Formal Verification of Weakly-Hard Systems

Chao Huang
Northwestern University
Evanston, Illinois
chao.huang@northwestern.edu

Wenchao Li
Boston University
Boston, Massachusetts
wenchao@bu.edu

Qi Zhu
Northwestern University
Evanston, Illinois
qzhu@northwestern.edu

ABSTRACT

Weakly-hard systems are real-time systems that can tolerate occasional deadline misses in a bounded manner. Compared with traditional systems with hard deadline constraints, they provide more scheduling flexibility, and thus expand the design space for system configuration and reconfiguration. A key question for such a system is precisely to what degree it can tolerate deadline misses while still meeting its functional requirements. In this paper, we provide a formal treatment to the verification problem of a general class of weakly-hard systems. We discuss relaxation and over-approximation techniques for managing the complexity of reachability analysis, and develop algorithms based upon these for verifying the safety of weakly-hard systems. Experiments demonstrate the effectiveness of our approach in understanding the impact of and guiding the selection among different weakly-hard constraints.

CCS CONCEPTS

• **Computer systems organization** → **Real-time systems**; *Embedded systems*; • **Software and its engineering** → *Formal methods*;

KEYWORDS

Weakly-hard, Formal Verification, Safety

1 INTRODUCTION

Timing constraints are critical for real-time systems. However, it has been observed that the traditional hard real-time constraints can be too rigid, and can lead to over-provisioning of system resources or inability to cope with deadline misses that may arise in practical settings [9]. An example of this is the retrofitting automotive electronic system for security [10, 52]. Recent studies showed that the hard real-time model is incompatible with even lightweight defense mechanisms due to timing violations caused by the additional overhead [28].

Weakly-hard systems employ a timing model that aims at relaxing the constraints of a real-time system by allowing certain degrees of deadline misses. A common example of the weakly-hard model is the (m,K) constraint, which specifies that among any K consecutive task activations, at most m instances could violate their execution deadlines [6, 36][1]. Such weakly-hard constraints can enhance the system's capability to accommodate timing violations. Compared with traditional hard deadlines, they significantly expand the feasible system configuration space, and provide more scheduling slack and flexibility; while compared with soft deadlines, they can provide deterministic guarantees on system safety, stability, performance, data quality and other properties.

[1]Hard constraints can viewed as a special case of weakly-hard constraints, with $m = 0$, while soft constraints can be viewed as a special case with no restriction on m.

Certain types of systems also fit naturally with a weakly-hard model. In the Internet-of-Things (IoT) space, transiently-powered devices are embedded systems that operate based on unreliable energy sources such as energy harvesters [5, 44]. These systems exhibit an intermittent pattern of computation where a computation may need to be suspended until sufficient energy is gathered (and thus potentially violating its deadline) [4, 14]. Another class is multi-rate distributed systems that are sensitive to message losses. For instance, a sensor might be sending messages at a faster rate than the receiving controller can process them [27]. Such message losses can be translated to deadline misses under a weakly-hard model.

The central question for weakly-hard systems is *precisely to what degree the systems can tolerate such deadline misses*. The well-known (m,K) model specifies a high-level constraint on the frequency of deadline misses. However, further verification is needed to ensure that such timing violations do not impact the system's functionality and safety. In this paper, we focus on the safety verification problem of nonlinear weakly-hard systems. We show that such a system can be naturally modeled as a hybrid automaton that captures the discrete nature of a sampled-data system, the system's dynamics, as well as the nondeterminism in deadline misses. Existing verification techniques for hybrid systems, however, cannot be directly applied to this model. There are three main difficulties: (1) uncertainty of when deadline misses occur, (2) exact state being unknown at each sampling instant, and (3) infinite number of discrete and continuous transitions. To address these difficulties, we relax the original problem through three steps, namely *split*, *localize* and *classify* (details in Section 4), which result in a finite-time problem of *local safety* and *inductiveness*. We show that these are sufficient conditions for the original safety problem and thus guarantee soundness of our approach. By separately considering local safety and inductiveness based on over-approximation techniques, we can compute the safe region of the system at the typical worst-case response time (TWCRT) W. The safe initial region can then be computed using region of attraction (ROA) techniques [20, 23, 45, 46]. Thus, the safety of the system under a given initial state can be checked by simply testing if it belongs to the safe initial region. Prior works on safety verification of weakly hard systems are limited to linear/piecewise affine dynamical systems [15, 16]. To the best of our knowledge, this is the first work that addresses the infinite-time verification problem of weakly-hard systems with nonlinear dynamics.

Our paper makes the following contributions.

- We propose a relaxation approach to convert the infinite-time verification problem of weakly hard systems with nonlinear dynamic into a finite-time one of local safety and inductiveness.
- We develop an over-approximation based technique to analyze the local safety and inductiveness.

- We demonstrate the effectiveness of our approach on multiple examples with linear or nonlinear dynamics.

The rest of the paper is structured as follows. Section 2 gives an overview of related work. Section 3 introduces the system model and the verification problem. Section 4 and Section 5 describe the relaxation techniques and the over-approximation based analysis approach respectively. Section 6 presents experimental results. We conclude and lay out future directions in Section 7.

2 RELATED WORK

Weakly-hard constraints: The notion of (m,K) constraints was first introduced in [18]. The authors in [6] formally defined weakly-hard constraints for real-time systems and presented schedulability analysis for periodic tasks under fixed-priority scheduling. Other schedulability analysis works [7, 26, 43] were presented under various assumptions such as bi-modal execution and non-preemptiveness. Weakly-hard constraints were also studied to bound the temporal behavior of overloaded systems [2, 19, 36, 48], where typical worst-case analysis (TWCA) is conducted for tasks that are activated periodically with sporadic overload. Recently, a job-level scheduling policy was presented for weakly-hard constraints to improve system schedulability [12].

Besides schedulability analysis, analyzing and optimizing control stability is another topic studied with weakly-hard constraints. In [37], periodic task instances are statically separated into mandatory and optional instances based on the (m,K) constraints, and only the mandatory ones are guaranteed to complete in time. The work in [17] extends this work to improve the performance of optional instances, and the work in [31] considers additional non-periodic execution. In [34], a general state-based weakly-hard model is proposed to measure the performance cost of deadline misses. In [30], a deadline miss is modeled as a probabilistic event. Several approaches have also been proposed for control-schedule co-design under possible deadline misses [8, 13, 42]. The work in [22] proposes to leverage weakly-hard constraints for modeling disturbances in networked systems and analzying system properties.

Only a few prior works have studied the safety verification problem of weakly-hard systems [15, 16]. Both of these rely on using satisfiability modulo theories (SMT) solvers and can only handle linear/piecewise affine dynamical systems.

Sampled-data systems: Sampled-data systems [11] are simple version weakly-hard systems with no deadline misses. Most of the current works focus on the stability analysis of different dynamical systems. Linear dynamical systems were considered in [24, 39], while nonlinear dynamical systems were considered in [32, 33]. In recent years, sampled-data systems with control inputs missing have attracted much interests [25, 50]. These works share the same pre-condition that when an input misses for some reason, the system will apply *zero* input, that is, the system runs in the open loop. Different from these works, we consider weakly-hard systems where the input in the last sampling period will be applied when the current execution deadline is not met.

Hybrid systems: Hybrid systems [3] are dynamical systems with both continuous evolution and discrete jumps. They are suitable for modeling weakly-hard systems which are sampled-data systems with a bounded degree of nondeterminism on deadline misses. Researchers of hybrid systems mainly focus on reachability, that is, compute the set of states that a hybrid system can reach in finite or infinite time. For instance, deterministic hybrid systems are studied in [35, 47, 49], while stochastic hybrid system are studied in [1, 21, 41]. However, due to the sampled-data nature of weakly-hard systems, existing analysis approaches of hybrid systems cannot be directly applied to weakly-hard systems.

To the best of our knowledge, this is the first work that addresses the infinite-time verification problem of weakly-hard systems with nonlinear dynamics.

