AN IMPLEMENTATION OF AN OPTIMAL BLOCKING SUPERVISOR FOR DISCRETE EVENT SYSTEMS

Özgür Turay Kaymakçı Salman Kurtulan
e-mail: kaymakci@elk.itu.edu.tr *e-mail: kurtulan@elk.itu.edu.tr*
Istanbul Technical University, Faculty of Electrics & Electronics, Department of Electrical Engineering, 34390 Maslak, Istanbul, Turkey

Key words: Discrete Event Systems, Optimal Blocking Supervisor, Blocking, Supervisory Control Theory

ABSTRACT

In this study, the theoretical background of a recently proposed new algorithm for finding an optimal blocking supervisor for discrete event systems is proposed. Afterwards an implementation of it is given. The recently proposed new algorithm, which tries to find an optimal blocking supervisor, depends on a new cost function over language and a new numerical performance measure. As a result at the end of the execution of the algorithm an optimal solution is always obtained.

I. INTRODUCTION

From a planning and control perspective, the manufacturing systems, which consist of robots, actuators, sensors, motors, etc., can be seen as a dynamical system whose states evolve according to the occurrence of physical events , thus can be classified as event driven systems or discrete event systems (DESs). In the past due to its simplicity of these systems, the intuitive methods or ad-hoc solutions have been adequate. However, the increasing complexities of these systems and the detailed system requirements have created a need for formal approaches for analysis and control of these kinds of systems. In this respect the theoretical background of discrete event systems and supervisory control theory was established.

Generally discrete event systems are modeled as regular languages that can be represented by finite state automata. Usually the behaviour of the system is not satisfactory and must be "modified" by a control. In this perspective, supervisory control theory (SCT) gives us the possibility to build in the modifications via supervisor as much as possible. The supervisory control theory (SCT) was firstly introduced by Ramadge and Wonham[10]. The theory enables the synthesis of closed loop control for DESs by making some assumptions on the system that is to be controlled, and on the supervisor that is to control the system. Although the SCT has received a wide acceptance in academic communities and also some applications of SCT have been reported in the literature, it has not been accepted in the industry yet. This is mainly due to the difficulties arising in physical implementation of SCT. The state explosion problem is good example for this class.

When we focus on the industrial examples, all the desired strings are not usually marked by the supervisor that is built according to nonblocking and controllability conditions. The designed supervisor could constitute a conservative solution due to preventing all probable blockings in the system. So one may be willing to risk blocking if there will be serious increase in the marked strings. Therefore to obtain the "best" supervisor, a new approach has to be improved so a new performance measure and optimization procedure is developed.

In the literature several researchers have interested in optimization problem for discrete event systems and also have proposed optimal control of DES in different approaches. The milestones of solving this problem in the literature are as follows: Chen and Lafortune were interested in blocking in discrete event systems and introduced two operators on a set of languages that optimizes the performance measure in the sense of set inclusion [2]. Passino and Antsaklis proposed a cost function on the set of transitions and put a control objective for restricting the plant behaviour in such a way that optimal control of a discrete event system is equivalent to follow a trajectory of optimal cost [9]. Like Passino and Antsaklis, Brave and Heymann also introduced a cost function on the set of transitions and dealed with the optimal control problem as an optimal attractor problem. Different from them, the problem was formulated not only for a fixed single initial state but for any initial states[1]. 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. In this work Kumar and Garg assumed that a certain state may be visited only once, so the corresponding transitions are controlled only once [7]. Sengupta and Lafortune also used control and event costs to find the optimal nonblocking supervisor and solved the problem by dynamic programming [12]. Also this study constitutes the cornerstone of the recently done works on optimization in discrete event systems. Later Surana and Ray constructed a signed measure over discrete event systems and the measure is defined on state transitions. Also Fu and Ray used this signed measure for formulating an unconstrained optimal control policy. The policy is obtained by selectively disabling controllable events to maximize the measure [3]. These strategies have addressed performance enhancement of discrete event systems but none of them were interested in finding the optimal blocking supervisor with a unique numerical performance measure.

This paper introduces the recently proposed an optimization algorithm to find the optimal blocking supervisor also gives an implementation of it. The paper is organized in 4 sections. Section 2 briefly gives the preliminaries. Section 3 describes the motivation, algorithm and implementation of it through an example. The paper is summarized and concluded in section 4.

