STATE-SPACE SUPERVISION OF RECONFIGURABLE DISCRETE EVENT SYSTEMS

by

Humberto E. Garcia
Argonne National Laboratory-West
Engineering Division
P.O. Box 2528
Idaho Falls, ID 83403-2528

and

Asok Ray
Department of Mechanical Engineering
The Pennsylvania State University
University Park, PA 16802

Paper to be Submitted
to the
CDC'95
34th IEEE Conference
on Decision and Control

New Orleans, Louisiana

December 13-15, 1995

*Work supported by the U.S. Department of Energy, Reactor Systems, Development and Technology, under Contract W-31-109-Eng-38.

State-Space Supervision of Reconfigurable Discrete Event Systems

Humberto E. Garcia
Argonne National Laboratory
P.O. Box 2528
Idaho Falls, ID 83403-2528
garcia@anl.gov

Asok Ray
The Pennsylvania State University
Mechanical Engineering Department
University Park, PA 16802
a2r@ecl.psu.edu

Abstract

The Discrete Event Systems (DES) theory of supervisory and state feedback control offers many advantages for implementing supervisory systems. Algorithmic concepts have been introduced to assure that the supervising algorithms are correct and meet the specifications. It is often assumed that the supervisory specifications are invariant or, at least, until a given supervisory task is completed. However, there are many practical applications where the supervising specifications update at real time. For example, in a Reconfigurable Discrete Event System (RDES) architecture, a bank of supervisors is defined to accommodate each identified operational condition or different supervisory specifications. This adaptive supervisory control system changes its supervisory configuration to accept coordinating commands or to adjust for changes in the controlled process. This paper addresses reconfiguration at the supervisory level of hybrid systems along with a RDES underlying architecture. It reviews the state-based supervisory control theory and extends it to the paradigm of RDES and in view of process control applications. The paper addresses theoretical issues with a limited number of practical examples. Due to page limitation, all the proofs are omitted but can be found in [3]. This control approach is particularly suitable for hierarchical reconfigurable hybrid implementations as those in [4].

1. Introduction

In many control applications, the supervisory system is often required to be adaptive and flexible to accommodate time-varying supervisory specifications or persistently changing environments. DES with time-varying specifications or unknown disturbances are loosely defined as time-varying DES. To deal with these systems, two approaches can be mentioned [11], namely, robust supervision and adaptive supervision. In robust supervision, a given supervisor is designed to perform adequately in the event of identified uncertainties. However, uncertainties are not resolved. In an adaptive supervisory system, uncertainties are resolved by identifying the plant conditions and updating the control algorithms accordingly. Among the growing literature on DES, work in uncertain and/or time-varying systems has been rather limited. Recently, the concepts of robust and adaptive supervision have been introduced [11] under the Ramadge-Wonham framework [12]. In [2], a supervisory control scheme based on limited lookahead control is described where the next control action is determined based on the projection of the process behavior. The approach taken here is to partition the controlled DES along with its operating conditions into subprocesses and operating regimes, and a supervisor is devised for each pair. More than one supervisor may be synthesized for a given subprocess [3, 4]. However, because a process is in exactly one operating condition at any given time, only one supervisor for each subprocess is operational at that instant. The class of discrete event systems that can reconfigure its supervisory algorithms are called *Reconfigurable Discrete Event Systems* (RDES). This paper focuses on the issues for the underlying RDES architecture created by the reconfigurable supervision scheme. To this end, a review of basic ideas of supervisory control in the state-based framework is provided with modeling aspects introduced in view of process control. Analytical guidelines related to the implementation of RDES are then presented. However, limited examples are given here. For more an extended discussion, the reader is referred to [3] and [4].

