Synthesizing Controllers for Hybrid Systems*

Deepak Kapur[1] and R.K. Shyamasundar[2]

[1] Department of Computer Science, State University of New York, Albany, NY, USA, kapur@cs.albany.edu
[2] Computer Science Group, Tata Institute of Fundamental Research, Bombay 400 005, India, shyam@tcs.tifr.res.in

Abstract. A methodology for synthesizing control laws of hybrid systems is proposed using the hybrid automaton framework. The objective is to synthesize guards for making phase transitions to ensure that the system satisfies global invariance over the whole state space. Classical analysis is used to derive a controller given the phases, transitions and global invariance of a hybrid system. The methodology seems to be general in the sense that control laws can be synthesized even if the hybrid system is modeled by an automaton more general than bounded-drift linear hybrid automaton model frequently discussed in the literature. The main requirement is that it should be possible to generate a closed form expression for state variables in every phase, as a function of time elapsed in that phase. Conditions on system parameters can be identified for which the control laws can be synthesized. Optimality criteria can be incorporated for selecting among different control strategies. The methodology is illustrated using three examples. This work is in contrast to most of the work on hybrid systems in which the focus has been on the *analysis* problem.

1 Introduction

Hybrid systems combine discrete and continuous computations. A hybrid system model descibes activities that modify their variables continuously over intervals of positive duration in addition to the familiar transitions that change the values of variables instantaneously, representing the discrete components. Many systems that interact with a physical environment such as a digital module controlling a process or a manufacturing plant, a digital-analog guidance of transport systems, control of a robot, flexible manufacturing systems, etc., can benefit from the study of hybrid models.

Design, analysis (verification) and synthesis (control) of hybrid systems have been an important area of active research [GNRR93, AKNS95, AHS96]. In verification, an objective is to show that a given hybrid system will satisfy a certain (temporal) property and will never enter any *unsafe* state. In contrast, in synthesis, the focus is on designing a *controller* that ensures the given system will never enter in any unsafe state.

* The work was partially supported by an NSF Indo-US grant INT-9416687. The first author was partially supported by an NSF grant nos. CCR-9303394, CCR-9308016, and CCR-9404930.

The study of hybrid systems is interdisciplinary, lying at the junction of computer science and control. Automata-theoretic and logical approaches have been primarily used for verification and synthesis. Hybrid automata [ACHHH95, De94] are used for modeling hybrid systems. Following [De94], informally, a hybrid transition system has finitely many discrete states, called *phases* (or *locations*), and infinitely many continuous states (henceforth just called states). A phase is used for modeling a dynamic subsystem specified by a set of differential equations. Transitions among phases are made based on conditions on continuous states. Since the behavior of a hybrid automaton can be complex, much research has been devoted to identifying subclasses of hybrid automata for which verification problems are decidable. Model checking techniques have been studied for verification [Ho95].

The controller synthesis problem discussed in this paper is that of identifying conditions on transitions among phases such that the continuous state of a given hybrid automaton always satisfies a given global invariant. We propose an approach using the classical techniques. The crux lies in deriving the *switching* points from one phase to another using some policy/optimality criterion. The method involves the following steps for each phase:

1. Solve the associated differential equations to obtain expressions for state variable in terms of the time elapsed in that phase and system parameters.
2. Determine how these values are evolving (increasing, decreasing, or both increasing and decreasing). And,
3. Identify whether the phase is *control-critical*, i.e., the value of a continuous variable constrained by the global invariant evolves in the direction opposite to the previous phase. If so, deduce possible constraints on system parameters so that the global invariant remains satisfied.

The methodology is illustrated using three examples. A *linear* water-level controller (in which the rates of change of water level are constants in different phases) [Ho95] is used as a running example as the methodology is presented. A slightly nontrivial version of a *nonlinear* water-level controller (in which the rates of change of water level are nonlinear functions as well as depend upon a system parameter) [SM94] is discussed later. This is followed by a discussion of a more general version of nuclear reactor temperature control [ACHHH95, Ho95]. It is illustrated that adopting different policies can lead to different controllers.

The work on the synthesis of controllers for discrete and timed systems has been elucidated in [AMP95]. The core idea lies in finding a winning strategy for certain two-player games defined by an automaton or by a timed automaton (both of which have the set of designated final states). Using the notion of safe games, the basic algorithm reduces to iteratively finding the subset of predecessors that would always lead to final states and thus arriving at a selection (or control) predicate at each of the transitions. In the context of timed systems, "time" becomes a third player which can interfere in favor of both the other players. Again, the controller synthesis is approached on similar lines with the care that players are not allowed to win through *Zenonism* tricks (i.e., preventing the time from progressing).

One may notice that in the synthesis of discrete and timed systems, one tries to augment the conditions for the transitions to achieve the desired property from the designated final states whereas in hybrid control, one tries to derive the control events and law from the continuous dynamics of the system and the plausible transitions among phases.

2 Hybrid Automaton Model

We adopt the framework proposed in [De94], and reproduce relevant definitions below.

