Modeling Cyber Physical Systems: A case study of Floodgate management system using Hybrid automata and State space analysis

Viyyapu Durga Prasad

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Electrical Engineering

June 2015

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

(Signature)

Viyyapu Durga Prasad
EE12M1042
(Roll No.)

Approval Sheet

This Thesis entitled Modeling Cyber Physical Systems: A case study of Floodgate management system using Hybrid automata and State space analysis by Viyyapu Durga Prasad is approved for the degree of Master of Technology from IIT Hyderabad

(Dr. Sumohana Channappayya) Examiner
Department of Electrical Engineering
IIT, Hyderabad

(Dr. P. Rajalakshmi) Examiner
Department of Electrical Engineering
IIT Hyderabad

(Dr. Mohammed Zafar Ali Khan) Adviser
Department of Electrical Engineering
IIT Hyderabad

(Dr.G.V.V.Sharma) Chairman
Department of Electrical Engineering
IIT Hyderabad

Acknowledgements

I sincerely express my gratitude to everyone who supported me throughout the course of this work. I am thankful to my advisor Dr.Mohammed Zafar Ali Khan for his aspiring guidance, invaluably constructive criticism and friendly advice during the work.

A significant part of this thesis was benefited from the discussions that took place in the weekly meetings of Cyber Physical Systems project at IIT Hyderabad. I am thankful to all the participants of those meetings, who attended the presentations and shared their views from time to time.

I am also thankful to my friends at IIT Hyderabad. Although it is not possible to mention all of them, A. Venkat Reddy deserve a special mention for the enormous amount of help he provided.

Dedication

To my family and to everyone who has been part of my learning experiences.

Abstract

This thesis seeks methods for modeling cyber physical systems(CPSs) and the related issues. They enable innovation in a wide range of domains including robotics, smart homes, vehicles, and buildings, medical implants, and future-generation sensor networks. Advances in CPS will enable capability, adaptability, scalability, resiliency, safety, security, and usability that will far exceed the simple embedded systems of today. In this thesis two methods are used to model and analyze the flood gate management system (FMS). Specific technologies described include hybrid automata and State space analysis, the use of domain-specific ontologies to enhance modularity, and the joint modeling of functionality and implementation.

In order to realize cyber physical system in hybrid automata, several engineering aspects need attention. This thesis focuses on a few related modeling issues. Specifically, compact representation and realization in state space analysis of physical systems are discussed. Further, the proposed hybrid autoamta for flood gate management system is shown to be safe and minimize the floodings.

Contents

Chapter 1

Introduction

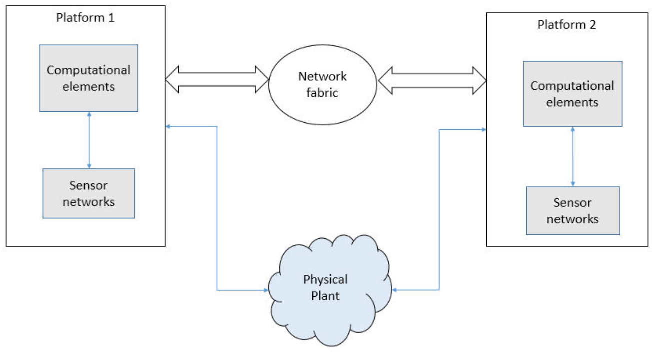

Figure 1.1: General Structure of Cyber Physical System

In recent years, Cyber Physical Systems got good attention from researchers because of the diverse applications. Today, a precursor generation of cyber-physical systems can be found in areas as diverse as aerospace, automotive, chemical processes, civil infrastructure, energy, healthcare, manufacturing, and transportation, entertainment, and consumer appliances. A cyber-physical system (CPS) is a system of collaborating computational elements controlling physical entities. Advances in CPS will enable capability, adaptability, scalability, resiliency, safety, security, and usability that will far exceed the simple embedded systems of today. CPS technology will transform the way people interact with engineered systems just as the Internet has transformed the way people interact with information. New smart CPS will drive innovation and competition in sectors such as agriculture, energy, transportation, building design and automation, healthcare, and manufacturing. These are integration of physical process and the computational elements which monitor and control the physical processes. The design of such systems, therefore, requires understanding the joint dynamics of computers, software, networks, and physical processes. It is this study of joint dynamics that sets this discipline apart. Compounding the challenge,When studying Cyber Physical Systems, certain key problems emerge that are rare in so-called general-purpose computing [3]. For example, in general-purpose software, the time it takes to perform a task is an issue of performance, not correctness. It is not incorrect to take longer to perform a task. It is merely less convenient and therefore less valuable. In Cyber Physical Systems, the time it takes to perform a task may be critical to correct functioning of the system. In Cyber Physical Systems, moreover, many things happen at once. Physical processes are compositions of many things occurring at the same time, unlike software processes, which are rooted in sequential steps.[4] describe computer science as procedural epistemology, knowledge through procedure. In the physical world, by contrast, processes are rarely procedural. The main challenges of these Cyber Physical Systems are intrinsic heterogeneity,concurrency and sensitive to timing make the design of these Systems complex.

There are three main parts in Figure 1.1. First, the physical plant is the physical part of a cyber-physical system. It is simply that part of the system that is not realized with computers or digital networks. It can include mechanical parts, biological or chemical processes, or human operators. Second, there are one or more computational platforms, which consist of sensors, actuators, one or more computers, and (possibly) one or more operating systems. Third, there is a network fabric, which provides the mechanisms for the computers to communicate. Together, the platforms and the network fabric form the cyber part of the cyber-physical system.

The challenges and opportunities for CPS are thus significant and far-reaching,[1] [2]. New relationships between the cyber and physical components require new architectural models that redefine form and function. The designing and implementing of most of the cyber-physical systems involves three major parts [5]. They are modeling, design, and analysis. **Modeling** is the process of gaining a deeper understanding of a system through imitation. Models imitate the system and reflect properties of the system. Models specify what a system does. **Design** is the structured creation of artifacts. It specifies how a system does what it does. **Analysis** is the process of gaining a deeper understanding of a system through dissection. It specifies why a system does what it does (or fails to do what a model says it should do).