3 MODEL WEAKLY-HARD SYSTEMS

Figure 1 shows a typical weakly-hard system, where $x \in \mathcal{R}^n$ is the state variable, $u \in \mathcal{R}^m$ is the input variable. The state equation is

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where $f : \mathcal{R}^n \to \mathcal{R}^n$ and $g : \mathcal{R}^n \to \mathcal{R}^{n\times m}$ are polynomial functions. For any $t \geq 0$, let $i = \lfloor (t-W)/T \rfloor$,

$$u(t) = u_i = \begin{cases} 0, & i < 0,\ t \in [0, W] \\ \pi(x(iT)), & i \geq 0,\ t \in [iT+W, (i+1)T+W] \end{cases} \tag{2}$$

where $\pi(x)$ is Lipschitz continuous with Lipschitz constant c_L, that is, there exists $c_L \in \mathcal{R}^+$ such that

$$\forall x_1, x_2 \in \mathcal{R}^n, \quad \|\pi(x_1) - \pi(x_2)\| \leq c_L \|x_1 - x_2\|.$$

Remark 1. *If π is linear, its linear coefficient is a Lipschitz constant. For general nonlinear π, it needs a case-by-case analysis to obtain c_L. For instance, recent work showed that a function represented as a neural network with various types of activation functions is Lipschitz continuous and the corresponding Lipschitz constant can be calculated accordingly [38].*

In this paper, we consider weakly-hard systems that are defined by (m,K) constraint:

Definition 3.1. An (m,K) weakly-hard system is a sampled-data real-time system, where there are at most m deadline misses among any K consecutive executions.

The (m,K) weakly-hard system in Figure 1 runs as follows. At the instant $t = 0$, the system applies the input 0 as a default input for the typical worst-case response time (TWCRT) W. Meanwhile, the system samples the current state and tries to compute the input function u_1. The computation time does not exceed the TWCRT, therefore the system can apply u_1 from $t = W$ to $T + W$. The same procedure repeats at the next sampling instant $t = T$. This time the computation does not meet the deadline, therefore the system is not able to obtain the newest input u_2 and have to use the last input u_1. The above procedure will continue for infinite time.

We assume that the equilibrium point of the dynamical system (1) is known *a priori*. Without loss of generality, let the origin be the equilibrium, that is, $f(0) = 0$. We assume that the state space of the system is $\mathcal{R}^n$, and the safe region of the system is

$$\mathcal{X}_s = B(d), \tag{3}$$

where $B(r) \triangleq \{x | \|x\| \leq r\}$ representing a ball around the origin with radius r, and d represents the safety distance threshold from

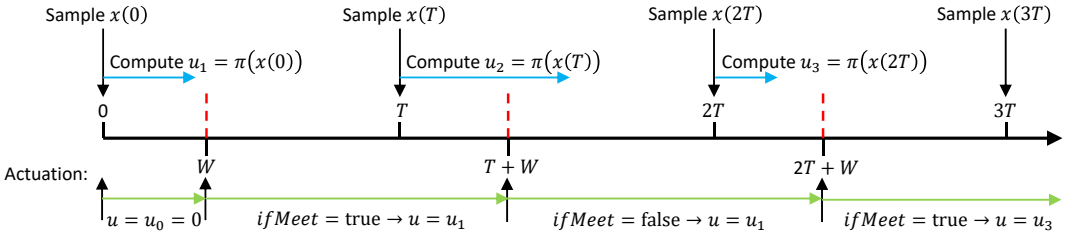


Figure 1: Structure of a Weakly-Hard System

0. Then the unsafe state set can be described as the complement of feasible region $\mathcal{X}_s$:

$$\mathcal{X}_u = \mathcal{X}_s^c = B^c(d). \tag{4}$$

Let $x(0)$ be the initial state. The safety verification of this weakly-hard system is to check whether the system will never enter a given unsafe state set $\mathcal{X}_u$ from a given initial state $x(0)$.

The weakly-hard system can be modeled as a hybrid automaton (Figure 2). It is generally difficult to verify such a system directly based on the traditional hybrid automata techniques such as bounded model checking or barrier certificate checking, due to the following factors:

- **[Unpredictable Deadline-Meeting]**: Even though we are given a bound on the deadline miss rate, we cannot know if the deadline miss will actually occur at a certain sampling period.
- **[Unknown Parameters]**: A classical hybrid automaton is purely autonomous, where no sampling is involved. The periodic sampling $x(NT)$, $N \geq 0$ brings unknown parameters into the automaton and cannot be handled with techniques such as barrier certificate.
- **[Infinite Discrete Transitions]**: The transition guard conditions depend on the period number which increases as time progresses. This means that we have an infinite number of transitions and need to consider infinite-time verification.

Before tackling above difficulties, we would like to first introduce an important assumption throughout the rest of the paper.

Assumption 1. *If the system is a hard real-time system, that is, the computation deadline is met in every sampling period, then the open loop control law π can make the system safe, that is,*

$$\|x(t)\| \leq d, \forall t \geq 0$$

Meanwhile, the system is exponentially stable for the hard real-time system under the control law π, i.e. the system is asymptotically stable and there exists $\alpha > 0$, $\lambda > 0$, such that

$$\|x(t)\| \leq \alpha e^{-\lambda t} \|x(0)\|.$$

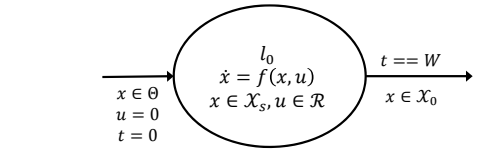
Remark 2. *Note that Assumption 1 is not restrictive in practice. If the designed control law cannot make the hard real-time system stable and safe, there is no reason to expect the system can satisfy the same safety property with deadline misses. On the flip side, analyzing exponential stability for a general control system and determining λ and α are not trivial. Our rationale behind making this assumption is largely based on the recent results from the control community on this problem [24, 33, 39, 40].*

4 PROBLEM CONVERSION

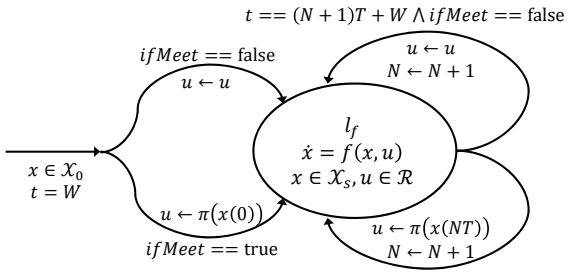
In order to solve the verification problem mentioned above, we propose to first convert the problem into three steps, namely *split*, *localize*, and *classify*. In the *split* step, we split the hybrid automaton (Figure 2) into two parts, the first incomplete period (input is fixed to 0 and lasts only for W), and the following complete periods. In the *localize* step, we consider every K consecutive complete periods as a union. In the *classify* step, we separately consider the cases of deadline miss and deadline meet.

By these three steps, we convert the original infinite time verification problem of a hybrid system with uncertainty into the verification of finite-time deterministic hybrid systems, which is much easier to handle. It is worthy noting that, each conversion step is theoretically *sufficient*, that is, a solution to the new problem after conversion must be a solution to the original one.

4.1 Split



(a) initial period



(b) subsequent periods

Figure 3: Split Models

Note that different from other inputs, the first input $u_0 = 0$ is only applied for W. Meanwhile, the system is purely deterministic in $t \in [0, W]$. Thus our first step is to split the original system into two sub-automata (Figure 3), which are named as *initial period* (Figure 3a) and *subsequent periods* (Figure 3b), respectively. Let

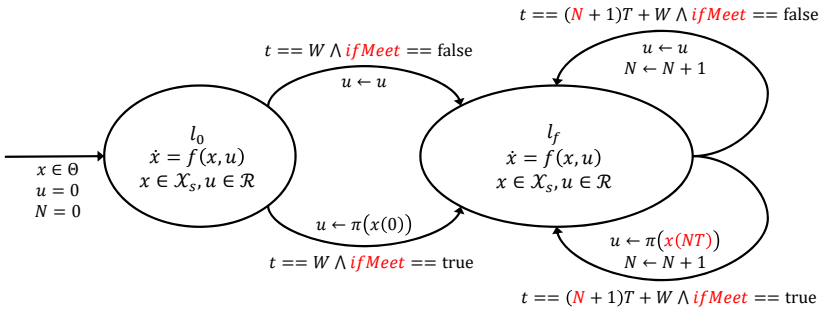


Figure 2: Hybrid Automaton Model of a Weakly-Hard System

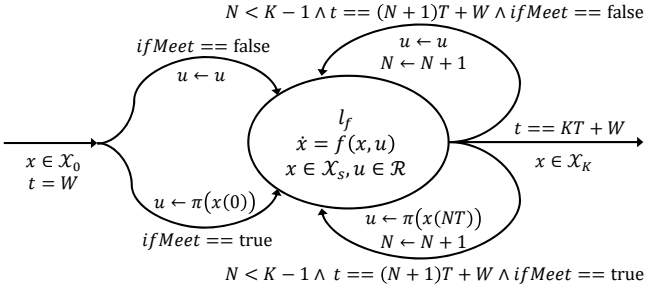


Figure 4: Localize the Model Shown in Figure 3b

$\mathcal{X}(t)$ represent the state set of the system at the instant t. When $t = NT + W$, we may sometimes use $\mathcal{X}_N$ for simplicity. Then $\mathcal{X}_0 = \mathcal{X}(W)$ can be considered either the output of the initial period or the input of the subsequent periods.

