12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Robust Supervisory Control Against Intermittent Loss of Observations ⋆

Marcos Vinícius S. Alves * **João Carlos Basilio** *
Antonio Eduardo C. da Cunha ** **Lilian Kawakami Carvalho** *
Marcos Vicente Moreira *

* *COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, 21949-900 Rio de Janeiro, Brazil (e-mails: mvalves@poli.ufrj.br, basilio@dee.ufrj.br, lilian@dee.ufrj.br, moreira@dee.ufrj.br).*
** *Seção de Ensino de Engenharia Elétrica (SE/3), Instituto Militar de Engenharia (IME), Rio de Janeiro, Brazil (e-mail: carrilho@ime.eb.br)*

Abstract: We address in this paper the design of robust supervisors for discrete-event systems subject to intermittent loss of observations. We present two definitions of robust observability: a more restrictive one that requires that the language achieved by the supervisor that control the nominal plant be also achieved by the robust supervisor, and a weaker one that also takes into account possible observation of the events that are subject of intermittent loss of observations. Necessary and sufficient conditions for the existence of robust supervisors that make the controlled system achieve weakly and strongly robust observable languages are also presented. A running example illustrates all the results presented in the paper.

Keywords: Discrete-event systems, Supervisory control, Robustness, Automaton.

1. INTRODUCTION

Robustness is one of the main objectives of feedback control. In the field of discrete event systems (DES), robustness have been considered for both supervisory control (Lin, 1993; Cury and Krogh, 1999; Takai, 2002; Park and Lim, 2000; Bourdon et al., 2002; Takai, 2004; Rohloff, 2005; Saboori and Hashtrudi-Zad, 2006; Sanchez and Montoya, 2006; Rohloff, 2012) and fault diagnosis (Paoli and Lafortune, 2005; Basilio and Lafortune, 2009; Athanasopoulou et al., 2010; Carvalho et al., 2011, 2012; Takai, 2012; Carvalho et al., 2013) in several senses.

The robust supervisory control problem was first considered by Lin (1993), who formulated the problem of designing a robust supervisor, assuming partial observation, with the view to providing the desired behavior assuming that the discrete event system was not modeled by a single nominal automaton but by a class of automata. The problem formulated by Lin (1993) was extended by Bourdon et al. (2002) who considered one specification for each possible plant model, but assuming full observation. Later on Saboori and Hashtrudi-Zad (2006) extended the previous work assuming partial observation.

In a different context, Cury and Krogh (1999) formulated a new robust supervisory control problem that consisted in designing a maximally permissive nonblocking supervisor for the nominal plant model that maximizes the set of plants that, under the control action of the designed supervisor, have the desired specifications. In order to solve the proposed problem, Cury and Krogh (1999) imposes the condition that the maximum legal behavior is a subset of the language generated by the nominal plant, later on removed by Takai (2002) and Takai (2004).

Park and Lim (2000, 2005) propose a new way to model uncertainties in the system behavior by associating uncertainties to unknown unobservable events which by including in the plant model transitions labeled by a so-called Δ event.

More recently, Rohloff (2005, 2012) and Sanchez and Montoya (2006) consider the problem of safe supervisory control in the presence of sensor faults assuming that sensors can fail any time but once the fault occurs, it is permanent.

In this paper we consider the problem of robust supervisory control assuming intermittent loss of observations. This is a more general approach than that considered by Rohloff (2005, 2012) and Sanchez and Montoya (2006), since permanent loss of observations can be seen as a particular case of intermittent loss of observations. We use the model proposed by Carvalho et al. (2012), in the context of fault diagnosis, to extend the definitions of observability to robust observability; indeed, two definitions of robust observability are presented here: *(i)* strong robust observability, that is derived directly from the definition of robust diagnosability proposed by Carvalho et al. (2012); *(ii)* weak robust observability, that takes advantage of possible observation (prior to the actual loss of observation) of the event that is likely to become unobservable to increase the achieved language permissiveness.

⋆ This work was partially supported by the Brazilian Research Council (CNPq) and by Carlos Chagas Foundation (FAPERJ).