In this thesis we focused on the modeling part of the Cyber Physical Sytems. A model of a physical system is a description of certain aspects of the system that is intended to yield insight into properties of the system. Modeling is understood as the abstraction of reality,resulting in the formal specification of a conceptualization and underlying assumptions and constraints.In model-based design [6] and model-driven development [7], models play an essential role in the design process. They form the specifications for systems and reflect the evolution of the system design. They enable simulation and analysis, both of which can result in earlier identification of design defects than prototyping. Models can have formal properties. We can say definitive things about models. For example, we can assert that a model is deterministic which means for a specific input it will always produce specific output. It is not possible to give such assertion with any physical realization of a system. If model is a good abstraction of the physical system then the definitive assertion about the model gives the confidence in the physical realization. Such confidence is useful, particularly for embedded systems where malfunctions can lead to problems. Studying models of systems gives us insight into how those systems will behave in the physical world.

In this thesis we will illustrate the model of floodgate management system using hybrid automaton and state space analysis. Flooding is one of the most damaging of natural disasters. Structural approaches to flood management consist of reservoirs and dams equipped with floodgates, along with protocols for their operation. However, in spite of the infrastructure being in place,floods can occur because of flaws in the floodgate operation protocols or human error in its implementation. These errors may happening mainly because of misjudging the timing of actuation of flood gates. So if we have a central control system which operates the gates of reservoirs based on the sensor data we can minimize the flooding problem.

Central to this discussion lies the Hybrid Automata [8]. A hybrid automaton is a model of a system with interacting continuous and discrete dynamics. Hybrid automata is a mathematical method which is used to model and analyze the Hybrid Systems. The importance of systems with interacting digital and analog computations is increasing dramatically. Areas such as aeronautics, automotive vehicles, bio engineering, embedded software, process control, and transportation are growing tremendously. Hybrid automata have proved to be an efficient way to model systems with both continuous and discrete dynamics. Their rich structure allow them to accurately predict the behavior of quite complex systems. Based on computer science and control theory, tools are now evolving for analyzing and designing hybrid systems within the hybrid automata framework.As embedded computing becomes ubiquitous, hybrid systems are increasingly employed in safety-critical applications, making reliability a prime concern. For this purpose, the hybrid automaton has been proposed as a formal model for hybrid systems.

1.1 Scope of the Thesis

Scope of this thesis is to model the flood gate management system using the hybrid automaton and partially validate the floodgate management system using the state space analysis. Those include proposing the new system model for flood gate management system and analyze this moodel using modeling techniques. For example, van der Schaft et al,have presented the system model of hybrid systems [9]. Lygeros et al,have presented a system specifications for hybrid systems [10].

1.2 Literature Survey

As discussed in the previous section, this thesis studies modeling of the flood gate management system by hybrid automata approach. The literature available in this context can roughly be divided into a few categories such as (a)modeling cyber physical systems(b) A hybrid automata approach to analyze the Hybrid Systems (c)finally, State space analysis for analysing the disctre systems. In addition to these, a few researchers also discuss modeling of systems by different models. Although this thesis does not propose any concrete solution for the modeling the floodgate management system using state space approach the related literature is studied to some extent as it gives useful insights for finding efficient representation methods.

To begin with, importance of modeling of various type of physical systems discussed. A significant amount of the literature employing Hybrid automata for modeing the hybrid systems are discussed. Such approaches are reviewed in [11] [12].

Chapter 2

Modeling Floodgate Management System

This chapter explores the basic idea of the system model for flood gate management system. Next,we will understand what exactly the hybrid automata mathematically [13]. A major part of this chapter is dedicated for the proposed flood gate system model by hybrid automaton.

2.1 System Model

We begin this section by defining some terminology and the statement of the problem addressed in this thesis. There is series of n reservoirs r_1, r_2, r_3, ...,r_n. The last reservoir r_n drains into river is r_{n+1}. There is water channel $e_{i,i+1}$ between reservoirs r_i to r_{i+1} for $1 \leq i \leq n$. Water always flow from r_i to r_{i+1} and not in the other direction. There is a floodgate g_i at reservoir r_i installed at the beginning of the channel $e_{i,i+1}$. The floodgate g_i can be open, which results in a flow of water from r_i to r_{i+1} at a rate $f_{i,i+1}$ units of volume per unit time (for $1 \leq i \leq n$). When it is closed, the flow stops.

Each reservoir r_i has an upper threshold of water level U_i associated with it, beyond which if the water level rises,the reservoir floods. However, a strategy can assume a more conservative upper threshold $u_i \leq U_i$, as will be seen later. There is also a lower threshold.

For each channel $e_{i,i+1}$, there exists a delay $d_{i,i+1}$ for the water to travel from r_i to r_{i+1}. Thus, when water is released from r_i, it reaches r_{i+1} after $d_{i,i+1}$ time units. Finally, associated with each floodgate g_1 is a delay t_i incurred for opening the floodgate. Note that all the above terminology and parameters are infrastructural in nature.

Now, we define terminology for the dynamical quantities, typically collected by sensors. We denote by x_i and dx_i the current water level and its rate of change respectively at reservoir r_i. Thus, if precipitation at r_i is p_i,

$$dx_i = p_i + f_{i-1,i} - f_{i,i+1} \tag{2.1}$$

For $2 \leq i \leq n$ and $dx_i = p_i - f_{i,i+1}$ for $i = 1$. We assume that each reservoir is equipped with sensors that report these parameters. The sensor data is collected at a central control room. The control room can actuate the floodgates into opening or closing.