It is important to note that this "split" step is an equivalent transformation. Thus, we have following theorem.

Theorem 4.1. *The system in Figure 2 is safe* ***iff*** *both the initial period in Figure 3a and the subsequent periods in Figure 3b are safe.*

Since the initial period is a purely deterministic continuous system, we can analyze it based on classical techniques. In the next two steps, we will focus on subsequent periods.

4.2 Localize

Note that the verification of the subsequent periods is still an infinite-time problem. In this *localize* step, we try to convert it into a finite-time one. We point out that different from the *split* step, the *localize* step is not equivalent but sufficient. In the following, we describe this conversion step in detail.

Specifically, we consider the system in Figure 4, namely K-period, which describes the behavior of the control system with at most m deadline misses in K periods. It is because the set of subsequent periods is a subset of the set of the systems composed of K-period.

Proposition 4.2. *For an (m,K) weakly-hard system, its subsequent periods is composed of consecutive infinite K-periods.*

Remark 3. *Note that the opposite of Proposition 4.2 is not true. For instance, let $m = 1$ and $K = 2$. Assume the second deadline misses in a certain K-period, and the first deadline misses in the next K-period. It is obvious that deadline misses occur for consecutive two times, and thus the system is not a $(1,2)$ weakly-hard system.*

Let E_N represent the event when the system meets the deadline in the N-th period. If the deadline is met, $E_N = 1$, otherwise $E_N = 0$. Then, we can use a K-sequence $\tilde{E} = \{E_1, \cdots, E_K\}$ to represent the behavior trace of the system in a K-period, namely deadline miss/meet sequence. The (m,K) specification indicates that all the possible behaviors of the system is a sequence set $\mathcal{E}(m,K) = \{\tilde{E} | \sum_{N=1}^{K} E_N \geq K - m\}$. The safety problem of the subsequent periods can be reformulated as follows:

Theorem 4.3. *The subsequent periods is safe* ***if*** *we can find a set $\mathcal{X}_0 \subseteq \mathcal{X}_s$, such that for any start state $x \in \mathcal{X}_0$, and any $\tilde{E} \in \mathcal{E}(m,K)$,*

- ***Local Safety****: The system is safe within $[W, KT + W]$;*
- ***Inductiveness****: The reachable set $\mathcal{X}_K$ after $KT + W$ satisfies $\mathcal{X}_K \subseteq \mathcal{X}_0$.*

Proof. Let S' be the system that consists of infinite consecutive K-period starting from $\mathcal{X}_0$. By the property of local safety and inductiveness, we know that S' is safe. Combining Proposition 4.2, the theorem is true. □

Remark 4. *Readers may find the concept of $\mathcal{X}_0$ is similar to* forward invariant. *Note that $\mathcal{X}_0$ is weaker since we do not need $x(t) \in \mathcal{X}_0$, $t \in (W, KT + W)$. The analogous concept can be found in [51], which is used for controller synthesis of an industrial oil pump.*

By *localize* step, we focus our attention to the finite time system K-period. In the next step, we further refine the safety problem, and give the formal description of the problem after converted.

4.3 Classify

The K-period consists of K consecutive periods, where the input computation of each period can either meet or not meet the deadline. We can therefore consider these two cases separately.

When $N \in [0, K-1]$, we use the function $R(A,t,E) : \mathcal{R}^n \times [0,T] \times \{0,1\} \rightarrow \mathcal{R}^n$ to denote the reachable set function from the set A after time t with the event E (0 for deadline miss and 1 for deadline meet) in a sampling period. Then we have the following theorem.

Theorem 4.4. *The subsequent periods is safe* ***if*** *we can find a set $\mathcal{X}_0 \subseteq \mathcal{X}_s$, such that for any start state $x \in \mathcal{X}_0$,*

- ***Local Safety****: The system is safe within $[W, KT + W]$;*

- ***Inductiveness**: $\mathcal{X}_0$ satisfies:*

$$\bigcup_{\tilde{E}\in\mathcal{E}(m,K)} R(\cdot,T,E_{K-1})\circ\cdots\circ R(\cdot,T,E_0)(\mathcal{X}_0)\subseteq\mathcal{X}_0 \tag{5}$$

Recall the initial period. Once $\mathcal{X}_0$ is obtained, we can see that if from a certain initial state $x(0)$, the system with zero input, that is $\dot{x}=f(x)$, enters $\mathcal{X}_0$ at $t=W$ and is safe within $[0,W]$, then the system is safe over $t\in[0,\infty)$. Thus we need to compute the *region of attraction* (ROA) that is defined as:

Definition 4.5. The set Θ is called *region of attraction* of the system $\dot{x}=f(x)$, **if** it consists of all the state from which the system can enter $\mathcal{X}_0$ at $t=W$. Formally,

$$\Theta\triangleq\{x_0\mid\exists x(\cdot),\text{s.t. }\dot{x}=f(x)\wedge x(0)=x_0\wedge x(W)\in\mathcal{X}_0\wedge x(t)\in\mathcal{X}_s,\forall t\in[0,W]\}$$

We can now formally state the problem we will try to solve in this paper by combining Theorem 4.1, 4.4 and Definition 4.5:

Problem 1. *For an (m,K) weakly-hard system with dynamic* (1)*, control policy* (2)*, unsafe region $\mathcal{X}_u$* (4)*, find a set Θ, such that there exists a set $\mathcal{X}_0$, which satisfies*

- *Θ is ROA of $\dot{x}=f(x)$;*
- local safety *and* inductiveness *in Theorem 4.4.*

After Θ is obtained, the safety of the system with a given initial state $x(0)$ can be easily checked by testing if $x(0)\in\Theta$.

5 TECHNIQUES AND ALGORITHMS

The key step of solving Problem 1 is to find $\mathcal{X}_0$. However, it is difficult to obtain the exact $\mathcal{X}_0$ for an arbitrary nonlinear system. Inspired by the classical idea that uses over-approximation, instead of exact reachable set, to verify system safety, we propose an over-approximation based approach to estimate $\mathcal{X}_0$ with guarantee of soundness. Specifically, we use a closed ball $B(r(t))$ around the origin with radius $r(t)$ to over-approximate the reachable set $\mathcal{X}(t)$ at any time (Figure 5). For convenience, we use r_N to represent $r(t)$, when $t=NT+W$, $N=0,1,\cdots,K$. Thus, we can estimate the reachable set by just considering a one-dimensional variable, the radius $r(t)$, which greatly reduces the complexity. In the following, we describe the corresponding algorithms in detail.

First, we assume that $\mathcal{X}_0=B(r_0)$, where r_0 is the parameter we hope to obtain. Obviously, the system satisfies local safety if the over-approximated state set is contained inside the safe region $\mathcal{X}_s$.

Theorem 5.1. *$\forall t\in[W,KT+W]$, $\mathcal{X}(t)\subseteq\mathcal{X}_s$ if $B(r(t))\subseteq\mathcal{X}_s$.*

For inductiveness, we know that if the over-approximated state set $B(r_K)$ is contained inside $\mathcal{X}_0$, $\mathcal{X}_K$ is also contained inside $\mathcal{X}_0$.

