Control of Asynchronous Dynamical Systems with Rate Constraints on Events

Arash Hassibi[1] Stephen P. Boyd Jonathan P. How

Information Systems Laboratory
Stanford University
Stanford, CA 94305-9510, USA

Abstract— In this paper we consider dynamical systems which are driven by "events" that occur asynchronously. It is assumed that the event rates are fixed, or at least they can be bounded on any time period of length T. Such systems are becoming increasingly important in control due to the very rapid advances in digital systems, communication systems, and data networks. Examples of such systems include, control systems in which signals are transmitted over an asynchronous network; distributed control systems in which each subsystem has its own objective, sensors, resources and level of decision making; parallelized numerical algorithms in which the algorithm is separated into several local algorithms operating concurrently at different processors; and queuing networks. We present a Lyapunov-based theory for asynchronous dynamical systems and show how Lyapunov functions and controllers can be constructed for such systems by solving linear matrix inequality (LMI) and bilinear matrix inequality (BMI) problems. Examples are also presented to demonstrate the effectiveness of the approach.

Keywords: Asynchronous dynamical systems, multi-rate systems, Lyapunov theory, linear matrix inequality (LMI), bilinear matrix inequality (BMI).

1 Introduction

Due to the very rapid advances in digital systems, communication systems, and data networks, asynchronous dynamical systems are becoming increasingly important in control from a practical point of view. Asynchronous dynamical systems can model a vast array of systems such as control systems in which signals are transmitted over an asynchronous communication network; distributed control systems comprising of many interacting subsystems in which each subsystem has its own objective, sensors, resources and level of decision making; parallelized numerical algorithms in which the algorithm is separated into several local algorithms operating concurrently at different processors; and queuing networks.

Roughly speaking, asynchronous dynamical systems are systems that incorporate both discrete and continuous dynamics, with the discrete dynamics governed by finite automata, and the continuous dynamics represented by ordinary differential (or difference) equations at each discrete state. The discrete dynamics is driven asynchronously by discrete events, which are assumed to occur at a fixed rate. In other words, events trigger discrete state transitions, and the discrete state in turn, determines the dynamics governing the continuous state.

Control theory to date has mainly concentrated on continuous synchronous centralized control. Current control methods are almost all based on uniform sampling in time, with all sensors, actuators and processors synchronized. Signals are transmitted perfectly — none are lost, and if there is any delay at all, it is fixed. In the future, however, more and more systems are built around packet-switched networks, where signals can be lost, delayed by varying amounts, etc. The current approach is to design the network so that data transmission is perfectly predictable — regular sensor samples arriving every 20 milliseconds, say. Therefore, the challenge is to develop a new system and control theory (and practice) that works in the more natural network environment: asynchronous and packetized.

Asynchronous dynamical systems can have very complex behavior, even those with very simple dynamics. Hence, as in robust control, it is not surprising that many problems for such systems are known or conjectured to be computationally intractable (NP-hard) or even undecidable. For example, the stability problem of a linear time-varying system (which is a simple asynchronous system) cannot be solved in polynomial-time [1]. Therefore, it is very unlikely to formulate control problems for asynchronous systems *exactly* as computationally efficient (polynomial-time) optimization problems. It is expected, however, to develop some *semi-heuristic* methods that are very effective on certain types of problems. By semi-heuristic we mean a method that guarantees its results when it works, but is not guaranteed to work for all input data. Such a method results, for example, when we search over a fixed, finite-dimensional class of Lyapunov functions that guarantee some specification for a given asynchronous system — it may not be possible to find such a function, but if one is found, the result is unambiguous.

[1]Contact author. E-mail: `arash@isl.stanford.edu`. Research supported by Air Force (under F49620-97-1-0459), AFOSR (under F49620-95-1-0318), and NSF (under ECS-9222391 and EEC-9420565).

This research is an effort in this direction.

Previous work on asynchronous dynamical systems in control has mainly focused on the stability of asynchronous multi-rate sampled-data systems and finite-difference system of equations with unknown (bounded) delays (cf. [2, 3, 4, 5]). The framework considered in this work is more general, and in addition to allowing possible unknown delays in the continuous dynamics, we also assume that the continuous dynamics are affected by a finite set of discrete events that occur at a fixed known rate. This enables us to model, for example, loss in transmitted signals, "large" periods of silence in communication, multi-rate communication and processing, and asynchronous inputs in queuing systems. Note that no statistical assumption is made on the arrival times of the events.

In this paper, we introduce a Lyapunov-based approach for analysis and control of such systems. More importantly, we show that computing Lyapunov functions to prove some level of performance or to design controllers for such systems can be cast as optimization problems involving linear matrix inequalities (LMIs) and bilinear matrix inequalities (BMIs). LMI problems can be efficiently solved (globally) using widely available software (see, *e.g.*, [6, 7, 8]), and BMI problems can be solved (locally) by alternating between LMIs. A very important and useful feature of this approach is that the results and analysis tools of robust control can be easily mixed with those presented here for asynchronous systems. This enables us to analyze very complex dynamical systems that, for example, are asynchronous, include logic variables, and have various nonlinearities, structured uncertainties and unknown delays.

In the next section we give a more formal definition of asynchronous dynamical systems with constraints on the rate of events, followed by a couple of examples of such systems. In §3 we present a Lyapunov-based theory for asynchronous dynamical systems. In §4 we show how Lyapunov functions can be numerically computed for such systems using SDP. Examples are given in §5 and the paper is concluded in §6.

