Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 317 (2015) 71–83

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Theorem-Proving Analysis of Digital Control Logic Interacting with Continuous Dynamics

Geoffrey C. Hulette[1], Robert C. Armstrong, Jackson R. Mayo, Joseph R. Ruthruff

Sandia National Laboratories, P.O. Box 969, Livermore, California 94551-0969, USA

Abstract

This work outlines an equation-based formulation of a digital control program and transducer interacting with a continuous physical process, and an approach using the Coq theorem prover for verifying the performance of the combined hybrid system. Considering thermal dynamics with linear dissipation for simplicity, we focus on a generalizable, physically consistent description of the interaction of the real-valued temperature and the digital program acting as a thermostat. Of interest in this work is the discovery and formal proof of bounds on the temperature, the degree of variation, and other performance characteristics. Our approach explicitly addresses the need to mathematically represent the decision problem inherent in an analog-to-digital converter, which for rare values can take an arbitrarily long time to produce a digital answer (the so-called Buridan's Principle); this constraint ineluctably manifests itself in the verification of thermostat performance. Furthermore, the temporal causality constraints in the thermal physics must be made explicit to obtain a consistent model for analysis. We discuss the significance of these findings toward the verification of digital control for more complex physical variables and fields.

Keywords: formal methods, theorem proving, hybrid systems, cyber-physical systems

1 Introduction

Formal verification of hybrid or cyber-physical systems [2] can be viewed as a broader extension of numerical software verification – one in which real-valued variables and functions are modeled not merely for purposes of understanding their representation in a digital computation, but as actual physical phenomena with which a digital computation interacts. This viewpoint indicates a need both (1) to extend formal verification techniques for reasoning about digital computation to include continuous dynamics, and (2) to ensure the consistency of such hybrid models with physics, including the physics of digital computation itself. That is, since all extant systems are believed to be ultimately physically continuous, it is important to understand

[1] Email: ghulett@sandia.gov

http://dx.doi.org/10.1016/j.entcs.2015.10.008

under what circumstances parts of a system can be modeled as digital, and how to reason formally about the entire system.

Much research and development work has targeted enabling formal verification of hybrid systems, typically in the form of model checking for so-called hybrid automata [3,9]. We argue that this existing work is in different ways too broad and too narrow: Modeling approaches that freely combine discrete and continuous dynamics can readily introduce ill-posed and unphysical behavior due to the delicate interaction between the two types of dynamics [7]. And reasoning about hybrid systems via model checking is limited to properties that can be verified conservatively by enumeration of discrete regions within the continuous state space; even approaches using theorem proving have implemented model-checking strategies [14] or have relied on restrictive logics to formally model hybrid systems [12]. Work exists on formally analyzing continuous differential equations via theorem proving, but without modeling a coupling to digital logic [13]. We propose an approach that can leverage the full power of higher-order logic in the Coq theorem prover [5] to reason about physically consistent hybrid digital-physical models. Unlike model-checking approaches, our goal is not to completely automate the verification, but rather to provide maximum power and scalability for reasoning rigorously about properties of interest, leveraging understanding of system design for both the digital and physical elements.

In the remainder of this paper we present the physical modeling considerations that motivate this work (Sec. 2); a simple hybrid thermostat model used to illustrate our approach (Sec. 3); an analysis of that model using informal mathematics to convey the key ideas (Sec. 4); a corresponding formal analysis in the Coq theorem prover (Sec. 5); and a conclusion (Sec. 6).

Excerpts of the formal analysis are shown in this paper, and the full Coq implementation is available online [1].

2 Physics of Hybrid Modeling

The novelty of this work lies in a formal proof for an almost trivial cyber-physical system but with faithful modeling of continuous physical variables as real numbers coupled consistently to the digital control program. A noted limitation [13] of typical approaches to cyber-physical problems is that continuous physics is first "digitized" and the resulting, completely digital model is then analyzed [3]. We observe that this common strategy can obscure important physical constraints. One such constraint is causality, the requirement that a physical effect cannot precede its cause in time. Another is the Arbiter's Problem [4], also known as Buridan's Principle [10], a fundamental property of physics stating that a discrete decision based on a continuous variable (i.e., an analog-to-digital conversion) cannot be guaranteed to complete in bounded time; this property must be accounted for in any analysis seeking formal guarantees about discrete decisions on real numbers. Interestingly, both of these physical constraints are also closely related to considerations of computability, as is natural if the viewpoint is taken that the physical universe itself may arise from an

underlying computational process [15].

