Supervisor Synthesis for Controller Upgrades

Johannes Kloos
MPI-SWS, Germany

Rupak Majumdar
MPI-SWS, Germany

***Abstract*—During the life cycle of a cyber-physical system, it is sometimes necessary to upgrade a working controller with a new, but unverified, one which provides better performance or additional functionality. To make sure that system invariants are not broken because of bugs in the new controller, an architecture is used in which both controllers are implemented on the platform, and a supervisor process checks that the actions of the new controller keep the system within its safe states. If an invariant may be violated, the supervisor switches over to the old controller that ensures correct behavior, but possibly degraded performance. A key question in the design of such supervisors is the switching strategy: when should the supervisor reinstate the new controller after it has switched to the old one? In general, one would prefer to use the new controller as much as possible, provided it does not violate safety. However, arbitrarily switching back to the new controller can cause the system to become unstable, even when each controller in isolation ensures stability. We provide a supervisor synthesis procedure that uses a simple counting strategy for the supervisor. Our synthesized supervisor ensures that switching between the controllers ensures stability of the system, while maintaining its safety properties and providing a lower bound on the use of the new controller. We prove the correctness of the strategy and show on an example that it can provide close to optimal use of the new controller against many disturbance scenarios.**

I. Introduction

Software controllers for physical systems lie at the core of many safety-critical systems. Such controllers are hard to design reliably, and are often the major portion of the development cost. Thus, once a controller has been validated (usually through extensive simulations as well as production performance), engineers are reluctant to change the code. Unfortunately, controllers may have to be updated as part of the design life cycle, e.g., to use a new implementation that has better performance, to accommodate a new architectural feature, or to use a new component that provides additional functionality [5], [8]. While the new controller may already be tested under nominal conditions to guarantee asymptotic stability, it may not be possible to verify its operation under all disturbance scenarios. Thus, it is often the case that while both controllers ensure stability of the plant on their own, under some disturbance scenarios, the new controller may produce trajectories that violate some additional safety properties of the system.

One way to make use of the new controller while ensuring safety is to keep both controllers around, and provide two modes of operations: regular operation using the new controller and degraded operation using the old controller. An obvious strategy is to switch from the new controller to the old controller the first time the new controller is about to violate the safety property, and from then on, use the old controller. However, this would mean that the system runs in degraded mode from this point on. A better strategy would be to switch back to the new controller when possible, so that the benefits of the new controller are available to the system. The problem of generating a switching strategy is not trivial: even if each controller ensures stability, it is well known that switching between them arbitrarily can cause the resulting system to be unstable (see, e.g., [12], as well as Figure 1 below). In this paper, we study the synthesis of supervisors that ensure stability, safety, as well as maximize the use of the new controller.

Let us be more precise. The input to the controller upgrade problem consists of two controllers C_{old} and C_{new} for a physical plant modeled as a continuous-time dynamical system, and an invariant I on the system states that must be maintained. Each controller, by itself, is guaranteed to ensure that the controlled system is stable (that is, converges to the reference behavior). Additionally, under the action of controller C_{old}, the system is guaranteed to ensure the invariant I. We want to design a supervisor that switches between the controllers to ensure that the switched control system is asymptotically stable and maintains invariant I, but also maximizes the long-run average use of C_{new}.

While there are general theoretical approaches to find optimal supervisors (e.g., using mean-payoff parity games [3], ensuring stability and safety using appropriate parity constraints [2]), they are at least as hard as the verification problem for controllers. Thus, these approaches may not be computationally feasible. Instead, we restrict attention to restricted classes of supervisor strategies and look for sufficient conditions.

In particular, we consider *counting* supervisors that do not depend on the exact sequence of controller actions and system states to determine when to switch, but only on the number of steps since the last switch. We show how to construct a counting strategy of the supervisor that ensures that the supervised system is globally asymptotically stable and does not violate the invariant, while giving a lower bound on the long-run average use of C_{new}. Our strategy is based on *average dwell time* arguments for switched control systems [12], [21]. We consider two versions of counting strategies: a *non-adaptive* version, which only counts the number of steps C_{old} is run since the last switch, and an *adaptive* version, which counts steps for both C_{old} and C_{new}. Both strategies

978-3-9815370-0-0/DATE13/©2013 EDAA

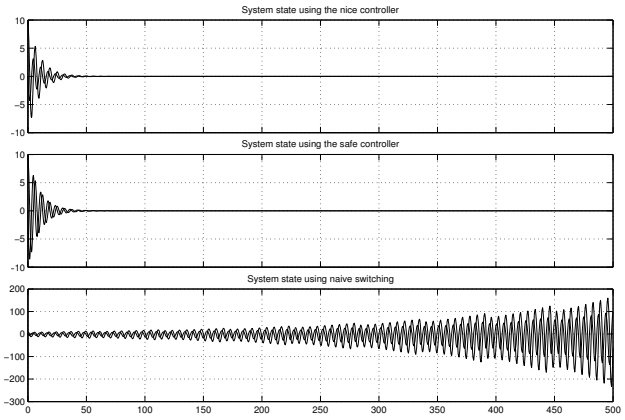

Fig. 1. Behavior of the running example: naive switching causes instability

are simple to implement using a finite state counter.

