2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Synthesis of Optimal Fault-Tolerant Supervisor for Discrete Event Systems

Q. Wen, Member IEEE, R. Kumar, Fellow, IEEE, J. Huang, Member IEEE
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa 50011

***Abstract*— In an earlier work [1], [2], we introduced a framework for fault-tolerant supervisory control of discrete event systems and presented a necessary and sufficient condition for its existence. Here we propose an approach to synthesize an optimal fault-tolerant supervisory controller. Given a discrete event plant with both faulty and nonfaulty behaviors, an optimal fault-tolerant supervisor we synthesize enforces a set of behaviors in which (i) a recovery is guaranteed within a bounded delay following any fault, (ii) the enforced set of nonfaulty behaviors are maximized, and (iii) the enforced set of faulty behaviors prior to the recovery are minimized. An example is given to illustrate the approach.**

Keywords: discrete event systems, fault tolerant control, supervisory control, stability, convergence.

I. Introduction

In an earlier work [1], [2] we introduced a framework for fault-tolerant supervisory control of discrete event systems and presented a necessary and sufficient condition for its existence. Fault-tolerance is a property requiring that a system continues to function, possibly with a degraded performance, even when some of its components fail. Given a plant G, possessing both faulty and nonfaulty behaviors, and a submodel G^N for the nonfaulty part, the goal of fault-tolerant control is to enforce a certain specification K^N for the nonfaulty plant G^N and another (perhaps more liberal) specification $K \supseteq K^N$ for the overall plant G, and further to ensure that the plant recovers from any fault within a bounded delay, so that following the recovery the system state is equivalent to a nonfaulty state (as if no fault ever happened). The condition for the existence of a fault-tolerant controller involves the usual notions of controllability, observability and relative-closure, together with the notion of stability [3], which is used to establish bounded delay recovery from a fault.

Some previous work involved controller switching upon the occurrence of a fault as in [4], or re-computation of a controller as in [5]. The resulting controlled system can tolerate some faults but the system performance after faults will remain degraded since the notion of recovery from faults was not incorporated. Case studies involving synthesis of fault-tolerant supervisors can also be found in [6], [7]. Fault-tolerance in Petri Net is considered in [8], where liveness enforcing strategies are designed to deal with failures using system reconfigurations. In [9], authors considered a pair of specifications, representing the desired and the (more liberal) tolerable behavior for a plant.

The research was supported in part by the National Science Foundation under the grants NSF-ECS-0424048, NSF-ECS-0601570 and NSF-ECCS-08013763.

In this paper we study the synthesis of an optimal fault-tolerant supervisory controller when the required existence conditions (as reported in [1], [2]) are not satisfied. An optimal fault-tolerant supervisor we synthesize enforces a set of behaviors in which (i) a recovery is guaranteed within a bounded delay following any fault, (ii) the enforced set of nonfaulty behaviors are maximized, and (iii) the enforced set of faulty behaviors prior to the recovery are minimized. Given (G, G^N), where G is a plant and G^N is its nonfaulty part, and a state-based specification (X^g, X^g_m) representing legal states and legal final states respectively, we compute a subplant $(\tilde{G}, \tilde{G}^N)$ such that (i) $\tilde{G}^N$ ($\sqsubseteq G^N$) is a maximal controllable subplant of G^N for which there exists G' with $\tilde{G}^N \sqsubseteq G' \sqsubseteq G$ and $(G', \tilde{G}^N)$ is fault-tolerant, (ii) $\tilde{G}$ is a minimal such G', and (iii) safety and nonblockingness properties are satisfied. The above is guided by the goal to maximize the achievable nonfaulty behaviors and at the same time minimize the faulty behaviors that must be tolerated, without having to sacrifice safety, nonblockingness, and recovery. We show that $\tilde{G}^N$ can be uniquely chosen (since the corresponding property is closed under union), whereas nonunique minimal choices exist for $\tilde{G}$ (the corresponding property is closed under the intersection over decreasing chains). We present an algorithm for computing $(\tilde{G}, \tilde{G}^N)$ and illustrate the algorithm through an example.

II. Notation and Preliminaries

A DES to be controlled, called plant, is modeled as an automaton, denoted by a five tuple $G := (X, \Sigma, \alpha, x_0, X_m)$, where X denotes the set of states, Σ denotes the finite set of events, $\alpha : X \times \Sigma \to X$ denotes the partial deterministic state transition function, $x_0 \in X$ denotes the initial state, and $X_m \subseteq X$ denotes the set of marked states. For $x \in X$, we use $\Sigma(x) \subseteq \Sigma$ to denote the set of events defined at x, i.e., $\Sigma(x) := \{\sigma \in \Sigma \mid \alpha(x, \sigma) \text{ is defined}\}$. Σ^* is used to denote the set of all finite-length sequences of events, called traces, which includes the zero-length trace ϵ. The length of a trace s, denoted as $|s|$, is defined to be the number of events in the trace. A subset of Σ^* is called a language. The generated language of G, denoted as $L(G) \subseteq \Sigma^*$, contains all traces s for which $\alpha(x_0, s)$ is defined. The marked language of G, denoted as $L_m(G)$, contains all generated traces that reach a marked state.

Given two automata $G_1 := (X_1, \Sigma, \alpha_1, x_{0_1}, X_{m_1})$ and $G_2 := (X_2, \Sigma, \alpha_2, x_{0_2}, X_{m_2})$, G_1 is said to be a subautomaton of G_2, denoted as $G_1 \sqsubseteq G_2$, if there exists an injective

map $h : X_1 \to X_2$ such that $\forall s \in L(G_1) : h(\alpha_1(x_{0_1}, s)) = \alpha_2(x_{0_2}, s)$.

