Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

On the Design of Optimal and Robust Supervisors for Deterministic Finite State Automata

Constantino M. Lagoa

Abstract— **In this paper, the problem of optimal and robust controller design for finite state automata is addressed. The approach presented is based on the language measure introduced in Wang and Ray. However, it differs from previous approaches to optimal controller design by using a new definition of the performance of the supervised automaton. This new definition is, in our opinion, more appropriate in cases where the performance weights are related to the relative frequency of the events.**

I. Introduction

The main motivation for the problem addressed in this paper is the recent development of quantitative language measures and their use in supervisor design; see [1], [2], [3], [4], [5]. In this early work, the performance of supervised plants was assessed by discarding the weights of the events that were disabled by the supervisor. Algorithms for optimal and robust supervisor design were developed based on this way of assessing the performance of the supervised plant.

However, there are cases in which simply discarding the weights of the disabled events is not the best approach. Namely, if one interprets the performance weights of a given event as being related to the relative frequency at which this specific event occurs then, in our opinion, when an event is disabled its weight should be "distributed" among the events that are still enabled. In other words, to have a consistent relation between event weights and the frequency of events, when one event is disabled, all the weights in the language measure should be modified accordingly.

Given the motivation above, in this paper we propose a new way of addressing the performance of a supervised deterministic finite state automaton (DFSA) which addresses cases like the one mentioned above. Also, given that in many instances one only has estimates of the relative frequency of events, an algorithm is presented which, given bounds on the uncertainty of estimates, converges to the worst–case performance. Moreover, an algorithm for optimal robust supervisor design is presented; i.e, a procedure is presented which converges to a supervisor that maximizes the worst–case performance of the supervised plant.

A. Previous Work

The problem of robust control of discrete-event dynamical systems (DEDS) has been addressed by several researchers. Park and Lim [6] have studied the problem of robust control of nondeterministic DEDS. The performance measure used was nonblocking property of the supervised plant. Necessary and sufficient conditions for existence of a robust nonblocking controller for a given finite set of plants are provided. However, no algorithm for controller design is provided. The problem of designing nonblocking robust controllers was also addressed by Cury and Krogh [7] with the additional constraint that the infinite behavior belongs to a given set of allowable behaviors. In this work, the authors concentrated on the problem of designing a controller that maximizes the set of plants for which their supervised behavior belong to the admissible set of behaviors. Takai [8] addresses a similar problem. However, it considers the whole behavior (not just the infinite behavior) and it does not consider nonblockingness. Lin [9] adopted a different approach, where both the set of admissible plants and the performance are defined in terms of the marked language. Taking the set of admissible plants as the plants whose marked language is in between two given the behaviors, the authors provided conditions for solvability of the problem of designing a discrete event supervisory controller such that the supervised behavior of any of the admissible plants contains a desired behavior K.

To address a subject related to that of this paper several researchers have proposed optimal control algorithms for deterministic finite state automata (DFSA) based on different assumptions. Some of these researchers have attempted to quantify the controller performance using different types of cost assigned to the individual events. Passino and Antsaklis [10] proposed path costs associated with state transitions and hence optimal control of a discrete event system is equivalent to following the shortest path on the graph representing the uncontrolled system. Kumar and Garg [11] introduced the concept of payoff and control costs that are incurred only once regardless of the number of times the system visits the state associated with the cost. Consequently, the resulting cost is not a function of the dynamic behavior of the plant. Brave and Heymann [12] introduced the concept of optimal attractors in discrete-event control. Sengupta and Lafortune [13] used control cost in addition to the path cost in optimization of the performance index for trade-off between finding the shortest path and reducing the control cost.

A limitation of the work mentioned above is that the controllers are designed so that the closed loop system has certain specified characteristics. No performance measure is given that can compare the performance of different controllers. To address this issue, Wang and Ray [1] and

This work has been supported by the National Science Foundation under grants ECS-9984260 and ANI-0125653.

Constantino Lagoa is with Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802, USA lagoa@engr.psu.edu.

Ray and Phoha [2] have proposed a signed real measure for regular languages. This novel tool of addressing the performance of DFSAs enable the developing of a new approach to supervisor design. The design of optimal supervisor has been reported by Fu, Ray and Lagoa in [3] and [4] for without and with event disabling cost, respectively. Moreover, the design of robust supervisors has been addressed in Fu, Lagoa and Ray in [5]. Although based on these early results, the approach taken in this paper differs on the way that the performance of the supervised plant is assessed. It provides, in our opinion, a better approach when the performance weights are related to the relative frequency of events.