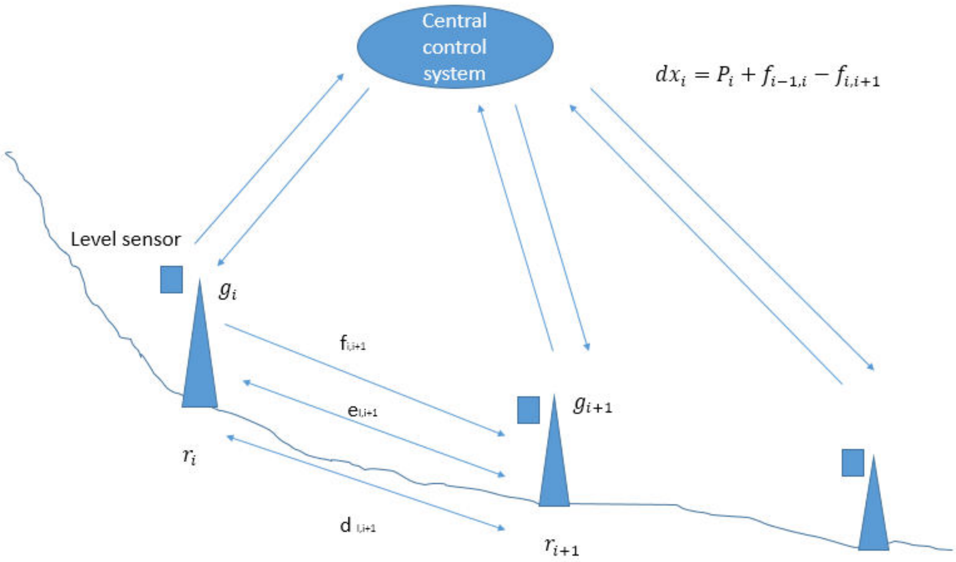

Figure 2.1: Floodgate system model with three reservoirs

Figure 2.1 depicts the floodgate management system with three reservoirs r_1, r_2, r_3(say). The last reservoir r_3 drains into river is r_4. There is water channel e_1 between reservoirs r_1 to r_2. Water always flow from r_1 to r_2 and not in the other direction. There is a floodgate g_1 at reservoir r_1 installed at the beginning of the channel $e_{1,2}$. The floodgate g_1 can be open, which results in a flow of water from r_1 to r_2 at a rate $f_{1,2}$ units of volume per unit time. When it is closed, the flow stops.

2.2 Hybrid Automata

As earlier mentioned Hybrid automata is model of the systems which exhibits both continues and discrete behavior. A Hybrid Automaton is used to model the Hybrid Systems. Hybrid Systems [14] [15] are dynamical systems with interacting continuous-time dynamics (differential equations) and discrete-event dynamics (automata).

A Hybrid Automaton H is a collection $H = (Q, X, f, Init, Dom, E, G, R)$
where

- Q is the finite collection of discrete variables with values in Q, $Q = \{q_1, q_2, ...q_n\}$.it represents the number of discrete states, the system exhibits.
- X is the finite collection of continuous variables with values in $X = R^n$, $X = \{x_1, x_2, ...x_n\}$. The number n is called the dimension of H.it represents the number of continuous states the system have.
- f(.,.): $Q \times X \rightarrow R^n$ is the vector field.it represents the with which function the variables are changing.
- $Init \subseteq Q \times R^n$ is the set of initial states.it represents the starting point of the process of the system.

- Dom(.):$Q \rightarrow p(X)$ is the domain of H. it represents the domain of the each state.
- $E \subseteq Q \times Q$ is a set of edges.it represents the different edges of the system.
- G(.):$E \rightarrow p(X)$ is a guard condition.it represents the conditions for changing from one state to another state.
- R(.,.): $E \times X \rightarrow p(X)$ is a reset map.it represents the reset for the system model.

There are number of physical systems modeled using hybrid automata. In [16],Thomas A. Henzingerz explains the how to model the hybrid systems example of water tank model.

Example (Water Tank): The two tank system, shown in Figure 2.2, consists of two tanks containing water. Both tanks are leaking at a constant rate. Water is added at a constant rate to the system through a hose, which at any point in time is dedicated to either one tank or the other. It is assumed that the hose can switch between the tanks instantaneously.

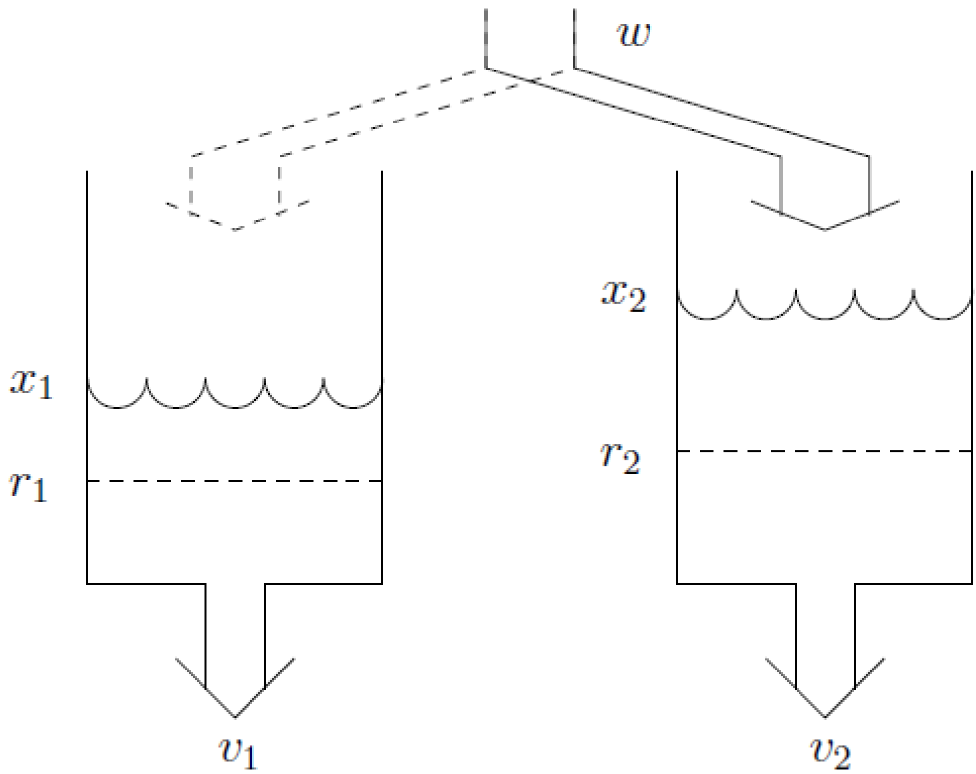

Figure 2.2: Water Tank System.

For $i \in 1, 2$, let x_i denote the volume of water in Tank i and $v_i > 0$ denote the constant flow of water out of Tank i. Let w denote the constant flow of water into the system. The objective is to keep the water volumes above r_1 and r_2, respectively, assuming that the water volumes are above r_1 and r_2 initially. This is to be achieved by a controller that switches the inflow to Tank 1 whenever $x_1 \leq r_1$ and to Tank 2 whenever$x_2 \leq r_2$. It is straightforward to define an autonomous hybrid automaton to describe this process:

- Q $= \{q_1, q_2\}$;
- $X = R^2$;
- Init $=Q \times \{x \in R^2 | x1 \geq r1 and x2 \geq r2\}$;
- $f(q_1, x) = (wv_1, v_2)$ and $f(q_2, x) = (v_1, wv_2)$;
- $Dom(q_1) = \{x \in R^2 | x_2 \geq r_2\}$ and $Dom(q_2) = \{x \in R^2 | x_1 \geq r_1\}$;
- $R(q_1, x) = (q_2, x) if x_2 \leq r_2, R(q_2, x) = (q_1, x) if x_1 \leq r_1andR(q, x) =$; otherwise.