For traces s and t, we use $s \leq t$ to denote that s is a prefix of t and $s < t$ to denote that s is a proper prefix of t. For a language $K \subseteq \Sigma^*$, $pr(K)$, called the prefix-closure of K, denotes the set of all prefixes of traces in K, i.e., $pr(K) = \{s \in \Sigma^* \mid \exists t \in K : s \leq t\}$. It is clear that $K \subseteq pr(K)$, and K is said to be prefix-closed if $K = pr(K)$. A language K is said to be relative-closed with respect to G, if $pr(K) \cap L_m(G) = K \cap L_m(G)$.

We use $K\backslash s$ to denote the set of traces that occur in the language K after the trace s has occurred, i.e., $K\backslash s := \{t \in \Sigma^* \mid st \in K\}$. For traces s and t, we use $s \sqsubseteq_K t$ to denote that the sets of traces that occur in K after s are contained in those after t, i.e., $K\backslash s \subseteq K\backslash t$. We write $s \cong_K t$ if $s \sqsubseteq_K t$ and $t \sqsubseteq_K s$. $s \sqsubseteq_K t$ implies the behaviors following s are subsumed by the behaviors following t, whereas $s \cong_K t$ implies the equivalence of the behaviors following s and t. We write $s \sqsubseteq_G t$ to denote $s \sqsubseteq_{L_m(G)} t$ and $s \sqsubseteq_{L(G)} t$. Also $s \cong_G t$ if $s \sqsubseteq_G t$ and $t \sqsubseteq_G s$.

Definition 1: [3] Given a plant $G = (X, \Sigma, \alpha, x_0, X_m)$ and a state set $\widehat{X} \subseteq X$, $x \in X$ is said to be $\widehat{X} - attractable$ in G if there exists $m \in \mathcal{N}$ such that for all t for which $\alpha(x, t)$ is defined and either $|t| \geq m$ or t deadlocks, exists $t' \leq t$ with $|t'| \leq m$ such that $\alpha(x, t') \in \widehat{X}$. $x \in X$ is said to be controllably $\widehat{X} - attractable$ in G if there exists a supervisor S such that x is $\widehat{X} - attractable$ in $G\|S$.

We use $\Omega_G(\widehat{X})$, the region of attraction of $\widehat{X}$, to denote the set of all $\widehat{X}$-attractable states, and $\widehat{X}$ is called an attractor for the set $\Omega_G(\widehat{X})$. We use $\Omega^c_G(\widehat{X})$, the region of controllable attraction of $\widehat{X}$, to denote the set of all controllably $\widehat{X}$-attractable states, and $\widehat{X}$ is called a controllable attractor for the set $\Omega^c_G(\widehat{X})$. A state set $\tilde{X} \subseteq X$ is said to be attractable to $\widehat{X}$ if $\tilde{X} \subseteq \Omega_G(\widehat{X})$ and controllably attractable to $\widehat{X}$ if $\tilde{X} \subseteq \Omega^c_G(\widehat{X})$. Clearly, $\widehat{X} \subseteq \Omega_G(\widehat{X}) \subseteq \Omega^c_G(\widehat{X})$.

For control purposes, the event set of G is partitioned into the set of controllable events $\Sigma_c \subseteq \Sigma$ and the set of uncontrollable events $\Sigma_u \subseteq \Sigma$. A language K is said to be controllable (with respect to G and Σ_u) if $pr(K)\Sigma_u \cap L(G) \subseteq pr(K)$.

A supervisor is another automaton $S := (Y, \Sigma, \beta, y_0, Y_m)$. The supervised plant is the synchronous composition of G and S, denoted $G||S := (X \times Y, \Sigma, \gamma, (x_0, y_0), X_m \times Y_m)$, where for $(x, y) \in X \times Y$ and $\sigma \in \Sigma$, $\gamma((x, y), \sigma)$ is defined if and only if both $\alpha(x, \sigma)$ and $\beta(y, \sigma)$ are defined and in which case, $\gamma((x, y), \sigma) = (\alpha(x, \sigma), \beta(y, \sigma))$. It can be concluded that the generated and the marked languages of the supervised plant satisfy: $L(G||S) = L(G) \cap L(S)$ and $L_m(G||S) = L_m(G) \cap L_m(S)$, respectively. A supervisor S is said to be (i) nonmarking if $L_m(G||S) = L(G||S) \cap L_m(G)$, (ii) nonblocking if $pr(L_m(G||S)) = L(G||S)$, and (iii) Σ_u-compatible if it does not disable any uncontrollable event (equivalently if $L(G||S)$ is controllable).

The following notion was introduced in [1].

Definition 2: Consider a pair of languages (H, H^N) with $H^N \subseteq H$. The pair (H, H^N) is said to be fault-tolerant if exists $m \in \mathcal{N}$ such that for $s \in pr(H) - pr(H^N)$, $st \in pr(H)$ with $|t| \geq m$ or st deadlocks, there exist $u \in pr(H^N)$ and $t' \leq t$ with $|t'| \leq m$, $st' \cong_H u$ and $st' \cong_{pr(H)} u$. In this case, m is called the *delay-bound of fault-tolerance*. Given a plant G with its nonfaulty part G^N, (G, G^N) is said to be fault-tolerant if $(L(G), L(G^N))$ is fault-tolerant. A supervisor S is said to be fault-tolerant if $(G\|S, G^N\|S)$ is fault-tolerant.

