have also given the formula for one particular solution, and parameterized the set of all causal solutions with a free parameter $Y$. The results can be applied to design a controller in exact model matching problem, or to design a compensator in the disturbance decoupling problem.

ACKNOWLEDGMENT

The author wishes to thank Prof. Jacob van der Woude from Technische Universität Delft for pointing out the connection between the causal factorization problem and the disturbance decoupling problem, and for providing the reference [13].

REFERENCES


Robust and Adaptive Supervisory Control of Discrete Event Systems

Feng Lin

Abstract—To control a system with model uncertainty, two approaches are available: 1) robust control that controls the system without actually resolving uncertainties; 2) adaptive control that invokes an identification scheme to resolve uncertainties and updates the control algorithm accordingly. In this note we discuss both robust and adaptive supervisory control in discrete event systems. We assume that the system $G$ to be controlled is not known exactly. We only know that it belongs to a set or its "lower" and "upper" bounds. The task of robust supervision is to synthesize a supervisory controller that realizes a given desired behavior for all possible $G$. We derive a necessary and sufficient condition for the existence of such a robust supervisor. Based on this condition, a robust supervisory control and observation problem of synthesizing a robust supervisor whose behavior is both legal and acceptable is solved. We also discuss adaptive supervision. As the system progresses, the information on occurrences of events may help us to resolve or reduce uncertainties. This information can be used to improve the performance of the closed loop systems by introducing adaptive supervision.

I. INTRODUCTION

If the dynamics of a plant (system to be controlled) is partially known, then control of the system becomes more complex. In general, two approaches are available to control such an uncertain system: The first is robust control that controls the system without actually resolving uncertainties. The second is adaptive control that invokes an identification procedure to resolve uncertainties and updates the control algorithm accordingly. Both approaches have been studied extensively for continuous variable systems. Their importance in control has been recognized by control scientists for decades. Obviously, the issue of dealing with uncertainties exists not only in control of continuous variable systems but also in control of discrete event systems. In this note, we study robust and adaptive control of discrete event systems in the framework of supervisory control.

For continuous variable systems, a system to be controlled is modeled either by a transfer function in the frequency domain or by a differential (or difference) equation in the state space representation. Sometimes, however, due to uncertainties in parameters or inaccuracy in modeling, a single nominal model may not be adequate to model the system. Instead, we assume that, although we do not know the exact model of the system, it belongs to a set of models. The objective of robust control is to design a controller to achieve a given desired behavior for any model in the set. Here the desired behavior is usually described by the stability of the closed loop system or optimality with respect to a given performance measure such as a quadratic performance index.

In discrete event systems, although models and control mechanisms are very different from continuous variable systems, many principles of control remain the same. In supervisory control, a discrete event system to be controlled is modeled by

Manuscript received August 8, 1991, revised May 22, 1992 and June 25, 1992. This work was supported by the National Science Foundation under Grant ECS-9088947. The author is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202.

IEEE Log Number 9212676.
an automaton. The behavior of the system is usually described by the language generated by the automaton. Each string in the language represents a possible trajectory of the system. A controller (or supervisor) is then designed, whose task is to enable or disable some controllable events based on a record of occurrences of some observable events so that the closed loop system achieves a given desired behavior. In supervisory control, the desired behavior is described by another language. Each string in that language represents a desired trajectory.

If there are uncertainties in the system, then a single automaton model may not be adequate to model the system. Hence, we assume that the model of the system belongs to a set of models. The objective of robust supervision is to design a supervisor to achieve the desired behavior for any model belonging to the set.

In this note, we will derive a necessary and sufficient condition for the existence of such a robust supervisor. This condition is expressed in terms of controllability and observability with respect to the union of the languages generated by the automata in the set of models. Comparing our result with that of conventional supervision, we find that the only difference between the two is that now controllability and observability is with respect to the union of the languages generated by the set of models instead of the language generated by one model. One interpretation of this result is rather interesting: it says that we can also view uncertainties in modeling as possible dynamics of the system. We will show that these two points of view are essentially equivalent as far as supervisor synthesis is concerned.

With this result in mind, we can solve a Robust Supervisory Control and Observation Problem of synthesizing a robust supervisor whose behavior is both legal and acceptable in a way similar to that of Supervisory Control and Observation Problem presented in [7].