2 Asynchronous dynamical systems with rate constraints on events

2.1 Definition

In this paper, an asynchronous dynamical system (ADS) with rate constraints on events is a tuple

$$\mathcal{A} = (\mathbf{R}_+, \{1, \ldots, N\}, \mathbf{R}^n, E, R, \mathcal{I}, F) \qquad (1)$$

where $\mathbf{R}_+$ is time, $\{1, \ldots, N\}$ is the discrete state-space, $\mathbf{R}^n$ is the continuous state-space, E is the set of events, R is the set of event rates, $\mathcal{I} : \{1, \ldots, N\} \to 2^E$ is the discrete state-event function, and F is the set of continuous dynamical system functions. Below is a brief description of each of these and other related concepts used in this paper.

- *Time.* This is the set $\mathbf{R}_+$. Time is denoted by t.
- *Discrete state, continuous state, state-space.* The discrete and continuous states at time t are denoted by $s(t)$ and $x(t)$ respectively. For all t, $s(t) \in \{1, \ldots, N\}$ and $x(t) \in \mathbf{R}^n$. The state-space of $\mathcal{A}$ is the set $\{1, \ldots, N\} \times \mathbf{R}^n$.
- *Set of continuous dynamical system functions.* $F = \{f_1, \ldots, f_N\}$ where $f_i : \mathbf{R}^n \to \mathbf{R}^n$. These functions determine the dynamics of x at each discrete state $s = i$ (more below).
- *Set of events.* $E = \{E_1, \ldots, E_M\}$ is the set of possible events E_i.
- *Event indicator functions.* Define the ith event indicator function $e_i : \mathbf{R}_+ \to \{0, 1\}$ for $i = 1, \ldots, M$ as

$$e_i(t) = \begin{cases} 1 & E_i \text{ has occurred at time } t \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

- *Rate of events.* $R = \{r_1, \ldots, r_M\}$ is the set of event rates in which r_i satisfying $0 \le r_i \le 1$ is the rate of occurrence of event E_i over time. Roughly, over any time period $[t,\ t+T]$ for large enough T, $r_i T$ is the total amount of time that E_i has occurred. In other words,

$$\lim_{T\to\infty} \frac{1}{T} \int_t^{t+T} e_i(\tau)\, d\tau = r_i.$$

- *Discrete state-event sets.* The function $\mathcal{I} : \{1, \ldots, N\} \to 2^E$ assigns a set of events to each discrete state. By definition, $\mathcal{I}(i)$ is the ith discrete state-event set. We assume that

$$\mathcal{I}(i) = \left\{E_{i_1}, E_{i_2}, \ldots E_{i_{M_i}}\right\} \qquad (3)$$

where $e_j^{(i)} \in E$ for $j = 1, \ldots, M_i$ The discrete state event sets determine the evolution of the discrete state under the influence of the events—$s(t) = i$ if and only if the events in $\mathcal{I}(i)$ have occurred (more below).

- *State dynamics.* The evolution of the continuous state x over time is given by the integral equation

$$x(t) = \int_0^t f_{s(\tau)}(x)\, d\tau + x(0).$$

In other words, at discrete state s, the continuous dynamics are governed by the dynamical equation $\dot{x} = f_s(x)$. The evolution of the discrete state s is such that $s(t) = i$ if and only if $\mathcal{I}(i)$ is the set of events occurred at time t. Hence, the discrete state $s(t)$ remains unchanged until a new event occurs, so that $\mathcal{I}(j)$ becomes the new set of occurred events. At that point $s(t)$ changes to j.

- *Trajectory.* Roughly speaking, a trajectory of $\mathcal{A}$ is any function $(s, x) : \mathbf{R}_+ \to \{1, \ldots, N\} \times \mathbf{R}^n$ where s and x satisfy the discrete and continuous state dynamics of $\mathcal{A}$ respectively.

- *Transition diagram.* It is possible to associate to $\mathcal{A}$ a directed graph $\mathcal{G}(\mathcal{A})$ called a transition diagram. In this graph the nodes correspond to the discrete states and the arcs between nodes correspond to events that take the discrete state from one node to the other. Hence, the nodes are labeled with the discrete state numbers and the arcs are labeled with the events (see examples below). Any trajectory of the discrete state corresponds to the sequence of nodes along some forward path in $\mathcal{G}(\mathcal{A})$.

2.2 Extensions

It is possible to extend the definition of ADS in the previous section to more general cases. For example, we can simply add inputs and outputs by considering the continuous dynamics $\dot{x} = f_i(x, u)$ and $y = g_i(x, u)$ at discrete state i. Or, we can add delays in the continuous dynamics so that $\dot{x} = f_i(x, x(t-\tau))$. We can also assume that over any time period $[t,\ t+T]$ (T is given), r_i, the rate of occurrence of the ith event is bounded such that $\bar{r}_i - \delta_i \le r_i \le \bar{r}_i + \delta_i$ ($\bar{r}_i, \delta_i$ are given). In addition, we can assume bounds on the interarrival times of different events.

2.3 Examples

In this subsection we present two examples for asynchronous systems. It should be noted that although the descriptions for these systems appear to be simple, the behavior of these systems can be extremely complex.

2.3.1 Control over asynchronous network: Consider the asynchronous control system of Figure 1. The plant is a linear system with exogenous input w, control input u, and regulated output z. The controller is a simple static linear feedback and should be designed such that z is "small" in the presence of w.