We have implemented the approach on top of Simulink, and we show experimentally on a standard benchmark example that it leads to almost optimal use of the new controller in a variety of settings, while maintaining asymptotic stability and the safety property by design. Further, we show that the simple counting strategies achieve performance close to manually constructed optimal strategies.

Related Work. Supervisor synthesis for discrete systems was studied extensively, both from a discrete event systems perspective [17], [10], as well as from an automata-theoretic synthesis perspective [16], [15], [20]. Control of timed and hybrid systems have been studied in various contexts [15], [9], [13], [19], although algorithmic results are usually intractable or undecidable. D'Souza et al. [6], [7] introduced the notion of conflict-tolerant features to allow synthesis of a supervisor that switches between different controllers while maximizing, in a certain sense, the use of each controller. These methods address invariance properties, but not global asymptotic stability.

The systems we consider are a special case of switching systems. It is well-known that introducing switching into a stable dynamical system can generate instability [12]. Early work in the analysis of switching systems considered the cases where the number of switches or the inter-switch time was bounded. We base our synthesis on a general result of Vu et al. [21] about input-to-state stability for general switching systems.

II. Preliminaries

We recall the background on dynamical systems, and use standard control-theoretic notation (see, e.g., [11]). For two vectors $v, w \in \mathbb{R}^n$, we write $v \cdot w$ for the standard scalar product. For a differentiable function f, we write ∇f for its gradient.

Dynamical Systems For the purpose of this paper, a *time-varying dynamical system with input or* general dynamical system is given by a differential equation $\dot{x} = f_{t_0}(t, x, u)$, where $f_{t_0} : \mathbb{R}_{\geq t_0} \times \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$. If $\dot{x} = f(x, u)$, the system is called *time-invariant*. If f is linear, the system is *linear*.

The *solution* of a differential equation for a general dynamical system is given by $\phi_{t_0} : \mathbb{R}_{\geq t_0} \times \mathbb{R}^N \times \mathcal{S} \to \mathbb{R}^N$, where $\mathcal{S} \subseteq (\mathbb{R}_{\geq t_0} \to \mathbb{R}^K)$ describes the set of *input signals*, containing the constant 0 signal, such that:

- $\phi(t_0, x_0, u) = x_0$ for all $u \in \mathcal{S}$,
- $\left(\frac{\partial}{\partial t}\phi\right)(t, x_0, u) = f(t, \phi(t, x_0, u), u(t))$ for $t \geq t_0$.

We assume the existence and uniqueness of solutions of each differential equation, this is guaranteed under standard assumptions on f [11]. For simplicity of notation, we assume that $t_0 = 0$, and omit all mentions of t_0.

Let $\mathcal{U} \subseteq \mathbb{R}^N$. A dynamical system is called *safe with respect to* $\mathcal{U}$ if its solution ϕ satisfies: if $x_0 \in \mathcal{U}$ then for all $t \geq 0$ and $u \in \mathcal{S}$, $\phi(t, x_0, u) \in \mathcal{U}$.

We will use the notion of stability as the fundamental correctness property of dynamical systems, in the form of input-to-state stability (ISS, in short). Let $\dot{x} = f(t, x, u)$ be a general dynamic system and ϕ its solution. Then the system is called *input-to-state stable* (*ISS*) if there is a $\mathcal{KL}$ function β and a $\mathcal{K}$ function γ such that for all $t \geq 0$, $x_0 \in \mathbb{R}^N$ and $u \in \mathcal{S}$, we have $|\phi(t, x_0, u)| \leq \beta(|x_0|, t) + \gamma(\sup_{0 \leq \tau \leq t} |u(\tau)|)$ (see [11], Definition 4.7, also for definitions of classes $\mathcal{K}$, $\mathcal{K}_\infty$, and $\mathcal{KL}$).

A general dynamical system is called *globally asymptotically stable* (*GAS*) if there is a $\mathcal{KL}$ function β such that $|\phi(t, x_0, 0)| \leq \beta(|x_0|, t)$. It can be given as the special case of ISS where $\mathcal{S} = \{0\}$. For linear systems, ISS and GAS are in fact equivalent ([11], Lemma 4.6).

A function $V : \mathbb{R}^N \to \mathbb{R}$ is called an *ISS Lyapunov function* for $f : \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ if there are $\mathcal{K}_\infty$ functions α_1, α_2 and $\mathcal{K}$ functions α_3, χ such that

1) $\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|)$ for all $x \in \mathbb{R}^N$,
2) $|x| \geq \chi(|u|) \implies (\nabla V)(x) \cdot f(x, u) \leq -\alpha_3(|x|)$ for all $x \in \mathbb{R}^N$, $u \in \mathbb{R}^K$.