Under the assumption of complete observation, given a legal behavior, we can synthesize an optimal supervisor that generates the largest possible language. This supervisor is least restrictive in the sense that it will disable as few events as possible. By introducing a lifetime to each occurrence of events, we can show (see [6]) that, under a certain fairness condition, the closed loop system with least restrictive supervisor runs fastest in the sense that each occurrence of events will occur earliest. The reader is referred to [6] for this interesting result.

This robust supervisory control scheme is very simple indeed. As we will show, however, if we can resolve or reduce uncertainties while the system is progressing, then we can improve the performance of the system by introducing adaptive supervisory control.

The basic idea of adaptive supervisory control is this: After some events have occurred in the system, we will have more information about the system. This will allow us to reduce uncertainties by eliminating some of the possible models. When a model is excluded from the set of possible models, we can reconfigure the supervisor to take advantage of that additional information.

Several problems related to robust and adaptive supervision have been discussed by other researchers. For example, supervisory control using limited lookahead policies [22, 3] takes into account changes in the system and can be used in combination with our adaptive control scheme. Also, discrete event system identification has been studied by researchers in automation theory (e.g., [16, 20]) as well as by researchers in discrete event systems (e.g., [11]). However, to the best of our knowledge, the robust and adaptive supervision presented in this note have never been formally studied in the context of supervisory control.

II. EXISTENCE OF ROBUST SUPERVISING CONTROL

Let $\Sigma$ be a nonempty finite alphabet, which will be interpreted as events, and denote by $\Sigma^*$ the set of all finite traces (strings) of elements of $\Sigma$, including the empty trace $\varepsilon$ ("$\varepsilon"$ is called the Kleene closure) [5]. A subset $L \subseteq \Sigma^*$ is a language over $\Sigma$. If $s, s' \in \Sigma^*$ with $s'r = s$, then $s'$ is a prefix of $s$. The closure $L$ of $L$ is the language consisting of all the prefixes of traces in $L$. $L$ is closed if $L = L$.

A discrete event system to be controlled is modeled by a generator [9], which is a deterministic automaton $G = (Q, \Sigma, \delta, q_0, \mathcal{Q}_a)$, where $Q$ is the state space, $\Sigma$ is the set of events, $\delta: \Sigma^* \times Q \rightarrow Q$ is the transition function, $q_0$ is the initial state, and $\mathcal{Q}_a \subseteq Q$ is the set of marked states. The closed language generated by $G$ is $L(G) = \{ s \in \Sigma^*: \delta(s, q_0) \in \mathcal{Q}_a \}$.

The language marked by $G$ is $L_a(G) = \{ s \in L(G): \delta(s, q_0) \in \mathcal{Q}_a \}$.

It is assumed that some events are controllable, in the sense that their occurrence can be prevented; and some events are observable, in the sense that their occurrence can be observed. Thus, $\Sigma$ is partitioned as $\Sigma = \Sigma_c \cup \Sigma_u = \Sigma_a \cup \Sigma_{ua}$, where $\Sigma_c$ is the set of controllable events, $\Sigma_u$ is the set of uncontrollable events, $\Sigma_a$ is the set of observable events, and $\Sigma_{ua}$ is the set of unobservable events.

The basic problem in supervisory control is to design a supervisor whose task is to enable and disable the controllable events based on a record of occurrences of the observable events such that the resulting closed loop system obeys some prespecified operating rules. Formally, a supervisor is a pair $S = (R, \phi)$, where $R = (X, \Sigma, x_0, X_a)$ is a generator called recognizer with $x_0: \Sigma \times X \rightarrow X$ satisfying $(\forall \sigma \in \Sigma_a)(\xi(\sigma, x) = x)$, and $\phi: \Sigma \times X \rightarrow \{0, 1\}$ is a feedback map satisfying $\phi(\sigma, x) = 1$ if $\sigma \in \Sigma_a$, $x \in X$; $\phi(\sigma, x) = 0$ if $\sigma \in \Sigma_c$, $x \in X$.

