A Hybrid Systems Approach to Communication Networks: Zeno Behavior and Guaranteed Simulations

Shams Karimkhan
Electrical Engineering
Wright State University
shamshmk@yahoo.com

Graduate Mentor: **Alessandro Abate**
Research Supervisor: **Dr. Jonathan Sprinkle**
Faculty Mentor: **Prof. S. Shankar Sastry**

July 29, 2005

Summer Undergraduate Program in
Engineering at Berkeley (SUPERB) 2005

Department of Electrical Engineering and Computer Sciences
College of Engineering
University of California, Berkeley

A Hybrid Systems Approach to Communication Networks: Zeno Behavior and Guaranteed Simulations

Shams Karimkhan

Abstract

In nature objects are time-dependent and with time they change in shape, position and composition. System engineering describes these phenomena with mathematical models. In this project we will use MATLAB to model and simulate Hybrid Systems (HS). A HS models a discrete program with an analog environment. The analog part is described by an ordinary differential equation (ODE). In order to know how the system works we need to find the solution to the ODE. Due to the fact that it is almost impossible to find the exact solution we need to use numerical approximation. By doing this we introduce some uncertainty, or rather some errors. The next step is controlling the errors, i.e. refining the simulations, in order to make sure that the simulated solution will have the same "shape" (behavior, evolution) of the actual simulation. This idea will be possibly embedded in the currently available simulation tools for HS. A TCP will be modeled as a HS and the Zeno behavior will be studied as well as error handling.

1 INTRODUCTION

Hybrid dynamics provide a convenient framework for modeling systems in a wide range of engineering applications such as mechanical systems, electrical circuits and embedded computation systems [3]. In order to understand and analyze a Hybrid System first we define a Hybrid System and its properties (sec.2.1). A Hybrid system has both continuous and discrete dynamics. The continuous dynamics are described by ODEs and the discrete dynamics by deference equations. Since it's almost impossible to solve an ODE we must use numerical approximations. Furthermore we will explain what a Zeno execution is. (sec.2.2). Zeno in HS is a pathological behavior in which there are infinite number of discrete transitions in a finite time. As for modeling we will model a TCP (Transmission Control Protocol) as a hybrid system and analyze it in a Zeno execution (sec.3).The model used is a network with two users, a wired link and a wireless link. In the end we introduce error handling in order to guarantee simulations, i.e. make sure the approximated solution will have the same behavior as the actual solution (sec.4.2).

2 DEFINITION

A dynamical system describes the *evolution* of a *state* over *time*. Based on the type of the state, a dynamical system can be classified into:

1. **Continuous**, if the state takes values in Euclidean space R^n for some $n \geq 1$.

2. **Discrete**, if the state takes values in a countable or finite set $\{q_1, q_2, ...\}$.

3. **Hybrid**, if part of the state takes values in R^n while another part takes values in a finite set.

We can also classify the dynamical systems based on the set of time over which the state evolves.

1. **Continuous time**, if the set of times is a subset of the real line. The evolution of the state of a continuous time system is described by an **Ordinary Differential Equation**(ODE).

2. **Discrete time**, if the set of times is a subset of the integers. The evolution of the state of a discrete time system is described by a **difference equation**.

3. **Hybrid time**, when the evolution is over continuous time but there are also discrete "instances" which we call *events*.

Continuous state systems can be further classified according to the equations used to describe the evolution of their state

1. **Linear**, if the evolution is governed by a linear differential equation or deference equation for continuous time or discrete time respectively.

2. **Nonlinear**, if the evolution is governed by a nonlinear differential equation for continuous time or difference equation for discrete time.

2.1 Hybrid Systems

Roughly speaking, Hybrid Systems are dynamical systems that involve the interaction of different types of dynamics, i.e. inter action of continuous state dynamics and discrete state dynamics. In other words a Hybrid System is a dynamical system that describes the evolution in time of the values of a set of discrete and continuous state variables: [3]. In this paper we define a hybrid system as a 6-tuple

