Formal Synthesis of Stabilizing Controllers for Periodically Controlled Linear Switched Systems

Atreyee Kundu [a] Miriam García Soto [b]
Pavithra Prabhakar [c]

Abstract

In this paper, we address the problem of synthesizing periodic switching controllers for stabilizing a family of linear systems. Our broad approach consists of constructing a finite game graph based on the family of linear systems such that every winning strategy on the game graph corresponds to a stabilizing switching controller for the family of linear systems. The construction of a (finite) game graph, the synthesis of a winning strategy and the extraction of a stabilizing controller are all computationally feasible. We illustrate our method on an example.

1 Introduction

Stability is a fundamental property in control system design that stipulates that small perturbations in the initial state or input to the system lead to small deviations in the resulting behavior of the system, and that the effect of small perturbations vanishes over time. Supervisory control consists of switching between a set of operational modes/dynamics to achieve a control objective. In this paper, we study the problem of synthesizing a switching control to achieve stability. More precisely, given a finite set of linear dynamical systems $\dot{x} = A_p x$, where $p \in \mathcal{P}$ and $\mathcal{P}$ is a finite set of indices corresponding to operational modes, we intend to compute a switching strategy which specifies for each time instant a mode p whose dynamics is active, such that the resulting switched system is stable. In particular, we consider periodically controlled systems where the

[a]Atreyee Kundu is an INSPIRE faculty fellow with the Department of Electrical Engineering, Indian Institute of Science Bangalore, India. Her research was supported in part by DST INSPIRE Faculty Award IFA17-ENG225 of the Department of Science and Technology, Govt. of India.

[b]Miriam García Soto is a postdoctoral scholar with the Institute of Science and Technology Austria. She was partially supported by the Austrian Science Fund (FWF) under grants S11402-N23 (RiSE/SHiNE) and Z211-N23 (Wittgenstein Award).

[c]Pavithra Prabhakar is an associate professor of Computer Science and Peggy and Gary Edwards Chair in Engineering at Kansas State University. She was partially supported for this work by a National Science Foundation CAREER Award No. 1552668 and an Office of Naval Research Young Investigator Award No. N00014-17-1-257.

change to a new dynamics occurs only at time instants that are multiples of τ, that is, at time $\tau, 2\tau, 3\tau, \cdots$. Periodically controlled switched systems model real time behaviors of supervisory controller when implemented on a digital platform, wherein, the sensing of the plant state and application of mode switch happens synchronously with the processor clock periodically.

Switched systems [12] have been extensively investigated in the context of stability. Stability analysis for switched systems is challenging even when the dynamics of the modes is linear. For instance, it is shown in [2] that switching between two stable systems can result in an unstable system. Hence, it is essential to design the switching logic carefully so as to ensure stability. From a computational point of view, it was established that analyzing stability is undecidable for switched and hybrid systems even when the dynamics is a constant rate dynamics [24]. Research on stability analysis for switched linear systems explores restrictions on the dynamics and switching law such as requiring the system matrices to be pairwise commutative or symmetric, for enforcing stability [16, 29]. More generally, Lyapunov function based approaches have been explored for stability analysis and stabilizability of both linear and non-linear switched systems, and consist broadly of common and multiple Lyapunov function paradigms. Necessary and sufficient conditions for the existence of common and multiple Lyapunov functions [25, 26, 1] as well as constraints on the switching, such as, slow switching characterized by (average) dwell time [15, 7], and asymptotic characterizations [9, 10] that ensure stability have been investigated. For constrained switching, multiple Lyapunov functions have been studied in [3] including piecewise quadratic Lyapunov functions [8]. For the stabilization problem, Control Lyapunov functions, namely, quadratic and piecewise quadratic, have been explored in [28, 27, 18]. Necessary and sufficient conditions for stabilization have been explored in [13]. See [14] for a recent survey on stability and stabilization of switched linear systems. Synthesis of stabilizing switching signals for family of unstable linear systems appeared earlier in the literature, see e.g., [4, 6]. A *min-switching signal* [6] ensures global asymptotic stability of a linear switched system $\mathcal{S}$ given that the subsystem matrices satisfy a set of Bilinear Matrix Inequalities (BMIs).

In this paper, we propose an alternate approach based on abstractions for synthesizing a stabilizing controller, that requires solving linear programming problems and uses insights from automata and game theory [5]. Our broad approach for synthesizing a periodic switching control is an abstraction based approach which consists of constructing a finite weighted game graph that represents an abstraction (simplification) of the original system, and solving the controller synthesis problem on the simplified system. More precisely, we construct a finite game graph such that every control strategy on the finite game graph which satisfies certain conditions (every cycle in the resulting graph has product of the weights on the edges at most 1) corresponds to an actual periodic stabilizing switching control in the original system. However, the finite game graph is conservative in that it might have fewer strategies than that in the original system.

The game graph is constructed based on a finite partition of the state-space.