$R$ is considered to be driven externally by the stream of event symbols generated by $G$, while in turn, with $R$ in state $x$, the transitions $\sigma$ of $G$ are subject to the control $\phi(\sigma, x)$. If $\phi(\sigma, x) = 0$, then $\sigma$ is "disabled" (prohibited from occurring); while if $\phi(\sigma, x) = 1$, then $\sigma$ is "enabled" (permitted but not forced to occur). In this way, there results a closed loop feedback structure $S/G$, called the supervisory discrete event system. $S/G$ is itself an automaton $S/G = (\Sigma \times X \times Q, (\xi \times \delta)^{\psi}, (x_0, q_0), X_a \times \mathcal{Q}_a)$, where $(\xi \times \delta)^{\psi}: \Sigma \times X \times Q \rightarrow X \times Q$ is defined as $\{ (\xi(\sigma, x), \delta(\sigma, q)) \text{ if } \phi(\sigma, x) = 1 \\ \text{undefined } \text{if } \phi(\sigma, x) = 0 \}$.
The behavior of the supervised discrete event system is described by the language $LS(G)$ generated by $S/G$ and the language $L_n(S/G)$ marked by $S/G$. $S$ is said to be proper if $L_n(S/G) = L(S/G)$. In some applications, $G$ is not known exactly as shown in the following example.

Example 1: A robot is working in a manufacturing plant consisting of $n$ similar but not identical work stations. The robot moves from one station to another to perform similar jobs as required. Since all stations are similar, it is simple and cost effective to design only one supervisor that can control the robot in all stations. Formally, let $G$ be a set of all possible $G_i$'s.

As shown in the above example, we may only know that $G$ belongs to a set of possible models:

$$G \in \{G_a : a \in A\}$$

where $A$ is a finite set. If we denote

$$L_n(F) = \bigcap_{a \in A} L_n(G_a), \quad L(F) = \bigcap_{a \in A} L(G_a),
\frac{L_n}{L_n}(H) = \bigcup_{a \in A} L_n(G_a), \quad L(H) = \bigcup_{a \in A} L(G_a),$$

then $L_n(G)$ satisfies

$$L_n(F) \subseteq L_n(G) \subseteq L_n(H).$$

$F$ and $H$ can be viewed as the lower and upper bounds of $G$, respectively. The desired behavior of the closed loop system is described by a language $K \subseteq L_n(F)$.

We will now study the following problem: Under what condition does there exist a proper supervisor $S$ such that $L_n(S/G) = K$ for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$?

To answer this question, we need the notions of controllability and observability which can be found in [8], [7]. As shown in the following theorem, they play an important role in ensuring the existence of a robust supervisor.

Theorem 1: Let $K \subseteq L_n(F)$ be a nonempty language. There exists a proper supervisor $S$ such that $L_n(S/G) = K$ for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$ if and only if $K$ is controllable and observable with respect to $L_n(H)$.

Proof:

(Only if) Suppose that there exists a proper supervisor $S$ such that $L_n(S/G) = K$ for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$. Let us index the language marked by all those $G$ satisfying $(\ast)$ or $(\ast \ast)$ by $L_n(a)$, $a \in A$. Clearly,

$$\bigcup_{a \in A} L_n(a) = L_n(H).$$

Since the closure operation is closed under union ([5]),

$$L_n(G) = \bigcup_{a \in A} L_n(a) - \bigcup_{a \in A} \frac{L_n}{L_n}(a) = L(H).$$

The existence of $S$ implies the controllability of $K$ with respect to all $L_n(a)$.

$$\tilde{R} = L_n(G) \cap \frac{L_n}{L_n}(a) \subseteq \tilde{R}.$$

Therefore,

$$L_n(a) = \bigcup_{a \in A} L_n(G) \cap \tilde{L}_n \subseteq \tilde{R}.$$

That is, $K$ is controllable with respect to $L_n(H)$.

The existence of $S$ also implies the observability of $K$ with respect to all $L_n(a)$ (denote the projection from $2^n$ to $\Sigma^+$):

$$\forall s, s' \in \Sigma^+ \left( Ps = Ps' \implies \left( (\forall s \in \Sigma) \right) \right), \quad \left( \forall a \in A \right) \left( \forall s \in \tilde{K}, s' \in \frac{L_n}{L_n}(a) \implies s' \sigma s \in \tilde{K} \right).$$

Therefore, for all $s, s' \in \Sigma^+$, $P_s = Ps'$ implies

$$\forall s, s' \in \Sigma^+ \left( Ps = Ps' \implies \left( (\forall s \in \Sigma) \right) \right), \quad \left( \forall a \in A \right) \left( \forall s \in \tilde{K}, s' \in \frac{L_n}{L_n}(a) \implies s' \sigma s \in \tilde{K} \right).$$

That is, $K$ is observable with respect to $L_n(H)$.