$$H = \{Q, E, D, G, R, X\}$$

where

- $Q = \{q_1, q_2, ..., q_m\} \subseteq Z$ is a set of *discrete states* which is a subset of the integers.
- $E \subseteq Q \times Q$ is a set of *edges* which define relations between the domains. For $e = (i; j) \in E$ denote the source of e by $s(e) = i$ and the target of e by $t(e) = j$; sometimes the edges in E will be indexed, i.e., we will label the edges so that $E = \{e_1, ..., e_{|E|}\}$ where $|E|$ is the cardinality of E.
- $D = \{D_i\}_{i\in Q}$ is a set of *domains* where D_i is a subset of R^n.
- $G = \{G_e\}_{e\in E}$ is a set of *guards* where $G_e \subseteq D_{s(e)}$.
- $R = \{R_e\}_{e\in E}$ is a set of *reset maps* or transition maps; these are continuous maps from $G_e \subseteq D_{s(e)}$ to $R_e \subseteq D_{t(e)}$
- $X = \{x_i\}_{i\in Q}$ is a set of *vector fields* or ordinary differential equations (ODEs), such that x_i is Lipschitz on R^n. The solution to the ODE x_i with initial condition $x_0 \in D_i$ is denoted by $x_i(t)$ where $x_i(t_0)$ = x_0; we assume this solution is defined for all time [1].

2.2 Zeno behavior

Zeno of Elea was a Greek philosopher who developed different paradoxes about motion. One of the most famous paradoxes he came up with is called **"the dichotomy paradox"** and the idea is:
Suppose an object is moving from point A to point B, first it must traverse half the distance. Before it can do that, it must traverse a half of the half, and so on... It must, therefore, pass through an infinite number of points, and that is impossible in a finite time.
This is how we introduce Zeno executions in Hybrid Systems. We define a Zeno behavior in a hybrid system when we encounter infinite discrete transitions (events) within a finite time. Zeno executions can be classified into different types:

2.2.1 Chattering Zeno

Chattering Zeno results from having a switching surface in which the vector fields on each side "oppose" each other. As a result the trajectory will switch from one domain to the other and appear to be sliding along the switching surface. As we progress in time the time stamps go to zero, i.e. the time spent in each domain will decrease and approach zero.

2.2.2 Genuinely Zeno

In Genuinely Zeno the time stamps are always greater than zero. They are more complicated to detect and analyze.

2.3 Error Handling

As mentioned before in order to know how a hybrid system works we need to find the solution to the ordinary differential equation. Due to the fact that it is almost impossible to find the exact solution to an ODE, we need to use numerical approximation which results in having an uncertainty or error. Visually the error can be shown as circles (in two dimensions) or spheres (in three dimensions) around the approximated point where the actual solution can be anywhere within this circle or sphere. By propagating the error in time instead of having a circle or sphere we will have cones around the approximated value which show the boundary of errors. The cones will increase linearly when we calculate the local error and exponentially when it's calculated globally. If we are able to control these errors so when we encounter an event the bounding points of the cones are on the same guard we can then guarantee simulation.

3 MODELING

3.1 TCP(Transmission Control Protocol) Congestion control

TCP has been widely successful in wired internet. TCP Reno is the most popular TCP version today and in it's congestion avoidance stage, increases its windows size by one if there is no packets lost in the previous round trip time and halves the windows size if there is packet loss. This explains the AIMD topology; Additive Increase Multiplicative Decrease. The key assumption in most TCPs is that packet loss is a sign of congestion. This assumption breaks down when we introduce wireless networks which have an additional physical channel error associated with them. As a result we could have packets being dropped even when the link is not congested but the windows size will still halve and therefore we have underutilization of the wireless bandwidth [2]. In this paper a two user/two link network was modeled where the first link is wired and shared between the two users and the second link is wireless and only used by the first user. The model is shown in Fig. 3.1.

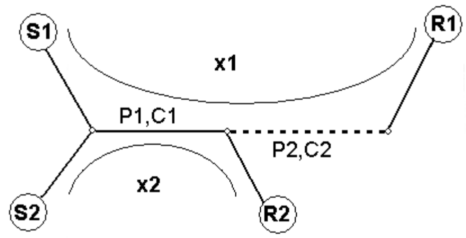

Figure 1. A model with two users and two links

3.2 Mathematical model

The network can be mathematically modeled Fig. 3.2 where the guards are the link capacities which are illustrated by the lines. The four different domains are:

1. $x_1 + x_2 < C_1$ and $x_1 < C_2$ where neither of the links are congested.
2. $x_1 + x_2 < C_1$ and $x_1 > C_2$ where only the first link is congested.
3. $x_1 + x_2 > C_1$ and $x_1 < C_2$ where only the second link is congested.
4. $x_1 + x_2 > C_1$ and $x_> < C_2$ where both links are congested.

