Randomized Algorithms for Synthesis of Switching Rules for Multimodal Systems

Hideaki Ishii, *Meber, IEEE*, Tamer Başar, *Fellow, IEEE*, and Roberto Tempo, *Fellow, IEEE*

***Abstract*—In this paper, we consider the design of globally asymptotically stabilizing state-dependent switching rules for multimodal systems, first restricting attention to linear time-invariant (LTI) systems with only two states for the switch, and then generalizing the results to multimodal LTI systems and to nonlinear systems. In all cases, the systems considered do not allow the construction of a single quadratic Lyapunov function and, hence, fall in the class of problems that require multiple Lyapunov functions and thus are nonconvex. To address the challenge of *nonconvexity*, we introduce probabilistic algorithms, and prove their probability-one convergence under a new notion of convergence. Then, to reduce complexity, we develop modified versions of the algorithm. We also present a class of more general nonconvex problems to which this approach can be applied. The results are illustrated using two- and three-dimensional systems with multiple switch states.**

***Index Terms*—Multiple Lyapunov function, nonconvexity, randomized algorithms, switched systems, switching rule design.**

I. Introduction

MULTIMODAL systems, or switched systems, represent a class of hybrid systems, and stability issues of such systems have been a subject of much recent research. Consider the linear case with two subsystems given by

$$\dot{x}(t) = A_{\sigma(t)}x(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, and $\sigma(t) \in \{1, 2\}$ is the state of the switch. The matrices A_1 and A_2 are assumed to be non-Hurwitz, that is, each A_i, $i = 1, 2$, has at least one eigenvalue with a nonnegative real part. The problem of this paper is to construct a state-dependent switching rule such that global asymptotic stability is achieved. We assume that there is no convex combination of matrices A_1 and A_2 that is Hurwitz because otherwise it is well known that there is a single quadratic Lyapunov function, which ensures quadratic stability.

In designing a switching scheme, it is attractive to employ a single Lyapunov function of a quadratic type because of its simplicity, but it is unnecessarily restricted. In the hybrid systems literature, methods that make use of multiple Lyapunov functions, or piecewise Lyapunov functions, have been studied extensively (see, e.g., [5], [7], [12], [15], [25], [26], and the references therein). Most of them, however, are for analysis purposes and, hence, the issue of design remains as a challenging problem.

In [17] and [24], one of the few such methods for constructing switching rules was developed. The method employs a piecewise quadratic Lyapunov function and can be described as follows. Let the unit sphere in $\mathbb{R}^n$ with center 0 be $\partial\mathcal{B}_0 := \{x \in \mathbb{R}^n : \|x\| = 1\}$. Given $\gamma > 0$ and positive–definite matrices $P_1, P_2 \in \mathbb{R}^{n\times n}$, suppose that for every $x \in \partial\mathcal{B}_0$

$$\begin{aligned} &x'(P_1A_1 + A_1'P_1)x \le -\gamma, \ \text{ if } x'(P_1 - P_2)x \ge 0, \ \text{ and} \\ &x'(P_2A_2 + A_2'P_2)x \le -\gamma, \ \text{ if } x'(P_1 - P_2)x \le 0. \end{aligned} \tag{2}$$

Then, we can construct a switching rule

$$\sigma(t) = \arg\max_{i=1,2} x(t)'P_ix(t). \tag{3}$$

Observe that the state–space $\mathbb{R}^n$ is partitioned into two sets $\mathcal{X}_1 := \{x : x'(P_1 - P_2)x \le 0\}$ and $\mathcal{X}_2 := \{x : x'(P_1 - P_2)x \ge 0\}$[1], and the switching signal $\sigma(t)$ is determined by the set in which the state $x(t)$ is (and, hence the term state-dependent switching). It can be easily checked that, if there is no chattering in the switching[2], the piecewise quadratic function $V(x) := \max_{i=1,2} x'P_ix$ decreases along trajectories of (1) and, thus, global asymptotic stabilization is achieved.

The switched system problem above is nonconvex and, hence, is computationally difficult to solve; for complexity issues arising in control, see [3]. A conventional method for constructing the matrices P_1 and P_2 is to use a bilinear matrix inequality (BMI) condition which is equivalent to (2), through the use of the so-called S-procedure [7], [24]. Although this method may simplify the problem to some extent, algorithms generally have high complexity. Another drawback in this context is that for switched systems with three or more subsystems, the condition of this type becomes only sufficient and, thus, conservative.

The objective of this paper is to develop a probabilistic, gradient-based algorithm which directly deals with the condition in (2). The proposed algorithm iteratively generates a pair (P_1, P_2)

Manuscript received September 15, 2003; revised March 19, 2004, July 22, 2004, and November 19, 2004. Recommended by Associate Editor A. Giua. This work was supported in part by the National Science Foundation under Grant CCR 00-85917 ITR.

H. Ishii is with the Department of Information Physics and Computing, The University of Tokyo, Tokyo 113-8656, Japan (e-mail: hideaki_ishii@ipc.i.u-tokyo.ac.jp).

T. Başar is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: tbasar@control.csl.uiuc.edu).

R. Tempo is with the IEIIT-CNR, Politecnico di Torino, 10129 Torino, Italy (e-mail: roberto.tempo@polito.it).

Digital Object Identifier 10.1109/TAC.2005.849187

[1]Clearly, the boundary points of the two sets $\mathcal{X}_i(x)$, $i = 1, 2$, overlap, and the term partition is used in a nonstrict manner.

[2]When the state x is on the boundary of $\mathcal{X}_i$, the switching signal σ may take the value of either 1 or 2. This should not affect the solution of (1) under the condition that there is no chattering, i.e., that there are only finite switchings in finite time. We should note, however, that the switching rule (3) does not take measures to prevent chattering. For details on related issues, see [15], [17], and [24].

and is guaranteed to converge in a probabilistic sense. The advantage of this approach is that it can be extended to the general case without introducing the conservatism mentioned above. On the other hand, a motivation for taking a probabilistic approach is the computational complexity of the problem. This approach has been a subject of active research in robust control; see, e.g., [22]. As mentioned above, the problem is nonconvex, and the structure of the problem makes the complexity high. For example, even to check deterministically whether given matrices P_1 and P_2 satisfy (2) entails indefinite quadratic programming, which is known to be NP-complete [4], [20]. In contrast, the complexity of the proposed probabilistic algorithms does not depend on the problem size such as the dimension of the switched systems.

Here, in particular, we follow an approach based on stochastic approximation algorithms [14], which was introduced in [21] in the context of robust control. This approach has been found useful in exploiting the structure of control problems and in developing low complexity randomized algorithms; for recent results, see, e.g., [9], [11], [13], [16], and [19]. These works, however, deal only with convex problems.

The novelty of this paper is in proposing a probabilistic algorithm under this scheme for a nonconvex problem, by studying the special structure of the condition (2). It turns out that the switched system problem belongs to a class that can be viewed as a generalization of the convex ones considered previously. As a result, a broader notion of convergence is required in the proposed algorithm: It is not possible to generate one sequence of (P_1, P_2)-pairs that is guaranteed to converge to a solution as in earlier works. Instead, in this paper, the algorithm generates a set of such sequences among which at least one is known to be converging. To maintain the number of sequences at a practical level, we also present its modified, heuristic version based on a branch-and-bound type method.[3] Furthermore, these techniques are applied to switching rule synthesis for more general cases; we consider switched linear systems with more than three subsystems as well as switched nonlinear systems.

This paper is organized as follows. In Section II, we state the problem and give several preliminary results. In Section III, we present the randomized algorithm together with the main theorem which shows its probabilistic convergence. This algorithm is then modified to reduce its complexity first by relaxing the requirement to probabilistic solutions in Section IV and further by introducing a branch-and-bound method in Section V. The multimodal case is presented in Section VI. In Section VII, we discuss the general class of problems to which the switched system problem belongs. Finally, some numerical examples are given in Section VIII, followed by concluding remarks in Section IX.

Notation: For $x \in \mathbb{R}^n$, $\|x\|$ denotes its Euclidean norm. The space of symmetric matrices in $\mathbb{R}^{n\times n}$ is equipped with the inner product $\langle X, Y\rangle = \mathrm{tr}XY$ and the Frobenius norm $\|X\| = \left(\Sigma_{i,k=1}^n x_{i,k}^2\right)^{1/2}$, where $x_{i,k}$ is the (i,k)th entry of the matrix X. Let $\mathcal{C} \subset \mathbb{R}^{n\times n}$ be the cone of nonnegative–definite matrices. We define the projection of a symmetric $n \times n$ matrix X on $\mathcal{C}$ by $X^+ := \arg\min_{Y\in\mathcal{C}} \|X - Y\|$.

We extend the inner product and the norm on the space of symmetric matrices in $\mathbb{R}^{n\times n}$ to the product space $\mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ in a natural way. The inner product is defined by $\langle (X_1, X_2), (Y_1, Y_2)\rangle = \langle X_1, Y_1\rangle + \langle X_2, Y_2\rangle$ and the norm by $\|(X_1, X_2)\| = \left(\|X_1\|^2 + \|X_2\|^2\right)^{1/2}$. We also use the notation $\mathcal{C}^2 = \mathcal{C} \times \mathcal{C}$ for the set of nonnegative–definite matrix pairs.

II. Problem Formulation

In this section, we give the problem statement and study the structure of the problem and in particular the solution set.

The problem of this paper is formally stated as follows: Given a scalar $\gamma > 0$, find positive–definite matrices P_1 and P_2, if they exist, such that for every $x \in \partial\mathcal{B}_0$ the condition (2) is satisfied.

We first introduce some sets in the space $\mathcal{C}^2$. Let $\mathcal{P}_s$ be the *solution set* for the problem:

$$\mathcal{P}_s := \Big\{ (P_1, P_2) \in \mathcal{C}^2 : \text{ for every } x \in \partial\mathcal{B}_0 \quad x'(P_1A_1 + A_1'P_1)x \le -\gamma \quad \text{if } x(P_1 - P_2)x \ge 0, \text{ and} \quad x'(P_2A_2 + A_2'P_2)x \le -\gamma \quad \text{if } x'(P_1 - P_2)x \le 0 \Big\}. \tag{4}$$

For $x \in \partial\mathcal{B}_0$, define the sets $\mathcal{P}_i(x), \mathcal{Q}_i(x) \subset \mathcal{C}^2$, $i = 1, 2$, by

$$\begin{aligned} \mathcal{P}_1(x) &:= \left\{(P_1, P_2) \in \mathcal{C}^2 : x'(P_1 - P_2)x \ge 0\right\} \\ \mathcal{P}_2(x) &:= \left\{(P_1, P_2) \in \mathcal{C}^2 : x'(P_1 - P_2)x \le 0\right\} \\ \mathcal{Q}_i(x) &:= \left\{(P_1, P_2) \in \mathcal{C}^2 : x'(P_iA_i + A_i'P_i)x \le -\gamma\right\}, \\ & \quad i = 1, 2. \end{aligned} \tag{5}$$

These sets allow us to study the properties of the problem in $\mathcal{C}^2$. First, there are two important facts about the problem; we state them as a lemma in the following.

Lemma 2.1:

i) The sets $\mathcal{P}_i(x)$, $i = 1, 2$, partition the set $\mathcal{C}^2$ for each $x \in \partial\mathcal{B}_0$.

ii) The solution set $\mathcal{P}_s$ can be written as $\mathcal{P}_s = \bigcap_{x\in\partial\mathcal{B}_0} \bigcup_{i=1,2} [\mathcal{P}_i(x) \cap \mathcal{Q}_i(x)]$. In other words, for each $x \in \partial\mathcal{B}_0$

$$\mathcal{P}_s \subset \bigcup_{i=1,2} [\mathcal{P}_i(x) \cap \mathcal{Q}_i(x)]. \tag{6}$$

Proof: i) This is obvious from the definition of $\mathcal{P}_i(x)$. ii) We must show that $(P_1, P_2) \in \mathcal{C}^2$ is in $\mathcal{P}_s$ if and only if for every $x \in \partial\mathcal{B}_0$, $[x'(P_1 - P_2)x \ge 0$ and $x'(A_1'P_1 + P_1A_1)x \le -\gamma]$ or $[x'(P_2 - P_1)x \ge 0$ and $x'(A_2'P_2 + P_2A_2)x \le -\gamma]$. For this, we prove

a) [(If A then B) and (If not A then C)] $>$

$\Leftrightarrow$ b) [(A and B) or (not A and C)].

[3] In this paper, we use the term branch-and-bound with a slight abuse of terminology. The proposed algorithm does not include pruning steps, whereas in the standard branch-and-bound algorithms, only candidates that are not solutions are eliminated; see, e.g., [1].

