On Optimal Control of a Class of Partially-Observed Discrete Event Systems

Hervé Marchand, Olivier Boivineau, Stéphane Lafortune

On Optimal Control of a Class of Partially-Observed Discrete Event Systems

Herv Marchand [a] , Olivier Boivineau [b], Stphane Lafortune [c]

[a]Inria Rennes, Campus Univ. de Beaulieu, 35042 Rennes, France

[b]Renault Technocentre 1, avenue du Golf 78288 Guyancourt, France

[c]Dept. of Elec. Eng. & Computer Science, Univ. of Michigan, 1301 Beal avenue, Ann Arbor, Michigan, USA

Abstract

We are interested in a new class of optimal control problems for Discrete Event Systems (DES). We adopt the formalism of supervisory control theory [10] and model the system as a ønite state machine (FSM). Our control problem is characterized by the presence of uncontrollable as well as unobservable events, the notion of occurrence and control costs for events and a worst-case objective function. We ørst derive an observer for the partially unobservable FSM, which allows us to construct an approximation of the unobservable trajectory costs. We then deøne the performance measure on this observer rather than on the original FSM itself. We then use the algorithm presented in [11] to synthesize an optimal submachine of the C-observer. This submachine leads to the desired supervisor for the system.

Key words: Discrete Event Systems, Supervisory Control Problem, Optimality, Partial Observation

1 Introduction and motivation

In general, the purpose of optimal control is to study the behavioral properties of a system (also called plant), modeled as a Discrete Event Systems (DES), to take advantage of a particular structure, and to generate a supervisor which constrains the system to a desired behavior according to quantitative and qualitative requirements. In the basic setup of supervisory control theory, optimality is with respect to set inclusion and thus all legal behaviors are equally good (zero cost) and illegal behaviors are equally bad (inønite cost). The work in [11] enriches this setup by the addition of quantitative measures in the form of occurrence and control cost functions and a worst-case objective function, to capture the fact that some legal behaviors are better than others. We are here interested in a new class of optimal control problems for DES. Compared to the work in [11,3,9], we wish to take into account partial observability. Several concepts and properties of the supervisory control problem under partial observation were studied in [2,7] among others. However, they only proposed a qualitative theory for the control of DESs. In [6], based on a notion of (un)desirable states, and a penalty cost when (un)desirable states can(not) be reached in the controlled system, the Optimal Control under Partial Observation problem is solved by reducing it to a particular class of optimal control with full observation. In this paper, we adopt a dicœerent strategy of optimization (i.e., cost formulation and computation) based on the work of [11]. The starting point of our solution is a FSM which represents the global behavior of a given system, including its unobservable dynamics. The ørst step is the derivation of an observer for the partially unobservable FSM, called a C-observer. This step is necessary since unobservable events alone cannot trigger a speciøc behavior of a controller. We deøne the performance measure on the C-observer rather than on the original FSM itself. However, we will make the necessary eœorts to keep track of the information that has disappeared with the initial structure. This observer allows us to remember an approximation of the unobservable costs between two observable events. This approximation corresponds to the worst, i.e., the highest, cost of the dicœerent unobservable trajectories than can occur between two observable events. In the second step, we use the theory in [11] to synthesize an optimal controller corresponding to the optimal restricted behavior, insofar as it is achievable by

* Corresponding author: H. Marchand. Fax: +332 99847171
Email addresses: hmarchan@irisa.fr (Herv Marchand), Olivier.Boivineau@renault.com (Olivier Boivineau), stephane@eecs.umich.edu (Stphane Lafortune).

an admissible (i.e., physically constructible) supervisor. We use back-propagation from the goal state to generate the supervisor, based on event cost functions. The supervisor is synthesized in a manner that gives them optimal sub-structure, consistent with the notion of DP-Optimality of [11]. Due to space limitations, proofs and examples of the results presented in this paper had to be omitted; they are available in [8].

2 Preliminaries

The system to be controlled is a øniteˆ state machine (FSM [4]) deøned by a 5-tuple $G = \langle \Sigma, Q, q_0, Q_m, \delta \rangle$, where Σ is the øniteˆ set of events, Q is the øniteˆ set of states, q_0 is the initial state, Q_m is the set of marked states, and δ is the partial transition function deøned over $\Sigma \times Q \longrightarrow Q$. The notation $\delta(\sigma, q)!$ means that $\delta(\sigma, q)$ is deøned, i.e., there is a transition labeled by event σ out of state q in machine G. Likewise, $\delta(s, q)$ denotes the state reached by taking the sequence of events deøned by trace s from state q in machine G. $\delta(q)$ denotes the active event set of x. The behavior of the system is described by the preøx-closed language $\mathcal{L}(G)$ [4], generated by G. Similarly, the language $\mathcal{L}_m(G)$ corresponds to the marked behavior of the FSM G, i.e., the set of trajectories of the system ending in $q_m \in Q_m$. An FSM is said to be blocking if $\mathcal{L}(G) \neq \overline{\mathcal{L}_m(G)}$ and non-blocking if $\mathcal{L}(G) = \overline{\mathcal{L}_m(G)}$, where $\overline{K}$ denotes the preøx closure of the language K. If G is blocking, it can reach a state q, where $\delta(q) = \emptyset$ but $q \notin Q_m$. This is called a deadlock because no event can be triggered. Another issue is when there is a set of unmarked states in G that forms a strongly connected component, but with no transition going out of the set. If the plant enters this set of states, then we get what is called a livelock. It is assumed that G is trim with respect to q_0 and Q_m (i.e. all the states of G are accessible from the initial state q_0 and co-accessible to one of the marked state $q_m \in Q_m$), which entails that G is non-blocking. We say that an FSM $A = \langle \Sigma_A, Q_A, q_{0A}, \delta_A \rangle$ is a submachine of G, denoted by $A \subseteq G$, if $\Sigma_A \subseteq \Sigma$, $Q_A \subseteq Q$, and $\forall \sigma \in \Sigma_A, q \in Q_A \;\; \delta_A(\sigma, q)! \Rightarrow (\delta_A(\sigma, q) = \delta(\sigma, q))$. It is clear that $\subseteq$ is a partial order on the set of FSMs. We also say that A is a submachine of G at q whenever $q_{0A} = q \in Q$ and $A \subseteq G$. Moreover, we will use $\mathcal{M}(G, q, q_m) = \{A \subseteq G : A \text{ is trim w.r.t. } q_m \in Q_m \text{ and } q_{0_A} = q\}$ to represent the set of trim submachines of G at q with respect to q_m. This set has a maximal element (in the sense that all others are submachines of it). It is denoted by $M(G, q, q_m)$. Later on in the paper, we will omit q_m in this notation (see Assumption 3). Moreover $\exists \sigma \in \delta_A(q_A)$ will mean that the event σ can be triggered in q_A and that $\delta_A(\sigma, q_A) \in Q_A$.

