# Combining Time and Frequency Domain Specifications For Periodic Signals.*

Aleksandar Chakarov[1], Sriram Sankaranarayanan [1],
Georgios Fainekos[2]
1. University of Colorado, Boulder, CO. first.lastname@colorado.edu
2. Arizona State University, Tempe, AZ. fainekos@asu.edu

August 26, 2011

**Abstract**

In this paper, we investigate formalisms for specifying periodic signals using time and frequency domain specifications along with algorithms for the signal recognition and generation problems for such specifications. The time domain specifications are in the form of hybrid automata whose continuous state variables generate the desired signals. The frequency domain specifications take the form of an "envelope" that constrains the possible power spectra of the periodic signals with a given frequency cutoff. The combination of time and frequency domain specifications yields mixed-domain specifications that constrain a signal to belong to the intersection of the both specifications.

We show that the signal recognition problem for periodic signals specified by hybrid automata is NP-complete, while the corresponding problem for frequency domain specifications can be approximated to any desired degree by linear programs, which can be solved in polynomial time. The signal generation problem for time and frequency domain specifications can be encoded into linear arithmetic constraints that can be solved using existing SMT solvers. We present some preliminary results based on an implementation that uses the SMT solver Z3 to tackle the signal generation problems.

## 1 Introduction

The combination of time and frequency domain specifications often arises in the design of analog or mixed signal circuits [16], digital signal processing systems [20] and control systems [3]. Circuits such as filters and modulators often specify time-domain requirements on the input signal. Common examples of time domain specifications include setup time and hold time requirements for flip-flops, the slew rate for clocks and bounds on the duty cycle for pulse width modulators [16]. Likewise, the behavior of many components are also specified in terms of their frequency responses. Such requirements concern the effect of a subsystem on the various frequency components of a input signal. The problem of combining these specification styles is therefore of great interest, especially in the runtime verification setting.

In this paper, we study models for specifying real-valued *periodic signals* using *mixed-domain specifications*. Such specifications combine commonly used automata-theoretic models that can specify the characteristics of a signal over time with frequency-domain specifications that constrain

*This work was supported, in part, by the US National Science Foundation (NSF) awards CNS-1017074 and CNS-1016994.

the distribution of amplitude (or the power) of the sinusoidal components over some range of frequencies. Given such a mixed-domain specification, we consider the signal recognition and generation problems. The signal generation problem seeks test cases for an analog or a mixed-signal circuit from its input specifications. Since specifications are often non-deterministic, an *exhaustive generator* explores all the possible cases encoded in the specification by generating a set of representative signals. Likewise, the signal recognition or monitoring problem decides whether a given signal conforms to specifications.

In this paper, we present an encoding that reduces both problems to constraints in linear arithmetic. While such an encoding is easily obtained time domain specifications, a naive encoding of the frequency domain constraints yields a system of non-linear constraints that are hard to solve. We demonstrate how such non-linear constraints can be systematically approximated to arbitrary precision using constraints from linear arithmetic. Finally, we present some preliminary results on a prototype implementation of our technique that uses the SMT solver Z3 to solve the resulting constraints [5]. Owing to space restrictions, we have omitted some of the finer details including proofs of key lemmas. An extended version containing proofs along with supplementary material containing the source code and models for our experiments are available upon request.

*Related Work* Automata, especially timed and hybrid automata, are quite natural formalisms for specifying the behavior of signals over time [1, 12]. Likewise, the study of Fourier transforms and power spectra of signals forms the basis for specifying analog and mixed signal systems [20]. The problem of matching observations to runs for timed and hybrid automata was studied by Alur et al. [2]. Whereas Alur et al. study the problem of matching a trace consisting of a set of events generated by discrete transitions, the traces here are partial observations over the run, sampled discretely. Therefore, while the timestamp generation problem is shown to be polynomial time by Alur et al., its analog in our setting is NP-complete.

Monitoring algorithms for discrete-time Boolean valued signals have been well-studied [26, 13, 10, 9, 7]. Such specifications can capture Boolean abstractions of discrete-time signals sampled over the output signals generated by hybrid/embedded systems. An off-line algorithm for temporal logic analysis of continuous-time signals was proposed by Nickovic et al. [15] and extended to an on-line algorithm [19]. Thati et al. [26] and Kristoffersen et al. [13] presented algorithms for monitoring timed temporal logics over timed state sequences. While fragments temporal logics and a restricted class of automata are well known to be efficiently monitorable, it is not easy to express properties of oscillators such as periodicity, rise times, duty cycles and bounds on derivatives in these fragments without introducing extraneous constraints or quantifiers. Fainekos et al. [6], considered the problem of monitoring continuous-time temporal logic properties of a signal based solely on discrete-time analysis of its sampling points. Tan et al. [24, 25] consider hybrid automaton specifications for synthesizing monitors for embedded systems, wherein the monitor's execution is synchronized with the model of the system during run-time. Specification and verification of the periodicity of oscillators has been considered by Frehse et al. [8] and Steinhorst et al.[23].

On the other hand, specification formalisms for frequency domain properties of systems have not received as much attention. Hedrich et al. [11] study the problem of verifying frequency domain properties of systems with uncertain parameters. Our encoding for frequency domain specifications is similar to techniques used in regression, wherein the goal is to find a function from a given family that best fits a given set of points, wherein the "best fit" can be defined as the sum of the distances between the data points and the function under some norm. The connection between regression and optimization is discussed in many standard textbooks on convex optimization [4].

## 2 Signals and Automata

Let $\mathbb{R}$ denote the set of real numbers. A signal $f(t)$ is a function $f : \mathbb{R} \mapsto \mathbb{R}$. A signal is *periodic* iff there is a time period $T > 0$ such that for all $t \geq 0$, $f(t+T) = f(t)$. Let $\Sigma$ represent the set of all signals $f : \mathbb{R} \mapsto \mathbb{R}$. Note that in most applications, the domain of a signal is the continuous time domain $t \in \mathbb{R}_{\geq 0}$. Let $\vec{\tau} = \langle t_0, t_1, \ldots, t_k \rangle$ be some set of time instants such that $0 \leq t_0 < t_1 \ldots < t_k$. A sample of a signal $f$ at the time instants $\vec{\tau}$ is given by $f(\vec{\tau}) = \langle f(t_0), f(t_1), \ldots, f(t_k) \rangle$.

*Hybrid Automaton:* Our discussion will focus mostly on hybrid automata with dynamics specified by rectangular differential inclusions.

**Definition 2.1** (Linear Hybrid Automata)**.** A *Linear Hybrid Automaton* $\mathcal{H}$ consists of a tuple $\langle Q, \vec{x}, \mathcal{T}, \mathcal{D}, \mathcal{I}, q_0, \Theta \rangle$:

1. $Q$ is a finite set of *discrete modes*,
2. $\vec{x}$ is a vector of finitely many continuous system variables.
3. $\mathcal{T}$ is a set of *discrete transitions*. Each transition $\tau \in \mathcal{T}$ is a tuple $\tau : \langle s, t, \rho_\tau \rangle$ where $s, t \in Q$ are the pre- and the post-modes respectively and $\rho_\tau[\vec{x}, \vec{x}']$ is a *transition relation* that relates the current value of $\vec{x}$ with the next state values $\vec{x}'$.
4. $\mathcal{D}$ maps each $q \in Q$ to a rectangular differential inclusion $\vec{\ell}(q) \leq \frac{d\vec{x}}{dt} \leq \vec{u}(q)$.
5. $\mathcal{I}$ maps each mode $q \in Q$ to a *mode invariant* set $\mathcal{I}(q)$.
6. $q_0$ is the start state and $\Theta$ is a logical assertion over $\vec{x}$ that specifies the initial conditions for the continuous variables.

A state of the hybrid automaton is a pair $(s, \vec{x})$ consisting of a discrete mode $s \in Q$ and a continuous state $\vec{x} \in \mathcal{I}(q)$. The semantics of a hybrid automaton are defined in terms of runs. In this paper, we will describe periodic signals by means of finite runs of a hybrid system.

**Definition 2.2** (Runs)**.** A *finite run* of a linear hybrid automaton $\mathcal{H}$ is a finite sequence of states and *actions*: $\sigma : (s_0, \vec{x}_0) \xrightarrow{a_1} (s_1, \vec{x}_1) \xrightarrow{a_2} (s_2, \vec{x}_2) \xrightarrow{a_3} \cdots \xrightarrow{a_N} (s_N, \vec{x}_N)$, wherein each action $a_i$ is of the form $\tau$ for some discrete transition or $(\mathsf{tick}(\delta_i), f_i)$, for some time interval $\delta_i \geq 0$ and function $f_i : [0, \delta_i) \mapsto \mathbb{R}^n$, such that:

- If action $a_i$ is a discrete transition $\tau_i$ then $\tau_i$ must be of the form $\langle s_{i-1}, s_i, \rho_i \rangle$ (i.e, the transition must take us from state $s_{i-1}$ to state $s_i$) and $(\vec{x}_{i-1}, \vec{x}_i) \models \rho_i$, i.e., the continuous variables change according to the transition relation.
- If $a_i$ is a "tick" of the form $(\mathsf{tick}(\delta_i), f_i)$, wherein $s_i = s_{i-1}$ (i.e., no mode change can occur). The function $f_i : [0, \delta_i] \mapsto \mathbb{R}^n$ is a continuous and piecewise differentiable function such that: (1) $f_i(0) = \vec{x}_i$, $f_i(\delta_i) = \vec{x}_{i+1}$, (2) $f_i(t)$ satisfies the mode invariant $\mathcal{I}(s_i)$ for all $t \in [0, \delta)$, and (3) $\frac{df_i}{dt} \in [\ell(s_i), u(s_i)]$ at all instances $t \in [0, \delta)$ where $f_i$ is differentiable.

