A Case Study on Formal Design of Hybrid Control Systems

Hong Ki Thae and Dang Van Hung
The United Nations University
International Institute for Software Technology
P. O. Box 3058, Macau
dvh@iist.unu.edu

Abstract

In this paper, we present an approach to the design of hybrid systems by combination of several comprehensive formalization techniques. We use Duration Calculus (DC) to specify the requirement and design at abstract level of system development. Then the high level designs are further refined in control theory. A formal verification may be done either in DC if it is possible, or in predicate calculus using the semantics of DC or theorems from control theory. We show our techniques through a double water tank case study which is one of the bench mark problem for modern process control engineering.

Keywords: *Embedded Systems, Discrete Design, Control Theory, Duration Calculus.*

1. Introduction

Hybrid systems are an active research area on the border between computer science and automatic control. A hybrid system is a dynamical system of discrete (Hybrid controllers) and continuous devices (physical plants). Hybrid controller is a control program implemented on digital computer for the control of physical plant. Therefore the design and implementation of such systems are studied in both of control engineering and computer science (see, e.g. [1], [6], [9]). Although many physical systems are hybrid in nature, the concept of model in control engineering is traditionally associated with differential or difference equations, typically derived by physical laws governing the dynamics of the system under consideration. On the other hand, in computer science hybrid systems are modeled with a real time reactive system which interacts with physical environment through sensors and actuators. In the last decades, however, there has been growing concerns with hybrid control systems from both control engineering and computer science. In general, the computer scientists focus on the logic and discrete aspects while the control scientists do the opposite. Perhaps a good approach should take a more balanced view. Although there exist many modeling frameworks to model the behavior of hybrid systems, there seems to be little research to bridge gap between high-level specification and practical implementation of hybrid controllers.

This paper presents an approach to the design and verification of hybrid control systems by using different formal methods. Our approach is to combine control theoretic method and formal methods whereby control theoretic methods are used for analysis of physical environments, while formal methods are used for the synthesis of the control system to substantiate the properties of whole system. The dynamic model of physical plant is used in conjunction with the control logic defined through formal model. In this approach, compositional verification is performed under different models. A required property is split into a number of conjuncts. Some of them are proved in the discrete formal verification methods using the controller properties. Others are verified in the continuous environment model by control theoretic analysis methods.

The formalism we use in this paper for the specification and verification of the hybrid control systems is the Duration Calculus (DC) [8], or more precisely, the Extended Duration Calculus (EDC) [9]. As a running example we use a hybrid controller for a double tank system. Our result in this paper has two folds: first, we show how to refine a design in EDC into a simple and well-known problem in the control theory, and how to formalize a controller in EDC; secondly, we give a general technique for the formal development of hybrid systems.

2. Interval Logic with Functions of Time

In this section, we give a summary of our specification and verification techniques for hybrid control systems. Namely, we use Extended Duration Calculus (EDC) as our specification language. Extended Duration Calculus is just an Interval Logic with chop in which temporal variables and temporal propositional letters are defined via functions of time. We refer the readers to [5, 9] for more details of interval logic and EDC.

Interval Logic with Chop (Dutertre 95) We are given the following symbol sets: the set $GVar$ of global variables $x, y, z, ...$, independent of *time*; the set $Tvar$ of temporal variables $v, v_1, v_2, ...$ interpreted as functions of time intervals, in which ℓ is an interval function denoting interval length; the set $FSymb$ of global function symbols $f, g, ...$ (such as *constants* and $+, -, *, ...$), and the set $RSymb$ of global relation symbols $G, H, ...$ (such as $true$ and $false$, and $=, \geq, ...$) which have standard meaning; and the set $PLetter$ of temporal propositional letters $X, Y, ...$ which will be interpreted as truth-valued functions of time interval.