Following [10], we have to take into account the possibility that certain events cannot be disabled by the supervisor or that certain events may be not observed by the supervisor. Therefore, some of the events in Σ are said to be uncontrollable, i.e., their occurrence cannot be prevented by a controller, while the others are controllable. Likewise, control will be applied on a plant that is partially observable, i.e. the supervisor will observe only a subset of the events generated by plant G. Hence some of the events in Σ are observable whereas the others will be unobservable. An unobservable event could model a failure event, an internal event, etc. In this regard, Σ can be partitioned as $\Sigma = \Sigma_c \cup \Sigma_{uc}$ with $\Sigma_c \cap \Sigma_{uc} = \emptyset$ and as $\Sigma = \Sigma_o \cup \Sigma_{uo}$ with $\Sigma_o \cap \Sigma_{uo} = \emptyset$, where Σ_c, Σ_{uc}, Σ_o and Σ_{uo} represent the set of controllable, uncontrollable, observable and unobservable events, respectively. Moreover, we make the following assumption:

Assumption 1 Unobservable events are assumed to be uncontrollable: $\Sigma_{uo} \subseteq \Sigma_{uc}$ (this implies $\Sigma_c \subseteq \Sigma_o$).

This assumption allows us to directly abstract away the unobservable trajectories of the system and to work on the resulting system (due to this assumption, all these trajectories are uncontrollable). It also simpliøes the cost trajectory computation (see Section 4.2). Moreover, in order to consider the control problem under partial observation, we need to make sure that it can have a solution. If the initial FSM G has an unobservable cycle, even if it may be possible to determine its existence, it would be impossible to alleviate the fact that it could make the system run indeønitely in that cycle, without the supervisor noticing. Hence Assumption 2.

Assumption 2 G has no unobservable cycles.

Under this assumption we can show that $\forall q \in Q$, $\{s \in \Sigma_o \Sigma_{uo}^* \;/\; \delta(s, q)!\}$ has øniteˆ cardinality.

We now include the last ingredient to be able to discuss optimality, namely a cost (or objective) function. As stated in [11], two cost values are associated to each event of Σ. We ørst introduce an occurrence cost function $c_e : \Sigma \to \mathbb{R}^+$. Occurrence cost functions are used to model the cost incurred in executing an event (energy, time, etc.). This function can be easily extented to a trace $s = \sigma_1 \ldots \sigma_n$ as follows : $c_e(s) = \sum_{i=1}^{n} c_e(\sigma_i)$. To represent the fact that disabling a transition possibly incurs a cost, we introduce a control cost function $c_c : \Sigma \to \mathbb{R}^+ \cup \{0, \infty\}$. The control cost function is inønity for events in Σ_{uc}. These cost functions are used to introduce a cost on the trajectories of $A \subseteq G$.

Finally, based on remarks in [11], and because we want to have an algorithm that solves the Optimal Controller Synthesis Problem, with a polynomial complexity in the number of states of the system, we reduce the model to a unique marked state [1] in G. Hence, the assumption:

[1] Note that from a theoretical point of view, this assumption is not necessary, but simpliøes the presentation. Moreover, from a computational point of view, in the case of acyclic FSMs, this assumption can be relaxed (see [11], Section 6.2).

Assumption 3 The initial system has a unique marked state q_m, i.e., $Q_m = \{q_m\}$.

3 The C-Observer

The framework in which we develop our control theory is that of partially observable FSMs. The supervisor that will be generated should be able to take decisions based on the states and/or events that it observes. Consequently, we base our model upon a partially observed system, seen through an observer. However, in order to take into account unobservable events in the optimality under which we apply our control, we must keep track of their costs. The idea is to collect an approximation of the costs between two observable events in the states of the observer we want to build. For example, consider two states p and q of G, connected by (at least) a trace of the form $\sigma s \in \Sigma_o \Sigma_{uo}^*$. As we only observe the ørst event, it is not possible to know which trajectory has been taken between these two states. Hence, from an optimal control point of view, we have to consider that the plant evolves through the trajectory with the highest cost (there is no way to control the system in such a way that this trajectory is not taken). In order to collect these costs, we build a deterministic observer, named C-observer (Observer with Costs), and deøne the notion of a macro-state, allowing both to mask the underlying nondeterminism and to keep track of the unobservable event costs of trajectories between two states. The C-observer constitutes the basic model on which the optimal control will be applied.