What complicates the control system is that there are two (sample & hold) *asynchronous* switches in this system. To model the sample & hold operation of the switch (Figure 2), we assume that the dynamics of the switch when closed is governed by $\dot{x}_{\rm s} = -\alpha x_{\rm s} + \alpha p$, $q = x_{\rm s}$ (where the time constant $1/\alpha$ is "small"), and the dynamics of the switch when open is governed by $\dot{x}_{\rm s} = -\bar{\alpha} x_{\rm s}$, $q = x_{\rm s}$ (where the time constant $1/\bar{\alpha}$ is "large"). The first switch has rate r_1 and when closed connects the state x of the plant to the controller. The second switch has rate r_2 and when closed connects the controller output u to the plant input. This situation could model, for example, a remote controller which is connected to the plant over an asynchronous network

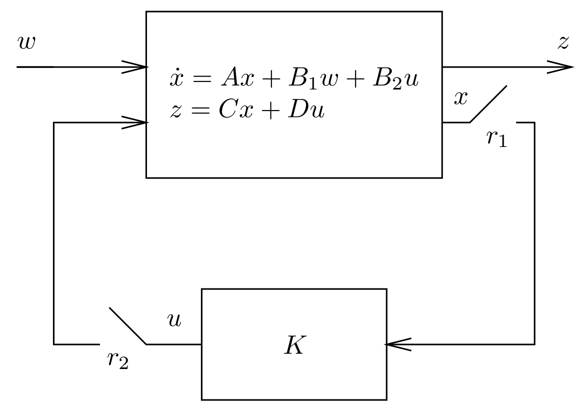

Figure 1: Asynchronous control system.

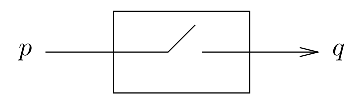

Figure 2: Asynchronous sample & hold switch.

such as the internet. Hence, in this case, the measurements x are sent to the controller over an asynchronous communication link of rate r_1 and the controller commands are sent to the plant over an asynchronous communication link of rate r_2.

This system can be easily put in the ADS framework of this paper. The transition diagram for this system is given in Figure 3, where E_1 and $\bar{E}_1$ denote the events of closing and opening the first switch, and E_2 and $\bar{E}_2$ denote the events of closing and opening the second switch.

At each discrete state of the system or node of the transition diagram, it is easy to write the dynamics of the system. For example, at the $E_1 E_2$ state, both switches are closed and the continuous dynamics are given by:

$$\begin{aligned}
&\dot{x} = Ax + B_1 w + B_2 x_{{\rm s},2}, \quad z = Cx + D x_{{\rm s},2},\\
&\dot{x}_{{\rm s},1} = -\alpha x_{{\rm s},1} + \alpha x,\\
&\dot{x}_{{\rm s},2} = -\alpha x_{{\rm s},2} + \alpha K x_{{\rm s},1},
\end{aligned}$$

where $x_{{\rm s},1}(t) \in \mathbf{R}^n$ and $x_{{\rm s},2}(t) \in \mathbf{R}$ are the states of the first and second switches respectively. Or at the $\bar{E}_1 E_2$ state:

$$\begin{aligned}
&\dot{x} = Ax + B_1 w + B_2 x_{{\rm s},2}, \quad z = Cx + D x_{{\rm s},2},\\
&\dot{x}_{{\rm s},1} = -\bar{\alpha} x_{{\rm s},1},\\
&\dot{x}_{{\rm s},2} = -\alpha x_{{\rm s},2} + \alpha K x_{{\rm s},1}.
\end{aligned}$$

(Note that a more realistic model for a control system over an asynchronous network would also include unknown delays in the data link.) In §5 we will come back to this problem, and we will design a controller

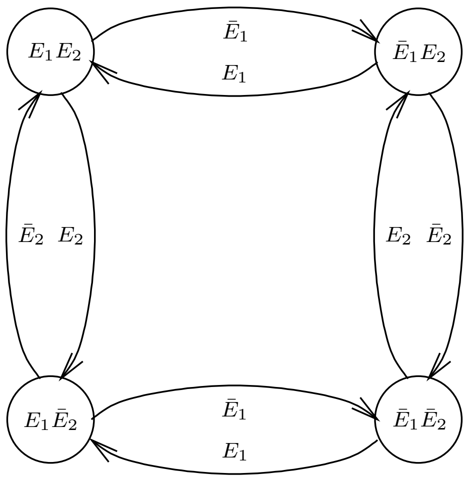

Figure 3: Transition diagram of asynchronous control system.

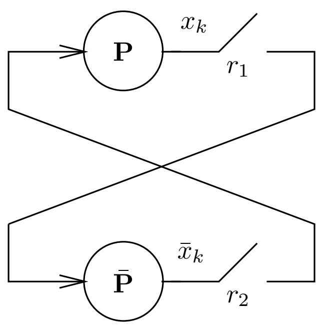

Figure 4: Asynchronous dual processor version for fixed-point problem.

gain K to regulate the output z in the presence of the disturbance w.

2.3.2 Parallelized algorithm: The fixed point problem $x_{k+1} = Ax_k + b$ converges to the fixed point $x = (I - A)^{-1}b$ from any initial condition if and only if all eigenvalues of A lie inside the unit circle in the complex plane. In this example, however, we assume that this fixed point problem is implemented in parallel by separating it into two local problems operating concurrently, say, on two different processors $\mathbf{P}$ and $\bar{\mathbf{P}}$. The situation is shown schematically in Figure 4, where we assume $\mathbf{P}$ performs the iteration for the upper half and $\bar{\mathbf{P}}$ performs the iteration for the lower half of x. Each processor does not have to wait for information to become available from the other processor. The only assumption is that $\mathbf{P}$ makes its state available to $\bar{\mathbf{P}}$ at rate r_1 and $\bar{\mathbf{P}}$ makes its state available to $\mathbf{P}$ at rate r_2. The question is whether this asynchronous implementation version of the fixed point problem is stable and converges to $x = (I - A)^{-1}b$.