**Example 2.1.** Consider the following signal specification for a square wave generator: (1) The signal has two stable phases: high $(5 \pm 0.5V)$ or low $(-5 \pm 0.5V)$. (2) If the signal transitions from one phase to another, the value of $v$ at the start of the transition must be in the range $[-4.6, 4.6]$. (3) The signal remains a minimum of 0.5 seconds in each mode. (4) The rate of signal rise during transition from low to high lies within $[5, 7.5]V/s$. (5) The rate of signal fall during transition from

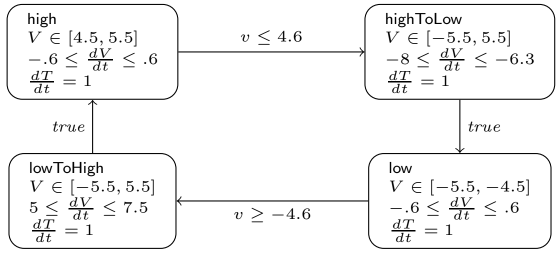


Figure 1: Hybrid automaton model for example signal specification.

high to low lies within $[-6.3, -8]V/s$. (6) In any stable phase, the rate of change lies between $[-.6, .6]V/s$.

Figure 1 shows a hybrid automaton that specifies the signal. The modes high and low specify the stable phases for the signal. Similarly, the modes highToLow and lowToHigh represent the transitions.

# 3 Periodic Signals In Time Domain

We will now explore the use of hybrid automata with piecewise constant dynamics to specify periodic signals. We will observe that the problem of checking if a sampled signal can be generated by some run of a hybrid automaton is NP-Complete. In fact, the problem of checking if *a given path through the automaton* generates the samples of a given signal is itself NP-complete. As a result, barring restrictions, linear hybrid automata by themselves are too rich a formalism for use in monitoring of signals. Thereafter, we focus on signal generation, presenting techniques for generating runs using a systematic exploration of the state-space of the automaton using LP solvers.

We augment the basic hybrid automaton by designating a set of modes as *final modes* and an output function $\vec{y} = f(\vec{x})$ that specifies the output signal as a function of the continuous state variables. Additionally, we require that the runs of the automaton $\sigma : (\vec{x}_0, s_0) \rightarrow (\vec{x}_1, s_1) \rightarrow \cdots \rightarrow (\vec{x}_N, s_N)$, satisfy the following constraints:

1. There is a *minimum dwell time* $\delta_{\min}$ for each mode such that whenever a run enters a mode $q$, it will remain in that mode for time at least $\delta_{\min}$ before taking a transition.

2. The terminal mode $s_N \in F$.

3. The initial state $(s_0, \vec{x}_0)$ and the terminal state $(s_N, \vec{x}_N)$ yield the same output $f(\vec{x}_0) = f(\vec{x}_N)$, so that the signal is periodic.

The minimum dwell time requirement seems quite natural for signal specifications, and furthermore, it considerably simplifies the complexity of signal membership checking and generation problems that we will discuss subsequently (also Cf. [2]). As a result of the requirements above, the output $\vec{y}(t)$ obtained on any finite run of the automaton can be thought of as constituting a single period of the signal. Repeating this output with time shifted yields the overall periodic signal.

**Definition 3.1** (Time Domain Periodic Signal Specification)**.** A time domain period signal specification consists of a hybrid automaton $\mathcal{H}$ with a set of final modes $F \subseteq Q$, an output function $\vec{y} = f(\vec{x})$ and a minimum dwell time $\delta_{\min}$.

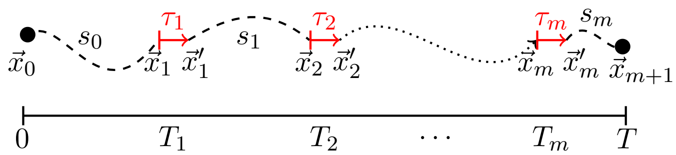


Figure 2: Run Encoding along a path $\pi$ with transitions $\tau_1, \ldots, \tau_m$.

| Constraints | Remarks |
|---|---|
| $\Theta[\vec{x}_0]$ | Initial condition |
| $f(\vec{x}_0) = f(\vec{x}_{m+1})$ | Periodicity of the trace |
| $\bigwedge_{i=1}^{m} \; T_i - T_{i-1} \geq \delta_{\min}$ | Minimum Dwell Time. |
| $\bigwedge_{k=1}^{m} \left( \begin{array}{c} \ell(\vec{s_k})(T_{k+1} - T_k) \; \leq \; (\vec{x}_{k+1} - \vec{x}'_k) \\ (\vec{x}_{k+1} - \vec{x}'_k \leq \; u(\vec{s_k})(T_{k+1} - T_k) \end{array} \right)$ | $\vec{x}'_k$ reachable from $\vec{x}_k$ in mode $s_k$ |
| $\bigwedge_{k=1}^{m} \left[ \mathcal{I}_{s_{k-1}}(\vec{x}_k) \; \wedge \; \mathcal{I}_{s_k}(\vec{x}'_k) \right]$ | Invariants for mode $s_k$ |

Figure 3: Constraints encoding the existence of a run along a path. **Note:** The guards, invariant sets, initial conditions of $\mathcal{H}$ are convex polyhedra. The function $f$ is affine.

### 3.1 Run Encoding

Let $\langle \mathcal{H}, F, f, \delta_{\min} \rangle$ be a hybrid automaton for a signal specification. Consider a syntactic path through $\pi : s_0 \xrightarrow{\tau_1} s_1 \xrightarrow{\tau_2} \cdots s_{m-1} \xrightarrow{\tau_m} s_m$ such that $s_0$ is initial, $s_m \in F$ and $m \leq \left\lfloor \frac{T}{\delta_{\min}} \right\rfloor$. We wish to encode the (possibly empty) set of runs that yield a periodic signal of time period $T$ along the path $\pi$ in terms of a linear program (LP) $\Psi_{T,\pi}$. We describe the variables that will be used in our encoding, as depicted in Figure 2. (A) $T_1, \ldots, T_m$ represent the transition times. We add two constants $T_0 = 0$ and $T_{m+1} = T$ to denote the start and end times of the trace, respectively. (B) $\vec{x}_0$ and $\vec{x}_{m+1}$ denote the initial and terminal values for continuous variables. (C) $\vec{x}_1, \vec{x}'_1, \ldots, \vec{x}_m, \vec{x}'_m$ encode the continuous states before and after each of the $m$ discrete transitions. The overall encoding is a conjunction of linear inequalities as described in Figure 3. This encoding is similar to the timestamp generation encoding provided by Alur et al. [2].

Note that the encoding yields a linear program $\Psi_{T,\pi}$, assuming that all transition relations, mode invariants are polyhedral and the output function $f$ is affine. Note that models of $\Psi_{T,\pi}$, if they exist, do not fully specify a run of the hybrid automaton. A run $\sigma$ of $\mathcal{H}$ *corresponds* to a model $(\vec{x}_0, \vec{x}'_1, T_1, \vec{x}_1, \ldots, \vec{x}'_m, T_m, \vec{x}_m, \vec{x}_{m+1})$ of $\Psi_{T,\pi}$ if the initial, terminal states, switching times and states before/after the discrete transitions of $\sigma$ coincide with those specified by the model.

**Theorem 3.1.** *The encoding of a run $\Psi_{T,\pi}$ is a linear assertion such that (a) each model of $\Psi_{T,\pi}$ corresponds to a run $\sigma$ of duration $T$, and (b) conversely, every run $\sigma$ of duration $T$ along the path $\pi$ corresponds to a model of $\Psi_{T,\pi}$ .*

### 3.2 Testing Membership

We first consider the problem of deciding signal membership given $N$ samples of periodic signal $g(t)$ with time period $T$, sampled at some fixed rate $\delta_s = \frac{T}{N}$ for a single time period. Let $g_0, \ldots, g_{N-1}$ be the signal values at times $0, \delta_s, \ldots, (N-1)\delta_s$, respectively. Since the signal is periodic, we have $g_N = g(N\delta_s) = g_0$. We assume that $\delta_s$ the sampling time, is strictly less than $\delta_{\min}$, the minimum dwell time.

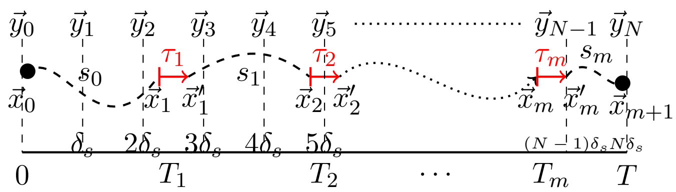


Figure 4: Encoding membership of a sampled trace.