Definition 1. A *hybrid transition system* H is a quintuple $(Q, R^n, \Sigma, E, \mathcal{D})$, where

1. Q is a finite set of discrete *phases*, also called *locations.*
2. R^n constitutes the set of continuous states, where R is the set of real numbers. Let V_c be a finite set of continuous variables, so the size of V_c is n.
3. Σ is a finite set of discrete events.
4. $E \subset Q \times \mathcal{P}(R^n) \times \Sigma \times \{R^n \to R^n\} \times Q$ is a finite set of edges, where $\mathcal{P}$ is the power set operator. Edges model the discrete event dynamics. An edge $e \in E$ denoted by $(q_e, X_e, V_e, r_e, {q'}_e)$ is interpreted as:
   - The transition e is enabled in phase q_e if the continuous state is in X_e. X_e characterizes the switching points from q_e to ${q'}_e$, and is specified by a formula, called a *guard.*
   - If the transition e is selected by the system (since the transition system may be nondeterministic, there may be many transitions enabled out of which one may be selected), the continuous state is reset using the function r_e before the phase ${q'}_e$ is entered.
5. $\mathcal{D}$ associates a finite set of differential equations to each phase expressing its dynamics. Let D_q be the differential equations associated with the phase $q \in Q$. Each differential equation is assumed to be of the form:

$$\dot{x} = e,$$

   where $x \in V_c$ is a continuous state variable, and e is an expression over V, where V includes V_c, other noncontinuous variables and parameters, if any.

As in [De94], a hybrid transition system executes in a sequence of phases. In each phase, it first evolves continuously by allowing time to pass, and then it evolves discretely by making an instantaneous transition. The evolution of the hybrid system can be described by the sequence of intervals over R^+. For example, $\tau = [\tau'_0, \tau_1], [\tau'_1, \tau_2], [\tau'_2, \tau_3], \cdots$, where $\tau'_0 = 0$ and $\forall i (\tau_i = \tau'_i \text{ and} \tau_{i+1} \geq \tau'_i)$ would denote an execution sequence of a hybrid system. The system evolves continuously in the interval (τ'_i, τ_{i+1}) while being in some phase q_i, and makes an instantaneous transition e_i at τ_{i+1} from q_i to q_{i+1}. If $\tau'_i = \tau_{i+1}$, then it makes a transition at τ_{i+1} without any continuous evolution in that interval. Let $q_0\ e_0\ q_1\ e_1\ \cdots q_i\ e_i\ q_{i+1}$ be a sequence of phases; if the transitions are obvious, then they can be omitted from the sequence.

The system behavior within a phase q is governed by the associated differential equations D_q. Let a *situation* be defined as a tuple $< state, time >$, where the continuous component of *state* is a function from V_c to R. Let $< s_1, t_1 >$ and $< s_2, t_2 >$ be two situations of the system with $t_1 < t_2$. An evolution from $< s_1, t_1 >$ to $< s_2, t_2 >$ is described by a set of functions F of time: $\{f_x(t) | x \in V_c\}$, differentiable in the interval $[t_1, t_2]$, such that:

a . $f_x(t_1) = s_1[x]$, the value of x in s_1, and $f_x(t_2) = s_2[x]$, the value of x in s_2.
b . The differential equation $\dot{x} = e$ in D_q that governs x, satisfies: $\dot{f}_x(t) = e(F)$ in the interval $[t_1, t_2]$, where $e(F)$ is obtained from e by replacing each occurrence of a variable $y \in V_c$ by $f_y(t)$.

An edge $e = (q_e, X_e, V_e, r_e, {q'}_e)$ in the transition system plays a crucial role. X_e specifies the set of continuous states when the system switches phases. Different X_e's correspond to different control strategies. In the sequel, for ease of reference, we use the term *phase transition system* to refer to hybrid transition systems without the switching states.

The specification of a hybrid system is usually augmented by invariants over continuous variables. The two main invariants used are:

- *System or global invariants:* These assertions must hold over all states irrespective of the phase the system is in. Examples might be physical requirements like maximum tolerable temperature, required range of water level, etc. They define the desired states a control strategy should lead the system to.
- *Phase or location invariants:* The global invariants constrain the values of continous variables in each phase based on the associated X_e's. For each location, the formula characterizing these values is called the *phase* or *location* invariant. Local invariants should be such that if a transition is made at the boundary of the local invariant of the exiting location (as soon as the local invariant is about to get violated), and after the reset map, the local invariant of the entering location must be satisfied.

The controller synthesis problem is thus synthesizing the phase invariants for each phase from global invariants and a control policy such as stay in a phase as long as possible (called *ALAP (as late as possible)* policy later). In this paper, a methodology is given for deriving X_es' (as well as phase invariants) from a partial specification of a given hybrid system. We show how X_es can be derived using exact analysis. We also discuss how one can arrive at different control strategies to achieve the global invariant using other criteria including cost.

3 A Methodology for Synthesizing Controllers

We give a methodology for synthesizing control laws for hybrid systems in which

1. the global invariance is specified as a subset of state variables to be constrained within a fixed range, i.e., $\mathbf{x_c} \{\leq, \geq\} \mathbf{z}$, where $\mathbf{z}$ is a constant (i.e., the global invariant is an n-dimensional box), and

2. the differential equations associated with each phase can be solved to generate a closed-form differentiable expression for each continuous state variable as a function of time, such that the global optima of each state variable in a given phase can be determined.