To explain this asynchronous fixed point problem in more detail, suppose that A and b are partitioned as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

Furthermore, assume that the states of $\mathbf{P}$ and $\bar{\mathbf{P}}$ at time k are partitioned (consistently) respectively as follows:

$$x_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}, \quad \bar{x}_k = \begin{bmatrix} \bar{x}_{1,k} \\ \bar{x}_{2,k} \end{bmatrix}.$$

Depending on the position of the switches in Figure 4 (described by events E_1,E_2,$\bar{E}_1$,$\bar{E}_2$ as in the previous example) the iteration would be different. For example, when both switches are open:

$$\begin{aligned} \mathbf{P}: \quad & x_{1,k+1} = A_{11}x_{1,k} + A_{12}x_{2,k} + b_1 \\ & x_{2,k+1} = x_{2,k} \end{aligned}$$

$$\begin{aligned} \bar{\mathbf{P}}: \quad & \bar{x}_{1,k+1} = \bar{x}_{1,k} \\ & \bar{x}_{1,k+1} = A_{21}\bar{x}_{1,k} + A_{22}\bar{x}_{2,k} + b_2. \end{aligned}$$

In other words, when the switches are open, the processors continue updating their states without using the state information from the other processor. When both switches are closed:

$$\begin{aligned} \mathbf{P}: \quad & x_{1,k+1} = A_{11}x_{1,k} + A_{12}\bar{x}_{2,k} + b_1 \\ & x_{2,k+1} = \bar{x}_{2,k} \end{aligned}$$

$$\begin{aligned} \bar{\mathbf{P}}: \quad & \bar{x}_{1,k+1} = x_{1,k} \\ & \bar{x}_{1,k+1} = A_{21}x_{1,k} + A_{22}\bar{x}_{2,k} + b_2, \end{aligned}$$

so each processor uses the state information from the other processor to update its state. The state update equations for the two other positions of the switches can be similarly written. We will come back to this example in §5.2 and prove stability of such a parallelized asynchronous fixed point problem using the methods presented in this paper.

3 Lyapunov-based theory for ADS

In this section we present a Lyapunov-based theory for analysis of ADS. In classical Lyapunov theory we require a Lyapunov function to decrease monotonically along state trajectories of the system to prove stability. Here, on the other hand, we require a Lyapunov-type function to decrease "on the average" along state trajectories of the asynchronous system. In the following subsections we give conditions on a Lyapunov-type function V which proves different performance measures for ADS.

3.1 Exponential stability

By definition, an ADS is exponentially stable if

$$\lim_{t\to\infty} e^{\alpha t}\|x(t)\| = 0$$

for some $\alpha > 0$. The largest such α is referred to as the decay rate of the system. Clearly, exponential stability implies uniform asymptotic stability. In what follows we present a Lyapunov-type argument to compute bounds on the decay rate of an ADS.

Suppose $V : \mathbf{R}^n \to \mathbf{R}_+$, that V is continuously differentiable and

$$\beta_1 \|x\|^2 \le V(x) \le \beta_2 \|x\|^2 \tag{4}$$

where $\beta_{1,2} > 0$. The decay rate of the ADS is greater than α if the scalars $\alpha_1, \alpha_2, \ldots, \alpha_M$ exist such that

$$r_1\alpha_1 + r_2\alpha_2 + \cdots r_M\alpha_M > \alpha > 0 \tag{5}$$

and

$$DV(x)f_i(x) \le -2(\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_{M_i}})V(x) \tag{6}$$

for $i = 1, \ldots, N$, in which i_j for $j = 1, \ldots, M_i$ are defined in (3). These conditions roughly state that V does not have to decrease monotonically at some rate α along state trajectories, but rather it should decrease at a rate α "on the average". For example, it might be possible that $\alpha_{i_1} + \cdots + \alpha_{i_{M_i}} < 0$ for some i so that V *increases* at discrete state i. However, as long as the "average rate" $r_1\alpha_1 + \cdots r_M\alpha_M > 0$ the system remains exponentially stable.

The proof is as follows. Suppose that the discrete state transitions of any trajectory of the system occur at times $0 = t_1 < t_2 < t_3 < \cdots$, so that $s(t)$ is constant for $t \in [t_k,\ t_{k+1}]$. Then, for $t \in [t_k,\ t_{k+1}]$, condition (6) gives

$$\frac{\dot V(x(t))}{V(x(t))} \le -2(\alpha_{s_1} + \alpha_{s_2} + \cdots + \alpha_{s_{M_s}})$$

or

$$\begin{aligned} \log V(x(t_{k+1})) - \log V(x(t_k)) \le \\ -2\alpha_{s_1}(t_{k+1} - t_k) + \cdots + \alpha_{s_{M_s}}(t_{k+1} - t_k). \end{aligned} \tag{7}$$

Note that whenever an event E_i occurs we have a contributing term $\alpha_i(t_{k+1} - t_k)$ at the right hand side of (7). Hence, summing up these inequalities for $k = 1, 2, \ldots, K-1$ gives

$$\begin{aligned} &\log V(x(t_K)) - \log V(x(0)) \le \\ &-2\alpha_1 \begin{pmatrix} \text{total time} \\ E_1 \text{ occurred} \end{pmatrix} - \cdots - 2\alpha_M \begin{pmatrix} \text{total time} \\ E_M \text{ occurred} \end{pmatrix}. \end{aligned}$$