We use the following strategy to search for a run $\sigma$ of the hybrid automaton $\mathcal{H}$ that coincides with the samples of $g(t)$.

1. Explore paths from $s_0$ to a final state $s_m \in F$ explicitly [1].
2. For each path $\pi$ with transitions $\tau_1, \ldots, \tau_m$, we encode the existence of a run along the path using $\Psi_{T,\pi}$, and
3. We conjoin $\Psi_{T,\pi}$ with a formula $\Gamma_{\pi,g}$ that encodes that the samples $g_0, \ldots, g_{N-1}$ conform to the run encoded in $\Psi$.

We encode the unknown continuous state at time $t = i\delta_s$ by variable $\vec{y}_i$. The encoding for $\Gamma_{\pi,g}$ will contain the following clauses:

*Continuous State and Output:* The signal value $g_i$ at $t = i\delta_s$, $i \in [0, N]$ must correspond to the continuous state: $f(\vec{y}_i) = g_i$.

*Mode change rule:* If a discrete transition happens between time $((i-1)\delta_s, i\delta_s)$ then $\vec{x}_j$ is reachable from $\vec{y}_{i-1}$ and likewise, $\vec{y}_i$ is reachable from $\vec{x}'_j$.

$$\bigwedge_{i,j=1}^{m} \left[ \begin{array}{c} (i-1)\delta_s \leq T_j \wedge \\ T_j < i\delta_s \end{array} \right] \Rightarrow \left[ \begin{array}{c} (i\delta_s - T_j)\ell(s_j) \leq (\vec{y}_i - \vec{x}'_j) \ \leq u(s_j)(i\delta_s - T_j) \ \wedge \\ (T_j - (i-1)\delta_s)\ell(s_{j-1}) \leq (\vec{x}_j - \vec{y}_{i-1}) \ \leq u(s_{j-1})(T_j - (i-1)\delta_s) \end{array} \right]$$

On the other hand, if no mode change happens in the interval $[(i-1)\delta_s, i\delta_s)$ then the mode at time $i\delta_s$ is the same as that at time $(i+1)\delta_s$. Furthermore, it is possible to reach the state $\vec{y}_i$ from $\vec{y}_{i-1}$ by evolving according to the dynamics at this mode:

$$\bigwedge_{i=1}^{N} \bigwedge_{j=1}^{m} \left[ \left( \begin{array}{c} T_j < (i-1)\delta_s \ \wedge \\ T_{j+1} \geq i\delta_s \end{array} \right) \right] \ \Rightarrow \ \left[ \ \delta_s \ell(s_j) \ \leq \ (\vec{y}_i - \vec{y}_{i-1}) \ \leq \ \delta_s u(s_j) \ \right]$$

*Simplifying the Encoding:* The encoding presented above can be simplified considerably by noting the minimum dwell time requirement on the runs. As a result of this requirement, we may deduce that the switching time for the $j^{th}$ transition $T_j$ must lie in the range $[j\delta_{\min}, T - (m+1-j)\delta_{\min}]$, wherein $\delta_{\min}$ is the minimum dwell time. As a result, some of the antecedents of the implications for the mode change rule are always false. This allows us to reduce the size of the encoding, in practice.

Let $g_0, \ldots, g_{N-1}$ be the signal samples at times $0, \delta_s, 2\delta_s, \ldots, (N-1)\delta_s$, wherein we assume that $\delta_s$ is smaller than the minimum dwell time. Let us assume that $\Gamma_{g,\pi}$ is the formula obtained over variables $\vec{x}_0, \ldots, \vec{x}_{m+1}$, $\vec{y}_0, \ldots, \vec{y}_N$, $T_1, \ldots, T_m$ using the encoding presented in this section.

[1] This search can also be encoded implicitly as a SAT formula.

**Theorem 3.2.** *The samples $g_0, \ldots, g_{N-1}$ of a periodic signal with sample time $\delta_s < \delta_{\min}$ are generated by some run of the hybrid automaton $\mathcal{H}$ if and only if the linear arithmetic formula $\Gamma_\pi \wedge \Psi_{T,\pi}$ is satisfiable for some path $\pi$ from an initial mode $s_0$ to a final mode $s_m \in F$ with $m \leq \lfloor \frac{T}{\delta} \rfloor$ discrete transitions.*

Given samples $g_0, \ldots, g_N$ of a signal, the algorithm thus far searches for a path $\pi$, a sequence of switching times and values of continuous states $\vec{x}_0, \ldots, \vec{x}_{m+1}$, $\vec{y}_0, \ldots, \vec{y}_N$ by solving a linear arithmetic formula using a SMT solver. Naturally, it is worth asking if there is an efficient algorithm for signal recognition using hybrid automata. We show that this is unlikely by proving the NP-completeness of the signal recognition problem. We observe the following surprising result for the seemingly simply problem of deciding if a given feasible path $\pi$ can yield a run generating the samples $g_0, \ldots, g_N$.

**Theorem 3.3.** *Let $g_0, \ldots, g_N$ be samples of a periodic signal $g(t)$ and $\pi$ be a path from initial to final mode in $\mathcal{H}$. Deciding if the given samples are generated by some run of along path $\pi$ is NP-complete.*

Membership in NP is clear from the SMT encoding to a linear arithmetic formula which can be solved by a non-deterministic polynomial time TM coupled with a LP solver which operates in polynomial time. The proof of NP-hardness is by reduction from CNF-SAT problem and is presented in an extended version of this paper available upon request. Our results show that significant restrictions are required on the linear hybrid automaton model to make it suitable for signal monitoring. For instance, such restrictions have to go beyond simply restricting the number of paths from the initial to the final mode.

## 3.3 Signal Generation

We will now consider the problem of generating signals *at random* from a given hybrid automaton specification. The signal generator explores all the paths in the hybrid automaton up to a depth bound. For each path $\pi$, the set of signals form a convex set given by the convex polyhedron $\Psi_{T,\pi}$ (Cf. Section 3.1). The notion of sampling uniformly at random from a convex set is defined rigorously in most standard textbooks [21]. Our generator samples a fixed number of solutions uniformly at random.

1. Systematically explore paths of length $m \leq \lfloor \frac{T}{\delta} \rfloor$ from initial to a final mode.
2. For each path $\pi$, encode the formula $\Psi_{T,\pi}$ to generate switching times and continuous state values $\vec{x}_i, \vec{x}'_i$ before and after transitions (Cf. Section 3.1).
3. Extract solutions uniformly at random from $\Psi_{T,\pi}$.
4. For each solution, generate sampled signals according the dynamics of each mode.

*Extracting Random Solutions from Linear Programs*

As shown in Section 3.1, let $\Psi_{T,\pi}$ be the LP for path $\pi$ over variables $\vec{x}_0, \vec{x}_1, \vec{x}'_1, \ldots, T_1, \ldots, T_m$ that we shall collectively refer to as $\vec{y}$. We assume that $\Psi$ is feasible. Our goal is to extract solutions at random from the polyhedron that represents all feasible solutions of $\Psi$. This is achieved by a simple Monte-Carlo sampling scheme known as hit-and-run sampling [21]. Let $\vec{y}_0$ be some feasible point in $\Psi$ obtained by using a LP solver. At each step, we generate a new solution $\vec{y}_{i+1}$, at random, from the current sample $\vec{y}_i$ (Cf. Fig. 5):

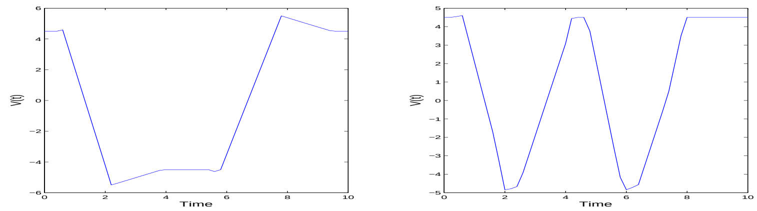


Figure 6: Periodic signals generated for the automaton in Example 2.1.

(1) Choose a random unit vector $\vec{v}$ uniformly. A simple scheme is to generate a vector $\vec{h}$ whose entries are uniform random numbers in $[0, 1]$ and compute $\vec{v} = \frac{1}{||\vec{h}||_2}\vec{h}$.
(2) Discover the interval $[l, u]$, such that $\forall \lambda \in [l, u]$, $\vec{y_i} + \lambda\vec{v} \in [[\Psi]]$. In other words, $\vec{v}$ yields a line segment containing the point $x$ along the directions $\pm\vec{v}$ and $[l, u]$ represent the minimum and maximum offsets possible along the direction $\vec{v}$ starting from $\vec{y_i}$. Since $[[\Psi]]$ is a polyhedron, bounds $[l, u]$ may be obtained by simply by substituting $\vec{x} \mapsto \vec{y_i} + \lambda\vec{v}$ in each inequality wherein $\lambda$ is an unknown. This yields upper and lower bounds on $\lambda$.
(3) Finally, we choose a value $\lambda \in [l, u]$ uniformly at random. The new solution sample is $\vec{y}_{i+1} = \vec{y_i} + \lambda\vec{v}$.