2.1 Causality

Causality means that the value of a physical variable depends only on its beginning state and what happens to it subsequently, i.e., an event in the future cannot affect any variable in the present or past, and any system failing to satisfy this constraint is considered unphysical. Computationally, causality manifests as a requirement that for recursively defined programs describing causal systems, the recursion must always be *well-founded*, i.e., must eventually terminate for any input.

For our application (described in detail in Sec. 3), we wish to evaluate a real-valued physical variable T at discrete time instants that form a potentially unbounded, ordered set $\{\ldots, t_n, t_{n-1}, \ldots, t_0\}$ (listed from future to past). We will rely on the physics concept of a "propagator" that directly composes the solution, rather than the usual differential-equation representation of the continuous physics. Of the several kinds of propagator that can apply to such physics, we start with the "macro" propagator, the *time evolution operator* $\mathcal{U}$:

$$T_i \equiv T(t_i) = \mathcal{U}(t_i, t_j, T_j, \mathcal{F}) \qquad \text{with } t_i > t_j \tag{1}$$

$$= \mathcal{U}(t_i, t_m, \mathcal{U}(t_m, t_j, T_j, \mathcal{F}), \mathcal{F}) \qquad \text{with } t_i > t_m > t_j, \tag{2}$$

where $\mathcal{U}$ advances to a final time from an initial time for an initial value of T, under the influence of a corresponding temporal sequence of events (external forcings) $\mathcal{F} = \{\ldots, f_n, f_{n-1}, \ldots, f_0\}$. The equivalence of the time evolution done all at once in Eq. (1) or in stages in Eq. (2) is a necessary consistency property that holds for allowed physical dynamics. $\mathcal{F}$ is chosen as a discrete set but could easily be extended to continuous time. Given that the transducer we will consider interacts with T only at discrete times (Sec. 3), discrete events are most relevant to this work.

In a formal analysis of Eq. (2), it is in effect necessary to show that the time evolution computation terminates or has a solution. This is physically ensured by the causality property:

$$\mathcal{U}(t_i, t_j, T_j, \mathcal{F}) = \mathcal{U}(t_i, t_j, T_j, \mathcal{F}_{ij}), \tag{3}$$

where $\mathcal{F}_{ij}$ is the subset $\{f_{i-1}, \ldots, f_{j+1}, f_j\}$ in Eq. (1). That is, the time evolution operator actually depends only on events that occur between the initial time and the final time. (As will be made clear subsequently, our convention is that T_i is the state existing just *before* the event f_i.) In Sec. 5, we will show how this is reflected in a proof of termination within the Coq theorem prover – as opposed to the self-referential inconsistency of a non-causal time evolution operator that depends on events occurring in the future.

The specific form of causality invoked here is based on the Markov property of physical dynamics, which asserts that for a suitable variable such as T (satisfying a well-posed differential equation), the evolution also does not depend on events *prior* to f_j once T_j is given. That is, the effects of prior events are captured in the initial

condition T_j and can subsequently be "forgotten". This property helpfully bounds the information needed to causally propagate the physical state.

2.2 Buridan's Principle

Formal verification is often undertaken in order to identify rare but potentially catastrophic corner-case behaviors. Buridan's Principle describes an often-overlooked issue in cyber-physical modeling, where discrete decisions about continuous variables are often required, in spite of the fact that such decisions cannot be guaranteed to complete in bounded time. Buridan's Principle manifests in the system as the decision potentially taking an arbitrarily long time to complete, or equivalently, remaining incomplete (with an intermediate, non-digital result) if it is examined after a fixed time. Many cyber-physical analyses digitize the physics prior to analysis, an approach that is convenient but fails to preserve the fundamental continuity properties that can lead to unexpected indecision in the real system – exactly the sort of corner-case behavior that formal verification seeks to uncover.

