Introducing Digital Design Methods to System Dynamics Education

Tzur Levin and Ilya Levin

Tel Aviv University, Schools of Education

Ramat Aviv, Israel

Tel: (972)-3-6407109, Fax: (972)-3-6407752

Email: levintzu@post.tau.ac.il

Abstract

One of the origins of system dynamics is control theory. Thus, it is worthwhile to investigate how the growing use of digital models as means of control should affect the field of system dynamics. In particular, it would be interesting to explore how the introduction of digital models could enrich the learning of system dynamics in K-12 curriculum.

This paper proposes to do so through hybrid systems modeling. Hybrid systems are dynamic systems in which a digital element controls a continuous process. The paper illustrates the relevance of hybrid systems to K-12 education, and describes how hybrid systems may be modeled by using icon-based software such as STELLA. Two examples showing an implementation that is suitable for high school students are presented as well as an outline for future research.

Keywords: hybrid systems, system dynamics, education, digital control, finite-state-machine.

1. Introduction

One of the origins of system dynamics is control theory (Andersen & Richardson, 1980). Thus, it is worthwhile to investigate how the growing use of digital models as means of control should influence the field of system dynamics. In particular, it would be interesting to explore how the introduction of digital models could enrich the learning of system dynamics in K-12 curriculum. This paper proposes to do so by modeling a special group of system called *hybrid systems*.

2. Types of Control in System Dynamics

Classical system theory uses calculus as the major mathematical tool for describing systems (Betralanffy, 1968). In particular, the system dynamics methodology makes use of differential and difference equations to model systems and run computer simulations (Forrester, 1968). These models are graphically expressed as stocks, flows, connectors and converters in icon-based tools for system modeling such as STELLA (High Performance Systems, 1985-2001). Applying these four types of icons for modeling all system forces students to adopt a certain way of thinking (Penner, 2001). This way of thinking may be called an *analog* way of conceiving systems.

a) Controlling with Feedback Loops

The fundamental assumption of System Dynamics is that dynamic behavior of a system is mainly a consequence of feedback loops (Goodman, 1974). Feedback loops connect elements of the system in a way that information on the system in a given time determines its future behavior (Forrester, 1968). Thus, they control the dynamic behavior of systems.

Feedbacks loops tend to be of a similar nature to the process they control: both obey the dynamics of stocks and flows. Thus changes in the size of the lynx population control the dynamics of the hares population in the classical ecological example. The two processes are represented as stocks with inflows and outflows, and may therefore be referred to as analog. A notable exception to the analog view of systems is the IF-THEN-ELSE logical expression sometimes used in feedback loops. Though this logical condition is of digital nature, we shall soon see that it is too simple to represent the controlling power of digital models.

b) Digital Control Systems

Following the computer revolution, digital methods are now widely applied in developing controlling systems (Dorf & Bishop, 2001). The use of the digital methods increases the ability of control systems to handle complexity, and is popular in computer-embedded-systems, and in particular in computer controlled robots (Brooks 1991; Mioduser, 1995).

A classical example of the power of the digital approach is the digital computer itself (Figure 1). In the central processing unit (CPU) of the digital computer, a digital control unit is physically distinct from an operation unit (Mano & Dime, 1997). The controlled unit (data path) performs the arithmetic, logical and other data processing tasks. The control unit receives input from the controlled unit, and provides signals that activate various micro-commands to be performed by the operation unit. This kind of separation between the units may exist also when the controlled part of the system is of an analog nature.

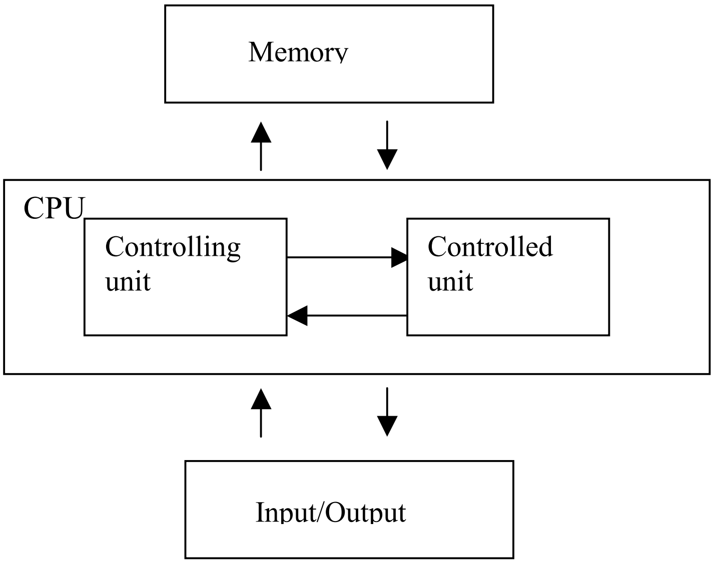

Figure 1 Digital computer structure (Mano & Dime, 1997)