The analysis of this scheme and proof of convergence to the uniform distribution follows from the theory of Markov Chain Monte Carlo sampling [21, 22]. However, care must be taken to ensure that the polyhedron $\Psi$ is not *skewed* along some direction $\vec{r}$. In the worst case, we may imagine $\Psi$ as a straight line segment. In such cases, it is essential to ensure that random unit vectors at each step belong to any subspace that $\Psi$ itself is contained in. Finally, the scheme works best if the initial point $\vec{y}_0$ is an interior point. Lovasz et al. [14] analyze the convergence of hit-and-run samplers for generating uniformly distributed points belonging to a convex set.

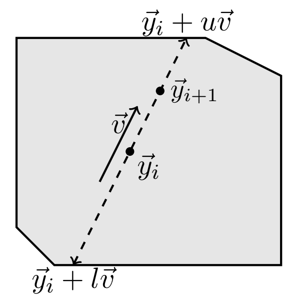


Figure 5: Hit-and-run sampling.

*From Switching Times To Sampled Signal*

Thus far, we have presented a scheme for encoding runs by means of a linear program $\Psi_{T,\pi}$ and choosing solutions at random efficiently from the polyhedron representing $\Psi$ by means of hit-and-run samplers. The next step is to construct signal samples $g_0, \ldots, g_{N-1}$ given the switching times $T_1, \ldots, T_m$, the continuous states $\vec{x}_0$, $\vec{x}_{m+1}$ at the beginning and end of the run, and the continuous states $\vec{x}_j, \vec{x}'_j$ before and after transition $\tau_j$, respectively.

Let $\delta_s$ be the sampling time. We will first generate the continuous state values $\vec{y}_0, \ldots, \vec{y}_N$ corresponding to the samples and thereafter, compute $g_i = f(\vec{y}_i)$.

From the switching times, it is known that all samples in the time interval $(T_j, T_{j+1})$ will belong to the mode $s_j$ (Cf. Figure 2). Our goal is to generate values $\vec{y}_i, \ldots, \vec{y}_{i+k}$ that lie between these time intervals, to ensure that (A) $\vec{y}_i$ is reachable from $\vec{x}'_j$ in time $i\delta_s - T_j$ evolving according to the mode $s_j$; (B) $\vec{y}_{i+l}$ for $1 \leq l \leq k$ is reachable from $\vec{y}_{i+l-1}$ in time $\delta_s$; and (C) $\vec{x}_{j+1}$ is reachable from $\vec{y}_{i+k}$.

Once again, these requirements can be encoded as a linear program since the dynamics at mode $s_j$ and the number of samples in the interval $(T_j, T_{j+1})$ are all known. We may then use hit-and-run sampler to choose values for the continuous variables $\vec{y}_i, \ldots, \vec{y}_{i+k}$ and thereafter, the signal samples by applying the function $f$.

**Example 3.1.** Consider the signal in Example 2.1. We will designate the state high as both the start and the end states. Figure 6 plots two signals that were generated using the models obtained for two paths $\pi_1, \pi_2$ of lengths 4 and 8 going around the cycle once and twice, respectively. For each path, we generate one solution for the switching times and one set of samples.

# 4 Frequency Domain Specifications

We will now consider the specification of periodic signals in the frequency domain by specifying constraints on its power spectrum. Let $g(t)$ be a continuous signal with time period $T > 0$. Its unique frequency domain representation can be derived by its Fourier series representation:

$$g(t) = a_0 + \sum_{k=1}^{\infty} \left( a_k \sin\left(\frac{2k\pi t}{T}\right) + b_k \cos\left(\frac{2k\pi t}{T}\right)\right)$$

The coefficient $a_0$ represents D.C component of the signal and coefficients $a_k, b_k$ represent the amplitude variable for the components at frequency $f = \frac{k}{T} = kf_0$. We will term $f_0 = \frac{1}{T}$ as the fundamental frequency. The amplitude at frequency $f_k = kf_0$ is given by $\sqrt{a_k^2 + b_k^2}$.

Let $G : [0, f_{\max}] \mapsto \mathbb{R}_{\geq 0}$ be a function mapping each frequency $f \in [0, f_{\max}]$ to a non-negative number $G(f)$. We assume that $G$ is a computable function so that $G(f)$ can be computed for any given $f$ to arbitrary precision. The function $G$ along with the maximum frequency $f_{\max}$ are said to form a *power spectral envelope*. Consider periodic signal $g(t)$ with fundamental frequency $f_0$ and Fourier coefficients $a_0, a_1, b_1, \ldots, a_n, b_n$.

**Definition 4.1** (Membership in Power Spectral Envelope)**.** The signal $g$ belongs to the power spectral envelope $\langle f_{\max}, G\rangle$, defined by $G : [0, f_{\max}] \mapsto \mathbb{R}_{\geq 0}$ if and only if:

1. The amplitudes vanish for all frequency components in $(f_{\max}, \infty)$: $\forall\ k \in \mathbb{N},\ (k \cdot f_0 > f_{\max}) \Rightarrow a_k = b_k = 0$.

2. The amplitudes for all frequency components in $(0, f_{\max}]$ are bounded by $G(f)$:

$$\forall\ k \in \mathbb{N},\ 0 < kf_0 < f_{\max}\ \Rightarrow\ \sqrt{a_k^2 + b_k^2} \leq G(kf_0)\,.$$

   In other words, the possible values of $a_k, b_k$ lie inside a circle of radius $G(kf_0)$ centered at $(0, 0)$.

3. The D.C component is bounded by $G(0)$, i.e, $-G(0) \leq a_0 \leq G(0)$.

In many situations, we are interested in signals being approximated within some tolerance limit by a signal that belongs to a given power spectral envelope $\langle f_{\max}, G\rangle$. Therefore, we define membership with $\epsilon$-tolerance for some $\epsilon \geq 0$.

**Definition 4.2** (Membership with $\epsilon$-tolerance)**.** A signal $s(t)$ satisfies $\langle f_{\max}, G\rangle$ with a tolerance $\epsilon \geq 0$ iff $s$ has a time period $T$ and there exists a signal $g$ that satisfies the frequency domain specification $\langle f_{\max}, G\rangle$ such that the distance between $s$ and $g$ is bounded by $\epsilon$, i.e, $(\forall\ t \in [0, T]),\ |s(t) - g(t)| \leq \epsilon$.

Let $\delta_s$ be a sampling time period. We say that $s(t)$ satisfies to a specification with a *sample tolerance* of $\epsilon$ iff $|s(k\delta_s) - g(k\delta_s)| \leq \epsilon,\ \forall\ k \in [0, \left\lfloor \frac{T}{\delta_s} \right\rfloor]$.

It is possible to relate continuous time tolerance to sample tolerance, provided absolute bounds may be placed on the derivatives of the signals $s$ and $g$.

**Theorem 4.1.** *Let $s, g$ be two signals with sample distance of $\epsilon$ and sample time $\delta_s$. Let $|\frac{ds}{dt}| \leq D_s$ and $|\frac{dg}{dt}| \leq D_g$. For all $t \geq 0, |g(t) - s(t)| \leq \epsilon + \frac{\delta_s}{2}(D_s + D_g)$.*

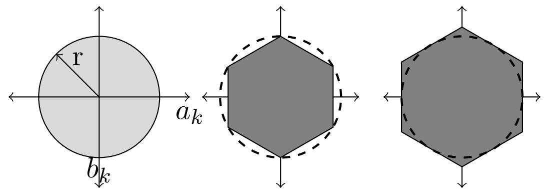


Figure 7: Relaxations and restrictions of amplitude constraint by polyhedral constraints.

A proof is provided in the extended version. Likewise, we prove that any signal belonging to a frequency domain specification $\langle f_{\max}, G\rangle$ has absolute bounds on its derivative.

**Theorem 4.2.** *The derivative of a signal $s$ with time period $T > 0$, whose Fourier series representation belongs to $\langle f_{\max}, G\rangle$, is bounded:*

$$\left|\frac{ds}{dt}\right| \leq \pi G_{\max} f_{\max}(1 + T f_{\max})\,,\ \textit{where } G_{\max} = \mathsf{sup}_{0\leq f\leq f_{\max}} G(f)\,.$$

**Example 4.1.** Consider the function $G(f) = \begin{cases} 1+8f & f \in [0, 0.5] \\ 7-4f & f \in [0.5, 1] \\ 0 & f > 1 \end{cases}$. We specify the set of all periodic signals whose time periods are in the range $T \in [5, 100]$ seconds, belonging to the envelope $\langle 1Hz, G\rangle$ with a tolerance of 0.01.

## 4.1 Encoding Membership

Let $g$ be some periodic signal with time period $T > 0$, sampled with time period $\delta > 0$. We represent $s$ in terms of its $N = \frac{T}{\delta}$ samples $g_0, g_1, \ldots, g_{N-1}$ wherein $g_k = g(k\delta)$. The sampling frequency $\frac{1}{\delta}$ is assumed to be at least $2f_{\max}$, the Nyquist limit to enable reconstruction of the original signal from its samples [20]. We wish to ascertain whether $g$ belongs to a given power spectral envelope $\langle f_{\max}, G\rangle$, with a given sample tolerance of $\epsilon \geq 0$. Membership is encoded in terms of linear inequality constraints over the unknown coefficients of the Fourier series representation of the signal $g(t)$.