In our approach, we account for Buridan's Principle by representing a decision on a continuous variable (temperature) with a continuous function that returns a 0 or 1 decision outside of an arbitrarily small interval but a value between 0 and 1 inside of it. As with causality, we also observe a close connection of Buridan's Principle to terminating computations or decidability. When continuous real values are represented computationally (e.g., using arbitrary-precision arithmetic), Boolean comparison of a real value to a threshold (or more generally, evaluation of any discontinuous function) is a computationally undecidable problem. This problem is often referred to as the Table Maker's Dilemma [8].

It is important to note that Buridan's Principle does not conflict with the physical propagation of discrete states by actual computers. *Given* a discontinuous set of initial states (e.g., voltages representing 0 or 1), an appropriate continuous nonlinear electrical circuit can implement logic perfectly by computing resulting discrete states at subsequent clock cycles [10]. Thus purely digital models, and traditional formal analyses thereof, are valid and valuable for a computational component that is set up in this way, but are not sufficient for understanding cyber-physical system behavior comprehensively including continuous inputs and outputs. The latter consideration calls for understanding, e.g., potential non-digital behaviors from indecision in a nominally digital device – either pragmatically bounding them in probability or, as in this paper, incorporating them as far as possible in a consistent model for exhaustive formal analysis.

3 Definition of the Thermostat Model

In physical terms, this model describes an idealized, thermally homogeneous object that gains heat from time to time via a rapid heat pulse (idealized as instantaneous) from a transducer, and loses heat to the environment via a linear cooling law. The transducer is designed to maintain the object's temperature T in a desired range above the ambient temperature by, at uniform time intervals, measuring T

and applying a heat pulse if T is below a threshold. While this is a relatively simple thermostat behavior, we view it as an instance of digital control logic coupled to physical dynamics, which could be generalized to more complex scenarios. In particular, even this simple thermostat embodies the challenges noted previously: modeling the cyber-physical system in a well-posed causal form, and accounting for indecision in analog-to-digital conversion.

For convenience, we take the unit of time to be the interval between sensor measurements, and take the zero point of the temperature scale to be the ambient temperature. Our model has four *positive* real parameters: the cooling coefficient α, the temperature rise H due to a heat pulse, the nominal threshold temperature T_*, and the temperature margin ϵ for indecision. The constraint $T_* > \epsilon$ is imposed (see below). An additional parameter is an "arbiter" function $\tilde{\theta}\colon \mathbb{R} \to \mathbb{R}$ that approximates the unit step function but allows for indecision rather than requiring an unrealistic discontinuity. For all $\Delta \in \mathbb{R}$, the arbiter must satisfy

$$\tilde{\theta}(\Delta) \in [0, 1], \tag{4}$$

$$\Delta > \epsilon \implies \tilde{\theta}(\Delta) = 1, \tag{5}$$

$$\Delta < -\epsilon \implies \tilde{\theta}(\Delta) = 0. \tag{6}$$

The behavior of the physical system is described by the temperature as a function of time, $T\colon \mathbb{R}_{\geq 0} \to \mathbb{R}$, assuming that the system starts running at time $t = 0$. Instead of the traditional differential-equation formulation, we use an integral equation that corresponds to the time evolution operator $\mathcal{U}$ discussed in Sec. 2. Moreover, following standard techniques in physics, we exploit the *linearity* of the thermal dynamics to express $\mathcal{U}$ via *superposition* in terms of "micro" propagators. The latter propagators represent individual, linearly combining contributions to the solution, and in general include a *kernel* that propagates initial or boundary conditions and a *Green's function* that propagates external forcing events.