A time-invariant dynamical system $\dot{x} = f(x, u)$ is ISS if and only if there exists an ISS Lyapunov function for f [18]. A similar approach is possible for GAS.

When modeling control problems, it is common to model the system in two parts, the *plant* and the *controller*. For controlled systems, the input to the system is split up into two parts, the controller input u and the disturbance input w. The system is then given via $\dot{x} = f(t, x, u, w), u = g(t, x)$, where $f : \mathbb{R}^+ \times \mathbb{R}^N \times \mathbb{R}^K \times \mathbb{R}^L \to \mathbb{R}^N$ models the behavior of the plant and $g : \mathbb{R}^+ \times \mathbb{R}^N \to \mathbb{R}^K$ models the controller. The dynamical system defined by $\dot{x} = f(t, x, u, w)$ is called the *open-loop system*.

The general dynamical system given by $\dot{x} = f(t, x, g(t, x), w)$ is called the *closed-loop system*. Note that for control systems, it is often useful to extend the definition of safety such that $\mathcal{U} \subseteq \mathbb{R}^N \times \mathbb{R}^K$, and demand that $(\phi(t, x_0, w), g(t, \phi(t, x_0, w))) \in \mathcal{U}$.

Switching Sampled-data Systems Given an index set $\mathcal{P}$, a family $f_p : \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ for $p \in \mathcal{P}$ that are locally Lipschitz in both arguments, and a function $\sigma : \mathbb{R}^+ \to \mathcal{P}$, the

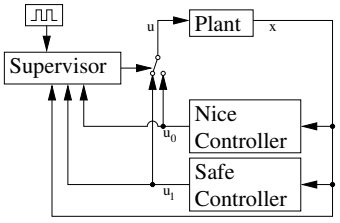

Fig. 2. System structure

switched system $\dot{x} = f_\sigma(x,u)$ is defined as $\dot{x} = f_{\sigma(t)}(x,u)$ (compare [21], Section 2). Similarly, one may define a switched control system by fixing a plant model $f : \mathbb{R}^N \times \mathbb{R}^K \times \mathbb{R}^L \to \mathbb{R}^n$ and a family of controllers $g_p : \mathbb{R}^N \to \mathbb{R}^K$ for $p \in \mathcal{P}$, with the dynamics $\dot{x}(t) = f(x, g_{\sigma(t)}(x(t)), w(t))$ (closed-loop system for a switched control system). By abuse of notation, we write $\dot{x} = f(x,u,w), u = g_\sigma(x)$.

Finally, a *sampled-data system* (see [4]) consists of a controlled dynamical system $\dot{x} = f(x,u,w)$ and a discrete controller, given as a function $g : \mathcal{S} \times \mathbb{R}^N \to \mathcal{S} \times \mathbb{R}^K$. Let $\tau_s > 0$ denote a sample time. Then the function $S : (\mathbb{R}^+ \to \mathbb{R}^N) \to (\mathbb{N} \to \mathbb{R}^N)$ defined as $S(x)(k) = x(k\tau_s)$ describes the *ideal sampler*, and $H : (\mathbb{N} \to \mathbb{R}^N) \to (\mathbb{R}^+ \to \mathbb{R}^N)$ the *zero-order hold*, $H(x)(t) = x(k)$ for $k \in \mathbb{N}$ such that $k\tau_s \le x < (k+1)\tau_s$. Then the sampled-data system denoted as $\dot{x} = f(x,u,w), u = g(x)$ is given by the controlled dynamical system $\dot{x} = f(x,u,w), u = H \circ g \circ S$.

For all these classes of systems, the notions of safety, GAS, ISS and Lyapunov function carry over from the case of general dynamical systems.

Stability of Switched Systems One common approach to guarantee stability for switched systems uses the notion of average dwell-time. A switching signal σ has *average dwell-time* τ_a if there exists an $N_\circ > 0$ such that $N_\sigma(t_1,t_2) \le N_\circ + \frac{t_2 - t_1}{\tau_a}$ for all $0 \le t_1 \le t_2$, where $N_\sigma(t_1,t_2)$ gives the number of switches in the interval $[t_1,t_2]$. The following result from [21] uses average dwell time to prove stability.

Theorem 1: ([21], Thm. 3.1) Consider the switched system $\dot{x} = f_\sigma(x,u)$ with $\mathcal{P}$ finite. Suppose that for every $p \in \mathcal{P}$, there exist positive definite functions V_p, as well as a class $\mathcal{K}_\infty$ function γ and numbers $\lambda_\circ > 0, \mu \ge 1$ such that for all $\xi \in \mathbb{R}^N, \eta \in \mathbb{R}^K$ and $p \in \mathcal{P}$, we have

- $\nabla V_p(\xi) \cdot f_p(\xi,\eta) \le -\lambda_\circ V_p(\xi) + \gamma(|\eta|)$,
- *For all $q \in \mathcal{P}$, $V_p(\xi) \le \mu V_q(\xi)$.*

Let σ be a switching signal having average dwell-time τ_a. If $\tau_a > (\ln \mu)/\lambda_\circ$, the switched system is input-to-state stable.