The structure of the paper is as follows: we present some basic material on DES in Section 2; in Section 3, we present a motivating example to highlight the need for also taking into account in the supervisor design possible sensor faults and communication channel problems; we formulate the robust supervisory control problem to be dealt with in the paper in Section 4; in Section 5, we present some results that will be needed later on the paper; in Section 6, we present the definitions of strong and weak robust observality and necessary and sufficient conditions for a language to be robustly observable in both senses and for the existence of robust supervisors; in Section 7, we present the robust supervisor design and its realization; finally, conclusions are drawn in Section 8.

2. BACKGROUND PRELIMINARIES

2.1 Definitions and notation

Let $G = (X, \Sigma, f, \Gamma, x_0, X_m)$ denote a deterministic finite state automaton, where X is the finite set of states, Σ is the finite set of events, $f : X \times \Sigma \to X$ is the transition function, partially defined in its domain, $\Gamma : X \to 2^{\Sigma}$ is the active event set, x_0 is the initial state, and X_m is the set of marked states. Assume that Σ can either be partitioned as $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, where Σ_o and Σ_{uo} denote, respectively, the sets of observable and unobservable events or $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$, where Σ_c and Σ_{uc} denote, respectively, the sets of controllable and uncontrollable events. Throughout the text, L and L_m denote, respectively, the languages generated and marked by automaton G.

The natural projection P_o is defined in the usual manner (Ramadge and Wonham, 1989), as $P_o : \Sigma^* \to \Sigma_o^*$, where Σ^* denotes the Kleene closure of Σ, with the following properties: *(i)* $P_o(\varepsilon) = \varepsilon$; *(ii)* $P_o(\sigma) = \sigma$, if $\sigma \in \Sigma_o$; *(iii)* $P_o(\sigma) = \varepsilon$, if $\sigma \in \Sigma_{uo}$ and; *(iv)* $P_o(s\sigma) = P_o(s)P_o(\sigma)$, for $s \in \Sigma^*$ and $\sigma \in \Sigma$, where ε denotes the empty string. The inverse projection operator P_o^{-1} is defined as $P_o^{-1}(t) = \{s \in \Sigma^* : P_o(s) = t\}$. Both the projection and the inverse projection operations can be extended to languages by applying $P_o(s)$ and $P_o^{-1}(s)$ to all strings s in the language.

Let $\Sigma_o = \Sigma_{ilo} \dot{\cup} \Sigma_{nilo}$ be a partition of Σ_o, where Σ_{ilo} is the set of observable events subject to intermittent loss of observations and Σ_{nilo} denotes the set of observable events not subject to intermittent loss of observations. Define $\Sigma'_{ilo} = \{\sigma' : \sigma \in \Sigma_{ilo}\}$ and $\Sigma_{dil} = \Sigma \cup \Sigma'_{ilo}$. We denote $P_{dil,o} : \Sigma_{dil}^* \to \Sigma_o^*$ as the natural projection from Σ_{dil}^* over Σ_o^*. The dilation operation (Carvalho et al., 2012) is the mapping $D : \Sigma^* \to 2^{(\Sigma_{dil})^*}$ with the following properties: *(i)* $D(\varepsilon) = \{\varepsilon\}$; *(ii)* $D(\sigma) = \sigma$, if $\sigma \in \Sigma \backslash \Sigma_{ilo}$; *(iii)* $D(\sigma) = \{\sigma, \sigma'\}$, if $\sigma \in \Sigma_{ilo}$ and; *(iv)* $D(s\sigma) = D(s)D(\sigma)$, $s \in \Sigma^*$, $\sigma \in \Sigma$. The dilation operation can be extended to languages by applying it to all sequences in the language, that is, $D(L) = \bigcup_{s \in L} D(s)$.