Terms θ and formulas ϕ are built using the following grammars

$$\begin{array}{rcl} \theta & ::= & x \mid \ell \mid v \mid f(\theta, ..., \theta) \\ \phi & ::= & X \mid G(\theta, ..., \theta) \mid \neg\phi \mid \phi \vee \phi \mid \exists x.\phi \mid \phi \,;\, \phi \end{array}$$

In addition to the standard abbreviations, we denote

$$\Diamond\phi \overset{def}{=} true\,;\,(\phi\,;\,true) \qquad \Box\phi \overset{def}{=} \neg\Diamond\neg\phi$$

The semantic of terms and formulas are defined as follows. Let **Intv** denote the set of time intervals. Let a value assignment $\mathcal{V}$ ($\mathcal{V} : GVar \rightarrow$ **Reals**) for global variables, an Interpretation $\mathcal{I}$ for the given symbols be given. The semantics of a term θ is a mapping $\theta_{\mathcal{I}}$: **Intv** $\rightarrow$ **Reals**. The semantics for a term of the kinds x or v, is exactly the one given by $\mathcal{V}$ and $\mathcal{I}$. For a term $\theta = f(\theta_1, ..., \theta_n)$, $\theta_{\mathcal{I}}([a, b])$ is defined defined as $f(c_1, ..., c_n)$ where c_i is $(\theta_i)_{\mathcal{I}}([b, e])$ for $i = 1, ..., n$.

The semantics of a formula ϕ is a mapping $\mathcal{I}[\![\phi]\!]$: **Intv** $\rightarrow$ $\{$*true*, *false*$\}$ defined as follows.
$\mathcal{I}[\![X]\!]([b, e]) =$ *true* iff $X_{\mathcal{I}}([b, e]) =$ *true*;
$\mathcal{I}[\![G(\theta_1, ..., \theta_n)]\!]([b, e]) =$ *true* iff $G(c_1, ..., c_n) =$ *true*, where $c_i = (\theta_i)_{\mathcal{I}}([b, e])$ for $i = 1, ..., n$;
$\mathcal{I}[\![\phi\,;\,\psi]\!]([b, e]) =$ *true* iff $\mathcal{I}[\![\phi]\!]([b, m]) =$ *true* and $\mathcal{I}[\![\psi]\!]([m, e]) =$ *true* for some $m \in [b, e]$.
For other cases of ϕ, the definition is just the same as in predicate calculus.

Extended Duration Calculus We assume that we are given a set $\mathcal{M}$ of real functions and a set $\mathcal{B}$ of Boolean functions of time that we are interested in. Note that for an n-ary relation R over **Reals**, for $f_1, \ldots, f_n \in \mathcal{M}$, $R(f_1, \ldots, f_n)$ is a Boolean function defined by $R(f_1, \ldots, f_n)(t) =$ *true* iff $R(f_1(t), \ldots, f_n(t)) =$ *true*. We define time-function-based temporal variables and temporal propositional letters for the interval logic as follows.

- For any real function $f \in \mathcal{M}$, **b**.f and **e**.f are temporal variables interpreted as the value of f at the beginning and the ending point of an interval, respectively.
- For any Boolean function $b \in \mathcal{B}$, $\lceil b \rceil$ ($\lceil b \rceil^c$) is a propositional letter which is evaluated to *true* over an interval $[c, d]$ iff b is interpreted as *true* almost everywhere (respectively, everywhere) in $[c, d]$.
- For any Boolean function $b \in \mathcal{B}$, $\lceil b \rceil^o$ is a propositional letter which is evaluated to *true* over an interval $[c, d]$ iff $c = d$ ($[c, d]$ is a point interval) and $b(c) =$ *true*.
- For a Boolean function b which is finitely variable, $\int b$ is a temporal variable that maps an interval $[c, d]$ to $\int_c^d b(t)dt$ (b is overloaded to be its characteristic function).

A formula in Interval Logic in which only time-function-based temporal variables and temporal propositional letters are allowed, is called an EDC formula. In this paper, we will identify a Boolean function with its characteristic function without fear of confusion. We are specially interested in the following kinds of functions of time:

- State Signal: a function of time the range of which is a countable set, and which is finitely variable.
- For any state signal x, we define Boolean functions $\uparrow x, \downarrow x, \updownarrow x$ as: $\uparrow x(t) = 1$ iff $x(t+) > x(t-))$; $\downarrow x(t) = 1$ iff $x(t+) < x(t-)$; $\updownarrow x(t) \overset{def}{=} \uparrow x(t) \vee \downarrow x(t)$.
- For a state signal x, for a real constant a and for $\sim \in \{=, >, <, \leq, \geq\}$, we define Boolean functions $\uparrow_{\sim a} x, \downarrow_{\sim a} x$ and $\updownarrow_{\sim a} x$ by: $\uparrow_{\sim a} x(t) = 1$ iff $x(t+) > x(t-) \wedge (x(t+) - x(t-) \sim a$; $\downarrow_{\sim a} x(t) = 1$ iff $x(t+) < x(t-) \wedge (x(t-) - x(t+) \sim a$; $\updownarrow_{\sim a} x(t) = 1$ iff $((x(t+) \neq x(t-)) \wedge |x(t+) - x(t-)| \sim a$

In the next sections, we show how EDC formulas are used as a specification and design language for the development of hybrid control systems.