A phase transition system is given without the switching points of the continuous states to switch from one phase to another phase. These requirements are less restrictive than the requirement that the state change in a phase (or a location) has to be governed by differential equations in which variables are bounded in a fixed range (called bounded-drift linear hybrid automata in [Ho95]).

We explain and illustrate the main steps of the methodology using a water-level controller, an example discussed extensively in the literature [HRP93, ACHHH95, Ho95].

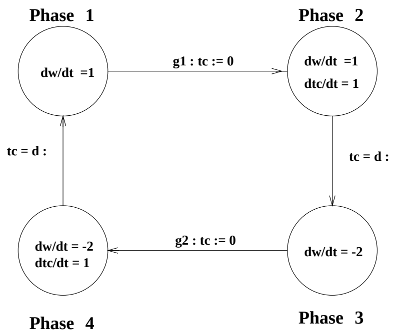

Figure 1: Water Level Controller

Water-Level Controller: The water level in a tank is controlled through a controller, which continuously senses the water level and turns a valve open and closed. The goal is to design a control strategy based on opening and closing of the valve so as to keep the water level, denoted by a state variable w, at all times between 1 ft. and 12 ft. The water-level changes as a piecewise-linear function over time. When the valve is open, the water level denoted by the variable w falls by 2 ft. per second; when the valve is closed, the water level rises by 1 ft per second. Initially, the valve is closed and the water level is 1 ft. It is assumed that there is a delay of d seconds between when the command to open/close a valve is issued, and the command becomes effective; d depends on the valve characteristics. The phase transition system is shown in Figure 1.

The dynamics of each of the four phases is given below. The global invariant is that the water level w is in between 1 and 12 ft. The controller is synthesized by deriving the local invariants for phases 1 and 3 from where the valve is signaled to open and close. The variable tc is used to model the timer to ensure that the valve indeed responds to the commands by the required time; after it is turned on, its rate of change is 1.

– *Phase 1:* **Valve closed** :$dw/dt = 1$; $tc := 0$.

- *Phase 2:* **Valve opening**: $dw/dt = 1$; $dtc/dt = 1$ and the valve completely opens in d seconds.
- *Phase 3:* **Valve open**: $dw/dt = -2$; $tc := 0$.
- *Phase 4:* **Valve closing**: $dw/dt = -2$; $dtc/dt = 1$, and valve completely closed in d seconds.[3]

3.1 Closed-form Expressions for Continuous State Component

The first step is to generate closed-form expressions for the continuous variables in each phase as a function of time elapsed in that phase. These expressions and the differential equations are used to determine whether variables constrained by the global invariants are increasing, decreasing or both.

For the water-level controller example, the differential equations for each phase are solved. Below, w_i denotes for the value of w in phase i, t_i us the time elapsed since the system last entered the i-th phase. Let $w_{i,0}$ be the initial value of w when the system last entered the i-th phase; $w_{i,f}$ is the final value of w when the system last exited phase i; $t_{i,f}$ is the time spent in the i-th phase. Let $w_{i,t}$ be the value of the variable w in phase i after time $t (\le t_{i,f})$ has elapsed since the system entered i-th phase last.

- *Phase 1*: $w_1 = t_1 + w_{1,0}$; here $w_{1,0}$ is the initial value of w as phase 1 is entered. t_1 is the time elapsed in phase 1.
- *Phase 2:* $w_2 = t_2 + w_{2,0}$, where $w_{2,0} = w_{1,f} = t_{1,f} + w_{1,0}$, $w_{1,f}$ being the value of w as phase 1 was exited, $t_{1,f}$ is the total time spent in phase 1. Also, $tc_2 = t_2$, which is bounded by d. So, $t_{2,f} = d_1$
- *Phase 3:* $w_3 = -2\ t_3 + w_{3,0}$
- *Phase 4:* $w_4 = -2\ t_4 + w_{4,0}$, $tc = t_4$, and hence $t_{4,f} = d$.

Since w is not reset on any transition: $w_{2,0} = w_{1,f}$, $w_{3,0} = w_{2,f}$, $w_{4,0} = w_{3,f}$.

3.2 Identifying Control-Critical Phases

A control strategy can be developed by identifying a maximal sequence of phases such that

1. in each phase in the sequence, the value of a constrained variable is evolving in the same direction (nonincreasing or nondecreasing) as well as the reset function, if any, associated with each transition made from one phase to another phase in the sequence resets the variable value in the same direction. Often, this can be inferred directly from the associated differential equations without needing to generate explicit closed-form expressions.
2. In the reset function associated with the next transition or in the next phase, the variable evolves in the opposite direction (if it was nonincreasing before, it increases; if it was nondecreasing before, it decreases). We will call such a phase to be *control-critical* since it is critical in determining the possible

[3] The analysis below extends even if the delays associated with valve opening and closing are different.

violation of the global invariant. The next phase can also have the variable decrease first and then increase if it was nonincreasing in earlier phases also (similarly, the variable can increase first and then decrease if it was nondecreasing earlier).