This is done as follows:

$$\begin{aligned}
\text{a)} &\Leftrightarrow [(\text{ not } A \text{ or } B) \text{ and } (A \text{ or } C)] \\
&\Leftrightarrow \{[\text{ not } A \text{ and } (A \text{ or } C)] \text{ or } [B \text{ and } (A \text{ or } C)]\} \\
&\Leftrightarrow \{[(\text{ not } A \text{ and } A) \text{ or } (\text{ not } A \text{ and } C)] \text{ or } \\
&\quad [(B \text{ and } A) \text{ or } (B \text{ and } C)]\} \\
&\Leftrightarrow [(\text{ not } A \text{ and } C) \text{ or } (A \text{ and } B) \text{ or } \\
&\quad (B \text{ and } C \text{ and } A) \text{ or } (B \text{ and } C \text{ and not } A)] \\
&\Leftrightarrow \text{b).}
\end{aligned}$$

Thus, we have $\mathcal{P}_s = \bigcap_{x\in\partial\mathcal{B}_0} \bigcup_{i=1,2} [\mathcal{P}_i(x) \cap \mathcal{Q}_i(x)]$. Now, (6) is immediate. □

We next make a few assumptions on the solution set $\mathcal{P}_s$. Define a subset $\widetilde{\mathcal{P}}_{s,\epsilon}$ of the solution set as follows. Given a scalar $\epsilon > 0$, let

$$\begin{aligned}
\widetilde{\mathcal{P}}_{s,\epsilon} := \{(P_1,P_2) \in \mathcal{C}^2 :\ &\text{for every} x \in \partial\mathcal{B}_0, \\
&x'(P_1A_1 + A_1'P_1)x \le -\gamma, \\
&\text{if } x'(P_1 - P_2)x \ge -\epsilon, \text{ and} \\
&x'(P_2A_2 + A_2'P_2)x \le -\gamma, \\
&\text{if } x'(P_1 - P_2)x \le \epsilon\}. \qquad (7)
\end{aligned}$$

Clearly, if $\widetilde{\mathcal{P}}_{s,\epsilon} \neq \emptyset$, then $\widetilde{\mathcal{P}}_{s,\epsilon} \subset \mathcal{P}_s$. Furthermore, this set is nonincreasing in the sense that for $\epsilon_1 \ge \epsilon_2 > 0$, we have $\widetilde{\mathcal{P}}_{s,\epsilon_1} \subset \widetilde{\mathcal{P}}_{s,\epsilon_2}$.

Throughout this paper, we assume that the following holds.

Assumption 2.2: The solution set $\mathcal{P}_s$ is nonempty, and further there exists a scalar $\epsilon > 0$ such that $\widetilde{\mathcal{P}}_{s,\epsilon}$ is nonempty.

We note that in general, the existence of a larger ϵ in the assumption means that the problem admits a larger solution set, as we will see later. Moreover, in the main algorithm presented in Section III, we will observe that the parameter ϵ has an influence over the convergence rate: Larger ϵ implies faster convergence. We also show in the numerical examples in Section VIII that the solutions obtained in fact satisfy this condition and, thus, it is not a vacuous condition.

Remark 2.3: Some technical comments on this assumption on feasibility of the problem are now in order. In general, this condition is difficult to check prior to running the proposed algorithms, and we need to know the size of ϵ. However, we first note that this assumption is standard in randomized algorithm techniques as in [6], [9], [11], [13], [16], [19], and [21]. It is usually stated in the form saying that the interior of the solution set $\mathcal{P}_s$ is nonempty; we can show that Assumption 2.2 is actually equivalent to this (see also Lemma 2.4). Furthermore, several methods are available to relax this assumption by studying probabilistic solutions instead of deterministic ones; see, e.g., [6], [18], and [22]. We have more discussions on this in Section IV.

In later sections, for more general classes of problems, we introduce similar conditions in Assumptions 6.1 and 7.2. The previous comments apply there as well. ▽

In view of this assumption, take $(P_1^*, P_2^*) \in \widetilde{\mathcal{P}}_{s,\epsilon}$. We first have a preliminary result, which we will make use of later. Here, $\sigma_{\max}(\cdot)$ denotes the maximum singular value of a matrix and $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue.

Lemma 2.4: Under Assumption 2.2, the following hold.

i) For $x \in \partial\mathcal{B}_0$ and $\alpha > 0$, if $(P_1,P_2) \in \mathcal{P}_i(x)$, then $(\alpha P_1, \alpha P_2) \in \mathcal{P}_i(x)$, $i = 1,2$.

ii) For $x \in \partial\mathcal{B}_0$ and $\alpha > 1$, if $(P_1,P_2) \in \mathcal{Q}_i(x)$, then $(\alpha P_1 + \Delta_1, \alpha P_2 + \Delta_2) \in \mathcal{Q}_i(x)$, $i = 1,2$, for any $\Delta_i = \Delta_i' \in \mathbb{R}^{n\times n}$, $i = 1,2$, satisfying

$$\|(\Delta_1,\Delta_2)\| \le \frac{\alpha - 1}{2} \times \min\left\{\frac{\gamma}{\sigma_{\max}(A_i)}, 2\lambda_{\min}(P_i) : \quad i = 1,2\right\}. \qquad (8)$$

iii) For $\alpha > 1$, $(\alpha P_1^* + \Delta_1, \alpha P_2^* + \Delta_2) \in \widetilde{\mathcal{P}}_{s,\epsilon}$ for any $\Delta_i = \Delta_i' \in \mathbb{R}^{n\times n}$, $i = 1,2$, satisfying $\|(\Delta_1,\Delta_2)\| \le [(\alpha - 1)/2]\min\{\epsilon, \gamma/\sigma_{\max}(A_i), 2\lambda_{\min}(P_i^*), \quad i = 1,2\}$.

Proof: i) This is obvious from the definition of $\mathcal{P}_i(x)$.

To show ii) and iii), take α and Δ_i satisfying the hypotheses and fix $x \in \partial\mathcal{B}_0$. ii) Suppose $(P_1,P_2) \in \mathcal{Q}_i(x)$, that is, $x'(P_iA_i + A_i'P_i)x \le -\gamma$ and $(P_1,P_2) \in \mathcal{C}^2$. Then

$$\begin{aligned}
x'[(\alpha P_i + \Delta_i)A_i + A_i'(\alpha P_i + \Delta_i)]x &\le -\alpha\gamma + 2\|x\|^2 \sigma_{\max} \\
&\quad \times (A_i)\sigma_{\max}(\Delta_i) \\
&\le -\gamma.
\end{aligned}$$

The last inequality holds because $\|x\| = 1$, $\sigma_{\max}(\Delta_i) \le \|\Delta_i\| \le \|(\Delta_1,\Delta_2)\|$, and (8). Moreover, again by (8), $(\alpha P_1 + \Delta_1, \alpha P_2 + \Delta_2) \in \mathcal{C}^2$, Thus, the pair is in $\mathcal{Q}_i(x)$.

iii) First note that, as in ii), $(\alpha P_1^* + \Delta_1, \alpha P_2^* + \Delta_2) \in \mathcal{C}^2$. Next, we must show that if

$$x'\left[(\alpha P_1^* + \Delta_1) - (\alpha P_2^* + \Delta_2)\right]x \ge -\epsilon \qquad (9)$$

then $(\alpha P_1^* + \Delta_1, \alpha P_2^* + \Delta_2) \in \mathcal{Q}_1(x)$, and that if $x'[(\alpha P_1^* + \Delta_1) - (\alpha P_2^* + \Delta_2)]x \le \epsilon$ then $(\alpha P_1^* + \Delta_1, \alpha P_2^* + \Delta_2) \in \mathcal{Q}_2(x)$. We show the first part here, as the second part can be proven in the same manner.

Suppose (9) holds. Then, it follows that $x'(P_1^* - P_2^*)x \ge -\epsilon$, which can be shown easily through perturbation arguments as those in the proof of ii) above. Now, since (P_1^*, P_2^*) is in $\widetilde{\mathcal{P}}_{s,\epsilon}$, we clearly obtain $(P_1^*, P_2^*) \in \mathcal{Q}_1(x)$. This implies that $(\alpha P_1^* + \Delta_1, \alpha P_2^* + \Delta_2)$ is in $\mathcal{Q}_1(x)$ as well using ii). □

Note that because of iii) in the previous lemma, we can always take (P_1^*, P_2^*) sufficiently large that, in the same iii), $\lambda_{\min}(P_i^*)$ is larger than ϵ and $\gamma/\sigma_{\max}(A_i)$; in this way, the size of $\|(\Delta_1,\Delta_2)\|$ would not depend on the matrices P_i^*.

Several remarks are now in order. In Fig. 1, the sets of interest in the space $\mathcal{C}^2$ are sketched for a given $x \in \partial\mathcal{B}_0$; this figure summarizes the results obtained so far. First, by Lemma 2.1 i), $\mathcal{C}^2$ is partitioned into $\mathcal{P}_1(x)$ and $\mathcal{P}_2(x)$; the boundary of the two sets is represented as a line because of i) in Lemma 2.4. Second, the sets $\mathcal{Q}_i(x)$ contain cones, as we see in Lemma 2.4 ii) and, hence, are unbounded. Third, the solution set $\mathcal{P}_s$ too contains a cone by iii) in the same lemma and, furthermore, is in $\bigcup_{i=1,2}[\mathcal{P}_i(x) \cap \mathcal{Q}_i(x)]$ due to (6).

Now, to construct a gradient-based iterative algorithm, it is important to address the following question: Starting from a given $(P_1,P_2) \in \mathcal{C}^2$ with information about $\mathcal{P}_i(x)$ and $\mathcal{Q}_i(x)$, to which direction should we move (P_1,P_2) to make it closer to the solution set $\mathcal{P}_s$? The best we know is that there are two

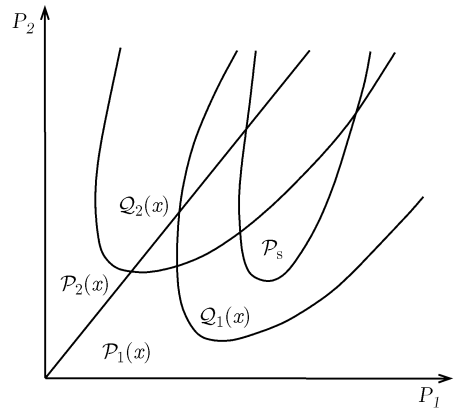


Fig. 1. Sketch of the sets in $\mathcal{C}^2$ for $x \in \partial\mathcal{B}_0$.

possible directions, one toward $\mathcal{P}_1(x) \cap \mathcal{Q}_1(x)$ and the other toward $\mathcal{P}_2(x) \cap \mathcal{Q}_2(x)$. This is due to the fact that the problem is not convex. We will make use of this idea in the algorithm presented in Section III.

We introduce some additional notation and a problem equivalent to the switched system problem. For $(P_1, P_2) \in \mathcal{C}^2$ and $x \in \partial\mathcal{B}_0$, let

$$\begin{aligned} v_1(P_1,P_2,x) :=& \{[-x'(P_1-P_2)x]^+\}^2 \\ &+ \{[x'(P_1A_1 + A_1'P_1)x + \gamma]^+\}^2 \\ v_2(P_1,P_2,x) :=& \{[-x'(P_2-P_1)x]^+\}^2 \\ &+ \{[x'(P_2A_2 + A_2'P_2)x + \gamma]^+\}^2. \end{aligned}$$

These functions $v_i(\cdot,\cdot,x)$ are convex in (P_1, P_2) for fixed x. Clearly, we have for $(P_1, P_2) \in \mathcal{C}^2$

$$v_i(P_1,P_2,x) = 0 \Leftrightarrow (P_1,P_2) \in \mathcal{P}_i(x) \cap \mathcal{Q}_i(x). \tag{10}$$

In view of Lemma 2.1 ii), the problem of this paper can now be expressed as the following equivalent one: Given $\gamma > 0$, find $(P_1, P_2) \in \mathcal{C}^2$ such that, for every $x \in \partial\mathcal{B}_0$

$$\text{either } v_1(P_1,P_2,x) = 0 \text{ or } v_2(P_1,P_2,x) = 0. \tag{11}$$

Condition (11) shows the nonconvexity of the problem more explicitly since each function v_i is convex in the P matrices. We will say more on this structure of the problem and its generalization in Sections VI and VII.

For completeness, we give their subgradients, which are used in the proposed algorithm in Section III, as follows:

$$\begin{aligned} \nabla_{P_1} v_1(P_1,P_2,x) =& -2\left[-x'(P_1-P_2)x\right]^+ (xx') \\ &+ 2\left[x'(P_1A_1 + A_1'P_1)x + \gamma\right]^+ \\ &\times (xx'A_1' + A_1xx') \\ \nabla_{P_2} v_1(P_1,P_2,x) =& 2\left[-x'(P_1-P_2)x\right]^+ (xx') \\ \nabla_{P_1} v_2(P_1,P_2,x) =& 2\left[-x'(P_2-P_1)x\right]^+ (xx') \\ \nabla_{P_2} v_2(P_1,P_2,x) =& -2\left[-x'(P_2-P_1)x\right]^+ (xx') \\ &+ 2\left[x'(P_2A_2 + A_2'P_2)x + \gamma\right]^+ \\ &\times (xx'A_2' + A_2xx'). \end{aligned} \tag{12}$$

As the arguments of the projection $[\cdot]^+$ are scalars, the derivation is straightforward.