The following result was obtained in [1].

Theorem 1: [1] Given a plant $G = (X, \Sigma, \alpha, x_0, X_m)$ with nonfaulty part $G^N = (X^N, \Sigma, \alpha^N, x_0, X^N_m)$, specification $\emptyset \neq K \subseteq L_m(G)$ for G and specification $\emptyset \neq K^N \subseteq L_m(G^N)$ for G^N satisfying $K^N \subseteq K$, there exists a nonmarking, nonblocking (with respect to both G^N and G), Σ_u-compatible and fault-tolerant supervisor S such that

1) $L_m(G^N||S) = K^N, L(G^N||S) = pr(L_m(G^N||S))$,
2) $L_m(G||S) = K$, and $L(G||S) = pr(L_m(G||S))$

if and only if

1) K is relative-closed and controllable w.r.t. G,
2) (K, K^N) is fault-tolerant, and
3) $K^N = K \cap L_m(G^N)$ and $pr(K^N) = pr(K) \cap L(G^N)$.

III. Formulation of Optimal Fault-Tolerant Control Synthesis Problem

Theorem 1 provides a condition under which a desired fault-tolerant supervisor exists. When this condition is satisfied, a trim recognizer of K can be chosen as a supervisor. Here we study the problem of synthesizing a fault-tolerant supervisor when the condition of Theorem 1 is not satisfied. A desirable goal is to maximize the achievable nonfaulty behaviors and at the same time minimize the faulty behaviors that must be tolerated, without sacrificing safety, nonblockingness and recovery. The motivation being, we allow maximal functionality of the system in the absence of faults, and at the same time limit the system's faulty behavior within a minimal range without sacrificing recovery.

It turns out that the supremal nonfaulty fault-tolerant behavior does not exist in general. That is, given a language pair (K, K^N), we cannot always find a fault-tolerant sublanguage pair $(\tilde{K}, \tilde{K^N})$, where $\tilde{K} \subseteq K$ and $\tilde{K^N} \subseteq K^N$, such that any other fault-tolerant sublanguage pair $(\widehat{K}, \widehat{K^N})$ satisfies $\widehat{K} \subseteq \tilde{K}$ and $\widehat{K^N} \subseteq \tilde{K^N}$.

Consider the following example for illustration. Figure 1 shows a plant G and its nonfaulty part G^N, where f is a faulty event and is the only uncontrollable event. From Figure 1, we can see that there are two subplant pairs (G_1, G_1^N) and (G_2, G_2^N), which are fault-tolerant. Note in (G_1, G_1^N) the faulty state 6 is equivalent to the nonfaulty state 3, whereas in (G_2, G_2^N) the faulty state 6 is equivalent to the nonfaulty state 2. Thus in both cases the system reaches a state that is equivalent to a nonfaulty state within one transition of the occurrence of the fault (i.e., the delay bound for recovery is one in both cases). However the language $(L(G_1) \cup L(G_2), L(G_1^N) \cup L(G_2^N))$ which is realized by (G, G^N) shown in Figure 1 is not fault-tolerant. This is because there exists no nonfaulty state that is equivalent to the faulty state 6 where the system can stay with recovery.

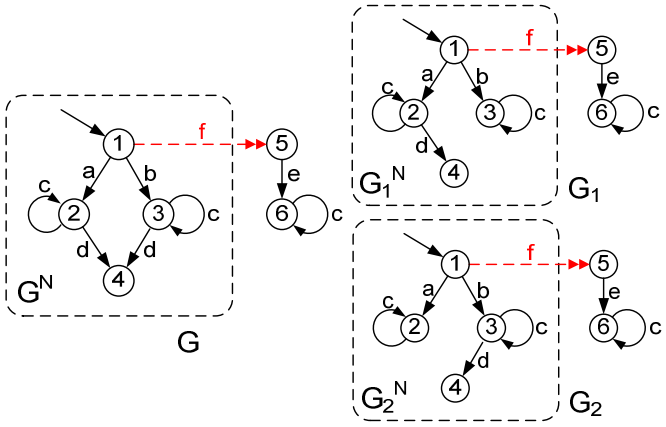

Fig. 1. Plant G with its subplants G_1 and G_2

It happens that maximal nonfaulty behaviors that are fault-tolerant also do not exist, for the limit of a class of monotonically increasing nonfaulty fault-tolerant behaviors may not be fault-tolerant. To see this, consider the monotonically increasing sequence of plant behaviors, where the nth behavior in the sequence is generated by the plant (G_n, G_n^N) (Figure 2). The nonfaulty part G_n^N contains n a's, whereas the overall plant G_n contains a faulty trace with same number of b's, i.e., the delay bound for recovery in (G_n, G_n^N) is n. The limiting plant behavior $(L(G_\infty) := \cup_n L(G_n), L(G_\infty^N) := \cup_n L(G_n^N))$ is not fault-tolerant since the faulty plant can execute an unbounded number of b's before a recovery to the nonfaulty part occurs.

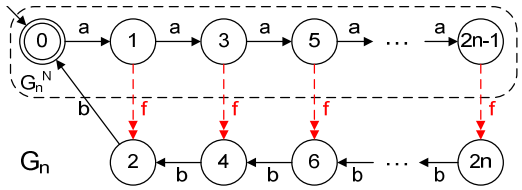

Fig. 2. Plant $(G_n, G_n^N), n \geq 1$

The examples above show that neither the supremal nor a maximal nonfaulty fault-tolerant sublanguage exists in general. Lack of supremal and even maximal nonfaulty fault-tolerant sublanguages motivate us to restrict our attention to state-feedback based control, under which the controlled plant is always a subplant of the uncontrolled plant. In this setting, we are able to show the existence of fault-tolerant control that maximizes the nonfaulty behavior while minimizing the faulty behavior without sacrificing safety, nonblockingness, and recovery.