Before formally giving the deønition of the C-observer, denoted by G_c, we need to check the original FSM G in order to account for unobservable events that may lead to q_m in G. Indeed, if an unobservable event leads to q_m in G, it may be impossible to determine whether or not the system has actually reached q_m.

Assumption 4 There exists in G a self-loop at q_m, labeled φ with $\delta(\varphi, q_m) = q_m$.

The φ event is just an (observable) indicator event (e.g., a sensor) that signals that q_m has been reached. Without loss of generality, we can assume it is controllable and has zero occurrence and control costs.

3.1 Deønition of the C-observer

The new structure that we deøne is called a C-observer. It is denoted by $G_c = \langle \Sigma_o, X, x_0, x_m, f \rangle$, where Σ_o is the set of observable events, X is the set of macro-states, x_0 is the initial macro-state, x_m is the marked macro-state, and f is the partial transition function deøned over $\Sigma_o^* \times X \longrightarrow X$. Starting from G, the set X of macro-states of G_c will be constituted of pairs in $Q \times \mathbb{R}^+$. More speciøcally, the admissible states that are considered are states that can be reached by a trace of events of the form $\Sigma_o \Sigma_{uo}^*$.

One way of looking at this choice of projection observation mask is that an observable action can lead to a sequence of unobservable events. If one initial observable event is taken, it is possible that several other events take place as a direct consequence without the possibility of observing them.

In order to formalize this idea, we introduce the set of triples $\mathcal{D}$ deøned by :

$$\mathcal{D} = \{(p, q, \sigma) \in Q \times Q \times \Sigma_o \ / \ \exists s \in \Sigma_{uo}^*,\ \delta(\sigma s, p) = q\}.$$

A triple (p, q, σ) belongs to set $\mathcal{D}$ if there is a trace between p and q whose ørst event is σ and whose following events are all unobservable. Note that more than one trace s could verify this condition. We now deøne the set of traces that verify the above conditions, for a given triple (p, q, σ) :

$$\forall (p, q, \sigma) \in \mathcal{D},\ \ \mathcal{S}(p, q, \sigma) = \{s \in \Sigma_{uo}^* \ / \ \delta(\sigma s, p) = q\}.$$

Knowing that $\forall q \in Q,\ \{s \in \Sigma_o \Sigma_{uo}^* / \delta(s, q)!\} < \infty$ (as pointed out in Section 2), we can easily prove that:

Proposition 1 $\forall (p, q, \sigma) \in \mathcal{D}, \mid \mathcal{S}(p, q, \sigma) \mid < \infty.$

We do not want to lose the cost of the unobservable events that have been projected. To this eœect we introduce the notion of locally computed cost associated with a triple (p, q, σ) of $\mathcal{D}$. Formally, it is given by a function, denoted by c_o, over $\mathcal{D} \rightarrow \mathbb{R}^+$, and deøned by:

$$\forall (p, q, \sigma) \in \mathcal{D},\ \ c_o(p, q, \sigma) = \max_{s \in \mathcal{S}(p,q,\sigma)} c_e(s). \qquad (1)$$

This way, we keep track of the worst unobservable trace that could lead from p to q. Using the previous notations, G_c is an FSM deøned as follows:

Deønition 2 Given an FSM G, the associated C-observer G_c is given by a tuple $\langle \Sigma_o, X, x_0, x_m, f \rangle$. It is an FSM whose elements are deøned as follows:

(1) A micro-state is a pair of $(q, c) \in Q \times \mathbb{R}^+$. A macro-state x is a set of micro-states, and X is the set of macro-states consisting of all (reachable) macro-states.
(2) The ønal macro-state is deøned by $x_m = (q_m, 0)$ and the initial macro-state x_0 as :

$$x_0 = \{(q, c_q),\ \exists s \in \Sigma_{uo}^*, \delta(s, q_0) = q \text{ and } c_q = \max_{t \in \Sigma_{uo}^*,\ \delta(t, q_0) = q} c_e(t)\}$$

(3) $\forall x \in X$, we deøne for any $(p, c_p) \in x$ and $\sigma \in \Sigma_o$, the set of Next micro-states $N_\sigma^x(p)$ [2] as

$$N_\sigma^x(p) = \{(q, c_o(p,q,\sigma)) \ / \ (p,q,\sigma) \in \mathcal{D}\}.$$

(4) The transition function f is recursively deøned by:

$$\forall x \in X \forall \sigma \in \Sigma_o, f(\sigma, x) = \{(q, c_q) \in \bigcup_{(p,c_p)\in x} N_\sigma^x(p), \ c_q = \max_{s \in \mathcal{S}(p,q,\sigma)} c_e(s)\}$$

(if two micro-states of the form $(q, .)$ in $f(\sigma, x)$, then we only consider the pair with the maximal cost)

(5) We only build the accessible part of G_c (i.e., the states $x \in X$ that are reachable from x_0 by f).

