IMPROVING THE BEHAVIOUR OF SUPERVISOR UNDER BLOCKING

Ozgür T. Kaymakci[a,*], Leyla Gören[a], Salman Kurtulan[a]

[a] *Control Systems Division, Faculty of Electrics and Electronics Engineering, Istanbul Technical University, Istanbul, Turkey*
** Corresponding Author e-mail: kaymakci@elk.itu.edu.tr*

Abstract: Sometimes, under control of a supervisor a discrete event system could constitute a conservative solution in this case relaxing the blocking becomes an inevitable fact to improve the performance. Meanwhile the balance between blocking and achievement has to be preserved. So the strings that drive the system to blocking and also the strings causing achievement have to be investigated in a numerical manner. For this purpose in this study a new performance measure is introduced. The elements of the performance measure depend on numeric values obtained from strings that correspond to blocking and success so the proposed formulation captures the fundamental trade-off motivated by the classical optimization approach. Then a new algorithm that explores the best result according to this performance measure is introduced.

Keywords: Discrete Event Systems, Finite State Machines, Performance Evaluation

1. INTRODUCTION

Discrete event systems are often modelled as regular languages that can be represented by finite state automata. Generally the behaviour of the system is not satisfactory and must be “modified” by a control action. In this perspective, supervisory control theory (SCT) is a powerful tool that gives us the possibility to build in the modifications via supervisor as much as possible. Therefore all the desired behaviour of the system is not usually marked by the supervisor that is built according to nonblocking and controllability constraints. Sometimes to prevent all probable blockings in the system, the designed supervisor could constitute a “conservative” solution. So one may be willing to risk blocking if there will be serious increase in the performance of the system. Therefore in this work, the performance of the system is examined under blocking and the attempt is to obtain an optimal language from the acceptable solution set. So a new performance measure and an optimization procedure are introduced. In the literature several researchers have proposed optimal control of DES. Chen and Lafortune dealed with the blocking in discrete event systems and introduced two operators on a set of languages that optimizes the performance measure according to set inclusion. (Chen and Lafortune, 1991). Kumar and Garg proposed a cost function with payoff and control costs and transformed the optimal control problem to combinatorial one and solved with network flow algorithm. Kumar and Garg assumed that a certain state may be visited only once, so the corresponding transitions are controlled only once (Kumar and Garg, 1995). Sengupta and Lafortune also used control and event costs to find the optimal nonblocking supervisor and solved the problem by dynamic programming (Sengupta and Lafortune, 1998). Surana and Ray constructed a signed measure over discrete event systems and the measure is defined on state transitions (Surana and Ray). Also Fu and Ray used this signed measure to formulate an unconstrained optimal control policy. The policy is obtained by selectively disabling controllable events to maximize the measure (Fu, J., et al., 2004). These strategies have addressed performance enhancement of discrete event systems but none of them were interested the blocking case and derived a numeric performance measure for it.

This paper introduces an optimization algorithm to find the optimal blocking supervisor based on a specified performance measure built on a metric space. The paper is organized in 5 sections. Section 2 briefly gives the preliminaries. Section 3 describes the motivation and gives the problem formulation. Section 4 presents optimization algorithm and gives an example. The paper is summarized and concluded in section 5.

2. PRELIMINARIES

Discrete event systems (DES) are dynamical systems which evolve in time by the occurrence of events at possibly irregular time intervals. Generally the system to be controlled is modelled with a Deterministic Finite State Machine (DFSM) defined by a 6-tuple $G=(X,\Sigma,f,\Gamma,x_0,X_m)$ where X is the set of states, Σ is the finite set of events, $f: X\times\Sigma\to X$ is the state transition function, $\Gamma: X\to 2^{\Sigma}$ is the active event set, x_0 is the initial state and $X_m\subseteq X$ is the set of marked states representing a completion of a given task or operation. Then the behaviour of the system G is described by a prefix-closed language $L(G)$, which is defined as $L(G)=\{s\in\Sigma^* \mid f^*(x_0,s)\in X\}$ where Σ^* denotes the set of all finite concatenations of events that belong to Σ, including the zero length string ε; the state transition function is extended to: $f^*: X\times\Sigma^*\to X$. $L(G)$ can be considered as the uncontrolled behaviour of the system. In this paper it is assumed that the uncontrolled behaviour of the system $L(G)$ is finite. Similarly the language $L_m(G)$ corresponds to the marked behaviour of the DFSM G.