Each region r in the partition corresponds to a player 1 node in the graph, and has an edge to a player 2 node (r, p) for every index $p \in \mathcal{P}$ of the dynamics. Player 1 chooses which of the dynamics to follow from a given region r, which is performed by following the corresponding edge. That is, choosing dynamics p from region r corresponds to following the edge r to (r, p). Player 2 follows the dynamics chosen for a time period τ. Hence, from (r, p), there is an edge to every r' such that from r following the dynamics p for time τ results in region r'. Note that Player 2 does not choose an edge, but needs to ensure stability for each of the edges. The strategy corresponds to choosing an edge from each Player 1 node such that all the paths in the graph obtained by choosing any of the Player 2 edges result in a stabilizing control. This is achieved by adding a weight to the edges from (r, p) to r' which provide a bound the factor by which the state of the system moves away from the origin (the equilibrium point) when moving from r to r' using p for time τ. A stabilizing controller then corresponds to a strategy in which the resulting graph does not have a cycle in which the product of the weights on the edges is > 1.

Our approach follows the broad approach for stabilizing controller synthesis in [23], which builds on the abstraction based stability analysis in [21, 20, 17, 19]. However, in comparison to [23], the current paper differs in several aspects. Firstly, here we consider the problem of periodic switching controllers, whereas, in [23], a state (facet) based switching controller is sought. Hence, in the game graph constructed, the nodes in [23] corresponds to facets specified along with the problem, whereas in this paper, the nodes correspond to regions, that is provided by the user or automatically determined using approaches such as counter-example guided abstraction refinement [17]. Further, here, the graph construction is simpler as compared to that from [23] in that certain auxiliary nodes are not required. Finally, the computation of the edges is simpler, since, we need to only check if it is possible to reach a target region from a source region at a specific time τ, rather than in an unbounded interval $[0, \infty)$ as in [23], where hybridization is applied prior to abstract game graph construction to tackle the issue. Further details regarding the computational aspects are discussed in Section 5.1.

2 Problem statement

In this section, we introduce the controller synthesis problem which consists of synthesizing a periodic switching sequence for a given family of linear dynamical systems such that the executions resulting from the application of the switching sequence on the family correspond to a stable system. We will define the concepts required to formalize the controller synthesis problem. Let $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{N}$ denote the set of real, rational and natural numbers, respectively.

2.0.1 Family of linear systems

Consider a family of continuous-time linear systems

$$\dot{x}(t) = A_p x(t),\ \ x(0) \in X_0,\ \ p \in \mathcal{P},\ \ t \in \mathbb{R}_{\geq 0}, \tag{1}$$

where $x(t) \in \mathbb{R}^d$ is the vector representing the state of the system at time t, and $\mathcal{P} = \{1, 2, \ldots, N\}$ is an index set where a $p \in \mathcal{P}$ is used to refer to the p-th linear dynamical system, and X_0 is a set of initial states. We also refer to the index p as the (operational) mode of the family of systems. We will use $\mathcal{F} = (\{A_p\}_{p\in\mathcal{P}}, X_0)$ to denote the family of linear systems (1). We assume that A_p, $p \in \mathcal{P}$ are full-rank matrices. Consequently, $0 \in \mathbb{R}^d$ is the unique equilibrium point for each system in $\mathcal{F}$.

2.0.2 Switching signal

Next, we define a switching signal which selects an *active subsystem* at every instant of time, i.e., the system from $\mathcal{F}$ that is currently being followed [11, $\mathcal{S}$1.1.2]. Formally, a *switching signal* σ is a function from $\mathbb{R}_{\geq 0}$ to $\mathcal{P}$ such that there exist a diverging sequence of times $0 =: \tau_0 < \tau_1 < \tau_2 < \cdots$ and a sequence of indices $p_0, p_1, p_2, \cdots$ with $p_i \in \mathcal{P}$, $i = 0, 1, 2, \ldots$ such that $\sigma(t) = p_i$ for $t \in [\tau_i, \tau_{i+1}[$, $i = 0, 1, 2, \ldots$. We call the time instants $\tau_0, \tau_1, \tau_2, \ldots$ a sequence of *switching instants* for σ. Note that the sequence of switching instants is not unique and only represents time instances such that the system remains in the same mode between two consecutive time instances. Moreover, a switching signal can be completely specified by providing its value at the switching instants.

In this paper, we are interested in switching signals where mode changes are allowed to only happen at periodic time instants. A switching signal σ is said to satisfy a *fixed dwell time* τ if there exists a sequence of switching time instants $\tau_0, \tau_1, \cdots$ for σ such that $\tau_{i+1} - \tau_i = \tau$, $i = 0, 1, 2, \ldots$. Given $\tau > 0$, let Σ_τ denote the set of all switching signals σ that satisfy a fixed dwell time τ.