In our thermal case, the kernel and the Green's function reduce to the same propagator

$$G(t, t') = e^{-\alpha(t-t')}\, \theta(t - t'). \tag{7}$$

The (exact) unit step function θ here is not an arbiter but a means for continuous-time dynamics to enforce causality – that an effect at time t cannot precede its cause at time t'. We take $\theta(0) = 0$, which amounts to the convention that $T(t')$ itself is *not* affected by a possible discrete event occurring at time t'; this permits us to more conveniently model "instantaneous" interaction with a transducer, because then defining a discrete event at time t' as a control response in terms of $T(t')$ does not lead to causality-violating circular dependence.

The time evolution operator (1) for our thermal system, then, advances the state from t_j to t_i by linearly superposing the effects of the initial condition $T(t_j)$ and the

subsequent heating events:

$$\begin{aligned} T(t_i) &= \mathcal{U}(t_i, t_j, T(t_j), \mathcal{F}) \\ &= T(t_j)\, G(t_i, t_j) + \int_{t_j}^{\infty} dt'\, q(t')\, G(t_i, t') \\ &= T(t_j)\, e^{-\alpha(t_i - t_j)} + \int_{t_j}^{t_i} dt'\, q(t')\, e^{-\alpha(t_i - t')} \quad \text{for } t_i > t_j \geq 0. \end{aligned} \tag{8}$$

Here, the part of the integral for $t' > t_i$ vanishes due to the causality-enforcing step function in G; and the function q is the thermal forcing term, which in our case is a sum over heat pulses. In accordance with our convention, any discrete forcing event at exactly time t_j *is* to be included in the integral because it is not already reflected in $T(t_j)$. Thus far we have exploited the linearity of the thermal dynamics with respect to q considered as an *external* forcing.

We now introduce the feedback from the transducer, which makes the fully coupled cyber-physical system *nonlinear*. Namely,

$$q(t) = \sum_{l=0}^{\infty} H\, \tilde{\theta}\big(T_* - T(l)\big)\, \delta(t - l). \tag{9}$$

The discrete sum over l reflects the fact that the digital control logic only interacts with the physics periodically at integral times. The control logic design seeks to provide a heat pulse of magnitude H if and only if the current temperature is below T_*. The constraint $T_* > \epsilon$ is imposed to ensure that $\tilde{\theta}(T_* - 0) = 1$, i.e., the transducer operates non-trivially and does not allow the object to simply remain at the (zero) ambient temperature. Buridan's Principle requires the use here of a continuous arbiter function $\tilde{\theta}$ rather than the exact step function θ, since the transducer cannot be guaranteed to provide a discrete response in bounded time. The fact that $\tilde{\theta}$ is a nonlinear function makes the time evolution operator (8) likewise nonlinear in T once the dependence of the forcing on T is included.

4 Informal Analysis of the Thermostat Model

A key characterization of the performance of this cyber-physical system is its ability to maintain the temperature in a desired range above the (zero) ambient temperature. Thus, we wish to prove the following as a theorem for some particular bounds $0 < A < B < \infty$:

$$\text{If} \quad T(0) \in [A, B], \quad \text{then} \quad T(t) \in [A, B] \quad \text{for all } t \in \mathbb{R}_{\geq 0}. \tag{10}$$

The utility of a specific thermostat would be judged by whether this result holds for values of A and B that reflect the system requirements.

The theorem (10) can be derived straightforwardly from the following lemma:

$$\text{For all } n \in \mathbb{N}, \ \text{ if } \ T(n) \in [A, B], \ \text{ then } \ T(t) \in [A, B] \ \text{ for all } \ t \in (n, n+1]. \tag{11}$$

The derivation is as follows: By first specializing $t = n+1$ in the lemma, and recalling the hypothesis that $T(0) \in [A, B]$, we obtain by induction that $T(n) \in [A, B]$ for all $n \in \mathbb{N}$. Now, whereas for $t = 0$ the theorem (10) is trivially valid, for $t > 0$ we specialize $n = \lceil t \rceil - 1$ in the lemma and obtain the required result. This establishes the temperature bounds for all $t \in \mathbb{R}_{\geq 0}$.

We now argue that the lemma (11) holds for any A and B that satisfy

$$0 < A \leq \min\left(\frac{H}{e^{\alpha}-1}, (T_* - \epsilon)e^{-\alpha}\right) \quad \text{and} \quad B \geq T_* + \epsilon + H. \tag{12}$$

Under the constraints of our model, this means that suitable bounds can be found with $0 < A < B < \infty$.