III. Problem Definition

Fig. 2 shows a full system consisting of the plant $\dot{x} = f(x,u)$ with $x \in \mathbb{R}^N$, $u \in \mathbb{R}^K$, two continuous-time controllers $g_j : \mathbb{R}^N \to \mathbb{R}^K$ for $j = 0,1$, and a sampled-time supervisor $g_s : \mathbb{R}^N \times \mathbb{R}^K \times \mathbb{R}^K \to \{0,1\}$ that selects which controller to use. The system evolves according to

$$\dot{x} = f(x,u_j) \quad u_0 = g_0(x) \quad u_1 = g_1(x) \quad (1)$$
$$j^+ = g_s(x,u_0,u_1,j)$$

where j^+ denotes the value of j in the next sample point. Fix the sample time to be τ_s, that is, the supervisor can switch between controllers at integer multiples of τ_s.

We assume that each controller g_0, g_1 ensures the closed loop system is ISS. Additionally, we assume that the closed loop system with controller g_0 is safe with respect to a safety property $\mathcal{U}$. We call g_0 the *safe* controller. We call g_1 the *nice* controller. In the following, let $\mathcal{P} = \{0,1\}$, where 0 corresponds to the safe controller, and 1 to the nice controller.

The switched system and the supervisor together define a switching sequence $\sigma(t)$. For a switching sequence $\sigma(t)$, define the *utility function* as

$$u(\sigma) = \liminf_{T\to\infty} \frac{1}{T} \int_0^T \sigma(t)dt$$

Intuitively, the utility describes the fraction of time the nice controller is in use.

The *supervisor synthesis problem* asks to design the supervisor g_s such that the closed loop system (1) is globally asymptotically stable (for systems without input) or input-to-state stable (for systems with input) and satisfies the safety property $\mathcal{U}$, and moreover, the utility $u(\sigma)$ is maximal.

As shown in [13], it is possible to generate g_s effectively so that the property $\mathcal{U}$ is guaranteed to always hold in the resulting switched system. Furthermore, the supervisor may be generated in such a way that $j^+ = 1$ is chosen if doing so will not, within a certain time period, allow the trajectory of the switched system to violate $\mathcal{U}$. This construction does not consider stability requirements, and may, in fact, lead to situations similar to that in Fig. 1, as will be demonstrated momentarily.

As a running example, consider the following system, with $x, u \in \mathbb{R}^2$:

$$\dot{x} = f(x,u) = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x + u$$
$$g_0(x) = -\frac{1}{100}x \quad g_1(x) = \begin{pmatrix} -\frac{1}{100} & 1 \\ 1/2 & -\frac{1}{100} \end{pmatrix} x.$$

We find that the closed-loop systems with both controllers are stable with Lyapunov functions $V_0(x) = x_1^2 + x_2^2$ and $V_1(x) = x_1^2 + 4x_2^2$, respectively.

Now, let $\tau_s = \pi/4$ and assume a safety constraint that states that when $x_2 = 0$ and $u = (u^{(1)}, u^{(2)})$, $u^{(2)} \ge 0$ must hold. This safety constraint is equivalent to forcing a switch to the safe controller whenever the state crosses the positive x axis during the next time step (i.e., a switch is forced at time $n\tau_s$ when the trajectory would cross the positive x axis at a time between $n\tau_s$ and $(n+1)\tau_s$). The corresponding set of states in which a switch to the safe controller is forced can be explicitly calculated as $\{x \mid x_1 \ge 0, 2x_1 \ge x_2 \ge 0\}$ using the time discretization of the system with step size τ_s. Simulating this system shows that it fails to stabilize, as shown in Fig. 1.

IV. Switching Strategies

A. Average Switching Time Game

One solution to ensure safety and stability is to modify the supervisor so that average dwell times are guaranteed. For

the running example, one finds that an average dwell time τ_a larger than roughly $50 \ln 2 \approx 34.657$ will do, using Theorem 1. Since switching occurs only at time quanta, it makes sense to consider τ_a/τ_s instead, which works out to $\tau_a/\tau_s > 4 \cdot 50(\ln 2)/\pi \approx 44.127$. In the following, $\tau_a/\tau_s = 45$ is assumed for this example.

Since the switching time from the nice controller to the safe controller is forced upon the supervisor, while the converse direction is decided by the supervisor, it is natural to represent the choice of switching times as an infinite game. For the description of the game, the following definition is needed:

Definition 1: Let $\mathbf{a}$ and $\mathbf{s}$ be sequences of natural numbers that have the same length (finite or infinite).