II. Brief Review of the Language Measure

This section briefly reviews the concept of signed real measure of regular languages introduced in [1]. Let the plant behavior be modelled as a deterministic finite state automaton (DFSA) $G_i \equiv (Q, \Sigma, \delta, q_i, Q_m)$ where Q is the finite set of states with $|Q| = n$ excluding the dump state [14] if any, and $q_i \in Q$ is the initial state; Σ is the (finite) alphabet of events; Σ^* is the set of all finite-length strings of events including the empty string ϵ; the possibly partial) function $\delta : Q \times \Sigma \to Q$ represents state transitions and $\hat{\delta^*} : Q \times \Sigma^* \to Q$ is an extension of δ; and $Q_m \subseteq Q$ is the set of marked states. The set Q_m is partitioned into Q_m^+ and Q_m^-, where Q_m^+ contains all *good* marked states that are desired to terminate on and Q_m^- contains all *bad* marked states that are not desired to terminate on, although it may not always be possible to avoid terminating on the bad states while attempting to reach the good states. The marked language associated with DFSA G_i $L_m(G_i)$ is partitioned into $L_m^+(G_i)$ and $L_m^-(G_i)$ consisting of good and bad strings that, starting from q_i, terminate on Q_m^+ and Q_m^-, respectively.

The language of all strings that, starting at a state $q_i \in Q$, terminates on a state $q_j \in Q$, is denoted as $L(q_i, q_j)$. That is, $L(q_i, q_j) \equiv \{s \in L(G_i) : \hat{\delta^*}(q_i, s) = q_j\}$. Furthermore, a consider a characteristic function $\chi : Q \to [-1,\ 1]$ satisfying

$$\chi(q_j) \in \begin{cases} (0,1] & \text{if } q_j \in Q_m^+ \\ \{0\} & \text{if } q_j \notin Q_m \\ [-1,0) & \text{if } q_j \in Q_m^- \end{cases}$$

Now, the event performance weight $\tilde{\pi} : \Sigma^* \times Q \to [0,1]$ is defined as

- $\tilde{\pi}[\sigma_k|q_j] = 0$ if $\delta(q_j, \sigma_k)$ is undefined; $\tilde{\pi}[\epsilon|q_j] = 1$;
- $\tilde{\pi}[\sigma_k|q_j] \equiv \tilde{\pi}_{jk} \in [0,1)$; $\sum_k \tilde{\pi}_{jk} < 1$;
- $\tilde{\pi}[\sigma_k\ s|q_j] = \tilde{\pi}[\sigma_k|q_j]\tilde{\pi}[s|\delta(q_j, \sigma_k)]$.

Given this, the signed real measure μ of a singleton string set $\{s\} \subset L(q_i, q_j) \subseteq L(G_i) \in 2^{\Sigma^*}$ is defined as:

$$\mu(\{s\}) \equiv \chi(q_j)\, \tilde{\pi}(s|q_i)\ \forall s \in L(q_i, q_j).$$

The signed real measure of $L(q_i, q_j)$ is defined as

$$\mu\left(L(q_i, q_j)\right) \equiv \sum_{s \in L(q_i,q_j)} \mu\left(\{s\}\right)$$

and the signed real measure of a DFSA G_i, initialized at the state $q_i \in Q$, is denoted as:

$$\mu_i \equiv \mu(L(G_i)) = \sum_j \mu\left(L(q_i, q_j)\right)$$

Taking $\mu \equiv [\mu_1\ \mu_2\ \cdots\ \mu_n]^T$, it was proven in [1] that

$$\mu = \Pi\mu + X$$

where Π is an $n \times n$ matrix whose (j,k) entry is

$$\pi_{jk} \equiv \pi(q_k|q_j) = \sum_{\sigma \in \Sigma : \delta(q_j,\sigma) = q_k} \tilde{\pi}(\sigma|q_j)$$

and

$$\pi_{jk} = 0 \text{ if } \{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k\} = \emptyset$$

and $X \equiv [\chi_1\ \ \chi_2\ \cdots\ \chi_n]^T$. So that the vector μ is well defined, it is assumed that there exist a $0 < \theta < 1$ such that, for all i

$$\sum_j \pi_{ij} = (1 - \theta).$$

The constant θ is related to the relative weight of short and long strings. If θ is close to zero the language measure depends mostly on "short" strings of events. The closer θ is to one, the higher the contribution of "long" strings to the language measure.

Remark: It should be noted that if one defines the function (or operator) $T : \Re^n \to \Re^n$

$$T(x) \equiv \Pi x + X.$$

finding the language measure μ is equivalent to finding the fixed point of T. This observation plays an important role in the proofs of the results in this paper

III. Performance of a Supervised plant

The main difference between the approach taken in this paper and previous ones resides on how one computes the performance of a supervised plant. In [3], [4], [5], if an event is disabled, then one would just associate the value zero with its corresponding weight. More precisely, if event σ_k is disabled when the current state of the automaton is q_j, then one would take

$$\tilde{\pi}[\sigma_k|q_j] = 0$$

without any modification of other event weights. However, in the case where $\tilde{\pi}[\sigma_k|q_j]$ is related to the relative frequency that event σ_k occurs given that one has "left" state q_j, then one has to take a different approach. In this case, when event σ_k is disabled, its weight should be distributed among the events that are currently enabled. One should note that this approach is only applicable to supervised plants that do not have any deadlock states. Hence, we now define the class of allowable supervisors.