3.3 Hybrid model

For modeling the network as a Hybrid System we approximated the price function1 as a step function instead of a sigmoid. By doing so we now have a Hybrid System that has a continuous value zero before congestion occurs and has a discrete transition to one when we have congestion. The hybrid model of the network is shown in Fig. 3.3. The state variables of the model are:

$$\begin{aligned}
\dot{x}_1(t) &= k_1 w_1^0 - k_1 x_1(t)[P_1(x_1(t) + x_2(t)) + P_2(x_1(t))]; \\
\dot{x}_2(t) &= k_2 w_2^0 - k_2 x_2(t)[P_1(x_2(t))].
\end{aligned}$$

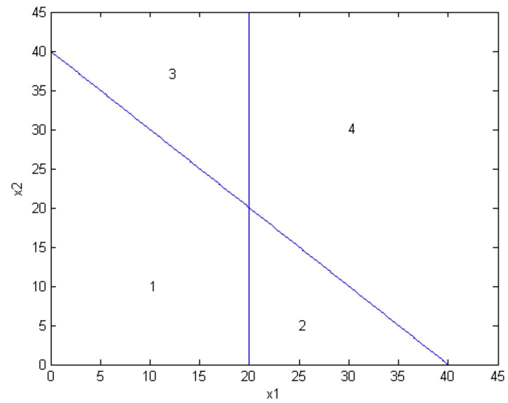

Figure 2. Mathematical model of the two user/two link network

Where w_r^0 is a weight and can be understood as pay per unit time. The source parameter w_r^0 is proportional to the number of connections opened by user r and changing the number of connections is equivalent to adjusting w_r proportionally.

P is the price function

$$P_j(x) = \frac{(x - C_j)^+}{x} \tag{1}$$

Where C_j is the capacity of link j and x is the aggregated rate passing through link j. The price function is strictly positive, increasing and continuous.
The wireless price is somewhat different than the wired price function due to the fact that a wireless link has the additional physical channel error which results in packet loss.

$$v_j(t) = q_j(\sum x_s(t)) = P_j(\sum x_s(t)) + (1 - P_j(\sum x_s(t))\epsilon_j \tag{2}$$

Where v_j is the noisy price function and $\epsilon_j \geq 0$ is the link packet loss rate due to physical error.

When the wireless link is not congested $q(.)$ is ϵ since the packet Loss is caused only by physical channel error. When the link is congested and packets are dropped at the router the $q(.)$ gradually increases. When this happens, i.e. the wireless link is congested we reduce the number of the user's connections with the network, which means we adjust w to a lower value.

4 SIMULATION RESULTS

With the use of ODE45 solvers in MATLAB both Chattering Zeno and Genuinely Zeno were simulated. The ODE45 function uses the Runge-Kutta numerical approximation method to solve ordinary differential equations. The Zeno behavior can be proved mathematically.