II. PRELIMINARIES

When the literature is investigated, it can be easily seen that there are more than one description for discrete event systems but the mostly accepted one is as follows: 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 $G=\left(X,\Sigma,f,\Gamma,x_0,X_m\right)$ where X is the set of states, Σ is the finite set of events, $f: X\times\Sigma\rightarrow X$ is the state transition function, $\Gamma: X\rightarrow 2^{\Sigma}$ 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 $L(G)=\left\{s\in\Sigma^* \,\middle|\, f^*\left(x_0,s\right)\in X\right\}$ 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: $f^*: X\times\Sigma^*\rightarrow X$. $L(G)$ 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 [12].

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 $\Gamma\left(x_i\right)=\varnothing\ \left(x_i\notin X_m\right)$ 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 [8]. 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 $\Sigma=\Sigma_c\cup\Sigma_{uc}\left(\Sigma_c\cap\Sigma_{uc}=\varnothing\right)$, 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 [11]. 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 [10, 11]. 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 [8]

III. PROBLEM FORMULATION AND AN EXAMPLE

When the desired marked language is taken into consideration, the minimally restrictive nonblocking solution (MRNBS) is sometimes 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 consequence, this kind of a strategy may constrain the behaviour of the system significantly. Sometimes in some situations the blocking in the system can be easily detected and resolved. Moreover for the system to conclude a task may be more essential than avoiding the occurrence of a possible blocking. In this sense at the design phase of the supervisor, a relaxation in the nonblocking requirement is needed for obtaining a more functional supervisor. But at this point, the question "How much relaxing?" arises. In the past this problem had been considered by Lafortune and Chen and then they introduced two operators on a set of languages that optimizes the performance measure [2]. But by these two operators the optimal language is obtained in the sense of set inclusion. Also for different initial solutions different incomparable final solutions can occur. Moreover 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 our previous works a new metric space and performance measure are introduced [4]. And for obtaining the best result a new algorithm is proposed [5].

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, which are not allowed by the controlled system, are denoted with $SM^C(L(S/G)) := \left\{ L_{am} \setminus L_m(S/G) \right\}$. These two sets are called Blocking Measure Set and Non-Satisfying Measure Set respectively [2]

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 kind of an assumption is not always true. Sometimes this set may include a very important string which concludes 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 due to set inclusion, the suggested solution by Chen and Lafortune does not give permission for addition of this string if a new blocking arises.

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_i where

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

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

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

With these two definitions given above, the discrimination of the strings and the languages are generated. So the worth of a language is obtained in terms of function β but when it is investigated deeply, it is easily seen that it has no determining factor on languages only gives a numerical value. Therefore 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 [6].

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:

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

Then the set 2^{Σ^*} and the distance function defined above forms a metric space $\left(2^{\Sigma^*}, d\right)$. The blocking measure set and non-satisfying measure set which were defined by Chen and Lafortune can be easily improved on defined metric set.

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

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

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

With these two definitions given above, not only the elements of these two sets are not known but also an opinion about their effects on the system can be obtained. By them it is clear that only trying to decrease the number of the elements of these two sets is not enough. Intuitively it is easily seen that the sum of these two performance criterions gives the performance measure of the system in this manner.

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

Different solutions can be compared because the denoted performance measure is a numerical performance measure. As expected the language that gives the minimum performance measure is the best solution. Sometimes pareto optimal solutions can be observed. But here this kind of a solution set is not in our consideration. 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 $arg\left\{\min_{L\in L_{cand}} \tilde{J}\right\}$ ▲

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 $BM(S_{IS}) = \{\alpha_1, \alpha_2, ..., \alpha_m\}$, $SM^C(S_{IS}) = \{\xi_1, \xi_2, ..., \xi_n\}$.

For guaranteeing the solution to be optimal below assumption are given.

To $\tilde{J}\left\{\left[L\setminus\overline{(L\setminus\alpha_i)^{\uparrow C}}\right]\cap\left[L\setminus\overline{(L\setminus\alpha_j)^{\uparrow C}}\right]\right\}=0$

$$\vee\left[L\setminus\overline{(L\setminus\alpha_i)^{\uparrow C}}\right]\supseteq\left[L\setminus\overline{(L\setminus\alpha_j)^{\uparrow C}}\right]$$

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

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

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}. For guaranteeing the uniqueness of the solution the below assumption is given

.