2.3 Hybrid automaton representation of reservoir

In this section we will represent the hybrid automation for the different reservoirs of the floodgate management system model shown in Figure 2.1 and analyze the model briefly.

2.3.1 Hybrid Automaton for first reservoir

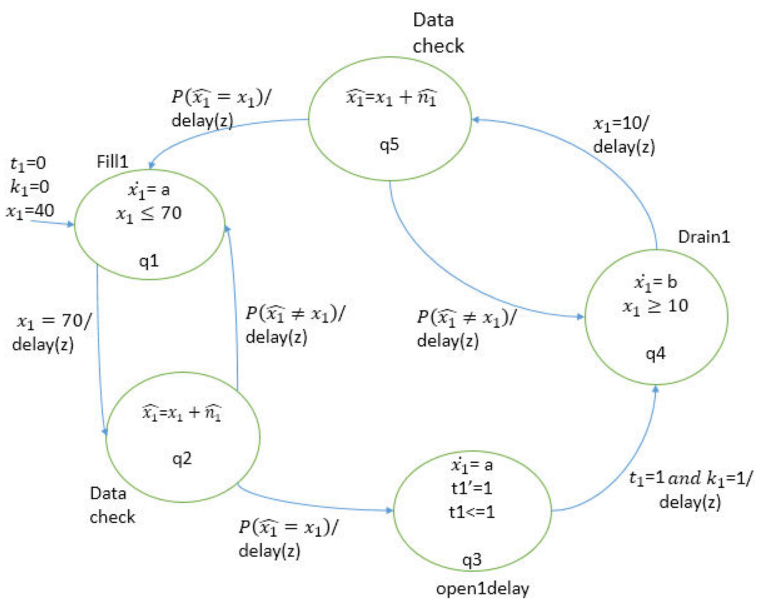

Figure 2.3: Hybrid Automata for first reservoir.

Figure 2.3 shows the hybrid automaton for the first reservoir. It undergoes five discrete states. They are *fill1, datacheck, open1delay, drain1, datacheck.*

According to the figure 2.3,the initial level is 40. When the inflow is there, the water level x_1 rises at the rate of a. If the water level reaches the threshold level($x_1 = 70$),the level sensor at the gate send data to the central control system and the system is in data check state. In *datacheck* state the central control system checks whether the data is correct or not. If it is correct the system will go to next state i.e *open1delay* where the water will continue to rise with the previous rate until the gate is open. Once the gate is open, the water will drain to lower stream and make the shared variable k_1 to 1. This shared variable will used to inform the other reservoirs about the release of the water. Now, the system is in *Drain1* state. it will continue in the same state until the water level reaches the lower threshold level i.e 10. Once the lower threshold is reached the sensor again send data to the central control system and the system goes to *datacheck* state. In this state the central control system checks the data sent by the sensor and the system goes to next state or previous state based on the sensor data. This process will repeat every time when the level reaches the threshold and make the reservoir 1 to maintain the constant level

In this process there will be delay in the communication network between sensor and the central control system. It is represented by z in the figure 2.3

Now, we will look at the analysis part of the hybrid automaton of the reservoir 1.

- $Q = \{q_1, q_2, q_3, q_4, q_5\}$
- $X = R^2$
- $Init = q_1$
- $f(q_1, x) = (a, 0), f(q_2, x) = (a, 0), f(q_3, x) = (a, 1), f(q_4, x) = (b, 0), f(q_5, x) = (b, 0)$
- $Dom(q_1) = \{x \in R^2/x_1 \leq 70\}, Dom(q_2) = \{x \in R^2/x_1 = 70\}, Dom(q_3) = \{x \in R^2/t_1 \leq 1\}, Dom(q_4) = \{x \in R^2/x_1 \geq 10\}, Dom(q_5) = \{x \in R^2/x_1 = 70\}$
- $R(q_1, x) = (q_2, x)$,if $x_1 \geq 70, R(q_2, x) = (q_1, x)$,if $x_1 \neq 70, R(q_2, x) = (q_3, x)$,if $x_1 = 70, R(q_3, x) = (q_4, x)$,if $t_1 \geq 1, R(q_4, x) = (q_5, x)$,if $x_1 \leq 10, R(q_5, x) = (q_4, x)$,if $x_1 \neq 10, R(q_5, x) = (q_1, x)$,if $x_1 = 10$

2.3.2 Hybrid Automaton for second reservoir

Figure 2.4 shows the hybrid automaton for the second reservoir. It undergoes nine discrete states. They are *fill2, WaterArrDelay2,Open1fill2, datacheck, open2delay, drain,open2delay, datacheck, datacheck.*

The automaton for the second reservoir, reservoir 2, is some what different because of the fact that it is downstream to reservoir 1 but upstream to reservoir 3. In Figure 2.4 it is shown that, the reservoir is also in the mode *Fill2* initially, with the initial water level at 20 units, and rising at 'a' units per unit time.