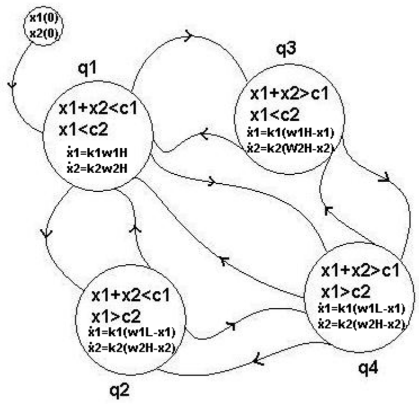

Figure 3. Hybrid model of the two user/two link network

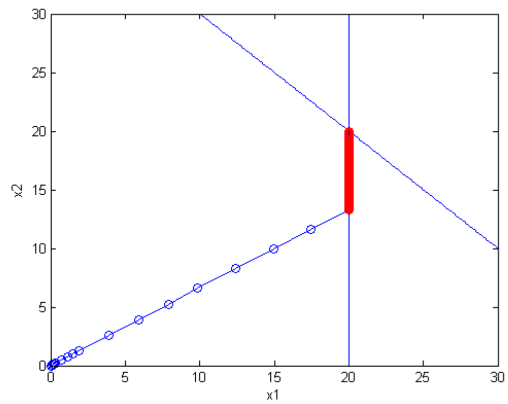

Figure 4. Chattering Zeno

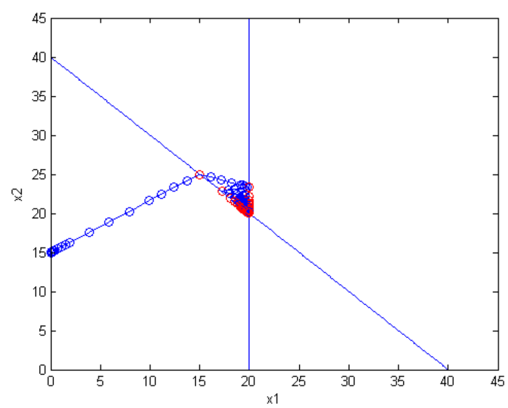

Figure 5. Genuinely Zeno

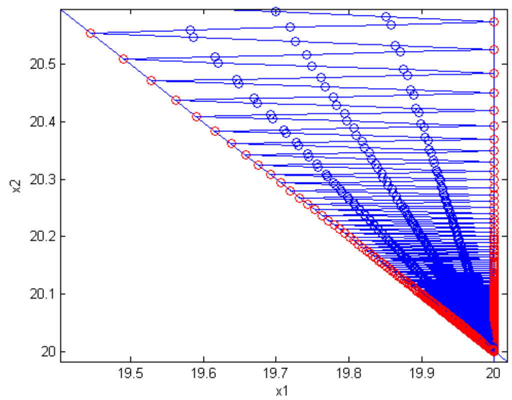

Figure 6. A closer look at the Genuinely Zeno

4.1 Zeno execution

4.2 Error Handling

After simulating the Zeno behavior we were interested in the amount of error introduced in the approximation. The error depends on the step size of the integration and also on the error tolerance. Error tolerance was set by RelTol (relative Tolerance) and AbsTol (Absolute Tolerance) parameters of the ODE45. The Error is shown in Fig. 4.2. This is only the propagation of the local error which is linear. It can be seen that as time goes by the error increases and the distance between the higher bound and lower bound becomes greater until one of the bounds will hit a different guard than the other. At this point it's desirable to control the error and reduce it so both the limiting boundaries can hit the same guard, i.e. guaranteed simulation.

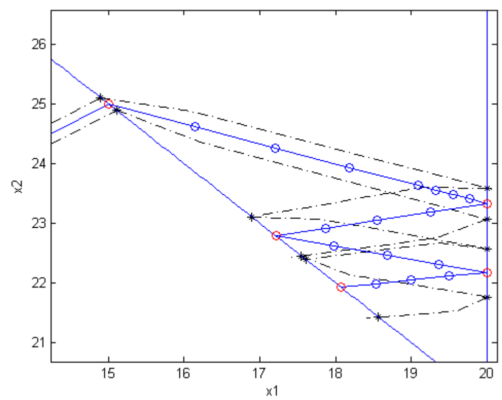

Figure 7. Propagation of local error

5 CONCLUSION

A TCP congestion control network was modeled as a hybrid system and the Zeno execution of this model was analyzed. The implemented error handling is the local error and not the global error. We can detect A Zeno behavior and try to avoid Zeno execution. Zeno execution is pathological and if a system is Zeno it may crash.

5.1 Future work

Implementing more complex TCP topologies(more users/link) and dynamics(non linear vector fields). This model could possibly be implemented in a larger scale, for example the internet.

References

[1] A. Abate, A. Ames, and S. Sastry. Characterizing deterministic hybrid systems through stochastic approximations. Technical report, University of California, Berkeley; EECS Department, 2004.

[2] M. Chen, A. Abate, A. Zakhor, and S. Sastry. Stability and delay consideration for flow control over wireless networks. Technical report, University of California, Berkeley; EECS Department, 2005.

[3] T. J. Koo, I. M. Mitchell, S. N. Simic, and S. S. Sastry. *Hybrid Systems - EECS291e Class Reader*. University of California, Berkeley, January 13 2003.