3. Specification of Double Tank System

The plant to be controlled and it's environment consist of two water tanks (plant) in series each of which has one drainage outlet, a pump (actuator) for bring water to the upper tank and to the lower tank through outlet of the former, and two level sensors which measure the levels of both tanks. We want to design a controller to control the water level in the lower tank with the pump on the upper tank. The process' output is u (input voltage to the pump) and inputs are y_1 and y_2 (voltage from level sensors).

The objective of the control is to keep the water level of the lower tank within predefined distance from dynamically changing reference value (set by operator) while to

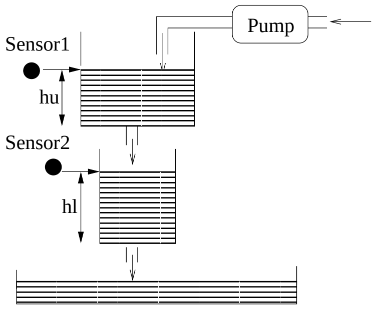

Figure 1. A double tank system

protect the system in such a way that the overflow from the tank can never occur (safety property). It is impossible to keep the water level within the required distance all the time of the system running due to the reference value changes. Therefore we just want a good reference tracking property and good regulation property of the controller. Informal requirement for the controller is summarised as:

1. The overflow from the lower tank never occur all the time of the system working.
2. Whenever the reference value is changed, the water level should be restabilized within the required distance from new set-point within a finite time bound.
3. In proper working state the system should be stable as long as the reference value does not change.

3.1. Formal Model of Plant

We describe the system behavior with respect to the functions of time hu, hl, u, R, and W. The values $hu(t)$ and $hl(t)$ represent the water level in upper tank and lower tank, respectively, at time t. The value $u(t) \in [0, 1]$ is the signal which is proportional to the throughput of the water pump at time t. The value $R(t)$ represents a desired water level in the lower tank at time t set by human operator discontinuously at a discrete time. The function $W : Time \rightarrow \{0,\ 1\}$ representing the working of the system is defined by $W(t) \overset{def}{=} 1$ iff the system is running at time t.

We assume the right continuity for the state signals R and W.

Assumption A1: For a reference value $R_0 \in U$

$$\Box(\lceil R = R_0 \rceil \Rightarrow \lceil R = R_0 \rceil^o ; \ell > 0) \wedge$$
$$\Box(\lceil W \rceil \Rightarrow \lceil W \rceil^o ; \ell > 0)$$

To derive physical laws which govern the time evolution of the plant we use the first principles of physics.

Mass balance for a tank with cross-section A gives

$$A\dot{h} = -q_{out} + q_{in},$$

where $h \geq 0$ is the water level, $q_{in} \geq 0$, $q_{out} \geq 0$ are the inflow and outflow of the tank, respectively. Bernoulli's law yields $q_{out} = c\sqrt{2gh}$, where c is the cross-section of the outlet hole and g is the acceleration of gravity. Assume that the flow generated by the pump is proportional to the applied voltage u, we get $q_{in} = \beta u$. Applying this equation to the two tanks gives:

$$\begin{aligned} A_1\dot{hu} &= -c_1\sqrt{2ghu} + \beta u, \\ A_2\dot{hl} &= c_1\sqrt{2ghu} - c_2\sqrt{2ghl}, \end{aligned}$$

where A_i denotes the cross-section of the ith tank, and c_i denotes the cross-section of the outlet hole of the ith tank. Let $a_{11} = c_1\sqrt{2g}/A_1$, $a_{21} = c_1\sqrt{2g}/A_2$, $a_{22} = c_2\sqrt{2g}/A_2$ and let $k = a_{22}/a_{21}$. We get:

$$\dot{hu} = -a_{11}\sqrt{hu} + \beta u, \quad \dot{hl} = a_{21}\sqrt{hu} - a_{22}\sqrt{hl}$$

It can be proved in mathematical analysis that

$$hu(t + \Delta) - hu(t) \leq (-a_{11}\sqrt{hu} + \beta)\Delta$$

Translating this equations to our formal specification language we have the following physical laws which govern the time evolution of the plant.

