Optimal control of discrete event systems under partial observation

Hervé Marchand, Olivier Boivineau, Stéphane Lafortune

Proceedings of the 40th IEEE
Conference on Decision and Control
Orlando, Florida USA, December 2001

Optimal Control of Discrete Event Systems under Partial Observation

Hervé Marchand[†], Olivier Boivineau[‡], Stéphane Lafortune[**]

[†] Inria Rennes, Campus Univ. de Beaulieu, 35042 Rennes, France, E-mail: hmarchan@irisa.fr
[‡] Electricité de France, Div. R & D. 1, avenue du Gal De Gaulle 92141 Clamart Cedex - France
[**] Dept. of Elec. Eng. & Computer Science, Univ. of Michigan,
1301 Beal avenue, Ann Arbor, Michigan, USA 48109-2122. E-mail: stephane@eecs.umich.edu

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 [7] and model the system as a finite 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 first derive an observer for the partially unobservable FSM, which allows us to construct an approximation of the unobservable trajectory costs. We define the performance measure on this observer rather than on the original FSM itself. Further, we use the algorithm of [8] to synthesize an optimal submachine of the observer. This submachine leads to the desired supervisor for the system.

1 Introduction and Motivation

We are interested in a new class of optimal control problems for Discrete Event Systems (DES) [7]. The system to be controlled is modeled as a finite state machine (FSM). Our control problem follows the theory in [8] and is characterized by the presence of uncontrollable events, the notion of occurrence and control costs for events and a worst-case objective function. However, compared to the work in [8] and compared to [3, 6], we wish to take into account partial observability. Several concepts and properties of the supervisory control problem under partial observation were studied in [1, 4] among others. However, they only propose a qualitative theory for the control of DESs.

The starting point of our solution is a FSM which represents the global behavior of a given system, including its unobservable dynamics. The first 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 specific behavior of a controller. We define the performance measure on the C-observer rather than on the original FSM itself. However, we will make the necessary efforts 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 different unobservable trajectories than can occur between two observable events. In the second step, we use the theory in [8] to synthesize an optimal controller corresponding to the optimal restricted behavior, insofar as it is achievable by 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 [8].

2 Preliminaries

The system to be controlled is modeled as a FSM defined by a 5-tuple $G = \langle \Sigma, Q, q_0, q_m, \delta \rangle$, where Σ is the set of events, Q is the (finite) set of states, q_0 is the initial state, q_m is the unique marked state, and δ is the partial transition function defined on $\Sigma^* \times Q$. The behavior of the system is described by the prefix-closed language $\mathcal{L}(G)$ [2], 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. Some of the events in Σ are 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. In this regard, Σ can be partitioned as $\Sigma = \Sigma_c \cup \Sigma_{uc}$ with $\Sigma_c \cap \Sigma_{uc} = \emptyset$ and $\Sigma = \Sigma_o \cup \Sigma_{uo}$ 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, unobservable events are assumed to be uncontrollable, i.e., $\Sigma_{uo} \subseteq \Sigma_{uc}$. In the sequel, we will only be interested in *trim* FSMs, i.e., FSMs for which all states of Q are accessible from q_0 and coaccessible to q_m [2]. We say that FSM $A = \langle \Sigma_A, Q_A, q_{0A}, q_m, \delta_A \rangle$ is a submachine of G, denoted $A \subseteq G$, if $\Sigma_A \subseteq \Sigma$, $Q_A \subseteq Q$, $\forall \sigma \in \Sigma_A, q \in Q_A \;\; \delta_A(\sigma, q)! \Rightarrow (\delta_A(\sigma, q) = \delta(\sigma, q))$ (The notation $\delta_A(\sigma, q)!$ means that $\delta_A(\sigma, q)$ is defined, i.e., there is a transition labeled by event σ out of state q in A). We say that A is a submachine of G at q whenever $q_{0A} = q \in Q$ and $A \subseteq G$. For any $q \in Q$, we will use $\mathcal{M}(G, q) = \{A : A$ is a trim submachine of G with respect to q_m 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 the maximal element contains all other elements as submachines. It is denoted by $M(G, q)$. In order to consider the control problem under partial observation, we need to make sure that the initial FSM G has no unobservable cycle. Otherwise it would be impossible to alleviate the fact that it could make the system run indefinitely in that cycle, without the supervisor noticing. We then assume that G has no unobservable cycles.