2.0.3 Switched system

We refer to the system resulting from the application of a switching signal on a family of dynamical systems as a switching system. A *switched system* generated by the family of systems $\mathcal{F}$ and a family of switching signals $\Sigma = \{\sigma^{x_0}\}_{x_0 \in X_0}$ is given by

$$\dot{x}(t) = A_{\sigma^{x_0}(t)} x(t),\ \ x(0) = x_0 \in X_0,\ \ t \in \mathbb{R}_{\geq 0}. \tag{2}$$

Henceforth, we refer to the linear switched system (2) by $\mathcal{S} = (\mathcal{F}, \Sigma)$. A solution of $\mathcal{S}$ from an initial state $x_0 \in X_0$ is a map $\Phi_{\mathcal{S}}^{x_0} : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ defined inductively as follows: Let $\tau_0, \tau_1, \cdots$ be a sequence of switching instances for σ^{x_0}. Then, for $i = 0, 1, 2, \cdots$ and $t \in [\tau_i, \tau_{i+1}[$,

$$\Phi_{\mathcal{S}}^{x_0}(t) = \exp(A_{\sigma^{x_0}(\tau_i)}(t - \tau_i))\Phi_{\mathcal{S}}^{x_0}(\tau_i),$$

where $\exp(A)$ represents the matrix exponential of A, namely, e^A.

2.0.4 Stability

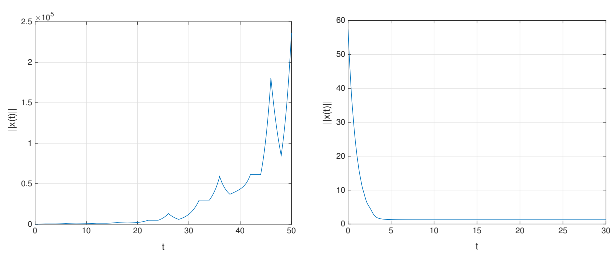
In this paper, we are interested in stability of $\mathcal{S}$ under switching signals σ that obey a fixed dwell time between every two consecutive switching instants. A switched system $\mathcal{S} = (\mathcal{F}, \Sigma)$ is said to be *Lyapunov stable* if for every $\varepsilon > 0$ there exists a $\delta = \delta(\varepsilon) > 0$ such that $\|\Phi_{\mathcal{S}}^{x_0}(0)\| \leq \delta \Rightarrow \|\Phi_{\mathcal{S}}^{x_0}(t)\| \leq \epsilon$ for all $x_0 \in X_0$ and $t \in \mathbb{R}_{\geq 0}$.

At this point, it is important to highlight that even when all systems in $\mathcal{F}$ are Lyapunov stable, it may be possible to obtain an unstable switched system $\mathcal{S}$ under a switching signal satisfying a fixed dwell time. We discuss the following example to demonstrate this matter:

Example 1 *Consider a family of systems $\mathcal{F}$ with*

$$A_1 = \begin{pmatrix} 0 & 1 \\ -0.1 & 0 \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} 0 & 1 \\ -4 & 0 \end{pmatrix}.$$

Fix a dwell time $\tau = 2$ units of time. Notice that both A_1 and A_2 are Lyapunov stable. Let a switching signal σ' alternate between modes 2 and 1, while being in each mode for τ units of time, that is, $0, 2, 4, 6, \cdots$ is a sequence of switching instances of σ', where $\sigma(2i) = 2$ if i is even and 1 otherwise. The linear switched system $\mathcal{S}$ generated under σ' with $x(0) = \begin{pmatrix} 86.3041 & 76.1538 \end{pmatrix}^\top$ is not Lyapunov stable. The state response $(\|x(t)\|)_{t \in \mathbb{R}_{\geq 0}}$ is shown in Figure 1a.



(a) Plot for Example 1 (b) Plot for Example 2

Figure 1: Plot of $(\|x(t)\|)_{t \in \mathbb{R}_{\geq 0}}$

Another case is that all systems in $\mathcal{F}$ are unstable. Then, it may be possible to obtain a stable switched system under a switching signal satisfying a fixed dwell time and selecting an active subsystem in $\mathcal{F}$, as in the following example.

Example 2 *Consider a family of systems $\mathcal{F}$ with*

$$A_1 = \begin{pmatrix} 0 & 1 \\ -0.1 & 0.2 \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} 0 & 1 \\ -4 & 0.2 \end{pmatrix}.$$

Fix a dwell time $\tau = 2$ units of time. Notice that both A_1 and A_2 are unstable. Let a switching signal σ' alternate between modes 2 and 1, while being in each mode for τ units of time, that is, $0, 2, 4, 6, \cdots$ is a sequence of switching instances *of σ', where $\sigma(2i) = 2$ if i is even and 1 otherwise. The linear switched system $\mathcal{S}$ generated under σ' with $x(0) = \begin{pmatrix}-43.6660 & 37.0513\end{pmatrix}^\top$ is Lyapunov stable. The state response $(\|x(t)\|)_{t\in\mathbb{R}_{\geq 0}}$ is shown in Figure 1b.*

2.0.5 Periodic Stabilizing Switching Controller Synthesis Problem

Our broad objective in this paper is to synthesize a family of switching controllers one for each initial state such that the resultant switched system is stable. This is formalized in the following:

Problem 1 *Given a family of linear systems $\mathcal{F} = (\{A_p\}_{p\in\mathcal{P}}, X_0)$ and a scalar $\tau > 0$, find a family of switching signals $\Sigma = \{\sigma^{x_0}\}_{x_0\in X_0} \subseteq \Sigma_\tau$ under which the linear switched system $\mathcal{S} = (\mathcal{F}, \Sigma)$ is Lyapunov stable.*

Though Example 1 admits a trivial solution for Problem 1, namely, selecting only one of the systems at any time, Example 2 does not. In this paper, we propose an abstraction based approach and employ game theoretic arguments for the synthesis of stabilizing controllers. We will, henceforth, call a Σ stabilizing if it is a solution to Problem 1.