For a string $s\in\Sigma^*$, $\bar{s}$ denotes the prefixes of s. Extending this definition to languages, prefix closure of a language L denoted as $\bar{L}$ is obtained. When a language L satisfies the condition $L=\bar{L}$ then it is called prefix closed. In the literature the length of a string or Myhill congruence index of a language is symbolized with $\|.\|$. Also p is a projection function defined on string where $p_j(s)$ represents the prefix of string s of length j (Sengupta and Lafortune, 1998).

If $L(G)\neq\overline{L_m(G)}$ then the DFSM G is said to be blocking. Two types of blocking can occur; these are deadlock and livelock. At deadlock, G can reach a state x_i, where $\Gamma(x_i)=\varnothing$ $(x_i\notin X_m)$ and at livelock; the DFSM can reach a set of unmarked states that form a strongly connected group of states, but with no transition out of this set (Lafortune and Chen, 1990). In this work, it is assumed that all the possible blockings are deadlock.

Some of the events in Σ are uncontrollable i.e. their occurrence cannot be prevented by a controller. A sensor output at a manufacturing system is a good example of this class. In this regard, Σ is partitioned as $\Sigma=\Sigma_c\cup\Sigma_{uc}$ $(\Sigma_c\cap\Sigma_{uc}=\varnothing)$, where Σ_c and Σ_{uc} represent the set of controllable events and the set of uncontrollable events respectively. Likewise the control action can be applied on a system that is partially observable, but in this work it is assumed that all the events are observable.

The control of DES was for the first time explicitly introduced in the work of Wonham and Ramadge (Ramadge and Wonham, 1987b). In this work, the aim of the supervisor is to generate a given language while restricting the system behaviour minimally. Here the supervisor's role is characterized such that at any given system's state, it determines a set of controllable events to be disabled so that the plant evolves over events without violating the specifications. In this perspective the existence of a supervisor is guaranteed if the desired language K satisfies Controllability Condition defined as $\bar{K}\Sigma_{uc}\cap L(G)\subseteq\bar{K}$. But the given language can not be always controllable with respect to Σ_{uc}. Then the idea of obtaining the maximum part of the given language is needed. Here the "maximal" means in terms of set inclusion. This maximum part of K is Supremal Controllable Sublanguage and denoted with $K^{\uparrow C}$ and to compute efficient algorithms is given (Ramadge and Wonham, 1987a; Wonham and Ramadge, 1987). With a similar approach violating the controllability condition, one finds the smallest prefix-closed and controllable language containing K. This language is known as Infimal Prefix-Closed Controllable Superlanguage of K and denoted with $K^{\downarrow C}$. Likewise algorithms also exist for its computation (Lafortune and Chen, 1990).

3. MOTIVATION AND PROBLEM FORMULATION

Sometimes the minimally restrictive nonblocking solution (MRNBS) is deemed inadequate owing to its too restrictive behaviour. In other words, the MRNBS gives a conservative result in the sense that it prevents all uncontrollable events that lead to blocking. As a result this kind of a strategy may constrain the behaviour of the system considerably. Also in some situations the blocking of the controlled system can be easily detected and resolved or for the system to conclude a task is more essential than avoiding the occurrence of a possible blocking. So at the design phase of the supervisor, there is needed to relax the nonblocking requirement. But at this point, the question "How much relaxing?" arises. The motivation of this work is to search a formal answer to this question. This problem has been considered by Lafortune and Chen and they introduced two operators on a set of languages that optimizes the performance measure (Chen and Lafortune, 1991). These two operators optimize the initial solution in the sense of set inclusion so for different initial solutions different incomparable final solutions can occur. Also in practice the interpretation of generated strings by the system are not the same. So the differences between the strings have to be taken into consideration before selecting the "best" supervisor. To overcome these drawbacks in this work a new structure is formulated where different languages generated under the supervision of different supervisors are compared. To fulfil this purpose, a new metric space and a new performance measure is suggested. At the end, to obtain the optimal blocking supervisor a new algorithm is introduced. Also in our previous work the same part of the problem is taken into consideration and to differentiate strings a different metric space is introduced and the result is discussed (Kaymakci and Kurtulan, 2004).