Theorem 5.2. *$\mathcal{X}_K\subseteq\mathcal{X}_0$ if $B(r_K)\subseteq\mathcal{X}_0$.*

The key of applying Theorem 5.1 and Theorem 5.2 is to estimate $r(t)$, $t\in[W,KT+W]$. In the following, we separately analyze the estimation of $r(t)$ under the cases of deadline miss or deadline meet.

By Assumption 1, we know that the correct input should cause the bound of system state to contract when the computation meets the deadline. On the other hand, the deadline miss may make the system divergent. If the input computation meets the deadline at the N-th sampling period, let $\zeta(N)$ be the length of the consecutive deadline meet sequence ended at the N-th sampling period, and let $\Delta t=t-NT-W\in[0,T]$. Then for $t\in[NT+W,(N+1)T+W]$,

$$r(t)\le\alpha e^{-\lambda((\zeta(N)-1)T+\Delta t)}r_{N-\zeta(N)+1}.$$

Note that the above constraint is uncountable in $t\in[NT+W,(N+1)T+W]$. We use two over-approximate constraints to describe it. For $t\in[NT+W,(N+1)T+W]$, since $e^{\Delta t}\ge 0$, we have

$$r(t)\le\alpha e^{-\lambda(\zeta(N)-1)T}r_{N-\zeta(N)+1}. \tag{6}$$

Specially, for the $t=(N+1)T+W$, we have

$$r(t)=r_{N+1}\le\alpha e^{-\lambda\zeta(N)T}r_{N-\zeta(N)+1}. \tag{7}$$

If the input computation misses the deadline at the N-th sampling cycle, the system state may move further away from the origin due to the wrong input function. Given any $x(NT+W)\in\mathcal{X}_N$, let $\bar{x}(t)$ and $\underline{x}(t)$ be the state at $t\in[NT+W,(N+1)T+W]$ evolving from $x(NT+W)$ with the last correct input $u(x((N-\delta(N))T+W))$ before $\delta(N)$ sampling periods and with the correct input $u(x(NT))$, respectively, we have

$$\begin{aligned}
&r(t)-\underline{r}(t)\\
&=\max_{x_N,x_{N-\delta(N)}}\|\bar{x}(t)\|-\left\|\underline{x}(t)\right\|\le\max_{x_N,x_{N-\delta(N)}}\left\|\bar{x}(t)-\underline{x}(t)\right\|\\
&=\max_{x_N,x_{N-\delta(N)}}\Big\|(x(NT+W)+\int_0^{\Delta t}(f(x)+g(x)\pi(x_{N-\delta(N)}))d\mu)-\\
&\qquad\qquad(x(NT+W)+\int_0^{\Delta t}(f(x)+g(x)\pi(x_N))d\mu)\Big\|\\
&=\max_{x_N,x_{N-\delta(N)}}\left\|\int_0^{\Delta t}g(x)(\pi(x_{N-\delta(N)})-\pi(x_N))d\mu\right\|\\
&\le\max_{x_N,x_{N-\delta(N)}}\int_0^{\Delta t}\left\|g(x)(\pi(x_{N-\delta(N)})-\pi(x_N))\right\|d\mu\\
&\le\max_{x_N,x_{N-\delta(N)}}\max_{x\in\mathcal{X}}\|g(x)\|\cdot\int_0^{\Delta t}\|\pi(x((N-\delta(N))T))-\pi(x(NT))\|\,d\mu\\
&\le\max_{x_N,x_{N-\delta(N)}}\max_{x\in\mathcal{X}}\|g(x)\|\,c_L(\|x((N-\delta(N))T)\|+\|x(NT)\|)\Delta t\\
&=\max_{x\in\mathcal{X}}\|g(x)\|\,c_L(r_{N-\delta(N)}+r_N)\Delta t
\end{aligned}$$

Let $\gamma=\max_{x\in\mathcal{X}}\|g(x)\|$. We call $\gamma c_L(r_{N-\delta(N)}+r_N)\Delta t$ the *deadline miss error*. Since $\underline{r}(t)$ corresponds to the case that deadline is met and $\zeta_N\ge 1$, we have

$$r(t)\le\underline{r}(t)+\gamma c_L(r_{N-\delta(N)}+r_N)t\le e^{-\lambda\Delta t}r_N+\gamma c_L(r_{N-\delta(N)}+r_N)\Delta t$$

Note that if we independently consider the system over $[W,KT+W]$, the case of $N<\delta_N$ is ill-defined. To handle it, we use d to bound $r_{N-\delta(N)}$ when $N<\delta_N$, which relies on the underlying precondition of safety. For $t\in[NT+W,(N+1)T+W]$, we have

$$r_t\le\begin{cases}\alpha e^{-\lambda\Delta t}r_N+\gamma c_L(d+r_N)\Delta t & N<\delta_N\\ \alpha e^{-\lambda\Delta t}r_N+\gamma c_L(r_{N-\delta(N)}+r_N)\Delta t & N\ge\delta(N)\end{cases}$$

Similar to the case when the deadline is met, we split the above constraint into two. For $t\in[NT+W,(N+1)T+W]$,

$$r_t\le\begin{cases}\alpha r_N+\gamma c_L(d+r_N)T & N<\delta_N\\ \alpha r_N+\gamma c_L(r_{N-\delta(N)}+r_N)T & N\ge\delta(N)\end{cases} \tag{8}$$

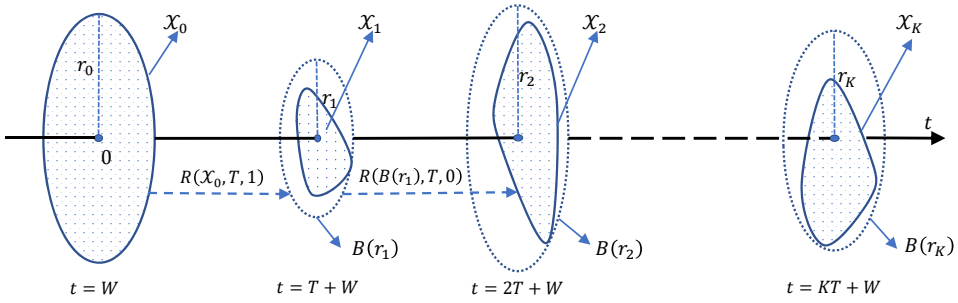


Figure 5: Over-approximation

Specially, for the $t = (N+1)T + W$, we have:

$$r_{N+1} \leq \begin{cases} \alpha e^{-\lambda T} r_N + \gamma c_L (d + r_N) T & N < \delta(N) \\ \alpha e^{-\lambda T} r_N + \gamma c_L (r_{N-\delta(N)} + r_N) T & N \geq \delta(N) \end{cases} \tag{9}$$

As mentioned in Theorem 4.3, a suitable $\mathcal{X}_0$ should satisfy both local safety and inductiveness. Now we respectively consider these two properties.

[Local Safety]: The system is safe if it is safe in the worst case. Thus we can consider the worst case based on the constraints (6), (7), (8), (9). Specifically, for a given sequence $\tilde{E} \in \mathcal{E}(m,K)$, we can solve the following linear programming (LP) problem:

$$\text{Find } r_0(\tilde{E})$$

such that

$$\begin{cases} r'_{N+1} = \begin{cases} e^{-\lambda(\zeta(N)-1)T} r_{N-\zeta(N)+1}, & E_N = 1, \\ \alpha r_N + \gamma c_L (d + r_N) T, & E_N = 0, N < \delta(N), \\ \alpha r_N + \gamma c_L (r_{N-\delta(N)} + r_N) T, & E_N = 0, N \geq \delta(N), \end{cases} \\ \quad 0 \leq N \leq K-1, \\ r_{N+1} = \begin{cases} \alpha e^{-\lambda \zeta(N) T} r_{N-\zeta(N)+1}, & E_N = 1, \\ \alpha e^{-\lambda T} r_N + \gamma c_L (d + r_N) T, & E_N = 0, N < \delta(N), \\ \alpha e^{-\lambda T} r_N + \gamma c_L (r_{N-\delta(N)} + r_N) T, & E_N = 0, N \geq \delta(N). \end{cases} \\ \quad 0 \leq N \leq K-1, \\ 0 \leq r'_N \leq d, \quad 1 \leq N \leq K, \\ 0 \leq r_N \leq d, \quad 0 \leq N \leq K, \end{cases} \tag{10}$$

Observing Constraint (10), we have the following important conclusion on soundness:

Theorem 5.3. *Given sequence $\tilde{E} \in \mathcal{E}(m,K)$, if r is a feasible solution of Constraint* (10)*, then the system satisfies local safety from any state $x(W) \in B(r)$.*

This LP problem provides the following property.