3 Main approach

We propose an abstraction based approach to solve Problem 1. Our broad approach to the problem follows that of [23], however, our focus in this paper is on periodically controller systems which simplifies some of the technicalities of [23]. We describe the main steps in the synthesis below:

1. First, we construct a simplified "game graph" for a given family of linear systems such that a "winning" strategy in the game graph corresponds a stabilizing family of switching signals.

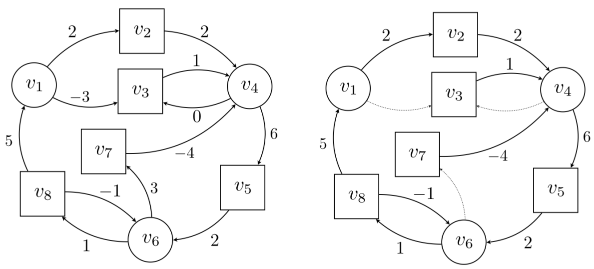
2. We solve for a winning strategy in the game graph.

3. We extract the family of switching signals from the winning strategy.

First, we provide some preliminaries on game graphs and winning strategies in Section 4. Solving for winning strategies is already studied in [23], and we refer to the same. Then we present the construction of the game graph, and extraction of the stabilizing controller from a winning strategy for the game graph in Section 5.

4 Games

Problem 1 will be reduced to a problem on a game graph, and the solution to this game problem will provide the solution to the initial one. The solution to Problem 1 consists of a switching signal for every initial state, with fixed dwell time, which renders the linear switched system stable. The search for such a



(a) Game graph G (b) Strategy over G

Figure 2: Game graph and strategy

kind of signal is transformed into the search for a strategy in a finite weighted game graph. In this section, we introduce certain notions in game theory which are required for our approach.

A game is played between two players, a defender and an adversary. The defender strives to achieve some objective, while the adversary tries to thwart the efforts of the defender. More precisely, the game is represented by a game graph, which is essentially a graph with two kinds of nodes corresponding to the defender and the adversary and the edges going between nodes of different kind. A play starts in some node, and consists of the corresponding player choosing an edge out of the node. Then, the edge is taken upon which a node of the opposite player is reached, followed by the opposite player choosing an edge out of the reached node. Hence, the two players alternately choose an edge and traverse the same. The result play, a sequence of nodes, is winning if it satisfies certain objective. A winning strategy consists of the choices of defender for each possible finite play such that irrespective of the choices of the adversary, the resulting play satisfies the given objective.

Game Graph A *game graph* is a weighted graph $G = (V, E, W)$ where $V = V_0 \cup V_1$, $V_0 \cap V_1 = \emptyset$, E is a subset of $V_0 \times V_1 \cup V_1 \times V_0$ and W is a weight function mapping edges to rational numbers, $W : E \rightarrow \mathbb{Q}$. Nodes are partitioned between the player and the adversary. The nodes in V_0 are those where the defender makes a choice, while the nodes in V_1 are those where the adversary makes a choice. Figure 2a shows a game graph, where the nodes at which defender makes a decision are represented by circles and those at which the adversary makes a decision are represented by squares.

Plays A *play* in a game graph G is an infinite sequence of nodes $\pi = v_1 v_2 \ldots$ such that $(v_i, v_{i+1}) \in E$ for every $i \geqslant 1$. The set of all plays in G is denoted as $\mathit{Plays}(G)$. The weight of a play $\pi = v_1 v_2 \ldots$ is bounded by b, denoted $W(\pi) \leqslant b$, if $\prod_{i=1}^{j} W(v_i, v_{i+1}) \leqslant b$ for every j. Note that the weight of a play is bounded by the product of the weights on the edges of prefixes of the play.

Strategy A strategy specifies the choices of the defender during a play. A *strategy* for a game graph $G = (V, E)$ is a function $\alpha : V_0 \to V_1$ such that for every $v \in V_0$, $(v, \alpha(v)) \in E$. In general, a strategy specifies the defender's choice from a defender node given the *history* of the play so far. However, we consider memoryless strategies in this paper, where the defender chooses the same edge from a defender node, irrespective of how the node was reached. A play $\pi = v_0 v_1 \dots$ of G is said to be *consistent with a strategy* α if for all $v_i \in V_0$, $\alpha(v_i) = v_{i+1}$.

In Figure 2b, the solid outgoing arrows from the circles show a memoryless strategy for the defender in the game graph, which specifies the choices of the defender when in those nodes. The strategy induces a subgraph, defined by all the solid arrows.

Winning Strategy We consider a bounded objective for the defender, which requires to ensure that the weight of a play consistent with the strategy is bounded from above. This boundedness is equivalent to ensuring that every cycle in the game graph has a product of weights equal to or smaller than one. Next, we define the winning strategy with respect to this objective. A strategy α is a *winning strategy* for a game graph G if there exists a value $M \in \mathbb{Z}$ such that for every $\pi \in \mathit{Plays}(G)$ consistent with α, the weight of π is bounded by M.