c) The Hybrid Systems Approach

Special attention is given in recent years to systems where a digital unit controls a continuous process. Such systems are the subject of a new branch of engineering called 'Hybrid Systems Theory' (Johansson, 2000; Kamil & Chui, 1996; Branicky, 1995). By *Hybrid systems* we mean dynamic systems that are a combination of analog systems and digital systems. The architecture of a hybrid system forms a two levels' control structure (Figure 2). On the low level, feedback loops are used for local control as part of the analog system. On the high level, meta-control logic functions switch between different modes of the analog system behavior.

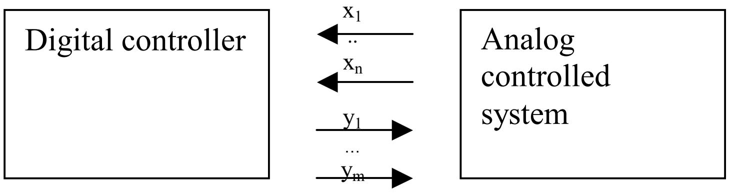

Figure 2 Hybrid system model

The analog subsystem is modeled by differential and/or difference equations. The digital controller is usually described in the form of a Finite State Machine (FSM), which reacts to events occurring in the controlled system. (Bencze & Frnanklin, 1994; Branicky, 1995; Mosterman & Biswas, 2000). The Finite State Machine receives input from the control system, and computes the output according to the state of the system, and its state transition rules. The model of the Finite State Machine will be described below. The general properties of hybrid systems are summarized in Table 1.

Table 1. Properties of the Hybrid System Components

	Digital	*Analog*
Mathematical model	Finite state machine	Differential/difference equations
Behavior	State transition according to input and transition rules	Change as a derivative/integral function
Graphical Model	State diagram	Stock-Flow diagram
Type of control	Event driven control	Local feedback loops

d) Relavance to k-12 Education

The inclusion of hybrid system modeling in k-12 system dynamics education has two advantages. First, it improves the understanding of specific systems and provides efficient means to solve control problems. Second, it provides a meaningful context for learning concepts from discrete math, computer science and digital design.

The systems suitable for hybrid modeling belong to one of two types: 1. Systems containing simple microprocessors such as climate control systems, elevator control panels, traffic controllers, drugs controllers etc. 2. System involving decision making, which effects a natural process, i.e. government decision concerning the market mechanism, hunting regulations to control prey-predator equilibrium, municipal council decisions to control urban development and so on.

While modeling such system, students get an opportunity to learn the principles of digital design. This knowledge is now limited to a small group of students, mainly from engineering and computer science classes. However, the influence of these concepts on the modern world should make them part of general education, in the framework of science and technology education for all. By applying the hybrid approach, students with different interests will find relevant contexts to learn the subject of digital control – be it a physical, medical, environmental or economical context.

3. Modeling FSM with System Dynamics Tools

Digital system may be as complex as the hardware of a digital computer, or simple as an On/Off switch. But in its core digital design consists of simple building blocks, which can be taught to high school students in relatively small number of lessons. This section describes what it takes to teach student to create simple digital controllers, modeled as Finite State Machines.

a) Fundementals of Digital Design

The root of digital design is in Boolean algebra, invented by the English mathematician George Boole in the 19^{th} century. Boolean Algebra enables a convenient notations for manipulation of logic statements, which is equivalent to truth tables (Katz, 1994). To allow designing digital machines, Boolean expressions are translated to logical gates. A logic gate is a device, which represents a Boolean function, and can be implemented in a variety of ways, the most useful of which are electrical circuits with switches and relays (Hillis, 1998).