2. Framework for DES

A. The Discrete Event Model

A discrete event system is represented by $M := (\Gamma, \Psi)$ where M is the discrete-event mechanism and Γ and Ψ are the *static* and *dynamic* components of M, respectively. Γ is defined as $\Gamma := (\mathcal{V}, \Sigma, \mathcal{P})$ where $\mathcal{V}$, Σ, and $\mathcal{P}$ are finite sets. The set $\mathcal{V}$ is the nonempty set of *internal state variables* $x_i \in \mathcal{V}$. The *internal* state space expanded by these x_i is denoted by $\mathcal{X}$. M also defines an *output state space* $\mathcal{X}^O$ that results from a specified functional composition $g(\cdot)$ of its internal state variables; i.e., $\mathcal{X}^O = g(\mathcal{X})$. This $\mathcal{X}^O$ characterizes the state space

of M that can be observed by other mechanisms. The set Σ is the event space or *input alphabet* representing the set of events defined for the process M. Finally, the set $\mathcal{P}$ is the collection of *state predicates* defined on $\mathcal{X}$. A predicate $P \in \mathcal{P}$ is a Boolean map $P: \mathcal{X} \rightarrow \{0,1\}$ that holds on $x \in \mathcal{X}$ if $P(x) = 1$. On the other hand, Ψ is defined as $\Psi := (I, d, f, O)$ where $I(\cdot)$ is called the *input* function, $O(\cdot)$ is the *output* function, $d(\cdot)$ is the *possible events* function and $f(\cdot)$ is a partial function called the *state transition* function. $I(\cdot)$ and $O(\cdot)$ are, in general, mappings of the form $I: \times_i \mathcal{X}_i \times 2^{\Sigma} \rightarrow \mathcal{X} \times \Sigma$ and $O: \mathcal{X} \times \Sigma \rightarrow \mathcal{X}^{O} \times 2^{\Sigma}$ where $\mathcal{X}_i$ is the output state space of the ith mechanism that M reads and $\mathcal{X}^{O}$ is the output state space of M. The possible events function $d: \mathcal{X} \rightarrow 2^{\Sigma}$ is a set-valued function that specifies the set of possible events defined at each state. Formally, $d(x) := \{\sigma \in \Sigma : D.\sigma.x = 1\}$ where the mapping $D.\sigma: \mathcal{X} \rightarrow \{0,1\}$ is a Boolean-valued expression in the variables of $\mathcal{V}$ representing the *enabling condition* for the event σ. The state transition function $f: \mathcal{X} \times \Sigma \rightarrow \mathcal{X}$ defines the dynamics of a given DES specifying how state changes occur in an M due to incoming events.

B. Control Approach

To control DES, certain events in the system are enabled or disabled to govern, whenever possible, transitions among states. Here, control specifications are given in terms of predicates on the set of states. The design problem is to formulate a control agent, hereafter called supervisor, that assigns control patterns (to be defined below) at each state so that a specified predicate can be satisfied. Based on their controllability, events are classified into controllable and uncontrollable events. While *controllable events* Σ_c can be disabled or prevented from occurring whenever desired, *uncontrollable events* Σ_u are those whose occurrence cannot be governed by a supervisor. The control law for a discrete event process is then realized by a *control pattern* for M as a Boolean function $U.\sigma: \mathcal{X} \rightarrow \{0,1\}$ that specifies if a controllable event σ is allowed to occur at a given state x. An event σ is enabled by $U.\sigma$ at x if $U.\sigma.x = 1$; it is disabled, otherwise. The *control input* is equal to $u[k] := \{\sigma \in \Sigma_C : \sigma \in d(x[k]), U.\sigma.x = 1\}$. The *disturbances* acting on a DES, denoted by w[k], is the set of uncontrollable events that may occur at a given state x[k] defined as $w[k] := \{\sigma \in \Sigma_u : \sigma \in d(x[k])\}$. Thus, the set of current possible events is dynamic and equal to $\sigma[k+1] \in u[k] \cup w[k]$. The DES mechanism to be controlled or *plant*, denoted by M_C, influences the state transitions of the controlling mechanism or *supervisor*, denoted by M_S, by means of observed plant state and, possibly, incoming events, while M_C is driven by a sequence $\{w[k]\}$ of disturbances and by a sequence $\{u[k]\}$ of control inputs determined by the consecutive states x[k] of M_S. This class of supervision will be called *event/state* feedback control [14].

3. Control of Time-Varying DES

The objective of a supervisor M_S is to modify the open loop response of a given plant M_C to track a desirable response as close as possible. Let $M_C[k]$ denote the time-varying DES model that characterizes M_C with $M_c[k] \in \{M_{c_i}\}_{i=1}^{m}$, where m denotes the number of possible operating conditions defined for M_C and M_{Ci} denotes the formulated model for the ith condition. To control $M_C[k]$, two supervisory approaches can be used.