This normal rate of filling up of the reservoir is disturbed by two events the water reaching the upper threshold or the water released by the upstream reservoir reaching reservoir 2. The rate of rise of the water levels differs for these two scenarios. In the first scenario i.e the water reaching upper threshold, the rate of rise of water level is same as the first reservoir. When the inflow is there, the water level x_2 rises at the rate of a. If the water level reaches the threshold level($x_2 = 70$),the level sensor at the gate send data to the central control system and the system is in *datacheck* state. In *datacheck* state the central control system checks whether the data is correct or not. If it is correct the system will go to next state i.e *open2delay* where the water will continue to rise with the previous rate until the gate is open. Once the gate is open, the water will drain at 'c' rate to lower stream

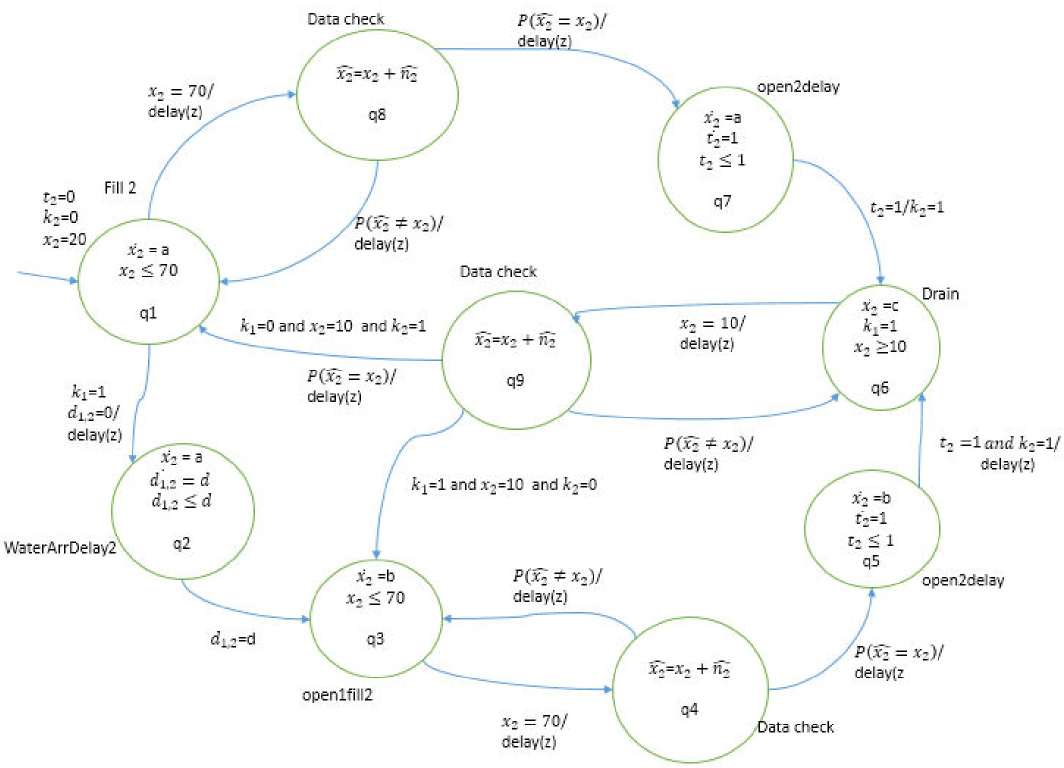

Figure 2.4: Hybrid Automata for second reservoir

and make the shared variable k_2 to 1. this shared variable will used to inform the other reservoirs about the release of the water. Now, the system is in *Drain* state. It will continue in the same state until the water level reaches the lower threshold level i.e 10. Once the lower threshold is reached the sensor again send data to the central control system and the system goes to *datacheck* state. In this state the central control system checks the data sent by the sensor and the system goes to next state or previous state based on the sensor data.

In the second scenario i.e the water is released by the upstream reservoir reaching reservoir 2,the rate of rise of water level is different from first reservoir. Initially, the automaton is in *fill2* state and the water level is continue to rise with 'a' rate. If the reservoir 2 gets to know through the shared variable k_1 that the flood gate is open at the reservoir 1,the automaton jumps to the *WaterArrDelay2* where it waits for the water to reach from the first reservoir to the second reservoir. After the channel delay of $d_{1,2} = d$ the water reaches the reservoir 2 and the water level at the reservoir continue to rise at the rate of 'b' in the *Open1fill2* state. If the water level reaches the threshold level($x_2 = 70$),the level sensor at the gate send data to the central control system and the system is in *datacheck* state. In *datacheck* state the central control system checks whether the data is correct or not. If it is correct the system will go to next state i.e *open2delay* where the water will continue to rise with the previous rate until the gate is open. Once the gate is open, the water will drain at 'c' rate to lower stream and make the shared variable k_2 to 1. This shared variable will used to inform the other reservoirs about the release of the water. Now, the system is in *Drain* state. It will continue in the same state until the water level reaches the lower threshold level i.e 10. Once the lower threshold is reached the sensor again send data to the central control system and the system goes to *datacheck* state. In the data check state, the central control system checks the water level of the reservoir and also the status of the flood gate status of the reservoir 1. If the floodgate at reservoir 1 is open the automaton goes to the *Open1fill2* state otherwise the automaton jumps to *fill2* state.

- where $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$
- $X = R^3$
- $Init = q_1$
- $f(q_1, x) = (a, 0, 0), f(q_2, x) = (a, 0, 2), f(q_3, x) = (b, 0, 0), f(q_4, x) = (b, 0, 0), f(q_5, x) = (b, 1, 0), f(q_6, x) = (c, 0, 0), f(q_7, x) = (a, 1, 0),$
  $f(q_8, x) = (a, 0, 0), f(q_9, x) = (c, 0, 0)$
- $Dom(q_1) = \{x \in R^3/x_2 \leq 70\}, Dom(q_2) = \{x \in R^3/d \leq 2\}, Dom(q_3) = \{x \in R^3/x_2 \leq 70\}, Dom(q_4) = \{x \in R^3/x_2 = 70\}, Dom(q_5) = \{x \in R^3/t_2 \leq 1\}, Dom(q_6) = \{x \in R^3/x_2 \geq 10\}, Dom(q_7) = \{x \in R^3/t_2 \leq 1\}, Dom(q_8) = \{x \in R^3/x_2 = 70\}, Dom(q_9) = \{x \in R^3/x_2 = 10\}$
- $R(q_1, x) = (q_2, x)$,if $x_2 \geq 70, R(q_2, x) = (q_3, x)$,if $d_2 \geq 2, R(q_3, x) = (q_4, x)$,if $x_2 > 70, R(q_4, x) = (q_3, x)$,if $x_2 \neq 70, R(q_4, x) = (q_5, x)$,if $x_2 = 70, R(q_5, x) = (q_6, x)$,if $t_2 > 1, R(q_6, x) = (q_9, x)$,if $x_2 < 10, R(q_7, x) = (q_6, x)$,if $t_2 > 1, R(q_8, x) = (q_7, x)$,if $x_2 = 70, R(q_8, x) = (q_1, x)$,if $x_2 \neq 70, R(q_9, x) = (q_1, x)$,if $x_2 = 10, R(q_9, x) = (q_6, x)$,if $x_2 \neq 10, R(q_9, x) = (q_3, x)$,if $x_2 = 10$

2.3.3 Hybrid Automaton for third reservoir

Figure 2.5 shows the hybrid automaton for the second reservoir. It undergoes nine discrete states. They are *fill3, WaterArrDelay3,Open2fill3, datacheck, open3delay, drain,open3delay, datacheck, datacheck.*

The automaton for the third reservoir, reservoir 3, is same as reservoir 2. In Figure 2.5, it is shown that, the reservoir is also in the mode *Fill3* initially, with the initial water level at 20 units, and rising at 'a' units per unit time.