For a given G and the admissible S, the resulting closed loop system is symbolized with S/G. Let the admissible language and admissible marked language be L_a, L_{am} respectively. Then the specifications and trivial assumptions on the controlled language are

$L_m(S/G) := L(S/G) \cap L_m(G)$

$L_m(S/G) \subseteq L_{am}, L(S/G) \subseteq L_a = \overline{L_a} \subseteq L(G)$,

$L_{am} = L_a \cap L_m(G) = L_a \cap L_{am}$ so L_{am} is $L_m(G)$ closed. Then the class of all admissible solutions is

$$L_{cand} := \left\{ K : \left(\overline{L_{am}^{\uparrow C}} \subseteq K \subseteq L_a^{\uparrow C} \cap L_{am}^{\downarrow C} \right) \wedge \left(K = \overline{K} \right) \wedge \left(\overline{K} \Sigma_{uc} \cap L(G) \subseteq \overline{K} \right) \right\}$$

The strings that drive the system to blocking can be represented as a set which is defined as $BM(L(S/G)) := \left\{ L(S/G) \setminus \overline{L_m(S/G)} \right\}$. Also the admissible marked strings, that are not allowed by the controlled system, are denoted with $SM^C(L(S/G)) := \{ L_{am} \setminus L_m(S/G) \}$. These two sets are called Blocking Measure Set and Non-Satisfying Measure Set respectively. For detailed information refer to (Chen and Lafortune, 1991).

At this point, there is no difference between all strings generated by the system. For example, all the strings in the non-satisfying measure set have the same significance. But in practice this is not always true. Sometimes this set may include a very important string which can conclude a very important task. To improve the performance, this kind of a string has to be added to controlled language in a formal way. But the suggested solution by Chen and Lafortune does not give permission for addition of this string if a new blocking arises. This is due to set inclusion. For this purpose, a new performance measure, which gives an opportunity to discriminate the languages, is formulized below.

Definition 1: The *importance* of a generated string is denoted as c where

$s \in \Sigma^*, c : \Sigma^* \setminus \varepsilon \to \mathbb{R}^+$ and $c(\varepsilon) = 0$ ▲

Definition 2: The importance of a language is defined as $\beta(L) : 2^{\Sigma^*} \to \mathbb{R}^+ + \{0\}$ where $L \in 2^{\Sigma^*}$ such that $L = \{s_1, s_2, s_3, \ldots, s_n\}$ .Then

$$\beta(L) := \begin{cases} \sum_{i=1}^{n} c(s_i) & L \neq \varnothing \\ 0 & otherwise \end{cases}$$ ▲

In this way, the significance of a language is obtained in terms of function β but this has no determining factor on languages only gives a numerical value. For comparing different languages a more formal structure has to be formed. So a new metric space for discrete event systems is formulated. For detailed information on metric space refer to (Kreyszig, 1978).

Property 1: If $\forall L_1, L_2 \in 2^{\Sigma^*}$ and $L_1 \subseteq L_2$ then $\beta(L_1) \leq \beta(L_2)$ ▲

Property 2: If $L \in 2^{\Sigma^*}$ and $L = \bigcup_{i=1}^{u} L_i$ where $L_k \cap L_l = \varnothing$ such that $k, l = 1 \ldots u$ and $k \neq l$ then

$\beta(L) = \beta(L_1) + \ldots + \beta(L_u)$ ▲

Property 3: If $\forall L_1, L_2 \in 2^{\Sigma^*}$ then $\beta(L_1 \cup L_2) \leq \beta(L_1) + \beta(L_2)$

The proofs of Property 1, 2 and 3 are trivial and omitted.

Definition 3: The distance function $d : 2^{\Sigma^*} \times 2^{\Sigma^*} \to \mathbb{R}^+ + \{0\}$ is defined in terms of the importance of the language as:

$d(L_1, L_2) := \beta(\{L_1 \setminus L_2\} \cup \{L_2 \setminus L_1\})$ ▲

Proposition 1: The set 2^{Σ^*} and the distance function defined above forms a metric space $\left(2^{\Sigma^*}, d\right)$.

Proof: It will be shown that the axioms of metric space (Kreyszig, 1978) are fulfilled.