A. Single Robust Supervisor Approach

A single *robust* supervisor M_S is designed to accommodate all possible variations of $M_C[k]$. Assume $M_C[k]$ to be represented by $M_c[k] = \hat{M}_c + \Delta M_c[k]$, where $\hat{M}_c$ is the nominal time-invariant model of the controlled DES and $\Delta M_c[k]$ represents the time-varying modeling uncertainties. The approach of using a single supervisor M_S suffers from several problems. First, the variability of $\Delta M_c[k]$ could be large enough that a single supervisor cannot be synthesized to account for all cases. Even if a supervisor exists, it may become too complex or overly conservative and hard to justify if large uncertainties occur rarely. If these rather rare cases are not considered during the design steps, the control system could be vulnerable to operational risk over the service life of the plant. System performance may also be significantly degraded to assure robustness.

B. Reconfigurable Supervisor Approach

In this case, the control efforts are exercised by employing a time-varying supervisor $M_S[k]$. A finite set of possible supervisors is defined as $M_s[k] \in \{M_{s_i}\}_{i=1}^{n}$ where n is the number of designed supervisors. The problem is to design a set of supervisors s.t $(\forall M_{c_i}) \exists M_{s_j} .\quad R_e(M_{c_i} / M_{s_j}, P_i) \leq P_i$ where M_{c_i} / M_{s_j} is the feedback composition of M_{Ci} and M_{Si}, P_i is the desirable operational predicate and $R_e(M,P)$ is the reachable predicate from an initial predicate P. Thus, supervisors are designed to accommodate only a given restricted subset of all possible plant conditions or supervisory specifications. This approach simplifies the design of supervisors as the tasks of analysis, synthesis, verification, upgrade and maintenance become easier. In addition, a reconfigurable system does not enforce a complete modification of an existing configuration but extends it to accommodate additional requirements.

4. Reconfigurable Supervisory Architecture

In a Reconfigurable Discrete Event System (RDES), as illustrated in Figure 1, a set of banks of

are designed for each processes operational conditions. Only one supervisor in each bank is acting at any time. When a higher decision maker in the hierarchy (in this case, the *coordinator*) decides to change the supervising algorithm, a switch to a new supervisor takes place. Figure 1 introduces the *control channel*. Via the control channel Con_i, the ith supervisor applies control to M_C. The ith control channel is a mapping $Con_i : 2^{\Sigma_c} \times \{0,1\} \rightarrow 2^{\Sigma_c} \cup \{\varepsilon\}$ implemented as follows:

$$Con_i(u, \gamma_i) := \begin{cases} u & \text{if } \gamma_i = 1 \\ \varepsilon & \text{otherwise} \end{cases}$$

where $\gamma_i[k] \in \{0,1\}$ is a control signal generated by the coordinator to govern the Con_i outputs. The scalar γ_i acts as a *supervisor enabling* signal. Specifically, the output of a given M_{Si} is communicated to M_C if its respective control channel has been enabled (i.e., $\gamma_i = 1$) by the coordinator. This, in turn, is equivalent to enabling or disabling a given supervisor's operation.

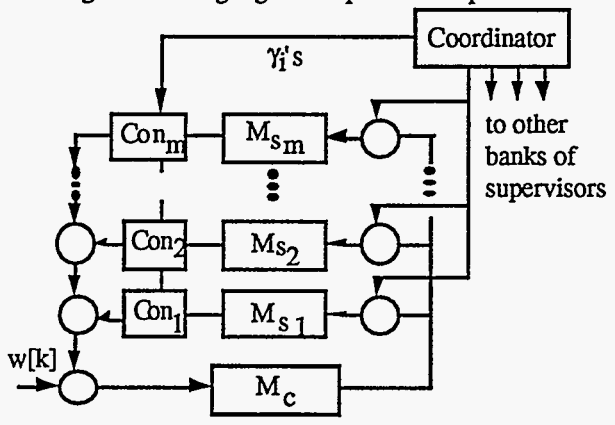

Figure 1. A Reconfigurable Discrete Event System.