As a starting point, from the governing equations (8) and (9), if we assume $t \in (n, n+1]$ and substitute $\{t_i, t_j\} = \{t, n\}$, we compute

$$T(t) = \Big(T(n) + H\,\tilde{\theta}\big(T_* - T(n)\big)\Big)\, e^{-\alpha(t-n)}. \tag{13}$$

We are given that $T(n) \in [A, B]$. We note that $t - n \leq 1$ and thus $e^{-\alpha(t-n)} \geq e^{-\alpha}$.

The proof of the lemma (11) is now by case analysis.

4.1 Proof of $T(t) \geq A$

4.1.1 Low $T(n)$ Range

If $T(n) \in [A, T_* - \epsilon)$, then $T_* - T(n) > \epsilon$ and so $\tilde{\theta}\big(T_* - T(n)\big) = 1$. Thus,

$$T(t) = \big(T(n) + H\big)e^{-\alpha(t-n)} \geq (A + H)e^{-\alpha}. \tag{14}$$

The imposed condition (12),

$$A \leq \frac{H}{e^{\alpha}-1}, \tag{15}$$

upon multiplying both sides by $1 - e^{-\alpha}$ and then adding $Ae^{-\alpha}$, gives

$$A \leq (A + H)e^{-\alpha}. \tag{16}$$

Accordingly, $T(t) \geq A$. Interpretation: If the temperature $T(n)$ is low enough, then a heat pulse occurs at time n and is sufficient, even with subsequent cooling, to keep the temperature at or above A.

4.1.2 High $T(n)$ Range

If $T(n) \in [T_* - \epsilon, B]$, then because $\tilde{\theta}$ is non-negative throughout its domain,

$$T(t) \geq T(n)\, e^{-\alpha(t-n)} \geq (T_* - \epsilon)e^{-\alpha} \geq A, \tag{17}$$

where the final inequality is from the condition (12) on A. Interpretation: If the temperature $T(n)$ is high enough, then even *without* a heat pulse, subsequent cooling does not take the temperature below A.

4.2 Proof of $T(t) \leq B$

4.2.1 Low $T(n)$ Range

If $T(n) \in [A, T_* + \epsilon]$, then because $\tilde{\theta}$ always returns a value ≤ 1, we have

$$T(t) \leq \big(T(n) + H\big)e^{-\alpha(t-n)} \leq T_* + \epsilon + H \leq B, \tag{18}$$

where the final inequality is the condition (12) imposed on B. Interpretation: If the temperature $T(n)$ is low enough, then even *with* a heat pulse, the temperature remains $\leq B$.

4.2.2 High $T(n)$ Range

If $T(n) \in (T_* + \epsilon, B]$, then $T_* - T(n) < -\epsilon$ and so $\tilde{\theta}\big(T_* - T(n)\big) = 0$. Thus,

$$T(t) = T(n)\, e^{-\alpha(t-n)} \leq T(n) \leq B. \tag{19}$$

Interpretation: If the temperature $T(n)$ is high enough, then no heat pulse occurs at time n and subsequently the temperature merely cools further below B.

Having covered all cases, we conclude that $T(t) \in [A, B]$, Q.E.D.

5 Formal Implementation

We have formalized our analysis within the Coq interactive theorem prover [5], which allows us to precisely define the various terms of our model and then state and prove theorems about those terms. To model continuous variables, we use the `Reals` module provided as part of Coq's standard library [11].

5.1 Accounting for Buridan's Principle

For our analysis, the arbiter function $\tilde{\theta}$, as discussed in Sec. 2, need not be defined explicitly but must have essential properties asserted corresponding to Eqs. (4)–(6):

```
Parameter eps : R.
Hypothesis eps_pos : 0 < eps.

Parameter theta_tilde : R → R.
Hypothesis theta_tilde_bound : ∀ d, 0 ≤ theta_tilde d ≤ 1.
Hypothesis theta_tilde_1 : ∀ d, d > eps → theta_tilde d = 1.
Hypothesis theta_tilde_0 : ∀ d, d < -eps → theta_tilde d = 0.
```

In accordance with Buridan's Principle, this formulation avoids the need to compare exact real numbers, side-stepping the associated undecidability problem while retaining enough structure to support our analysis. Verification of a discrete implementation (e.g., using floating-point comparison) would require proof that the implementation approximates the abstract definition.