- In accordance with the definition of function β, $d(L_1, L_2) \to \mathbb{R}^+ + \{0\}$, $\forall L_1, L_2 \in 2^{\Sigma^*}$
- $d(L_1, L_2) = \beta(\{L_1 \setminus L_2\} \cup \{L_2 \setminus L_1\})$

$d(L_2, L_1) = \beta(\{L_2 \setminus L_1\} \cup \{L_1 \setminus L_2\})$

then $d(L_1, L_2) = d(L_2, L_1)$ $\forall L_1, L_2 \in 2^{\Sigma^*}$

- If $d(L_1, L_2) = 0$ then $\{L_1 \setminus L_2\} \cup \{L_2 \setminus L_1\} = \varnothing$ (by definition of function β) $\Rightarrow L_1 = L_2$

If $L_1 = L_2$ then $d(L_1, L_2) = \beta(\varnothing) = 0$

$d(L_1, L_2) = 0 \Leftrightarrow L_1 = L_2$, $\forall L_1, L_2 \in 2^{\Sigma^*}$

- Assume that $\{L_1 \setminus L_2\} \cup \{L_2 \setminus L_1\} \supset (\{L_1 \setminus L_3\} \cup \{L_3 \setminus L_1\} \cup \{L_3 \setminus L_2\} \cup \{L_2 \setminus L_3\})$.

Let s be an element of $(L_3 \setminus L_1)\cup(L_3 \setminus L_2)$

Then $s \in \left(\{L_1 \setminus L_3\}\cup\{L_3 \setminus L_1\}\cup\{L_3 \setminus L_2\}\cup\{L_2 \setminus L_3\}\right)$

By the assumption, $s \in (L_1 \setminus L_2)\cup(L_2 \setminus L_1)$. But by the definition of s, $s \notin (L_1 \setminus L_2)\cup(L_2 \setminus L_1)$. This is a contradiction; as a result the assumption is not right.

$\{L_1 \setminus L_2\}\cup\{L_2 \setminus L_1\} \subseteq \left(\{L_1 \setminus L_3\}\cup\{L_3 \setminus L_1\}\cup\{L_3 \setminus L_2\}\cup\{L_2 \setminus L_3\}\right)$

$\beta\left(\{L_1 \setminus L_2\}\cup\{L_2 \setminus L_1\}\right) \le \beta\left(\{L_1 \setminus L_3\}\cup\{L_3 \setminus L_1\}\cup\{L_3 \setminus L_2\}\cup\{L_2 \setminus L_3\}\right)$ (by Property 1)

$\beta\left(\{L_1 \setminus L_3\}\cup\{L_3 \setminus L_1\}\cup\{L_3 \setminus L_2\}\cup\{L_2 \setminus L_3\}\right) \le \beta\left(\{L_1 \setminus L_3\}\cup\{L_3 \setminus L_1\}\right) + \beta\left(\{L_3 \setminus L_2\}\cup\{L_2 \setminus L_3\}\right)$ (by Property 3)

Then $d(L_1,L_2) \le d(L_1,L_3) + d(L_3,L_2)\ \forall L_1,L_2,L_3 \in 2^{\Sigma^*}$ ▲

Definition 4: The Non-Satisfying Measure and *Blocking Measure* for $L \in L_{cand}$ are respectively

$\widetilde{SM}^C(L) := d\left(L_{am}, L_m(S/G)\right)$

$\widetilde{BM}(L) := d\left(L(S/G), \overline{L_m(S/G)}\right)$ ▲

Now the blocking measure set and non-satisfying measure set are improved on the defined metric set. Thus the elements of these two sets are not only known but also an opinion about their effects on system can be obtained. It is also clear that only trying to decrease the number of the elements of these two sets is not enough. It is expected the sum of these two performance criterion gives the performance measure of a language.

Definition 5: For $L \in L_{cand}$, the performance measure of the language is defined as: $\tilde{J} := \widetilde{SM}^C(L) + \widetilde{BM}(L)$. ▲

The performance measure denoted is a numerical performance measure so different solutions can be compared. As expected the language that gives the minimum performance measure is the best solution. Then the optimal blocking supervisor problem can be defined as follows:

Definition 6: Let the uncontrolled behaviour of the system be $L(G)$, the performance measure be $\tilde{J}$, and then optimization problem is defined as

$$arg\left\{\min_{L \in L_{cand}} \tilde{J}\right\}$$ ▲

4. OPTIMIZATION ALGORITHM AND AN EXAMPLE

In this section the theoretical foundations of the optimal blocking supervisor will be presented. Let S_{IS} be a given supervisor that symbolizes the initial supervisor such that $L_{IS} = L(S_{IS}/G) \in L_{cand}$. For S_{IS}, $BM(S_{IS})$ and $SM^C(S_{IS})$ will be a finite set as $BM(S_{IS}) = \{\alpha_1, \alpha_2, ..., \alpha_m\}$, $SM^C(S_{IS}) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Lemma 1: If $L \in L_{cand}$ and $\alpha_i \in BM(L)$ then $\alpha_i \notin \overline{(L \setminus \alpha_i)^{\uparrow C}}$.