Once control-critical phases can be identified, bounds specified by a global invariant on the constrained variables can be used to determine the maximum time spent by the system in such phases as well as their phase invariants. If a constrained variable was otherwise increasing but it decreases in a control-critical phase, then the upper bound on the state variable specified in a global invariant can be used to get a constraint on that variable in that phase. Similarly, if a constrained variable was otherwise decreasing but it increases in a control-critical phase, then its lower bound as specified in a global invariant can be used. This information is backward propagated to other phases in which the constrained variable continues to evolve in the same direction.

After such analysis has been performed for each state variable, constraints are solved to identify the switching points based on a desired control policy. Guards associated with transition edges are then synthesized.

Phases may have been introduced in a phase transition model of a hybrid system to model delays involved in certain actions, e.g., opening and closing of valves in the running example, or pre-emptive signals at regular intervals using variables modeling clocks, e.g., a variation of water-tank example in chapter 5 in [Ho95]. Clock or timer is modeled using a clock variable whose derivative is 1, e. g., the variable tc in the running example. Such constraints also enforce the exact time or maximum time that could be spent in a phase.

The analysis so far has focussed on considering a single state variable constrained by the global invariant. It extends to many variables constrained by the global invariant.

In the water-level controller example, the global invariant constrains w from below as well as from above. In phases 1 and 2, w is increasing (since dw/dt is +ve in these phases), whereas w is decreasing in phases 3 and 4 (since dw/dt is −ve in these phases). Phases 2 and 4 are control-critical since w will reach a maximum value in phase 2 and a minimum value in phase 4.

Since w is increasing in phase 2, w gets to the maximum value on exit from phase 2. $w_{max} = w_{2,f} = t_{2,f} + w_{2,0}$, where $w_{2,0} = w_{1,f} = t_{1,f} + w_{1,0}$. Since $t_{2,f} = d$, $w_{max} = w_{2,f} = d + t_{1,f} + w_{1,0} \leq 12$. This gives $w_{1,f} \leq 12 - d$.

Similarly, w gets to the minimum value in phase 4 when it is to exit phase 4. $w_{min} = w_{4,f} = -2 * t_{4,f} + w_{4,0}$, where $w_{4,0} = w_{3,f} = t_{3,f} + w_{3,0}$. Since $t_{4,f} = d$, $w_{min} = w_{4,f} = -2d + t_{3,f} + w_{3,0} \geq 1$. This gives $w_{3,f} \geq 1 + 2d$.

As stated above, there can be critical-control phases in which a state variable may be increasing as well as decreasing (see example 2 in the next section). In such a phase, from its first and second derivatives, global minimum or maximum value of this variable can be determined, and the appropriate constraints from the global invariant can be deduced

3.3 Different Control Strategies

Constraints deduced from the above analysis can be used to develop different control strategies that ensure that the system never violates the given global invariant. One possible strategy is to use the constraints on variables as the guards themselves. We will call such a controller to be *generic* since it can be further refined to implement a specific policy. Another control strategy is to exit a phase *as late as possible*, *ALAP* policy, implying that if a phase is not exited and the system is allowed to evolve, the global invariant would be violated. The associated guards for this case are obtained by converting inequalities in the deduced constraints to equalities. We illustrate these strategies for the water-level controller example.

A generic controller for the example is to require that that the guard for transition from phase 1 to 2 is that $w \leq 12 - d$, and the guard for transition from phase 3 to 4 is that $w \geq 1 + 2 * d$. For $d = 2$, the respective guards are $w \leq 10$ and $w \geq 5$. Guards for other transitions are known as they are based on the timer value, namely, $tc = d$ for transition from phase 2 to 3, and $tc = d$ for transition from phase 4 to 1.

Using the ALAP controller policy. guard for transition from phase 1 to 2 is $w = 12 - d_1$, and the guard for transition from phase 3 to 4 is $w = 1 + 2 * d_2$. For $d_1 = d_2 = 2$, we get the desired values of water levels as assumed in [Ho95].

It is easy to see that the controllers can be synthesized even if lower and upper bounds on water level, say L and U, respectively, in the global invariant are given as parameters. Using the *ALAP* policy, from phase 1 to 2, the transition is made when the water level reaches $U - d$, and from phase 3 to 4, the transition is made when the water level reaches $L + 2 * d$.

The above approach is also helpful in identifying conditions when a controller can be synthesized. For the water level controller example, it must be the case that $U \geq L + 2d$ as well as $U - d \geq L$. This also makes sense as if the pump was very slow in responding to commands to open and close the valve, say $d = 6$ and $L = 1, U = 12$, then a controller cannot be synthesized to satisfy the global invariant since the first constraint above ($U \geq L + 2d$) cannot be satisfied.

3.4 Periodicity

Cycles in a phase transition system can give rise to additional constraints on continuous variables and parameters, depending upon the nature of a controller being synthesized. It is possible to generate a *viable* controller (in the sense of [De94, DV95]) which will enable the system to run forever. If guards on transitions can be synthesized that would lead to a *repeatable cycle* of phases being executed without the global invariant being violated, then it can be shown that the resulting controller is viable. A repeatable cycle of phases can be used to generate an infinite sequence of phases.