Law1 $\Box(\lceil(\dot{hu} = -a_{11}\sqrt{hu} + \beta u) \wedge$
$(\dot{hl} = a_{21}\sqrt{hu} - a_{22}\sqrt{hl})\rceil)$

Law2 $\Box(\lceil 0 \leq u \leq 1 \rceil^c \wedge \lceil 0 \leq hu \leq h_1 \rceil^c \wedge$
$\lceil 0 \leq hl \leq h_2 \rceil^c)$

Law3 $\Box(\lceil(a_{11} \geq 0) \wedge (a_{21} \geq 0) \wedge$
$(a_{22} \geq 0) \wedge (\beta \geq 0)\rceil^c)$

Law4 $\Box\lceil a_{22}\sqrt{c_2} < a_{11}\sqrt{c_1} < \beta \rceil^c$

Law5 $\Box(\mathbf{e}.hu - \mathbf{b}.hu \leq (-a_{11}\sqrt{\mathbf{b}.hu} + \beta)\ell)$

3.2. Formal Model of Sampling

We assume the control software in consideration behaves in a cyclic manner consisting of the following three phases: input phase to read the values of the state and the event signals from the physical units, decision making phase based on current states of physical units, and output phase to send out control commands to physical units. These cycles are synchronized by a specific time event signal sp of sampling. We make the following assumptions to characterize the sampling behavior of the system.

Assumption A2 $\Box((\lceil W \rceil \wedge \ell = T_s \Rightarrow \Diamond\lceil sp \rceil^o) \wedge (\lceil sp \rceil^o ; \lceil \neg sp \rceil^c ; \lceil sp \rceil^o) \wedge \lceil W \rceil^c \Rightarrow \ell = T_s)$

This captures precisely that during the system operation, sampling event occurs in every T_s time units where T_s is sampling time step.

We introduce the following state signals to represent the sampled values at a particular time. $\overline{hu} : Time \rightarrow U$, and

$\overline{hu}(t)$ is the sampled value of the water level in the upper tank at time t. $\overline{hl} : Time \to U$ and $\overline{hl}(t)$ is the sampled value of the water level in the lower tank at time t. $\overline{u} : Time \to R$ and $\overline{u}(t) \in [0, 1]$ is the sampled value of u at time t.

Assumption A3 Let $x(t)$ be a state signal. Then $\Box(\lceil \updownarrow x\rceil^o \Rightarrow \lceil sp\rceil^o)$, where $x(t) \in \{\ R(t), W(t), \overline{hu}, \overline{hl}, \overline{u}\}$.

This means that events which describes a discrete state changes can only occur at sampling points.

Assumption A4 For $f \in \{hu,\ hl,\ u\}$, $\lceil sp\rceil^o \Rightarrow \lceil \overline{f} = f\rceil^o$

This means that the sampled values are exactly the same as the values of the continuous state variables at the sampling instants.

Assumption A5 For $f \in \{hu,\ hl,\ u\}$
$\Box(\lceil \overline{f} = f\rceil \Rightarrow \lceil \overline{f} = f\rceil^o ; \ell > 0)$ (f is right-continuous).

Now, we give an axiom for sampling called forward induction rule the soundness of which is obvious.

Forward Induction Rule: Let b and b' be Boolean state signals. If $(\lceil sp \wedge b\rceil^o ; \ell > T_s) \Rightarrow ((\lceil b'\rceil^c ; \lceil b\rceil^o) \wedge \ell = T_s)$; *true* then $\lceil sp \wedge b\rceil^o$; *true* $\Rightarrow \lceil b'\rceil^c$.

3.3. Requirement Specification

In the following we introduce some auxiliary state signals to represent critical system states and defines some predicates on the critical states to represent an event occurrence conditions.