Proof: $\overline{(L \setminus \alpha_i)^{\uparrow C}} = (L \setminus \alpha_i)^{\uparrow C} \cup R$ where $(L \setminus \alpha_i)^{\uparrow C} \cap R = \varnothing$. Then to be a member of $\overline{(L \setminus \alpha_i)^{\uparrow C}}$, α_i is either a member of $(L \setminus \alpha_i)^{\uparrow C}$ or R. In addition to this it is easily seen that $\alpha_i \notin (L \setminus \alpha_i)^{\uparrow C}$. If $\alpha_i \in R$, there exists a $t \in (L \setminus \alpha_i)^{\uparrow C}$ such that $p_i(t) = \alpha_i, 1 \le i \le \| t-1 \|$. But $\alpha_i \in BM(L)$ so $\alpha_i s \notin L(G)$ such that $s \in \Sigma^*$. So $\alpha_i \notin \overline{(L \setminus \alpha_i)^{\uparrow C}}$ ▲

Proposition 2: If $L \in L_{cand}$ and $\alpha_i, \alpha_k \in BM(L)$ then $\overline{\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \setminus \alpha_k\right]^{\uparrow C}} = \overline{\left[\overline{(L \setminus \alpha_k)^{\uparrow C}} \setminus \alpha_i\right]^{\uparrow C}}$

Proof: For any pair of α_i, α_k, we can write

$\overline{(L \setminus \alpha_i)^{\uparrow C}} \subseteq L$ (because $L \in L_{cand}$)

$\overline{\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \setminus \alpha_k\right]^{\uparrow C}} \subseteq \overline{(L \setminus \alpha_k)^{\uparrow C}}$

$\overline{\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \setminus \alpha_k\right]^{\uparrow C}} \setminus \alpha_i \subseteq \overline{(L \setminus \alpha_k)^{\uparrow C}} \setminus \alpha_i$

$\overline{\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \setminus \alpha_k\right]^{\uparrow C}} \subseteq \overline{(L \setminus \alpha_k)^{\uparrow C}} \setminus \alpha_i$ (by lemma 1)

$\overline{\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \setminus \alpha_k\right]^{\uparrow C}} \subseteq \overline{\left[\overline{(L \setminus \alpha_k)^{\uparrow C}} \setminus \alpha_i\right]^{\uparrow C}}$

Since α_i and α_k are arbitrary elements then

$\overline{\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \setminus \alpha_k\right]^{\uparrow C}} = \overline{\left[\overline{(L \setminus \alpha_k)^{\uparrow C}} \setminus \alpha_i\right]^{\uparrow C}}$ ▲

This result can be extended to a general form with more than two strings in any order. But due to the similar formulation structure, it is not given here. Also for guaranteeing the optimality of the solution, which is obtained with the algorithm given below, these assumptions are given.

- $\tilde{J}\left\{\left[L \setminus \overline{(L \setminus \alpha_i)^{\uparrow C}}\right] \cap \left[L \setminus \overline{(L \setminus \alpha_j)^{\uparrow C}}\right]\right\} = 0$ $\vee \left[L \setminus \overline{(L \setminus \alpha_i)^{\uparrow C}}\right] \supseteq \left[L \setminus \overline{(L \setminus \alpha_j)^{\uparrow C}}\right]$ (A1)
- $\tilde{J}\left\{\left[(L \cup \xi_l)^{\downarrow C} \setminus L\right] \cap \left[(L \cup \xi_m)^{\downarrow C} \setminus L\right]\right\} = 0$ $\vee \left[(L \cup \xi_j)^{\downarrow C} \setminus L\right] \supseteq \left[(L \cup \xi_l)^{\downarrow C} \setminus L\right]$ (A2)
- $\xi_l^{\downarrow C} \cap \alpha_j = \varnothing$ (A3)

where $\alpha_i, \alpha_j \in BM(S_{IS})$, $\xi_l, \xi_m \in SM^C(S_{IS})$

According to the first part of A1 the intersection of strings, which are removed from the language due to two different blockings, have no effect on performance measure. In other words the removed common strings have no influence on blocking or success. Also the removed strings due to one blocking could include same sort of removed strings. Similarly the second assumption possess same type of constraints on languages but this time the concerned strings that drive system not blocking but success. Moreover these assumptions do not possess a too restrictive structure on target languages.