Without loss of generality, the specification is given as a state set pair (X^g, X_m^g), where $X^g \subseteq X$ is the set of legal states and $X_m^g \subseteq X^g \cap X_m$ is the set of legal final states. Since under a state-feedback control the controlled plant is a subplant of the uncontrolled plant, examining various state-feedback controllers is equivalent to examining various subplants of a given plant.

We first define the set of all fault-tolerant subplants, denoted $F(G, G^N)$ as follows:

Definition 3: Given a plant model (G, G^N) with $G^N \sqsubseteq G$, the class of *fault-tolerant subplants*, denoted $F(G, G^N)$, is the set of all subplants $(\tilde{G} \sqsubseteq G, \tilde{G}^N \sqsubseteq G^N)$ with states pair $(\tilde{X}, \tilde{X}^N)$ and marked states pair $(\tilde{X}_m, \tilde{X}_m^N)$, such that

- $\tilde{X} \subseteq X^g$, $\tilde{X}_m \subseteq X_m^g$;
- $L_m(\tilde{G})$ is relatively-closed and controllable w.r.t. G;
- $\tilde{X} \subseteq \Omega_{\tilde{G}}(\tilde{X}^N)$.

We next introduce the class of fault-tolerant nonfaulty subplants and show that this class is closed under union.

Definition 4: The class of *fault-tolerant nonfaulty subplants* is defined as: $F_G^N(G^N) := \{\tilde{G}^N \sqsubseteq G^N | \exists \tilde{G} \sqsubseteq G : (\tilde{G}, \tilde{G}^N) \in F(G, G^N)\}$.

The following theorem shows that the above class of fault-tolerant nonfaulty subplants is closed under union.

Theorem 2: Let Λ be an index set such that $\forall \lambda \in \Lambda$, $G_\lambda^N \in F_G^N(G^N)$. Then $\bigcup_{\lambda \in \Lambda} G_\lambda^N \in F_G^N(G^N)$.

Proof: We show the existence of $G' \sqsubseteq \cup_\lambda G_\lambda$ such that $(G', \cup_\lambda G_\lambda^N) \in F(G, G^N)$.

Suppose $(\cup_\lambda G_\lambda, \cup_\lambda G_\lambda^N)$ is fault-tolerant. Then we claim that we can choose $G' = \cup_\lambda G_\lambda$. Since for each λ, $X_\lambda \subseteq X^g$, we have $\cup_\lambda X_\lambda \subseteq X^g$. Since for each λ, $L_m(G_\lambda)$ is controllable which means no uncontrollable event is disabled in G to obtain each G_λ, it is the case that no uncontrollable event is disabled to obtain $\cup_\lambda G_\lambda$, i.e., $L_m(\cup_\lambda G_\lambda)$ is controllable. Since for each λ, $L_m(G_\lambda)$ is relatively-closed which implies that $X_\lambda \cap X_m \subseteq X_\lambda \cap X_m^g$, it is the case that $\cup_\lambda X_\lambda \cap X_m \subseteq \cup_\lambda X_\lambda \cap X_m^g$, i.e., $L_m(\cup_\lambda G_\lambda)$ is relatively-closed.

On the other hand suppose $(\cup_\lambda G_\lambda, \cup_\lambda G_\lambda^N)$ is not fault-tolerant. Then exists a cycle in the faulty part. Since for each λ, (G_λ, G_λ^N) is fault-tolerant which implies the faulty part of G_λ does not contain any cycle, i.e., a certain edge of each faulty-part cycle of $\cup_\lambda G_\lambda$ is missing in G_λ. Then each such edge must be labeled with a controllable event (since $L_m(G_\lambda)$ is controllable). Let G' be obtained from by removing each edge that contributes to a cycle in the faulty-part of $\cup_\lambda G_\lambda$ and is missing in $G_{\bar{\lambda}}$ for a $\bar{\lambda} \in \Lambda$. Then the faulty-part of G' is acyclic. Also since only controllable edges are removed to obtain G' from $\cup_\lambda G_\lambda$, the controllability property is preserved. Since $L_m(\cup_\lambda G_\lambda)$ is controllable (see above), we can claim that $L_m(G')$ is controllable. Further since G' is obtained from $\cup_\lambda G_\lambda$ by removing certain edges that appear as part of certain cycles, the state set is preserved upon the removal of such edges (i.e., $X' = \cup_\lambda X_\lambda$). It can then be concluded that relative-closure property is also preserved, and so $L_m(G')$ is also relatively-closed.