The first part of the assumption simply tells us that the intersection of strings, that 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 at the second part, it is told that the strings that are removed due to one blocking can include same sort of removed strings. Moreover this assumption does not possess a too restrictive structure on target languages.

Remark 1: A few more words for existence and uniqueness of the solution: The existence is guaranteed by the definition of L_{cand} and the uniqueness of the solution is obtained by Lemma 1 and the assumption given above. As a straight result 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.

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

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

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

In a similar way, in the name of guaranteeing the uniqueness of solution a new but similar assumption is included.

$$\tilde{J}\left\{\left[(L\cup\xi_j)^{\downarrow C}\setminus L\right]\cap\left[(L\cup\xi_l)^{\downarrow C}\setminus L\right]\right\}=0$$

$$\vee\left[(L\cup\xi_j)^{\downarrow C}\setminus L\right]\supseteq\left[(L\cup\xi_l)^{\downarrow C}\setminus L\right]$$

Remark 2: L_{cand} is also a complete lattice set for T_2. So the existence of the solution is guaranteed. Also in accordance with lemma 2 and the given assumption, the transformation gives a unique solution over non-satisfying measure set.

As it is seen, 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.

Remark 3: Since α_i and ξ_j are arbitrary elements so always $\tilde{J}\left[\left[\overline{(L\setminus\alpha_i)^{\uparrow C}}\cup\xi_j\right]^{\downarrow C}\right] \le \tilde{J}\left[\overline{\left[(L\cup\xi_j)^{\downarrow C}\setminus\alpha_i\right]^{\uparrow C}}\right]$ holds. Then applying T_1 before T_2, gives a smaller performance measure. Then using these two transformations in shown order gives us 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 $L_{FS} = (L_1 \cap L_{am})^{\downarrow C}$ 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 $BM(L_{FS}) = \{\alpha_1, ..., \alpha_m\}$, $SM^C(L_{FS}) = \{\xi_1, ..., \xi_n\}$

Step 2

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

Step 3

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

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 $\left\|BM(L_2)\right\| + \left\|SM^C(L_2)\right\| = m+n$

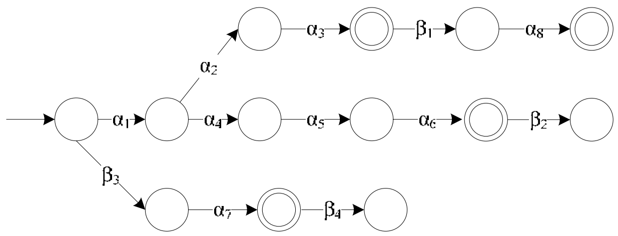

Figure 1

Example: Consider the automata G in Figure 1. Let $\Sigma_{uc} = \{\beta_1, \beta_2, \beta_3, \beta_4\}$, $L_{am} = \{\alpha_1\alpha_2\alpha_3, \alpha_1\alpha_4\alpha_5\alpha_6, \beta_3\alpha_7\}$, $L_a = \overline{L_{am}} \cup \{\alpha_1\alpha_2\alpha_3\beta_1, \alpha_4\alpha_5\alpha_6\beta_2, \beta_3\alpha_7\beta_4\}$. $c_i(\alpha_1\alpha_2\alpha_3) = 7$, $c_i(\alpha_1\alpha_2\alpha_3\beta_1) = 10$, $c_i(\alpha_1\alpha_4\alpha_5\alpha_6) = 10$, $c_i(\alpha_1\alpha_4\alpha_5\alpha_6\beta_2) = 2$ $c_i(\beta_3\alpha_7) = 5$, $c_i(\beta_3\alpha_7\beta_4) = 7$

Let $L_{IS} = \left\{\overline{\alpha_1\alpha_2\alpha_3\beta_1}, \overline{\alpha_1\alpha_4\alpha_5\alpha_6\beta_2}, \overline{\beta_3\alpha_7\beta_4}\right\}$. Then

$\widetilde{SM}^C(L) = 0$, $\widetilde{BM}(L) = 19$

$\tilde{J}(L_{IS}) = \widetilde{BM}(L_{IS}) + \widetilde{SM}^C(L_{IS}) = 19$

$K_{\max} = \varnothing$ 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.