Definition 7: For $\alpha_i, \alpha_j \in BM(L)$ and $L \in L_{cand}$ there exists a transformation $T_1 : L_{cand} \to L_{cand}$ such that

$$T_1(L, \alpha_i) := \begin{cases} \overline{(L \setminus \alpha_i)^{\uparrow C}} & \text{if } \tilde{J}\left[\overline{(L \setminus \alpha_i)^{\uparrow C}}\right] < \tilde{J}[L] \\ L & \text{otherwise} \end{cases} \quad \blacktriangle$$

According to the "if statement" in the definition of T_1, removing a string is bounded to a strict performance improvement. As the blocking set is finite; the transformation gives the best solution in m steps according to blocking measure set. And L_{cand} is a complete lattice set for T_1, so the transformed language is always a member of L_{cand}. Then the existence of final solution is guaranteed by the definition of L_{cand}

Remark 1: As a obvious result of A1 and proposition 2, the following relation always holds. $T_1\left[T_1(L, \alpha_i), \alpha_k\right] = T_1\left[T_1(L, \alpha_k), \alpha_i\right]$. Similarly this can also be extended to more than two words. Then the optimality of final language is guaranteed by Lemma 1 and A1.

Definition 8: For $\xi_j, \xi_l \in SM^C(L)$, $L \in L_{cand}$ then there exist a transformation $T_2 : L_{cand} \to L_{cand}$ such that

$$T_2(L, \xi_j) := \begin{cases} (L \cup \xi_j)^{\downarrow C} & \text{if } \tilde{J}\left[(L \cup \xi_j)^{\downarrow C}\right] < \tilde{J}[L] \\ L & \text{otherwise} \end{cases} \quad \blacktriangle$$

Like T_1, including a string is strictly bounded to performance improvement so in a similar way T_2 gives the best solution if it is applied to non-satisfying measure set completely.

Lemma 2: If $L \in L_{cand}$ and $\xi_j, \xi_l \in SM^C(L)$ then

$$\left[(L \cup \xi_j)^{\downarrow C} \cup \xi_l\right]^{\downarrow C} = \left[(L \cup \xi_l)^{\downarrow C} \cup \xi_j\right]^{\downarrow C} \quad \blacktriangle$$

The proof is straight forward.

Remark 2: As L_{cand} is a complete lattice set, the result of T_2 is always a member of L_{cand} so the existence of the solution is guaranteed. Also according to lemma 2 and the A2, the transformation over non-satisfying measure set gives the optimal solution. As the non-satisfying measure set is finite; the transformation gives the best solution in n steps.

It can be followed that, the transformations T_1 and T_2 deal with the blocking set and non-satisfying set respectively. If these two transformations are used together, an optimal blocking supervisor can be attained with respect to performance measure.

Lemma 3: Let $L \in L_{cand}$, $\alpha_i \in BM(L)$ and $\xi_j \in SM^C(L)$. For $\tilde{J}\left[\overline{(L \setminus \alpha_i)^{\uparrow C}}\right] < \tilde{J}[L]$ and $\tilde{J}\left[(L \cup \xi_j)^{\downarrow C}\right] < \tilde{J}[L]$.

Then $\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \cup \xi_j\right]^{\downarrow C} \subseteq \overline{\left[(L \cup \xi_j)^{\downarrow C} \setminus \alpha_i\right]^{\uparrow C}}$.

Proof:

$\overline{(L \setminus \alpha_i)^{\uparrow C}} \subseteq L$ (by $L \in L_{cand}$)

$\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \cup \xi_j\right]^{\downarrow C} \subseteq (L \cup \xi_j)^{\downarrow C}$

$\overline{(L \setminus \alpha_i)^{\uparrow C}} \cup \xi_j^{\downarrow C} \subseteq (L \cup \xi_j)^{\downarrow C} \setminus \alpha_i$ (by A3)

$\overline{(L \setminus \alpha_i)^{\uparrow C}} \cup \xi_j^{\downarrow C} \subseteq \overline{\left[(L \cup \xi_j)^{\downarrow C} \setminus \alpha_i\right]^{\uparrow C}}$

$\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \cup \xi_j\right]^{\downarrow C} \subseteq \overline{\left[(L \cup \xi_j)^{\downarrow C} \setminus \alpha_i\right]^{\uparrow C}}$ ▲

Remark 3: Since α_i and ξ_j are arbitrary elements and in accordance with definition 1 $\tilde{J}\left[\left[\overline{(L \setminus \alpha_i)^{\uparrow C}} \cup \xi_j\right]^{\downarrow C}\right] \le \tilde{J}\left[\overline{\left[(L \cup \xi_j)^{\downarrow C} \setminus \alpha_i\right]^{\uparrow C}}\right]$ always holds.