In the limit, the total time event E_i occurs is equal to $r_i t_K$ as $K \to \infty$. Therefore

$$\begin{aligned} \log V(x(t_K)) - \log V(x(0)) \le \\ -2\alpha_1 r_1 t_K - \cdots - 2\alpha_M r_M t_K \end{aligned}$$

or by (5)

$$\log V(x(t_K)) - \log V(x(0)) < -2\alpha t_K$$

so that

$$V(x(t_K)) < e^{-2\alpha t_K} V(x(0)).$$

Now using (4) we get

$$e^{\alpha t_K}\|x(t_K)\| < \sqrt{\frac{\beta_2}{\beta_1}}\|x(0)\|$$

or

$$\lim_{K\to\infty} e^{\alpha t_K}\|x(t_K)\| = 0.$$

Remark. If the evolution of x is given by a difference equation $x_{k+1} = f_s(x_k)$ instead of the differential equation $\dot x(t) = f_s(x(t))$, a sufficient condition for exponential stability, is the existence of V as in (4), and $\alpha_1, \alpha_2, \ldots, \alpha_M > 0$ satisfying

$$V(x_k{+}1) - V(x_k) \le (\alpha_{i_1}^{-2}\alpha_{i_2}^{-2}\cdots\alpha_{i_{M_i}}^{-2} - 1)V(x_k)$$

and

$$\alpha_1^{r_1}\alpha_2^{r_2}\cdots\alpha_M^{r_M} > \alpha > 1.$$

This last condition can be equivalently written as

$$r_1\log\alpha_1 + r_2\log\alpha_2 + \cdots + r_M\log\alpha_M > \log\alpha > 0.$$

Under these conditions $\lim_{k\to\infty}\alpha^k\|x_k\| = 0$.

3.2 Invariant sets over period T

In classical Lyapunov theory the level sets of Lyapunov functions of a system are invariant sets for the system. In order to be able to say something about invariant sets for ADS we need to have a bound on the event rates over some *finite* period of time T.

Specifically, suppose that over any period of time of length T

$$\bar r_i - \delta_i \le r_i \le \bar r_i + \delta_i.$$

Moreover, suppose that

$$\begin{aligned} \alpha = \min\{r_1\alpha_1 + \cdots + r_M\alpha_M \ \mid \bar r_i - \delta_i \le r_i \le \bar r_i + \delta_i, \\ i = 1, \ldots, M\}, \end{aligned}$$

and condition (6) holds. Then using a similar argument to the previous subsection it can be shown that

$$V(x(T)) < e^{-2\alpha T} V(x(0)).$$

So if $V(x(0)) \le \eta$, we have $V(x(T)) \le \eta$ since $e^{-2\alpha T} < 1$. In other words, the level sets of the Lyapunov-type function V are invariant if we consider samples of x at times $0, T, 2T, 3T$, etc.

3.3 Bound on return time

Suppose that as in the previous subsection we have bounds on the event rates over any time interval of length T. The *return time* of a stable ADS for the set $\mathcal{P} \subset \mathbf{R}^n$ is defined as the smallest T_{r} such that if

$x(0) \in \mathcal{P}$, then $x(t) \in \mathcal{P}$ for $t = T_\mathrm{r}, T_\mathrm{r} + T, T_\mathrm{r} + 2T$, etc. Let

$$\mathcal{E}_\eta = \{x \mid V(x) \le \eta\}$$

and suppose that

$$e^{-\alpha(kT)}\mathcal{E}_\eta \subseteq \mathcal{P} \subseteq \mathcal{E}_\eta. \tag{8}$$

Since $\mathcal{E}_\eta$ is an invariant ellipsoid for samples of x at integer multiples of T, and moreover $x(0) \in \mathcal{P}$ implies $x(t) \in e^{-\alpha t}\mathcal{E}_\eta$ for $t = 0, T, 2T, \ldots$, from (8) we can conclude that if $x(0) \in \mathcal{P}$ then $x(t) \in \mathcal{P}$ for $t = kT, kT + T, kT + 2T$, etc. In other words, $T_\mathrm{r} = kT$ is an upper bound on the return time.

3.4 Bound on $\mathcal{L}_\infty$ to RMS gain

In this subsection we assume that the dynamics of the continuous state at discrete state i of the ADS is given by

$$\dot{x} = f_i(x, w), \quad z = g_i(x, w),$$

where w is the exogenous input and z is the regulated output of the system. Here we give a condition for a bound on the $\mathcal{L}_\infty$ to RMS induced gain from w to z defined as

$$\sup_{\|w\|_\infty \neq 0,\ x(0)=0} \frac{\left(\limsup_{T\to\infty} \frac{1}{T}\int_0^T z^T z\,dt\right)^{1/2}}{\|w\|_\infty}. \tag{9}$$

The $\mathcal{L}_\infty$ to RMS gain of the ADS is less than γ if there exists $\gamma_1, \gamma_2, \ldots, \gamma_M$ such that

$$r_1\gamma_1^2 + r_2\gamma_2^2 + \cdots r_M\gamma_M^2 < \gamma^2 \tag{10}$$

and

$$DV(x)f_i(x, w) \le (\gamma_{i_1}^2 + \gamma_{i_2}^2 + \cdots + \gamma_{i_{M_i}}^2)w^T w - z^T z \tag{11}$$

for $i = 1, \ldots, N$, in which i_j for $j = 1, \ldots, M_i$ are defined in (3).