5.2 Causal Definition of Temperature

To express the temperature as a computation (i.e., a simulation of the cyber-physical system), it is most natural to define the temperature function recursively. As per Sec. 2, a principle of causality must be available to the theorem prover to demonstrate that the function terminates; otherwise it is both unphysical and unsound in Coq's logic.

First, we provide a function G (essentially the propagator introduced in Sec. 4, with some multipliers included) to factor out the non-recursive part of the calculation:

```
Definition G (t n : nat) (Tn : R) : R :=
  H × theta_tilde (Tstar - Tn) × exp (-a × INR (t - n)).
```

We then define a function *sum_0_to* taking two arguments: m, the upper bound of summation, and f, the function to be summed over. The lower bound of summation is implicitly zero. The definition is as follows:

```
Definition sum_0_to (m : nat) (f : ∀ x, x ≤ m → R) : R :=
  sum_inner m f m (le_n m).
```

This definition relies on a "helper" function, called *sum_inner*, which is defined as:

```
Fixpoint sum_inner (m : nat) (f : ∀ x, x ≤ m → R) (n : nat) : n ≤ m → R :=
  match n with
  | O ⇒ fun (pf : O ≤ m) ⇒ f O pf
  | S n' ⇒ fun (pf : S n' ≤ m) ⇒
    ((f (S n') pf) + sum_inner m f n' (le_Sn_le n' m pf))%R
  end.
```

The implementation of *sum_0_to* is unremarkable except in one respect: The function parameter f requires, in addition to the usual value parameter x, an extra parameter giving *evidence* (i.e., a proof) that x is less than or equal to the upper summation bound m. The *sum_inner* helper function can always construct this evidence (for any value of m), because it only invokes f with values ranging from 0 to m.

Now we define the temperature function, *temp_f*. This function computes the propagation of earlier events to the present through the function G defined previously:

```
Definition temp_f (t : nat) : R :=
  temp_f_inner t t (le_n t).
```

Once again, the hard work of *temp_f* is performed mostly within a helper function:

```
Definition temp_f_inner : ∀ (m t : nat), (t ≤ m) → R.
  refine (fix temp_f_inner m t :=
    match m,t with
    | _,O ⇒ fun _ ⇒ T0
    | O,S t' ⇒ fun (pf : S t' ≤ O) ⇒ except _
```

```
    | S m',S t' ⇒ fun (pf : S t' ≤ S m') ⇒
      let f := fun n pf' ⇒ G (INR t) n (temp_f_inner m' n _) in
      sum_0_to t' f
    end).
  try (match goal with H : S _ ≤ O ⊢ _ ⇒ inversion H end).
  apply le_trans with t'; auto using le_S_n.
Defined.
```

As part of the construction, an assertion that an event at a future time cannot contribute to the computation of the temperature at the current time must be made. To the theorem prover this requirement manifests itself as a termination condition for *temp_f_inner* and reflects the fact that a non-causal function that depends on both the future and the past is self-referential and inconsistent in the general case. Computationally, this is understood as an obligation to demonstrate that the recursively defined function is well-founded. Here, *temp_f_inner* and subsidiary functions are constructed such that they expect a parameter establishing proof that only times less than the current time will be evaluated and contribute to the result of the computation.

Coq is able to realize that *temp_f_inner* terminates (and equivalently, is causal) because:

(i) At $t = 0$, the result is a constant;

(ii) We explicitly prove that invoking the function with a time (parameter t) greater than the "present" (parameter m) is impossible – the proof parameter $(t \leq m)$ provides the evidence we need to rule this case out;

(iii) The recursive call to *temp_f_inner* is parametrized by m', which is smaller than m; thus the function is strictly decreasing in its parameter m;

(iv) *sum_0_to* only invokes *temp_f_inner* for times up to t', which is proven by transitivity to be less than or equal to m', and thus fulfills requirement (ii) that $(t \leq m)$ in the recursive call.