III. A Randomized Algorithm for Switched Systems

In this section, we present a randomized algorithm which finds a pair (P_1, P_2) to be used in the switching rule (3) for the switched system (1), and state and prove the main theorem that shows its convergence in a probabilistic sense.

The algorithm involves random generation of states x in the unit sphere $\partial\mathcal{B}_0$. We introduce the probability density function $f_x : \partial\mathcal{B}_0 \to \mathbb{R}_+$ for the states. We assume that

$$f_x(x) > 0 \text{ for } x \in \partial\mathcal{B}_0 \tag{13}$$

since all x on the sphere must satisfy the conditions in (2). One possible choice would be the uniform density function. In particular, we prove that the proposed algorithm converges in a finite number of steps with probability one, where the probability is taken with respect to the sequence of random samples.

In the iterative algorithm, we employ two types of indices. Square brackets are used to represent the step numbers: At step k, we have a state $x^{[k]}$. On the other hand, round brackets contain sequences of 1 and 2 whose lengths are equal to the step number; these are used for the candidate pairs (P_1, P_2), the scalars μ, ν, and so on; for example, at step 1, we obtain the matrix pairs $\left(P_1^{(1)}, P_2^{(1)}\right)$ and $\left(P_1^{(2)}, P_2^{(2)}\right)$; at step 2), we obtain $\left(P_1^{(1,1)}, P_2^{(1,1)}\right)$, $\left(P_1^{(1,2)}, P_2^{(1,2)}\right)$, $\left(P_1^{(2,1)}, P_2^{(2,1)}\right)$, and $\left(P_1^{(2,2)}, P_2^{(2,2)}\right)$; $\cdots$; and at step k, we obtain $\left(P_1^{(j_1,j_2,\ldots,j_k)}, P_2^{(j_1,j_2,\ldots,j_k)}\right)$, $j_1, j_2, \ldots, j_k = 1, 2$.

Now we are ready to give the iterative randomized algorithm for the switched system(1).

Algorithm 3.1:

0. Set an initial pair $(P_{10}, P_{20}) \in \mathcal{C}^2$ and set

$$r \in \left(0, \frac{\epsilon}{4}\right]. \tag{14}$$

1. At step 1, generate $x^{[1]} \in \partial\mathcal{B}_0$ according to the density function f_x in **(13)**. Obtain two pairs $\left(P_1^{(j)}, P_2^{(j)}\right)$, $j = 1, 2$, as follows:

For $j = 1, 2$, set **(15)**, as shown at the bottom of the page, where $\nabla_i v^{(j)} := \nabla_{P_i} v_j\left(P_{10}, P_{20}, x^{[1]}\right)$, $i = 1, 2$, and the step sizes $\mu^{(j)}$ are given by

$$\mu^{(j)} := \frac{v_j\left(P_{10}, P_{20}, x^{[0,1]}\right) + r\left\|\left(\nabla_1 v^{(j)}, \nabla_2 v^{(j)}\right)\right\|}{\left\|\left(\nabla_1 v^{(j)}, \nabla_2 v^{(j)}\right)\right\|^2}.$$

$$\left(P_1^{(j)}, P_2^{(j)}\right) = \begin{cases} \left(\left[P_{10} - \mu^{(j)}\nabla_1 v^{(j)}\right]^+, \left[P_{20} - \mu^{(j)}\nabla_2 v^{(j)}\right]^+\right), & \text{if } v_j\left(P_{10}, P_{20}, x^{[1]}\right) > 0 \\ (P_{10}, P_{20}), & \text{otherwise} \end{cases} \tag{15}$$

2. At step 2, generate $x^{[2]} \in \partial\mathcal{B}_0$ according to the density function f_x in **(13)**. Then, obtain four pairs $\left(P_1^{(j_1,j_2)}, P_2^{(j_1,j_2)}\right)$, $j_1, j_2 = 1, 2$, similarly to step 1, as follows. (For $j_1, j_2 = 1, 2$, replace (P_{10}, P_{20}) and $\left(P_1^{(j_2)}, P_2^{(j_2)}\right)$ with $\left(P_1^{(j_1)}, P_2^{(j_1)}\right)$ and $\left(P_1^{(j_1,j_2)}, P_2^{(j_1,j_2)}\right)$, respectively.)

For $j_1, j_2 = 1, 2$, set **(16)**, as shown at the bottom of the page, where $\nabla_i v^{(j_1,j_2)} := \nabla_{P_i} v_{j_2}\left(P_1^{(j_1)}, P_2^{(j_1)}, x^{[2]}\right)$, $i = 1, 2$, and the step sizes $\mu^{(j_1,j_2)}$ are given by

$$\mu^{(j_1,j_2)} := \frac{v_{j_2}\left(P_1^{(j_1)}, P_2^{(j_1)}, x^{[2]}\right) + r\left\|\left(\nabla_1 v^{(j_1,j_2)}, \nabla_2 v^{(j_1,j_2)}\right)\right\|}{\left\|\left(\nabla_1 v^{(j_1,j_2)}, \nabla_2 v^{(j_1,j_2)}\right)\right\|^2}.$$

$\vdots$

k. At step k, generate $x^{[k]} \in \partial\mathcal{B}_0$ according to f_x in **(13)**. Obtain 2^k pairs $\left(P_1^{(j_1,j_2,\ldots,j_k)}, P_2^{(j_1,j_2,\ldots,j_k)}\right)$, $j_1, j_2, \ldots, j_k = 1, 2$, similarly to step 1. (For $j_1, j_2, \ldots, j_k = 1, 2$, replace (P_{10}, P_{20}) and $\left(P_1^{(j_k)}, P_2^{(j_k)}\right)$ with $\left(P_1^{(j_1,\ldots,j_{k-1})}, P_2^{(j_1,\ldots,j_{k-1})}\right)$ and $\left(P_1^{(j_1,\ldots,j_{k-1},j_k)}, P_2^{(j_1,\ldots,j_{k-1},j_k)}\right)$, respectively.)

We say that an *update* in (P_1, P_2) occurs when a new pair is obtained through any iteration in, e.g., (15) and (16). Note that the choice of the initial pair $(P_{10}, P_{20}) \in \mathcal{C}^2$ is arbitrary.

Theorem 3.2: Suppose that Assumption 2.2 holds. Then, Algorithm 3.1 converges to a solution in a finite number of steps with probability one. Specifically, there exists $k \in \mathbb{N}$ such that at least one of the pairs in $\left\{\left(P_1^{(j_1,\ldots,j_k)}, P_2^{(j_1,\ldots,j_k)}\right) : j_1, \ldots, j_k = 1, 2\right\}$ is a solution.

The proof of the theorem is preceded by several technical results. Take $\alpha > 1$ with

$$\alpha \geq 1 + 2r \max\left\{\frac{1}{\epsilon}, \frac{\sigma_{\max}(A_i)}{\gamma} : \quad i = 1, 2\right\} \tag{17}$$

where r is given in (14) in the algorithm. Let

$$\left(\hat{P}_1^*, \hat{P}_2^*\right) := (\alpha P_1^*, \alpha P_2^*). \tag{18}$$

Now, denote by $\mathcal{B}_{\hat{P}^*}(r)$ the ball in $\mathcal{C}^2$ with center $\left(\hat{P}_1^*, \hat{P}_2^*\right)$ and radius r

$$\mathcal{B}_{\hat{P}^*}(r) := \left\{(P_1, P_2) \in \mathcal{C}^2 : \left\|(P_1, P_2) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\| \leq r\right\}. \tag{19}$$

This ball $\mathcal{B}_{\hat{P}^*}(r)$ has several properties.

Lemma 3.3:

i) $\mathcal{B}_{\hat{p}^*}(r) \subset \widetilde{\mathcal{P}}_{s,\epsilon}$.

ii) For every $x \in \partial\mathcal{B}_0$, one of the following conditions holds:

a) $\mathcal{B}_{\hat{P}^*}(r) \subset \mathcal{P}_1(x) \cap \mathcal{Q}_1(x)$;

b) $\mathcal{B}_{\hat{P}^*}(r) \subset \mathcal{P}_2(x) \cap \mathcal{Q}_2(x)$;

c) $\mathcal{B}_{\hat{P}^*}(r) \subset \bigcap_{i=1,2} \mathcal{Q}_i(x)$.

Proof: i) This is a direct consequence of the choice of α in (17) and Lemma 2.4 iii).

ii) Fix $x \in \partial\mathcal{B}_0$. By i), the ball $\mathcal{B}_{\hat{P}^*}(r)$ is contained in the solution set $\mathcal{P}_s$. Thus, the property of $\mathcal{P}_s$ in (6) implies that $\mathcal{B}_{\hat{P}^*}(r) \subset \bigcup_{i=1,2}[\mathcal{P}_i(x) \cap \mathcal{Q}_i(x)]$. Hence, clearly either a) or b) holds, or otherwise the ball is not contained in only one of $\mathcal{P}_i(x) \cap \mathcal{Q}_i(x)$, $i = 1, 2$. We must show that in the last case, c) follows.

By i) in Lemma 2.1, the center $\left(\hat{P}_1^*, \hat{P}_2^*\right)$ of the ball must be in either $\mathcal{P}_1(x)$ or $\mathcal{P}_2(x)$. We claim that, in either case, every (P_1, P_2) in the ball $\mathcal{B}_{\hat{P}^*}(r)$ satisfies $x'(P_1 - P_2)x \in [-\epsilon, \epsilon]$. To show this is straightforward using $r \leq \epsilon/4$ in (14); follow perturbation arguments as in the proof of Lemma 2.4.

Here, observe that by definition, if a pair (P_1, P_2) in $\widetilde{\mathcal{P}}_{s,\epsilon}$ satisfies $x'(P_1 - P_2)x \in [-\epsilon, \epsilon]$, then the pair necessarily is contained in $\bigcap_{i=1,2} \mathcal{Q}_i(x)$. By (i), the ball $\mathcal{B}_{\hat{P}^*}(r)$ is a subset of $\widetilde{\mathcal{P}}_{s,\epsilon}$ and, therefore, is in $\bigcap_{i=1,2} \mathcal{Q}_i(x)$. □

The next proposition is the key result for the proof of the theorem. The idea discussed at the end of Section II is stated in a formal way for the initial step in the algorithm. Recall that in step 1), we start with (P_{10}, P_{20}) and generate two pairs $\left(P_1^{(j)}, P_2^{(j)}\right)$, $j = 1, 2$. It is shown that at least one pair is closer to a solution if there were any updates in generating it.

Proposition 3.4: In Algorithm 3.1, suppose that $(P_{10}, P_{20}) \notin \mathcal{P}_s$. At step 1, there exists $j \in \{1, 2\}$ such that

i) if there is no update in(15) in generating $\left(P_1^{(j)}, P_2^{(j)}\right)$, then $\left(P_1^{(j)}, P_2^{(j)}\right) \in \mathcal{P}_j(x^{[1]}) \cap \mathcal{Q}_j(x^{[1]})$ and $\mathcal{P}_s \cap [\mathcal{P}_j(x^{[1]}) \cap \mathcal{Q}_j(x^{[1]})] \neq \emptyset$;

ii) otherwise, $\left\|(P_1^{(j)}, P_2^{(j)}) - (\hat{P}_1^*, \hat{P}_2^*)\right\|^2 \leq \left\|(P_{10}, P_{20}) - (\hat{P}_1^*, \hat{P}_2^*)\right\|^2 - r^2$.

Proof: Due to the definition of f_x in (13), $x^{[1]} \in \partial\mathcal{B}_0$, and hence with respect to the ball $\mathcal{B}_{\hat{P}^*}(r)$ there are three cases a)–c) as in Lemma 3.3 ii). We consider the three cases separately.

Case (a): Suppose $\mathcal{B}_{\hat{P}^*}(r) \subset \mathcal{P}_1(x^{[1]}) \cap \mathcal{Q}_1(x^{[1]})$. Then, with $j = 1$, i) and ii) hold.