5. Problems Due to Reconfiguring Supervisors

It is often assumed that the initial state of a given plant/supervisor pair is fixed, known a priori and one of the "legal" states. Starting from a pre-defined initial region, the objective of a supervisor is then to confine the process behavior within specified bounds. However, under reconfigurable supervision, the initial state space region may not be in the legal region for the current supervising condition. Assume that M_{Si} has been acting over M_C for some period such that any trajectories generated have been confined to the desired state region; i.e., $(\forall k > t_0 \wedge k < t_f)\, M_s[k] = M_{s_i} \wedge P_i.x[k] = 1$, where t_0 and t_f denote given instants, and P_i defines the control specification for the M_C/M_{Si} pair. Assume that at t_f, M_{Si} is disabled and a new supervisor M_{Sj} starts to act. Let P_j denote the "target" predicate for the new plant/supervisor configuration M_C/M_{Sj}. At time t_f, while $P_i.x[t_f] = 1$, it maybe possible that $P_j.x[t_f] = 0$. Thus, predicates are not always true for all states. Because $x[t_f]$ may not satisfy the new control specification P_j, it must be assured that the next desired predicate can be satisfied after a finite number of state transitions. Therefore, convergence from P_i to P_j must be guaranteed. A transitory and a steady state period can be identified and supervisory algorithms must be then designed to assure desirable responses.

6. Control Tasks on RDES

A. Task 1 *Control Invariance*:

This is achieved by a supervisor that ensures by control actions that a given predicate P_i remains invariantly true whenever it is initially satisfied. That is, $(\forall x \in \mathcal{X}_{P_i})\, (\forall \sigma \in u[k] \cup w[k])\ sp.f_\sigma.P_i \leq P_i$ where $u[k]$ is the control input to M_C generated by M_{Si}, $w[k]$ is the externally induced disturbance, and sp is the strongest postcondition predicate transformer. This task usually occurs in regulatory problems.

B. Task 2 *Region Avoidance*:

This is achieved by a supervisor that ensures by control actions that a given region will never be reached. In reference to RDES, a coordinator may switch from a supervisor M_{Si} to another M_{Sj} and a new predicate P_j is desired to be satisfied. For the general case of $P_j \neq P_i$, the supervisory system must then try to reach the target region P_j from this initial condition P_i. A series of state trajectories leading to P_j can be generated. However, it must be assured during this process that the system does not enter into "bad" regions where it may fail to converge or violate safety. Let $P_B \in \mathcal{P}$ define a forbidden state region. Then, any M_S must guarantee that $(\forall x \in \mathcal{X}_{\neg P_i})$ $(\forall \sigma \in u[k] \cup w[k])\ sp.f_\sigma.P_x \leq \neg P_B$, with $\neg$ denoting the negation operator.

C. Task 3 *Convergence*:

This is achieved by a supervisor that ensures by control actions that a given region can be reached from a starting space. That is, starting from a state satisfying P_i, the system should converge to a state in P_j after countable many transitions. It is often desired to achieve convergence in finite transitions. In this case, there must exist a positive integer "q" such that $\exists q \in \mathcal{Z}$ s.t. $P_j \wedge R_e(M, P_i, q) \neq P_\varnothing$ where $P\varnothing$ is the empty predicate and $R_e(M, P, q)$ is the *reachable predicate* that results from firing q event transitions from any state satisfying P. This task is particularly important in situations such as error recovery and system reconfiguration.

7. Solutions to the Control Tasks for RDES

A. Task I: *Control Invariance*

Proposition 7.1: Task I can only be satisfied if the predicate P is Σ_u-invariant [13].

A.1. Algorithms to Achieve Task I

Synthesis Algorithm 7.1.1:

Given that the desirable P is Σ_u-invariant and M_C is in a state x where P is satisfied, a controllable event σ is enabled at x iff its firing implies that the control-invariant condition $(\forall\sigma \in u[k])\ sp.f_\sigma.P \le P$ is preserved. That is, $(\sigma \in \Sigma_c)\ U.\sigma \Rightarrow (sp.f_\sigma.P \le P)$.