Implementing *temp_f_inner*, with its relatively complicated propagation of proof terms through recursive calls, was greatly eased by our use of the `refine` tactic, which allows "proof holes" in definitions. The holes then appear as obligations to be proven after the definition is complete. Crucially, the proofs can be discharged in Coq's interactive proof mode using tactics, which is much easier than providing the proof terms directly in the definition. This facility allowed us to combine programming and proving in a way that otherwise would have been quite challenging.

We have not yet completed the proof that this computational definition corresponds to the original integrated temperature equation (8). The proof is straightforward in principle but depends upon an extension to Coq's `Reals` standard library of theorems, which is the subject of ongoing work [6].

5.3 Proof of Temperature Bounds

Formalizing the proof of lemma (11) in Sec. 4 is straightforward. Here we present the interesting parts of the development, eliding the more tedious details. First, we define Eq. (13) as T, using *theta_tilde* from above, along with the other relevant parameters:

```
Definition T Tn tau (tau_bnd : 0 < tau ≤ 1) :=
  (Tn + H × theta_tilde (Tstar - Tn)) × exp (-a×tau).
```

The definition takes three parameters. The first, Tn, is the temperature at time n where $n \in \mathbb{N}$. The parameter tau represents the time increment relative to n at which we want to evaluate the temperature. The final parameter, tau_bnd, is a proof that tau lies in the interval $(0, 1]$. The definition above corresponds to Eq. (13), with $t = n + tau$. Using this definition, the statement and proof of lemma (11) are expressed as follows:

```
Theorem T_in_interval (Tn tau : R) (tau_bnd : 0 < tau ≤ 1) :
  A ≤ Tn ≤ B → A ≤ T Tn tau tau_bnd ≤ B.
Proof.
  intros HAB. decompose record HAB. split.

  destruct (Rlt_le_dec Tn (Tstar - eps)).
    apply Tn_heat_keeps_above; auto.
    apply Tn_no_heat_keeps_above; auto.

  destruct (Rle_lt_dec Tn (Tstar + eps)).
    apply Tn_heat_keeps_below; auto.
    apply Tn_no_heat_keeps_below; auto.
Qed.
```

The proof is structured as a four-way case analysis as described in Sec. 4, with each case discharged by applying a subsidiary lemma (e.g., $Tn\_heat\_keeps\_above$). The statements of the four lemmas (shown below, with proof bodies elided) match the conditions described in the informal proof.

```
Lemma Tn_heat_keeps_above (Tn tau : R) (tau_bnd : 0 < tau ≤ 1) :
  A ≤ Tn < Tstar - eps → A ≤ T Tn tau tau_bnd.
Proof. ... Qed.

Lemma Tn_no_heat_keeps_above (Tn tau : R) (tau_bnd : 0 < tau ≤ 1) :
  Tstar - eps ≤ Tn ≤ B → A ≤ T Tn tau tau_bnd.
Proof. ... Qed.

Lemma Tn_heat_keeps_below (Tn tau : R) (tau_bnd : 0 < tau ≤ 1) :
  A ≤ Tn ≤ Tstar + eps → T Tn tau tau_bnd ≤ B.
Proof. ... Qed.

Lemma Tn_no_heat_keeps_below (Tn tau : R) (tau_bnd : 0 < tau ≤ 1) :
  Tstar + eps < Tn ≤ B → T Tn tau tau_bnd ≤ B.
Proof. ... Qed.
```

The rather lengthy proofs for these lemmas, along with the detailed analysis of the entire thermostat system, are provided online [1].

6 Conclusion

We have presented an analysis for a simple but representative cyber-physical system, formalized within the Coq interactive theorem prover. Our method mixes discrete and continuous logic and shows how their interaction requires accounting for causality and Buridan's Principle. We make a case that these two constraints are universal to all cyber-physical systems combining discrete and continuous elements and that both constraints are essential to establishing confidence in the formal analysis.