The way G_c is built masks the nondeterministic nature of the projected FSM. The initial macro-state x_0 is computed from the unobservable reach of $(q_0, 0)$. The ønal macro-state x_m contains the single micro-state, namely, $\{(q_m, 0)\}$. Finally, f can be constructed recursively from the initial state. Indeed, we can construct the set of states of G_c using item (2) and then items (3) and (4) of Deønition 2 recursively. Note that due to Proposition 1, the recursion terminates. The structure that we obtain is another deterministic FSM, whose events are in Σ_o. The states of G_c are macro-states w.r.t. G. What the above means is that the C-observer knows the system model G but only observes the events in Σ_o. It will start with x_0 as its estimate of the state of G. Upon observing $\sigma \in \Sigma_o$, the C-observer will update its state estimate to $f(\sigma, x_0)$, as this set represents all the states where G could be after executing the event σ followed by an unobservable trace; and so on after each observation. Moreover, we have computed and kept a local cost to avoid losing track of the costs of the unobservable events that have disappeared from the structure.

There exist standard algorithms for building observers, without cost memorization (see, e.g., [5]). Such algorithms are in general exponential in the number of states of the initial system. In our case, the cost memorization required in C-observers can be done on-the-Æy when building G_c without changing the complexity.

We now give a few lemmas and properties that hold for G_c, in order to use them in subsequent results.

Lemma 3 Let $x \in X \backslash \{x_m\}$ be a state of G_c, and let $(q, c_q) \in x$ be a micro-state of x. We can state that

(1) either $\exists \sigma \in \Sigma_o$, $\exists q' \in Q$, $\delta(\sigma, q) = q'$ and, in this case, $\exists x' \in X$, s.t. $f(\sigma, x) = x'$, and $(q', .) \in x'$

(2) or $\exists \sigma \in \Sigma_{uo}$ and $\exists q' \in Q$ s.t. $\delta(\sigma, q) = q'$ and, in this case, $\exists (q', .) \in x$.

Moreover, $\forall (q, c_q) \in x$, $\exists s \in \Sigma_{uo}^* \Sigma_o$, $\delta(s, q)!$.

What the above lemma states is that whatever the state x that can be reached during the execution of the plant, there eventually exists a way out of this state (either directly via an observable event or via an unobservable trajectory which reaches a micro-state of x having the previous property). Next, we state that the C-observer realized from G inherits properties of G.

Proposition 4 G_c is non-blocking.

3.2 Extented notion of controllability

In this section, we formalize the method used (by a supervisor) to generate a submachine from a C-observer.

Submachines of a C-observer. The machine G_c that we obtain is an FSM that simply reÆects what an observer sees in system G. We wish to apply some control to the original system in order to optimize a certain performance criterion. In other words, we wish to reduce the system G_c, and therefore G, to a particular behavior. This leads us to deøne the notion of a submachine of G_c. In fact, even if the domains in which G and G_c are deøned are dicœerent, the notion of submachine is the same as the one given in Section 2 (i.e. a submachine of G_c is any structure that has its states in those of G_c, the same initial state and ønal state, and its events and transitions in those of G_c). Moreover, we are only interested in complete behaviors, i.e., we wish to obtain a controlled system that reaches the state x_m in G_c and therefore the state q_m in G. Hence, we wish to consider the submachines of G_c that have this property. This leads us to the notion of G-live submachines.

Deønition 5 Let $G_c = \langle \Sigma_o, X, x_0, x_m, f \rangle$ be the C-observer associated with $G = \langle \Sigma, Q, q_0, q_m, \delta \rangle$. A submachine $H = \langle \Sigma_o, X_H, x_{0,H}, x_m, f_H \rangle$ of G_c is said to be G-live if the following condition holds:

$$\forall x_H \in X_H \setminus \{x_m\}, \ \forall (q, c_q) \in x_H, \ \exists (q', c_{q'}) \in x_H \ s.t. \ \{[\exists s \in \Sigma_{uo}^*, \delta(s, q) = q'] \wedge [\exists \sigma \in f_H(x_H), \ \delta(\sigma, q')!]\}.$$

A submachine H of G_c is G-live whenever any micro-state of x_H has a transition that is either an observable transition for the initial FSM G, or an unobservable transition that leads to another micro-state of x_H from which there is a possibility of exiting the macro-state (except for the marked state). Quite naturally, using Lemma 3, we can state that G_c is G-live [8].

Controllability in this framework. The structure on which control will be applied is FSM G_c. We ørst have

[2] $N_\sigma^x(p)$ basically constitutes the set of states of G that can be reached via a trace $\sigma\Sigma_{uo}^*$ (from a micro-state of x), together with the associated approximation of the unobservable trace cost.

to adapt the classical deønition of controllability introduced by [10]. Indeed, even if the control policy remains the same (we do not want to disable uncontrollable events), we have to take care of the fact that, by removing controllable transitions, the obtained submachine of G inherits some properties of the initial FSM G_c.

Deønition 6 Let $G_c = \langle \Sigma_o, X, x_0, x_m, f \rangle$ be the C-observer associated with $G = \langle \Sigma, Q, q_0, q_m, \delta \rangle$. $H = \langle \Sigma_o, X_H, x_{0,H}, x_m, f_H \rangle$ is said to be a controllable submachine of G_c if the following conditions hold:

(1) $\forall x_H \in X_H$ that can be reached via a trace of $\mathcal{L}(H)$, $\forall \, \mathrm{e} \in \Sigma_{uc} \cap \Sigma_o, \quad f(\mathrm{e}, x_H)! \Rightarrow f_H(\mathrm{e}, x_H)!$,
(2) H is G-live.