Proposition 5.4. *Given sequence $\tilde{E} \in \mathcal{E}(m,K)$, if r is a feasible solution of Constraint* (10)*, then any $r' \in [0, r]$ is also a feasible solution of* (10).

By Proposition 5.4, we can expect to find the largest r_0 that satisfies local safety. Meanwhile, a suitable r_0 should satisfy Constraint (10) for any sequence in $\mathcal{E}$. Thus, let $\tilde{R}_{\text{safe}}(\tilde{E})$ be the feasible region of Constraint (10), we can find the largest r_0 for local safety by solving:

$$r_{\text{safe}} = \min_{\tilde{E} \in \mathcal{E}(m,K)} \max_{r_0, \cdots, r_K \in \tilde{R}_{\text{safe}}(\tilde{E})} r_0(\tilde{E}) \tag{11}$$

Essentially, Problem (11) is a minimax robust optimization problem. Note that the state space $\mathcal{E}(m,K)$ of $\tilde{E}$ is discrete and finite, which allows an enumeration of all possible patterns to find the solution. Specifically, we implement the algorithm as Algorithm 1.

Algorithm 1: MaxSafeRadius

Data: Dynamic system (1) with state constraint (3), the control law (2) with exponentially stable parameter (α, λ), radius of safe state region d, weakly-hard constraint (m,K), sampling period T

Result: Safe initial radius bound for K-period r_{safe}

```
E(m,K) ← generateSequences();
r_safe ← ∞;
for Ẽ ∈ E(m,K) do
    r ← max_{r_0,⋯,r_K ∈ R̃_safe(Ẽ)} r_0(Ẽ);
    if r_safe ≥ r then
        r_safe ← r;
    end
end
return r_safe;
```

[Inductiveness]: Similar to the analysis of local safety, we consider the worst case based on the constraints (6), (7), (8), (9). Specifically, for a given sequence $\tilde{E} \in \mathcal{E}(m,K)$, we need to solve the following linear programming problem:

$$\text{Find } r_0(\tilde{E})$$

such that

$$\begin{cases} r_{N+1} = \begin{cases} \alpha e^{-\lambda \zeta(N) T} r_{N-\zeta(N)+1}, & E_N = 1, \\ \alpha e^{-\lambda T} r_N + \gamma c_L (d + r_N) T, & E_N = 0, N < \delta(N), \\ \alpha e^{-\lambda T} r_N + \gamma c_L (r_{N-\delta(N)} + r_N) T, & E_N = 0, N \geq \delta(N). \end{cases} \\ \quad 0 \leq N \leq K-1, \\ r_0 \geq r_K, \quad r_0 \geq 0. \end{cases} \tag{12}$$

Observing Constraint (12), we have the following important conclusion on soundness:

Algorithm 2: MinInductiveRadius

Data: Dynamic system (1) with state constraint (3), the control law (2) with exponentially stable parameter (α, λ), radius of safe state region d, (m,K) constraint, sampling period T

Result: Inductive initial radius bound for K-period $r_{\text{inductive}}$

$\mathcal{E}(m,K) \leftarrow$ **generateSequences**();
$r_{\text{inductive}} \leftarrow 0$;
for $\tilde{E} \in \mathcal{E}(m,K)$ **do**
  $r \leftarrow \min_{r_0,\cdots,r_K \in \tilde{R}_{\text{inductive}}(\tilde{E})} r_0(\tilde{E})$;
  if $r_{inductive} \leq r$ **then**
    $r_{\text{inductive}} \leftarrow r$;
  end
end
return $r_{inductive}$;

Theorem 5.5. *Given sequence $\tilde{E} \in \mathcal{E}(m,K)$, if r is a feasible solution of Constraint* (12)*, then the system will reach $B(r)$ after K sampling periods from any initial state $x(W) \in B(r)$. That is, the system satisfies inductiveness.*

This LP problem provides the following property.

Proposition 5.6. *Given sequence $\tilde{E} \in \mathcal{E}(m,K)$, if r is a feasible solution of Constraint* (12)*, then any $r' \in [r, \infty)$ is also a feasible solution of* (12).

By Proposition 5.6, we can expect to find the smallest r_0 that satisfies inductiveness. Meanwhile, a suitable r_0 should satisfy Constraint (12) for any sequence in $\mathcal{E}(m,K)$. Thus, let $\tilde{R}_{\text{inductive}}(\tilde{E})$ be the feasible region of Constraint (12), we can find the smallest r_0 for inductiveness by solving:

$$r_{\text{inductive}} = \max_{\tilde{E} \in \mathcal{E}(m,K)} \min_{r_0,\cdots,r_K \in \tilde{R}_{\text{inductive}}(\tilde{E})} r_0(\tilde{E}) \tag{13}$$

Similar to Problem (11), we enumerate all the possible patterns to find the solution of Problem (13) by Algorithm 2.

Note that when we estimate $r_{\text{inductive}}$, the bounds in (8) and (9) rely on the local safety. Thus $r_{\text{inductive}}$ is valid only if $r_{\text{inductive}} \leq r_{\text{safe}}$ and it directly implies the following conclusion.

Proposition 5.7. *Any $r \in [r_{inductive}, r_{safe}]$ satisfies both local stability and inductiveness, if $r_{inductive} \leq r_{safe}$.*

When Proposition 5.7 is true, to get the largest $\mathcal{X}_0$, we let $\mathcal{X}_0 = B(r_{\text{safe}})$. Then we can obtain Θ by computing the (inner) region of attraction. In this paper, we adopt the moment-theory based method in [23]. Note that, if we specially consider the region of attraction Θ as a circle around a fixed center, the origin, with a variable radius r_Θ, then ROA becomes a one-dimensional problem. Thus we can also use bisection to find r_Θ based on barrier certificate approaches [35, 49]. The main algorithm is given in Algorithm 3.

Based on the above analysis, we give the conclusion on the soundness of our approach, which can be proved directly combining Proposition 5.4, 5.6 and 5.7.

Theorem 5.8 (Soundness). *Let Θ be the set obtained by Algorithm 3. If the initial state $x(0) \in \Theta$, then the (m,K) weakly-hard system is safe.*

Algorithm 3: Safe initial state set computation (Main algorithm)

Data: Dynamic system (1) with state constraint (3), the control law (2) with exponentially stable parameter (α, λ), radius of safe state region d, (m,K) constraint, sampling period T, typical worst-case response time W

Result: Safe initial state set Θ

$r_{\text{safe}} \leftarrow$ **MaxSafeRadius**();
$r_{\text{inductive}} \leftarrow$ **MinInductiveRadius**();
if $r_{safe} \geq r_{inductive}$ **then**
  $r_0 \leftarrow r_{\text{safe}}$;
  $\Theta \leftarrow$ **ROA**$(B(r_0), W)$;
  return Θ;
else
  return *"We cannot find the safe initial state set."*;
end

6 EXPERIMENT

We consider the following four kinds of deadline miss/meet sequence $\tilde{E}$ of K-period. They represent further refinements of the (m,K) constraint.

- **Case 1**: Arbitrary deadline miss/meet sequence consistent with (m,K). We use $\mathcal{E}_1(m,K)$ to represent the set that consists of all such sequences;
- **Case 2**: No consecutive deadline miss is allowed in $\tilde{E}$. We use $\mathcal{E}_2(m,K)$ to represent this set;
- **Case 3**: First deadline must be met in $\tilde{E}$. We use $\mathcal{E}_3(m,K)$ to represent this set;
- **Case 4**: $\tilde{E}$ simultaneously satisfies the requirements of Case 2 and Case 3. We use $\mathcal{E}_4(m,K)$ to represent this set.