For the water-level controller example, there is clearly a cycle involving all the phases. In order to generate an infinite sequence of phases using this cycle, the system is so required that when it enters the first phase in the i-th cycle, the water level is exactly the same as it exits the fourth phase in the i-th cycle, where $i > 0$. In that case, if after the first cycle, the hybrid system enters the first phase, then every phase will be visited infinitely often.

This requirement makes $w_{1,0} = w_{4,f}$, from which the relation that $2d+2t_{3,f} = d + t_{1,f}$, implying that a viable controller can be synthesized only if there exist values of $t_{1,f}, t_{3,f}$ for given values of d. And, $t_{1,f}$ and $t_{3,f}$ can be related to other parameters. Otherwise, a controller cannot be synthesized.

For synthesizing viable controllers, repeatable cycles in a phase transition system can be analyzed. Different repeatable cycles can lead to different control policies that generate infintie sequences of phases.

4 Two Nontrivial Examples

In the above example, the system dynamics was specified by simple differential equations in which the first derivatives of variables are constants. Below, we discuss two slightly nontrivial examples in which system dynamics exhibits nonlinear behavior. The second example taken from [SM94] is a variation over the first example. The rate of net inflow into the tank is a function of the extent to which the valve is open and a system parameter. The third example is a variation on the temperature controller for a nuclear reactor discussed in [Ho95]. This example is interesting since it illustrates how the proposed methodology can be used to evaluate different control laws, and an optimal control strategy based upon some other criterion such as cost can be designed.

4.1 Nonlinear Water Controller

The phase transition system for this example is similar to that in Figure 1. Equations governing the water level are different though as well as that there is no explicit timer. The resetting function for each transition is the identity function. In every phase, the water-level w is governed by

$$dw/dt = 8\ f - 16\ v,$$

where f is a system parameter, which could take values between 0 and 1, and v is another continuous variable between 0 and 1, measuring the extent to which the valve is open. If the valve is fully closed, $v = 0$; when it is fully open, $v = 1$. The controller is required to work no matter what f is. Other differential equations for each phase are given below.

The objective is to design a controller so as to keep the water level between 60 ft and 76 ft. The global invariant is $60 \leq w \leq 76$.

- Phase 1: **Valve closed**: $dv/dt = 0, v = 0$.
- Phase 2: **Valve opening**: $dv/dt = 1,\ v \leq 1$.
- Phase 3: **Valve open**: $dv/dt = 0, v = 1$.
- Phase 4: **Valve closing**: $dv/dt = -1\ ,\ v \geq 0$.

Just as in example 1, for each phase, closed-form expressions are obtained by solving equations in terms of the time elapsed in the phase.

- Phase 1: $w_1 = 8\ f\ t_1 + w_{1,0},\ v_1 = 0$.
- Phase 2: $v_2 = t_2 + v_{2,0},\ v_{2,0} = 0,\ 0 \leq v_2 \leq 1,\ w_2 = 8\ f\ t_2 - 8\ t_2^2 + w_{2,0}$.
- Phase 3: $v_3 = 1, w_3 = 8\ f\ t_3 - 16\ t_3 + w_{3,0}$.
- Phase 4: $v_4 = -t_4 + v_{4,0},\ v_{4,0} = 1$, and $w_4 = 8\ f\ t_4 - 16\ t_4 + 8\ t_4^2 + w_{4,0}$.

Further, $w_{2,0} = w_{1,f}, w_{3,0} = w_{2,f}, w_{4,0} = w_{3,f}$.

The rate of change of water-level dw_1/dt is positive for nonzero f in phase 1. Phase 2 is interesting; dw_2/dt can be positive to begin with depending upon the value of f, and then it goes from positive to negative. Phase 2 is thus control-critical; w attains a maximum value in this phase. When $dw_2/dt = 0$, $8f = 16v_2$. Since w_2 increases with f and the controller is required to work for every f between 0 and 1, the maximum value for f is assumed. So, for $f = 1$, when v is $1/2$, the water level is maximum (this is checked from the fact that the second derivative of w_2 is negative for values of $v > 1/2$). And, this happens when $t_2 = 1/2$. Thus,

$$w_{2,max} = 4 - 2 + w_{1,f} = 76,$$

and so $w_{1,f} = 74$.

In phase 3, dw_3/dt is negative. In phase 4, much like phase 2, dw_4/dt starts being negative and it becomes nonnegative. Phase 4 ics control-critical, and w attains a minimum value. Equating $dw_4/dt = 0$, we have $v = f/2$. Since w increases with f, the minimum water level will be reached the fastest when $f = 0$, so $v = 0$, i.e., when the valve is fully open, and this happens when $t_4 = 1$.

$$w_{4,min} = -16 + 8 + w_{3,f} = 60,$$

and so $w_{3,f} = 68$.

A generic controller would have the guards, $w \leq 74$ on the transition from phase 1 to phase 2, and $w \geq 68$ on the transition from phase 3 to 4. Assuming the ALAP policy, the guard for transition from phase 1 to phase 2 is $w = 74$, and for transition from phase 3 to 4, it is $w = 68$.