(If) Suppose $K$ is controllable and observable with respect to $L_n(H)$. Then we can construct a supervisor $S = (R, \psi)$ as follows:

$$R = (\Sigma, X, \xi, x_0, X_m),$$

where

$$X = \Sigma^*; \quad \xi(\sigma, x) = \begin{cases} x & \text{if } \sigma \in \Sigma_0; \\ x_0 & \text{if } \sigma \in \Sigma_1; \end{cases} \quad x_0 = e; \quad X_m = \{ x \in X : (3 \in K) Ps = x \}.$$

From the definition, $\xi(s, x_0) = Ps$. For all $x \in X$, define

$$\Sigma_0^\Sigma = \{ s \in \Sigma : (3 \in K) Ps = x \land s \sigma s \in L(H) \land \tilde{R} \};$$

$$\Sigma_1^\Sigma = \{ s \in \Sigma : (3 \in K) Ps = x \land s \sigma s \in L(H) \land \tilde{R} \}.$$}

Then controllability implies $\Sigma_0^\Sigma \subseteq \Sigma_0$, and observability implies $\Sigma_1^\Sigma \subseteq \Sigma_1$. Therefore, we can take $\psi$ to be any function satisfying

$$\psi(s, x) = 1 \text{ if } \sigma \in \Sigma_0^\Sigma;$$

$$\psi(s, x) = 0 \text{ if } \sigma \in \Sigma_1^\Sigma.$$
Since $K \subseteq \bar{K}$ and $K \subseteq L_\omega(F) \subseteq L_\omega(G)$, we have, by taking $s = s'$,
\[ s' \in K \Rightarrow s' \in \bar{K} \land s' \in L_\omega(G) \land (\exists s \in K)Ps = P'y. \]
On the other hand, by observability,
\[ s' \in \bar{K} \land s' \in L_\omega(G) \land (\exists s \in K)Ps = P'y \]
\[ \Rightarrow s' \in \bar{K} \land \exists s \in L_\omega(H) \land (\exists s \in K)Ps = P'y \]
\[ \Rightarrow s' \in K. \]
Hence, $L_\omega(S/G) = K$. This completes the proof. □

Recall [7] that, when $G$ is exactly known, the necessary and sufficient condition for the existence of a proper supervisor is that $K$ is controllable and observable with respect to $L_\omega(G)$. Comparing this condition with that in Theorem 1, we see that the difference is that controllability and observability is now with respect to $L_\omega(H)$ instead of $L_\omega(G)$. We can think that $H$ contains all possible dynamics of $G$ because $L_\omega(G) \subseteq L_\omega(H)$. Then the result of Theorem 1 says that uncertainties in modeling can be viewed as possible dynamics of the system.

In the next section, we will study a synthesis problem, where we are only interested in language $L(S/G)$. To this end, let us mark all states in the uncontrolled system, that is, $L(H) = L_\omega(H)$. Then we can state the following corollary.

Corollary 1: Let $K \subseteq L(F)$ be a nonempty language. There exists a proper supervisor $S$ such that $L(S/G) = K$ for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$ if and only if only if $K$ is closed, controllable and observable with respect to $L(H)$.

III. ROBUST SUPERVISION CONTROL AND OBSERVATION

We already derived a necessary and sufficient condition under which a robust supervisor exists. Using the results in Section II, we can synthesize a robust supervisor to satisfy certain specifications. Such specifications are usually given in terms of two constraints. One constraint is on the legality of the closed loop system: Each trajectory generated by the closed loop system must be legal (e.g., it must satisfy certain operating rules). The second constraint is on the flexibility of the closed loop systems: The system must be flexible enough to allow certain basic operations. Formally, we define two languages $A$ and $E$ such that
\[ A \subseteq E \subseteq L(F), \]
where $A$ is the minimal acceptable language and $E$ is the maximal legal language. Both are closed and nonempty.