If the game graph G is finite, that is, the number of nodes V is finite, then the winning strategy can be effectively computed. It can be reduced to another game called the energy game, that has been studied in the literature; details about computing the winning strategy for the game graph with respect to a bounded objective can be found in [23].

5 Game Graph Construction

In this section, we explain the main result of the paper. We describe the procedure to construct a game graph for solving Problem 1, and how to extract a stabilizing controller from a winning strategy for the game graph. Our game graph construction is parameterized by a family of linear systems $\mathcal{F}$ and a periodic time step τ, as well as a partition of the state-space into a finite number of regions. The game graph captures the evolution of the different dynamics in all the regions. More precisely, the defender nodes correspond to the regions in the partition. Intuitively, we intend to choose a dynamics to be followed in each of the regions. Hence, an edge from a defender node r, representing a region, to an adversary node (p, r) corresponds to the choice of the p-th mode (dynamics) in region r. Executing the dynamics $\dot{x} = A_p x$ while in region r for time τ, results in the system evolving to some region r'. Note that the region r' reached depends on the initial state in r from which we start executing. Hence, there might be multiple regions reached depending on where the systems starts in r. We add an edge from (p, r) to each of those regions r', along with a weight w that provides an upper bound on the (scaling) factor by which the state moves away from the equilibrium when going from r to r'. In the constructed game graph, we need to find a strategy for the defender (choice of dynamics to be followed in each region) such that the resulting system is stable. This correspond to ensuring that along any play that conforms with the strategy the weight is bounded. Hence, we reduce the problem of finding a stabilizing controller to that of finding a winning strategy in the game graph. Below we provide the formal details.

5.1 Construction of the Game Graph

A *partition* of $\mathbb{R}^d$ is a set of regions $\{r_1, \cdots, r_m\}$, where for each i, $r_i \subseteq \mathbb{R}^d$, for each $i \neq j$, $\mathring{r}_i \cap \mathring{r}_j = \emptyset$ and $\bigcup_i r_i = \mathbb{R}^d$. Let $\mathcal{F} = (\{A_p\}_{p\in\mathcal{P}}, X_0)$ be a family of linear systems of dimension d. A *game graph* induced by a family of systems $\mathcal{F} = (\{A_p\}_{p\in\mathcal{P}}, X_0)$, a partition $\mathcal{R}$ of $\mathbb{R}^d$, and a time step τ, denoted as $G(\mathcal{F}, \mathcal{R}, \tau) = (V, E, W)$, is given by

- the set of nodes $V = V_0 \cup V_1$, where
  - $V_0 = \mathcal{R}$,
  - $V_1 = \mathcal{P} \times \mathcal{R}$,
- set of edges $E = E_0 \cup E_1$, where
  - $E_0 = \{(r, (p, r)) \mid r \in \mathcal{R},\ p \in \mathcal{P}\}$,
  - $E_1 = \{((p, r), r') \mid \exists x \in r, y \in r', y = \exp(A_p\tau)x\}$,
- edge weights $w : E \to \mathcal{R}$ defined as

$$w(e) = \begin{cases} 1, & \text{if } e \in E_0, \\ \mu((p, r), r'), & \text{if } e \in E_1, \end{cases}$$

$$\mu((p, r), r') = \sup\{\frac{\|y\|}{\|x\|} \mid x \in r,\ y \in r',\ y = \exp(A_p\tau)x\}.$$

Note that computing the edges in E_1 requires solving a linear programming problem. First observe that the constraints $x \in r$ and $y \in r'$ both correspond to linear constraints with variables x and y given the regions r and r'. Similarly, since $\exp(A_p\tau)$ is a matrix, $y = \exp(A_p\tau)x$ is a linear constraint as well. Note that if we did not have a fixed time τ but an interval I in which τ resided, then, to compute E_1, we would need to over-approximate the relation between x and y given by $y = \exp(A_p\tau)x$, $\tau \in I$, since, the relation between x and y is given by an exponential function, thus obtaining a conservative E_1. Computing precise over-approximations of the relation $y = \exp(A_p\tau)x$, $\tau \in I$, is especially challenging when I is unbounded. Hence, in [22], hybridization is applied to over-approximate the dynamics $\dot{x} = A_px$ to a set of polyhedral inclusion dynamics $\dot{x} \in P$, $x \in X$ where P is a polyhedron and X is a polyhedral region. Now the solutions do not contain any exponential functions, and thus, the relation between x and y can be captured using linear constraints. Hence, the computation of the edges is a simpler affair in our current setting of periodic controllers. Similarly, the computation of the weights $\mu((p, r), r')$ requires solving an optimization problem over linear constraints, where the objective function is $\|y\| / \|x\|$. It is shown in [22] that the objective function can be simplified to $\|y\|$ by adding the constraint $\|x\| = 1$, when the regions are conical (that is, contain the origin and are positive scaling closed). The objective function $\|y\|$ considered as the infinity norm can be further reduced to solving a finite number of linear programming problems, by considering each component of y and its negation as objective functions in turn, and taking the maximum of the solutions of the linear programming problems [20].