Let $f_0 = \frac{1}{T}$ be the fundamental frequency. We will assume that $f_0 < f_{\max}$ (otherwise, membership is trivial). Let $m = \left\lfloor \frac{f_{\max}}{f_0} \right\rfloor$ represent the total number of potentially non-zero frequency components. We introduce the variables $a_0, a_1, \ldots, b_m$. The encoding consists of the following constraints:

*Sample Tolerance:* We encode that at each time instant $t = j\delta$, where $0 \leq j < N$, $s_j$ is approximated by the Fourier series:

$$\bigwedge_{j=0}^{N-1} -\epsilon \leq \left( g_j - \sum_{k=1}^{m} [a_k \sin(2\pi k f_0 j\delta) + b_k \cos(2\pi k f_0 j\delta)] - a_0 \right) \leq \epsilon\,.$$

Note that since $j$ and $\delta$ are known, the values of the trigonometric terms can be computed to arbitrary precision. As a result, the constraints above are linear inequalities over the unknowns $a_0, a_1, b_1, \ldots, b_m$.

*D.C. Component:* We encode requirements on $a_0$, $-G(0) \leq a_0 \leq G(0)$

*Amplitude Constraint:* For each $k \in [1, m]$, we wish to encode $\sqrt{a_k^2 + b_k^2} \leq G(kf_0)$. However, such a constraint is clearly non-linear. We present linear approximations of this constraints such that if any solution can be found for the linear restriction, then the solution satisfies the amplitude constraint above.

Geometrically, the constraint $\sqrt{a_k^2 + b_k^2} \leq G(kf_0)$ encodes that the feasible values of $(a_k, b_k)$ belong to the circle centered at origin of radius $G(kf_0)$ (see Figure 7). Let $P(r)$ be a polygon that under-approximates the circle of radius $r$ centered at the origin, and $Q(r)$ be a polygon that over-approximates the unit circle. It is well-known [2] that such polygons can approximate the circle to any desired accuracy. Therefore, we may restrict the constraint above by linear constraints $(a_k, b_k) \in P(G(kf_0))$, or relax it by linear constraints $(a_k, b_k) \in Q(G(kf_0))$. The overall encoding yields a linear program by conjoining the constraints above. The under approximate encoding is given by choosing $(a_k, b_k) \in P(r_k)$, wherein $r_k = G(kf_0)$, whereas the over approximate encoding is given by choosing the constraints $(a_k, b_k) \in Q(r_k)$.

*Signal Recognition:* Given a power spectral envelope $\langle f_{\max}, G \rangle$, a time period $T$ and signal samples $g_0, \ldots, g_{N-1}$ with timestep $\delta$, let $U_\epsilon(f_{\max}, G, T, g, \delta)$ be the restricted system and $O_\epsilon(f_{\max}, G, T, g, \delta)$ represent the relaxed constraints.

**Theorem 4.3.** *If $U_\epsilon$ is satisfiable then the signal $g(t)$ belongs to $\langle f_{\max}, G \rangle$ with sample tolerance $\epsilon$. If $O_\epsilon$ is unsatisfiable, then the signal $g(t)$ does not belong to $\langle f_{\max}, G \rangle$ with sample tolerance $\epsilon$.*

*Signal Generation:* Signal generation uses the same encoding $(O_\epsilon, U_\epsilon)$ with $g_0, \ldots, g_{N-1}$ as unknown variables as opposed to known samples of a signal. Once again, the hit-and-run sampling scheme used for choosing solutions at random can be employed to generate multiple samples.

*Mixed Domain Specifications* The problem of signal recognition can be solved by considering signal membership individually, in the time and frequency domains.

The encodings presented can be combined to generate signals. Let us assume that we are interested in generating a signal $g(t)$ with a fixed time period $T$. We choose some fixed sampling interface $\delta_s$, satisfying the Nyquist sampling criteria such that $\delta_s < \frac{1}{2f_{\max}}$. Let $g_i, i \in [0, N-1]$ denote the unknown signal sample to be generated at time $i\delta_s$. Once again, we generate the encodings $\Psi_{T,\pi}$ along paths $\pi$ to generate switching times and states before/after switching (Cf. Section 3.1). Next we generate LP $\Gamma_{g,\pi}$ that encodes the time domain correspondence of the signal samples w.r.t the run along path $\pi$ (Cf. Section 3.2). The sampled values from $\Psi_{T,\pi}$ are used to simplify $\Gamma_{g,\pi}$. The overall signal samples are generated by picking solutions from the LP $\Gamma_{g,\pi} \wedge U_\epsilon$ using a hit-and-run sampler.

# 5 Experiments

We will now report on our implementation, as a preliminary proof-of-concept for the ideas in this paper and some initial experimental results using these ideas.

*Implementation:* Our implementation reads in a hybrid automaton specification along with a frequency domain specification. The envelope function $G$ is specified by pairs $f_j, G(f_j)$ for a finite set of frequencies $f_j$. The value of $G(f)$ for $f \in (f_j, f_{j+1})$ is computed by linear interpolation. Our implementation first searches over paths in the hybrid automaton from the initial to the final states, constructing the LP $\Psi_{T,\pi}$ for each path. If this is found to be feasible, our approach constructs a SMT formula $\Gamma$ that encodes the existence of a signal sample corresponding to $\pi$. Currently, our approach uses Yices to obtain a single solution. Once such a solution is obtained, we may use the hit-and-run sampler to obtain other solutions. In fact, this process does not need further calls to

[2] Going back to the Greek mathematician Archimedes and the ancient Egyptians before him!

Table 1: Running times for signal generation benchmarks with various sets of time periods and sampling times. Legend: **#M:** # discrete modes, **#Tr:** # transitions, **#Samp:** # samples per period, **TP:** Time Period, **#FC**: Fourier Coefficients, **Time:** Signal generation time (Seconds), **#Path**: Paths explored, **#Sat:** satisfiable paths.

| Name | Time | | Freq. | | | Time Domain Only | | | Time + Freq Domain | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | #M | #Tr | #Samp | TP | #FC | Time | #Path | #Sat | Time | #Path | #Sat |
| SquareWave | 4 | 4 | 10 | 10 | 7 | 0 | 5 | 1 | 0.2 | 5 | 1 |
| | | | 15 | 15 | 13 | .2 | 7 | 2 | 30 | 7 | 0 |
| | | | 20 | 20 | 13 | .5 | 10 | 3 | 1300 | 10 | 0 |
| PulseWidth | 6 | 8 | 10 | 10 | 7 | .7 | 10 | 8 | 6.9 | 10 | 8 |
| | | | 15 | 15 | 11 | 8.7 | 15 | 13 | 391 | 15 | 7 |
| | | | 20 | 20 | 13 | 71.5 | 20 | 15 | - | T/O | - |
| Sq+SawtoothWave | 8 | 12 | 10 | 10 | 21 | 2.7 | 255 | 127 | 4.9 | 255 | 40 |
| | | | 15 | 15 | 31 | 149 | 8191 | 4095 | 1097 | 8191 | 32 |
| | | | 20 | 20 | 41 | 6349 | 262143 | 131071 | - | T/O | - |
| RoomHeater | 5 | 6 | 40 | 76 | - | 136 | 38 | 4 | - | n/a | - |

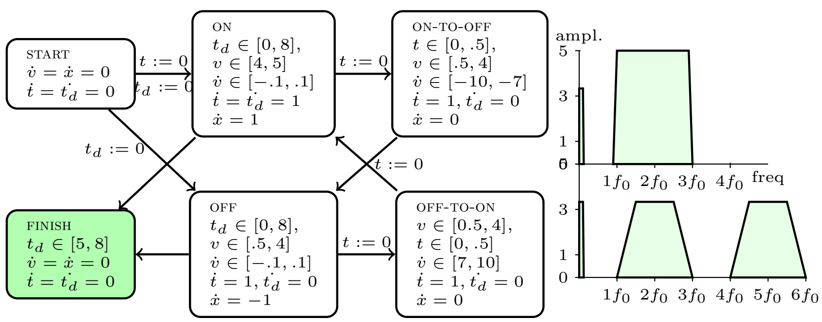


Figure 8: PWM signal time + frequency domain specification along with generated signals.

the solver. The alternative and potentially less expensive strategy of fixing a set of switching times by sampling from $\Psi_{T,\pi}$ and checking the conjunction of the time and frequency domain constraints remains to be implemented. The resulting samples are printed out in a suitable format that can be loaded into an environment such as Matlab. The encoding used in our implementation supports signal recognition as well.

We collected a set of benchmarks for commonly used specifications of various waveforms that are used in circuits including square waves that are commonly used to clock digital circuits (Cf. Example 2.1), sawtooth waves that are used in video monitors, the specification of a pulse-width modulator (PWM) waveform and a specification of an external disturbance temperature signal for testing the room heating benchmark available in Simulink/Stateflow(tm).