A. Class of Allowable Supervisors $\mathcal{S}$

Let G be the open loop plant and G^S be the supervised plant. The set of allowable supervisors is

$$\mathcal{S} \doteq \{S : G^S \text{ has no deadlock states}\}.$$

B. Performance Weights of Supervised Plant

Assume that at state q_j the set of events $\mathcal{D}_j^S$ are disabled by supervisor S. Moreover, denote by $\tilde{\pi}^S[\sigma_k|q_j]$ the event performance weights for the supervised system. Then,

$$\tilde{\pi}^S[\sigma_k|q_j] = \begin{cases} 0 & \text{if } \sigma_k \in \mathcal{D}_j^S \\ p_j^S \tilde{\pi}[\sigma_k|q_j] & \text{otherwise} \end{cases}$$

where

$$p_j^S = \frac{1-\theta}{\sum_{\sigma_k \notin \mathcal{D}_j^S} \tilde{\pi}[\sigma_k|q_j]}$$

The performance of the supervised plant μ^S is computed using the same approach as before. More precisely,

$$\mu^S = \Pi^S \mu^S + X$$

where Π^S is an $n \times n$ matrix whose (j,k) entry is

$$\pi_{jk}^S = \pi^S(q_k|q_j) = \sum_{\sigma \in \Sigma : \delta^S(q_j,\sigma)=q_k} \tilde{\pi}^S(\sigma|q_j)$$

and

$$\pi_{jk}^S = 0 \text{ if } \{\sigma \in \Sigma : \delta^S(q_j,\sigma) = q_k\} = \emptyset$$

and δ^S is the state transition function for the supervised system. Note that one still satisfies

$$\sum_j \pi_{ij}^S = (1-\theta).$$

IV. Uncertainty and Robust Performance

As mentioned in Section I, in many instances, the value of the performance weights is not exactly known. In this section, a way of addressing this problem is presented. A definition of the uncertainty structure addressed in this paper is presented. Moreover, an algorithm for computing worst-case performance is provided.

A. Uncertainty Structure

In this paper, it is assumed that one does not know the exact value of the performance weights. Only bounds are available. More precisely, it is assumed that

$$\tilde{\pi}[\sigma_k|q_j](\Delta) = \tilde{\pi}_0[\sigma_k|q_j](1 + \Delta_{\sigma_k|q_j})$$

where

$$\tilde{\pi}_0[\sigma_k|q_j] = \begin{cases} > 0 & \text{if } \{q : \delta(\sigma_k, q_j) = q\} \neq \emptyset; \\ = 0 & \text{otherwise.} \end{cases}$$

The admissible set for the uncertainty is

$$\boldsymbol{\Delta} = \left\{ \Delta : 0 < \Delta_{\sigma_k|q_j}^{\min} \leq \Delta_{\sigma_k|q_j} \leq \Delta_{\sigma_k|q_j}^{\max} \text{ and } \sum_{\sigma_k} \tilde{\pi}[\sigma_k|q_j](\Delta) = 1-\theta, j = 1,2,\ldots,n \right\}.$$

In other words, it is only known that each of the weights belongs to a given interval and that the sum of the weights corresponding to a given state is equal to $1-\theta$.

B. Uncertain Supervised Plant

As mentioned before, when a supervisor disables a subset of the events, the weights of the events are "distributed" among the events that have not been disabled. Hence, given uncertainty $\Delta \in \boldsymbol{\Delta}$ and a supervisor S which at state q_j disables a set of events $\mathcal{D}_j^S$, the performance weights of the supervised plant are defined as

$$\tilde{\pi}^S[\sigma_k|q_j](\Delta) = \begin{cases} 0 & \text{if } \sigma_k \in \mathcal{D}_j^S \\ p_j^S(\Delta)\tilde{\pi}[\sigma_k|q_j](\Delta) & \text{otherwise} \end{cases}$$

where

$$p_j^S(\Delta) = \frac{1-\theta}{\sum_{\sigma_k \notin \mathcal{D}_j} \tilde{\pi}[\sigma_k|q_j](\Delta)}$$

C. Additional Notation

Given a supervisor S, let $\Pi(S,\Delta)$ be the uncertain state transition matrix under supervisor S, i.e., $\Pi(S,\Delta)$ has entries

$$\pi_{ij}(S,\Delta) = \sum_{\sigma \in \Sigma : \delta(q_i,\sigma)=q_j} \tilde{\pi}^S[\sigma|q_i](\Delta).$$