Condition (1) imposes that any transition that needs to be disabled in G_c to generate H needs to be controllable. Condition (2) imposes that no submachine of a C-observer presents any deadlocks or livelocks. This condition imposes that any micro-state of a state x_H must have an active outgoing trace (in the original FSM from which G_c was derived) that is either unobservable (leading to another micro-state of x_H and eventually leading to a state from which there is an observable outgoing event) or observable (leading to another macro-state).

The supervisor. Now that we have the deønition of a controllable submachine of a C-observer, it is interesting to determine how such a submachine can be obtained via a supervisor acting upon G_c. Control cannot be blindly performed. Disabling a controllable event that was admissible in a state x of G_c can induce a deadlock in the initial FSM G, even if there seems to be a transition out of the state. Hence, we introduce the notion of Admissible Control Actions (ACA).

Deønition 7 Let G_c be the C-observer associated with G. We deøne the set of Admissible Control Actions (ACA) at state $x \in X$ as a function :

$$\Gamma_x = \{\gamma \subseteq \Sigma_c, \ \forall (q, c_q) \in x, \ \exists (q', c_{q'}) \in x \ s.t. \ \{[\exists s \in \Sigma_{uo}^*, \ \delta(s, q) = q'] \wedge [\exists \, \mathrm{e} \in f(x) \setminus \gamma, \ \delta(\mathrm{e}, q')!]\}$$

More precisely, Γ_x gives, for a state x of G_c all the possible sets of controllable events that can be disabled without risk of deadlock. In other words, given a state x of G_c and given a γ in Γ_x, if σ belongs to γ, it means that σ can be disabled because there actually exists at least one trajectory $s \in \Sigma_{uo}^*$ that leads the system in another micro-state of x' for which there exists an observable event σ' that makes the system leave the macro-state x and eventually reach a state $x' = f(x, \sigma')$ of G_c. Using Deønition 7, a supervisor of G_c is deøned by:

Deønition 8 Let G_c be the C-observer associated with G and $\Pi = (\Gamma_x)_{x \in X}$ be the set of admissible control actions, then a supervisor S is a function from X into 2^{Σ_c} such that $S(x) = \gamma \in \Gamma_x$.

Hence, a supervisor of G_c is obtained by choosing a particular γ in a state x. By deønition, the control action will always act on events in Σ_c, which ensures that S never disables an uncontrollable event. Conceptually, the supervisor controlling the plant G is placed in feedback with G and G_c. Only the observable events can be seen by S. Therefore G_c plays the role of an observer that will somehow rebuild a part of the state in which the system has evolved. At the level of G, if G_c evolves into x, then the eœect of S will be to disable in G all the events $\sigma \in S(x)$ that are admissible in the states $q \in Q$, such that $(q, .) \in x$ (without creating deadlock in the controlled system). Let us now remark that Deønition 8 is consistent with the deønition of a controllable submachine of the C-observer G_c. This is summarized by the following proposition:

Proposition 9 $H = \langle \Sigma_o, X_H, x_{0,H}, x_{m,H}, f_H \rangle \subseteq G_c$ is a controllable submachine of G_c if and only if there exists a supervisor S, such that $\forall x_H \in X_H, \ f_H(x_H) = f(x_H) \setminus S(x_H)$. ◇

Let us now deøne the behavior of the controlled system:

Deønition 10 Given a Supervisor S of G_c and $H \subseteq G$ the associated controllable submachine of G_c, then the language generated by G under the control of S is given by $\mathcal{L}(S/G) = P_o^{-1}[\mathcal{L}(H)] \cap \mathcal{L}(G)$, while the marked language is given by $\mathcal{L}_m(S/G) = \mathcal{L}(S/G) \cap \mathcal{L}_m(G)$, where P_o (resp. P_o^{-1}) corresponds to the natural projection over the observable events (resp. the inverse projection of P_o over the alphabet).

With this deønition, we can state the following property, making the link between G and S/G.

Proposition 11 With the preceding notations, $K = \mathcal{L}(S/G)$ is controllable with respect to $\mathcal{L}(G)$ and Σ_c and observable[3] with respect to $\mathcal{L}(G)$, Σ_o, and P_o. Moreover $\underline{\mathcal{L}_m(S/G)}$ is $\mathcal{L}_m(G)$-closed (i.e. $\mathcal{L}_m(S/G) = \mathcal{L}_m(G) \cap \overline{\mathcal{L}_m(S/G)}$).

4 Optimal supervisory control problem

The aim of optimal control is to study the behavioral properties of a system, to take advantage of a particular structure, and to generate a controller which constrains the system to a desired behavior according to quantitative and qualitative aspects [6,9,11]. This is performed by the addition of quantitative measures in the form of

[3] See the formal deønition in [4], Section 3.7 or in [10].

occurrence and control cost functions, to capture the fact that some legal behaviors are better than others.