5.2 Extraction of switching signal from a strategy

Next, we present how to obtain a switching signal for $\mathcal{F}$ and a given initial state x_0 from a strategy α on the game graph $G(\mathcal{F}, \mathcal{R}, \tau)$. The broad idea is to follow the dynamics p_0 specified by α for the region r_0 to which x_0 belongs, for the first τ time units, then follow the dynamics specified by α for the region r_1 that is reached by following p_0 for τ time units from x_0, and so on. Recall that to specify a switching signal, it suffices to specify its values at the switching instants. We refer to $\mathcal{R}(x)$ as the region in $\mathcal{R}$ to which x belongs. Let α be a strategy for $G(\mathcal{F}, \mathcal{R}, \tau)$, where $\mathcal{F} = (\{A_p\}_{p \in \mathcal{P}}, X_0)$. Given an initial state $x_0 \in X_0$, we define a switching signal $\sigma_\alpha^{x_0}$, where $0, \tau, 2\tau, \cdots$ is a sequence of switching instants, inductively at every $i\tau$, while also simultaneously defining the state x_i reached by following the switching signal at time $i\tau$. For $i = 0, 1, \cdots$,

- $\sigma_\alpha^{x_0}(i\tau) = p_i$ if $\alpha(\mathcal{R}(x_i)) = (p_i, \mathcal{R}(x_i))$; and
- $x_{i+1} = e^{A_{p_i}\tau} x_i$.

5.3 Main result

Our main result states that a stabilizing switching controller for a family of linear systems can be extracted from a winning strategy on the induced game graph.

Theorem 1 *Given a family of linear system $\mathcal{F}$, a partition $\mathcal{R}$ of $\mathbb{R}^d$, and a time τ, if α is a winning strategy for $G(\mathcal{F}, \mathcal{R}, \tau)$, then $\mathcal{S} = (\mathcal{F}, \Sigma_\alpha)$, where $\Sigma_\alpha = \{\sigma_\alpha^{x_0}\}_{x_0 \in X_0}$, is stable.*

Hence, Σ_α which consists of switching signals extracted from the winning strategy α is a family of stabilizing controllers for $\mathcal{F}$.

Theorem 1 states that our switching controller synthesis approach is sound in that if there is a winning strategy, then the extracted switched system is Lyapunov stable. Here, completeness results are not provided. The proof is similar to that in [23]. It relies on the fact that the scalings along solutions of the switched system $\mathcal{S}$ are upper-bounded by the product of weights along corresponding prefixes of the paths in the graph obtained by restricting the game graph to the choices specified by the winning strategy α. The weight along the paths are bounded, since the strategy is winning, and hence, the solutions do not diverge.

6 Numerical example

In this section we present numerical examples to demonstrate our result.

6.0.1 Analysis of Example 1

Consider the family of linear systems $\mathcal{F}$ from Example 1. We consider the state-space $\mathbb{R}^2$ partitioned into four regions $\mathcal{R} = \{r_1, r_2, r_3, r_4\}$ corresponding to the four quadrants $r_1 = \{(x_1, x_2) \,|\, x_1 \geqslant 0,\, x_2 \geqslant 0\}$, $r_2 = \{(x_1, x_2) \,|\, x_1 \leqslant 0,\, x_2 \geqslant 0\}$, $r_3 = \{(x_1, x_2) \,|\, x_1 \leqslant 0,\, x_2 \leqslant 0\}$, $r_4 = \{(x_1, x_2) \,|\, x_1 \geqslant 0,\, x_2 \leqslant 0\}$. We will employ our technique to find a stabilizing switching signal.

The first task is to construct a game graph $G(\mathcal{F}, \mathcal{R}, \tau) = (V, E, W)$ with $V = V_0 \cup V_1$, $E = E_0 \cup E_1$. We have

- $V_0 = \{r_1, r_2, r_3, r_4\}$,
- $V_1 = \{(1, r_1), (2, r_1), (1, r_2), (2, r_2), (1, r_3), (2, r_3), (1, r_4), (2, r_4)\}$,
- $E_0 = \{(r_1, (1, r_1)), (r_1, (2, r_1)), (r_2, (1, r_2)), (r_2, (2, r_2)), (r_3, (1, r_3)), (r_3, (2, r_3)), (r_4, (1, r_4)), (r_4, (2, r_4))\}$,
- $E_1 = \{((1, r_2), r_1), ((1, r_2), r_3), ((1, r_2), r_4), ((2, r_2), r_1), ((2, r_2), r_3), ((2, r_2), r_4), ((1, r_4), r_1), ((1, r_4), r_2), ((1, r_4), r_3), ((2, r_4), r_1), ((2, r_4), r_2), ((2, r_4), r_3)\}$,
- $w((1, r_2), r_1) = 7.46$, $w((1, r_2), r_3) = 0.68$,
  $w((1, r_2), r_4) = 3.91$, $w((2, r_2), r_1) = 7.45$,
  $w((2, r_2), r_3) = 0.10$, $w((2, r_2), r_4) = 0.01$,
  $w((1, r_4), r_1) = 1.29$, $w((1, r_4), r_2) = 0.68$,
  $w((1, r_4), r_3) = 3.91$, $w((2, r_4), r_1) = 1$,
  $w((2, r_4), r_2) = 0.01$, $w((2, r_4), r_3) = 0.13$.