1) The sequence $\mathbf{r}_{\mathbf{a},\mathbf{s}}$ is given by $(\mathbf{r}_{\mathbf{a},\mathbf{s}})_n := \sum_{i=1}^{n-1} \mathbf{a}_i + \mathbf{s}_i$. If $\mathbf{a}$ and $\mathbf{s}$ are clear from the context, $\mathbf{r}$ is used instead of $\mathbf{r}_{\mathbf{a},\mathbf{s}}$.
2) The *induced switch sequence* $\sigma_{\mathbf{a},\mathbf{s}}$ is given by

$$\sigma_{\mathbf{a},\mathbf{s}}(t) = \begin{cases} 0 & \exists i \in \mathbb{N} : t/\tau_s \in [(\mathbf{r}_{\mathbf{a},\mathbf{s}})_i + \mathbf{a}_i, (\mathbf{r}_{\mathbf{a},\mathbf{s}})_{i+1}) \\ 1 & \text{otherwise} \end{cases}$$

The *average switching time game* is an infinite game with two players, the supervisor and the adversary, who choose positive natural numbers denoting time points to switch from the nice controller to the safe one (for the adversary), and points to switch back (for the supervisor). Two values are given initially, namely $\tau_a > 0$ and a natural number $N_\circ \geq 1$. The states of the game are pairs of finite sequences $(\mathbf{a}, \mathbf{s})$, indicating the history of switching intervals by the adversary ($\mathbf{a}$) and the supervisor ($\mathbf{s}$). The initial state is $([], [])$. When $|\mathbf{a}| = |\mathbf{s}|$, the adversary picks a new point $k > 0$ and the new state is $(\mathbf{a} \cdot k, \mathbf{s})$. Then, the supervisor picks $k' > 0$, and the new state is $(\mathbf{a} \cdot k, \mathbf{s} \cdot k')$, where $\cdot$ denotes appending an element to a sequence. The winning condition $\mathcal{W}$ for the supervisor is given as the set of all (finite and infinite) sequences that have average dwell-time τ_a.

It is easy to show that there is a winning strategy of the supervisor, which waits "long enough" in the old controller. This strategy can even ignore the moves of the adversary.

Proposition 1: Given a safe and a nice controller that are both ISS, the strategy choosing $\mathbf{s}_i = 2\left\lceil \frac{\tau_a}{\tau_s} \right\rceil - 1$ for all $i \in \mathbb{N}$ is a winning strategy for $N_\circ \geq 2$.

Lemma 1: Let a nice and a safe controller be given that are both ISS. If the switching intervals chosen by the adversary have a finite average, i.e., $\overline{\mathbf{a}} = \lim_{n\to\infty} \sum_{i=1}^{n} \frac{\mathbf{a}_i}{n}$ exists, the maximal achievable utility is bounded: For every supervisor-chosen switching interval sequence $\mathbf{s}$, $u(\sigma_{\mathbf{a},\mathbf{s}}) \leq \frac{\overline{\mathbf{a}}\tau_s}{2\tau_a}$.
In case the adversary always picks a constant $k \in \mathbb{N}$, there is a supervisor strategy that bounds the utility by $\frac{k\tau_s}{\max\{2\tau_a, (k+1)\tau_s\}}$. Note that, in particular, for $k = 1$ and $\tau_a/\tau_s \in \mathbb{N}$, the above strategy achieves the optimal utility.

B. An Adaptive Strategy

Proposition 1 shows the supervisor can win *blindly*, i.e., by ignoring the adversary's moves. Such a strategy can be conservative. We now give an adaptive winning strategy, that considers the sequence $\mathbf{a}$.

Consider the strategy that, in the n-th step, chooses

$$\mathbf{s}_n = \min\{t \in \mathbb{N} | t \geq 1, \sigma_{[\mathbf{a}_1,\ldots,\mathbf{a}_n,1],[\mathbf{s}_1,\ldots,\mathbf{s}_{n-1},t,R]} \text{ has average dwell-time } \tau_a\}$$

for a given $N_\circ$ and R as above. Call this strategy the *online strategy*.

Proposition 2: Given a safe and a nice controller that are both ISS, the optimistic online strategy is well-defined and a winning strategy for $N_\circ \geq 2$.

Note that this strategy can be given in an equivalent, efficiently computable, form as follows. Define $\rho := \frac{\tau_a}{\tau_s}, c := (1 - N_\circ)\rho + \max\{0, \rho - e\}$, and assume that $N_\circ \geq 2$.