The proof is as follows. Suppose that the discrete state transitions of any trajectory of the system occur at times $0 = t_1 < t_2 < t_3 < \cdots$, so that $s(t)$ is constant for $t \in [t_k,\ t_{k+1}]$. Then, for $t \in [t_k,\ t_{k+1}]$, condition (11) gives

$$\dot{V}(x(t)) \le (\gamma_{i_1}^2 + \gamma_{i_2}^2 + \cdots + \gamma_{i_{M_i}}^2)w^T w - z^T z$$

or

$$\begin{aligned}
&V(x(t_{k+1})) - V(x(t_k)) \le \\
&(\gamma_{i_1}^2 + \cdots + \gamma_{i_{M_i}}^2)\int_{t_k}^{t_{k+1}} w^T w\,dt - \int_{t_k}^{t_{k+1}} z^T z\,dt \le \\
&\quad (\gamma_{i_1}^2 + \cdots + \gamma_{i_{M_i}}^2)(t_k - t_{k+1})\|w\|_\infty^2 - \int_{t_k}^{t_{k+1}} z^T z\,dt.
\end{aligned}$$

Now summing up these inequalities for $k = 1, 2, \ldots, K-1$ gives

$$\begin{aligned}
V(x(t_K)) - V(x(0)) \le \gamma_1^2 &\begin{pmatrix}\text{total time}\\ E_1 \text{ occurred}\end{pmatrix}\|w\|_\infty^2 + \cdots \\
&+ \gamma_M^2\begin{pmatrix}\text{total time}\\ E_M \text{ occurred}\end{pmatrix}\|w\|_\infty^2 - \int_0^{t_K} z^T z\,dt.
\end{aligned}$$

In the limit, the total time event E_i occurs is equal to $r_i t_K$ as $K \to \infty$. Therefore, since $x(0) = 0$

$$\begin{aligned}
V(x(t_K)) \le \gamma_1^2 r_1 t_K \|w\|_\infty^2 + \cdots \\
+ \gamma_M^2 r_M t_K \|w\|_\infty^2 - \int_0^{t_K} z^T z\,dt.
\end{aligned}$$

Using (10) and since $V(x(t_K)) \ge 0$ we get

$$\frac{\frac{1}{t_K}\int_0^{t_K} z^T z\,dt}{\|w\|_\infty^2} \le \gamma^2.$$

In other words, the $\mathcal{L}_\infty$ to RMS gain of the system is less than γ.

> **Remark.** Note that the method just described for bounding the $\mathcal{L}_\infty$ to RMS gain of a given ADS would not work if any of the dynamical systems $\dot{x} = f_i(x, w)$, $z = g(x, w)$ are unstable (and therefore have an induced $\mathcal{L}_2$ gain of infinity). This conservatism can be fixed by requiring
>
> $$\begin{aligned}DV(x)f_i(x, w) \le (\gamma_{i_1}^2 + \cdots + \gamma_{i_{M_i}}^2)w^T w - \\ z^T z - 2(\alpha_{i_1} + \cdots + \alpha_{i_{M_i}})V(x)\end{aligned} \tag{12}$$
>
> with $r_1\alpha_1 + \cdots + r_M\alpha_M > 0$ instead of (11).

4 Numerical computation of Lyapunov functions for ADS

In the previous section we presented a Lyapunov-based theory for ADS. In this section we demonstrate methods to actually *compute* Lyapunov functions for such systems using semidefinite programming (SDP). This is done by searching over a fixed finite-dimensional class of Lyapunov functions.

Different classes of Lyapunov functions have been proposed in the control literature for analyzing various types of (nonlinear) dynamical systems. These include quadratic Lyapunov functions, quadratic plus integral of the nonlinearity Lyapunov functions, quadratic plus integral quadratic terms Lyapunov functions, piecewise-quadratic Lyapunov functions, path-dependent Lyapunov functionals, etc. See, for example, [9, 10, 11, 12, 13, 14] and references therein. Each of these classes of Lyapunov functions are effective on certain types of nonlinear dynamical systems. For example, the quadratic plus integral of nonlinearity Lyapunov function is useful for analyzing the Luré system and leads to the well-known Popov criterion. Or Lyapunov functions involving integral quadratic constraints (IQCs) can be used to analyze many different nonlinear uncertain systems, such as systems with unknown delays, odd monotone sector-bounded nonlinearities, hysteresis nonlinearities, etc.

In this section we search over simple quadratic Lyapunov functions V given by

$$V(x) = x^T P x, \quad P \succ 0.$$

and we assume that the functions f_i are *linear* so that the continuous dynamics is linear at each discrete state of the ADS, *i.e.*,

$$\dot{x} = A_s x, \quad s = 1, \dots, N.$$

We show that finding such a V that proves stability or bounds the $\mathcal{L}_\infty$ to RMS gain of a given ADS can be cast as LMI or BMI problems which can then be solved (globally or locally) using widely available software. Constructing quadratic Lyapunov functions for proving other performance measures can be handled similarly. We also briefly explain how controller synthesis can be performed in this framework.

Note that the following results can be easily extended when the search is performed over more sophisticated classes of Lyapunov functions suitable for different types of *nonlinear* f_i's. For example, if the continuous dynamics at each discrete state is a Luré system or a system involving unknown time delays, in order to get less conservative results, we can respectively search over Popov Lyapunov functions or IQCs for unknown time delays.

4.1 Analysis of exponential stability

With $V(x) = x^T P x$ and $f_i(x) = A_i x$ condition (6) can be written as

$$x^T (A_i^T P + P A_i) x \leq -2(\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_{M_i}}) V(x)$$

which is equivalent to the matrix inequality

$$A_i^T P + P A_i \preceq -2(\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_{M_i}}) P. \tag{13}$$

Hence (13) with $P \succ 0$ and (5) give a sufficient condition for asymptotic stability of the ADS in the design parameters P, and α_i for $i = 1, \dots, N$. (13) is an LMI in P for fixed α_i's, and a linear inequality constraint in α_i's for fixed P. Therefore, the condition for exponential stability is given in terms of a BMI. This BMI can be solved locally by alternating between LMIs by solving SDPs, or globally using a branch and bound technique on the α_i's to find, if any, a feasible set of design parameters.