Proof: i) If there is no update, it follows that $\left(P_1^{(1)}, P_2^{(1)}\right) = (P_{10}, P_{20}) \in \mathcal{P}_1(x^{[1]}) \cap \mathcal{Q}_1(x^{[1]})$ by (10). Moreover, the ball $\mathcal{B}_{\hat{P}^*}(r)$ is in $\mathcal{P}_s$ by Lemma 3.3 i), and, by the hypothesis, the ball clearly intersects with $\mathcal{P}_1(x^{[1]}) \cap \mathcal{Q}_1(x^{[1]})$.

ii) An update in (15) means that

$$v_1(P_{10}, P_{20}, x^{[1]}) > 0. \tag{20}$$

$$\left(P_1^{(j_1,j_2)}, P_2^{(j_1,j_2)}\right) = \begin{cases} \left(\left[P_1^{(j_1)} - \mu^{(j_1,j_2)}\nabla_1 v^{(j_1,j_2)}\right]^+, \left[P_2^{(j_1)} - \mu^{(j_1,j_2)}\nabla_2 v^{(j_1,j_2)}\right]^+\right), & \text{if } v_{j_2}\left(P_1^{(j_1)}, P_2^{(j_1)}, x^{[2]}\right) > 0 \\ \left(P_1^{(j_1)}, P_2^{(j_1)}\right), & \text{otherwise} \end{cases} \tag{16}$$

Let

$$(\hat{P}_1, \hat{P}_2) := (\hat{P}_1^*, \hat{P}_2^*) + \frac{r}{\left\|\left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right)\right\|} \times \left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right).$$

Notice that $(\hat{P}_1, \hat{P}_2) \in \mathcal{B}_{\hat{P}^*}(r)$. Hence, by hypothesis, $(\hat{P}_1, \hat{P}_2) \in \mathcal{P}_1(x^{[1]}) \cap \mathcal{Q}_1(x^{[1]})$, that is, by (10)

$$v_1(\hat{P}_1, \hat{P}_2, x^{[1]}) = 0. \tag{21}$$

Now, by the update law (15) and by the properties of the projection $[\cdot]^+$ (see, e.g., [11, Lemma 5.3])

$$\begin{aligned}
&\left\|\left(P_1^{(1)}, P_2^{(1)}\right) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\|^2 \\
&\quad \le \left\|(P_{10}, P_{20}) - \mu^{(1)}\left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\|^2 \\
&\quad = \left\|(P_{10}, P_{20}) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\|^2 \\
&\qquad + (\mu^{(1)})^2 \left\|\left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right)\right\|^2 \\
&\qquad - 2\mu^{(1)} \left\langle \left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right), \right. \\
&\qquad\qquad \left. (P_{10}, P_{20}) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\rangle.
\end{aligned} \tag{22}$$

This last term can be written as

$$\begin{aligned}
&\left\langle \left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right), (P_{10}, P_{20}) - (\hat{P}_1^*, \hat{P}_2^*)\right\rangle \\
&\quad = \left\langle \left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right), (P_{10}, P_{20}) - (\hat{P}_1, \hat{P}_2)\right\rangle \\
&\qquad + \left\langle \left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right), (\hat{P}_1, \hat{P}_2) - (\hat{P}_1^*, \hat{P}_2^*)\right\rangle.
\end{aligned} \tag{23}$$

Here, because of the subgradients $\nabla_1 v^{(1)}$ and $\nabla_2 v^{(1)}$ and (21), the first term in the right-hand side of (23) can be expressed as

$$\begin{aligned}
&\left\langle \left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right), (P_{10}, P_{20}) - (\hat{P}_1, \hat{P}_2)\right\rangle \\
&\quad \ge v_1(P_{10}, P_{20}, x^{[1]}) - v_1(\hat{P}_1, \hat{P}_2, x^{[1]}) \\
&\quad \ge v_1(P_{10}, P_{20}, x^{[1]}).
\end{aligned} \tag{24}$$

On the other hand, the second term in the right-hand side of (23) becomes

$$\begin{aligned}
&\left\langle \left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right), (\hat{P}_1, \hat{P}_2) - (\hat{P}_1^*, \hat{P}_2^*)\right\rangle \\
&\qquad = r \left\|\left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right)\right\|
\end{aligned} \tag{25}$$

by the definition of $(\hat{P}_1, \hat{P}_2)$. Then, substituting (23)–(25) into (22) yields

$$\begin{aligned}
\left\|\left(P_1^{(1)}, P_2^{(1)}\right) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\|^2 &\le \left\|(P_{10}, P_{20}) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\|^2 \\
&\quad - \left(\frac{v_1(P_{10}, P_{20}, x^{[1]}) + r\left\|\left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right)\right\|}{\left\|\left(\nabla_1 v^{(1)}, \nabla_2 v^{(1)}\right)\right\|}\right)^2.
\end{aligned}$$

Using (20), we now obtain the inequality in ii) with $j = 1$. $\triangledown$

Case b): Suppose $\mathcal{B}_{\hat{P}^*}(r) \subset \mathcal{P}_2(x^{[1]}) \cap \mathcal{Q}_2(x^{[1]})$. Then, with $j = 2$, i) and ii) hold.

This can be proven similarly to the Case a). Before we consider the last case, notice that, by i) in Lemma 2.1, (P_{10}, P_{20}) is in either $\mathcal{P}_i(x^{[1]})$, $i = 1, 2$.

Case c): Suppose $\mathcal{B}_{\hat{P}^*}(r) \subset \bigcap_{i=1,2} \mathcal{Q}_i(x)$. Let $j \in \{1, 2\}$ be such that $(P_{10}, P_{20}) \in \mathcal{P}_j(x^{[1]})$. Then, with this j, i) and ii) hold.

Proof: i) It follows from i) in Lemma 3.3 that the ball $\mathcal{B}_{\hat{P}^*}(r)$ is contained in the solution set $\mathcal{P}_s$. This together with the hypothesis suggests that the ball intersects with $\mathcal{P}_i(x^{[1]}) \cap \mathcal{Q}_i(x^{[1]})$ for both $i = 1, 2$. Hence, if there is no update, that is, if (P_{10}, P_{20}) is in $\mathcal{P}_j(x^{[1]}) \cap \mathcal{Q}_j(x^{[1]})$, then clearly i) holds.

ii) Since (P_{10}, P_{20}) is in $\mathcal{P}_j(x^{[1]})$, if an update occurs, it means that the pair is not in $\mathcal{Q}_j(x^{[1]})$. Let $w_j(P_1, P_2, x) := ([x'(P_j A_j + A_j' P_j)x + \gamma]^+)^2$. Then, clearly, $v_j(P_{10}, P_{20}, x^{[1]}) = w_j(P_{10}, P_{20}, x^{[1]}) > 0$ and also, by (12), $\nabla_{P_i} v_j(P_{10}, P_{20}, x^{[1]}) = \nabla_{P_i} w_j(P_{10}, P_{20}, x^{[1]})$ for $i, j = 1, 2$. The inequality in ii) can be shown as in the Case a), but by replacing $\mathcal{P}_j(x^{[1]}) \cap \mathcal{Q}_j(x^{[1]})$ with $\mathcal{Q}_j(x^{[1]})$, $v_j(P_{10}, P_{20}, x^{[1]})$ with $w_j(P_{10}, P_{20}, x^{[1]}) > 0$, and similarly with the gradients. $\triangledown\square$

Proof of Theorem 3.2: Observe that, in Algorithm 3.1, in the beginning of step k, we have 2^{k-1} pairs $(P_1^{(j_1,\dots,j_{k-1})}, P_2^{(j_1,\dots,j_{k-1})})$ and that, during this step, for each pair, we obtain two new pairs $(P_1^{(j_1,\dots,j_{k-1},j_k)}, P_2^{(j_1,\dots,j_{k-1},j_k)})$, $j_k = 1, 2$.

Assume that, at step k, $\left(P_1^{(j_1,\dots,j_{k-1})}, P_2^{(j_1,\dots,j_{k-1})}\right) \notin \mathcal{P}_s$. We claim that there exists $j_k \in \{1, 2\}$ such that

i) if there is no update in obtaining $\left(P_1^{(j_1,\dots,j_{k-1},j_k)}, P_2^{(j_1,\dots,j_{k-1},j_k)}\right)$, then $\left(P_1^{(j_1,\dots,j_{k-1},j_k)}, P_2^{(j_1,\dots,j_{k-1},j_k)}\right) \in \mathcal{P}_{j_k}(x^{[k]}) \cap \mathcal{Q}_{j_k}(x^{[k]})$ and $\mathcal{P}_s \cap [\mathcal{P}_{j_k}(x^{[k]}) \cap \mathcal{Q}_{j_k}(x^{[k]})] \neq \emptyset$;

ii) otherwise

$$\begin{aligned}
&\left\|\left(P_1^{(j_1,\dots,j_{k-1},j_k)}, P_2^{(j_1,\dots,j_{k-1},j_k)}\right) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\|^2 \\
&\qquad \le \left\|\left(P_1^{(j_1,\dots,j_{k-1})}, P_2^{(j_1,\dots,j_{k-1})}\right) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\|^2 - r^2.
\end{aligned}$$

Notice, however, that since the iterations in step k are the same as those in step 1, it is sufficient to prove this claim for the case $k = 1$. This is shown in Proposition 3.4.

Applying this claim recursively, we have that for any given $k \in \mathbb{N}$ there is at least one sequence, denoted by $(j_1^*, \dots, j_k^*)$ in the sequel, satisfying the following: For each $m = 1, 2, \dots, k$, its truncated subsequence $(j_1^*, \dots, j_m^*)$ gives a pair $\left(P_1^{(j_1^*,\dots,j_m^*)}, P_2^{(j_1^*,\dots,j_m^*)}\right)$ such that, if it is not a solution, then with j_m^* i) and ii) hold; moreover, the number of subsequences for which ii) holds cannot be more than $\left\lfloor \left\|(P_{10}, P_{20}) - (\hat{P}_1^*, \hat{P}_2^*)\right\|^2 / r^2 \right\rfloor$, because otherwise the pair $\left(P_1^{(j_1^*,\dots,j_k^*)}, P_2^{(j_1^*,\dots,j_k^*)}\right)$ is a solution.

On the other hand, while the pair corresponding to this sequence is not a solution, the probability of generating $x^{[k]}$ that

forces an update to occur is nonzero, and hence the algorithm cannot stop updating. □

We have a corollary to this result, whose proof is contained in that of Theorem 3.2. It gives an upper bound on the number of updates for the algorithm to converge under the assumption that a bound on the solution pair $(\hat{P}_1^*, \hat{P}_2^*)$ in (18) is known.

Corollary 3.5: Suppose that Assumption 2.2 holds and that $\left\|(P_{10}, P_{20}) - (\hat{P}_1^*, \hat{P}_2^*)\right\| \leq D$ for some $D > 0$. Then, in Algorithm 3.1, the number of updates that occur in obtaining a solution pair $\left(P_1^{(j_1,\ldots,j_k)}, P_2^{(j_1,\ldots,j_k)}\right)$ from the initial pair (P_{10}, P_{20}) is smaller than $\lfloor (D/r)^2 \rfloor$. More specifically, there are at most $\lfloor (D/r)^2 \rfloor$ truncated subsequences $(j_1, \ldots, j_m)$, $m \in \{1, 2, \ldots, k\}$, whose corresponding pairs are obtained after updates.

The convergence of Algorithm 3.1 relies on the convexity of the functions $v_i(\cdot, \cdot, x)$ for fixed x. However, this convergence is a generalization of that introduced in, e.g., [6], [21] in the following sense: The algorithm generates a set of candidate (P_1, P_2)-pairs, of which at least one has been moving toward the solution set constantly each time there was an update; as discussed in Section II, this convergence is due to the nonconvex nature of the problem. In contrast, in previous works in which convex problems were considered, it would be only one candidate that is generated at each step by an algorithm.

Remark 3.6: The advantage of the approach in this paper is that it directly deals with the condition (2) and that the algorithm can be extended to the general case for three or more subsystems. We present results for this general case in Section VI. As we mentioned in the Introduction, there is no conservatism introduced in this extension. This is in contrast to the conventional approach using the S-procedure to reduce the problem to a BMI condition [7], [24]; for the general case, this condition is conservative. We also note that the BMI formulation may reduce the complexity to some extent and provide us with insights into the problem. In general, however, BMI problems are nonconvex and therefore finding a solution is computationally difficult; for more discussions, see, e.g., [10]. ▽

IV. Reducing Complexity: Probabilistic Solutions

In this section, we relax the objective of finding a deterministic solution to that of finding a probabilistic one and obtain an algorithm that converges in a finite number of steps. In particular, we employ the methods described in [18] and [22] and extend them to the nonconvex problem of switched systems.

More generally, the notion of probabilistically robust solutions has been studied in the field of robust control of uncertain systems. It makes use of probabilistic information on the system uncertainty and enables us to employ low-complexity algorithms such as Monte Carlo simulation techniques at the expense of introducing a small risk; for more on this general subject, see [23] and [22].