Theorem 2: Given an adversary sequence $\mathbf{a}$*, construct two sequences* $\mathbf{u}, \mathbf{v}$ *as follows:* $\mathbf{u}_0 = 0$*,* $\mathbf{u}_n = \max\{\mathbf{u}_{n-1} + 2\rho - \mathbf{a}_n - \mathbf{v}_{n-1}, \rho + c\}$*,* $\mathbf{v}_n = \max\{1, \lceil \mathbf{u}_n \rceil\}$*. Then* $\mathbf{v} = \mathbf{s}$*, where* $\mathbf{s}$ *is generated by the on-line strategy.*

Note that each $\mathbf{u}_n$ is bounded by $R = 2\lceil \frac{\tau_a}{\tau_s} \rceil - 1$ as in Proposition 1. This implies the following corollary.

Corollary 1: The on-line strategy can be implemented using finite memory, and each step runs in constant time.

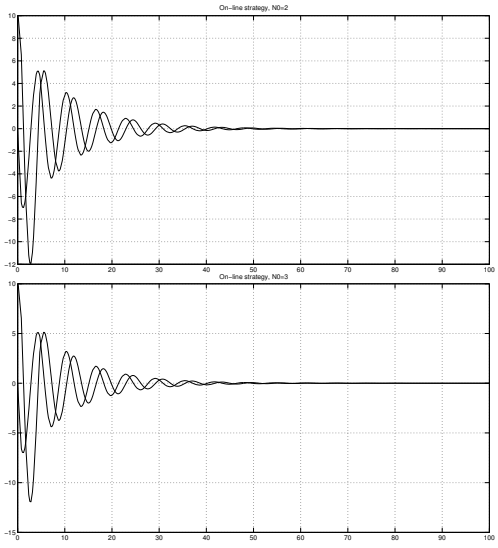

Fig. 3. Globally asymptotically stable switching for the running example: Example runs

V. Experimental Analysis

In this section, the switching technique is applied to two systems, the running example described above, and a bicycle steering controller. Both systems were implemented in Matlab/Simulink for experimentation.

A. The running example

For the running example, Fig. 1 shows that each controller by itself achieves global asymptotic stability. Moreover, the safe controller achieves system safety by itself but the nice controller may be unsafe. Naive switching can ensure safety but can make the system unstable. Recall that the conditions

from Theorem 1 imply that $\rho \geq 45$ guarantees global asymptotic stability of the switched system. The results using the online strategy are visualized in Fig. 3. Switching without average dwell-time guarantee technically achieves safety (for the given safety condition). Periodic switching occurs, with a utility value of 0.754. However, the resulting system is *not stable*. Switching with the on-line strategy for $N_\circ = 2$ and $\rho = 45$ achieves safety, global asymptotic stability, and periodic switching, with a utility value of 0.06283.

Is the achieved utility value acceptable, especially since the (unstable) switching strategy that only considers safety achieves a utility that is larger by an order of magnitude? We consider the following aspect of the question: Among switching strategies that have the same average dwell time guarantee, how much improvement is possible? In this specific example, it is possible to answer the question precisely by making use of the exact knowledge of the solution of the dynamical system. For this purpose, it is important to note that the structure of the system implies that if the system runs 7 steps with the nice controller, it will definitely be forced to switch to the safe controller. This can be seen from the discretization of the system.

To answer the first question, one may find an upper bound for $u(\sigma)$ using Lemma 1: $u(\sigma) \leq \frac{\bar{\mathbf{a}}\tau_a}{2\tau_s} = \frac{\bar{\mathbf{a}}}{2\rho} < \frac{7}{2\cdot 45} = 0.0\overline{7}$. Thus, in absolute terms, the utility achieved by the on-line strategy cannot be increased by a significant amount for the given value of ρ. In relative terms, the on-line strategy achieves about 80% of the theoretical limit for the utility value with the given average dwell time bound.

B. The bicycle steering controller

The second example concerns the control of a bicycle. The example is taken from [1] and the controllers come from [14]. The second reference provides two controllers, a LQR (Linear Quadratic Regulator) controller and a fixed-point controller. In practice, due to quantization error and related problems, the LQR controller will not control the system as precisely as the fixed-point controller, while at the same time requiring more involved calculations. Therefore, the fixed-point controller is to be used preferentially to the LQR controller.

The plant model is given as

$$\dot{x} = \begin{pmatrix} 0 & \frac{g}{h} \\ 1 & 0 \end{pmatrix} x + \begin{pmatrix} u + w \\ 0 \end{pmatrix}$$

where $g = 9.81$ and $h = 1.5$ (SI units). The LQR controller is given by $g_1(x) = -5.2236x_1 - 13.1428x_2$, and the fixed-point controller by $g_2(x) = -10x_1 - 16x_2$.

The safety condition chosen for this example is that the torque that can be applied to the system is limited, in particular, $|u| \leq 100$ (i.e., $\mathcal{U} = \{(x, u)||u| \leq 100\}$. Note that in this case, the extended notion of safety, including the controller inputs in the state, is used. An easy calculation shows that, with regard to this condition, the LQR controller is safe in a larger set of states than the fixed-point controller. Therefore, we implement both controllers and a switching supervisor to make use of the larger safe set. The stabilization goal is to ensure that the system still stabilizes in the presence of small disturbances. This can be formulated as an ISS problem: The system should be input-to-state stable, where the disturbances are considered as the inputs.