Remark 5. *Mathematically, for a given (m,K), we have*

$$\mathcal{E}_4(m,K) \subseteq \mathcal{E}_2(m,K), \mathcal{E}_3(m,K) \subseteq \mathcal{E}_1(m,K).$$

Their practical meaning is as follows. Case 1 is the most general case following the definition of (m,K) weakly-hard systems. However, in many situations, the safety requirement can be hard to satisfy under Case 1, since multiple consecutive deadline misses will rapidly increase the deadline miss error and lead to unsafety. Thus, we consider Case 2 as a weaker situation, where the deadline miss error could be reduced by the next several deadline meets. Case 3 is another special weaker case compared to Case 1. Observe Constraint (8) *and* (9)*. If the deadline is missed in the first sampling period of the K-period, we should use d instead of $r_{N-\delta(N)}$ to estimate the bound, which is much looser. Thus, by letting the deadline be met in the first sampling period, we have more chance to verify a given weakly-hard system. If no conclusion on safety can be obtained under either Case 2 or Case 3, we can consider the weakest situation Case 4.*

We introduce following examples selected from related works.

Example 1. *[24, 39, 40] The linear dynamic system is:*

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -0.9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + u, \quad \textit{where } u = \begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

The safety distance is $d = 6$. The sampling period is $T = 0.3$ and the TWCRT $W = 0.1$. The sampled system with no deadline miss is exponentially stable with parameters $\alpha = 1.1$ and $\lambda = 1.8$.

Example 2. *[24, 39, 40] The linear dynamic system is:*

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -0.1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + u,, \quad where \ u = \begin{bmatrix}0 & 0\\ -0.375 & -1.15\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}.$$

The safety distance is $d = 2$. The sampling period is $T = 1$ and the TWCRT $W = 0.2$. The sampled system with no deadline miss is exponentially stable with parameters $\alpha = 1.8$ and $\lambda = 0.4$.

Example 3. *[40] The linear dynamic system is:*

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -2 & -0.1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + u,, \quad where \ u = \begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}.$$

The safety distance is $d = 3$. The sampling period is $T = 1.6$ and the TWCRT $W = 0.3$. The sampled system with no deadline miss is exponentially stable with parameters $\alpha = 2$ and $\lambda = 0.37$.

Example 4. *[33] The nonlinear dynamic system is:*

$$\dot{x} = x^2 - x^3 + u,, \quad where \ u = -2x.$$

The safety distance is $d = 4$. The sampling period is $T = 0.6$ and the TWCRT $W = 0.1$. The sampled system with no deadline miss is exponentially stable with parameters $\alpha = 1.4$ and $\lambda = 1$.

Our experimental evaluation consists of three parts. First, we take Example 1 for instance and demonstrate the safe radius r_0 of $\mathcal{X}_0$ obtained by our approach with different configurations. Second, we show the minimal K value that provides feasible solution under different configurations for all four cases. Finally, we give the simulation results under selected configuration for Example 1. All experiments were performed on Intel i7-8700 machine with 16GB memory and the LP problems were solved using YALMIP [29].

6.1 Experimental Results of Example 1

Table 1 shows the safe radius r_0 of $\mathcal{X}_0$ with different (m,K) constraints under different deadline miss/meet sequence cases for Example 1, where $1 \le m \le K \le 15$.

First, most examples completed in seconds, while the longest computation time occurred when $m = 7$ and $K = 15$, which took about 10 minutes. Since we reduce the original problem to a one-dimensional problem, the complexity of our approach is independent of the system size, but relies on m and K. In the worst case, our approach terminates after solving $2 \cdot \binom{K}{m}$ LP problems.

Second, we can see that for each of the four cases, our approach can obtain a feasible safe radius r_0 of $\mathcal{X}_0$ with a satisfactory ratio m/K. For instance, in Case 1, the system can be safe with at most 1 deadline miss in every 3 consecutive sampling periods, that is, $m/K = 33\%$. For all four cases, the allowable deadline miss ratio $m/K \in [0.273, 0.418]$, which means a significant portion of system resources can be saved.

Third, we can find that for a given case and a given m, if there exists a positive integer K_0, such that the safe radius $r_0(K_0)$ of $\mathcal{X}_0$ exists for (m,K_0), then for any $K \ge K_0$, the safe radius $r_0(K)$ also exists. In addition, $r_0(K) = r_0(K_0)$, $K \ge K_0$. It is due to that the value of the safe radius r_0 is determined by r_{safe} if exists. For a given m, the worst case for safety occurs when the first m deadlines are all missed in K-period. Thus, once r_{safe} exists for a given (m,K_0), it will remain the same for (m,K), $K \ge K_0$. As for the inductiveness, due to the property of exponential stability, for a given m, the larger the K is, the more contracted the system is, which makes it easier to guarantee inductiveness, that is, $r_{\text{inductive}}$ monotone decreases with K. Combining the requirement of $r_{\text{inductive}} \le r_{\text{safe}}$, we can arrive at the above conclusion.

Finally, we can find that for a given (m,K), the safe radius r of $\mathcal{X}_0$ exists for Case 4, if one exists for Case 2 or Case 3. r_0 exists for Case 2 or Case 3, if one exists for Case 1. In addition, if r_0 exists for all four cases, let $r_0(\text{Case}_i)$ be the solution under Case i, $i = 1,2,3,4$, we have $r_0(\text{Case}_1) \le r_0(\text{Case}_2), r_0(\text{Case}_3) \le r_0(\text{Case}_4)$. This property is due to the partial order among $\mathcal{E}_i(m,K)$, $i = 1,2,3,4$ mentioned in Remark 5.

6.2 A Glance on All Experimental Results

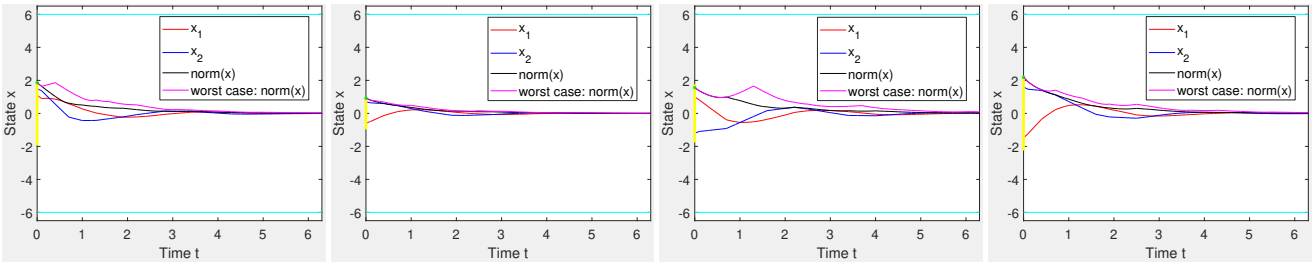
Table 2: Minimal K value that provides feasible solution: This table shows the minimal K value for a given m such that the safe radius r_0 can be found by our approach under the weakly-hard constraint (m,K) for each example in different cases of alternating meet and miss sequence. We use "-" to represent there is no feasible $K \le 15$ for a given m.

		m					
	type	1	2	3	4	5	6
Example 1	Case 1	3	-	-	-	-	-
	Case 2	3	7	11	-	-	-
	Case 3	3	5	8	10	12	15
	Case 4	3	5	8	10	12	15
Example 2	Case 1	-	-	-	-	-	-
	Case 2	-	-	-	-	-	-
	Case 3	8	14	-	-	-	-
	Case 3	8	14	-	-	-	-
Example 3	Case 1	-	-	-	-	-	-
	Case 2	-	-	-	-	-	-
	Case 3	7	12	-	-	-	-
	Case 4	7	12	-	-	-	-
Example 4	Case 1	-	-	-	-	-	-
	Case 2	-	-	-	-	-	-
	Case 3	5	9	13	-	-	-
	Case 4	5	9	13	-	-	-

Due to the space limit, we show the minimal feasible K for a given m of all four examples in Table 2. The complete experimental results with concrete r_0 can be found in the appendix. Since the exponential stability parameters α and λ are much larger and smaller respectively in Examples 2-4, the deadline miss ratio m/K is lower than it in Example 1. Specifically, in Case 1 and Case 2, we cannot find a feasible K for any amount of deadline misses. However, if we restrict the deadline miss/meet sequence to Case 3 and Case 4, we can still find at least 12.5%, 14.3%, 20.0% deadline miss ratio for Examples 2, 3, 4, respectively, which also leads to a significant resource saving.