A network of logical gates creates a logical device that executes logical functions. There are two basic types of logical devices (Mano & Dime, 1997): *Combinational logic devices* are devices that receive input values and calculate output values based on the arrangement of logic gates. *Sequential logic devices* are devices that calculate output according to the inputs and the history of the system. Sequential devices contain a memory, and can therefore execute time-varying function (Hillis, 1998). The most important model of the sequential device is the Finite State Machine (Katz, 1994). In the context of digital control, Finite State Machine is the main object of study for k-12 students and beyond.

b) Finite State Machine Representations

The Finite State Machine (FSM) as a fundamental model of finite digital systems is commonly used in computer science and digital control engineering (Hopcroft et al, 2001; Mano & Dime, 1997; Varshavsky & Pospelov, 1988). As a controller, the FSM receives input and calculates output according to the state of the systems. Didactic examples of FSMs are automatic vending machines, combination locks, traffic-light controllers and elevator control panel (Levin & Mioduser, 1996).

A graphical representation of the Finite-State Machine is the *state diagram*. The tabular form of the state diagram is that of *state table*. The state diagram and the state table describe the behavior of the FSM. The inner structure is modeled by the canonical representation, also referred to as “abstract synthesis” (Levin & Mioduser, 1996). This representation is useful in design and implementation of logical circuits. The structure of FSM is that of two interacting components: a combinational scheme and a memory register (Figure 3).

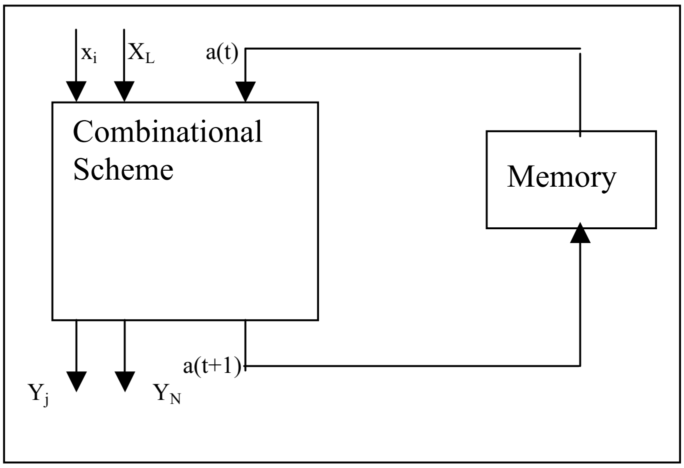

Figure 3. Structure scheme of the controlling FSM

c) Modeling in STELLA

For didactical reasons, the FSM should be modeled in the same software tool in which the controlled system is constructed. The following section shows how the FSM can be modeled and integrated with the controlled system using STELLA (High Performance Systems, 1985-2001). The representation of the FSM in STELLA is based on the canonical representation, which requires implementation of both the combinational scheme and the memory register.

Our proposal for implementing the combinational scheme in STELLA is by using a logical device called *multiplexer*. The multiplexer is a combinational logic device, which receives several input variables and selects one of them to be its output (Mano & Dime, 1997). For example, consider a case where the multiplexer receives two inputs from the controlled system, x_1 and x_2, and that the current state of the system is either a_1 or a_2. The multiplexer returns the new state of the system – a_1 *or a2* - according to the values of x_1, x_2 and the current state.

The multiplexer-based implementation of the combinational scheme of the controller can be represented in STELLA as tree-graph with converters as nodes. The nodes on the lowest level of the graph represent values of input variables. The graph executes a function of selection between the variables based on conditional statements (if-then). The output of the multiplexer affects a corresponding flow (Figure 4).

Memory is implemented in Stella by means of a stock with two flows. Each state has a numerical value (a_1=1 a_2=2 etc.), and the value of the current state is kept in the stock (Figure 6). With each "move" of the controlling systems, the value of stock nulls via the outflow, and a new value is added via the inflow, to be used as the value for the next calculation. Thus, the effect of delay, which is essential to the FSM, is achieved.