It remains to show that $(G', \cup_\lambda G_\lambda^N)$ is fault-tolerant. Since the faulty-part of G' is acyclic, it suffices to show that for any faulty state $x \in X' = \cup_\lambda X_\lambda$ exists a path in G' to the nonfaulty part $\cup_\lambda X_\lambda^N$. Pick any such state x. Then exists λ such that x is a faulty state of G_λ. From the fault-tolerance of (G_λ, G_λ^N), exists a path in G_λ from x to $G_\lambda^N \sqsubseteq \cup_\lambda G_\lambda^N$. If this path does not contain any of the edges that were removed to obtain G', then we are done. Otherwise this path visits a state $\bar{x}$ from where an edge that is present in $\cup_\lambda G_\lambda$ but is missing in $G_{\bar{\lambda}}$ has been removed. From the fault-tolerance

of $(G_{\bar{\lambda}}, G^N_{\bar{\lambda}})$ exists a path in $G_{\bar{\lambda}} \sqsubseteq G'$ from $\bar{x}$ to $G^N_{\bar{\lambda}} \sqsubseteq G'$. This concludes the proof. ■

Since $F^N_G(G^N)$ is closed under union, it possesses a supremal element, $\sup F^N_G(G^N) \in F^N_G(G^N)$, with the property that if $\tilde{G}^N \in F^N_G(G^N)$, then $\tilde{G}^N \sqsubseteq \sup F^N_G(G^N)$.

In the above, we defined a class of fault-tolerant nonfaulty subplants. Next, given a nonfaulty subplant, we define a class of overall subplant that is fault-tolerant.

Definition 5: Given $\tilde{G}^N \sqsubseteq G^N$, the class of *overall subplants that are fault-tolerant w.r.t.* $\tilde{G}^N$ is defined as:

$$F_{\tilde{G}^N}(G) := \{\tilde{G} \sqsubseteq G | (\tilde{G}, \tilde{G}^N) \in F(G, G^N)\}.$$

$\tilde{G}$ is an *infimal* element of $F_{\tilde{G}^N}(G)$ if $\tilde{G} \in F_{\tilde{G}^N}(G)$, and $G' \in F_{\tilde{G}^N}(G)$ implies $\tilde{G} \sqsubseteq G'$. $\tilde{G}$ is a *minimal* element of $F_{\tilde{G}^N}(G)$ if $\tilde{G} \in F_{\tilde{G}^N}(G)$ and $G' \sqsubseteq \tilde{G}$ implies $G' \notin F_{\tilde{G}^N}(G)$.

The following result establishes certain closure properties of $F_{\tilde{G}^N}(G)$ under intersection.

Theorem 3: $F_{\tilde{G}^N}(G)$ does not possess infimal element but possess minimal elements whenever it is nonempty.

Proof: For the first part, we show that $F_{\tilde{G}^N}(G)$ is not closed under intersection. As seen from Figure 3, $G_1, G_2 \in F_{G^N}(G_1 \cup G_2)$ (since for each $i = 1, 2$, $(G_i, G^N) \in F(G_1 \cup G_2, G^N)$). However it is clear from Figure 3 that $(G_1 \cap G_2, G^N) \notin F(G_1 \cup G_2, G^N)$, i.e., $G_1 \cap G_2 \notin F_{G^N}(G_1 \cup G_2)$.

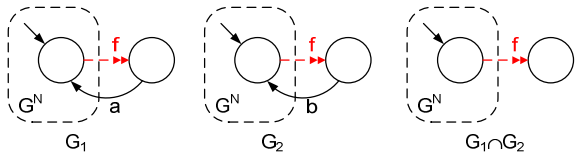

Fig. 3. $(G_1, \tilde{G}^N)$ and $(G_2, \tilde{G}^N)$ are fault-tolerant, but $(G_1 \cap G_2, \tilde{G}^N)$ is not

Next we consider the second part. Since G has finite number of states and transitions, the number of subplants of G is finite. So, whenever $F_{\tilde{G}^N}(G)$ is nonempty, there exist at least one subplant of G in $F_{\tilde{G}^N}(G)$ for which there exists no subplant in the set $F_{\tilde{G}^N}(G)$, and so the existence of a minimal overall subplant that is fault-tolerant with respect to $\tilde{G}^N$ follows. ■

The set of all the minimal elements of $F_{\tilde{G}^N}(G)$ is denoted as $\text{MIN}F_{\tilde{G}^N}(G)$, and a minimal element is denoted as $\min F_{\tilde{G}^N}(G)$.

IV. Optimal Fault-Tolerant Control Synthesis

The computation of an optimal fault-tolerant control discussed above requires the computation of the region of controllable attraction and a minimal set of controllable transition for achieving the controllable attractability given in Algorithm 1. It is obtained by extending that given in [10] to keep track of a minimal set of controllable transitions that must be enabled to achieve the controllable attractability.

Algorithm 1: Consider a plant $G = (X, \Sigma, \alpha, x_0, X_m)$ and a state set $\widehat{X} \subseteq X$.

1) Initialization step:
   Set $k = 0$, $\Omega_{k-1} = \emptyset$, $\Omega_k = \widehat{X}$, and $\Delta_k = \emptyset$.
2) Iteration step:
   - $\Omega_{k+1} = \Omega_k \cup \{x \in X - \Omega_k | \alpha(x, \Sigma) \cap \Omega_k - \Omega_{k-1} \neq \emptyset, \alpha(x, \Sigma_u) \subseteq \Omega_k\}$.
   - $\Delta_{k+1} = \Delta_k \cup \{(x, \sigma, x') \mid x \in \Omega_{k+1} - \Omega_k, \sigma \in \Sigma_x, x' \in \Omega_k\}$, where $\Sigma_x = \emptyset$ if $\alpha(x, \Sigma_u) \neq \emptyset$ and otherwise $\Sigma_x = \{\sigma_x\}$ such that $\alpha(x, \sigma_x) \in \Omega_k$.
3) Termination step:
   If $\Omega_{k+1} = \Omega_k$, then stop and set $\Omega^c_G(\widehat{X}) = \Omega_k$ and $\Delta_G(\widehat{X}) = \Delta_k$; else set $k = k + 1$ and go to step 2.