Our goal is to synthesize a robust supervisor so that the closed loop system is both legal (i.e., contained in $E$) and acceptable (i.e., containing $A$). Therefore, we propose the following.

Robust Supervision Control and Observation Problem (RSCOP)

Synthesize a proper supervisor $S$ such that, for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$,
\[ A \subseteq L(S/G) \subseteq E. \]

To discuss the solvability of RSCOP, we define two classes of languages:
\[ C(E) = \{ K \subseteq E : K \text{ is closed and controllable wrt } L(H) \}; \]
\[ O(A) = \{ K \supseteq A : K \text{ is closed and observable wrt } L(H) \}. \]
As shown in [9], [7], $C(E)$ is closed under arbitrary union and $O(A)$ is closed under arbitrary intersection. Therefore sup $C(E)$ and inf $O(A)$ exist. Note that without assuming closeness of the languages, inf $O(A)$ may not exist. That is the reason that we assume $A$ and $E$ are closed in this section.

Theorem 2: RSCOP is solvable if and only if
\[ \inf O(A) \subseteq \sup C(E). \]

Proof:

(Only if) Suppose that RSCOP is solvable. Then there exists a proper supervisor $S$ such that, for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$,
\[ A \subseteq L(S/G) \subseteq E. \]

By Corollary 1, $L(S/G)$ is closed, controllable and observable with respect to $L(H)$. Therefore,
\[ \inf O(A) \subseteq L(S/G) \subseteq \sup C(E). \]
(If) Suppose
\[ \inf O(A) \subseteq \sup C(E). \]

Then define
\[ K = \inf O(A) \subseteq L(H). \]
Clearly $K$ is closed. Furthermore,
\[ \inf O(A) \subseteq K \subseteq \sup C(E) \subseteq L(H). \]

Since sup $C(E)$ is controllable with respect to $L(H)$, $\sup C(E) \subseteq L(H)$.

Therefore,
\[ \inf O(A) \subseteq K \subseteq \sup C(E) \subseteq L(H). \]

By Lemmas 3.2 and 3.3 of [7], $K$ is controllable and observable with respect to $L(H)$. Hence, by Corollary 1, there exists a proper supervisor $S$ such that $L(S/G) = K$ for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$. In other words,
\[ A \subseteq L(S/G) \subseteq E. \]

IV. ADAPTIVE SUPERVISION

In this section, we consider a different supervisory control problem: Given a (maximal) legal language $E$, not necessarily closed, we would like to synthesize a supervisor $S$ such that $L(S/G) \subseteq E$ and $L(S/G)$ is the largest. As shown in [6], such a supervisor is optimal in the sense that the closed loop system runs fastest. We assume that all events are observable in the rest of the note. Therefore, observability is always satisfied and we have the following corollary.

Corollary 2: Assume that all events are observable. Let $K \subseteq L_\omega(F)$ be a nonempty language. There exists a proper supervisor $S$ such that $L_\omega(S/G) = K$ for all possible $G$ satisfying $(\ast)$ or $(\ast \ast)$ if and only if $K$ is controllable with respect to $L_\omega(H)$.

An optimal supervisor should realize as large a language as possible. To this end, we denote the supremal controllable sublanguage of $E$ with respect to $L_\omega(H)$ by $E_{\omega}(H)$. A formula to calculate $E_{\omega}(H)$ can be found in [1] when $E$ is closed.

To discuss adaptive supervision, let us assume that $G$ satisfies $(\ast)$:
\[ G \subseteq G' = \{ G_a : a \in A \}. \]

Denote $\cup_{a \in A} L_\omega(G_a)$ by $L_\omega(G)$. Clearly, the best that a robust supervisor can do is to realize $E_{\omega}(G)$. If however, we can reduce uncertainties by eliminating some models from $G$, that is, if we know $G \subseteq G' \subseteq G$, then we can synthesize a supervisor $S'$ that realizes $E_{\omega}(G)$. We will show, in the following theorem, that $E_{\omega}(G) \subseteq E_{\omega}(G')$.

Theorem 3: If $G' \subseteq G$, that is $L_\omega(G') \subseteq L_\omega(G)$, then $E_{\omega}(G') \subseteq E_{\omega}(G)$.\]

Proof: Since $E_{\omega}(G') \subseteq E$, we only need to prove that $E_{\omega}(G')$ is controllable with respect to $L_\omega(G')$. But this is obvious
because
\[ E_{x_n} \cap L_n(G') \subseteq E_{x_n} \cap L_n(G) \subseteq E. \]
\[ \square \]
This theorem shows that, if we can resolve or reduce uncertainties, then we can improve the performance of the closed loop system by using adaptive supervision.