It is sufficient to consider only some important states which are critical for reasoning in terms of the safety, performance or efficiency. To do this we partition original state space using the threshold values previously given $0 < C_r < C_1$, where C_r represents the safety bound of the lower tank. Then we define

$$Safe(t) \overset{def}{=} \begin{cases} 1 & \text{if } 0 \le hl < C_r \\ 0 & \text{if } C_r \le hl \end{cases}$$
$$Steady(t) \overset{def}{=} \begin{cases} 1 & \text{if } |R(t) - hl(t)| \le \Delta \\ 0 & \text{if } |R(t) - hl(t)| > \Delta \end{cases}$$

where $R(t)$ is the desired value of the water level in the lower tank at time t (set by the operator), and $hl(t)$ the actual value of the water level in the lower tank, respectively.

We model a controller C as a Boolean state signal $C : Time \to \{0, 1\}$. $C(t) = 1$ means that controller C is used at time t (in this paper, we use the controllers *TMC* and *PID*, and will be formalized later). We introduce the following auxiliary state signal to constrain the reference value change at time t:

$$Disable(t) \overset{def}{=} \begin{cases} 1 & \text{change of } R \text{ is disable at time } t \\ 0 & \text{change of } R \text{ is enable at time } t \end{cases}$$

Any change of the reference value which is larger than the maximal allowable control error causes the change of error signal which larger than the maximal allowable value.

(P1) $\updownarrow_{>\Delta} R \Rightarrow \downarrow Steady$

The state expression $\neg Steady$ denotes an undesirable but unavoidable state of the system because the reference value can be changed at a time by the operator. However, the accumulated time for its presence should be small enough in comparison to the observation time. The following formulas capture the requirements mentioned at the beginning of this section.

1. *Safety Requirement*: $\mathbf{REQ_1}\ : \Box(\lceil W\rceil \Rightarrow \lceil Safe\rceil)$.

2. *Performance Requirement*: $\mathbf{REQ_2}\ : \Box(\lceil W\rceil \wedge \ell > L_o \Rightarrow \int Steady \ge \gamma\ell)$, where $0 < \gamma < 1$ is a given constant.

3. *Stability Performance*: $\mathbf{REQ_3}\ : \Box(\lceil W \wedge \neg \updownarrow R\rceil^c \wedge (\lceil Steady\rceil^o ; \ell > 0) \Rightarrow \lceil Steady\rceil)$

4. Design by Refinement of Requirements

We cannot implement our requirements without having the assumption that the system always starts at a safe state. Therefore, we need another assumption for the initial state:

Assumption A6 $\Box(\lceil\uparrow W\rceil^o \Rightarrow (\lceil hu \le k^2 C_r \wedge hl \le C_r\rceil^o) \wedge (R_{min} \le R \le R_{max})$

So, the assumptions and physical laws for our system are:

$$\textbf{ASS} \overset{def}{=} \textbf{A1} \wedge \textbf{A2} \wedge \textbf{A3} \wedge \textbf{A4} \wedge \textbf{A5} \wedge \textbf{A6}$$
$$\textbf{LAW} \overset{def}{=} \textbf{Law1} \wedge \textbf{Law2} \wedge \textbf{Law3} \wedge \textbf{Law4} \wedge \textbf{Law5}$$

The refinement of the requirements is then to derive more detailed designs such that their conjunction **DESIGN**, together with the above assumptions and laws will verify our the requirements:

$\{$**ASS, LAW, DESIGN**$\} \vdash \textbf{REQ}_1 \wedge \textbf{REQ}_2 \wedge \textbf{REQ}_3$

For the simplicity of presentation, in the sequel, we will assume that the system starts at time 0 (the states at time 0 satisfy the condition for the initial state) and works for ever. Therefore, $\Box\lceil W\rceil$ could be used as an assumption.

It is useful to note that the pump is the only actuator in the system which can enable or disable a violation of the safety requirement. From the **Law1** it follows that if the outflow from the lower tank is larger than the outflow from the upper tank then the water level of the lower tank does not increase:

$$a_{21}\sqrt{hu} \le a_{22}\sqrt{hl} \Leftrightarrow \Box(\lceil \dot{hl} \le 0\rceil)$$

From this fact it follows that the water level of the lower tank can never exceed the danger water level C_r unless the water level of the upper tank exceeds the corresponding water level $k^2 C_r$, where $k = a_{22}/a_{21}$. Let **Design1** be

$$\lceil sp \wedge \overline{hu} - a_{11}\sqrt{\overline{hu}} > k^2 C_r - \beta T_s\rceil^o \Rightarrow \lceil u = 0\rceil^o$$

Let $C_r^*(hu)$ denote $k^2 C_r + a_{11}\sqrt{\overline{hu}}\, T_s - \beta\, T_s$.