It is likely that any analysis that involves discrete decisions on continuous variables, endemic to cyber-physical systems, needs to consider Buridan's Principle. Real-world digital systems are constructed from continuous physical processes, but exploit those processes' intrinsic timescales to synchronize periodically with a clock, making their *purely* digital function immune from this concern: for example a Babbage engine or a solid-state electronic processor [10]. However, external physical processes sampled by a digital system will lack this synchronization and thus a decision problem subject to Buridan's Principle will be necessary: no matter how much digital processing is performed, the possibility of indecision will remain. The simple example given here shows how the principle arises in a formal proof of thermostat performance (see Sec. 5.1).

Although Buridan's Principle is rather subtle and may be a hazard for cyber-physical models where the representation of physics is pre-digitized [13], the causal property of physical systems is less subtle, arising mostly as a consistency condition in the formal analysis. Human analysts generally are aware that future events cannot affect the past. Theorem provers do not have this knowledge built-in and need this constraint manifested concretely to complete the analysis (see Sec. 5.2).

Future work will focus on more complex and realistic digital control logic and multi-dimensional physical systems that have many transducers interacting through space and time. Interactions between transducers may be mediated by digital communication only or physical processes only, or a combination of both. Besides being more realistic, multiple interacting transducers will introduce richer manifestations of the causality and Buridan's Principle constraints discussed here.

Acknowledgement

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration (NNSA) under contract DE-AC04-94AL85000. This document is SAND2014-2587C.

References

[1] http://dancer.ca.sandia.gov/NSV/.

[2] Alur, R., *Formal verification of hybrid systems*, in: *Proceedings of the Ninth ACM International Conference on Embedded Software*, EMSOFT '11 (2011), pp. 273–278.

[3] Alur, R., C. Courcoubetis, T. A. Henzinger and P.-H. Ho, *Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems*, in: *Hybrid Systems* (1993), pp. 209–229.

[4] Barros, J. C. and B. W. Johnson, *Equivalence of the arbiter, the synchronizer, the latch, and the inertial delay*, IEEE Trans. Comput. **32** (1983), pp. 603–614.

[5] Bertot, Y. and P. Castéran, "Interactive Theorem Proving and Program Development," Springer, 2004.

[6] Boldo, S., C. Lelay and G. Melquiond, *Improving Real Analysis in Coq: a User-Friendly Approach to Integrals and Derivatives*, in: *CPP'12*, Lecture Notes in Computer Science **7679** (2012), pp. 289–304.

[7] Derler, P., E. A. Lee and A. Sangiovanni-Vincentelli, *Modeling cyber-physical systems*, Proceedings of the IEEE (special issue on CPS) **100** (2012), pp. 13–28.

[8] Érik Martin-Dorel, "Contributions to the Formal Verification of Arithmetic Algorithms," Ph.D. thesis, École Normale Normale Supérieure de Lyon (2012).

[9] Henzinger, T. A., *The theory of hybrid automata*, in: *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, LICS '96 (1996), pp. 278–292.

[10] Lamport, L., *Buridan's Principle*, Found. Phys. **42** (2012), pp. 1056–1066.

[11] The Coq development team, "The Coq proof assistant reference manual," (2004).
URL http://coq.inria.fr

[12] Platzer, A. and J.-D. Quesel, *KeYmaera: A hybrid theorem prover for hybrid systems*, in: *Automated Reasoning*, Lecture Notes in Computer Science **5195**, Springer, 2008 pp. 171–178.

[13] Sanwal, M. U. and O. Hasan, *Formal verification of cyber-physical systems: Coping with continuous elements*, ICCSA'13 (2013), pp. 358–371.

[14] Tveretina, O., *Towards the safety verification of real-time systems with the Coq proof assistant*, CSIT '07 **2**, Wisla, Poland, 2007.

[15] Zenil, H., editor, "A Computable Universe: Understanding and Exploring Nature as Computation," World Scientific Publishing Company, 2012.