$BM(L_{IS}) = \{\alpha_1\alpha_2\alpha_3\beta_1, \alpha_1\alpha_4\alpha_5\alpha_6\beta_2, \beta_3\alpha_7\beta_4\}$

$SM^C(L_{IS}) = \varnothing$, $L_{FS} = L_{IS}$

Step 2

The transformation T_1 will be applied to blocking set.

$T_1(L_{FS}, \alpha_1\alpha_2\alpha_3\beta_1) = \left\{\overline{\alpha_1\alpha_4\alpha_5\alpha_6\beta_2}, \overline{\beta_3\alpha_7\beta_4}\right\}$ because $\tilde{J}\left[T_1(L_{FS}, \alpha_1\alpha_5\beta_1)\right] < \tilde{J}(L_{FS})$

$T_1(L_{FS}, \alpha_1\alpha_4\alpha_5\alpha_6\beta_2) = \left\{\overline{\alpha_1\alpha_4\alpha_5\alpha_6\beta_2}, \overline{\beta_3\alpha_7\beta_4}\right\}$ because $\tilde{J}\left[T_1(L_{FS}, \alpha_1\alpha_2\beta_1)\right] > \tilde{J}(L_{FS})$

$T_1(L_{FS}, \beta_3\alpha_7\beta_4) = \left\{\overline{\alpha_1\alpha_4\alpha_5\alpha_6\beta_2}, \overline{\beta_3\alpha_7\beta_4}\right\}$ because $\tilde{J}\left[T_1(L_{FS}, \alpha_1\alpha_2\beta_1)\right] > \tilde{J}(L_{FS})$

Step 3

There is not any iteration step in this part because $SM^C(L_{IS}) = \varnothing$

Then $L_{FS2} = \left\{\overline{\alpha_1\alpha_4\alpha_5\alpha_6\beta_2}, \overline{\beta_3\alpha_7\beta_4}\right\}$

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

IV. CONCLUSION

The aim of the paper is to study the blocking in supervisory control of DES. For this purpose in our recent work a new performance measure was proposed and two different distance functions were given. Also a new optimization algorithm was suggested. This paper contributes a better understanding of the properties of blocking and gives a unique optimal blocking supervisor among a set of admissible supervisors. This paper is concluded by an example showing that the task of finding the optimal blocking supervisor

IV. REFERENCES

1. Y. Brave and M. Heymann, On Optimal Attraction in Discrete Event Processes. Proc. First European Control Conference, Grenoble, France, July 2-5, 1998
2. E. Chen and S. Lafortune. Dealing with Blocking in Supervisory Control of Discrete-Event Systems. *IEEE Transactions on Automatic Control*, Vol 36(6), pages 724-735, 1991
3. J. Fu, A. Ray and C. Lagoa, Unconstrained Optimal Control of Regular Languages, *Automatica*, Vol 40(4), 2004
4. Ozgur T. Kaymakçı and S. Kurtulan. 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
5. O. Kaymakci, S. Kurtulan. Improving the Behaviour Under Blocking, 16th IFAC World Congress, 4-8 July 2005, Prag, Çek Cumhuriyeti
6. E. Kreyszig, Introductory functional analysis with applications, John Wiley & Sons, 1978.
7. R. Kumar and V. Garg, Optimal Supervisory Control of Discrete Event Dynamical Systems. *SIAM Journal of Control and Optimization*, Vol 33(2), pages 419-439, 1995.
8. S. Lafortune and E. Chen, The Infimal Closed Controllable Superlanguage and its Application in Supervisory Control, *IEEE Transactions on Automatic Control*, Vol 35(4) pages398-405, 1990
9. K. M. Passino and P.J. Antsaklis, On Optimal Control of Discrete Event Systems. IEEE Decision and Control Conference, Tampa, Florida. Pages 2713-2718, 1989.
10. P. J. Ramadge and W. M. Wonham, Modular Feedback Logic for Discrete Event Systems. *SIAM Journal Control and Optimization*, Vol 25(5), pages 1202-1218, 1987
11. P. J. Ramadge and W. M. Wonham, Supervisory Control of a Class of Discrete Event Systems. *SIAM Journal Control and Optimization*, Vol 25(1), pages 206-230, 1987
12. R. Sengupta and S. Lafortune, An Optimal Control Theory for Discrete Event Systems. *SIAM Journal of Control and Optimization*, Vol 36(2), pages 488-541, 1998.