For a given admissible value of the uncertainty $\Delta \in \boldsymbol{\Delta}$, the performance of the plant under the supervisor S, denoted by $\mu(S,\Delta)$, is the solution of

$$\mu(S,\Delta) = \Pi(S,\Delta)\mu(S,\Delta) + X.$$

D. Robust Performance

Consider an uncertain automaton controlled by a supervisor S. The *robust performance of supervisor* S, denoted by $\underline{\mu}(S)$ is defined as the worst-case performance, i.e.,

$$\underline{\mu}(S) = \min_{\Delta \in \boldsymbol{\Delta}} \mu(S,\Delta)$$

where the above minimum is taken elementwise. Even though the minimization is done element by element, this performance is achieved for some $\Delta^* \in \boldsymbol{\Delta}$. The precise statement of this result is given below and its proof is provided in Section V-B.

Lemma 1: Let S be a supervisor. Then, there exists a $\Delta^* \in \boldsymbol{\Delta}$ such that, for all admissible $\Delta \in \boldsymbol{\Delta}$,

$$\underline{\mu}(S) = \mu(S,\Delta^*) \leq \mu(S,\Delta)$$

where the above inequality is implied elementwise.

An algorithm for computing $\underline{\mu}(S)$ is presented below.

Algorithm 1: Computation of worst–case performance of supervisor S.

Step 0. Let $k = 0$ and select $\Delta^0 \in \boldsymbol{\Delta}$. Let *eps* be the desired precision level.

Step 1. Let $\Pi_i(S,\Delta)$ denote the i-th row of the matrix $\Pi_i(S,\Delta)$. Find Δ^{k+1} such that for $i = 1,2,\ldots,n$[1]

$$\Pi_i(S,\Delta^{k+1})\mu(S,\Delta^k) = \min_{\Delta \in \boldsymbol{\Delta}} \Pi_i(S,\Delta)\mu(S,\Delta^k).$$

[1]Note that this such a Δ^{k+1} can always be found since the uncertainty in each entry row of the matrix $\Pi(S,\Delta)$ is independent of the uncertainty in the other rows.

Step 2. If $\|\mu(S,\Delta^{k+1})-\mu(S,\Delta^k)\|_\infty < eps$ stop. Else let $k \leftarrow k+1$ and go to Step 1

E. Comments on Numerical Implementation

Note that the algorithm above requires solving several optimization problems in Step 1. Although not convex, these optimization problems are quasi-convex and can be easily solved using a bisection algorithm. We now elaborate on this.

In Step 1. of the algorithm one is required to solve a problem of the form

$$\min_{\Delta\in\mathbf{\Delta}} \Pi_i(S,\Delta)\mu.$$

Given the definition of $\Pi_i(S,\Delta)$ and the uncertainty admissible set, this problem is equivalent to

$$\min \nu$$

subject to

$$\sum_j \sum_{\sigma\in\Sigma,\sigma\notin\mathcal{D}_i^S:\delta(q_i,\sigma)=q_j} \tilde{\pi}^S[\sigma|q_i](\Delta)\mu_j \leq \frac{\nu}{1-\theta}\sum_{\sigma\notin\mathcal{D}_i^S}\tilde{\pi}[\sigma_k|q_j](\Delta)$$

$$\sum_{\sigma_k}\tilde{\pi}[\sigma|q_i](\Delta) = 1-\theta$$

$$\tilde{\pi}[\sigma|q_i](\Delta) = \tilde{\pi}_0[\sigma|q_i](1+\Delta_{\sigma|q_i}); \quad \Delta_{\sigma|q_i}^{\min} \leq \Delta_{\sigma|q_i} \leq \Delta_{\sigma|q_i}^{\max}$$

which can easily solved using a bisection algorithm to search for ν and solving a linear inequality feasibility problem at each of the iterations.

F. Convergence of Algorithm 1

We now establish the convergence of the algorithm for computing worst–case performance.

Theorem 1: Given a supervisor S, Algorithm 1 converges to its robust performance, i.e., $\mu(S,\Delta^k)\rightarrow \underline{\mu}(S)$. Furthermore, if there exists an $\varepsilon>0$ such that for $i\neq j$ $|\underline{\mu}_i(S)-\underline{\mu}_j(S)|>\varepsilon$ then the algorithm converges in a finite number of iterations. More precisely, it will converge to the robust performance after at most N iterations where

$$N = \left\lceil \frac{\log\varepsilon - \log(2\|\mu(S,\Delta^0)-\underline{\mu}(S)\|_\infty)}{\log(1-\theta)} \right\rceil + 1$$

and $\lceil x \rceil$ denotes the smallest integer greater or equal to x.