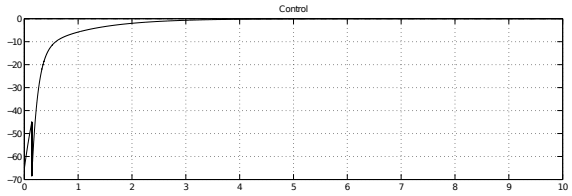

Fig. 4. Behavior of the control values in the bicycle model: undisturbed case. Only one switching event occurred; it corresponds to the point in time where the control value has a strong negative spike. The state of the plant has been omitted for brevity.

Choosing a sample time $\tau_s = 1ms$, the set of states in which the system could be safely run using the fixed-point controller was under-approximated using a simulation-based approach, with limited disturbances with $\|w\|_\infty \leq 10$. The set of states that will not evolve, within one millisecond, to an unsafe state, is given by $\Sigma := \{x||x_1 + 0.625x_2 + 0.625| \leq 5.62\}$. Using Matlab, the Lyapunov functions for the closed-loop systems were calculated as

$$V_0(x) = x^T \cdot \begin{pmatrix} 0.7277 & -0.5 \\ -0.6 & 0.5058 \end{pmatrix} x$$

$$V_1(x) = x^T \cdot \begin{pmatrix} 0.523 & -0.5 \\ -0.6 & 0.4838 \end{pmatrix} x$$

Thus, $\lambda_\circ \approx 17.8516$ and $\mu \approx 2.60268$, so $\tau_a > (\ln \mu)/\lambda_\circ \approx 0.026739$. We chose $\tau_a := 0.027$, so $\rho = \tau_a/\tau_s = 27$. The online strategy with $N_\circ = 2$ achieves safety and stabilization.

Preliminary experimentation showed that the undisturbed system (i.e., the case $w = 0$) would quickly enter a state where the nice controller could be used perpetually (Fig. 4). So, we considered the system with disturbance inputs that would cause periodic switching. We fixed the disturbance signal to be:

$$w(t) = \begin{cases} -20x_1 & \exists n \in \mathbb{N} : nT \leq t \leq (n+b)T \\ 0 & \text{otherwise} \end{cases}$$

with parameters b (duty cycle of the disturbance) and T (period of the disturbance). The result for following parameter combinations for b and T are reported below:

(1) $b = 5\%$, $T = 0.2$, (2) $b = 30\%$, $T = 0.2$
(3) $b = 5\%$, $T = 0.5$, (4) $b = 5\%$, $T = 1$.

These different values were chosen to detect the influence of different time-scales and disturbance shapes on the system. Combining Theorem 1 with the fact that a Lyapunov function for a linear system is also an ISS Lyapunov function allows us to conclude that the same average dwell time bound applied in this case as well. In each experiment, the utility function and the fraction of time where the system was outside Σ were calculated, and it was recorded whether any safety conditions were violated.

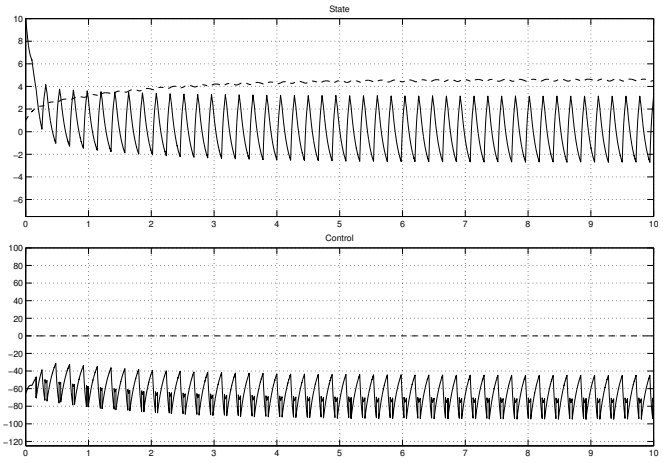

Fig. 5. Evolution of the bicycle example with disturbance: interesting case ($T = 0.2$, $b = 5\%$)

T	b	Utility	Time fraction in Σ	Efficiency
0.2	5%	82.27%	82.4%	99.85%
0.2	30%	82.27%	82.4%	99.85%
0.5	5%	71.07%	71.07%	99.99%
1	5%	71.4%	71.4%	99.99%

TABLE I
MEASUREMENT RESULTS FOR THE BICYCLE EXAMPLE – INTERESTING CASES

A typical example of system behavior with the listed disturbances is shown in Fig. 5. In none of the experiments was any safety violation detected. The other measurements are listed in Table I. The utility column gives the measured utility value, while "Time fraction in Σ" gives the fraction of time where the system state was inside Σ, i.e., the amount of time that the nice controller could be safely used. Finally, the efficiency column gives the ratio between those columns in percent.