> **Remark.** If the evolution of x is given by the difference equation $x_{k+1} = A_s x_k$ instead of the differential equation $\dot{x}(t) = A_s x(t)$, it can be shown that the sufficient condition for exponential stability given in the Remark of §3.1 with $V(x) = x^T P x$ is equivalent to a matrix inequality which is an LMI in P for fixed α_i's and is a linear inequality in $\log \alpha_i$'s for fixed P.

4.2 Computing bound on $\mathcal{L}_\infty$ to RMS gain

Here we assume that

$$f_i(x, w) = A_i x + B_i w, \quad g_i(x, w) = C_i x$$

and therefore condition (11) with $V(x) = x^T P x$ becomes

$$\begin{bmatrix} A_i^T P + P A_i + C_i^T C_i & P B_i \\ B_i^T P & -(\gamma_{i_1}^2 + \cdots + \gamma_{i_{M_i}}^2) I \end{bmatrix} \prec 0. \tag{14}$$

This with $P \succ 0$ and (10) give a sufficient condition for an $\mathcal{L}_\infty$ to RMS gain of less than γ. The optimum bound on the gain can be computed by minimizing γ^2 subject to these constraints using SDP.

4.3 Joint controller and Lyapunov function design

It is possible to jointly design a controller to achieve a given level of performance and a Lyapunov function that proves that level of performance. This can be done, for example, by replacing the open-loop system matrices A_i, B_i, and C_i by the closed-loop system matrices in the equations above. In general the matrix inequalities now become bilinear in the matrix P and control parameters and a so-called V-K iteration is required to solve the problem. In many cases, however, it is possible to convert the BMI to a linear matrix inequality by a change of variables (see, *e.g.*, [9]). In such cases, the problem can be solved globally using SDP.

5 Numerical examples

5.1 Control over asynchronous network

In this example we consider controller design under the setup of §2.3.1. The system to be controlled is the simple mechanical system of Figure 5 with $k_1 = 1$, $k_2 = 1$, $b_1 = 1$, and $b_2 = 1$. The exogenous input w and regulated output z are the force applied to the first mass and the displacement of the first mass respectively. The control input u is the force applied to the second mass. The goal is to design a constant gain controller defined by $K \in \mathbf{R}^{1 \times 4}$ in the asynchronous environment of Figure 1 with $r_1 = r_2 = 0.9$ to reduce the RMS gain from input w to output $z = x_1$. The open-loop RMS gain is $\gamma_{\text{OL}} = 1.5$, and the modes of the mechanical system are

$$p_{1,2} = -0.1309 \pm j0.9511, \quad p_{3,4} = -0.0191 \pm j0.5878.$$

The problem of designing K is an output-feedback problem and can be cast as a BMI using the method of §4.2. The resulting BMI can be solved (locally) using a $V - K$ iteration. That is, for a fixed controller K we design a Lyapunov function V that proves (a bound on) system performance, and then for a fixed Lyapunov function V we design a controller K to improve (a bound on) system performance, and we iterate until there is no improvement in system performance.

To get an initial controller gain K for the $V - K$ iteration, we designed an $\mathcal{H}_\infty$ linear constant state-feedback controller for the plant using the regulating output $z_u = [x_1 \ 0.1u]^T$ with $\gamma = 1$. The $V - K$ iteration for the asynchronous control system of Figure 5 then resulted in

$$K = \begin{bmatrix} -1.5336 & -0.3252 & -2.0247 & -0.1021 \end{bmatrix}$$

and a provable level of $\mathcal{L}_\infty$ to RMS gain from w to $z = x_1$ of less than $\gamma_{\text{CL,async}} = 1$. This is a 50% improvement over the open-loop gain. Note that the optimum state-feedback $\mathcal{H}_\infty$ controller in a *synchronous* environment would give $\gamma_{\text{CL,sync}} = 0.4$.

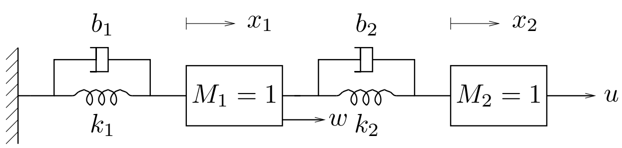

Figure 5: Simple mechanical system considered in the asynchronous environment of Figure 1.

5.2 Parallelized algorithm

Here we study the stability of the parallelized asynchronous fixed point problem of §2.3.2 using the analysis methods of this paper. Specifically, suppose that

$$x_{k+1} = Ax_k, \quad A = \begin{bmatrix} 0.8558 & -0.2895 \\ 0.7295 & -0.6558 \end{bmatrix}$$

The eigenvalues of A are 0.7 and -0.5 so the (synchronous) fixed point problem is stable with a decay rate of $1/0.7 = 1.43$, so x_k converges to the fixed point $x = 0$ as $k \to \infty$.