Finally, to take into account the numerical aspect of the optimal control problem, two cost values are associated to each event of Σ. We introduce an occurrence cost function $c_e : \Sigma \to \mathbb{R}^+$ and a control cost function $c_c : \Sigma \to \mathbb{R}^+ \cup \{0, \infty\}$. Control costs are used to represent the fact that disabling a transition possibly incurs a cost. The control cost function is infinity for events in Σ_{uc}. The cost functions are then used to introduce a cost on the trajectories of a submachine of G.

3 The C-Observer with respect to Σ_{uo}

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 first event, it is not be 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 define the notion of a macro-state, allowing both to mask the underlying nondeterminism by abstracting away from the nondeterministic submachine 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 giving formally the definition 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. We therefore update G by adding 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 The C-Observer definition

The new structure that we define 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 defined 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 specifically, the admissible states that are considered are states that can be reached by a trace of events constituted by an observable first event followed by a sequence of unobservable events. In language formalism, the latter trace should be in $\Sigma_o \Sigma_{uo}^*$. We present more formally the way the states of the system G_c are built. First, we introduce the set of triples $\mathcal{D}$ defined by :

$$\mathcal{D} = \{(p, q, \sigma) \in Q \times Q \times \Sigma_o \;/\; \exists s \in \Sigma_{uo}^*,\ \delta(\sigma s, p) = q\}. \quad (1)$$

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

$$\forall (p, q, \sigma) \in \mathcal{D},\ \ \mathcal{S}(p, q, \sigma) = \{s \in \Sigma_{uo}^* \;/\; \delta(\sigma s, p) = q\}. \quad (2)$$

Using (2), we can easily deduce the following property:

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

Finally, we do not want to lose the cost of the unobservable events that have been projected. To this effect 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 $\mathcal{D} \to \mathbb{R}^+$, and defined by:

$$\forall (p, q, \sigma) \in \mathcal{D},\ \ c_o(p, q, \sigma) = \max_{s \in \mathcal{S}(p,q,\sigma)} c_e(s) \qquad (3)$$

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

Definition 1 *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 defined as follows:*

1. *X is the set of macro-states. $x \in X$ is defined by a set of pairs $(q, c) \in Q \times \mathbb{R}^+$, called micro-states;*

2. *The final macro-state is defined by $x_m = \{(q_m, 0)\}$ and the initial macro-state x_0 as :*

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

3. $\forall x \in X$ and $\forall \sigma \in \Sigma_o$, *define*

$$\forall (p, c_p) \in x,\ A^x_\sigma(p) = \{(q, c_o(p, q, \sigma)) \ / \ (p, q, \sigma) \in \mathcal{D}\}.$$

$A^x_\sigma(p)$ 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.

4. *The transition function f is recursively defined by:*

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

Hence, if there exists different 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 the system (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. x_0 is computed from the unobservable reach of $(q_0, 0)$. x_m is a single marked 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 point *(2)* and then point *(3)* and *(4)* of Definition 1 recursively. Note that due to Property 1, the recursion terminates. The structure that we obtain is another deterministic FSM, whose events are taken in Σ_o. States of G_c are macro-states with respect to G. However, 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.

Lemma 1 **[5]** *Let $x \in X - \{x_m\}$ 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 by 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 1 **[5]** *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. We wish to apply some control to the original system in order to verify 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 define the notion of a submachine of G_c. In fact, even if the worlds in which they are defined (for G and G_c) are different, 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 final state and its events and transitions in those of G_c).

Moreover, we are only interested in complete behavior, i.e. we wish to obtain a controlled system that reaches the state x_m and therefore the state q_m. Hence, we wish to consider the submachine of G_c that have this property. Hence the notion of *G-live* submachines.

Definition 2 *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 1, we can state that :

Proposition 2 **[5]** *If G_c is the C-observer associated with G, then G_c is G-live.*

Controllability in this framework. The structure on which control will be applied is FSM G_c. We first have to adapt the classical definition of controllability introduced by [7]. 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 $= G_c$. Hence the new definition of controllability:

Definition 3 *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 $\mathcal{L}(H)$, $\forall \sigma \in \Sigma_{uc} \cap \Sigma_o,\ \ f(\sigma, x_H)! \Rightarrow f_H(\sigma, x_H)!$,*

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 (thereby leading to another micro-state of x_H and eventually leading to a state from which there is an observable outgoing event) or observable (thereby leading to another macro-state of H).

The supervisor. Now that we have the definition 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. However, control cannot be blindly performed. Disabling an event that was admissible in a state x of G_c can induce a deadlock in the initial FSM G. Hence, we introduce the notion of *Admissible Control Actions (ACA)*.