4.1 Transformation of G_c

We need to transform the C-observer, in order to exactly øt within the framework developed by [11]. Indeed, unlike in the case of total observability where costs are deøned on events only, we have incorporated cost information in the macro-states of G_c. These costs were attached to the states in order to keep track of the unobservable cost of the trajectory between two macro-states (see Section 3.1). Basically, the transformation we will perform on G_c, consists in ıshiftingȷ the cost of the macro-state to the events that can be executed in this macro-state. However, we do not blindly take the worst cost of the micro-states contained in the macro-state. For a given x, and a given σ admissible in x, we consider the worst cost of the pairs $(q, c_q) \in x$ such that σ belongs to the active event set of q in G. The transformation is performed as follows. Let $x \in X$ and let $f(x)$ be the set of events that G_c can execute in x. For each $\sigma \in f(x)$, we rename σ as σ_x and we attach to this new event the cost $c(\sigma_x)$:

$$c(\sigma_x) = \max_{(q, c_q) \in x, \delta(\sigma, q)!} \{c_q\} + c_e(\sigma) \qquad (2)$$

The controllability status of the event as well as the control cost of the events do not change (namely, we have $c_c(\sigma_x) = c_c(\sigma)$). Call Σ'_o the new set of event. The transition function f remains the same (i.e., $f(x, \sigma_x)$ is deøned and equal to x' whenever $x' = f(x, \sigma)$).

The new C-observer G'_c we obtain is still an FSM. It is deøned by $\langle \Sigma', X, x_0, x_m, f \rangle$. Compared to G_c, the global structure of G'_c does not change. The only dicœerence is that we changed the original set of events of G_c in such a way that costs are now deøned on events only, as carried out in [11]. From now on, G'_c is a deterministic and trim FSM. To each event is attached two values, which respectively correspond to its event and control costs. The only dicœerence with [11] lies in the notion of controllability that, in our framework, takes into account the notion of liveness of the underlying system G. However, this does not acœect the use of the theory of [11] to compute the optimal supervisor of G_c, and therefore the optimal supervisor of G. Indeed, as in our case, the theory is based on the notion of acceptable control actions that have to be computed at ørst. In [11], a control action in a state x is admissible whenever it does not disable uncontrollable events and it does not produce local deadlock (i.e., no output event.)

Remark 12 Note that given G_c, the way we are shifting the costs of the unobservable trajectories in order to obtain G'_c constitutes the best approximation that we can do without memory. A better one could be obtained by unfolding the C-observer G_c in order to take into account the history of the plant (e.g., the last n observable events that occurred in the system). However, even if the approximation would be better (at each step, the number of admissible pairs (q, c_q) in a macro-state would have been lower leading to a possibly lower unobservable cost), the counterpart would be the complexity of G'_c.

4.2 Trajectory costs of a submachine of G'_c

In order to be able to discuss optimality, we now explain how to compute the cost of a trajectory of G'_c.

Control cost function over the states. In order to model this particular aspect, let us deøne the control cost of an event according to a state. We ørst introduce $\Sigma_d(x, H) = f_{G'_c}(x) \setminus f_H(x)$ as the set of disabled events at state x for the system to remain in submachine H of G'_c. Whereas in [11] the control cost function was deøned on an event, in the case of partial observation, it is deøned on a state as follows: considering a submachine H of G'_c, we have

$$C_c(x, H) = \begin{cases} \infty \text{ if } \Sigma_d(H, x) \notin \Gamma_x \\ \sum_{\sigma' \in \Sigma_d(H,x)} c_c(\sigma') \text{ otherwise} \end{cases} \qquad (3)$$

The control cost of a state x is equal to ∞ whenever there does not exist a particular control policy $\gamma \in \Gamma_x$ that restricts the behavior of G'_c to H (i.e. when an uncontrollable event has been removed or when a removal of a controllable event induces a deadlock).

The global cost of a submachine of G'_c. We are now ready to deøne the cost of a trajectory s of a submachine H and the objective cost function of $H \subseteq G'_c$.

Deønition 13 Let $H = \langle \Sigma'_o, X_H, x_{0,H}, x_{m,H}, f_H \rangle$ be a submachine of G'_c derived from G and $\mathcal{L}_m(H)$ be the marked language generated by H, then

(1) for all y in H and trajectory $s = \sigma'_1 ... \sigma'_n, \forall i, 1 \leq i \leq n, \sigma'_i \in \Sigma'_o$ such that $f_H(y, s)$ exists, the cost of s is given by :

$$C_O(y, H, s) = \sum_{i=1}^{n} c(\sigma'_i) + \sum_{i=0}^{n} C_c(f_H(y, \|s\|_i), H), \qquad (4)$$

where $\|s\|_i$ denotes the preøx of s of length i,

(2) the objective cost function denoted by $C_{Sup}(H)$ is given by:

$$C_{Sup}(H) = \sup_{s \in \mathcal{L}_m(H)} (C_O(x_0, H, s)) \qquad (5)$$

The cost of a trajectory is the sum of the occurrence costs of the events composing it, to which is added the

cost of controlling events on the way to remain in machine H. If an uncontrollable event is disabled, the cost of a trajectory becomes inønite because of the second term of (4). Finally, $C_{Sup}(H)$ represents the worst case behavior that is possible in submachine H. Note that the purpose of ıcontracting a submachineȷ is to remove trajectories with high event costs. However this process is accompanied by rising control costs, hence the tradeoœ in the optimization problem we now deøne.

4.3 The optimization problem

We are only interested in machines that achieve a task (we only consider plants having a behavior which terminates at a marked state). We want to extract the submachines that have a minimal cost function among all the trim and controllable submachines of G'_c.

Deønition 14 $\forall x \in X, H_o \in \mathcal{M}(G'_c, x)$ is an optimal submachine of FSM G'_c if $C_{Sup}(H_o) = \min_{H \in \mathcal{M}(G'_c,x)} C_{Sup}(H) < \infty$.