Under the setup of §2.3.2, for the asynchronous version of this problem

$$\begin{aligned} A_{11} &= 0.8558, \quad A_{12} = -0.2895, \\ A_{21} &= 0.7295, \quad A_{22} = -0.6558. \end{aligned}$$

We assume that $r_1 = r_2 = 0.8$ so that $\mathbf{P}$ and $\bar{\mathbf{P}}$ communicate asynchronously 80% of the time. Using the method of §4.1, a $P \in \mathbf{SR}^{4\times4}$ can be computed such that $V(x) = x^T Px$ proves exponential stability for this asynchronous fixed point problem. Specifically we got

$$P = \begin{bmatrix} 17.3374 & -0.7766 & -3.9773 & -5.4321 \\ -0.7766 & 2.4066 & -1.4905 & 1.9887 \\ -3.9773 & -1.4905 & 4.4845 & -2.3577 \\ -5.4321 & 1.9887 & -2.3577 & 10.3086 \end{bmatrix}$$

$$\begin{aligned} \alpha_1 &= 1.1669, \quad \alpha_2 = 1.2242, \\ \bar{\alpha}_1 &= 0.5897, \quad \bar{\alpha}_2 = 0.5626 \end{aligned}$$

where α_1, α_2, $\bar{\alpha}_1$ and $\bar{\alpha}_2$ correspond to the events E_1, E_2, $\bar{E}_1$ and $\bar{E}_2$ respectively. These values prove a decay rate of at least $\alpha = 1.11$ for the asynchronous parallelized implementation of the algorithm. If we assume that $r_1 = r_2 = 0.9$ it is possible to prove a decay rate bound of $\alpha = 1.25$. Hence, in this case, the asynchronous parallelized implementation using two processors which communicate 90% of the time is guaranteed to outperform the synchronous implementation, since the decay rate of the synchronous iteration is $1.43 < 1.25^2 = 1.56$ (assuming $\mathbf{P}$ and $\bar{\mathbf{P}}$ can operate twice as fast because they only perform half of the computation).

Note that abs(A), *i.e.*, the matrix with elements equal to the absolute values of the elements of A, has an eigenvalue of $1.2261 > 1$. Therefore, according to [4, p.435] for example, in an asynchronous implementation of this problem with communication delays, if delays of greater than two are allowed the iteration can become unstable. The setup in this example is different, however, and we assume there are no delays when the processors communicate. On the other hand, we assume that the processors do not communicate at all times, but only asynchronously at fixed rates given by the values of r_1 and r_2. Hence we allow, for example, "large" periods of time over which no communication occurs, or possibly data loss in transmission.

Note that to deal with unknown communication delays, or quantization and finite precision errors, one can use a richer class of Lyapunov functions V from robust control that, for example, incorporate IQC terms to handle unknown delays or quantization nonlinearities (see, *e.g.*, [12]).

6 Conclusions

In this paper we introduced a Lyapunov-based method for analysis and controller design for asynchronous dynamical systems with rate constraints on events. It was shown that for various performance measures, the analysis and controller design can be cast as LMI or BMI problems which can then be solved (globally or locally) using SDP. Examples were also included to demonstrate the effectiveness of the approach.

Although we only considered computing quadratic Lyapunov functions for asynchronous systems in which the continuous dynamics is linear at each discrete state, it is possible to consider more complicated continuous dynamics and extend these results by searching over richer classes of Lyapunov functions from robust control. For example, we can assume unknown delays and various nonlinearities in the continuous dynamics. Basically, it is possible to mix many results and analysis tools from nonlinear robust control within this asynchronous framework.

References

[1] V. D. Blondel and J. N. Tsitsiklis. NP-hardness of some linear control design problems. *SIAM J. on Control and Optimization*, pages 2118–27, 1997.

[2] A. F. Kleptsyn, V. S. Kozyakin, M. A. Krasnosel'skii, and N. A. Kuznetsov. Effect of small synchronization errors on stability of complex systems. I, II, III. *Avtomatika i Tele-*

mekhanika. Translated in: Automation and Remote Control, 44, 45, 45(7, 3, 8):861–7, 309–14, 1014–8, 1983, 1984.

[3] V. S. Ritchey and G. F. Franklin. A stability criterion for asynchronous multirate linear systems. *IEEE Trans. on Automatic Control*, 34(5):529–35, 1989.

[4] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[5] Y.-F. Su, A. Bhaya, E. Kaszkurewicz, and V. S. Kozyakin. Further results on stability of asynchronous discrete-time linear systems. In *Proc. IEEE Conf. on Decision and Control*, pages 915–20, San Diego, CA, 1997.

[6] L. Vandenberghe and S. Boyd. SP*: Software for Semidefinite Programming. User's Guide, Beta Version.* Stanford University, October 1994. Available at `http://www-isl.stanford.edu/people/boyd`.

[7] S.-P. Wu and S. Boyd. SDPSOL: *A Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure. User's Guide, Version Beta.* Stanford University, June 1996.

[8] P. Gahinet and A. Nemirovskii. *LMI Lab: A Package for Manipulating and Solving LMIs*. INRIA, 1993.

[9] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[10] A. I. Lur'e and V. N. Postnikov. On the theory of stability of control systems. *Applied mathematics and mechanics*, 8(3), 1944. In Russian.

[11] N. N. Krasovskii. Application of Lyapunov's second method for equations with time delay. *Prikl. Mat. Mek.*, 20(3):315–327, 1956.

[12] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Trans. Aut. Control*, 42(6):819–830, June 1997.

[13] A. Hassibi and S. Boyd. Quadratic stabilization and control of piecewise-linear systems. In *Proc. American Control Conf.*, pages 3659–64, Philadelphia, PA, 1998.

[14] A. Hassibi, S. P. Boyd, and J. P. How. A class of Lyapunov functionals for analyzing hybrid dynamical systems. In *Proc. American Control Conf.*, San Diego, CA, 1999.