Synthesis Algorithm 7.1.2 (From Lemma 2.2 [10]):

Given that M_C is started in any state satisfying a given predicate P_1, and a weaker predicate $P_2 \ge P_1$ is control-invariant, the following condition on predicate reachability $R_e(M_c \setminus u, P_1) \le P_2$ holds if the corresponding feedback predicate $U.\sigma$ for the (static) feedback u is chosen as $(\forall\sigma \in \Sigma_c)\ U.\sigma := wlp.f_\sigma.P_2$ where $R_e(M\setminus u,P)$ denotes the predicate reachable from P and wlp is the weakest liberal precondition.

Synthesis Algorithm 7.1.3 :

Given that P is valid at x, a controllable event σ is enabled iff, upon firing of σ, P continues to be valid under any possible subsequent sequence of uncontrollable events. That is, $(\forall\sigma \in \Sigma_c)\ \sigma \in u(x)$ iff $R_e({}^uM_c, sp.f_\sigma.P_x) \le P$ where P_x is the predicate valid exactly at x and ${}^uM$ is the submechanism of M generated by disabling all controllable events.

B. Task II: Region Avoidance

Let P_T and P_B denote the target and bad predicates, respectively. Avoidance of P_B must be satisfied not only at the given current state but also at any state that may uncontrollably lead to P_B. That is, the predicate to be avoided is $P_B \vee R_e^{-1}({}^uM_c, P_B)$ with $R_e^{-1}(M,P)$ denoting the attractable predicate from where P can be reached. Since $P_B \le R_e^{-1}({}^uM_c, P_B)$ always holds, the above predicate can be characterized by $R_e^{-1}({}^uM_c, P_B)$. Thus, to avoid P_B, a given $\sigma \in \Sigma_c$ is enabled if, after its firing, the trajectory does not enter a region where the previous predicate is valid; i.e., $(\sigma \in \Sigma_c)\ \sigma \in u(x)$ if $R_e^{-1}({}^uM_c, P_B).f_\sigma(x) = 0$. In addition, if the evolving state trajectory has entered the region $R_e^{-1}({}^uM_c, P_B)$, a controllable event should be enabled to move out from this region; i.e., $(\sigma \in \Sigma_c)\ \sigma \in u(x)$ if $R_e^{-1}({}^uM_c, P_B).x = 1$.

B.1. Algorithms to Achieve Task II

Synthesis Algorithm 7.2.1: The implication

$(\forall\sigma \in \Sigma_c)\ U.\sigma.x \Rightarrow$

$R_e^{-1}({}^uM_c, P_B).x \vee \neg R_e^{-1}({}^uM_c, P_B).f_\sigma(x)$ should always be satisfied.

Synthesis Algorithm 7.2.2 :

A controllable event σ is enabled iff, upon firing of σ, the permissible region $\neg P_B$ continues to remain valid under any uncontrollable event sequence or if at the current state the trajectory can uncontrollably reach the undesirable region P_B. That is, $(\forall\sigma \in \Sigma_c)\ \sigma \in u(x)$ iff $(R_e({}^uM_c, sp.f_\sigma.P_x) \le \neg P_B) \vee (P_x \le R_e^{-1}({}^uM_c, P_B))$.

C. Task III: *Convergence*

Definition 7.1 : Let P_1 and P_2 denote the initial and target predicates, respectively. Let $\mathcal{K}$ denote a set of nonnegative integers such that $\mathcal{K} := \{j \in Z^+ : R_e(M_c / M_s, P_1, j) \wedge P_2 \ne P_\varnothing\}$ where Z^+ denotes the set of all nonnegative integers, and $P\varnothing$ denotes the predicate defined such that $\forall x \in \mathcal{X}\ P\varnothing.x = 0$. If $\mathcal{K} \ne \varnothing$ for a given plant/supervisor configuration, then the *(minimum) predicate for convergence* from P_1 to P_2, denoted as $P_C(P_1,P_2)$, is defined as $P_C(P_1,P_2) = R_e(M_C/M_S, P_1, i)$ where $i = \min_{j \in K}(j)$.