Table 1: Safe radius r_0 of $\mathcal{X}_0$ for Example 1: This table shows the value of the safe radius r found by our approach under the weakly-hard constraint (m,K) for Example 1 in different cases of deadline miss/meet sequence. We use "-" to represent there is no feasible r that can be found by our approach for a given (m,K).

			K														
type			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Case 1	m	1	-	-	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947
Case 2	m	1	-	-	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947
		2	-	-	-	-	-	-	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.962
		3	-	-	-	-	-	-	-	-	-	-	0.352	0.352	0.352	0.352	0.352
Case 3	m	1	-	-	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995
		2	-	-	-	-	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500
		3	-	-	-	-	-	-	-	1.759	1.759	1.759	1.759	1.759	1.759	1.759	1.759
		4	-	-	-	-	-	-	-	-	-	1.318	1.318	1.318	1.318	1.318	1.318
		5	-	-	-	-	-	-	-	-	-	-	-	0.953	0.953	0.953	0.953
		6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.824
Case 4	m	1	-	-	3.995	3.995	3.995	3.996	3.996	3.996	3.996	3.996	3.996	3.996	3.996	3.996	3.996
		2	-	-	-	-	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309
		3	-	-	-	-	-	-	-	2.741	2.741	2.741	2.741	2.741	2.741	2.741	2.741
		4	-	-	-	-	-	-	-	-	-	2.270	2.270	2.270	2.270	2.270	2.270
		5	-	-	-	-	-	-	-	-	-	-	-	1.880	1.880	1.880	1.880
		6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1.557



(a) Case 1, m=1, K=3, $x_0 = \left[1, 1.5\right]^T$ **(b) Case 2,** m=2, K=7, $x_0 = \left[-0.6, 0.7\right]^T$ **(c) Case 3,** m=3, K=8, $x_0 = \left[1, -1.2\right]^T$ **(d) Case 4,** m=4, K=10, $x_0 = \left[-1.5, 1.6\right]^T$

Figure 6: Simulation Results of Selected Configurations of Example 1

6.3 Simulation Results of Example 1

We ran simulations of several configurations of Example 1 (Figure 6) to assess how conservative the over-approximation is. In each subfigure, the yellow line represents the safe initial state set obtained by our approach, the green point represents the distance (2-norm) of the initial state from the origin, the red curve represents the trajectory of x_1, the blue curve represents the trajectory of x_2, the black curve represents the trajectory of the distance of x from the origin, $\|x\|$, the purple curve represents the trajectory of $\|x\|$ under the worst case, and the two cyan lines are the safety bound of $\|x\|$. We can see the state trajectories from a safe initial state contract away from the boundary quickly even in the worst case, which indicates our approach might be too conserved. In future, we will consider tightening the estimation for the safe radius r of $\mathcal{X}_0$.

7 CONCLUSION

In this paper, we study the safety verification problem of weakly-hard systems with nonlinear dynamics. We propose a relaxation technique to convert the original infinite-time safety problem into finite-time problem of local safety and inductiveness. We have developed an over-approximation based technique to analyze these two properties and give a sufficient initial condition to ensure safety. Our experiment results show that our approach allows the designer to better understand the impact of different (m,K) constraints on the safety of the system. Our future work includes tightening the estimation of the safe initial set, and applying the proposed technique to larger systems such as adaptive cruise control and model predictive control of self-driving vehicles.

ACKNOWLEDGMENTS

We gratefully acknowledge the support from the National Science Foundation awards 1834701, 1834324, 1839511, and 1724341. This work is also funded in part by the DARPA BRASS program under agreement number FA8750-16-C-0043 and NSF grant CCF-1646497.

REFERENCES

[1] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. 2008. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica* 44, 11 (2008), 2724–2734.

[2] L. Ahrendts, S. Quinton, T. Boroske, and R. Ernst. 2018. Verifying Weakly-Hard Real-Time Properties of Traffic Streams in Switched Networks. In *ECRTS*, Vol. 106. 15:1–15:22.

[3] E. Alur, C. Courcoubetis, T. A. Henzinger, and P. Ho. 1993. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid systems*. Springer, 209–229.

[4] A. Colin K. Maeng B. Lucia, V. Balaji and E. Ruppel. 2017. Intermittent Computing: Challenges and Opportunities. In *Summit on Advances in Programming Languages*. 8:1–8:14.

[5] D. Balsamo, A. S. Weddell, A. Das, A. R. Arreola, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini. 2016. Hibernus: A Self-Calibrating and Adaptive System for Transiently-Powered Embedded Devices. *TCAD* 35, 12 (2016), 1968–1980.

[6] G. Bernat, A. Burns, and A. Liamosi. 2001. Weakly hard real-time systems. *IEEE transactions on Computers* 50, 4 (2001), 308–321.

[7] G. Bernat and R. Cayssials. 2001. Guaranteed on-line weakly-hard real-time systems. In *RTSS*. 22–35.

[8] T. Bund and F. Slomka. 2014. Controller/platform co-design of networked control systems based on density functions. In *ACM SIGBED International Workshop on Design, Modeling, and Evaluation of Cyber-Physical Systems*. ACM, 11–14.

[9] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen. 2003. How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. *IEEE Control Systems Magazine* 23, 3 (June 2003), 16–30.

[10] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *USENIX Security*.

[11] T. Chen and B. A. Francis. 2012. *Optimal sampled-data control systems*. Springer Science & Business Media.

[12] H. Choi, H. Kim, and Q. Zhu. 2019. Job-Class-Level Fixed Priority Scheduling of Weakly-Hard Real-Time Systems. In *IEEE Real-Time Technology and Applications Symposium (RTAS)*.

[13] H. S. Chwa, K. G. Shin, and J. Lee. 2018. Closing the Gap between Stability and Schedulability: A New Task Model for Cyber-Physical Systems. In *IEEE Real-Time Technology and Applications Symposium (RTAS)*.

[14] A. Colin and B. Lucia. 2016. Chain: Tasks and Channels for Reliable Intermittent Programs. *SIGPLAN Not.* 51, 10 (Oct. 2016), 514–530.

[15] P. S. Duggirala and M. Viswanathan. 2015. Analyzing real time linear control systems using software verification. In *RTSS*. IEEE, 216–226.

[16] G. Frehse, A. Hamann, S. Quinton, and M. Woehrle. 2014. Formal Analysis of Timing Effects on Closed-Loop Properties of Control Software. In *RTSS*. 53–62.

[17] M. B. Gaid, D. Simon, and O. Sename. 2008. A Design Methodology for Weakly-Hard Real-Time Control. *IFAC* 41, 2 (2008), 10258 – 10264.

[18] M. Hamdaoui and P. Ramanathan. 1995. A dynamic priority assignment technique for streams with (m, k)-firm deadlines. *IEEE Trans. Comput.* 44, 12 (1995), 1443–1451.

[19] Z. A. H. Hammadeh, R. Ernst, S. Quinton, R. Henia, and L. Rioux. 2017. Bounding deadline misses in weakly-hard real-time systems with task dependencies. In *DATE*. 584–589.

[20] D. Henrion and M. Korda. 2014. Convex computation of the region of attraction of polynomial control systems. *IEEE Trans. Automat. Control* 59, 2 (2014), 297–312.

[21] C. Huang, X. Chen, W. Lin, Z. Yang, and X. Li. 2017. Probabilistic Safety Verification of Stochastic Hybrid Systems Using Barrier Certificates. *TECS* 16, 5s (2017), 186.

[22] C. Huang, K. Wardega, W. Li, and Q. Zhu. 2019. Exploring Weakly-hard Paradigm for Networked Systems. In *Workshop on Design Automation for CPS and IoT (DESTION'19)*.

[23] M. Korda, D. Henrion, and C. N. Jones. 2013. Inner approximations of the region of attraction for polynomial dynamical systems. *IFAC* 46, 23 (2013), 534–539.

[24] T. H. Lee and J. H. Park. 2017. Stability Analysis of Sampled-Data Systems via Free-Matrix-Based Time-Dependent Discontinuous Lyapunov Approach. *IEEE Trans. Automat. Control* 62, 7 (2017), 3653–3657.

[25] Tae H. Lee, Ju H. Park, S. M. Lee, and O. M. Kwon. 2014. Robust sampled-data control with random missing data scenario. *Internat. J. Control* 87, 9 (2014), 1957–1969.