This normal rate of filling up of the reservoir is disturbed by two events the water reaching the upper threshold or the water released by the upstream reservoir reaching reservoir 3. The rate of rise of the water levels differs for these two scenarios.

In the first scenario i.e the water reaching upper threshold, the rate of rise of water level is same as the first reservoir. When the inflow is there, the water level x_2 rises at the rate of a. If the water level reaches the threshold level($x_3 = 70$),the level sensor at the gate send data to the central control system and the system is in *datacheck* state. In *datacheck* state the central control system checks whether the data is correct or not. If it is correct the system will go to next state i.e *open3delay* where the water will continue to rise with the previous rate until the gate is open. Once the gate is open, the water will drain at 'c' rate to lower stream and make the shared variable k_3 to 1. this shared variable will used to inform the other reservoirs about the release of the water. Now, the system is in *Drain* state. It will continue in the same state until the water level reaches the lower threshold level i.e 10. Once the lower threshold is reached the sensor again send data to the central control system and the system goes to *datacheck* state. In this state the central control system checks the data sent by the sensor and the system goes to next state or previous state based on the sensor data.

In the second scenario i.e the water is released by the upstream reservoir reaching reservoir 3,the rate of rise of water level is different from first reservoir. Initially, the automaton is in *fill3* state and the water level is continue to rise with 'a' rate. If the reservoir 2 gets to know through the shared variable k_1 that the flood gate is open at the reservoir 2,the automaton jumps to the *WaterArrDelay3* where it waits for the water to reach from the second reservoir to the third reservoir. After the channel delay of $d_{2,3} = d$ the water reaches the reservoir 3 and the water level at the reservoir continue to rise at the rate of 'b' in the *Open2fill3* state. If the water level reaches the threshold level($x_3 = 70$),the level sensor at the gate send data to the central control system and the system is in *datacheck* state. In *datacheck* state the central control system checks whether the data is correct or not. If it is correct the system will go to next state i.e *open3delay* where the water will continue to rise with the previous rate until the gate is open. Once the gate is open, the water will drain at 'c' rate to lower stream and make the shared variable k_3 to 1. This shared variable will used to inform the other reservoirs about the release of the water. Now, the system is in *Drain* state. It will continue in the same state until the water level reaches the lower threshold level i.e 10. Once the lower threshold is reached the sensor again send data to the central control system and the system goes to *datacheck* state. In the *datacheck* state, the central control system checks the water level of the reservoir and also the status of the flood gate status of the reservoir 2. If the floodgate at reservoir 2 is open the automaton goes to the *Open2fill3* state otherwise the automaton jumps to *fill3* state.