In the setup of the switched system problem, such solutions are defined as follows. First, for a matrix pair (P_1, P_2), let

$$p(P_1, P_2) := \text{Prob}\left\{v_1(P_1, P_2, x) = 0 \text{ or } v_2(P_1, P_2, x) = 0\right\}$$

that is, $p(P_1, P_2)$ is the probability that the pair satisfies the condition (11) with respect to the probability density function f_x on $\partial\mathcal{B}_0$ in (13). Let $p^* \in (0, 1)$. We say that a pair (P_1, P_2) is a *probabilistic solution with probability* p^* if

$$p(P_1, P_2) \geq p^*. \tag{26}$$

The quantity $p(P_1, P_2)$ cannot generally be computed exactly, but only an estimate of it can be obtained by adopting an *empirical* approach. More precisely, such an estimate can be determined using a set $\{x^{[l]}\}_{l=1}^N$ of random samples generated under f_x. Let

$$\hat{p}_N(P_1, P_2) := \frac{1}{N}\left|\left\{x^{[l]} : v_1\left(P_1, P_2, x^{[l]}\right) = 0 \text{ or } v_2\left(P_1, P_2, x^{[l]}\right) = 0, \; l = 1, 2, \ldots, N\right\}\right| \tag{27}$$

where $|\cdot|$ denotes the cardinality of a set. This estimate is usually called the *empirical probability*. Notice that the estimate itself is a random variable.

For an estimate of the above type to be reliable in a sequential algorithm where such estimates are made repeatedly, the question is how large the number N of samples must be. For this purpose, we introduce the function $N(\cdot)$ for sample size [18]. Let $\delta \in (0, 1)$ and let

$$N(\rho) := \left\lceil \frac{\left(\ln \frac{\pi^2(\rho+1)^2}{6\delta}\right)}{\ln \frac{1}{p^*}} \right\rceil, \qquad \rho \in \mathbb{N}. \tag{28}$$

We develop an algorithm which is a modified version of Algorithm 3.1. This algorithm has a predefined termination criterion for which the probability of not delivering a probabilistic solution with probability p^* is no greater than δ. In this sense, we say that the pair (P_1, P_2) that the algorithm produces is a *probabilistic solution with probability* p^* *and confidence level greater than* $1-\delta$. Moreover, another key idea is that the number of candidate matrix pairs is reduced by making use of the bound on updates in Corollary 3.5.

In the algorithm, we introduce a new variable ρ_0 and three new sequences $\{\rho^{(j_1,\ldots,j_k)}\}$, $\{y^{(j_1,\ldots,j_k)}\}$, and $\{z^{(j_1,\ldots,j_k)}\}$. These serve as counters to determine when to stop the algorithm. The variable $\rho_0 \in \mathbb{Z}_+$ is a global counter and is the number of updates that occurred in the algorithm up to that point. Thus, it provides labels when new matrix pairs are generated, where the initial pair (P_{10}, P_{20}) is labeled as 0. Each matrix pair $\left(P_1^{(j_1,\ldots,j_k)}, P_2^{(j_1,\ldots,j_k)}\right)$ is given such a label as the number $\rho^{(j_1,\ldots,j_k)} \in \mathbb{Z}_+$. The counter $y^{(j_1,\ldots,j_k)} \in \mathbb{Z}_+$ represents the number of updates that occurred to obtain the pair $\left(P_1^{(j_1,\ldots,j_k)}, P_2^{(j_1,\ldots,j_k)}\right)$ from the initial pair, while $z^{(j_1,\ldots,j_k)}$ is the number of steps after the last update for this pair. Also, $\mathcal{S}_k$ is a set of sequences $(j_1, \ldots, j_k)$ whose corresponding (P_1, P_2)-pairs are being considered as candidates.

The following is the randomized algorithm for finding probabilistic solutions.

Algorithm 4.1:

0. Set the parameters p^*, $\delta \in (0,1)$, the radius $r \in (0, \epsilon/4]$ of the ball in the solution set in **(19)**, and the initial matrix pair (P_{10}, P_{20}). Let $\mathcal{S}_1 = \{1,2\}$, $\mathcal{S}_2 = \{(j_1, j_2) : j_1, j_2 = 1,2\}$, and $\mathcal{S}_k = \emptyset$, $k \geq 3$. Also let $\rho_0 = 0$.

1. At step 1, execute step 1 of Algorithm 3.1. Then, for $j \in \mathcal{S}_1$, let

$$\left(\rho^{(j)}, y^{(j)}, z^{(j)}\right) = \begin{cases} (\rho_0 + 1, 1, 0), & \text{if } v_j\left(P_{10}, P_{20}, x^{[1]}\right) > 0 \\ (0,0,1), & \text{otherwise} \end{cases}$$

and let $\rho_0 = \rho_0 + 1$ if $v_j\left(P_{10}, P_{20}, x^{[1]}\right) > 0$.

$\vdots$

k. At step k, execute step k of Algorithm 3.1. Then, for $(j_1, \ldots, j_k) \in \mathcal{S}_k$,

i) let the equation shown at the bottom of the page hold and let $\rho_0 = \rho_0 + 1$ if $v_{j_k}\left(P_1^{(j_1,\ldots,j_{k-1})}, P_2^{(j_1,\ldots,j_{k-1})}, x^{[k]}\right) > 0$.

ii) If $z^{(j_1,\ldots,j_k)} = N(\rho^{(j_1,\ldots,j_k)})$, then return.

iii) If $y^{(j_1,\ldots,j_k)} = \lfloor (D/r)^2 \rfloor$, then generate random vectors $\{x^{[l]}\}_{l=1}^{N(\rho^{(j_1,\ldots,j_k)})}$ and check whether the empirical probability is equal to one. If it is, then return.

iv) If $y^{(j_1,\ldots,j_k)} < \lfloor (D/r)^2 \rfloor$, then update the set $\mathcal{S}_{k+1}$ by replacing it with $\mathcal{S}_{k+1} \cup \{(j_1, \ldots, j_k, j_{k+1}) : j_{k+1} = 1,2\}$.

The next proposition gives a probabilistic statement on the previous algorithm.

Proposition 4.2: Suppose that Assumption 2.2 holds and that $\left\|(P_{10}, P_{20}) - \left(\hat{P}_1^*, \hat{P}_2^*\right)\right\| \leq D$ for some $D > 0$. Then, Algorithm 4.1 terminates in a finite number of steps, and the pair obtained is a probabilistic solution with probability p^* [in the sense as defined in (26)] and confidence level greater than $1 - \delta$.

Proof: Recall that in the proof of Theorem 3.2, the sequence denoted by $(j_1^*, \ldots, j_k^*)$ has the property that the matrix pairs corresponding to its truncated sequences move constantly closer to the solution set after updates. We first claim that at any step k, the set $\mathcal{S}_k$ of candidate sequences contains $(j_1^*, \ldots, j_k^*)$. In step k iii), if $y^{(j_1,\ldots,j_k)} = \lfloor (D/r)^2 \rfloor$ but the empirical probability is less than one, then the sequence $(j_1, \ldots, j_k)$ cannot be $(j_1^*, \ldots, j_k^*)$. In steps k iii) and k iv), sequences containing this $(j_1, \ldots, j_k)$ as a truncated subsequence are not included in $\mathcal{S}_{k+1}, \mathcal{S}_{k+2}, \ldots$.

It is clear from the construction of the algorithm and the claim above that the algorithm terminates in a finite number of steps. In the worst case, there exists a finite k such that $y^{(j_1^*,\ldots,j_k^*)}$ equals $\lfloor (D/r)^2 \rfloor$ and, thus, by Corollary 3.5, it stops in step k iii).

We next show that the probability that the pair obtained at termination is a probabilistic solution with p^* is greater than $1 - \delta$. For this, we follow arguments similar to that in [18] and [22]. Notice that all distinct pairs generated in the algorithm are labeled as $0, 1, \ldots$; these are kept in $\rho^{(j_1,\ldots,j_k)}$. Denote the pair corresponding to the label $\rho \in \mathbb{Z}_+$ by $\left(\bar{P}_1^{[\rho]}, \bar{P}_2^{[\rho]}\right)$. We then define the following two events:

1) $M_\rho : \left(\bar{P}_1^{[\rho]}, \bar{P}_2^{[\rho]}\right)$ is generated, and no update occurs for the successive $N(\rho)$ random samples;
2) $B_\rho : \left(\bar{P}_1^{[\rho]}, \bar{P}_2^{[\rho]}\right)$ is generated, and $p\left(\bar{P}_1^{[\rho]}, \bar{P}_2^{[\rho]}\right) < p^*$.

Let $\text{Prob}'(\cdot)$ denote the infinite product of the probability measure induced by the density function f_x, i.e., it is a measure for an infinite sequence of random samples [2, Sec. 5.11]. Observe that for any ρ, $\text{Prob}'(M_\rho \cap B_\rho) = \text{Prob}'(M_\rho | B_\rho)\, \text{Prob}'(B_\rho) \leq \text{Prob}'(M_\rho | B_\rho) \leq (p^*)^{N(\rho)}$, where $\text{Prob}'(M_\rho | B_\rho)$ is the conditional probability of M_ρ given B_ρ. Thus, the probability that the algorithm terminates with a pair which is not a probabilistic solution is

$$\text{Prob}'\left[\bigcup_{\rho=0}^{\infty}(M_\rho \cap B_\rho)\right] \leq \sum_{\rho=0}^{\infty}(p^*)^{N(\rho)} \leq \frac{6\delta}{\pi^2}\sum_{\rho=0}^{\infty}\frac{1}{(\rho+1)^2} = \delta.$$

□

The previous statement evaluates the pair obtained at the end of the algorithm in a probabilistic sense. For this reason, it involves two probability measures: One is the measure induced by the density function f_x and is used for the probabilistic solution defined in (26). The other is for the entire sequence $x^{[1]}, x^{[2]}, \ldots, x^{[k]}$ generated in the algorithm; the length k is finite but not fixed in advance (it is the step when the algorithm terminates), and each sample is chosen independently based on the first measure. As a consequence, the number of samples used in the algorithm depends on two parameters p^* and δ.

While the algorithm seeks for probabilistic solutions instead of deterministic ones, an advantage is the explicit termination criterion, which distinguishes the algorithm above from Algorithm 3.1. We also emphasize that, as a probabilistic algorithm, the convergence rate of the current algorithm does not depend on the dimension of the A-matrices. The problem of finding a reasonable bound on the number of steps, however, remains open.

Another feature of this approach is that it may keep the number of candidate pairs low by leaving out the pairs not leading to a solution efficiently. In step k iii), if the empirical probability for the pair associated with the sequence $(j_1, \ldots, j_k)$ is smaller than 1, then all sequences of the form $(j_1, \ldots, j_k, j_{k+1}, \ldots, j_l)$ are excluded from $\mathcal{S}_l$ for $l > k$. This can be considered as a pruning stage in the algorithm, and is

$$\left(\rho^{(j_1,\ldots,j_k)}, y^{(j_1,\ldots,j_k)}, z^{(j_1,\ldots,j_k)}\right) = \begin{cases} \left(\rho_0 + 1, y^{(j_1,\ldots,j_{k-1})} + 1, 0\right), & \text{if } v_{j_k}\left(P_1^{(j_1,\ldots,j_{k-1})}, P_2^{(j_1,\ldots,j_{k-1})}, x^{[k]}\right) > 0 \\ \left(\rho^{(j_1,\ldots,j_{k-1})}, y^{(j_1,\ldots,j_{k-1})}, z^{(j_1,\ldots,j_{k-1})} + 1\right), & \text{otherwise} \end{cases}$$

effective if the bound $(D/r)^2$ is small and if updates occur often so that the candidate pair set $\mathcal{S}_k$ can be kept small throughout the run. Nevertheless, in general, the number of such candidate pairs still may increase exponentially, 2^k, with respect to k.

Remark 4.3: The proposed probabilistic algorithm is based on the techniques in [18] and [22] for convex problems. In these references, further extensions are developed for the case without an assumption on the existence of solutions (such as Assumption 2.2). These employ criteria for termination when a solution is unlikely to exist. As mentioned in Remark 2.3, these methods can also be adapted for the nonconvex problem of this paper. To avoid technical issues, however, we do not proceed along these lines. ▽

V. Modified Algorithm Using a Branch-and-Bound Type Method

We observed that in both Algorithms 3.1 and 4.1, the number of candidate (P_1, P_2)-pairs grows exponentially. This may not be practical from a computational viewpoint, especially because many pairs are not moving toward the solution set. In this section, we propose a practical version of the randomized algorithm, where the number of candidates is bounded by a fixed number, based on a branch-and-bound type method.

In the next algorithm, as the one in the previous section, we seek for probabilistic solutions with probability p^*. The idea is that, at each step, a fixed number (denoted by 2^M) of candidate pairs is generated using Algorithm 3.1, and among them the one with the maximum empirical probability is chosen to be the next candidate pair. Since parts of Algorithm 3.1 are called, to avoid confusion, we denote the matrix pair generated at the kth step by $\left(\tilde{P}_1^{[k]}, \tilde{P}_2^{[k]}\right)$. We also note that the indices are slightly different, but they follow the rules mentioned before Algorithm 3.1. For simplicity, we avoid the use of the counters y and z of Algorithm 4.1.