[26] J. Li, Y. Song, and F. Simonot-Lion. 2006. Providing Real-Time Applications With Graceful Degradation of QoS and Fault Tolerance According to (m, k)-Firm Model. *IEEE Transactions on Industrial Informatics* 2, 2 (2006), 112–119.

[27] W. Li, L. Gérard, and N. Shankar. 2015. Design and verification of multi-rate distributed systems. In *MEMOCODE*. 20–29.

[28] C. Lin, B. Zheng, Q. Zhu, and A. Sangiovanni-Vincentelli. 2015. Security-Aware Design Methodology and Optimization for Automotive Systems. *ACM Transactions on Design Automation of Electronic Systems* 21, 1, Article 18 (December 2015), 26 pages.

[29] J. Löfberg. 2004. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *CACSD*. Taipei, Taiwan.

[30] R. Majumdar, I. Saha, and M. Zamani. 2011. Performance-aware scheduler synthesis for control systems. In *EmSoft*. ACM, 299–308.

[31] P. Marti, A. Camacho, M. Velasco, and M. E. M. Ben Gaid. 2010. Runtime Allocation of Optional Control Jobs to a Set of CAN-Based Networked Control Systems. *IEEE Transactions on Industrial Informatics* 6, 4 (2010), 503–520.

[32] D. Nešić, A. R. Teel, and P. V. Kokotović. 1999. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems & Control Letters* 38, 4-5 (1999), 259–270.

[33] H. Omran, L. Hetel, M. Petreczky, J. Richard, and F. Lamnabhi-Lagarrigue. 2016. Stability analysis of some classes of input-affine nonlinear systems with aperiodic sampled-data control. *Automatica* 70 (2016), 266–274.

[34] P. Pazzaglia, L. Pannocchi, A. Biondi, and M. D. Natale. 2018. Beyond the Weakly Hard Model: Measuring the Performance Cost of Deadline Misses. In *ECRTS*, Vol. 106. 10:1–10:22.

[35] S. Prajna and A. Jadbabaie. 2004. Safety verification of hybrid systems using barrier certificates. In *HSCC*. Springer, 477–492.

[36] S. Quinton and R. Ernst. 2012. Generalized weakly-hard constraints. In *ISoLA*. Springer, 96–110.

[37] P. Ramanathan. 1999. Overload management in real-time control applications using (m, k)-firm guarantee. *IEEE Transactions on Parallel and Distributed Systems* 10, 6 (1999), 549–559.

[38] W. Ruan, X. Huang, and MZ. Kwiatkowska. 2018. Reachability analysis of deep neural networks with provable guarantees. IJCAI.

[39] A. Seuret. 2010. Exponential stability and stabilization of sampled-data systems with time-varying period. In *IFAC Workshop on Time Delay Systems*.

[40] A. Seuret and M. M. Peet. 2013. Stability analysis of sampled-data systems using sum of squares. *IEEE Trans. Automat. Control* 58, 6 (2013), 1620–1625.

[41] F. Shmarov and P. Zuliani. 2015. Probreach: verified probabilistic delta-reachability for stochastic hybrid systems. In *HSCC*. ACM, 134–139.

[42] D. Soudbakhsh, L. TX. Phan, A. M. Annaswamy, and O. Sokolsky. 2016. Co-design of arbitrated network control systems with overrun strategies. *IEEE Transactions on Control of Network Systems* (2016).

[43] Y. Sun and M. D. Natale. 2017. Weakly Hard Schedulability Analysis for Fixed Priority Scheduling of Periodic Real-Time Tasks. *TECS* 16, 5s (2017).

[44] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. R. Smith. 2015. Powering the Next Billion Devices with Wi-fi. In *CoNEXT*. ACM, 4:1–4:13.

[45] U. Topcu, A. K. Packard, P. Seiler, and G. J. Balas. 2010. Robust region-of-attraction estimation. *IEEE Trans. Automat. Control* 55, 1 (2010), 137–142.

[46] G. Valmorbida and J. Anderson. 2014. Region of attraction analysis via invariant sets. In *ACC*. IEEE, 3591–3596.

[47] Gera Weiss and Rajeev Alur. 2007. Automata based interfaces for control and scheduling. In *HSCC*. Springer, 601–613.

[48] W. Xu, Z. A. H. Hammadeh, A. KrÃűller, R. Ernst, and S. Quinton. 2015. Improved Deadline Miss Models for Real-Time Systems Using Typical Worst-Case Analysis. In *ECRTS*. 247–256.

[49] Z. Yang, C. Huang, X. Chen, W. Lin, and Z. Liu. 2016. A linear programming relaxation based approach for generating barrier certificates of hybrid systems. In *FM*. Springer, 721–738.

[50] W. Zhang and L. Yu. 2010. Stabilization of Sampled-Data Control Systems With Control Inputs Missing. *IEEE Trans. Automat. Control* 55, 2 (2010), 447–452.

[51] H. Zhao, N. Zhan, D. Kapur, and K. G. Larsen. 2012. A hybrid approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In *FM*. Springer, 471–485.

[52] B. Zheng, W. Li, P. Deng, L. Gerard, Q. Zhu, and N. Shankar. 2015. Design and Verification for Transportation System Security. In *DAC*.

A COMPLETE EXPERIMENT RESULTS

The complete experiment results of four examples shown in Section 7 under four weakly hard cases with different (m, K) constraints are shown in Table 3.

Table 3: Safe radius r_0 for Examples 1-4

Example 1			K														
type			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Case 1	m	1	-	-	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947
Case 2	m	1	-	-	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947	1.947
		2	-	-	-	-	-	-	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.962	0.962
		3	-	-	-	-	-	-	-	-	-	-	0.352	0.352	0.352	0.352	0.352
Case 3	m	1	-	-	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995	3.995
		2	-	-	-	-	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.500
		3	-	-	-	-	-	-	-	1.759	1.759	1.759	1.759	1.759	1.759	1.759	1.759
		4	-	-	-	-	-	-	-	-	-	1.318	1.318	1.318	1.318	1.318	1.318
		5	-	-	-	-	-	-	-	-	-	-	-	0.953	0.953	0.953	0.953
		6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.824
Case 4	m	1	-	-	3.995	3.995	3.995	3.996	3.996	3.996	3.996	3.996	3.996	3.996	3.996	3.996	3.996
		2	-	-	-	-	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309	3.309
		3	-	-	-	-	-	-	-	2.741	2.741	2.741	2.741	2.741	2.741	2.741	2.741
		4	-	-	-	-	-	-	-	-	-	2.270	2.270	2.270	2.270	2.270	2.270
		5	-	-	-	-	-	-	-	-	-	-	-	1.880	1.880	1.880	1.880
		6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1.557
Example 2			K														
type			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Case 1	m	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Case 2	m	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Case 3	m	1	-	-	-	-	-	-	-	0.269	0.269	0.269	0.269	0.269	0.269	0.269	0.269
		2	-	-	-	-	-	-	-	-	-	-	-	-	-	0.065	0.065
Case 4	m	1	-	-	-	-	-	-	-	0.269	0.269	0.269	0.269	0.269	0.269	0.269	0.269
		2	-	-	-	-	-	-	-	-	-	-	-	-	-	0.065	0.065
Example 3			K														
type			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Case 1	m	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Case 2	m	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Case 3	m	1	-	-	-	-	-	-	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.326
		2	-	-	-	-	-	-	-	-	-	-	-	0.071	0.071	0.071	0.071
Case 4	m	1	-	-	-	-	-	-	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.326
		2	-	-	-	-	-	-	-	-	-	-	-	0.071	0.071	0.071	0.071
Example 4			K														
type			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Case 1	m	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Case 2	m	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Case 3	m	1	-	-	-	-	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053
		2	-	-	-	-	-	-	-	-	0.388	0.388	0.388	0.388	0.388	0.388	0.388
		3	-	-	-	-	-	-	-	-	-	-	-	-	0.143	0.143	0.143
Case 4	m	1	-	-	-	-	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053	1.053
		2	-	-	-	-	-	-	-	-	0.388	0.388	0.388	0.388	0.388	0.388	0.388
		3	-	-	-	-	-	-	-	-	-	-	-	-	0.143	0.143	0.143