The control law can be synthesized even if lower and upper bounds on water level in the global invariant are given as parameters. It is possible to determine conditions on L, U for which a controller cannot be synthesized.

4.2 Nuclear Reactor Temperature Control

In [ACHHH95, Ho95], variations of a nuclear reactor temperature controller are discussed. The system has a reactor core and two control rods that control the temperature of the reactor core. One of the rods cools slowly and the other rod cools faster. Further, once a rod is immersed in the core, it has to remain outside for a certain period of time. In the discussion in [ACHHH95, Ho95], the controller is straightforward: Put one of the rods in if the temperature reaches U, and take the rod out if temperature reaches L. If the time for which rods have to be outside is beyond a certain period, then rods 1 and 2 are alternated. We generalize the formulation so that it is possible to consider different possible control strategies based on cost criteria for cooling. The generalization illustrates the power of the methodology.

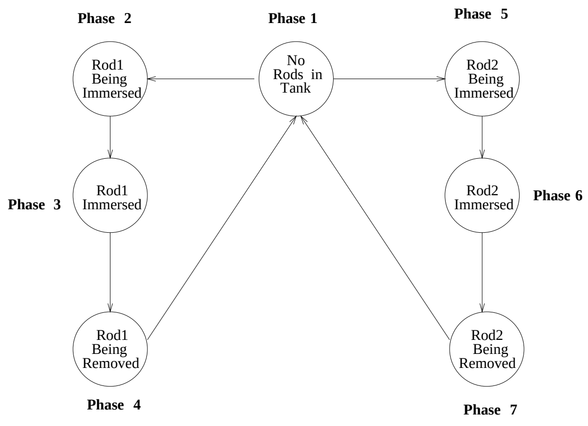

Figure 2. Temperature Controller of a Nuclear Reactor

The phase transition system is shown in Figure 2. Let T stand for the minimum time for which rods should not be immersed after they have been taken out (for simplicity, it is assumed to be the same for both rods even though the analysis extends to the case if this time parameter is different for the two rods). The temperature of the reactor is assumed to be governed by:

$$dx/dt = x/f - (t_c + k_1\ p_1 + k_2\ p_2),$$

where p_i is the fraction of i-th rod immersed in the reactor, k_i is the cooling factor of the i-th rod. It is assumed, for simplicity, that at most one rod is immersed at any time, i.e., at least one of p_1 and p_2 is 0.

Let c_i be the cost rate for keeping the i-th rod inside the reactor. For instance, if only rod 1 is used and it is inside the reactor (starting from when it is immersed inside) for t_1 time in any given cycle of time period t_m, then it costs $c_1\ t_1$ units to keep the reactor in the prescribed temperature range during that cycle, and average cost is $c_1\ t_1/t_m$ units per second.

The objective is to design a controller so as to

(i) ensure that the temperature of the reactor is between L and U,
(ii) rods are outside for at least T seconds after use, and
(iii) A cycle costs no more than K per unit on the average per second.

The global invariant is thus $L \leq x \leq U$.

To simplify the analysis, some of the above parameters – c_1, c_2, k_1, k_2, L and U are fixed. It is assumed that it takes 2 seconds to immerse a rod or to take out a rod from the reactor. And, $f = 1/10, t_c = 50, k_1 = 6$, and $k_2 = 10$, $L = 510, U = 550$, giving us a system similar to the one in [ACHHH95, Ho95].

For measuring time when a rod is outside the reactor after it was taken out last, as well as to measure time, a rod is inside the reactor, time counters are introduced. Let o_j be the time j-th rod is outside after it was taken out of the reactor last time. Let i_j be the time j-th rod is inside the reactor. The derivatives of these variables are either 1 or 0, depending upon whether the corresponding counters are in effect.

The differential equations associated with each phase are given below.

Phase 1: **No rod inside**: $do_1/dt = 1, do_2/dt = 1$. Further $di_1/dt = di_2/dt = 0$.
Phase 2: **First rod immersing**: $di_1/dt = 1, do_2/dt = 1, do_1/dt = 0, di_2/dt = 0$. The transition from phase 1 to phase 2 initializes i_1 to 0. And, $dp_1/dt = 1/2$, the rate at which rod 1 is being immersed, with $p_2 = 0$.
Phase 3: **First rod completely immersed**: $di_1/dt = 1, do_2/dt = 1, do_1/dt = 0, di_2/dt = 0$. And, $dp_1/dt = 0$ with $p_1 = 1, p_2 = 0$.
Phase 4: **First rod removing**: $di_1/dt = 1, do_2/dt = 1, do_1/dt = 0, di_2/dt = 0$, and $dp_1/dt = -1/2$, the rate at which rod 1 is being removed, with $p_2 = 0$.

The transition from phase 4 to phase 1 resets the counter for 1st rod outside, i.e., $o_1 = 0$, and records time during which it is inside. Similarly, corresponding to the immersion and removal of rod 2 in the reactor, there are phases 5, 6 and 7 with similar differential equations.