Recall the definition of $w(e)$, $e \in E$ from Section 5. The scalars $\mu((p, r), r')$'s are computed by solving a linear program with $\|x\| = 1$. Observe that considering x such that $\|x\| = 1$ suffices because we are dealing with linear dynamics, which means that if an execution from x reaches y, then an execution from mx reaches my for $m > 0$. The game graph $G(\mathcal{P}, \mathcal{R})$ is shown in Figure 3. We then obtain the following stabilizing switching signal α for $G(\mathcal{F}, \mathcal{R}, \tau)$: $\alpha(r_1) = (2, r_1)$, $\alpha(r_2) = (2, r_2)$, $\alpha(r_3) = (2, r_3)$ and $\alpha(r_4) = (2, r_4)$.

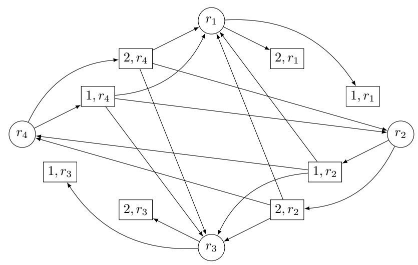


Figure 3: Game graph $G(\mathcal{F}, \mathcal{R}, \tau)$ for Example 1

6.0.2 Analysis of Example 2

We consider the family of linear unstable systems $\mathcal{F}$ from Example 2 and provide a stabilizing switching signal. Let the state-space $\mathbb{R}^2$ be partitioned into four regions $\mathcal{R} = \{r_1, r_2, r_3, r_4\}$ that correspond to the four quadrants as before. Again, we choose a fixed dwell time $\tau = 2$ units of time. We are interested in synthesizing a switching signal $\sigma \in \Sigma_\tau$ under which the switched system $\mathcal{S}$ is Lyapunov stable.

For solving the said problem, we first construct a game graph $G(\mathcal{F}, \mathcal{R}, \tau) = (V, E, W)$ with $V = V_0 \cup V_1$, $E = E_0 \cup E_1$, where

- $V_0 = \{r_1, r_2, r_3, r_4\}$,
- $V_1 = \{(1, r_1), (2, r_1), (1, r_2), (2, r_2), (1, r_3), (2, r_3),$
  $(1, r_4), (2, r_4)\}$,
- $E_0 = \{(r_1, (1, r_1)), (r_1, (2, r_1)), (r_2, (1, r_2)), (r_2, (2, r_2)),$
  $(r_3, (1, r_3)), (r_3, (2, r_3)), (r_4, (1, r_4)), (r_4, (2, r_4))\}$,
- $E_1 = \{((1, r_2), r_1), ((1, r_2), r_3), ((1, r_2), r_4), ((2, r_2), r_1),$
  $((2, r_2), r_3), ((1, r_4), r_1), ((1, r_4), r_2),$
  $((1, r_4), r_3), ((2, r_4), r_1), ((2, r_4), r_3)\}$,
- $w((1, r_2), r_1) = 7.54, \quad w((1, r_2), r_3) = 0.61,$
  $w((1, r_2), r_4) = 2.68, \quad w((2, r_2), r_1) = 7.54,$
  $w((2, r_2), r_3) = 0.10, \quad w((1, r_4), r_1) = 1.29,$
  $w((1, r_4), r_2) = 0.61, \quad w((1, r_4), r_3) = 2.68,$
  $w((2, r_4), r_1) = 1, \quad w((2, r_4), r_3) = 0.20.$

The game graph $G(\mathcal{F}, \mathcal{R}, \tau)$ is shown in Figure 4. We then obtain the following stabilizing switching signal α for $G(\mathcal{F}, \mathcal{R}, \tau)$: $\alpha(r_1) = (2, r_1)$, $\alpha(r_2) = (2, r_2)$, $\alpha(r_3) = (2, r_3)$ and $\alpha(r_4) = (1, r_4)$.

7 Conclusion

In this paper, we presented an abstraction based approach for the synthesis of periodic switching controllers for stabilizing a family of linear systems. The

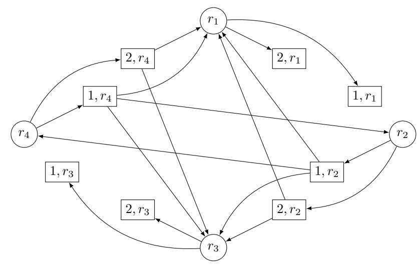


Figure 4: Game graph $G(\mathcal{F}, \mathcal{R}, \tau)$ for Example 2

broad approach of constructing a game graph and extracting a periodic controller is inspired by the results of [23]. In the future, we intend to consider periodic controllers with delays as well as systems with non-linear dynamics. The challenges with the latter arise from the fact that often closed form solutions are not known for non-linear dynamics and hence, computing precise over-approximations of even bounded time reachable set is non-trivial, which is required for the computation of edges and weights in the abstract game graph.