The cost $C_{Sup}(H_o)$ of H_o represents the minimum worst case cost incurred to reach x_m from x_0 when the behavior of G'_c is restricted to a submachine of it. As some events in some states are not controllable (which induces an inønite cost), optimality is met when there is no other control policy with lower worst-case cost that allows to reach the marked state x_m with certainty. At a lower level (in the world of G), the control policy induced by submachine H_o corresponds to the one with lowest worst-case cost, knowing that G could evolve through unobservable trajectories with the worst possible cost. However the way we compute the cost of trajectories by taking into account an upper approximation of the unobservable trajectories (see Section 4.1 and Deønition 13) reduces the uncertainty; we do not consider all the unobservable trajectories but only the ones that are admissible knowing that a particular event is executed. The following theorem gives necessary and suŒcient conditions for the existence of optimal submachines:

Theorem 15 An optimal submachine of G'_c exists if and only if there exists a submachine H of G'_c such that H is trim, controllable, with no cycles. ◇

Intuitively, this theorem states that an optimal solution exists when there are acyclic controllable submachines of G'_c. The controllability assumption ensures that the cycles can be broken using controllable events alone. The optimal submachine that includes all the other optimal submachines will be denoted by ${H^\uparrow}_o$ (see [11]).

Usually, the solution to the optimal supervisory control problem is not unique. Moreover, all the optimal solutions do not structurally have optimal sub-solutions, which means that they do not satisfy the principle of Dynamic Programming (see e.g., [1]). In fact, in the previous paragraph, optimality is obtained only regarding the paths between the initial and the ønal state, and never the postøx paths between any state of the corresponding FSM and the ønal state. In this section, we will show that whenever an optimal solution exists, a solution having optimal sub-structure also exists. We call this latter type a DP-optimal solution (DP stands for Dynamical Programming) and deøne it as follows :

Deønition 16 A submachine H_{DO} of G'_c is DP-optimal if it is optimal and $\forall x' \in X_{H_{DO}}$, $M(H_{DO}, x')$ is an optimal submachine in $\mathcal{M}(G'_c, x')$.

DP-Optimality is then obtained when any terminal path from any state of a submachine to the goal state x_m is optimal in the previous sense. If a particular DP-Optimal FSM includes all other DP-Optimal FSMs as submachines of itself, then we call it the maximal DP-Optimal submachine. The maximal DP-Optimal submachine of a machine G'_c at q w.r.t. x_m will be denoted by $M^o_D(G'_c, x)$. The existence of a DP-Optimal submachine of G'_c is given by the following theorem.

Theorem 17 [11] If an optimal submachine of G'_c exists, then the unique maximal DP-Optimal submachine $M^o_D(G'_c, x_0)$ of G w.r.t. x_m also exists.

The DP-Optimal algorithm. Consider an FSM $G = \langle \Sigma, Q, q_0, q_m, \delta \rangle$ and its corresponding transformed C-observer $G'_c = \langle \Sigma'_o, X, x_0, x_m, f \rangle$. Then there exists an algorithm [11], named DP-Opt, with a worst-case complexity $\mathcal{O}(|X|^2|\Sigma_o|log(|\Sigma_o|) + |X|^3|\Sigma_o|)$ (Theorem 6.10 of [11]) [4], that constructs the desired maximal DP-Optimal submachine of the FSM G'_c w.r.t. x_0 and x_m. We refer the reader to [11] for a complete description of DP-Opt. Since the number of states of G_c is in the worst case exponential in the number of states of G, the real complexity of the DP-Opt algorithm that gives access to an optimal supervisor under partial observation is indeed exponential in the number of states of the initial system (but polynomial if we only consider the C-observer).

4.4 The supervisor

The supervisor computation consists of dioerent steps. Once the C-observer G_c derived from the initial FSM G is computed, we ørst have to transform it into G'_c by attaching the cost induced by the unobservable trajectories to the events in order to øt within the framework of [11] (see Section 4.1). From this machine, using the algorithm of [11], we compute (if it exists) the DP-Optimal solution $M^o_D(G'_c, x_0)$ of G'_c. At this point, we disable in G_c the corresponding sets of events in Σ'_o and for all $x \in X$, we retrieve $\Sigma_d(G_c, x)$, the set of disabled events at state

[4] Note that we have $|\Sigma_o|$ and not $|\Sigma'_o|$ in this expression, because we only have to account for the number of admissible events in a state x, which is bounded by $|\Sigma_o|$.

x for the system to remain in submachine $M_D^o(G_c, x_0)$ of G_c. Call f_c the new transition function, given by :

$$f_c : X \times \Sigma_o \to X$$
$$(x, \sigma) \mapsto \begin{cases} f(x, \sigma) \text{ if it is deøned and if } \sigma \notin \Sigma_d(G_c, x) \\ \text{undeøned otherwise} \end{cases}$$

Now, a supervisor S of G_c can be derived from $M_D^o(G_c, x_0)$ by attaching to this FSM an output function O, that for a given state x delivers the set of disabled events $\Sigma_d(G_c, x)$. The supervisor $S = \langle \Sigma_o, X, x_o, x_m, f_c, O \rangle$ will in fact be used for two purposes. It ørst plays the role of an observer that is able to partly rebuild the state in which the system has evolved. Based on this information, S sends back to the system the set of events that have to be disabled in order to force the closed loop system to eventually reach the marked state q_m by minimizing the global cost of the trajectory.