V. Proofs of Lemma 1 and Theorem 1

A. Additional Notation

Given a supervisor S and uncertainty value $\Delta\in\mathbf{\Delta}$, let $T_\Delta^S:\Re^n\rightarrow\Re^n$ be defined as

$$T_\Delta^S(\mu) \doteq \Pi(S,\Delta)\mu + X$$

Furthermore, let $T^S:\Re^n\rightarrow\Re^n$ be given by

$$T^S(\mu) = \min_{\Delta\in\mathbf{\Delta}} T_\Delta^S(\mu)$$

where the above minimum is taken entry by entry. Note that $T_\Delta^S(\cdot)$ is well-defined since, as mentioned in Section IV, the uncertainty in each row of $\Pi(S,\Delta)$ is independent of the uncertainties in all other rows. Finally, given $x\in\Re^n$, define the ∞-norm $\|x\| = \max_i |x_i|$. Given $x,y\in\Re^n$, it follows that $x\leq y$ if $x_i\leq y_i$ for all $i=1,\ 2,\ \ldots,n$. It also follows that $x<y$ if $x\leq y$ and $x_i<y_i$ for some i.

Before providing the proofs of Lemma 1 and Theorem 1, a number of relevant properties of the functions $T_\Delta^S(\cdot)$ and $T^S(\cdot)$ are established. These properties are not proven since they are a direct consequence of the fact that all the entries of $\Pi(S,\Delta)$ are positive and that each row sums to $1-\theta$.

Fact 1: Let S be a supervisor and $\Delta\in\mathbf{\Delta}$ be given, then T_Δ^S is a contraction. Moreover, given any vectors x, y,

$$\left\|T_\Delta^S(x)-T_\Delta^S(y)\right\| \leq (1-\theta)\|x-y\|.$$

Fact 2: Let S be a supervisor and let $\Delta,\Delta'\in\mathbf{\Delta}$ be given. If $T_\Delta^S(\mu(S,\Delta'))\leq\mu(S,\Delta')$ then $\mu(S,\Delta)\leq\mu(S,\Delta')$.

Fact 3: Let S be a supervisor. Then, $T_\Delta^S(\mu(S,\Delta'))<\mu(S,\Delta')$ implies that $\mu(S,\Delta)<\mu(S,\Delta')$.

Fact 4: Let S be a controllable supervisor. Then, the operator T^S is a contraction. Moreover, given any vectors x, y, $\left\|T^S(x)-T^S(y)\right\|\leq(1-\theta)\|x-y\|$.

Fact 5: Let S and S' be two supervisors, if $T^S(\underline{\mu}(S'))\geq\underline{\mu}(S')$ then $\underline{\mu}(S)\geq\underline{\mu}(S')$. In addition, if $T^S(\underline{\mu}(S'))>\underline{\mu}(S')$ then $\underline{\mu}(S)>\underline{\mu}(S')$.

Having these preliminary facts, one can now proceed with the proofs of Lemma 1 and Theorem 1.

B. Proof of Lemma 1

First, note that, by Fact 4, T^S is a contraction. Hence, there exists a $\underline{\mu}(S)$ such that

$$\underline{\mu}(S) = T^S(\underline{\mu}(S)) = \min_{\Delta\in\mathbf{\Delta}} T_\Delta^S(\underline{\mu}(S)).$$

Since $T_\Delta^S(\underline{\mu}(S))$ depends continuously on Δ and the minimization is done over the compact set $\mathbf{\Delta}$, then there exists a $\Delta^*\in\mathbf{\Delta}$ such that $\underline{\mu}(S) = T_{\Delta^*}^S(\underline{\mu}(S))$ and, therefore, $\underline{\mu}(S)=\mu(S,\Delta^*)$. Furthermore, T_Δ^S is monotone, i.e.,

$$x\leq y \Rightarrow T_\Delta^S(x)\leq T_\Delta^S(y).$$

This is a consequence of the fact that all the entries of the matrix $\Pi(S,\Delta)$ are positive. This implies monotonicity of the operator T^S; i.e., one can see that

$$x\leq y\Rightarrow T^S(x) = \min_{\Delta\in\mathbf{\Delta}} T_\Delta^S(x)\leq T^S(y) = \min_{\Delta\in\mathbf{\Delta}} T_\Delta^S(y).$$

Now, given any $\Delta\in\mathbf{\Delta}$ and the associated performance $\mu(S,\Delta)$, the definition of T^S implies that

$$T^S(\mu(S,\Delta)) \leq T_\Delta^S(\mu(S,\Delta)) = \mu(S,\Delta).$$

Therefore, it follows that

$$\begin{aligned}(T^S)^2(\mu(S,\Delta)) &\doteq T^S\left[T^S(\mu(S,\Delta)\right] \\ &\leq T^S(\mu(S,\Delta)) \leq \mu(S,\Delta)\end{aligned}$$

Repeating the above reasoning, for any k,

$$(T^S)^k(\mu(S,\Delta)) \leq \mu(S,\Delta)$$

Hence, the Contraction Mapping Theorem [15] implies

$$\underline{\mu}(S) = \lim_{k\to\infty}(T^S)^k(\mu(S,\Delta)) \leq \mu(S,\Delta)$$

Therefore, $\underline{\mu}(S) = \mu(S,\Delta^*)$ and the proof is complete.