Remark 1: The complexity of Algorithm 1 can be seen to be linear in the size of the plant G. This is because at most $|X|$ iterations are being performed, and in each iteration a constant amount of computation is being done.

The following algorithm computes a fault-tolerant subplant $(\tilde{G}^N, \tilde{G})$ such that $\tilde{G}^N = \sup F^N_G(G^N)$ and $\tilde{G} \in \text{MIN}F_{\tilde{G}^N}(G)$.

Algorithm 2: Consider plant $G = (X, \Sigma, \alpha, x_0, X_m)$ with nonfaulty part $G^N = (X^N, \Sigma, \alpha^N, x_0, X^N_m)$, and specification (X^g, X^g_m).

Uncontrollable/blocking states removal:

1) Initialization step:
   $k = 0$, $D_k = X - X^g$.
2) Iteration step:
   If $x_0 \in D_k$, then terminate (no solution exists); else
   - $D = D_k \cup \{x \in X - D_k \mid \alpha(x, \Sigma^*_u) \subseteq D_k\}$;
   - $D_{k+1} = D \cup \{x \in X - D \mid \alpha|_{X-D}(x, \Sigma^*) \cap X^g_m = \emptyset\}$.
3) Termination step:
   If $D_{k+1} \neq D_k$, set $k = k + 1$ and iterate; else remove from G the states in D_k and all their incoming and outgoing transitions to obtain $G_0 = (X_0, \Sigma, \alpha_0, x_0, X_{m,0})$, and let $G^N_0 = (X^N_0, \Sigma, \alpha^N_0, x_0, X^N_{m,0})$ be its nonfaulty part.

Optimal fault-tolerant subplant computation:

1) Initialization step:
   $k = 0$.
2) Iteration step:
   If $x_0 \notin X^N_k$, then terminate (no solution exists); else
   - $X' = \{x \in X^N_k \mid \alpha_k(x, \Sigma^*_u) \subseteq X_k\}$;
   - $X^N_{k+1} = \{x \in X' \mid \alpha_k|_{X'}(x, \Sigma^*) \cap X_{m,k} \neq \emptyset\}$;
   - $X_{k+1} := \alpha_k(X^N_{k+1}, \Sigma^*) \cap \Omega^c_{G_k}(X^N_{k+1})$;
   - $G_{k+1} = (X_{k+1}, \Sigma, \alpha_{k+1}, x_0, X_{m,k+1}) := G_k|_{X_{k+1}}$; $G^N_{k+1} = (X^N_{k+1}, \Sigma, \alpha^N_{k+1}, x_0, X^N_{m,k+1}) := G^N_k|_{X^N_{k+1}}$.
3) Termination step:
   If $X_{k+1} \neq X_k$, set $k = k + 1$ and iterate; else remove all controllable transitions from G_k that leave the state set X_k, and also all controllable transitions in the faulty part of G_k that are not present in $\Delta_{G_k}(X^N_{k+1})$. This yields a desired subplant $(\sup F^N_G(G^N), \min F_{\sup F^N_G(G^N)}(G))$.

The steps of Algorithm 2 can be understood as follows. The "uncontrollable/blocking states removal" step computes the supremal relative-closed and controllable subplant by removing those states which can uncontrollably reach an

illegal state in $X - X^g$ or can never reach a legal final state in X_m^g. This provides a supremal safe and nonblocking subplant (G_0, G_0^N). Additional pruning is required to also ensure fault-tolerance (besides safety and nonblockingness) so all faulty states are attractable to the nonfaulty ones. This is accomplished in the "optimal fault-tolerant subplant computation" step. The kth iteration computes a subplant (G_{k+1}, G_{k+1}^N) of (G_k, G_k^N) by first computing X_{k+1}^N which consists of all nonfaulty states in X_k^N that are controllable and nonblocking with respect to the states in X_k of the overall plant G_k. This maximizes the set of nonfaulty behaviors. Since faulty states in X_{k+1} must be controllably attractable to X_{k+1}^N, only those states in X_k that belong to $\Omega_{G_k}^c(X_{k+1}^N)$ are kept in X_{k+1}. Further to achieve the minimality of the behaviors in the faulty part, only the reachable states of X_{k+1}^N are kept in X_{k+1}, i.e., we set X_{k+1} to be the set of reachable and controllably attractable states of X_{k+1}^N. The iteration continues if $X_{k+1} \neq X_k$ and otherwise it terminates yielding a desired optimal fault-tolerant subplant. To ensure the minimality of the behavior in the faulty part only a minimal set of controllable transitions needed to ensure attractability to the nonfaulty part are kept (which are computed as $\Delta_{G_k}(X_{k+1}^N)$ in Algorithm 1).

Remark 2: It can be verified that the number of iterations of the steps "uncontrollable/blocking states removal" as well as "optimal fault-tolerant subplant computation" is bounded by the number of states in G, and each such iteration has a complexity that is linear in the size of plant G. It follows that the complexity of Algorithm 2 is quadratic in the size of plant G.

The correctness of Algorithm 2 is next established.

Theorem 4: Given overall plant G, nonfaulty plant G^N and specification (X^g, X_m^g), Algorithm 2 generates the pair $(\tilde{G}^N, \tilde{G})$ with $\tilde{G}^N = \sup F_G^N(G^N)$ and $\tilde{G} \in \text{MIN}F_{\tilde{G}^N}(G)$.