Then applying T_1 before T_2, gives a smaller performance measure. Under these three given assumptions, using the two transformations in given order gives the optimal blocking supervisor. This solution can also be applied in an algorithmic structure.

Step 1

- Pick any $L_{IS} \in L_{cand}$ and calculate $L_1 = L_{IS} \cup K_{\max}^{\downarrow C}$ and $L_{FS1} = L_2 = (L_1 \cap L_{am})^{\downarrow C}$ respectively where $K_{\max} := \sup\left\{K : (K \subseteq L_{am} \setminus L_{IS}) ve \left(K^{\downarrow C} \subseteq L_{IS} \cup L_{am}\right)\right\}$
- Find $BM(L_{FS1}) = \{\alpha_1, ..., \alpha_m\}$, $SM^C(L_{FS1}) = \{\xi_1, ..., \xi_n\}$

Step 2

$L_2 = T_1(L_2, \alpha_i)$. Repeat this step for $\forall \alpha_i \in BM(L_{FS1})$

At the end of iteration $L_3 = L_2$

Step 3

$L_3 = T_2(L_3, \xi_j)$. Repeat this step for $\forall \xi_j \in SM^C(L_{FS1})$

At the end of iteration $L_{FS2} = L_3$

Remark 4: Due to T_1 and T_2 are defined on finite sets, the number of iterations needed to arrive an optimal blocking supervisor is

$\left\|BM(L_{FS2})\right\| + \left\|SM^C(L_{FS2})\right\| = m + n$

Example: Consider the generator G in Figure 1. Let $\sum_{uc}=\{\beta_1,\beta_2\}$, $L_{am}=\{\alpha_1\alpha_5,\alpha_1\alpha_2\alpha_8,\alpha_1\alpha_2\alpha_3,\alpha_1\beta_2\alpha_4\}$, $L_a=\overline{L_{am}}\cup\{\alpha_1\alpha_5\beta_1,\alpha_1\alpha_2\beta_1,\alpha_1\alpha_2\alpha_3\beta_1\}$. $c(\alpha_1\alpha_2\alpha_8)=10$, $c(\alpha_1\alpha_2\beta_1)=3$, $c(\alpha_1\alpha_2\alpha_3)=12$, $c(\alpha_1\alpha_2\alpha_3\beta_1)=10$, $c(\alpha_1\alpha_5)=3$, $c(\alpha_1\alpha_5\beta_1)=7$

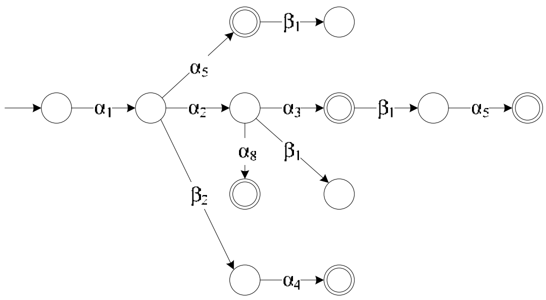

Figure 1

Let $L_{IS}=\left\{\overline{\alpha_1\alpha_5\beta_1},\overline{\alpha_1\beta_2\alpha_4},\overline{\alpha_1\alpha_2\alpha_8},\alpha_1\alpha_2\beta_1\right\}$. Then

$\widetilde{SM^C}(L)=17$, $\widetilde{BM}(L)=10$

$\tilde{J}(L_{IS})=\widetilde{BM}(L_{IS})+\widetilde{SM^C}(L_{IS})=27$

$K_{\max}=\varnothing$ so $L_1=A_{SM}(L_{IS})=L_{IS}$ and $L_{FS1}=A_{BM}(L_1)$ $=L_{IS}$ (L_{FS} denotes the final solution) When the example is solved by Chen and Lafortune's optimization technique and the final solution remains the same. So no change at performance measure occurs. Now the problem will be solved by the search algorithm presented above.

Step 1 L_{IS} is defined.

$BM(L_{IS})=\{\alpha_1\alpha_5\beta_1,\alpha_1\alpha_2\beta_1\}$

$SM^C(L_{IS})=\{\alpha_1\alpha_2\alpha_3\}$, $L_{FS}=L_{IS}$

Step 2

The transformation T_1 will be applied to blocking set.