References

[1] P. Bauer, K. Premaratne, and J. Duran, *A necessary and sufficient condition for robust asymptotic stability of time-variant discrete systems*, IEEE Transactions on Automatic Control, 38 (1993), pp. 1427–1430.

[2] M. Branicky, *Stability of hybrid systems*, in Encylopedia of Life Support Systems, vol. Theme 6.43:Control Sytems, Robotics and Automation, 2004.

[3] R. DeCarlo, M. Branicky, S. Pettersson, and B. Lennartson, *Perspectives and results on the stability and stabilizability of hybrid systems*, Proceedings of the IEEE, 88 (2000), pp. 1069–1082.

[4] J. C. Geromel and P. Colaneri, *Stability and stabilization of continuous-time switched linear systems*, SIAM Journal on Control and Optimization, 45 (2006), pp. 1915–1930.

[5] E. Grädel, W. Thomas, and T. Wilke, eds., *Automata Logics, and Infinite Games: A Guide to Current Research*, Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[6] W. P. M. H. Heemels, A. Kundu, and J. Daafouz, *On lyapunov-metzler inequalities and s-procedure characterisations for the stabilisation of switched linear systems*, IEEE Transactions on Automatic Control, 62 (2017), pp. 4593 – 4597.

[7] J. P. Hespanha and A. S. Morse, *Stability of switched systems with average dwell-time*, in CDC, Dec 1999, pp. 2655–2660.

[8] M. Johansson, *Piecewise Linear Control Systems*, Springer, 2003.

[9] A. Kundu and D. Chatterjee, *Stabilizing switching signals for switched systems*, IEEE Transactions on Automatic Control, 60 (2015), pp. 882 – 888.

[10] A. Kundu and D. Chatterjee, *Stabilizing switching signals: a transition from point-wise to asymptotic conditions*, Systems and Control Letters, 106 (2017), pp. 16–23.

[11] D. Liberzon, *Switching in systems and control*, Systems & Control: Foundations & Applications, Birkhäuser Boston Inc., Boston, MA, 2003.

[12] D. Liberzon, J. P. Hespanha, and A. S. Morse, *Stability of switched systems: a lie-algebraic condition*, Systems Control Letters, 37 (1999), pp. 117–122.

[13] H. Lin and P. J. Antsaklis, *Switching stabilizability for continuous-time uncertain switched linear systems*, 2007.

[14] ———, *Stability and stabilizability of switched linear systems: A survey of recent results*, IEEE Transactions on Automatic Control, 54 (2009), pp. 308–322.

[15] A. S. Morse, *Supervisory control of families of linear set-point controllers - part 1: Exact matching*, IEEE Transactions on Automatic Control, 41 (1998), pp. 1413–1431.

[16] K. Narendra and J. Balakrishnan, *A common lyapunov function for stable lti systems with commuting a-matrices*, IEEE Transactions on Automatic Control, 39 (1994), pp. 2469–2471.

[17] Pavithra and M. G. Soto, *Counterexample guided abstraction refinement for stability analysis*, in CAV, 2016, pp. 495–512.

[18] S. Pettersson, *Synthesis of switched linear systems*, in CDC, vol. 5, 2003, pp. 5283–5288.

[19] P. Prabhakar, S. Duggirala, S. Mitra, and M. Viswanathan, *Hybrid automata-based cegar for rectangular hybrid automata*, in VMCAI, 2013.

[20] P. Prabhakar and M. G. Soto, *Abstraction based model-checking of stability of hybrid systems*, in CAV, 2013, pp. 280–295.

[21] ———, *An algorithmic approach to stability verification of polyhedral switched systems*, in ACC, 2014, pp. 2318–2323.

[22] P. Prabhakar and M. G. Soto, *Hybridization for stability analysis of switched linear systems*, in HSCC, 2016.

[23] P. Prabhakar and M. G. Soto, *Formal synthesis of stabilizing controllers for switched systems*, in HSCC, 2017, pp. 111–120.

[24] P. Prabhakar and M. Viswanathan, *On the decidability of stability of hybrid systems*, in HSCC, 2013, pp. 53–62.

[25] R. Shorten and K. Narendra, *Necessary and sufficient conditions for the existence of a common quadratic lyapunov function for two stable second order linear time-invariant systems*, in ACC, vol. 2, Jun 1999, pp. 1410–1414.

[26] R. Shorten, K. Narendra, and O. Mason, *A result on common quadratic lyapunov functions*, IEEE Transactions on Automatic Control, 48 (2003), pp. 110–113.

[27] E. Skafidas, R. J. Evans, A. V. Savkin, and I. R. Petersen, *Stability results for switched controller systems*, in Automatica, vol. 35, 1999, pp. 553–564.

[28] G. Zhai, *Quadratic stabilizability of discrete-time switched systems via state and output feedback*, in CDC, vol. 3, 2001, pp. 2165–2166.

[29] G. Zhai, H. Lin, and P. Antsaklis, *Controller failure time analysis for symmetric h infin; control systems*, in CDC, vol. 3, Dec 2003, pp. 2459–2464.