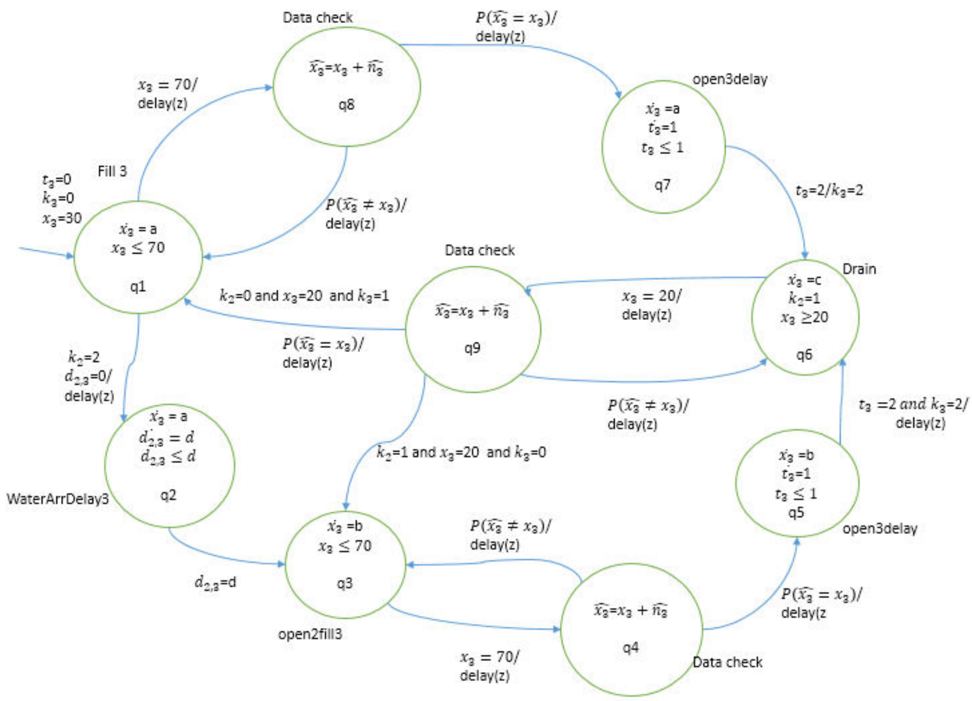

Figure 2.5: Hybrid Automata for third reservoir

- where $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$
- $X = R^3$
- $Init = q_1$
- $f(q_1, x) = (a, 0, 0), f(q_2, x) = (a, 0, 2), f(q_3, x) = (b, 0, 0), f(q_4, x) = (b, 0, 0), f(q_5, x) = (b, 1, 0), f(q_6, x) = (c, 0, 0), f(q_7, x) = (a, 1, 0),$
  $f(q_8, x) = (a, 0, 0), f(q_9, x) = (c, 0, 0)$
- $Dom(q_1) = \{x \in R^3/x_3 \leq 70\}, Dom(q_2) = \{x \in R^3/d \leq 2\}, Dom(q_3) = \{x \in R^3/x_3 \leq 70\}, Dom(q_4) = \{x \in R^3/x_3 = 70\}, Dom(q_5) = \{x \in R^3/t_3 \leq 1\}, Dom(q_6) = \{x \in R^3/x_3 \geq 10\}, Dom(q_7) = \{x \in R^3/t_3 \leq 1\}, Dom(q_8) = \{x \in R^3/x_3 = 70\}, Dom(q_9) = \{x \in R^3/x_3 = 10\}$
- $R(q_1, x) = (q_2, x)$,if $x_3 \geq 70, R(q_2, x) = (q_3, x)$,if $d_3 \geq 2, R(q_3, x) = (q_4, x)$,if $x_3 > 70, R(q_4, x) = (q_3, x)$,if $x_3 \neq 70, R(q_4, x) = (q_5, x)$,if $x_3 = 70, R(q_5, x) = (q_6, x)$,if $t_3 > 1, R(q_6, x) = (q_9, x)$,if $x_3 < 10, R(q_7, x) = (q_6, x)$,if $t_3 > 1, R(q_8, x) = (q_7, x)$,if $x_3 = 70, R(q_8, x) = (q_1, x)$,if $x_3 \neq 70, R(q_9, x) = (q_1, x)$,if $x_3 = 10, R(q_9, x) = (q_6, x)$,if $x_3 \neq 10, R(q_9, x) = (q_3, x)$,if $x_3 = 10$

Finally to conclude the chapter, the modeling of the floodgate management system is studied using hybrid automata.

Chapter 3

State space realization of Flood gate management System

Models of Cyber Physical Systems include both discrete and continuous components. Loosely speaking, continuous components evolve smoothly, while discrete components evolve abruptly. Hybrid systems allow for time domains that have both continuous and discrete parts.

This chapter gives the insight into the state space analysis for discrete time systems and the state space realization of the flood gate management system. This method enables us to find the stability and controllability of the systems which undergoes different states. The state-space representation provides a compact and convenient way to model and analyze systems with multiple inputs and outputs.

3.1 State space analysis

A state-space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations. State space Description provides the dynamics as a set of coupled first order differential equations in a set of internal variables known as state variables.State is the smallest set of variables, so that the knowledge of these variable at initial time t_0,together with the knowledge of input for time $t \geq t_0$,determine the behavior of the system.

Major advantages of the state space analysis is,once the system state is known,output of the system can be immediately obtained from the output equation. Thus solution of the state equations provides the information about the system state as well as the system output. State equations are the equations relating the current state and output of a system to its current input and past states.

The most general state-space representation of a continuous time linear system is written in the following form

$$\dot{X}(t) = AX(t) + BU(t)$$
$$Y(t) = CX(t) + DU(t)$$

Here
A is the system matrix
B is the input matrix
C is the output matrix
D is the transmission matrix

3.1.1 Discrete-time State-Space Realizations

A discrete system is a system with a countable number of states.Because discrete systems have a countable number of states, they may be described in precise mathematical models [27] [28] .The most general state-space representation of a linear discrete-time systems is given by

$$\begin{aligned} x((k+1)T) &= A(kT)x(kT) + B(kT)u(kT) \\ y(kT) &= C(kT)x(kT) + D(kT)u(kT) \end{aligned}$$

where
u(kT) is the input vector
y(kT) is the output vector
x(kT) is the state vector
T is the sampling period.

If the linear discrete-time system is time invariant, then it can by represented by the following state-space equations

$$\begin{aligned} x((k+1)T) &= Ax(kT) + Bu(kT) \\ y(kT) &= Cx(kT) + Du(kT) \end{aligned}$$

3.1.2 State-space Representation of Time-invariant Scalar Difference Equations

Consider the following scalar difference equation

$$y(k+n) + a_1 y(k+n-1) + a_2 y(k+n-2) + ... + a_n y(k) = bu(k)$$

where k denotes the k_{th} sampling instant, y(k) is the system output at the k_{th} sampling instant, and u(k) is the input at the k_{th} sampling instant.

Let us define

$$
\begin{aligned}
x_1(k) =& y(k) \\
x_1(k+1) =& x_2(k) \\
x_2(k+1) =& x_3(k) \\
x_3(k+1) =& x_4(k) \\
& ... \\
& ... \\
x_{n-1}(k+1) =& x_n(k) \\
x_n(k+1) =& -a_1 x_n(k) - a_2 x_{n-1}(k) - a_3 x_{n-2}(k) - .... - a_n x_1(k) + bu(k)
\end{aligned}
$$

$$
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ . \\ . \\ . \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & . & . & . & . & 0 \\ 0 & 0 & 1 & . & . & . & . & 0 \\ 0 & 0 & 0 & 1 & . & . & . & 0 \\ . & & & & & & & \\ ... & & & & & & & \\ 0 & 0 & 0 & 0 & . & . & . & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & . & . & . & . & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ . \\ . \\ . \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ b \end{bmatrix} u(k)
$$

$$
y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & . & . & 0 & \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ . \\ . \\ . \\ . \\ . \\ x_n(k) \end{bmatrix}
$$

3.1.3 State space representation of a flood gate management system

This section presents the state space realization of the flood gate management system. The floodgate management system can be modeled by using discrete state space representation

$$
\begin{aligned}
x_1(k) &= y(k) \\
x_1(k+n_1) &= x_2(k) \\
x_2(k+n_2) &= x_3(k) \\
x_3(k+n_3) &= x_4(k) \\
x_4(k+n_4) &= x_5(k) \\
x_5(k+n_5) &= -a_1x_5(k) - a_2x_4(k) - a_3x_3(k) - a_4x_2(k) - a_5x_1(k) + bu(k)
\end{aligned}
$$

$$
\begin{bmatrix} x_1(k+n_1) \\ x_2(k+n_2) \\ x_3(k+n_3) \\ x_4(k+n_4) \\ x_5(k+n_5) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -a_5 & -a_4 & -a_3 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ b \end{bmatrix} u(k)
$$

$$
y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix}
$$

Here

$$
A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -a_5 & -a_4 & -a_3 & -a_2 & -a_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ b \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

Chapter 4

Summary and Discussion

The two main focus areas of this work, described in section 1.1 are (i) modeling the cyber physical systems using hybrid automata , which enables one to analyze(ii) the systems which exhibits the both discrete and continuous nature. The thesis reports findings on the modeling hybrid systems.