$T_1(L_{FS},\alpha_1\alpha_5\beta_1)=\left\{\overline{\alpha_1\beta_2\alpha_4},\overline{\alpha_1\alpha_2\alpha_8},\alpha_1\alpha_2\beta_1\right\}$ because $\tilde{J}\left[T_1(L_{FS},\alpha_1\alpha_5\beta_1)\right]<\tilde{J}(L_{FS})$

$T_1(L_{FS},\alpha_1\alpha_2\beta_1)=\left\{\overline{\alpha_1\beta_2\alpha_4},\overline{\alpha_1\alpha_2\alpha_8},\alpha_1\alpha_2\beta_1\right\}$ because $\tilde{J}\left[T_1(L_{FS},\alpha_1\alpha_2\beta_1)\right]>\tilde{J}(L_{FS})$

Step 3

$T_2(L_{FS},\alpha_1\alpha_2\alpha_3)=\left\{\overline{\alpha_1\beta_2\alpha_4},\overline{\alpha_1\alpha_2\alpha_3\beta_1},\alpha_1\alpha_2\alpha_8,\alpha_1\alpha_2\beta_1\right\}$

because $\tilde{J}\left[T_2(L_{FS},\alpha_1\alpha_2\alpha_3)\right]<\tilde{J}(L_{FS})$

Then $L_{FS2}=\left\{\overline{\alpha_1\beta_2\alpha_4},\overline{\alpha_1\alpha_2\alpha_3\beta_1},\alpha_1\alpha_2\alpha_8,\alpha_1\alpha_2\beta_1\right\}$

Also $\tilde{J}(L_{FS2})<\tilde{J}(L_{FS1})$

The optimization algorithm presented above gives a better result according to numerical performance measure.

5. CONCLUSION

In this paper the issue of blocking in supervisory control of DES is studied. A new numerical performance measure is proposed in terms of two different distance functions. Two transformations are presented and using them an optimization algorithm is constructed to minimize the performance measure. This paper contributes a better understanding of the properties of blocking and gives an optimal blocking supervisor in a set of admissible supervisors. This paper is concluded by showing that under the given assumptions the task of finding the optimal blocking supervisor requires $m+n$ iteration steps where n and m refers the number of the elements in $BM(L)$ and $SM^C(L)$, respectively. Future work will be concerned with blocking where besides deadlock livelock is considered. Another issue would be the investigation of relaxing the assumptions on languages.

REFERENCES

Chen, E. and Lafortune, S. (1991). Dealing with Blocking in Supervisory Control of Discrete-Event Systems. *IEEE Transactions on Automatic Control*, **Vol 36(6)**, pages 724-735 .

Fu, J., Ray, A. and Lagoa C. (2004) Unconstrained Optimal Control of Regular Languages, *Automatica*, **Vol 40(4)** submitted for printing.

Kaymakci, Ö. T. and Kurtulan S. (2004) A Metric Space Approach for a Class of Discrete Event Systems. 12th Mediterranean Conference on Control and Automation (MED'04), June 6-9,2004, Kuşadası, Turkey

Kreyszig, E.,(1978) *Introductory functional analysis with applications*, John Wiley & Sons.

Kumar, R. and Garg, V. (1995). Optimal Supervisory Control of Discrete Event Dynamical Systems. *SIAM Journal of Control and Optimization*, **Vol 33(2),** pages 419-439.

Lafortune, S. and Chen, E. (1990) The Infimal Closed Controllable Superlanguage and its Application in Supervisory Control, *IEEE Transactions on Automatic Control* , Vol 35(4) pages398-405

Ramadge, P. J. and Wonham, W. M., (1987a). Modular Feedback Logic for Discrete Event Systems. *SIAM Journal Control and Optimization*, **Vol 25(5),** pages 1202-1218

Ramadge, P. J. and Wonham, W. M., (1987b). Supervisory Control of a Class of Discrete Event Systems. *SIAM Journal Control and Optimization*, **Vol 25(1),** pages 206-230

Sengupta, R. and Lafortune, S. (1998). An Optimal Control Theory for Discrete Event Systems. *SIAM Journal of Control and Optimization*, **Vol 36(2),** pages 488-541.

Surana, A. and Ray, A. Measure of Regular Languages. Discrete Event Dynamic Systems: Theory and Applications, submitted for printing.

Wonham, W.M. and Ramadge, P.J. On the Supremal Controllable Sublanguage of a given language, SIAM J. Control Optimization Vol. 25 , No. 3, pp 637-659, May 1987