C. Proof of Theorem 1

In Step 1 of Algorithm 1, we have

$$\begin{aligned}T^S_{\Delta^{k+1}}(\mu(S,\Delta^k)) = T^S(\mu(S,\Delta^k)) = \min_{\Delta\in\boldsymbol{\Delta}} T^S_{\Delta}(\mu(S,\Delta^k))\\ \leq T^S_{\Delta^k}(\mu(S,\Delta^k)) = \mu(S,\Delta^k)\end{aligned}$$

and, hence, by Fact 2, $\mu(S,\Delta^{k+1}) \leq \mu(S,\Delta^k)$. and one has a monotonic performance sequence. To see that one converges to the robust performance $\underline{\mu}(S)$, first note that the monotonicity of T^S_Δ implies that

$$\begin{aligned}\mu(S,\Delta^{k+1}) &= T^S_{\Delta^{k+1}}(\mu(S,\Delta^{k+1}))\\ &\leq T^S_{\Delta^{k+1}}(\mu(S,\Delta^k)) = T^S(\mu(S,\Delta^k)).\end{aligned}$$

Hence, given the definition of robust performance $\underline{\mu}(S)$, one can see that

$$\begin{aligned}\mu(S,\Delta^{k+1}) - \underline{\mu}(S) &\leq T^S(\mu(S,\Delta^k)) - \underline{\mu}(S)\\ &= T^S(\mu(S,\Delta^k)) - T^S(\underline{\mu}(S)).\end{aligned}$$

Since, both sides of the inequalities above are positive and the fact that T^S is a contraction with "contraction factor" $1-\theta$, we conclude that

$$\begin{aligned}\|\mu(S,\Delta^{k+1}) - \underline{\mu}(S)\| &\leq \|T^S(\mu(S,\Delta^k)) - T^S(\underline{\mu}(S))\|\\ &\leq (1-\theta)\|\mu(S,\Delta^k) - \underline{\mu}(S)\|.\end{aligned}$$

Hence, $\|\mu(S,\Delta^k) - \underline{\mu}(S)\| \leq (1-\theta)^k\|\mu(S,\Delta^0) - \underline{\mu}(S)\|$. We conclude the reasoning by noting that $0<\theta<1$.

Now, consider the case where the worst-case performance satisfies $|\underline{\mu}_i(S) - \underline{\mu}_j(S)| > \varepsilon$ for $i\neq j$ and $\varepsilon>0$. To prove that, in this case, one has convergence in a finite number of steps note that a solution of the optimization problem in Step 1 of Algorithm 1 can be determined in the following way: Let j^* be such that

$$\mu_{j^*}(S,\Delta^k) = \min_j \mu_j(S,\Delta^k).$$

Then, maximize over $\Delta\in\boldsymbol{\Delta}$ the value $\pi^S_{i,j^*}(\Delta)$. Now, with $\pi^S_{i,j^*}(\Delta)$ fixed, maximize the value of $\pi^S_{i,j^{**}}(\Delta)$, where j^{**} is such that

$$\mu_{j^{**}}(S,\Delta^k) = \min_{j, j\neq j^*} \mu_j(S,\Delta^k).$$

Now, repeat the reasoning above for the remaining values of j. Given this, one can see that one does not need to know the exact values of the entries of $\mu(S,\Delta^k)$ to determine Δ^{k+1}. One only needs to know the relative order of the entries.

Now, recall that

$$\|\mu(S,\Delta^{k+1}) - \underline{\mu}(S)\| \leq (1-\theta)^k\|\mu(S,\Delta^0) - \underline{\mu}(S)\|.$$

Hence, if

$$k \geq N = \left\lceil \frac{\log\varepsilon - \log(2\|\mu(S,\Delta^0) - \underline{\mu}(S)\|_\infty)}{\log(1-\theta)} \right\rceil + 1$$

one has $\|\mu(S,\Delta^k) - \underline{\mu}(S)\| \leq \varepsilon/2$ and, since $|\underline{\mu}_i(S) - \underline{\mu}_j(S)| > \varepsilon$ for $i\neq j$, the relative order of the entries of $\mu(S,\Delta^k)$ and $\underline{\mu}_j(S) = \mu(S,\Delta^*)$ are the same. Therefore, given the above remarks on the algorithm, if one has a unique minimizer, we conclude that $\Delta^{k+1} = \Delta^*$. If the minimizer is not unique, then there are several solutions to the optimization problem and one still obtains

$$\underline{\mu}(S) = \mu(S,\Delta^{k+1}).$$

VI. Optimal Robust Controller Design

In previous sections, the problem of determining the worst-case performance of a supervised plant was addressed. In this section, an algorithm for robust optimal supervisor design is provided. More precisely, we present an algorithm which converges to the supervisor which exhibits the best worst–case performance subject to the constraint that no state is a deadlock state.