Proposition 7.2: Let $\wp(P_1,P_2)$ denote the set of state paths $\mathcal{X}^*$ defined as $\wp(P_1,P_2) := \{\Pi \in \mathcal{X}^* : P_1.\Pi(1) = P_2.\Pi(q) = 1,\ (\forall i \in \{1,...,q\})\ P_C(P_1,P_2).\Pi(i) = 1$ with $q = |\Pi|\}$ where Π denotes a state path on $\mathcal{X}$, $\Pi(i)$ the ith state on the state sequence imposed by Π, and $|\Pi|$ the length of Π. Assume that $\mathcal{K} \ne \varnothing$. Then, $\wp(P_1,P_2) \ne \varnothing$.

Theorem 7.1: Let $\wp(P_1,P_2)$ be defined as given in Proposition 7.2. Assume that $\mathcal{K} \ne \varnothing$ and $P_C(P_1,P_2)$ is controllable from P_1. Then, $(\forall x \in P_1)\ \exists \Pi \in \wp(P_1,P_2)$ s.t. $x = \Pi(1)$.

C.1. Algorithms to Achieve Task III

First, in light of Theorem 7.1 and Theorem 2.1 in [10], Algorithm 7.3.1 is given. Next, based on Theorem 7.1 and Theorem 3.1 in [9], an alternative algorithm for supervisor synthesis is given in 7.3.2.

Synthesis Algorithm 7.3.1:

i) Let $P_C(P_1,P_2) \ne P_\varnothing$.

ii) If $R_C(P_1,P_2)$ is not controllable from P_1, find the supremal sub predicate of $R_C(P_1,P_2)$, i.e., $supC(P_1,R_C(P_1,P_2))$.

iii) Assume then that

$$\sup C\left(P_1, P_c(P_1,P_2)\right) \wedge P_2 \neq \varnothing \qquad (1)$$

If (1) is not satisfied for the current $R_c(P_1,P_2)$, increase the value of i until (1) is valid.

iv) Then, any controllable event σ may be enabled at a state x only if its firing leads to a state from where it is known that P_2 can be reached; that is,

$$\left(\forall \sigma \in \Sigma_c\right)\sigma \in u(x) \text{ if } sp.f_\sigma.P_x \leq \sup C\left(P_1, P_c(P_1,P_2)\right)$$

Synthesis Algorithm 7.3.2 :

i) Assume the conditions i) - iii) taken by Algorithm 7.3.1.

ii) Then, any controllable event σ may be enabled at a given state x iff, after the firing of σ, the predicate sup $C(P_1,R_c(P_1,P_2))$ is satisfied under any possible subsequent sequence of uncontrollable events. That is, $\left(\forall \sigma \in \Sigma_c\right)\sigma \in u(x)$ iff

$$R_e({}^{u}M_c, sp.f_\sigma.P_x) \leq \sup C\left(P_1, P_c(P_1,P_2)\right).$$

8. Predicate Convergence on Regions of Attraction

The possibility of driving the controlled DES from arbitrary initial states to a specified target region is of interest in a more general framework. To this end, stabilization properties and asymptotic behavior needs to be investigated. To the best of the authors' knowledge, the first study involving concepts of stabilization of DES was introduced in [1] followed by redefinition of classical concepts of dynamics as invariant sets and attractors. This section extends some of the results in [1] for the proposed framework.

A. Region of Strong Attraction

Definition 8.1: Let $P_2 \leq P_1$ where P_1 and P_2 are two predicates on $\mathcal{X}$. Then P_2 is a *concentric strong attractor* for P_1 under the discrete event mechanism M, denoted as $P_2 \overset{M}{\Leftarrow} P_1$, if the following conditions are met:

i) $(\forall \sigma \in \Sigma) P_2 \leq wlp.f_\sigma.P_2$ (rel P_2)

ii) $R_e(M,P_1) \leq R_e^{-1}(M,P_2)$

iii) ($\forall x \in \mathcal{X}$ with

$R_e(M,P_1).x = \neg P_2.x = 1$) $f(x,s) = x \Rightarrow s = \varepsilon$

where ε denotes the empty event sequence.