The equation governing the temperature of the nuclear reactor in each phase is:

$$dx/dt \;=\; x/10 \;-\; 50 - 6p_1 - 10p_2.$$

In phase 1, dx/dt is positive if $x > 500$, implying that the temperature is increasing. In phase 2, $dx/dt = 0$ when x attains the maximum value of 550, implying that $p_1 = 5/6$, i.e., the first rod is immersed 5/6th into the reactor. Before this point, dx/dt is positive, and it becomes negative. In phase 3, dx/dt is $-ve$ as long as $x < 560$ implying that the termperature is decreasing. In phase 4, dx/dt goes from $-ve$ to $+ve$. At $dx/dt = 0$, it attains the minimum value – 510, when $p_1 = 1/6$, meaning that the first rod is still 1/6 immersed as it is being pulled out. Similar analysis can be done for phases 5, 6 and 7 for the second rod.

There can be at least three possible control strategies:

1. Only the first rod is used for cooling.
2. Only the second rod is used for cooling.
3. Both rods are used in alternation.

Below, a detailed analysis in the case when only the first rod is used, is given. Details are omitted for the second and third control strategies.

Strategy I **Using the first rod only**: The equation for the first phase is $dx/dt \;=\; x/10 \;-\; 50$ whose solution is:

$$x_1 = 500 + (x_{1,0} - 500) \times e^{t_1/10}. \tag{1}$$

For the second phase, the equation is, $dx_2/dt = \; x_2/10 \;-\; 50 - 6t_2$, whose solution is (obtained by substituting $x_2/10 - 6t$ by z) is:

$$x_2/10 = 110 + (x_{2,0}/10 - 110) \times e^{t_2/10} + 6 \times t_2, \tag{2}$$

where t_2 is the time the system has been in the second phase. Solving for x_2 at $t_2 = 5/6$, $x_{2,5/6} = 550$, we get $x_{2,0} = 547.97 = x_{1,f}$.

Similarly for the third phase,

$$x_3 = 560 + (x_{3,0} - 560) \times e^{t_3/10}. \tag{3}$$

For the fourth phase,

$$x_4/10 = -4 + (x_{3,f}/10 + 4) \times e^{t_4/10} - 6 \times t_4. \tag{4}$$

Solving for x_4 at $t_4 = 5/6$ and $x_{4,5/6} = 510$, we get $x_{3,f} = 512.03$.

Using these values, time elapsed in each phase when rod 1 is used for cooling can be calculated. Solving equation 4 for phase 4 at $t_4 = 1$, $x_{4,f} = 510.09$ which is also $x_{1,0}$ of the next cycle in the steady-state. Substituting these in 1,

$$e^{t_{1,f}/10} = (x_{1,f} - 500)/(x_{1,0} - 500) = 4.754,$$

from which $t_{1,f} = 15.59$ seconds.

Time spent in phase 3 is obtained by substituting in equation 3:

$$x_{3,f} - 560 = (x_{3,0} - 560) \times e^{t_{3,f}/10}.$$

Since $x_{3,0} = x_{2,f}$, $x_{2,f} = 549.91$ by substituting $t_{2,f} = 1$ in 2. Using these values in above equation for x_3, $t_{3,f} = 15.59$ seconds as well.

A similar analysis can be done in case only rod 2 is being used. In the discussion below, we omit the details.

Strategy II **Using** Rod_2 **only**: The governing equation is $dx/dt = x/10 - 50 - 10p_2$. When the temperature reaches the maximum during the immersion of the second rod, it is half-immersed since when $dx_5/dt = 0$; $x_5 = 550$, $p_2 = 1/2$. Similarly, when the temperature reaches the minimum when rod 2 is being removed, it is still 1/10 immersed since $dx_7/dt = 0$ and $x_7 = 510,$, $p_2 = 1/10$. Equations can be solved for different phases, as in the case of the first rod. Time spent in phase 6 is $t_{6,f} = 14.08$ seconds, and time spent in phase 1 in the steady-state when rod 2 is only being used is 15.795 seconds.

We can associate the cost factors for the possible startegies as follows:

1. **Only rod 1 is to be used**: The time T before rod 1 can be reused ≤ 15.59 seconds when the system is in phase 1. And, the cost incurred per unit time for this cycle is $c_1(2+t_{3,f})/(t_{1,f}+2+t_{3,f}) = 0.5301c_1$, since $2+t_{3,f}$ is the time for which rod 1 is inside, and $2 + t_{1,f} + t_{3,f}$ is the total cycle time.
2. **Only rod 2 is to be used**: If only rod 2 is to be used, then the time T before rod 2 can be reused is ≤ 15.795 seconds, slightly more than in the first case as to be expected since the cooling rate of rod 2 is more. In this case, the cost incurred per unit time for this cycle is $c_2(2+t_{6,f})/(t_{1,f}+2+t_{6,f}) = 0.5045c_2$. It does not appear to be very advantageous to use rod 2 with a higher cooling factor if c_2 is significantly higher than c_1.