We now present the branch-and-bound type algorithm.

Algorithm 5.1:

0. Set an initial pair $\left(\tilde{P}_1^{[0]}, \tilde{P}_2^{[0]}\right) \in \mathcal{C}^2$, and set $r \in (0, \epsilon/4]$ and $M \in \mathbb{N}$.

k. At step $k \geq 1$, from $\left(\tilde{P}_1^{[k-1]}, \tilde{P}_2^{[k-1]}\right)$, obtain $\left(\tilde{P}_1^{[k]}, \tilde{P}_2^{[k]}\right)$ as follows.

i) Set $(P_{10}, P_{20}) = \left(\tilde{P}_1^{[k-1]}, \tilde{P}_2^{[k-1]}\right)$ and generate 2^M pairs $\left(P_1^{(j_1,j_2,\ldots,j_M)}, P_2^{(j_1,j_2,\ldots,j_M)}\right)$, $j_1, j_2, \ldots, j_M = 1, 2$, following the steps up to M in Algorithm 3.1. Then, set

$$\left(\tilde{P}_1^{[k]}, \tilde{P}_2^{[k]}\right) = \arg\max \hat{p}_{N(k2^M)}\left(P_1^{(j_1,j_2,\ldots,j_M)}, P_2^{(j_1,j_2,\ldots,j_M)}\right) \quad (29)$$

where the maximum is taken over the 2^M pairs and $N(\cdot)$ is the sample size function in **(28)**.

ii) If the maximum empirical probability in **(29)** (i.e., the one for $\left(\tilde{P}_1^{[k]}, \tilde{P}_2^{[k]}\right)$) is equal to one, then return.

For this algorithm, we have the following lemma.

Lemma 5.2: The pair (P_1, P_2) obtained at the termination of Algorithm 5.1 is a probabilistic solution with probability p^* (in the sense as defined in (26)) and confidence level greater than $1 - \delta$.

Proof: This can be shown in a similar way as the proof of Proposition 4.2 by following the arguments in [18] and [22]. A technical difference is that we label all the pairs generated in the algorithm, that is, all $\left(P_1^{(j_1,j_2,\ldots,j_M)}, P_2^{(j_1,j_2,\ldots,j_M)}\right)$, $j_1, j_2, \ldots, j_M = 1, 2$, obtained in each step k i). At the end of step k, labels up to $k2^M$ are given. □

We have several remarks. First, the number of candidate (P_1, P_2)-pairs is at most 2^M at any step in the algorithm. Hence, it is a more practical alternative for implementation of the algorithms in the previous sections. Second, the probability in the result is only for the pair that may be found in the algorithm; even if Assumption 2.2 holds, there is no guarantee for convergence. This is because the pruning stage in (29) is based on empirical probability; thus, as indicated earlier, the term branch-and-bound method is used in some approximated sense (see, e.g., [1]). In the original Algorithm 3.1, we observed that there is a sequence of (P_1, P_2)-pairs which converges to a solution in a monotonic way (i.e., the pairs corresponding to $\{(j_1^*, \ldots, j_k^*)\}$ in the proof of Theorem 3.2). Similarly, we may expect that in the branch-and-bound algorithm, the empirical probability increases as the step size grows. This, however, is difficult to show in general. Even if the pair at one step is contained in the set $\bigcup_{i=1,2}[\mathcal{P}_i(x) \cap \mathcal{Q}_i(x)]$ for some x, the pair at the next step may not be contained after an update.

VI. General Multimodal Systems

In this section, we describe an extension of the main results for the two subsystems case to the general L subsystems case.

The switched linear system with L subsystems is expressed as

$$\dot{x}(t) = A_{\sigma(t)}x(t), \text{ where } x(t) \in \mathbb{R}^n \text{ and } \sigma(t) \in \mathcal{I}_L \quad (30)$$

where $\mathcal{I}_L := \{1, 2, \ldots, L\}$ is the index set. All matrices A_i, $i \in \mathcal{I}_L$, are assumed to be non-Hurwitz because otherwise the problem becomes trivial. Further, we assume that these A-matrices do not have a convex combination which is Hurwitz. We note that, unlike in the two subsystems case, this assumption does not rule out the possibility that a quadratic Lyapunov function exists for the system. In other words, the existence of a Hurwitz combination is only a sufficient condition for quadratic stability; see [8] for details.

We aim at finding an asymptotically stabilizing switching rule $\sigma(t) = \arg\max\{x(t)'P_ix(t) : i \in \mathcal{I}_L\}$, where $P_i \in \mathbb{R}^{n\times n}$, $i \in \mathcal{I}_L$, are positive definite. That is, we would like to find L matrices $P_1, \ldots, P_L$ such that for each $x \in \partial\mathcal{B}_0$ and $i \in \mathcal{I}_L$

$$\text{if } x'P_ix = \max_{j\in\mathcal{I}_L} x'P_jx, \text{ then } x'(P_iA_i + A_i'P_i)x \leq -\gamma. \quad (31)$$

Once these matrices are found, we have a piecewise quadratic Lyapunov function of the form $V(x) := \max_{i\in\mathcal{I}_L} x'P_ix$ for (30).

Randomized algorithms similar to Algorithms 3.1, 4.1, and 5.1 can be constructed for this problem. In particular, we present

one that is analogous to Algorithm 3.1 in the following. Here, as the number L of the functions v_i increases, the complexity of the algorithm increases; at step k in the algorithm, L^k candidate sequences $(P_1, \ldots, P_L)$ of matrices are generated.

Again, denote by $\mathcal{P}_s$ the solution set, i.e., the set of all L sequences $(P_1, \ldots, P_L)$ of nonnegative matrices satisfying the condition (31). We introduce a subset $\widetilde{\mathcal{P}}_{s,\epsilon}$ for $\epsilon > 0$ of the solution set as

$$\widetilde{\mathcal{P}}_{s,\epsilon} := \Big\{ (P_1, \ldots, P_L) \in \mathcal{C}^L : \forall\, x \in \partial\mathcal{B}_0 \qquad \forall\, i \in \mathcal{I}_L \\ \text{if } x' P_i x \geq V(x)\epsilon, \ \text{ then } x'(P_i A_i + A_i P_i)x \\ \leq -\gamma \Big\}. \tag{32}$$

To show convergence of the algorithm, we need to make a technical assumption similar to Assumption 2.2.

Assumption 6.1: The solution set $\mathcal{P}_s$ is nonempty, and moreover there exists a scalar $\epsilon > 0$ such that the set $\widetilde{\mathcal{P}}_{s,\epsilon}$ in (32) is nonempty.

With a slight abuse of notation, we write the sequence of L symmetric matrices as $P = (P_1, \ldots, P_L) \in \mathcal{C}^L$. In the space of such sequences, we define the inner product by $\langle P, Q \rangle = \Sigma_{i \in \mathcal{I}_L} \langle P_i, Q_i \rangle$ and further the Frobenius norm by $\|P\| = \left(\Sigma_{i \in \mathcal{I}_L} \|P_i\|^2 \right)^{1/2}$. We also denote the projection of a sequence of symmetric matrices onto $\mathcal{C}^L$ by $P^+ = \left(P_1^+, \ldots, P_L^+\right)$.

Now, we rewrite the problem into the form similar to (11). Define L functions v_i, $i \in \mathcal{I}_L$, by

$$v_i(P, x) := \Sigma_{j \neq i} \left\{ [-x'(P_i - P_j)x]^+ \right\}^2 \\ + \left\{ [x'(P_i A_i + A_i' P_i)x + \gamma]^+ \right\}^2 \\ \text{for } P \in \mathcal{C}^L \text{ and } x \in \partial\mathcal{B}_0. \tag{33}$$

These functions are convex in the parameters of the matrices $P = (P_1, \ldots, P_L)$ for any fixed x. The problem (31) is equivalent to finding $P \in \mathcal{C}^L$ such that

$$\text{for every } x \in \partial\mathcal{B}_0, \text{ there exists } i \in \mathcal{I}_L \\ \text{satisfying } v_i(P, x) = 0. \tag{34}$$

The equivalence of the two problems can be shown as in Lemma 2.1. This problem clearly is nonconvex in P although it is stated in terms of convex functions $v_i(\cdot, x)$. We would like to exploit this structure.

The iterative randomized algorithm for the switched system problem for the L subsystems case is given as follows.

Algorithm 6.2:

0. Set $r \in (0, \epsilon/4]$ and set an initial sequence $P_0 = (P_{10}, \ldots, P_{L0}) \in \mathcal{C}^L$.

1. At step 1, generate $x^{[1]} \in \partial\mathcal{B}_0$ according to the density function f_x in **(13)**. Obtain L sequences $P^{(j)} \in \mathcal{C}^L$, $j \in \mathcal{I}_L$, as follows:

$$P^{(j)} = \begin{cases} \left[P_0 - \mu^{(j)} \nabla_P v_j \left(P_0, x^{[1]}\right)\right]^+, & \text{if } v_j\left(P_0, x^{[1]}\right) > 0 \\ P_0, & \text{otherwise} \end{cases}$$

where the step size is

$$\mu^{(j)} = \Big(v_j \left(P_0, x^{[1]}\right) + \\ r \left\| \nabla_P v_j \left(P_0, x^{[1]}\right) \right\| \Big) \Big/ \left\| \nabla_P v_j (P_0, x^{[1]}) \right\|^2.$$

$\vdots$

k. At step $k \geq 2$, generate $x^{[k]} \in \partial\mathcal{B}_0$ according to f_x. Obtain L^k sequences $P^{(j_1, \ldots, j_k)} \in \mathcal{C}^L$, $j_1, \ldots, j_k \in \mathcal{I}_L$, as shown in the equation at the bottom of the page,
where the second equation at the bottom of the page holds.

A result parallel to Theorem 3.2 can be developed showing that, in this gradient-based randomized algorithm, under Assumption 6.1, the algorithm converges in a finite number of steps with probability one, and more specifically, that at least one of the L^k matrix sequences at step k is constantly moving toward a certain ball with radius r contained in the solution set $\mathcal{P}_s$. As mentioned earlier, the proposed approach has the advantage of not introducing the conservatism that arises in the bilinear matrix inequality conditions.

VII. Generalizations and Discussions

In this section, we discuss some directions regarding further generalizations of the approach described in this paper.

The L multimodal switched systems problem in (34) in the previous section has a fairly general structure. Motivated by this, we consider here a class of problems of the following type: Given a bounded subset $\mathcal{X}$ of $\mathbb{R}^n$ and L functions $v_i : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $i \in \mathcal{I}_L := \{1, 2, \ldots, L\}$, suppose that $v_i(\cdot, x)$ is convex for any fixed $x \in \mathcal{X}$. Find $z \in \mathbb{R}^m$ such that, for every $x \in \mathcal{X}$, there is at least one $i \in \mathcal{I}_L$ such that $v_i(z, x) \leq 0$.

The randomized algorithm in Algorithm 6.2 can be applied to this class of problems with minor modifications, and the same convergence properties hold as well. The case with $L = 1$ is special in that the problem is convex and basically is the class of problems considered in, e.g., [6], [9], [11], [13], [16], [19], and

$$P^{(j_1, \ldots, j_k)} = \begin{cases} \left[P^{(j_1, \ldots, j_{k-1})} - \mu^{(j_1, \ldots, j_k)} \nabla_P v_{j_k} (P^{(j_1, \ldots, j_{k-1})}, x^{[k]})\right]^+, & \text{if } v_{j_k}(P^{(j_1, \ldots, j_{k-1})}, x^{[k]}) > 0 \\ P^{(j_1, \ldots, j_{k-1})}, & \text{otherwise} \end{cases}$$

$$\mu^{(j_1, \ldots, j_k)} = \frac{\left(v_{j_k}\left(P^{(j_1, \ldots, j_{k-1})}, x^{[k]}\right) + r \left\| \nabla_P v_{j_k}\left(P^{(j_1, \ldots, j_{k-1})}, x^{[k]}\right) \right\|\right)}{\left\| \nabla_P v_{j_k} (P^{(j_1, \ldots, j_{k-1})}, x^{[k]}) \right\|^2}$$

[21], following the probabilistic robustness approach. Hence, the problem class presented before can be viewed as a nontrivial generalization of them. This explains the need to introduce the generalized notion of convergence for the switched system problem as well as the previous problem.

We show in the following that the stabilization of switched nonlinear systems through its switching rule design can be formulated in this framework.