Strong attraction assures that the system eventually converges to a specified target region if initialized in a state belonging to its region of attraction. Condition i) indicates that any state in the region defined by P_2 will stay inside of it under any event firing. Therefore, Σ-invariance is a necessary condition for a given predicate P to be a concentric strong attractor. Condition ii) indicates that any state reachable from P_1 is attractable to P_2. Therefore, any state trajectory leaving P_1 can then be driven to end in P_2. Finally, condition iii) indicates that for any state reachable from P_1 but not in P_2, there exists no sequence of events other than the empty string that causes no state changes.

Theorem 8.1: Let SA(M,P) denote the class of all predicates for which P is a concentric strong attractor; i.e., $SA(M,P) := \{ Q \in \mathcal{P} : P \leq Q, \; P \overset{M}{\Leftarrow} Q \}$. Then, SA(M,P) has a maximal element.

Corollary to Theorem 8.1: Let SA(M,P) denote the class of all predicates for which P is a concentric strong attractor. Then, a maximal element of SA(M,P) exist, denoted by $SA_{max}(M,P)$, such that

$\neg \exists Q \in SA(M,P)$ s.t. $SA_{max}(M,P) < Q$.

The maximal element mentioned in the above corollary is called the *ball of strong predicate attraction* of P under M. If P is not Σ-invariant, then $SA_{max}(M,P) = \varnothing$. In such a case, the maximal Σ-invariant subpredicate of P is found to replace P in the above equations.

Proposition 8.1: A target region given by a predicate P_2 can be reached from another region P_1 if P_1 is contained within the ball of strong predicate attraction of P_2; that is, $P_1 \leq SA_{max}(M,P_2)$.

B. Region of Weak Attraction

Definition 8.2 : Let $P_2 \leq P_1$ where P_1 and P_2 are two predicates on $\mathcal{X}$. Then, P_2 is a *concentric weak attractor* for P_1 with respect to the plant M_c, denoted as $P_2 \overset{M_c}{\leftarrow} P_1$, if there exists a supervisor M_S such that, for the closed-loop system M_c/M_S, P_2 is a concentric strong attractor of P_1; i.e., $P_2 \overset{M_c/M_s}{\Leftarrow} P_1$.

Remark 8.1: It follows from Definitions 8.1 and 8.2 that strong attraction implies (by definition) weak attraction.

Proposition 8.2: Let $P_2 \leq P_1$ where P_1 and P_2 are predicates on $\mathcal{X}$ such that P_2 is Σ_u-invariant. Then, P_2 can be a concentric weak attractor for P_1 if there exists a supervisor M_S such that the following conditions hold:

i) $R_e(M_c/M_s,P_1) \leq R_e^{-1}(M_c/M_s,P_2)$

ii) ($\forall x \in \mathcal{X}$ with

$\left(\neg P_2 \wedge R_e(M_c/M_s,P_1)\right).x = 1$) $f(x,s) = x \Rightarrow s = \varepsilon$

where ε the empty event sequence.

Corollary to Proposition 8.2: If all events are controllable, then the necessary and sufficient condition needed to assure convergence is given by $P_1 \leq R_e^{-1}(M_c/M_s,P_2)$.

Theorem 8.2: Let $WA(M_C,P)$ denote the class of all predicates for which P is a weak attractor; i.e.,

$$WA(M_C,P) := \{ Q \in \mathcal{P} : P \leq Q,\ P \xleftarrow{M_c} Q \}$$

Then, $WA(M_C,P)$ has a maximal element.

Corollary to Theorem 8.2: Let $WA(M_C,P)$ denote the class of all predicates for which P is a concentric weak attractor. Then, a maximal element of $WA(M_C,P)$ exists, denoted by $WA_{max}(M_C,P)$, such that $\neg\exists Q \in WA(M_c,P)$ s.t. $WA_{max}(M_c,P) < Q$.

The maximal element satisfying the above corollary is called the *ball of weak predicate attraction* of P with respect to M_C. If P is not Σ_u-invariant, then $WA_{max}(M_c,P) = \varnothing$. In this case, the maximal Σ_u-invariant sub predicate of P is found to replace P in the above equations.

Proposition 8.3: A target region defined by a given predicate P_2 can be reached from another region P_1 under supervision if the latter is evolved by the ball of weak attraction of the former; that is, $P_1 \leq WA_{max}(M_c,P_2)$.