An automaton model G_{dil} that takes into account intermittent loss of observations was proposed by Carvalho et al. (2012), being formed by adding to the transitions labeled with event $\sigma \in \Sigma_{ilo}$ parallel transitions labeled with the corresponding event $\sigma' \in \Sigma'_{ilo}$. G_{dil} is formally defined as follows:

$$G_{dil} = (X, \Sigma_{dil}, f_{dil}, \Gamma_{dil}, x_0, X_m), \tag{1}$$

where $\Gamma_{dil}(x) = D[\Gamma(x)]$, and f_{dil} is defined as follows: $\forall \sigma_{dil} \in \Gamma_{dil}(x) : \sigma_{dil} \in D(\sigma)$, $f_{dil}(x, \sigma_{dil}) = f(x, \sigma)$, with $\sigma \in \Gamma(x)$. As proved in Carvalho et al. (2012), $L(G_{dil}) = D(L)$ and $L_m(G_{dil}) = D(L_m)$.

2.2 Supervisory control problem

When it is necessary to restrict the behavior of G in order to satisfy some performance specification, we introduce a feedback path together with a structure called supervisor. A supervisor acts by restricting event occurrences based on the observations of the strings generated by G.

When there is full observation of events ($\Sigma_{uo} = \emptyset$), the supervisor makes its decision based on the actual string s generated by G. It acts by changing the active event set of G, *i.e.*, $\Gamma_N(x) = \Gamma(x) \cap S(s)$, where $\Gamma_N(x)$ denotes the new active event set of state $x = f(x_0, s)$. A supervisor will be referred to as admissible if it does not disable uncontrollable events.

If $\Sigma_{uo} \neq \emptyset$, then the supervisor decides which events are to be disabled based on the projection of the generated string on Σ_o^*. It is worth remarking that under partial observation, two different strings s_1 and s_2 with the same projection lead to the same control action. This is equivalent to saying that the supervisor makes its decision based on $P_o(s)$ and not on s, and, for this reason, it is usually denoted as $S_P[P_o(s)]$. Formally, a partial observation supervisor, or simply, a P-supervisor, is a mapping

$$\begin{aligned} S_P : P_o(L) &\to 2^{\Sigma} \\ P_o(s) &\mapsto S_P[P_o(s)] \end{aligned}$$

such that $\Gamma_N[f(x_0, s)] = \Gamma[f(x_0, s)] \cap S_P[P_o(s)]$.

Let K and $L = \overline{L}$ be languages defined over Σ^*. We say that K is controllable with respect to L and Σ_{uc} if $\overline{K}\Sigma_{uc} \cap L \subseteq \overline{K}$. In addition, we say that K is observable with respect to L, P_o and Σ_c if for all $s \in \overline{K}$ and for all $\sigma \in \Sigma_c$, ($s\sigma \notin \overline{K}$) and ($s\sigma \in L$) implies that $P_o^{-1}[P_o(s)]\sigma \cap \overline{K} = \emptyset$. Finally, we say that K is normal with respect to L and P_o if $\overline{K} = P_o^{-1}[P_o(\overline{K})] \cap L$. Throughout the text whenever a language K is simply referred to as controllable (resp. observable) it should be understood that K is controllable with respect to L and Σ_{uc} (resp. observable with respect to L, P_o and Σ_c).

Finally, assume that K is not controllable (resp. normal). Then $K^{\uparrow C}$ (resp. $K^{\uparrow N}$) is the maximal controllable (resp. normal) sublanguage of K if the following conditions hold true: *(i)* $K^{\uparrow C} \subseteq K$ (resp. $K^{\uparrow N} \subseteq K$) is controllable (resp. normal); *(ii)* if there exists $K' \subset K$ controllable (resp. normal) then $K' \subseteq K^{\uparrow C}$ (resp. $K' \subseteq K^{\uparrow N}$).

3. A MOTIVATING EXAMPLE

Consider automaton G depicted in Figure 1(a), where $\Sigma = \{\alpha, \beta, \gamma, \delta\}$ is the set of events, and $\Sigma_o = \Sigma$ and $\Sigma_c = \{\alpha, \delta\}$ are, respectively, the sets of observable and controllable events. Assume that the language generated by G, $L(G)$, must be modified in order to satisfy the specification language K generated by automaton H, whose state diagram is shown in Figure 1(b).

Let us first consider the supervisor design assuming full observation. Since β and γ are uncontrollable, they cannot