*Pulse-Width Modulator Waveform* Figure 8 shows time domain and frequency domain specifications for signals generated by a PWM waveform. The waveform consists of a square pulse represented by $v$ that alternates between on and off. An associated signal $x$ rises whenever the $v$ is high and falls when $v$ is low. In effect, $x$ represents the waveform $v$ by a sequence of 1s and 0s represented by $v$. We add two requirements (a) the % of time period $v$ must be high (also known as the duty cycle) must be between $50\% - 80\%$, and (b) the waveform $v$ must belong to one of the two power-spectral envelopes shown in Fig. 8(right). Note that while the former is a time

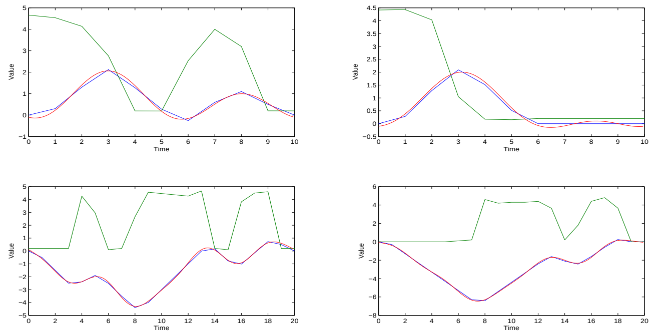


Figure 9: Some signals generated for the PWM specification in Figure 8. The time domain samples (blue) and the frequency domain samples (red) are overlaid on each other.

domain constraint on $v$, the latter is a frequency domain constraint on $x$. Fig 9 shows some of the waveforms output by our implementation. The sample tolerance between time and frequency domain signals was specified to be 0.1 and the sampling rate was chosen to be roughly $2.5 f_{\max}$ (slightly larger than the Nyquist rate).

Table 1 shows some of the results obtained by running the benchmark examples. Three of the examples have frequency domain specifications while the room heating benchmark had no frequency domain part. Overall, the benchmarks show that it is possible to exhaustively explore relatively small time domain specifications to obtain sample signals. Nevertheless, the complexity of exploration using SMT solvers is quite sensitive to the sampling rate. The addition of frequency domain constraints increases the complexity of these specifications many-fold. We believe that the handling of large floating point coefficients using exact arithmetic in tools such as Yices and Z3 is a bottleneck for frequency domain constraints and also to a limited extent for time domain constraints. A new generation of SMT solvers that combine the efficiency of floating point solvers with exact arithmetic solvers to guarantee the results may hold promise for tackling these constraints [17]. We are currently implementing strategies that avoid the use of SMT solvers by first fixing the transition timings by sampling from $\Psi_{T,\pi}$ and then finding if signal samples exist.

# 6 Conclusion

The overall goal of this paper was to explore the very first steps towards combining time domain and frequency domain specifications for mixed signal and DSP systems. In the future, we wish to consider restrictions of the time domain specifications for efficient monitoring. The generation of non-periodic signals by specifying the shape of their Fourier transforms is a natural next step. The results in this paper will be integrated into our ongoing work on Monte Carlo Methods for falsification of safety properties for hybrid systems [18].

# References

[1] R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126(2):183–235, 1994.

[2] R. Alur, R. P. Kurshan, and M. Viswanathan. Membership questions for timed and hybrid automata. In *RTSS'98*, pages 254–264. IEEE, 1998.

[3] K. Åstrom and R. M. Murray. *Feedback Systems: An Introduction for Engineers and Scientists.* Princeton University Press, 2005.

[4] S. Boyd and S. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. Online `http://www.stanford.edu/~boyd/cvxbook.html`.

[5] L. M. de Moura and N. Bjørner. Z3: An efficient SMT solver. In *TACAS*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008.

[6] G. E. Fainekos and G. J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009.

[7] B. Finkbeiner and H. Sipma. Checking finite traces using alternating automata. *Form. Methods Syst. Des.*, 24:101–127, Mar 2004.

[8] G. Frehse, B. H. Krogh, R. A. Rutenbar, and O. Maler. Time domain verification of oscillator circuit properties. *Electron. Notes Theor. Comput. Sci.*, 153:9–22, June 2006.

[9] M. Geilen. On the construction of monitors for temporal logic properties. In *Proceedings of the 1st Workshop on Runtime Verification*, volume 55 of *ENTCS*, pages 181–199, 2001.

[10] K. Havelund and G. Rosu. Monitoring programs using rewriting. In *Proceedings of the 16th IEEE international conference on Automated software engineering*, 2001.

[11] L. Hedrich and E. Barke. A formal approach to verification of linear analog circuits with parameter tolerances. In *Proceedings of the conference on Design, Automation and Test in Europe (DATE)*, pages 649–655, Washington, DC, USA, 1998. IEEE Computer Society.

[12] T. A. Henzinger. The theory of hybrid automata. In *LICS'96*, pages 278–292. IEEE, 1996.

[13] K. J. Kristoffersen, C. Pedersen, and H. R. Andersen. Runtime verification of timed LTL using disjunctive normalized equation systems. In *Proceedings of the 3rd Workshop on Runtime Verification*, volume 89 of *ENTCS*, pages 1–16, 2003.

[14] L. Lovasz. Hit-and-run is fast and run. *Mathematical Programming*, 86:443–461, 1999.

[15] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Proceedings of FORMATS-FTRTFT*, volume 3253 of *LNCS*, pages 152–166, 2004.

[16] J. Millman and C. C. Halkias. *Electronic Devices and Circuits*. McGraw-Hill Inc, 1967.

[17] D. Monniaux. On using floating-point computations to help an exact linear arithmetic decision procedure. In *CAV*, volume 5643 of *LNCS*, pages 570–583. Springer, 2009.

[18] T. Nghiem, S. Sankaranarayanan, G. E. Fainekos, F. Ivancic, A. Gupta, and G. J. Pappas. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. In *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, pages 211–220, 2010.

[19] D. Nickovic and O. Maler. Amt: A property-based monitoring tool for analog systems. In *FORMATS*, volume 4763 of *LNCS*, pages 304–319. Springer, 2007.

[20] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. Prentice Hall, 1975.

[21] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. Wiley Series in Probability and Mathematical Statistics, 2008.

[22] R. L. Smith. The hit-and-run sampler: a globally reaching markov chain sampler for generating arbitrary multivariate distributions. In *Proceedings of the 28th conference on Winter simulation*, pages 260–264. IEEE Computer Society, 1996.

[23] S. Steinhorst and L. Hedrich. Model checking of analog systems using an analog specification language. In *Proceedings of the conference on Design, automation and test in Europe*, DATE '08, pages 324–329, New York, NY, USA, 2008. ACM.

[24] L. Tan, J. Kim, and I. Lee. Testing and monitoring model-based generated program. In *Proceedings of the 3rd Workshop on Run-time Verification*, volume 89 of *ENTCS*, pages 1–21, 2003.

[25] L. Tan, J. Kim, O. Sokolsky, and I. Lee. Model-based testing and monitoring for hybrid embedded systems. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration*, pages 487–492, 2004.

[26] P. Thati and G. Rosu. Monitoring algorithms for metric temporal logic specifications. In *Runtime Verification*, volume 113 of *ENTCS*, pages 145–162. Elsevier, 2005.

# A Continuous Time Tolerance vs Sample Tolerance

In this section we present an auxiliary lemma to guarantee that the signal we produce does not exhibit any deviant behavior. The section culminates with a lemma that guarantees that no sudden spikes in the signal we generate occur between sample points.

**Definition A.1.** Let $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $r : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be two signals. Let $\delta_s$ be the time between samples. We say that $g$ and $r$ are within $\epsilon$ sampling tolerance if at every sample time $i\delta_s$: $|g(i\delta_s) - r(i\delta_s)| \leq \epsilon$.

Unfortunately, this constraint is not sufficient to ensure the quality of the generated signal. Although two signals may be close to each other at sample points, they move arbitrarily far apart at intermediate point.

In order to ensure that the overall tolerance *between* sample points is bounded, we will impose restrictions on the derivatives of the signals. For any signal $g(t)$ we will require that at any time $t_0 \in \mathbb{R}_{\geq 0}$ the derivative $\frac{\mathrm{d}g}{\mathrm{d}t}$ is in the range $[-D_g, D_g]$ for some real constant $D_g$.

**Claim A.1.** *Let $g(t)$ and $r(t)$ be two signals within $\epsilon$ sampling tolerance with bounded derivatives $-D_g \leq \frac{dg}{dt} \leq D_g$ and $-D_r \leq \frac{dr}{dt} \leq D_r$ for some real constants $D_g$ and $D_r$. Then the* difference *signal $(g - r)(t)$ is bounded from above by $\frac{\delta_s}{2}(D_r + D_g) + \epsilon$ and from below by $-\frac{\delta_s}{2}(D_r + D_g) - \epsilon$.*

*Proof.* Without loss of generality, the claimed upper bound follows from the fact that the greatest increase in the differential signal $(g - r)(t)$ occurs when $g(t)$ and $r(t)$ begin $\epsilon$ distance apart, $g(t)$ increases at maximum rate $D_g$ for $\delta_s/2$ time then decreases at $-D_g$ rate while $r(t)$ decreases at maximum rate $-D_r$ then increases at rate $D_r$ so that the signals end up $\epsilon$ distance apart at the next sample point.