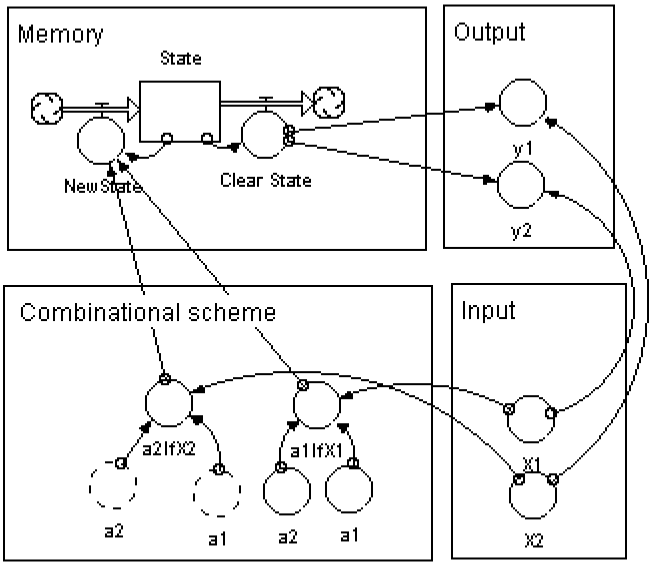

Figure 4. FSM implementation in STELLA

4. Examples for Use in Class

After the fundamentals have been taught, the class is now ready for hybrid modeling. In the system dynamics tradition, the context for the modeling should be a certain problem (Andersen& Richardson, 1980). Two examples are hereby presented.

a) Artificial Climate Control

A simulation of an artificial climate system is an example associated with both physics and technology education. The physical process is that of air cooling and heating. The technological aspect concerns the way an air-condition functions. To gain basic understanding of the subject we simplify both aspects of the systems. First, we model the physical process as change in temperature (a side effect of energy change). Second, we focus solely on the controlling element of the air-condition system, ignoring the way cold and hot air is produced. Further improvement of the accuracy and richness of the model may be achieved at a more advanced stage of the teaching.

The problem presented to the students is a typical control problem:

> *Create a digital thermostat that will regulate the temperature of a room according to the chosen value.*

The traditional way to model this system is by using a proportional-integral controller, which turns a radiator on and off. An alternative approach is based on a digital controller, represented as FSM. The controller will monitor the range of the room temperature, and will turn the radiator on and off according to the state of system. The system is described by its continuous and discrete subsystems, input and output vectors and a state vector (Figure 5):