Optimality of S. Proposition 11 tells us that the closed-loop language $\mathcal{L}(S/G)$ lies in the class of controllable, and observable sublanguage of $\mathcal{L}(G)$. Note also that Assumption 1 implies that $\mathcal{L}(S/G)$ is also normal (c.f. [4]). We cannot however compare $\mathcal{L}(S/G)$ with the optimal language $\mathcal{L}(S_{full\ obs}/G)$ that would have been computed by the DP-Opt algorithm under the assumption that $\Sigma_o = \Sigma$. Such comparisons are meaningless when $\Sigma_o \neq \Sigma$, since a partial observation supervisor can only react upon the occurrence of observable events. To characterize the optimality properties of S we must look at observer-based supervisors and projected languages. The results is [11] tells us that $M_D^o(G_c', x_0)$ is the maximal DP-optimal of G_c', and consequently $M_D^o(G_c, x_0)$ deøned above is the maximal DP-optimal submachine of G_c. Moreover, $P_o[\mathcal{L}(S/G)]$ is the maximal optimal sublanguage of $P_o[\mathcal{L}(G)]$ in the context of the language formulation of the optimal control problem (see [11] for further details). By formulating the optimization problem over the submachines of G_c, we are eœectively requiring that the corresponding supervisors (that implement the solutions) be ımemorylessȷ, i.e., that they be based on the states of the observer of G. Therefore, S is an optimal supervisor in the class of observer-based supervisors. In other words, any other supervisor S' as deøned by (5) would correspond to a submachine H of G_c and therefore it would necessarily induce a worst case cost higher or equal to that induced by $M_D^o(G_c, x_0)$. It remains an open question to determine if by allowing supervisors with memory, i.e., based on a øner state space than that of G_c. c.f. Remark 3), we could obtain controllable and observable sublanguages of $\mathcal{L}(G)$ with a lower worst case cost than that of the above $\mathcal{L}(S/G)$.

5 Conclusion

In this paper, we have introduced a new type of optimal control for discrete event systems. Previous works in optimal control dealt with numerical performances in supervisory control theory when the system to be controlled is under full observation. In contrast, our aim was to account for partial observability, while controlling the system. The system to be controlled is represented by an FSM G with a unique marked state representing the state of interest (the achievement of a task for example) and some unobservable events. The ørst step is the derivation of an observer for the partially unobservable FSM, called a C-observer G_c, which allows us to remember an approximation of the unobservable trajectory costs. We then presented a new deønition of controllability derived from the classical one introduced by Ramadge & Wonham [10], that allows us to avoid the unobservable blocking of G. We then deøned the performance measure on this observer rather than on the FSM itself. In the second step, we ørst transformed G_c into G_c' by shifting the cost of each macro-state to the events that can be executed in this macro-state. We then used the algorithm presented in [11] to synthesize an optimal submachine of the C-observer, which led to the desired supervisor for the system. The behavior of the obtained controlled system is optimal with respect to Σ_o, in the sense that G_c carries on the best approximation of the unobservable trajectories between two observable events (without observing the whole system, it is not possible to have more information about these trajectories). Moreover it is optimal for G_c and therefore for G as explained in Section 4.4. This optimality notion is due to [11].

Acknowledgements

We would like to thank the reviewers for the comments and suggestions that helped to improve the paper. The research of the third author is supported in part by NFS grant CCR-0082784 and by the DDR&E MURI on Low Energy Electronics Design for Mobile Platforms managed by ARO under grant DAAH04-96-1-0377.

References

[1] R. Bellman. Dynamic programming. Princeton University Press, 1957.

[2] R. D. Brandt, V. K. Garg, R. Kumar, F. Lin, S. I. Marcus, and W. M. Wonham. Formulas for calculating supremal and normal sublanguages. Systems and Control Letters, 15(8):111˘117, 1990.

[3] Y. Brave and Heymann M. On optimal attraction in discrete-event processes. Information Sciences, 67:245˘276, 1993.

[4] C. Cassandras and S. Lafortune. Introduction to Discrete Event Systems. Kluwer Academic Publishers, 1999.

[5] J. E. Hopcroft and J. D. Ullman. Introduction to automata theory, Languages, and computation. Addison-wesley, Reading, MA., 1979.

[6] R. Kumar and V. Garg. Optimal supervisory control of discrete event dynamical systems. SIAM Journal of Control and Optimization, 33(2):419˘439, March 1995.

[7] F. Lin and W. M. Wonham. On observability of discrete˘ event systems. Information Sciences, 44(3):173˘198, 1988.

[8] H Marchand, O. Boivineau, and Lafortune S. Optimal control of discrete event systems under partial observation. Technical Report 1359, IRISA, November 2000. Available via WWW from http://www.irisa.fr/bibli/publi/pi/2000/1359/1359.html.

[9] K. M. Passino and P. J. Antsaklis. On the optimal control of discrete event systems. In Proc. of 28th Conf. Decision and Control, pages 2713˘2718, Tampa, Floride, December 1989.

[10] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. Proceedings of the IEEE; Special issue on Dynamics of Discrete Event Systems, 77(1):81˘98, 1989.

[11] R. Sengupta and S. Lafortune. An optimal control theory for discrete event systems. SIAM Journal on Control and Optimization, 36(2), March 1998.