Since $(g - r)(t)$ increases at maximum rate $D_g + D_r$ half of the time and decreases for the rest, the difference signal achieves a maximum of $\frac{\delta_s}{2}(D_g + D_r) + \epsilon$.

Analogously, the signal achieves an inter-sample point minimum of $-\frac{\delta_s}{2}(D_g + D_r) - \epsilon$. □

This result guarantees a bound on the differential signal provided that the derivatives are bounded. Next, we state a lemma that guarantees some bounds on the derive of a signal based on the envelope frequency spectrum function.

**Lemma A.1.** *Let $g(t)$ be a signal and $G$ be the envelope spectrum function with fundamental frequency $f_0$ and maximum cutoff frequency $f_{max}$. Let $G_{max}$ denote the maximum value achieved by $G$. Then the derivative of $g(t)$ is in the range:*

$$-\frac{\pi G_{max} f_{max}}{f_0}(f_{max} + f_0) \leq \frac{dg}{dt} \leq \frac{\pi G_{max} f_{max}}{f_0}(f_{max} + f_0)$$

*Proof.* Begin with the Fourier series expansion for $g$:

$$\begin{aligned} g(t) &= a_0 + \sum_{k=1}^{m}(a_k \sin(2\pi k f_0 t) + b_k \cos(2\pi k f_0 t)) \\ &= a_0 + \sum_{k=1}^{m} \sqrt{a_k^2 + b_k^2} \sin(2\pi k f_0 t + \phi) \end{aligned}$$

where $m = f_{max}/f_0$ and $\phi = \begin{cases} \arcsin\left(\frac{b_k}{\sqrt{a_k^2+b_k^2}}\right), & \text{if } a_k \geq 0 \\ \pi - \arcsin\left(\frac{b_k}{\sqrt{a_k^2+b_k^2}}\right), & \text{if } a_k < 0 \end{cases}$

Note that the last step is obtained by repeated application of the trigonometric identity.

Let us now consider the derivative $\frac{\mathrm{d}g}{\mathrm{d}t}$:

$$\begin{aligned}
\frac{\mathrm{d}g}{\mathrm{d}t} &= \sum_{k=1}^{m} \sqrt{a_k^2 + b_k^2}(2\pi k f_0)\cos(2\pi k f_0 t + \phi) \\
&\text{where } \sqrt{a_k^2 + b_k^2} \leq G_{max}, \text{ for all } k = 1, \ldots, m \\
&\leq \sum_{k=1}^{m} G_{max}(2\pi k f_0)\cos(2\pi k f_0 t + \phi) \\
&= G_{max}(2\pi k f_0)\sum_{k=1}^{m} k\cos(2\pi k f_0 t + \phi) \\
&\text{where } -1 \leq \cos(2\pi k f_0 t + \phi) \leq 1 \text{ for all } k, \text{ and } \sum_{k=1}^{m} k = \frac{m(m+1)}{2} \\
&\leq G_{max} 2\pi f_0 \frac{m(m+1)}{2} \\
&= \frac{\pi G_{max} f_{max}}{f_0}(f_{max} + f_0).
\end{aligned}$$

Analogously, we can show the lower bound holds.
Therefore,

$$-\frac{\pi G_{max} f_{max}}{f_0}(f_{max} + f_0) \leq \frac{\mathrm{d}g}{\mathrm{d}t} \leq \frac{\pi G_{max} f_{max}}{f_0}(f_{max} + f_0)$$

. □

Together LemmaA.1 and ClaimA.1 guarantee that a signal stays within a certain range between sample point and no sudden (unbounded) spikes are allowed.

# B NP-Completeness of Path Signal Recognition

Let $(\mathcal{H}, F, f)$ be a time domain specification and $g_0, \ldots, g_{N-1}$ be the samples of a periodic signal $g(t)$ with time period $T$ and sampling rate $\delta$. Let $\pi$ be a path

$$\pi : s_0 \xrightarrow{\tau_1} s_1 \xrightarrow{\tau_2} \cdots \xrightarrow{\tau_m} s_m$$

from the initial mode of $\mathcal{H}$ to a final mode $s_m \in F$. The Path Signal Recognition problem asks if there exist switching times for taking the transition and a run $\sigma$ along path $\pi$ such that generates the signal $g$. Formally, we specify the problem as follows:

**Inputs:** Time domain specification $(\mathcal{H}, F, f)$, path $\pi$ and signal samples $g_0 = g(0), g_1 = g(\delta), \ldots, g_{N-1} = g((N-1)\delta)$ with period $T = N\delta$.

**Output:** Yes, if there is a set of transition timings $T_1, \ldots, T_m$, valuations to continuous variables $\vec{x}_i, \vec{x}_i'$ before and after taking transition $\tau_i$, initial state $\vec{x}_0$, terminal state $\vec{x}_{m+1}$ and valuations to continuous state variables $\vec{y}_0, \ldots, \vec{y}_N$ such that $f(\vec{y}_i) = g(i\delta) = g_i$. No, otherwise.

**Theorem B.1.** *The Path Signal Recognition Problem belongs to the class NP.*

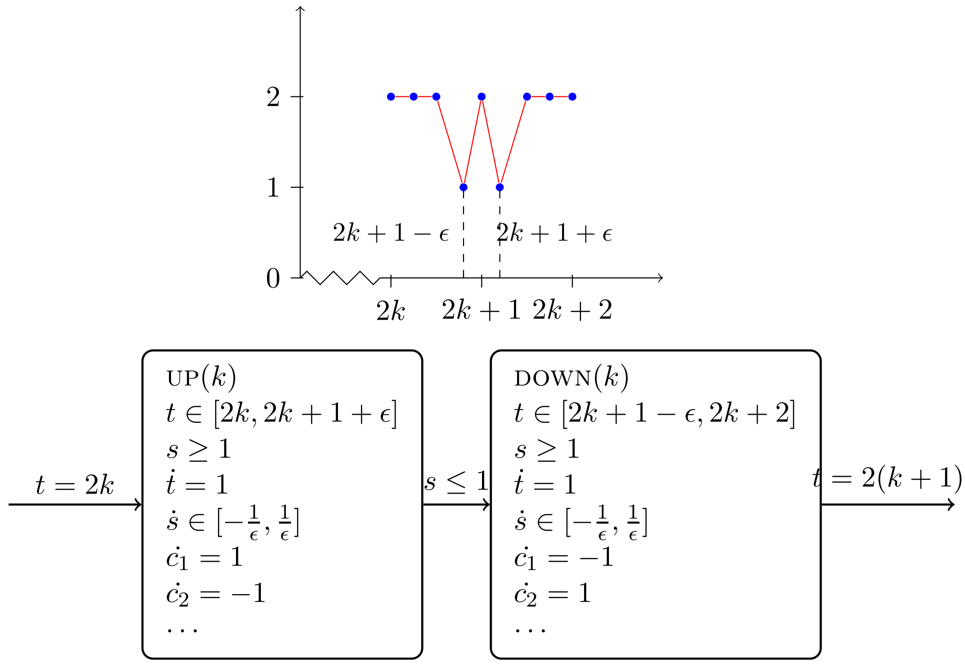


Figure 10: Signal fragment and modes of automaton corresponding to the variable selection gadget.

*Proof.* Proof follows directly from the encoding to linear arithmetic constraints in Section 3.2. It is easy to show that any formula $\varphi$ that consists of arbitrary Boolean combination of linear arithmetic constraints can be solved as follows:

- Assign truth values to the atomic formulae, non-deterministically.
- Use an LP solver to check that the assignments are indeed consistent.

By the direction of reduction, we conclude membership of the signal recognition problem in the class NP. ☐

**Theorem B.2.** *The Path Signal Recognition Problem is NP-hard.*

*Proof.* The overall proof is by reduction from 3-CNF-SAT.Let $\varphi$ be a given 3-CNF-SAT problem with variables $x_0, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. Each clause has at most distinct literals, wherein each literal is of the form $x_i$ or its negation $\overline{x_i}$.

Our proof will create a path $\pi$ with $2n+3$ modes and transitions, and signal samples $g$, sampled at some fixed sampling rate $\epsilon = \frac{1}{K}$ for some large enough integer constant $K > 20n$ (say). The time period of the signal is $2n+3$ with a minimum dwell time smaller than $1-\epsilon$.

The discrete modes are as follows:

1. Modes DOWN$(k)$ and UP$(k)$ that will be used to represent variable choices for the variable $x_k$ where $k \in [0, n]$.
2. Final mode $m$.