Algorithm 2: Optimal robust supervisor design

Step 0. Let $k=0$ and let eps be the desired precision level. Let S^0 be a controllable supervisor and determine its worst-case performance $\underline{\mu}(S^0)$ using Algorithm 1.

Step 1. Let $j_1, j_2, \ldots, j_n$ be such that

$$\underline{\mu}_{j_1}(S^k) \geq \underline{\mu}_{j_2}(S^k) \geq \cdots \geq \underline{\mu}_{j_n}(S^k).$$

Step 2. Let S^{k+1} be the supervisor that disables all controllable events. For $i=1$ to n do

a. Let $l=1$.

b. If no events leading out of state i are enabled enable all events from state q_i to state q_{j_1} and go to the next step of the for loop.

c. Determine $\nu = \min_{\Delta\in\boldsymbol{\Delta}} \Pi_i(S^{k+1},\Delta)\underline{\mu}(S^k)$. where $\Pi_i(S^{k+1},\Delta)$ is the i-th row of the matrix $\Pi(S^{k+1},\Delta)$.

d. If $\nu < (1-\theta)\underline{\mu}_{j_l}(S^k)$ modify S^{k+1} by enabling all controllable events from state q_i to state q_{j_l}. Else go to the next step of the for loop.

e. Let $l=l+1$, If $l>n$ go to the next step of the for loop. Else go to c.

Step 3. Given S^{k+1}, determine $\underline{\mu}(S^{k+1})$ using Algorithm 1. If $\|\underline{\mu}(S^{k+1}) - \underline{\mu}(S^k)\|_\infty < eps$ stop. Else let k=k+1 and go to Step 1.

The result below establishes the convergence of the design algorithm presented above.

Theorem 2: The sequence of supervisors S^k obtained by Algorithm 2 satisfies $\underline{\mu}(S^{k+1}) > \underline{\mu}(S^k)$. Moreover, denoting *by $\underline{\mu}^*$ the optimal worst-case performance, the performance of the supervisors obtained satisfies*

$$\|\underline{\mu}(S^k) - \underline{\mu}^*\|_\infty \leq (1-\theta)^k \|\underline{\mu}(S^0) - \underline{\mu}^*\|_\infty$$

Proof: A sketch of the proof of the theorem above is now provided. Let $T : \Re^n \to \Re^n$ be defined as

$$T(\mu) \doteq \max_{S \in \mathcal{S}} T^S(\mu).$$

Some relevant properties of $T(\cdot)$ are established in the following two lemmas. The proofs are omitted since they are similar to the proofs of the facts in Section V.

Fact 6: The transformation T is a contraction. Moreover, given any vectors x, y, $\|T(x) - T(y)\| \leq (1-\theta)\|x - y\|$.

Fact 7: There exists a S^* such that $\underline{\mu}^* = \underline{\mu}(S^*) = T(\underline{\mu}(S^*))$. Furthermore, for all $S \in \mathcal{S}$, $\underline{\mu}^* \geq \underline{\mu}(S)$.

A sketch of the proof of Theorem 2 is now provided. First, note that S^{k+1} in Step 2 of Algorithm 2 is such that

$$T^{S^{k+1}}(\underline{\mu}(S^k)) = \max_{S \in \mathcal{S}} T^S(\underline{\mu}(S^k)).$$

To see this, first one should note that one can maximize each of the entries of $T^S(\underline{\mu}(S^k))$ independently since events are enabled or disabled independently at each state. Therefore, we now concentrate on the i-th element of $T^S(\underline{\mu}(S^k))$; i.e., it turns out that, at each step, S^{k+1} maximizes

$$\min_{\Delta \in \boldsymbol{\Delta}} \Pi_i(S, \Delta)\underline{\mu}(S^k)$$

where $\Pi_i(S, \Delta)$ is the i-th row of $\Pi(S, \Delta)$. This can be proven using the following results.

Fact 8: Each of the iterations of the subroutine in Step 2 strictly increases the value of

$$\min_{\Delta \in \boldsymbol{\Delta}} \Pi_i(S^{k+1}, \Delta)\underline{\mu}(S^k).$$

Fact 9: If a supervisor S that achieves the maximum of

$$\min_{\Delta \in \boldsymbol{\Delta}} \Pi_i(S, \Delta)\underline{\mu}(S^k)$$

and enables events leading from state q_i to state q_l, then it also enables all events leading from state q_i to states q_m that satisfy $\underline{\mu}_m(S^k) > \underline{\mu}_l(S^k)$.