Example 7.1: Consider the switched system $\dot{x}(t) = f_{\sigma(t)}(x(t))$, where $x(t) \in \mathbb{R}^n$ and $\sigma(t) \in \{1,2\}$. Suppose that $f_i : \mathcal{X} \to \mathbb{R}^n$, $i = 1,2$, is locally Lipschitz with a bounded domain $\mathcal{X} \subset \mathbb{R}^n$ containing the origin and that the origin is an equilibrium for each subsystem $\dot{x} = f_i(x)$. The problem is to find a switching rule of the form (3) such that the origin of the switched system is locally asymptotically stable. In particular, we look for a piecewise quadratic Lyapunov function $V(x) = \max_{i=1,2} x'P_ix$ whose time derivative along solutions of the system is negative; that is, as in the linear case, for each $x \in \mathcal{X}$, there exists $i \in \{1,2\}$ such that $v_i(P_1,P_2,x) := \left\{[-x'(P_i - P_j)x]^+\right\}^2 + \left\{[2x'P_if_i(x) + \gamma x'x]^+\right\}^2 \le 0$, where $j = 1,2$, $j \neq i$. Once such a Lyapunov function is found, the origin is asymptotically stable within a level set $\{x : V(x) \le c\} \subset \mathcal{X}$ with some $c > 0$. Variants of Assumption 2.2 and Algorithm 3.1 for the nonlinear case can be developed in a parallel manner. ▽

The formulation of this paper applies to problems outside the context of switching rule synthesis for switched systems as well. One such problem is the stability analysis of linear differential inclusions using piecewise quadratic Lyapunov functions; see [4], [25], and the references therein. To capture these extensions, we introduce a technical assumption similar to but more general than Assumption 6.1. Define $\widetilde{\mathcal{P}}_i(x) := \{z \in \mathbb{R}^m : v_i(z,x) \le 0\}$ for $x \in \mathcal{X}$, $i \in \mathcal{I}_L$. (These correspond to $\mathcal{P}_i(x) \cup \mathcal{Q}_i(x)$ in the 2-modal switched system problem.)

Assumption 7.2: The solution set is nonempty and, further, there exists a ball $\mathcal{B}_{z^*}(r) \subset \mathbb{R}^m$ with center z^* and radius $r > 0$ such that: i) the ball is contained in the solution set and that: ii) for each $x \in \mathcal{X}$, $\mathcal{B}_{z^*}(r) \subset \widetilde{\mathcal{P}}_i(x)$ for some $i \in \mathcal{I}_L$.

Under this assumption, we can construct a randomized algorithm of the type in Algorithm 6.2 and prove its probabilistic convergence. This assumption is mild and may be relaxed depending on the problem, as we mentioned in Remark 2.3. For the switched (linear) system problem for two subsystems, it was assumed that the set $\widetilde{\mathcal{P}}_{s,\epsilon}$ in (7) is nonempty in Assumption 2.2; this is equivalent to i) in the assumption above. Moreover, the ball $\mathcal{B}_{\hat{P}^*}(r)$ has the properties stated in Lemma 3.3 ii); we point out that these properties resemble ii) in Assumption 7.2 as well.

Remark 7.3: We now discuss ellipsoid algorithm techniques (see, e.g., [4]) and their limitations when applied to the class of problems we have considered. The use of ellipsoid algorithms, instead of gradient ones, is one way to improve the convergence rate of the proposed algorithms in the probabilistic framework as well. In [13], such an algorithm is proposed for a problem with $L = 1$.

When applied to the general L case in (31), however, such an ellipsoid algorithm requires an additional assumption on top of Assumption 7.2. To guarantee its convergence in a finite number of steps, we need to assume that the ball $\mathcal{B}_{z^*}(r)$, whose location is usually unknown, is contained in the initial ellipsoid. This is due to the nonconvexity of the problem and can be explained as follows: The solution points contained in the initial ellipsoid are distributed among the L^k ellipsoids obtained at step k, and hence the volume of solution points in each candidate ellipsoid may become smaller and smaller; thus, finite-time convergence cannot be proven in general. The additional assumption stated above guarantees that any ellipsoid containing solution points contains the ball $\mathcal{B}_{z^*}(r)$, and hence that the algorithm stops after a finite number of iterations.

Another drawback of this additional assumption is that, in practice, it requires the initial ellipsoid to be large, since the location of the ball is unknown. This can make the branch-and-bound version of the algorithm numerically unstable for the following reason. Generally speaking, in ellipsoid algorithms, after an update the center of the new ellipsoid may not be closer to the solution set than the previous one. Thus, at the bounding stage, the chance of picking a wrong ellipsoid (i.e., one not containing any solution points) may be high. Note that once this happens, the algorithm does not converge. ▽

VIII. Numerical Examples

In this section, we present some numerical examples to illustrate the results.

We first applied the algorithm in Section V to construct a switching rule for the two-dimensional system whose A-matrices are given by

$$A_1 = \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1.5 & 2 \\ -2 & -0.5 \end{bmatrix}. \tag{35}$$

Their eigenvalues are 0 for A_1 and $0.5 \pm 1.73j$ for A_2. This is one of the typical examples in switched linear systems, where each subsystem is unstable, but by switching between them asymptotic stability can be achieved. Moreover, there is no Hurwitz convex combination of A_1 and A_2. This particular example has been studied in the literature; for instance, in [7], a multiple Lyapunov function is designed through a sufficient condition which involves linear matrix inequalities (LMIs).

In the simulation, we used Algorithm 5.1 with $r = 0.05$, $\gamma = 0.1$, and the initial pair $(P_{10}, P_{20}) = (I_2, 2I_2)$. Note that this pair is not a solution because $P_{10} < P_{20}$ and A_2 is unstable. For the computation of the empirical probability in (27), a set $\{x^{[l]}\}_{l=1}^N$ of states with $N = 2000$ was generated according to the uniform density function on $\partial\mathcal{B}_0$. [By (28), $N(\rho) = 2000$ corresponds to, e.g., $(\rho, p^*, \delta) = (120, 0.990, 1.00 \times 10^{-4})$, which covers all the examples in this section.] Each time parts of the original Algorithm 3.1 are called in the algorithm, $2^4 = 16$ pairs $(P_1^{(j_1,j_2,j_3,j_4)}, P_2^{(j_1,j_2,j_3,j_4)})$, $j_1, j_2, j_3, j_4 = 1, 2$, are generated, that is, with $M = 4$. After five steps, we obtained a satisfactory pair

$$\begin{aligned}(P_1, P_2) &= (\tilde{P}_1^{[5]}, \tilde{P}_2^{[5]}) \\ &= \left(\begin{bmatrix} 0.0475 & -0.084 \\ -0.0841 & 1.497 \end{bmatrix}, \begin{bmatrix} 0.180 & 0.456 \\ 0.456 & 1.153 \end{bmatrix} \right).\end{aligned} \tag{36}$$

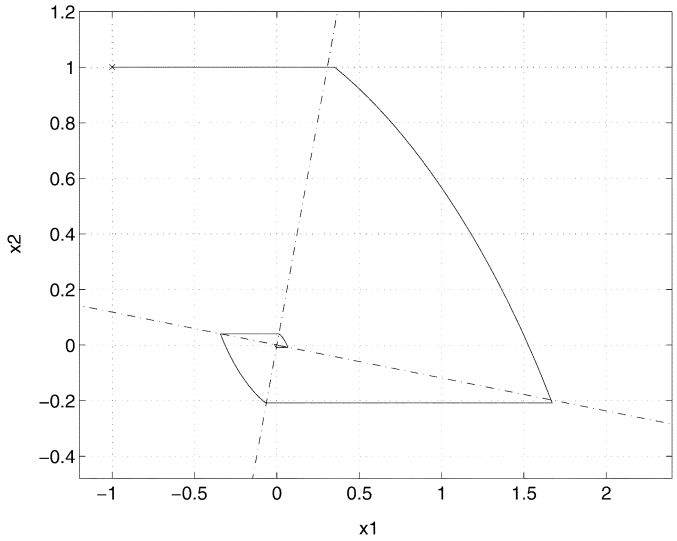


Fig. 2. Trajectory of the two subsystems (35).

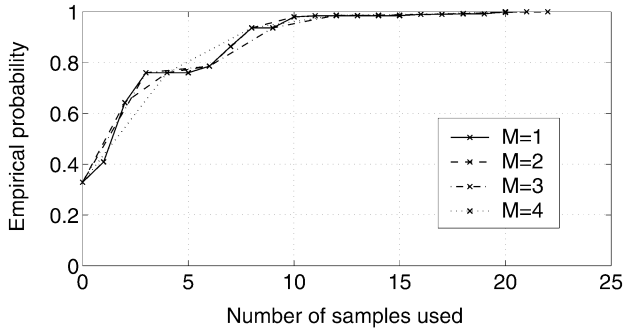


Fig. 3. Empirical probability of (P_1, P_2) to be in $\mathcal{P}_s$ for (35).

There were eleven updates in total. We confirmed that, for this pair, the equivalent bilinear matrix inequality condition holds; thus, it is a deterministic solution of the problem.

We also found that this pair satisfies the technical condition in Assumption 2.2. By checking an equivalent LMI condition, we found that it is in $\widetilde{\mathcal{P}}_{s,\epsilon}$ with $\epsilon = 0.03$. Furthermore, a ball in $\mathcal{C}^2$ centered at this pair with radius 0.0005 seems to be contained in this set $\widetilde{\mathcal{P}}_{s,\epsilon}$; we checked for 20 000 pairs randomly generated in this ball.

The trajectory of the system for the initial condition $[-1\ 1]'$ is plotted in Fig. 2. The previous pair is used in the switching rule (3); it partitions the state–space into four regions, by the two dashed lines in the plot, almost corresponding to the quadrants. In the first and third quadrant, the subsystem 2 is used and in the second and fourth the subsystem 1. The trajectory clearly goes to the origin and there is no chattering in the switching.

The algorithm was run under the same conditions but with $M = 1, 2, 3$. Satisfactory pairs were generated after 20 steps and 12 updates for $M = 1$, 11 steps and 13 updates for $M = 2$, and 7 steps and 12 updates for $M = 3$. We note that in these runs, each step requires M random samples of $x \in \partial\mathcal{B}_0$ and, hence, these runs used 20 samples for $M = 1$, 22 samples for $M = 2$, and so forth. The empirical probability of the (P_1, P_2)-pairs to be in the solution set is given in Fig. 3; here, to make the comparison fair, the horizontal axis is taken to be the number of samples used (and not the number of steps). In this example, the size of M did not have much effect on the convergence rates.

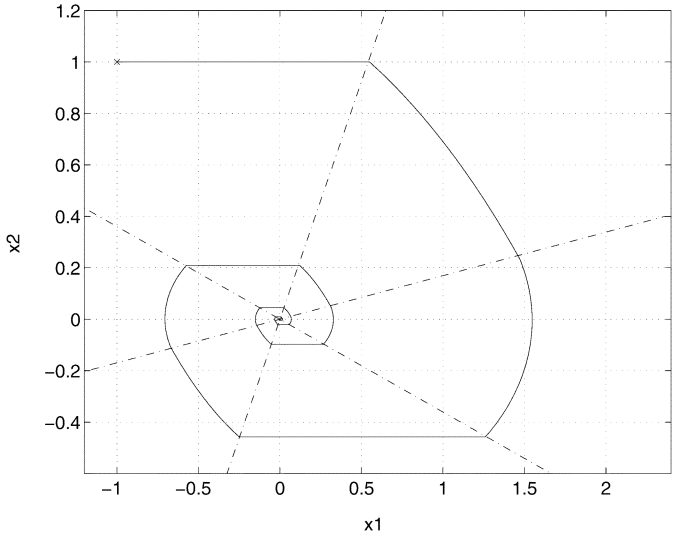


Fig. 4. Trajectory for the three subsystems (37).

We next present an example with three subsystems. In addition to A_1 and A_2 in (35), we use the matrix A_3 given by

$$A_3 = \begin{bmatrix} 0.01 & 4 \\ -1 & 0.1 \end{bmatrix}.$$

There is no Hurwitz convex combination of these three A-matrices. For this case, an obvious solution is to use the pair in (36) and to take P_3 very small so that the system never switches to the subsystem with A_3. A branch-and-bound version of Algorithm 6.2 was run using the parameters $r = 0.3$, the initial triple $(P_{10}, P_{20}, P_{30}) = (3I_2, 2I_2, I_2)$, and $M = 3$; other parameters were the same. After 13 iterations and 17 updates, we arrived at the solution triple

$$(P_1, P_2, P_3) = \left(\begin{bmatrix} 0.115 & -0.290 \\ -0.290 & 2.67 \end{bmatrix}, \begin{bmatrix} 0.448 & 0.583 \\ 0.583 & 1.62 \end{bmatrix}, \begin{bmatrix} 0.570 & 0.200 \\ 0.200 & 1.88 \end{bmatrix} \right). \quad (37)$$

The trajectory is given in Fig. 4. It looks similar to the one in Fig. 2, but the switching signal takes all three states. We note that this solution does not satisfy the BMI condition and hence exhibits its conservatism.