The continuous variables include:

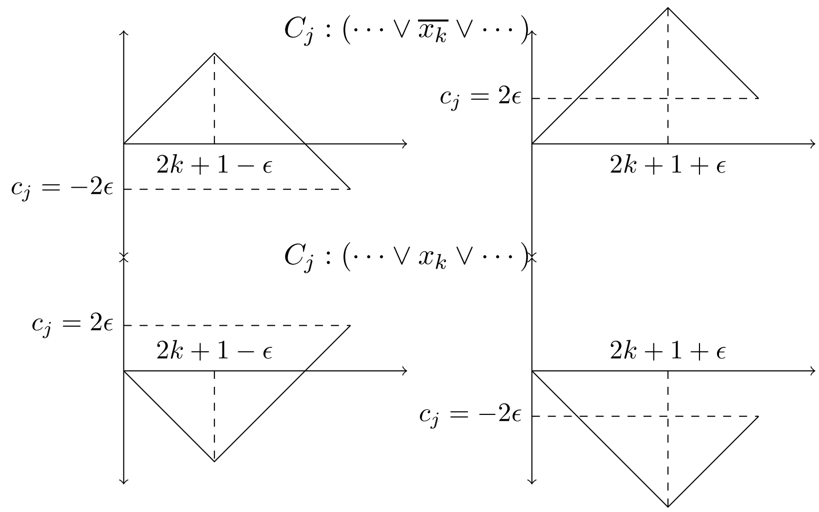


Figure 11: Effect of transition switching time for gadget $x_k$ on clause variable $C_j$. [**top (left,right)**] $\overline{x_k}$ is part of $C_j$ and [**bottom (left,right)**]) $x_k$ is part of $C_j$. Likewise, [**left (top,bottom)**] effect of assigning $x_k := \mathit{true}$ by switching at $t = k+1-\epsilon$ and [**right (top,bottom)**] effect of assignming $x_k := \mathit{false}$ by switchign at time $t = k+1+\epsilon$.

1. Clause variables $c_1, \ldots, c_m$, wherein $c_i$ represents the $i^{th}$ clause.

2. Variable $t$ that acts as a timer.

3. Variable $s$ that corresponds directly to the output signal, i.e, the output function $f(t, s, c_1, \ldots, c_m) = s$.

The signal samples are chosen as follows:

- For any time interval $t \in [2k, 2k+2]$ where our construction will mimic truth value choice for variable $x_k$, the signal samples are all $g_l = 2$ except for samples at two times $2k+1-\epsilon$ and $2k+1+\epsilon$ where $g_l = 1$.

- For the last one second in its period $t \in [2n+2, 2n+3]$ all samples are $g_l = 2$.

The signal samples are so designed that a transition can either occur at time $2k+1-\epsilon$ or $2k+1+\epsilon$ where the value of the signal dips to 1 but not both. We will use this to model the truth assignment choice for $x_k$ throught the *variable selection gadget*.

Figure 10 shows the variable selection gadget for variable $x_k$ where $0 \leq k \leq n$. Informally, the gadget consists of two modes $\textsc{up}(k)$ and $\textsc{down}(k)$. The guards coming in and out of the gadget ensures that

1. The value of timer entering $\textsc{up}(k)$ is precisely $t = 2k$.

2. Exactly two seconds are spent between modes $\textsc{up}(k)$ and $\textsc{down}(k)$.

3. Finally, the transition between $\textsc{up}(k)$ and $\textsc{down}(k)$ happens when the value of the variable $s \leq 1$. Furthermore, the mode invariants ensure that $t \in [2k+1-\epsilon, 2k+1-\epsilon]$ when the switch takes place.

Figure 10 also shows the signal fragment during the times $[2k, 2k+2]$ in connection with the modes that will be encountered. As a result, we conclude that the transition between UP($k$) and DOWN($k$) can happen either at $t = 2k+1-\epsilon$ or at $t = 2k+1+\epsilon$ (but not both, of course). We will use the former choice to mimic the variable assignment $x_k := \mathit{true}$ and the latter to mimic $x_k := \mathit{false}$. This is achieved by setting the derivatives of the clause variables as follows:

1. For each clause $C_j$ that contains the literal $x_k$, we set $\dot{c_k} = -1$ in mode UP($k$) and $\dot{c_k} = +1$ in mode DOWN($k$).

2. For each clause $C_j$ that contains the literal $\overline{x_k}$, we set $\dot{c_k} = 1$ in mode UP($k$) and $\dot{c_k} = -1$ in mode DOWN($k$).

3. For each clause $C_j$ that does not contain either $\overline{x_k}$ or $x_k$, we set $\dot{c_j} = 0$ in both modes.

The dynamics for $t$ are always $\dot{t} = 1$ in all modes and finally the dynamics of $s$ are in the range $[\frac{-1}{\epsilon}, \frac{1}{\epsilon}]$ so that $s$ can rise of fall by at least 1 unit within any timeframe of $\epsilon$.

For $j = 1, \ldots, m$ let $c_j(t)$ represent the values of the continuous variable $c_j$ at time $t$. We are interested in the difference $c_j(2k+2) - c_j(2k)$ as a function of the choice of switching time between DOWN($k$) and UP($k$).

**Lemma B.1.** *If the switching time from* UP($k$) *to* DOWN($k$) *occurs at time* $t = 2k+1-\epsilon$ *then*

- *For every clause $C_j$ containing the literal $x_k$, we have $c_j(2k+2) - c_j(2k) = +2\epsilon$.*
- *For every clause $C_j$ containing the literal $\overline{x_k}$, we have $c_j(2k+2) - c_j(2k) = -2\epsilon$.*
- *If a clause contains neither $x_k$ or $\overline{x_k}$ its value is preserved.*

*Proof.* Since we switch at $t = 2k+1-\epsilon$, we spend $1-\epsilon$ time in UP($k$) and $1+\epsilon$ time in DOWN($k$). If $x_k$ is contained positively in clause $C_j$ then

$$\begin{aligned} c_j(2k+2) - c_j(2k) &= (1-\epsilon)\tfrac{dc_j}{dt}|_{up(k)} + (1+\epsilon)\tfrac{dc_j}{dt}|_{down(k)} \\ &= (1-\epsilon)(-1) + (1+\epsilon)1 \\ &= +2\epsilon \end{aligned}$$

Proof is pictorially depicted as part of Figure 11 (top left, top right). □

**Lemma B.2.** *If the switching time from* UP($k$) *to* DOWN($k$) *occurs at time* $t = k+1+\epsilon$ *then*

- *For every clause $C_j$ containing the literal $x_k$, we have $c_j(2k+2) - c_j(2k) = -2\epsilon$.*
- *For every clause $C_j$ containing the literal $\overline{x_k}$, we have $c_j(2k+2) - c_j(2k) = 2\epsilon$.*
- *If a clause contains neither $x_k$ or $\overline{x_k}$ its value is preserved.*

Proof is pictorially depicted as part of Figure 11 (bottom left, bottom right).

Finally, the overall automaton $\mathcal{H}$ consists of $n+1$ copies of the variable gadget corresponding to the choice of variables $x_0, \ldots, x_n$ and a final mode $m$ with the invariant

$$t \geq 2(n+1) \ \wedge\ c_1 \geq -4\epsilon \ \wedge\ c_2 \geq -4\epsilon \ \wedge\ \cdots\ c_m \geq -4\epsilon$$

The dynamics of all variables are $\dot{c_j} = 0, \dot{t} = 0, \dot{s} = 0$ at the final mode.

All variables $c_j$ are set to 0 initially. The time $t$ is set to *zero* and the variable $s$ is set to 2.

Informally, every clause has at most three distinct literals in it. As a result, the value of a clause may change in three of the gadgets that correspond to the variables in it. In each gadget, it may increase by $+2\epsilon$ if the choice of the variable reflected in the choice of switching time from $\textsc{up}(k)$ to $\textsc{down}(k)$ satisfies the clause. Otherwise, the value decreases by $-2\epsilon$. Therefore, a clause $C_j$ has at least one of its literals to be true iff the value of the corresponding variable $c_j$ is at least $-4\epsilon$ when the final mode $m$ is entered.

**Lemma B.3.** *If there is a satisfiable assignment, then the signal samples are recognized by a run along the path $\pi$.*

*Proof.* Let us assume that a satisfiable assignment $\gamma$ is given. If $\gamma(x_k)$ is true for a variable $x_k$ then the choice of switching time from $\textsc{up}(k)$ to $\textsc{down}(k)$ is set to $k+1-\epsilon$. Otherwise, the switching time is set to $k+1+\epsilon$. We see that the samples for the first $2n+2$ seconds correspond to the signal samples. We now observe that $c_j \geq -4\epsilon$ for each $j = 1, \ldots, m$ when the final mode is entered. This is because, each clause $C_j$ has at least one literal that is satisfied. Therefore, the run would have increased the value of $c_j$ variable by $2\epsilon$. The smallest possible value for $c_j$ is therefore $-4\epsilon$. ☐

**Lemma B.4.** *If there is a run along the path $\pi$ that generates the given samples then there is a satisfiable assignment.*

*Proof.* First we gather that the switching times from each mode $\textsc{up}(k)$ to $\textsc{down}(k)$ must be at $t = k+1-\epsilon$ or $t = k+1+\epsilon$. We can take the former to denote $x_k := \mathit{true}$ and the latter to denote $x_k := \mathit{false}$. Given that the final mode is reached, we conclude that $c_j(2k) - c_j(0) \geq -4\epsilon$ for all $c_j$. From this, we gather that at least one of the values of the three literals in $C_j$ must be true. ☐

This concludes the reduction from 3-SAT to the Path Signal Recognition problem and as a result, the problem is NP-hard. ☐

Combining the NP-hardness and membership in NP, we conclude the NP-completeness of the path signal recognition problem.