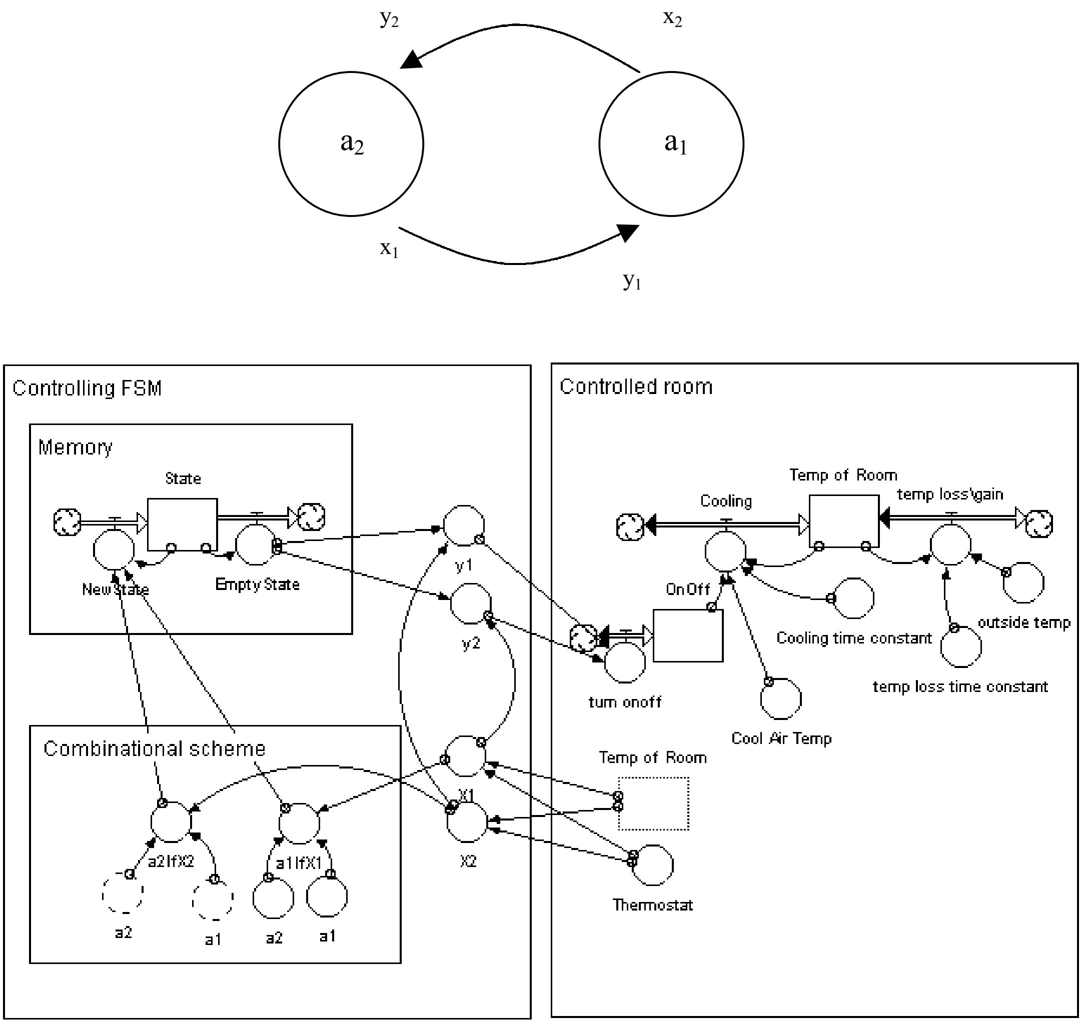

Figure 5. Controlling FSM and implementation in STELLA of a climate system

Input vector: x_1 = (Room Temp ≥ Thermostat+2) ; x_2 = (Room Temp ≤ Thermostat-2)

Output vector: y_1 means turn on the radiator; y_2 means turn off the radiator.

System states: a_1 means radiator on, while a_2 means radiator off.

The graph of the room temperature shows its regulated behavior over time (Figure 6). The simulation may be used for what-if scenarios, to evaluate the effects of the level of insulation, and to explore changes in the thermostat settings. More advanced models may describe the flow in the physical system in terms of energy changes and include more features of control such as humidity and smoke control.

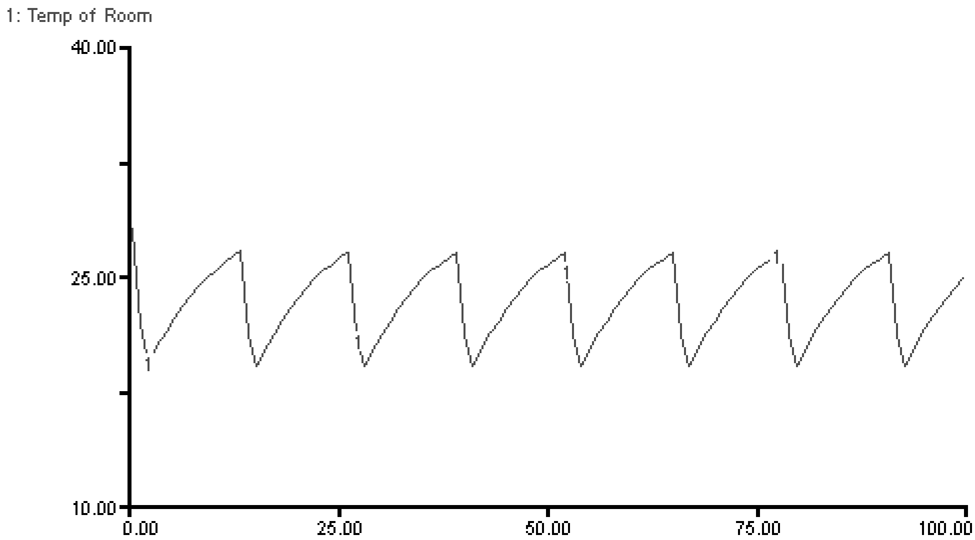

Figure 6. Graph of temperature

b) Drug Control

Medical treatment can be viewed as adopting control strategies to regulate the behavior of natural processes. Consider this example: a patient is to receive a dangerous drug. The drug enters the blood stream and then moves on to the stomach, where its effectiveness can be measured. A minimum concentration of the drug in the stomach is required to have therapeutic value. However a high concentration of the drug might be dangerous and even kill the patient.

Students are asked to model the process of drug absorption in the blood, and to design a digital control unit to regulate the consumption of the drug in real time. The controller has to monitor the amount of drug in the patient's body, and to determine whether to give the patient an additional dosage. The tricky point in this exercise is the issue of delay. Since the drug has to go first through the blood stream and only then reaches the stomach, it takes time until control decisions are felt in the body.