To estimate the maximal achievable utility value, we note that the fraction of time spent in Σ forms a crude upper bound: the system will only use the nice controller when the system state is in Σ or has just left it, and return to using the nice controller when the system state has re-entered Σ. Thus, the ratio between measured utility value and the fraction of time spent in Σ gives an estimate of the quality of the switching strategy in terms of utilization of the nice controller.

As can be seen from Table I, for the example, the achieved efficiency is very high (greater than 99% in all cases). Furthermore, in all cases where the system eventually settled on using the nice controller, efficiency is actually 100%. Therefore, the on-line strategy is highly efficient in this example.

That last observation motivates some further analysis: The efficiency of the switching strategy goes up when the difference between time scales for disturbances and dwell-time goes up. For disturbances that occur on a time-scale that is significantly larger than ρ, the switching utility is roughly the same as the fraction of time where the safety condition does not enforce a switch to the safe controller, which forms a trivial lower bound for the utility value. Also, very small values of τ_a allow for practically trivial stabilization with switching and actually occur in practice. Furthermore, if $\tau_a \ll \tau_s$, naive switching still gives ISS! This allows for a trade-off with regard to choosing the sample time: a smaller τ_s allows a more precise under-approximation of the strongly safe state set, but requires more CPU power and slightly more effort for stabilization.

We make the following conclusions. First, the online strategy works well in stabilizing a practical system while ensuring safety and high utility. Second, the utility value is close to the achievable maximum when average dwell time is small compared to either the time-scale on which disturbances appear or the sample time.

REFERENCES

[1] K. Astrom and R. Murray. *Feedback systems an introduction for scientists and engineers*. Princeton University Press, 2008.

[2] P. Bouyer, T. Brihaye, M. Jurdziński, R. Lazić, and M. Rutkowski. Average-Price and Reachability-Price Games on Hybrid Automata with Strong Resets. In *FORMATS '08*, LNCS 5215, pages 63–77. Springer, 2008.

[3] K. Chatterjee, T. Henzinger, and M. Jurdzinski. Mean-Payoff Parity Games. In *LICS '05*, pages 178–187. IEEE, 2005.

[4] T. Chen and B. A. Francis. *Optimal sampled-data control systems*. Springer, 1995.

[5] U. Drolia, Z. Wang, Y. Pant, and R. Mangharam. AutoPlug: An Automotive Test-bed for Electronic Controller Unit Testing and Verification. In *IPSN '11*. IEEE, 2011.

[6] D. D'Souza and M. Gopinathan. Conflict-Tolerant Features. In *CAV '08*, LNCS, pages 227–239. Springer, 2008.

[7] D. D'Souza, M. Gopinathan, S. Ramesh, and P. Sampath. Supervisory control for real-time systems based on conflict-tolerant controllers. In *CASE '09*, pages 555–560. IEEE, 2009.

[8] K. Heckemann, M. Gesell, T. Pfister, K. Berns, K. Schneider, and M. Trapp. Safe automotive software. In *KES (4)*, LNCS 6884, pages 167–176. Springer, 2011.

[9] T. Henzinger, B. Horowitz, and R. Majumdar. Rectangular Hybrid Games. In *CONCUR'99 Concurrency Theory*, LNCS 1664, pages 320–335. Springer, 1999.

[10] R. Hill, J. Cury, M. de Queiroz, D. Tilbury, and S. Lafortune. Multi-level hierarchical interface-based supervisory control. *Automatica*, 46(7):1152–1164, 2010.

[11] H. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[12] D. Liberzon. *Switching in Systems and Control*. Birkhäuser, 2003.

[13] P. Mahdavinezhad, P. Gohari, and A. G. Aghdam. Supervisory Control of Discrete-Event Systems with Output: Application to Hybrid Systems. In *ACC '07*, pages 4291–4296. IEEE, 2007.

[14] R. Majumdar, I. Saha, and M. Zamani. Synthesis of Minimal-Error Control Software. In *EMSOFT '12*. ACM, 2012.

[15] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In *STACS '95*, LNCS 900, pages 229–242. Springer, 1995.

[16] A. Pnueli and R. Rosner. On the synthesis of an asynchronous reactive module. In *ICALP '89*, LNCS 372, pages 652–671. Springer, 1989.

[17] P. J. Ramadge and W. M. Wonham. Supervisory Control of a Class of Discrete Event Processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

[18] E. Sontag and Y. Wang. On characterizations of the input-to-state stability property. *Systems and Control Letters*, 24(5):351–359, 1995.

[19] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2010.

[20] W. Thomas. On the synthesis of strategies in infinite games. In *STACS '95*, LNCS 900, pages 1–13. Springer, 1995.

[21] L. Vu, D. Chatterjee, and D. Liberzon. Input-to-state stability of switched systems and switching adaptive control. *Automatica*, 43(4):639–646, 2007.