One is now ready to prove Theorem 2. Given the results above, the supervisor determined in Step 2 satisfies

$$\begin{aligned} T^{S^{k+1}}(\underline{\mu}(S^k)) &= \max_{S \in \mathcal{S}} T^S(\underline{\mu}(S^k)) = T(\underline{\mu}(S^k)) \\ &> T^{S^k}(\underline{\mu}(S^k)) = \underline{\mu}(S^k) \end{aligned}$$

Hence, by Lemma 5, $\underline{\mu}(S^{k+1}) > \underline{\mu}(S^k)$. Now, note that

$$\begin{aligned} \underline{\mu}^* - \underline{\mu}(S^{k+1}) &= \underline{\mu}^* - T^{S^{k+1}}(\underline{\mu}(S^{k+1})) \\ &\leq \underline{\mu}^* - T^{S^{k+1}}(\underline{\mu}(S^k)) \\ &= T(\underline{\mu}^*) - T(\underline{\mu}(S^k)). \end{aligned}$$

Since both sides of the inequalities above are positive and T is a contraction with "contraction factor" $1-\theta$, one obtains

$$\begin{aligned} \|\underline{\mu}^* - \underline{\mu}(S^{k+1})\| &\leq \|T(\underline{\mu}^*) - T(\underline{\mu}(S^k))\| \\ &\leq (1-\theta)\|\underline{\mu}^* - \underline{\mu}(S^k)\|. \end{aligned}$$

Hence,

$$\|\underline{\mu}^* - \underline{\mu}(S^k)\| \leq (1-\theta)^k \|\underline{\mu}^* - \underline{\mu}(S^0)\|.$$

The proof is concluded by noting that $0 < \theta < 1$.

■

VII. Concluding Remarks

In this paper, a new way of addressing the performance of supervised automata is presented. The approach taken is based on recent results on quantitative measures of regular languages and addresses the case where the weights of the language measure are related to the relative frequency of events. Algorithms for worst–case performance assessment and optimal robust supervisor design are presented. It is shown that the algorithms have exponential convergence and that, in many cases, the worst–case performance assessment algorithm converges in a finite number of steps.

Effort is now being put on the case where one cannot observe directly the current state of the automaton. Hence, the problem of optimal observer design and its use in the design of optimal supervisors is now being studied.

References

[1] X. Wang and A. Ray, "A language measure for performance evaluation of discrete event supervisory control systems," *Applied Mathematical Modelling*, vol. 28, no. 9, pp. 817–833, 2004.

[2] A. Ray and S. Phoha, "Signed real measure of regular languages for discrete-event automata," *International Journal of Control*, vol. 76, no. 18, pp. 1800–1808, 2003.

[3] J. Fu, A. Ray, and C. M. Lagoa, "Unconstrained optimal control of regular languages," *Automatica*, vol. 40, pp. 639–646, April 2004.

[4] J. Fu, A. Ray, and C. M. Lagoa, "Optimal control of regular languages with event disabling cost," in *Proceedings of the 2003 American Control Conference*, (Denver, CO), pp. 1691–1695, 2003.

[5] J. Fu, C. M. Lagoa, and A. Ray, "Robust optimal control of regular languages," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, (Maui, Hawaii), pp. 3209–3214, December 2003.

[6] S.-J. Park and J.-T. Lim, "Robust and nonblocking supervisory control of nondeterministic discrete event systems using trajectory models," *IEEE Transactions on Automatic Control*, vol. 47, pp. 655–658, April 2002.

[7] J. E. R. Cury and B. H. Krogh, "Robustness of supervisors for discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 44, pp. 376–379, February 1999.

[8] S. Takai, "Design of robust supervisors for prefix-closed language specifications," in *Proceedings of the 38th IEEE Conference on Decision and Control*, (Phoenix, AZ), pp. 1725–1730, 1999.

[9] F. Lin, "Robust and adaptive supervisory control of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 38, pp. 1848–1852, December 1993.

[10] K. M. Passino and P. J. Antsaklis, "On the optimal control of discrete event systems," in *Proceedings of the 28th IEEE Conference on Decision and Control*, (Tampa, FL), pp. 2713–2718, 1989.

[11] R. Kumar and V. K. Garg, "Optimal supervisory control of discrete event dynamical systems," *SIAM Journal on Control and Optimization*, vol. 33, no. 2, pp. 419–439, 1995.

[12] Y. Brave and M. Heymann, "On optimal attraction of discrete-event processes," *Information Science*, vol. 67, no. 3, pp. 245–276, 1993.

[13] R. Sengupta and S. Lafortune, "An optimal control theory for discrete event systems," *SIAM Journal on Control and Optimization*, vol. 36, pp. 488–541, March 1998.

[14] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[15] A. W. Naylor and G. W. Sell, *Linear Operator Theory in Engineering and Science*. New York: Springer-Verlag, 1982.