3. **Rods are used together:** We could alternate between the use of rod 1 and rod2, then the time before either rod can be used has to be less than or equal to 47.465 seconds (minimum of 15.795 + 2 + 14.08 + 15.59 = 47.465 and 15.795 + 2 + 15.59 + 15.59 = 48.975). These calculations are approximate because of reasons given below.
   And, the cost incurred per unit time is given by
   $$(c_1(2+t_{3,f})+c_2(2+t_{6,f}))/(t_{1,f}+2+t_{3,f}+2+t_{5,f}+t'_{1,f}) = \ .2704c_1+.2472c_2.$$

Based on T that can be tolerated, different control strategies may be possible, and the best among them may be picked based on the cost. If T is lower than 15.59 seconds, then either of the three strategies can be chosen, and the cost criterion can be brought into play to determine which one is optimal. If T is lower than 15.795 seconds but more than 15.59 seconds, then only two control strategies are feasible. If T is more than 15.795 but lower than 47.465 seconds, then a controller can be synthesized but rods 1 and 2 must alternate. If T is more than 47.465 then a controller cannot be synthesized.

It is also possible to perform a parametric analysis and relate different control strategies in a way that T depends upon U, L, k_1, k_2 etc.

5 Concluding Remarks

We have proposed a methodology for synthesizing control laws for a hybrid system whose phase transition system and a global invariance are given. As illustrated, the proposed methodology can be used to synthesiz generic controllers which could be further refined as well as controllers based on ALAP (as late as possible) policy. It is possible to identify conditions on parameters under which a controller may not be realizable as the global invariance cannot be guaranteed. The methodology is quite powerful and is applicable to the typical examples that have been used in the hybrid system literature. The interesting features of the methodology are that the analysis is exact and the only restrictions on the continuous dynamics are that the underlying equations (differential or otherwise) be solvable resulting in closed-form expressions.

To make the methodology mechanizable, the following capabilities are needed:

- A solver for differential equations.
- A solver to determine the zero set of expressions (which is needed to determine when variables get optimal values).
- Determining the sign of expressions based on the range of values variables can take.
- Determining boundary conditions under which variables could potentially violate range constraints expressed in a global invariant.

If the rate predicates are expressed to be in a fixed range (as in the case of bounded-drift linear hybrid automaton), it is clear that each of the capabilities is supported by a computer algebra system such as Mathematica or Maple insofar as it has a decent package for solving linear inequalities. Since computer algebra systems have more sophisticated algorithms, so it should be possible to exploit these algorithms for even a wider class of problems. The two examples discussed in the previous sections do not fall in the family of bounded-drift linear hybrid automata. The machinery needed to synthesize controllers for them is well-supported in computer algebra systems. Further work is in progress in formalizing and analyzing the power of the method and in building a mechanizable system supporting the methodlogy.

Acknowledgments: The authors thank Akash Deshpande, Paliath Narendran, Michael Kourjanski, and the referees for comments on earlier drafts of this paper.

References

[ACHH93] R. Alur, C. Courcoubetis, T. A. Henzinger and P.-H. Ho, *Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems*, LNCS, 736, Springer-Verlag, 1993, pp. 209-229, enhanced version is in [ACHHH95].

[ACHHH95] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger and P.-H. Ho, X. Nicollin , A. Olivero, J. Sifakis, and S. Yovine, *The algorithmic analysis of hybrid systems*, Theoretical Computer Science, 138, pp.3-34, 1995

[AHS96] R. Alur, T.A. Henzinger, and E.D. Sontag, editors, Hybrid Systems III: *Verification and Control*, LNCS, 1066, Springer-Verlag, 1996.

[AKNS95] P. Antsaklis, W. Kohn, A. Nerode and S. Shastry (Eds.), *Hybrid Systems II*, LNCS, 999, Springer Verlag, 1995.

[AMP95] E. Asarin, O. Maler, and A. Pnueli, *Symbolic Controller Synthesis for Discrete and Timed Systems*, Proc. of Hybrid Systems II, LNCS 999, 1995.

[De94] A. Deshpande, *Control of Hybrid Systems*, Ph.D. Thesis, University of California at Berkeley, 1994.

[DV95] A. Deshpande and P. Varaiya, *Viability in Hybrid Systems*, in [AKNS95], pp. 128-147.

[GNRR93] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel (Eds.), *Hybrid Systems I*, LNCS, 736, Springer-Verlag, 1993.

[HRP93] N. Halbwachs, P. Raymond, Y.-E. Proy. *Verifying linear hybrid systems by means of convex approximations*, Proc. SAS 94, LNCS 864, Sept. 1994.

[HKPV95] T. A. Henzinger, W. Kopke, A. Puri, P. Varaiya, *What's decidable about Hybrid automata*, Proc. of the 27th Annual ACM Symposium on Theory of Computing STOC 1995.

[Ho95] P.-H. Ho, *The Algorithmic Analysis of Hybrid Systems*, Ph.D Thesis, Cornell University, 1995.

[SM94] H.B. Sipma and Z. Manna, *Specification and Verification of Controlled Systems*, Proc. FTRTFT, LNCS, 863, 641-659, Sept. 1994.

[TE95] M. Tittus and B. Egardt, *Hybrid Objects*, in [AKNS95], pp. 493-508.