Proof: We first prove that $\tilde{G}^N$ is the supremal element of $F_G^N(G^N)$. It is straightforward to see that $(\tilde{G}, \tilde{G}^N)$ is fault-tolerant since all states in $\tilde{G}$ are in the region of controllable attraction of $\tilde{G}^N$. Further $(\tilde{G}, \tilde{G}^N)$ is safe since illegal states cannot be uncontrollably reached from any state in X_k (this is assured by the "uncontrollable/blocking states removal" step). $(\tilde{G}, \tilde{G}^N)$ is also nonblocking since X_{k+1}^N is nonblocking with respect to $X_{m,k}$ for each k and on termination $X_{k+1}^N = X_k^N$ (meaning at termination X_k^N is nonblocking with respect to $X_{m,k}$). To show the supremality of $\tilde{G}^N$ we claim that any nonfaulty state not present in $\tilde{X}^N$ cannot be in an optimal solution. This is because at each of the kth iteration only the states from X_k^N are removed to obtain X_{k+1}^N that are uncontrollable/blocking with respect to X_k. Only the states from X_k are kept to obtain X_{k+1} that is controllably attractable to and also reachable from X_{k+1}^N.

Next we prove that the plant $\tilde{G}$ is an element of $\text{MIN}F_{\tilde{G}^N}(G)$. It is clear that all faulty states in $\tilde{G}$ are reachable from and controllably attractable to $\tilde{G}^N = \sup F_G^N(G^N)$. We show the minimality of $\tilde{G}$ by showing that by removing any controllable transition will make $(\tilde{G}, \tilde{G}^N)$ become non-fault-tolerant. From the construction of $\Delta_{G_k}(X_{k+1}^N)$, at any faulty state at most one controllable transition is enabled and if an uncontrollable transition is defined at a faulty state, no controllable transition is enabled. From the minimality of the controllable transitions included in $\Delta_{G_k}(X_{k+1}^N)$ (as guaranteed by Algorithm 1), we can conclude that removal of any controllable transition from the faulty part of $(\tilde{G}, \tilde{G}^N)$ will make it non-fault-tolerant. ■

V. Illustrative Example

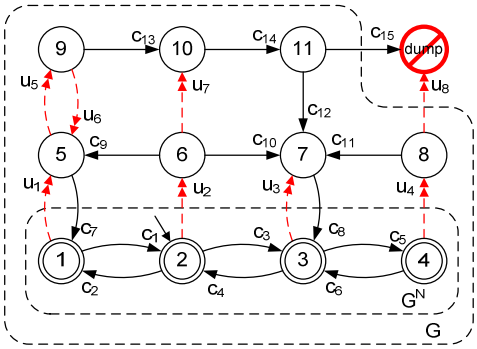

Fig. 4. Plant (G, G^N) used in Section V

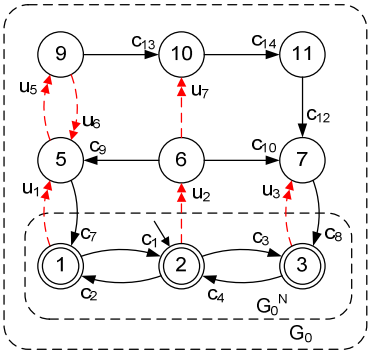

Fig. 5. Controllable and nonblocking subplant (G_0, G_0^N)

The following example illustrates the Algorithm 2. Consider the plant (G, G^N) given in Figure 4. Encircled states denote the final states. There is a single illegal state labeled *dump*; the remaining states form the state set X^g. The specification for the marked states is given as $X_m^g = X^g \cap X_m = X_m$. The dotted double arrowed transitions are uncontrollable and the remaining ones are controllable. All transitions from a state in X^N to a state in X are considered faulty (and also uncontrollable). Note each transition has a distinct event label and so it can be identified by the event labeling the transition.

The subplant (G_0, G_0^N) obtained after the removal of uncontrollable and blocking states is shown in Figure 5. Note the states 4 and 8 get removed since they can reach the illegal state uncontrollably. Also note that $X_0^N = \{1, 2, 3\}$, and $X_0 = \{1, 2, 3, 5, 6, 7, 9, 10, 11\}$.

Since $x_0 = \{2\} \in X_0$, the computation of the fault-tolerant subplant proceeds as follows.

1) Iteration no. 1:
   - $X_1^N = \{1, 2, 3\}$;

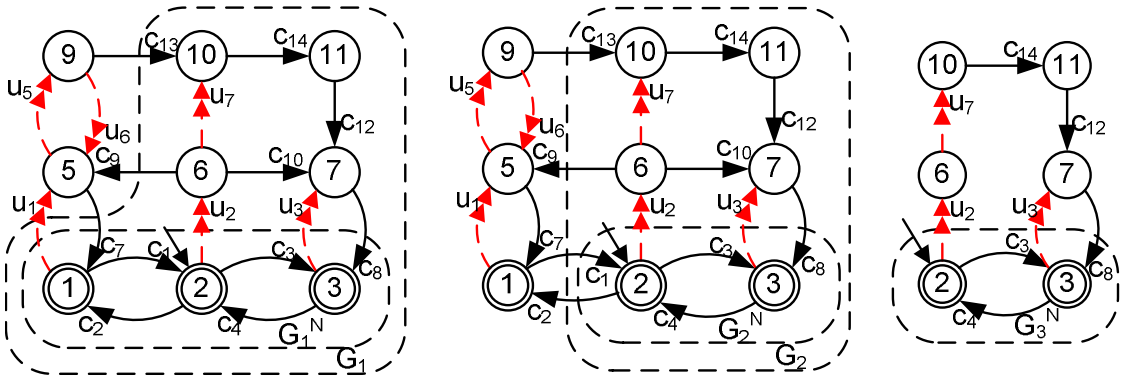

Fig. 6. (G_1, G_1^N), (G_2, G_2^N) and final controlled plant (G_3, G_3^N) obtained from iterations of Algorithm 2

- $\Omega^c_{G_0}(X_1^N) = \{1, 2, 3, 6, 7, 10, 11\}$, $\Delta_{G_0}(X_1^N) = \{c_8, c_{12}, c_{14}\}$;
- $X_1 = \{1, 2, 3, 6, 7, 10, 11\}$. The resulting (G_1, G_1^N) is shown in Figure 6.