Theorem 1 ***ASS*** $\wedge$ ***LAW*** $\wedge$ ***Design1*** $\vdash$ ***REQ***$_1$.

Because of the space limit, the proof of all theorems in this paper is omitted. The readers are referred to [10] for the proof details.

Requirement **REQ**$_2$ is very familiar to the people in the Duration Calculus community, which is a motivation for introducing DC and has been refined into the following designs:

DesignS1 : $\Box\lceil\neg steady\rceil \Rightarrow \ell \le L_d$

DesignS2 : $\Box\lceil\neg steady\rceil; \lceil steady\rceil; \lceil\neg steady\rceil \Rightarrow \ell \ge L_s$

DesignS3 : $(1-\frac{L_d}{L_s})(1-\frac{L_d}{L_0}) > \gamma$

It has been proved formally (see [8]) that

Theorem 2 ***DesignS1*** $\wedge$ ***DesignS2*** $\wedge$ ***DesignS3*** $\vdash$ ***REQ***$_2$

The designs **DesignS1** and **DesignS2** are not detailed enough to be implemented and need to be refined further.

Recall that the reference value may be changed at a (sampling) time randomly by the operator which causes the unsteady. But we can restrict the frequency of the change for the reference value otherwise it is impossible to meet any reasonable requirement and to implement it. We therefore enforce a kind of stability and switching conditions which we hope that **DesignS2** will be satisfied. In order to prevent the trivial implementation of the system, we have the design decision

DesignH1 : $\lceil Steady\rceil \wedge \ell > L_s \Rightarrow \ell = L_s; \lceil\neg Disable\rceil$

DesignH1 is just for the efficiency, and has no role for the correctness. A lower level controller is then designed such that for any time interval in which the reference value does not change, the water level in the lower tank will converge to the required accuracy within L_d $(0 \le L_d < L_s)$ time units. The refinement of the above designs is then:

DesignH2 : $\Box(\lceil Disable\rceil \wedge \ell > L_d \Rightarrow \ell < L_d; \lceil Steady\rceil)$

DesignH3 : $(1-\frac{L_d}{L_s})(1-\frac{L_d}{L_o}) > \gamma$

DesignH4 : $\Box(\lceil\updownarrow R\rceil^o; true \Rightarrow \lceil\neg Disable\rceil^o; \lceil Disable\rceil; true)$

DesignH5 : $\Box\lceil Disable \Rightarrow \neg\updownarrow R\rceil^c$

DesignH6 : $\Box\lceil\neg Steady\rceil \Rightarrow \lceil Disable\rceil$

DesignH7 : $\Box(\lceil\uparrow steady\rceil^o; \lceil steady\rceil \Rightarrow (\lceil Disable\rceil \vee \ell \ge L_s))$

DesignH8 : **REQ**$_3$

Note that we take **REQ**$_3$ as a design decision for the proof of the correctness, and it will be refined further. Let

Design2 $\overset{def}{=}$ **DesignH1** $\wedge$ **DesignH3** $\wedge$ **DesignH4**$\wedge$ **DesignH5** $\wedge$ **DesignH6** $\wedge$ **DesignH7**.

Theorem 3 ***ASS*** $\wedge$ ***LAWS*** $\wedge$ ***Design2*** $\wedge$ ***DesignH2*** $\wedge$ ***DesignH8*** $\vdash$ ***DesignS1*** $\wedge$ ***DesignS2*** $\wedge$ ***DesignS3***

The **DesignH2** and **DesignH8** are not detailed enough for the implementation and need to be refined further.