Definition 4 *Let $G_c = \langle \Sigma_o, X, x_0, x_m, f\rangle$ be the C-observer associated with $G = \langle \Sigma, Q, q_0, q_m, \delta\rangle$. We define the set of* Admissible Control Actions (ACA) *at state $x \in X$ as a function :*

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

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 a given 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 Definition 4, a *supervisor* of G_c is defined by:

Definition 5 *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 given by :*

$$\begin{aligned} S : X &\rightarrow 2^{\Sigma_c} \\ x &\mapsto \gamma \in \Gamma_x \end{aligned} \tag{4}$$

In other words, a supervisor of G_c is obtained by choosing a particular γ in a state x. By definition, the control action will always belongs to 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. According to this information, the supervisor determines whether the observation corresponds to a (conditionally) controllable event and if it has to enable/disable this event in order to keep the closed loop system behaving "desirably".

To conclude this section, let us remark that Definition 5 is consistent with the definition of a controllable submachine of the C-observer G_c. This is summarized by the following proposition:

Proposition 3 *$H \subseteq G_c$ is a controllable submachine of G_c if and only if there exists a supervisor S, such that $\forall x_H \in X_H,\ f_H(x_H) = f(x_H) \setminus S(x_H)$.*

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 [3, 6, 8]. This is performed by the addition of quantitative measures in the form of occurrence and control cost functions, to capture the fact that some legal behaviors are better than others.

4.1 Transformation of G_c

We first need to transform the C-observer, in order to exactly fit within the framework developed by [8]. Indeed, unlike in the case of total observability where costs are defined in events only, we have incorporated cost information in the macro-states of the 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 "shifting" the cost of the macro-state to the events that can be executed in this 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 $c_e(\sigma_x)$ defined by :

$$c_e(\sigma_x) = \max_{(q, c_q) \in x, \delta(\sigma, q)!} \{c_q\} + c_e(\sigma) \tag{5}$$

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 defined and equal to x' whenever $x' = f(x, \sigma)$).

The new C-observer G'_c we obtain is still a FSM. It is defined by $\langle \Sigma'_o, X, x_0, x_m, f\rangle$. Compared to G_c, the global structure of G'_c does not change. The only difference is that we change the original alphabet of G_c in such a way that costs are now defined on events only, as carried out in [8]. 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 difference with [8] 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 affect the use of the theory of [8] 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 first. In [8], 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.)

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 define the control cost of an event according to a state. We first 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 [8] the control cost function was defined on an event, in the case of partial observation, it is defined on a state as follows: considering a submachine H of G'_c, we have

$$C_c(x, H) = \begin{cases} \infty \text{ if } \Sigma_d(H, x) \notin \Gamma_x \\ \sum_{\sigma' \in \Sigma_d(H,x)} c_c(\sigma') \text{ otherwise} \end{cases} \tag{6}$$

The 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 controllable event has been removed, then inducing a deadlock).

Cost of a trajectory and of a submachine of G'_c. We are now ready to define the cost of a trajectory s of a submachine H as well as the objective cost function of a submachine H of G'_c.

Definition 6 *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 :*

$$C_O(y, H, s) = \sum_{i=1}^{n} c_e(\sigma_i) + \sum_{i=0}^{n} C_c(f_H(y, \|s\|_i), H), \tag{7}$$

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

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

$$C_{Sup}(H) = \sup_{s \in \mathcal{L}_m(H)} (C_O(x_0, H, s)) \tag{8}$$

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 infinite because of the second term of (7). Finally, $C_{Sup}(H)$ represents the worst case behavior that is possible in submachine H. The next lemma characterizes the interaction of event and control costs:

Lemma 2 [5] *Let $H_1 \subseteq H_2 \in \mathcal{M}(G'_c, x)$ and $s \in \mathcal{L}_m(H_1)$, then $C_O(x, H_1, s) \geq C_O(x, H_2, s)$.*

This lemma states that the cost associated with a trajectory admissible in a machine is lower than the cost of the same trajectory generated by one of its submachines. The purpose of "contracting a submachine" is to remove trajectories with high event costs. However this process is accompanied by rising control costs, hence the optimization problem we now define.

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). Among all the trim and controllable submachines of G'_c, since we want to deal with optimal solutions, we want to extract the submachines that have a minimal objective cost function.

The optimal submachines of G'_c. Considering the trim hypothesis, we denote $\mathcal{M}(G'_c, x)$ as the set of trim submachines of G'_c starting at state x with respect to the unique final state x_m and denote by $M(G'_c, x)$ its maximal element (see Section 2). We now define the optimization problem.