Chapter 2 has presented an empirical study on flood gate management system. It was observed that hybrid automata is very convenient to represent the flood gate management system. Further, state space analysis useful to partially represent the floodgate management system. The next step is to do further study in the state space realization of the cyber physical systems.

References

[1] E. A. Lee, Cyber physical systems: Design challenges,[in Proc. Int. Symp.Object/Component/Service-Oriented Real-Time Distrib. Comput., Orlando, FL, 2008,pp. 363369. [Online]. Available:http://dx.doi.org/10.1109/ISORC.2008.25.

[2] E. A. Lee, Disciplined heterogeneous modeling,[in Proc. Model Driven Eng. Lang. Syst., D. C. Petriu, N. Rouquette, and O. Haugen, Eds., 2010, pp. 273287.[Online]. Available: http://chess.eecs.berkeley.edu/pubs/679.html.

[3] Modeling CyberPhysical Systems By Patricia Derler, Member IEEE, Edward A. Lee, Fellow IEEE, and Alberto Sangiovanni Vincentelli, Fellow IEEE

[4] H. Abelson and G. J. Sussman, Structure and Interpretation of Computer Programs, 2nd ed. Cambridge, MA: MIT Press, 1996.

[5] S. Edwards, L. Lavagno, E. A. Lee, and A. Sangiovanni-Vincentelli, Design of embedded systems: Formal models, validation, and synthesis,[Proc. IEEE, vol. 85, no. 3, pp. 366390, Mar. 1997.

[6] J. Sztipanovits and G. Karsai, Model-integrated computing,[IEEE Computer, vol. 34, no. 4, pp. 110111, Apr. 1997.

[7] B. Selic, The pragmatics of model-driven development,[IEEE Softw., vol. 20, no. 5, pp. 1925, Sep./Oct. 2003.

[8] Karl Henrik Johansson, John Lygeros, Jun Zhang, and Shankar Sastry "Hybrid Automata:A Formal Paradigm for Heterogeneous Modeling"*IEEE International Symposium on Computer-Aided Control System Design Anchorage, Alaska, USA September 25-27, 2000*

[9] A. J. van der Schaft and J. M. Schumacher. Complementarity modeling of hybrid systems. IEEE Transactions on Automatic Control,43(4):483-490, April

[10] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. Automatica, 35(3), March 1999.

[11] L. P. Carloni, R. Passerone, A. Pinto, and A. Sangiovanni-Vincentelli, Languages and tools for hybrid systems design,[Found. Trends Electron. Design Autom., vol. 1, no. 1/2, pp. 1204, 2006.

[12] J. R. Burch, R. Passerone, and A. L. Sangiovanni-Vincentelli, Overcoming heterophobia: Modeling concurrency in heterogeneous systems,[in Proc. Int. Conf. Appl. Concurrency Syst. Design, 2001, p. 13.

[13] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, Hybrid Systems I, Lecture Notes in Computer Science 736, pages 209-229. Springer-Verlag, 1993

[14] R. Balluchi, M. Di Benedetto, C. Pinello, C. Rossi, and A. Sangiovanni- Vincentelli. Hybrid control for automotive engine management: The cut-off case. In T. A. Henzinger and S. Sastry, editors, Hybrid Systems: Computation and Control, volume 1386 of Lecture Notes in Computer Science. Springer-Verlag, 1998.

[15] E. A. Lee. (1999). Modeling concurrent real-time processes using discrete events. Ann. Softw. Eng. [Online]. 7, pp. 2545. Available: http://dx.doi.org/10.1023/A:1018998524196

[16] The Theory of Hybrid Automata,Thomas A. Henzingerz ,Electrical Engineering and Computer Sciences University of California at Berkeley

[17] E. A. Lee and H. Zheng,Operational semantics of hybrid systems,[in Hybrid Systems: Computation and Control (HSCC), vol. 3414, M. Morari and L. Thiele, Eds.Zurich, Switzerland: Springer-Verlag, 2005, pp. 2553. [Online].

[18] X. Liu, J. Liu, J. Eker, and E. A. Lee,Heterogeneous modeling and design of control systems,[in Software-Enabled Control: Information Technology for Dynamical Systems, T. Samad and G. Balas, Eds. New York: Wiley-IEEE Press, 2003.

[19] O. Maler, Z. Manna, and A. Pnueli, From timed to hybrid systems,[in Proc. Real-Time, Theory Practice REX Workshop, 1992, pp. 447484

[20] Control Engineering ENGT4300: Notes-2 Discrete-time State-Space Realizations by Prof. Victor Sreeram

[21] E. A. Lee. (1999). Modeling concurrent real-time processes using discrete events. Ann. Softw. Eng. [Online]. 7, pp. 2545. Available: http://dx.doi.org/10.1023/A:1018998524196

[22] E. A. Lee, Cyber physical systems: Design challenges,[in Proc. Int. Symp.Object/Component/Service-Oriented Real-Time Distrib. Comput., Orlando, FL, 2008,pp. 363369. [Online]. Available:http://dx.doi.org/10.1109/ISORC.2008.25.

[23] E. A. Lee, Disciplined heterogeneous modeling,[in Proc. Model Driven Eng. Lang. Syst., D. C. Petriu, N. Rouquette, and O. Haugen, Eds., 2010, pp. 273287.[Online]. Available: http://chess.eecs.berkeley.edu/pubs/679.html.

[24] E. A. Lee and H. Zheng,Operational semantics of hybrid systems,[in Hybrid Systems: Computation and Control (HSCC), vol. 3414, M. Morari and L. Thiele, Eds.Zurich, Switzerland: Springer-Verlag, 2005, pp. 2553. [Online].

[25] X. Liu, J. Liu, J. Eker, and E. A. Lee,Heterogeneous modeling and design of control systems,[in Software-Enabled Control: Information Technology for Dynamical Systems, T. Samad and G. Balas, Eds. New York: Wiley-IEEE Press, 2003.

[26] O. Maler, Z. Manna, and A. Pnueli, From timed to hybrid systems,[in Proc. Real-Time, Theory Practice REX Workshop, 1992, pp. 447484

[27] Control Engineering ENGT4300: Notes-2 Discrete-time State-Space Realizations by Prof. Victor Sreeram

[28] Bucci.G, Fedeli, A.Sassoli, L.Vicario, Timed state space analysis of real-time preemptive systems Software Engineering, IEEE Transactions on Volume:30 , Issue: 2