REQ$_3$ is a stability requirement and can only be verified using control theory and exact knowledge of the control algorithm. For the readability, first we assume that *TMC* (stands for time minimal control) and *PID* are Boolean state signals intended to represent the controllers used by the system to control the input u of the pump. The refinement is then:

DesignT0 : $\Box(\lceil\updownarrow R\rceil^o; \lceil steady\rceil \Rightarrow \lceil PID\rceil)$

DesignT1 : $\Box(\lceil\updownarrow R\rceil^o; \lceil\neg steady\rceil \Rightarrow \lceil TMC\rceil)$

DesignT2 : $\Box(\lceil\uparrow steady\rceil^o; \lceil\neg\updownarrow R\rceil \Rightarrow \lceil PID\rceil)$

DesignT3 : $\Box(\lceil\neg steady\rceil \wedge \lceil TMC\rceil) \Rightarrow \ell < L_d$

DesignT4 : $\Box(\lceil steady\rceil^*; \lceil\neg\updownarrow R\rceil) \wedge \lceil PID\rceil \Rightarrow \lceil steady\rceil$

DesignT5 : $\lceil\neg steady\rceil \Rightarrow \lceil TMC\rceil$

Note that **DesignT5** is provable easily from the others using predicate calculus (but not in duration calculus) because we have to extend the reference interval to the left. Let **Design3** $\overset{def}{=}$ **DesignT1** $\wedge$ **DesignT2** $\wedge$ **DesignT3** $\wedge$ **DesignT4** $\wedge$ **DesignT5**.

Theorem 4 ***ASS*** $\wedge$ ***LAW*** $\wedge$ ***Design3*** $\wedge$ ***Design2*** $\vdash$ ***REQ***$_3$ $\wedge$ ***DesignH2***.

Let **DESIGN** be **Design1** $\wedge$ **Design2** $\wedge$ **Design3**. From the above theorems it follows that **DESIGN** verify the requirements under **ASS** and **LAW**.

Except for **DesignT3** and **DesignT4**, all other designs are detailed enough for translating into a computer program. The translation is easy and is not presented here.

Requirement **DesignT3** can be satisfied if there exists a controller which can steer water level of the lower tank from any initial state hu^o, hl^o to any allowable final state hu^f, hl^f within an time interval which is shorter than L_d time interval.

Let's consider the dynamics of the double tank.

$$\dot{hu} = -a_{11}\sqrt{hu} + \beta u \qquad (1)$$

$$\dot{hl} = a_{21}\sqrt{hu} - a_{22}\sqrt{hl} \qquad (2)$$

and the following optimal control problem:

$$J = \max_{u(t)\in U^*} \int_{t_i}^{t_f} -dt \qquad (3)$$

with the dynamics in 1 and 2, and the constraints

$$\begin{array}{l} [hu(t_i), hl(t_i)]^T = [hu^o, hl^o]^T \wedge \\ (0 \le hu^o \le k^2 C_r) \wedge (0 \le hl^o \le C_r) \\ [hu(t_f), hl(t_f)]^T = [hu^f, hl^f]^T \wedge \\ (|k^2 R - hu^f| + |R - hl^f| \le \Delta) \end{array}$$

where $U^* \overset{def}{=} \{u | (u : Time \to R) \wedge u$ is right continuous$\wedge$ $(0 \le u(t) \le 1)\}$.

The theory of optimal control is well established (see, e.g. [7]) and we can use the Pontryagin maximum principle to show that solution for the optimal control problem (3) exists with that u is piece-wise continuous and satisfies the following necessary condition:

$$H(hu^*, hl^*, u^*) = \underset{u(t) \in U^*}{max} H(hu, hl, u), \tag{4}$$

where $H(hu, hl, u) = -1 + \lambda_1(t)(-a_{11}\sqrt{hu(t)} + \beta u) + \lambda_2(t)(a_{21}\sqrt{hu(t)} - a_{22}\sqrt{hl(t)})$ is the Hamiltonian of the double tank, and λ_1 and λ_2 are the solutions of the following adjoint equations:

$$\dot{\lambda_1(t)} = -\frac{a_{11}}{2\sqrt{hu}(t)}\lambda_1(t) + \frac{a_{21}}{2\sqrt{hu}(t)}\lambda_2(t) \tag{5}$$

$$\dot{\lambda_2(t)} = -\frac{a_{22}}{2\sqrt{hu}(t)}\lambda_2(t) \tag{6}$$

It follows that the time minimal control signal u^*, which maximizes H, satisfies the following conditions which is defined as Boolean state signal *TMC*:

$$\mathit{TMC}(t) \overset{def}{=} \quad u^*(t) = \begin{cases} 1 & \text{if } \lambda_1(t) > 0 \\ 0 & \text{if } \lambda_1(t) \leq 0 \end{cases}$$