The algorithm works for higher order systems as well. However, the problem often encountered is that chattering can arise easily. Because the algorithm itself does not take account of chattering, the design can be difficult even for a simple example. To illustrate this point, consider the three-dimensional system with

$$A_1 = \begin{bmatrix} 0 & 10 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1.5 & 2 & 0 \\ -2 & -0.5 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}. \quad (38)$$

The matrices A_1 and A_2 have an additional dimension to those in (35) with eigenvalues 0 and -0.5, respectively. The state x_3 corresponding to these eigenvalues is decoupled from the other states and hence is nonincreasing for any trajectories. Therefore,

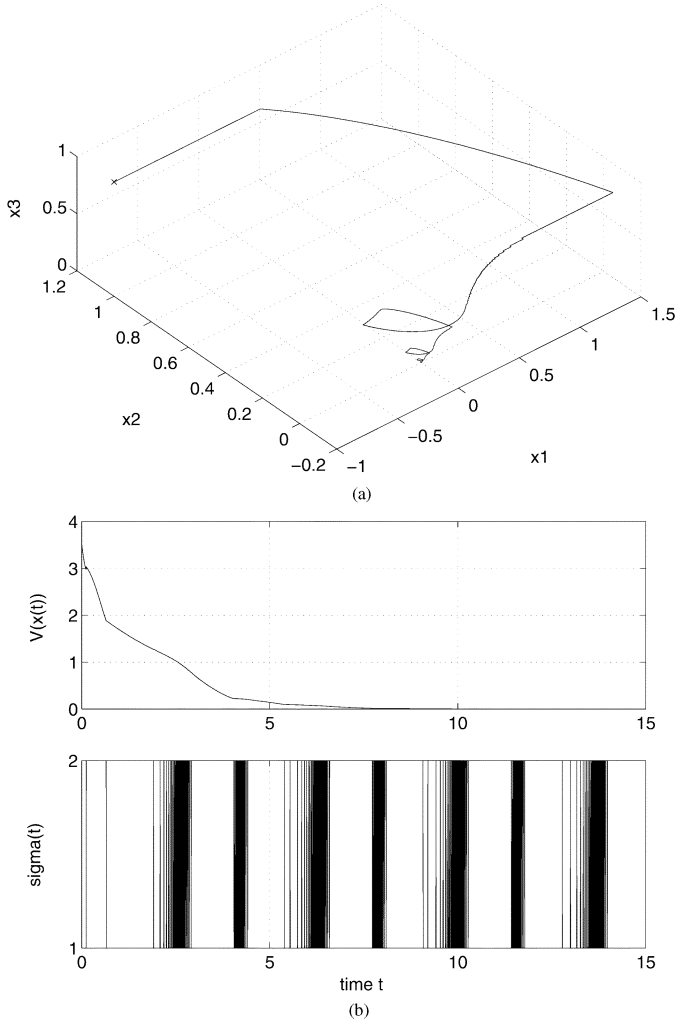


Fig. 5. Results for the system with A-matrices given by (38). (a) A trajectory (b) The Lyapunov function $V(x(t))$ and the switching signal $\sigma(t)$.

we can construct a (P_1, P_2)-pair using the one in (36) to show that the origin of the system is asymptotically stable under the switching law; for example

$$(P_1, P_2) = \left(\begin{bmatrix} 0.0475 & -0.084 & 0 \\ -0.0841 & 1.497 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0.180 & 0.456 & 0 \\ 0.456 & 1.153 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right).$$

Under this law, x_3 does not affect the switching. It is obvious that this pair gives a solution for the problem.

We ran the branch-and-bound algorithm for this system in almost the same way as above with the initial pair $(I_3, 2I_3)$, $N = 2000$, and $M = 4$. The solution we obtained after 6 steps and 13 updates is

$$(P_1, P_2) = \left(\begin{bmatrix} 0.211 & -0.349 & 0.200 \\ -0.349 & 1.71 & -0.105 \\ 0.200 & -0.105 & 1.52 \end{bmatrix}, \begin{bmatrix} 0.293 & 0.413 & 0.132 \\ 0.413 & 1.10 & 0.001\,85 \\ 0.132 & 0.001\,85 & 1.70 \end{bmatrix} \right).$$

A time response for the initial condition $[-1\ 1\ 1]'$ is given in Fig. 5(a). We see that $x(t)$ goes to the origin. Also for this trajectory, the plots for the Lyapunov function $V(x(t))$ and the switching signal $\sigma(t)$ are presented in Fig. 5(b). Note that $V(x(t))$ is decreasing, but there is chattering, or fast switching, in $\sigma(t)$; in this case, the condition (2) does not guarantee stability [17], [24].

IX. Conclusion

In this paper, we presented a probabilistic algorithm that generates a stabilizing switching rule for switched linear systems for the two subsystem case. The algorithm is guaranteed to converge with probability one under a broader notion of convergence compared to those in earlier works on randomized algorithms. We extended this algorithm to multimodal switched systems and showed that this can further be applied to a special class of nonconvex problems including a switched nonlinear systems problem. Future research topics include the study of this class of problems in control.

Acknowledgment

The authors would like to thank Y. Oishi and the anonymous reviewer for their comments which helped them to improve this paper. This research was performed in part while R. Tempo was visiting the Coordinated Science Laboratory at the University of Illinois, Urbana-Champaign, as a CSL Visiting Research Professor. The support of this program is gratefully acknowledged.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*. Reading, MA: Addison-Wesley, 1983.
[2] R. B. Ash, *Real Analysis and Probability*. New York: Academic, 1972.
[3] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, pp. 1249–1274, 2000.
[4] S. Boyd, L. El Ghaoui, E. Ferron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994, vol. 15, SIAM Studies in Applied Mathematics.
[5] M. S. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 475–482, Apr. 1998.
[6] G. Calafiore and B. T. Polyak, "Stochastic algorithms for exact and approximate feasibility of robust LMIs," *IEEE Trans. Autom. Control*, vol. 46, no. 6, pp. 1755–1759, Nov. 2001.
[7] R. DeCarlo, M. S. Branicky, S. Pettersson, and B. Lennartson, "Perspectives and results on the stability and stabilizability of hybrid systems," *Proc. IEEE*, vol. 88, no. 7, pp. 1069–1082, Jul. 2000.
[8] E. Feron, "Quadratic stabilizability of switched systems via state and output feedback," MIT Center for Intelligent Control Systems, Cambridge, MA, Tech. Rep. CICS-P-468, 1996.
[9] Y. Fujisaki, F. Dabbene, and R. Tempo, "Probabilistic design of LPV control systems," *Automatica*, vol. 39, pp. 1323–1337, 2003.
[10] K.-C. Goh, M. G. Safonov, and J. H. Ly, "Robust synthesis via bilinear matrix inequalities," *Int. J. Rob. Nonlinear Control*, vol. 6, pp. 1079–1095, 1996.
[11] H. Ishii, T. Başar, and R. Tempo, "Randomized algorithms for quadratic stability of quantized sampled-data systems," *Automatica*, vol. 40, pp. 839–846, 2004.
[12] M. Johansson and A. Rantzer, "Computation of piecewise quadratic Lyapunov functions for hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 555–559, Apr. 1998.
[13] S. Kanev, B. De Schutter, and M. Verhaegen, "An ellipsoid algorithm for probabilistic robust controller design," *Syst. Control Lett.*, vol. 49, pp. 365–375, 2003.
[14] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer-Verlag, 2003.

[15] D. Liberzon, *Switching in Systems and Control*. Boston, MA: Birkhäuser, 2003.
[16] D. Liberzon and R. Tempo, "Common Lyapunov functions and gradient algorithms," *IEEE Trans. Autom. Control*, vol. 49, no. 6, pp. 990–994, Jun. 2004.
[17] J. Malmborg, B. Bernhardsson, and K. J. Åström, "A stabilizing switching scheme for multi controller systems," in *Proc. 13th IFAC World Congr.*, 1996, pp. 229–234.
[18] Y. Oishi, "Probabilistic design of a robust state-feedback controller based on parameter-dependent Lyapunov functions," in *Proc. 42nd Conf. Decision and Control*, 2003, pp. 1920–1925.
[19] Y. Oishi and H. Kimura, "Computational complexity of randomized algorithms for solving parameter-dependent linear matrix inequalities," *Automatica*, vol. 39, pp. 2149–2156, 2003.
[20] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*. Berlin, Germany: Springer-Verlag, 1987, vol. 268, Lecture Notes on Computer Science.
[21] B. T. Polyak and R. Tempo, "Probabilistic robust design with linear quadratic regulators," *Syst. Control Lett.*, vol. 43, pp. 343–353, 2001.
[22] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. London, U.K.: Springer-Verlag, 2004.
[23] M. Vidyasagar, "Statistical learning theory and randomized algorithms for control," *IEEE Control Syst. Mag.*, vol. 18, no. 6, pp. 69–85, Jun. 1998.
[24] M. Wicks and R. DeCarlo, "Solution of coupled Lyapunov equations for the stabilization of multimodal linear systems," in *Proc. Amer. Control Conf.*, 1997, pp. 1709–1713.
[25] L. Xie, S. Shishkin, and M. Fu, "Piecewise lyapunov functions for robust stability of linear time-varying systems," *Syst. Control Lett.*, vol. 31, pp. 165–171, 1997.
[26] H. Ye, A. N. Michel, and L. Hou, "Stability theory for hybrid dynamical systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 461–474, Apr. 1998.

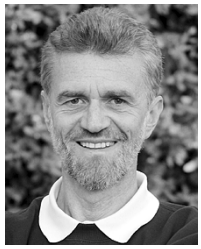
Hideaki Ishii (S'97–M'02) was born in 1972 in Tokyo, Japan. He received the B.Eng. degree in engineering systems from the University of Tsukuba, Tsukuba, Japan, in 1996, the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1998, and the Ph.D. degree in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2002.

He was a Postdoctoral Research Associate in the Coordinate Science Laboratory at the University of Illinois at Urbana-Champaign from 2001 to 2004. He is currently a Research Associate in the Department of Information Physics and Computing, The University of Tokyo, Tokyo, Japan. His research interests are in networked control systems, probabilistic algorithms, and hybrid systems.

Tamer Başar (S'71–M'73–SM'79–F'83) received the B.S.E.E. degree from Robert College, Istanbul, Turkey, and the M.S., M.Phil, and Ph.D. degrees in engineering and applied science from Yale University, New Haven, CT.

After stints at Harvard University, Cambridge, MA, and Marmara Research Institute, Gebze, Turkey, he joined the University of Illinois at Urbana-Champaign in 1981, where he is currently the Fredric G. and Elizabeth H. Nearing Professor of Electrical and Computer Engineering. He has published extensively in systems, control, communications, and dynamic games, and has current interests in robust nonlinear and adaptive control, modeling and control of communication networks, control over wireless links, resource management and pricing in networks, risk-sensitive estimation and control, and robust identification.

Dr. Başar is currently the Editor-in-Chief of *Automatica*, Editor of the Birkhäuser Series on Systems and Control, Managing Editor of the *Annals of the International Society of Dynamic Games (ISDG)*, and a Member of editorial and advisory boards of several international journals. He has received several awards and recognitions over the years, among which are the Medal of Science of Turkey (1993), Distinguished Member Award (1993), Axelby Outstanding Paper Award (1995) and Bode Lecture Prize (2004) of the IEEE Control Systems Society (CSS), and the Quazza Medal (2005) of the International Federation of Automatic Control (IFAC). He is a Member of the National Academy of Engineering, a Member of the European Academy of Sciences, a past President of CSS, and a past President of ISDG.



Roberto Tempo (M'90–SM'98–F'00) was born in Cuorgné, Italy, in 1956. He graduated in electrical engineering from Politecnico di Torino, Torino, Italy, in 1980.

From 1981 to 1983, he was with the Dipartimento di Automatica e Informatica, Politecnico di Torino. In 1984, he joined the National Research Council of Italy (CNR) at the research institute IEIIT, Torino, where he is a Director of Research of Systems and Computer Engineering since 1991. He has held visiting and research positions at Kyoto University, Kyoto, Japan, the University of Illinois at Urbana-Champaign, German Aerospace Research Organization, Oberpfaffenhofen, Germany, and Columbia University, New York. His research activities are mainly focused on robustness analysis and control of uncertain systems, and identification of complex systems in the presence of bounded errors. He is author or coauthor of more than 120 research papers published in international journals, books, and conferences. He is also a coauthor of the book *Randomized Algorithms for Analysis and Control of Uncertain Systems* (London, U.K.: Springer-Verlag, 2005).

In 1993, he received the "Outstanding Paper Prize Award" from the International Federation of Automatic Control (IFAC) for a paper published in the journal *Automatica*. Dr. Tempo has been an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, for which he is now Editor, Technical Notes and Correspondence, and *Systems and Control Letters*. Currently, he is an Editor and Deputy Editor-in-Chief of *Automatica*. He is Program Chair for the first joint CDC-ECC Conference, to be held in Seville, Spain, in 2005. He was Vice-President for Conference Activities of the IEEE Control Systems Society for the period 2002–2003 and a Member of the EUCA Council in 1998–2003.