Synthesis Algorithm 8.3.1 :

Let P_1 and P_2 denote the predicates specified to be enforced whenever the closed-loop system is characterized by the plant/supervisor pairs M_C/M_{S1} and M_C/M_{S2}, respectively. If the coordinator decides a reconfiguration from supervisor M_{S1} to M_{S2}, then the plant is needed to be moved from P_1 to the region defined by P_2. Therefore, convergence to P_2 from P_1 must be guaranteed. If Proposition 8.3 is satisfied, it is assured that a supervisor exists that can drag any state in P_1 to a state in P_2.

9. Summary

Reconfiguration refers to the capability of changing system configuration based on operational conditions. This paper proposes a reconfigurable supervisory approach to control plants subjected to unknown disturbances including time-varying discrete event systems (DES). In a reconfigurable approach, the controlled DES process along with its operating conditions is partitioned into sets of subprocesses and operating regimes, and a supervisor is devised for each pair of subprocess and subprocess operating condition. The current operating condition of the plant is identified and a supervisor is selected by a high level decision-maker (e.g., a coordinator) among the available ones to act directly on the plant. Experimental evaluations of concepts presented in this paper are partially given elsewhere [4, 5]. Additional experimental work is needed and tests are currently being planned for future research.

References

1. Y. Brave and M. Heymann, *Stabilization of discrete-event processes,* International Journal of Control, Vol. 51, No. 5, 1990, pp. 1101-1117.
2. S. Chung, S. Lafortune, F. Lin, *Limited Lookahead Policies in Supervisory Control of Discrete Event Systems,* IEEE Trans. on AC, Vol. 37, No. 12, 1992, pp. 1921-1935.
3. H. E. Garcia and A. Ray, *State-Space Supervisory Control of Reconfigurable Discrete Event System* , Inter. Journal of Control, Vol. 63, No. 2, Feb. 1996.
4. H. E. Garcia, A. Ray, and R. M. Edwards, *A Reconfigurable Hybrid System and Its Application to Power Plant Control,* IEEE Trans. on CST,. Vol. 3, No. 2, June 1995, pp. 157-170.
5. H. E. Garcia, "A Reconfigurable Hierarchical Hybrid Supervisory Control System," Ph.D. Dissertation, The Pennsylvania State University, December 1993.
6. Aleks Gollu and Pravin Varaiya, *Hybrid Dynamical Systems,* Proceedings of the 28th CDC, Tampa, FL, Dec 1989, pp. 2708-2712.
7. R. Kumar, V. Garg, S. I. Marcus, *Predicates and Predicate Transformers for Supervisory Control of Discrete Event Dynamical Systems,* IEEE Trans. on AC, Vol. 38, No. 2, Feb. 1993.
8. Y. Li, *Control of Vector Discrete-Event Systems,* System Control Group Report No.9106, University of Toronto, 1991.
9. Y. Li and W. M. Wonham, *A State-Variable Approach to the Modeling and Control of Discrete-Event Systems,* Proc. 2nd Allerton Conf. on CCC, 1988, pp. 1140-1149.
10. Y. Li and W. M. Wonham, *Controllability and Observability in the State-Feedback Control of Discrete-Event Systems,* Proc. of the 27th CDC, Austin, Texas, Dec. 1988, pp. 203-208.
11. F. Lin, *Robust and Adaptive Supervisory Control of Discrete Event Systems,* Proc. of ACC, Chicago, IL, June 1992, pp. 2804-2808.
12. P. J. Ramadge and W. M. Wonham, *The Control of Discrete Event Systems,* Proceedings of the IEEE, Vol. 77, No. 1, Jan. 1989, pp. 81-98.
13. P. J. Ramadge and W. M. Wonham, *Modular Feedback Logic for Discrete Event Systems,* SIAM J. Control and Optimization, Vol.25, No.5, Sep. 1987, pp. 1202-1218.
14. R. S. Sreenivas and B. H. Krogh, *On Condition/Event Systems with Discrete State Realizations,* DEDS: Theory & Applications 1, (1991), pp. 209-236.
15. T. Ushio, *Controllability and control-invariance in discrete-event systems,* Inter. Journal of Control, Vol. 50, No. 4, 1989, pp. 1507-1515.