Let us thus consider procedures to resolve or reduce uncertainties. Let us call such a procedure identification. The goal of identification is to eliminate models from G. The information that we have is occurrences of events up to a moment in time. Clearly, after a string of events \( t \) has occurred in G, or in \( S/G \) if a supervisor has been employed, the set of all possible models is reduced to
\[ G(t) = \{ G_a : a \in E : t \in L(G_a) \}. \]

If \( G(t) \) is strictly contained in \( G \), then we can reconfigure the supervisor to take the advantage of the additional information.

Therefore, we introduce the following adaptive control scheme. Let us denote the supervisor after string \( t \) has occurred by \( S(t) \). Initially, \( G(0) = G \), and the supervisor realizes \( E_{x_0}^{1}(G) \). We will use this supervisor until we have reduced uncertainties by eliminating some models. At that time, we will reconfigure the supervisor. This procedure will continue until uncertainties are resolved.

To formalize this scheme mathematically, let us introduce the following notations:
1) The set of all supervisors that realize a given language \( K \) is denoted by
\[ S(K) = \{ S : L_{x_n}(S/G) = K \}. \]
An element of \( S(K) \) is denoted by \( S(K) \).

2) The post-language of \( K \) after a string \( s \) is denoted by \( K/s = \{ |u \in \Sigma^* : su \in K \} \).

With these definitions, an adaptive supervisor \( S(t) \) is given by
\[ S(t) = S \left( E_{x_0}^{1}(G) \right); \]
for the empty trace. For a trace \( s \) generated by the closed loop system \( s \in L(S(t)/G) \),
\[ S(s) = \begin{cases} \{ s \} & \text{if } G(s) = G(s); \\ \left( E_{x_0}^{1}(G) \right)/s & \text{otherwise}. \end{cases} \]

This adaptive supervisor is indeed better than a robust supervisor. It has the property that the closed loop system generates no illegal strings and is most flexible as shown in the following theorem.

**Theorem 4:** For the supervisor \( S(t) \) defined above, we have
\[ L(S(t)/G) \subseteq E. \]
For any other supervisor \( S'(t) \) of \( G \), we have
\[ L(S'(t)/G) \subseteq E \Rightarrow L(S'(t)/G) \subseteq L(S(t)/G). \]

**Proof:** By the construction of \( S(t) \), for all \( s \in L(S(t)/G) \), we have
\[ L(S(t)/G)/s = \left( E_{x_0}^{1}(G) \right)/s \subseteq E/s. \]
Therefore,
\[ L(S(t)/G) \subseteq E. \]

For any other supervisor \( S'(t) \) of \( G \),
\[ L(S'(t)/G) \subseteq E \]
implies that, for all \( s \in L(S'(t)/G) \),
\[ L(S'(t)/G)/s \subseteq E/s. \]

Since \( L(S'(t)/G)/s \) is controllable with respect to \( L_{x_0}(G)/s \),
\[ L(S'(t)/G)/s \subseteq L(S(t)/G)/s. \]
Therefore,
\[ L(S(t)/G) \subseteq L(S(t)/G). \]
\[ \square \]

This adaptive control scheme can be used in combination with the limited lookahead policies developed in [2], [3]. Using limited lookahead policies, the computational complexity of adaptive supervision is essentially the same as that of conventional supervision. This is because limited lookahead policies can easily handle “time-varying” systems. In general, the complexity of limited lookahead policies depends on the window size. On the other hand, as shown in [3], we can efficiently calculate limited lookahead policies by means of dynamic programming.

**V. ACKNOWLEDGMENT**

The author would like to thank the reviewers for their helpful comments and suggestions.

**REFERENCES**


**Robustness of Discrete-time Adaptive Controllers: An Input–Output Approach**

Aniruddha Datta

**Abstract**—In this note, we present a general input–output approach that can be used for the design and analysis of robust discrete-time adaptive control schemes. In particular, we consider a model reference adaptive controller. The design approach involves the use of the Cor...