Definition 7 *$\forall x \in X, H_o \in \mathcal{M}(G'_c, x)$ is an optimal submachine of the 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 infinite 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 certainly. At a lower level (in the world of G), the control policy induced by submachine H_o corresponds to the one with lower worst-case cost, knowing that G could evolve through unobservable trajectories with the worst possible cost. In general, there will exists several optimal submachines for an FSM.

As in the case of total observation [8], the following lemma is stated to note that optimal solutions lie within the class of controllable submachines.

Lemma 3 [5] *Let $H \in \mathcal{M}(G, x)$. If $C_{Sup}(H) < \infty$ then H is controllable.*

From Lemma 3, uncontrollable submachines are not candidates for optimality since the cost for restricting the system to those submachines is infinite. The following theorem gives necessary and sufficient conditions for the existence of optimal submachines:

Theorem 1 [5] *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 controllable submachines of G'_c in which there does not exist cycles. The controllability assumption ensures that the cycles can be broken using controllable events alone. The submachine that includes all the other optimal submachines will be called the maximal optimal submachine and will be denoted by $H^{\uparrow}{}_o$.

The DP-optimal submachines of G'_c. In general, 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. In fact, in the previous section, optimality is obtained only regarding the paths between the initial and final state and never the postfix paths between any state of the corresponding FSM and the final 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 define it as follows :

Definition 8 *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 $\mathcal{M}(G'_c, x')$.*

We have already seen that optimality actually exists when the worst-case cost from the initial state x_0 is to x_m is finite once minimized. DP-Optimality is 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 $M^o_D(G'_c, x)$. Note that all DP-Optimal submachines are acyclic. The existence of a DP-Optimal submachine of G'_c is given by the following theorem (the proof can be found in [8]).

Theorem 2 *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 a FSM $G = \langle \Sigma, Q, q_0, q_m, \delta\rangle$ with a unique initial state q_0, and a unique marked state q_m and its corresponding transformed C-observer $G'_c = \langle \Sigma'_o, X, x_0, x_m, f\rangle$. Then there exists an algorithm [8], 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 [8]), that constructs the desired maximal DP-Optimal submachine $M^o_D(G'_c, x_0)$ of the FSM G'_c w.r.t. x_0 and x_m. The algorithm also returns the worst inevitable cost $c^g_{sup}(M^o_D(G'_c, x_0))$. We refer the reader to [8] for a complete description of **DP-Opt**.

4.4 The supervisor

The supervisor computation consists of different steps. Once the C-observer G_c derived from the initial FSM G is computed, we first have to transform it into G'_c by attaching the cost induced by the unobservable trajectories to the events in order to fit within the framework of [8] (see Section 4.1). From this machine, using the algorithm of [8], 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 event at state x for the system to remain in submachine $M^o_D(G_c, x_0)$ of G_c. Call f_c the new transition function. It is formally given by :

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

Now, a supervisor S of G_c can be derived from $M^o_D(G_c, x_0)$ by attaching to this FSM an output function O that for a given states 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 first plays the role of an observer that is able to rebuild part of 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.

5 Conclusion

In this paper, we have introduced a new type of optimal control for DESs by adding the notion of partial observation. The system to be controlled is represented by an FSM G with a unique marked state and some unobservable events. The first step was the derivation of a C-observer G_c from the partially unobservable FSM, which allows us to remember an approximation of the unobservable trajectory costs. We then presented a new definition of controllability derived from the classical one introduced by [7], that allows us to avoid the blocking of G without observing it. We then define the performance measure on this observer rather than on the FSM itself. In the second step, we first transform G_c into G'_c by shifting the cost of the macro-state to the events that can be executed in this macro-state. We then use the algorithm presented in [8] to synthesize an optimal submachine of the C-observer, which leads to the desired supervisor for the system. The behavior of the obtained controlled system is optimal w.r.t. Σ_o, in the sense that G_c carries on the best approximation of the unobservable trajectories. Moreover it is optimal for G'_c and therefore for G_c. This optimality status is due to [8].

References

[1] 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.

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

[3] R. Kumar and V. K. Garg. Optimal control of discrete event dynamical systems using network flow techniques. In *Proc. of 29th Allerton Conf. on Communication, Control and Computing*, Champaign, IL, USA, October 1991.

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

[5] H Marchand, O. Boivineau, and Lafortune S. Optimal control of discrete event systems under partial observation. Technical Report CGR-00-10, Control Group, College of Engineering, Univeristy of Michigan, USA, September 2000.

[6] 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.

[7] 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.

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