It follows that there exists the optimal trajectory which describes time evolution of the double tank state from initial state (hu^o, hl^o) to final state (hu^f, hl^f). Let $(hu^*(u^*, hu^o, hl^o, t), hl^*(u^*, hu^o, hl^o, t))$ be optimal trajectory. Then we can compute the final time t_f from the following equations:

$$hu^f = hu^*(u^*, hu^o, hl^o, t^f)$$
$$hl^f = hl^*(u^*, hu^o, hl^o, t^f)$$

Let $\mathbf{h} = [hl, hu]^T$. The time spent to steer state from $\mathbf{h^o}$ to $\mathbf{h^f}$ is $T^*(\mathbf{h^o}, \mathbf{h^f}) = t^f - t^i$. Define

$$T^*_{max} = \max_{0 \leq \mathbf{h^o}, \mathbf{h^f} \leq [k^2C_r, C_r]^T} T^*(\mathbf{h^o}, \mathbf{h^f})$$

Then we get the following theorem, which says that $T^*_{max} \leq L_d$ is a refinement of **DesignT3** in control theory.

Theorem 5 $T^*_{max} \leq L_d \wedge$ ***LAW*** $\wedge$ ***ASS*** $\vdash$ ***DesignT3***

DesignT4 is a stability requirement and can only be refined using control theory and exact knowledge of the control algorithm. Refinement of the *PID* control is very much involved in Laplace transform and a set of differential equations which are rather complicated. For the space limit, it is omitted it here, and we refer the readers, who are interested in PID control to our report [10] for the details.

5. Conclusion

We have presented an approach to the design of hybrid systems by the combination of several comprehensive formalization techniques. The specification language used at abstract level of the system development is Extended Duration Calculus, which is intuitive, easy to understand, closed to natural languages and precise. We also use EDC to model the abstract controller, which can be refined further in control theory. We use predicate calculus as a background framework to ensure the smooth transit between Duration Calculus and Control Theory. We have shown how our combination techniques work in a double water tank case study, and how a design in DC is refined into the design of controllers. In this way the control logic of reactive model and physical dynamic model can be separated explicitly and be treated independently from each other.

In this approach, compositional verification is performed under different models. A required property is split into a number of conjuncts. Some of them are proved in the discrete formal verification methods using the controller properties. Others are verified in the continuous environment model by control theoretic analysis methods. We should mention here that unfolding our designs and specifications into their semantics in the first order logic would not only make them long and difficult to read but also make the verification unmanageable.

References

[1] Alur, R., C. Courcoubetis, T. A., T. A. Henzinger, and P -H. Ho. Hybrid Automata: An algorithmic Approach to the Specification and Verification of Hybrid systems. LCNS 736, pp. 36-59,Springer Verlag, 1993.

[2] Åström, Karl Johan and Björn Wittenmark. Adaptive Control, Addison-Wesley, 1989.

[3] Bemporad,A and M.Morari. Control of Systems Integrating Logic, Dynamics, and Constraints. Automatica **35**(3),pp.407-427,1999.

[4] Dang Van Hung and Wang Ji. On The Design of Hybrid Control Systems Using Automata Models. LCNS 1180, Springer, 1996, pp. 156-167.

[5] B. Dutertre. On First Order Interval Temporal Logic Report no. CSD-TR-94-3 Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England, 1995.

[6] Leeb Gunter and Nancy Lynch. Proving Safety Properties of the Steam Boiler Controller. LNCS 1165, Springer-Verlag, pp.318-338, 1996.

[7] Macki, JackM and Aaron Strauss. Introduction of Optimal Control Theory. Springer-Verlag, pp.66-73, 1995.

[8] Zhou Chaochen, C. A. R. Hoare and A. P. Ravn. A Calculus of Durations. *Information Processing Letters*, 40(5), pp. 269-276, 1991.

[9] Zhou Chaochen, A. P. Ravn and M. R. Hansen An Extended Duration Calculus For Hybrid Real-Time Systems. LCNS 736, pp. 36-59, Springer Verlag, 1993.

[10] Hong Ki Thae and Dang Van Hung. Formal Design of Hybrid Control Systems: Duration Calculus Approach. Technical Report 221, UNU/IIST, P.O. Box 3058, Macau, November 2000.