This problem may be given to students in biology, medical and paramedical classes, as well as control students in technology and engineering classes. The problem can be stated as follows:

> *Design an automatic controlling mechanism to treat a patient using a dangerous drug. The concentration of the drug in the stomach should only be values between the minimum effective and the toxic levels.*

Students are expected to design the physiological model first, and then to create the digital controlling element. Both the continuous and the discrete subsystems are to be implemented in Stella as shown in Figure 7.

Input vector:

x_1 = (Blood concentration ≤ Minimum therapeutic concentration+0.05)

x_2 = (Blood concentration ≥ toxic concentration-0.25)

Output vector: y_1 means set the dosage to 20 mg; y_2 means set the dosage to 5 mg.

System states: a_1 means 20 mg dosage is being taken; a_2 means 5 mg dosage is being taken.

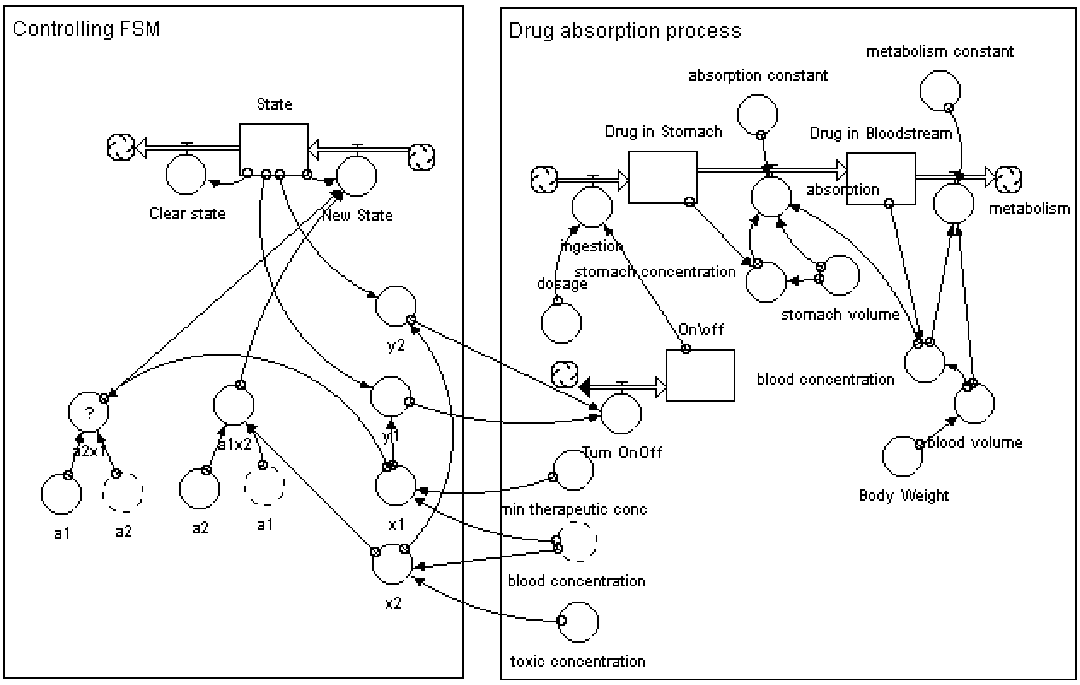

Figure 7. Drug control implementation in Stella

The graph in Figure 8 describes the drug concentration in the blood over time for the duration of the simulation.