2) Iteration no. 2:
- $X_2^N = \{2, 3\}$;
- $\Omega^c_{G_1}(X_2^N) = \{2, 3, 6, 7, 10, 11\}$, $\Delta_{G_1}(X_2^N) = \{c_8, c_{12}, c_{14}\}$;
- $X_2 = \{2, 3, 6, 7, 10, 11\}$. The resulting (G_2, G_2^N) is shown in Figure 6.

3) Iteration no. 3:
- $X_3^N = \{2, 3\}$;
- $\Omega^c_{G_2}(X_3^N) = \{2, 3, 6, 7, 10, 11\}$, $\Delta_{G_2}(X_3^N) = \{c_8, c_{12}, c_{14}\}$;
- $X_3 = \{2, 3, 6, 7, 10, 11\}$.

Since $X_3 = X_2$, the iteration stops. After removing the controllable transitions $\{c_2, c_9\}$ that leave the state set $X_3 = X_2$ and also all controllable transitions in the faulty part of G_2 that are not included in $\Delta_{G_2}(X_3^N) = \{c_8, c_{12}, c_{14}\}$, we get the desired fault-tolerant subplant shown in Figure 6.

We can see that state 2 and 3 are the only nonfaulty states from where after the occurrence of a fault it is possible to recover within a bounded delay. State 1 does not have this property since it is possible to uncontrollably reach state 5 from where a bounded delay recovery is not possible (state 5 is contained in a cycle of uncontrollable transitions). On the other hand state 4 does not have this property since it is possible to uncontrollably reach the illegal state from state 4. It can be seen then that the computed nonfaulty part is supremal. The faulty states 6, 7, and 10 must be present in the overall subplant since those states are uncontrollably reached from the nonfaulty states 2 and 3. Since the only way to recover from the faulty state 10 is through state 11, state 11 must also be included in the overall subplant. Finally removing any controllable transition in the faulty part renders the overall subplant "fault-intolerant". It follows that the computed faulty part is minimal.

VI. Conclusion

A notion of fault-tolerant supervisory control was introduced in our prior work [1], [2] where the controlled system must not only satisfy the desired safety and progress properties but must also be fault-tolerant, i.e., following the occurrence of any fault a recovery to a nonfaulty or nonfaulty-equivalent state must occur within a bounded delay. Here we formulated the notion of an optimal fault-tolerant supervisor to be one that maximizes the nonfaulty behavior and at the same time minimizes the faulty behavior that must be tolerated, and also ensures safety, nonblockingness, and bounded-delay recovery. We showed that while the problem in general does not admit an optimal solution, an optimal solution does exist over the class of state-feedback control policies. We presented an algorithm to find such an optimal solution. The complexity of the algorithm is *quadratic* in the size of a given plant.

References

[1] Q. Wen, R. Kumar, J. Huang, and H. Liu, "Fault-tolerant supervisory control of discrete event systems: Formulation and existence results," in *Proceedings of Dependable Control of Discrete Systems*, Pairs, France, 2007.

[2] ——, "A framework for fault-tolerant supervisor control of discrete event systems," *IEEE Transaction on Automatic Control*, 2007, to Appear.

[3] Y. Brave and M. Heymann, "On stabilization of discrete event processes," *International Journal of Control*, vol. 51, no. 5, pp. 1101–1117, 1990.

[4] H. Darabi, M. A. Jafari, and A. L. Buczak, "A control switching theory for supervisory control of discrete event systems," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 131–137, 2003.

[5] K. R. Rohloff, "Sensor failure tolerant supervisory control," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, 2005, pp. 3493 – 3498.

[6] K.-H. Cho and J.-T. Lim, "Failure diagnosis and fault tolerant supervisory control systems," *IEICE Transactions on Information and System*, vol. E79-D, no. 9, pp. 1223 – 1231, 1996.

[7] ——, "Synthesis of fault tolerant supervisor for automated manufacturing systems: A case study on photolithographic process," *IEEE Trans. on Robotics and Automation*, pp. 348 – 351, 1998.

[8] M. V. Iordache and P. J. Antsaklis, "Resilience to failure and reconfigurations in the supervision based on place invariants," *Proceedings of the 2004 American Control Conference*, pp. 4477 – 4482, 2004.

[9] S. Lafortune and F. Lin, "On tolerable and desirable behaviors in supervisory control of discrete event systems," *Discrete Event Dynamical System: Theory and Application*, vol. 1, no. 1, pp. 61–92, 1991.

[10] R. Kumar, V. K. Garg, and S. I. Marcus, "Language stability and stabilizability of discrete event dynamical systems," *SIAM Journal of Control and Optimization*, vol. 31, no. 5, pp. 1294–1320, September 1993.