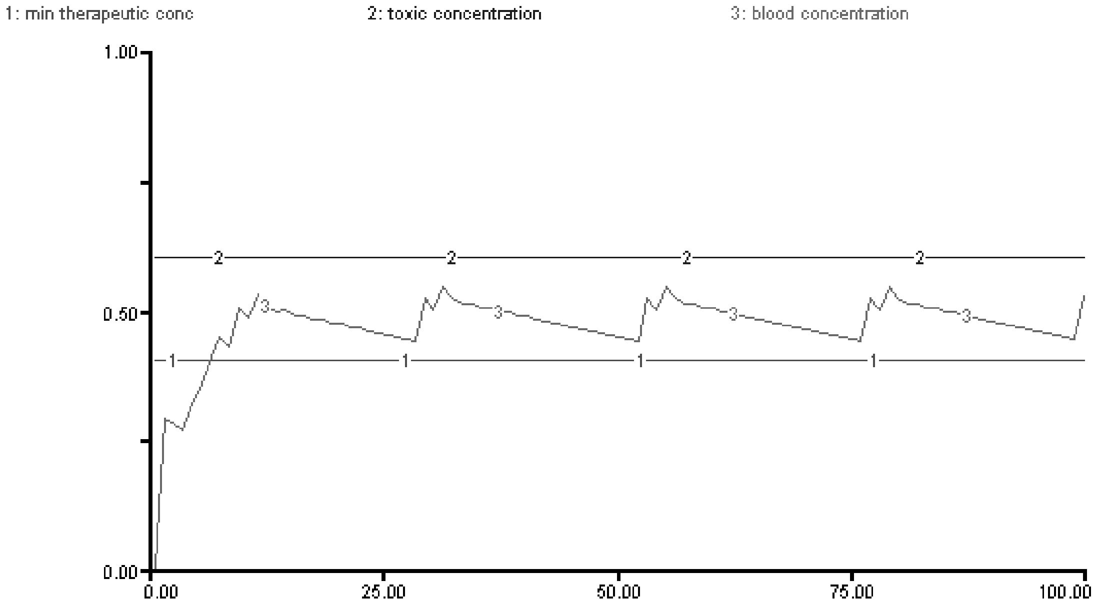

Figure 8. Concentration graph

5. Summary and Research Agenda

Despite the advance made in the theory of hybrid systems, efficient teaching oriented models of hybrid systems have not been developed. This paper is a first step towards filling this hiatus within the system dynamics approach to k-12 education. Future research on hybrid system modeling in education should follow two directions:

1. On the theoretical level, more examples need to be constructed and analyzed in diverse subject matters.
2. On the didactical level, empirical research on the pedagogical aspects of solving control problem in the hybrid approach is required.

Reference

1. Andersen David F.& Richardson George. 1980. Toward Pedagogy of System Dynamics. *TIMS Studies in the Management Sciences*, vol. 14, 91-106.
2. Bencze William J.& Franklin, Gene F. 1995 A Separation Principle for Hybrid Control System Design. *IEEE Control Systems Magazine*, vol.15, no.2, p. 80-4.
3. Branicky, Michael 1995. *Studies in Hybrid Systems: Modeling, Analysis, and Control*. Unpublished doctoral dissertation. MIT, Cambridge.
4. Dorf, Richard C. & Bishop Robert H. 2001. Modern Control Systems (ninth edition). New Jersey: Prentice-Hall.
5. Forrester, W. Jay. 1968. *Principles of Systems (second edition).* Cambridge,

Mass: Wright-Allen Press.

6. Forrester, W. Jay. 1994. *Learning through System Dynamics as Preparation to the 21st Century.* Keynote Address for System Thinking and Dynamic Modeling Conference for K-12 Education. Concord Academy. Concord, Ma, USA.

7. Hillis, W. Daniel. 1998. The pattern on the stone: the simple ideas that make computers work. New York: Basic Books.

8. Hopcroft J.E, Motwani R. & Ullman J.D. 2001. *Introduction to Automata Theory, Languages, and Computation. Second Edition*. New York: Addison-Wesley.

9. Johansson, Karl. 2000. *Hybrid Systems. Spring 2000 Lecture notes*. Berkeley. http://robotics.eecs.berkeley.edu/~johans/ee291e.html

10. Kamil, Ahmad & Chui Heung Wing. 1996. Hybrid Control Systems and Optimal Control. *Surprise*, vol. 4. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/ahak/report.html

11. Katz, Randy H. 1994. Contemporary logic design. Redwood City, CA : Benjamin Cummings.

12. Mano, M. Morris & Dime, Charles R. 1997. *Logic and Computer Design Fundamentals*. New Jersey: Prentice-Hall.

13. Mosterman Pieter J. & Biswas Gautam B. 2000. A comprehensive methodology for building hybrid models of physical systems. *Artificial Intelligence*, 121, 171–209.

14. Penner, David. 2001. Cognition, Computers, and Synthetic Science: Building Knowledge and Meaning Through Modeling. In Review of Research Education 25, Secada Walter G. (Editor). American Education Research Association: Washington; 1-35.

15. Varshavsky, V.I. & Pospelov, P.A. 1988. *Puppets without Strings*. Trans. By A. Dandarov. Moscow: Mir Publishers.