System Dynamics Learning through Separation of a Control Unit

Tzur Levin, Ilya Levin, Vadim Talis
Tel Aviv University, School of Education
Israel

Abstract

In computer systems a control unit is separated from an operation unit, and a channel of communication connects between them. We argue that this concept of design may have a pedagogical advantage in studying dynamic systems in general. We describe a constructive model for designing and analyzing dynamic systems, in which students design digital control units to system models in various subject matters.
We describe examples for using this model in robot design and in simulation of ecological system. We discuss ways to implement the model to control remote systems through the Web.

Keywords: System dynamics, technology, design, education, control, Internet

Introduction

With the digital revolution, a new era had begun in the field of control theory. Digital control methods were added to the classical continuous techniques for controlling autonomous systems and various computer-embedded-systems. This development has been recognized in the technology education context, where digital control has been long recognized as an essential component (Levin &Mioduser, 1996).

Digital control finds its way to the general technology classroom mainly through computer-controlled kits like the Lego-Logo learning environment, where behavior of robots is controlled by computer program implemented on a personal computer (Resnick & Ocko, 1991). While designing, debugging and testing the control unit, students understand the relations between the robot and its environment (Martin, 1994). The same constructive principle may be used for learning about systems in other contexts.

We suggest applying the digital control design to simulation of dynamic systems in diverse technological, natural and social systems. In this paper we explain how it can be done, and why it should be tried.

Control and Operation Units

The separation of the control unit from the operation unit is a common design approach in digital systems. Its classical manifestation is the digital computer itself (Figure 1). In the central processing unit (CPU) of the digital computer, the two units are physically distinct, each responsible for a different function (Mano, 1997). The operation unit (data path) performs the arithmetic, logical and other data processing tasks. The control unit receives input on the state of the operation, and provides signals that activate various micro-commands to be performed by the operation unit. This concept of design is also popular in computer-embedded-systems, and in particular in computer controlled robots.

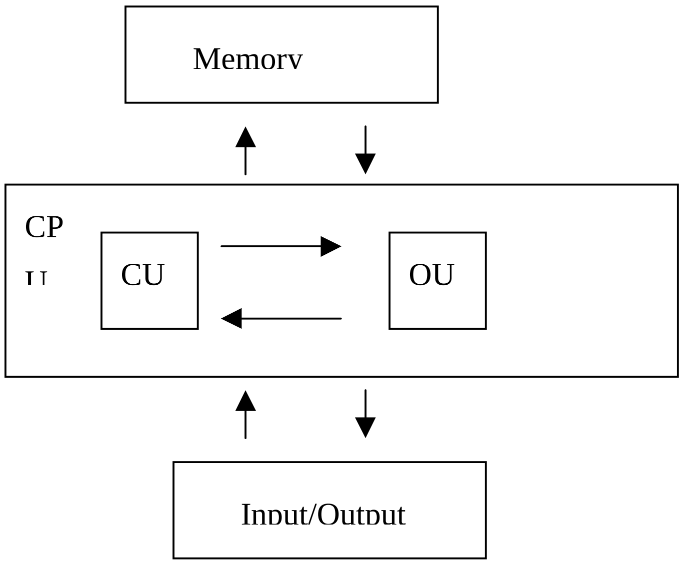

Figure 1: Digital computer structure

The Lego-Logo learning environment brings the same concept of design to a classroom (Resnick M. and Ocko S. 1991; Mioduser, Levin, Talis, 1995). The operating unit consists of the Lego building elements – bricks, wheels, and engines. The control unit is Logo-based control program implemented on a personal computer. A set $X=\{x_1,\ldots x_L\}$ of binary signals is transferred from the OU to the CU to communicate the world state of the system. A set of microoperations, represented by the binary signals $Y=\{y_1,\ldots,y_N\}$ is sent by the CU to the OU to control the OU's behavior (*Figure 2*). The goal of the CU is the generation of a sequence of signals Y, distributed in time. The functioning of the OU is dictated by this sequence.

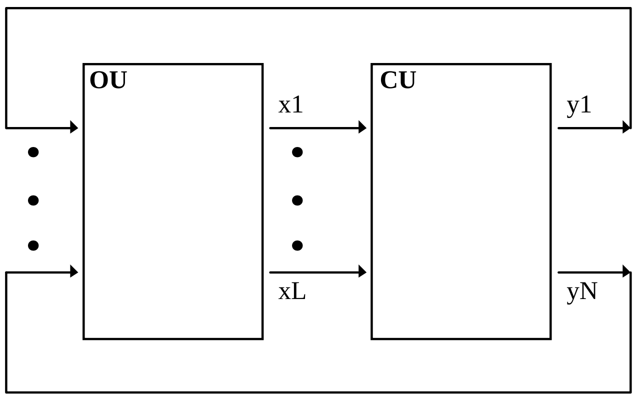

Figure 2: The connection between the control and operation units

We suggest implementing this design model to other cases of dynamic systems modeling. In order to do so, we extract control units from continuous modeling of given systems, and design them digitally.

Applying the model to other systems

Ever since its first manifestations in the 1940's, the system approach has been dominating diverse scientific and engineering disciplines, and has recently become key concept in the area of technology education (de Vries, 1996). With the development of computer based modeling and simulation software, construvistic tools for building virtual models have been developed to enhance learning (Maier and Grobler, 2000).

One of these tools is STELLA, software for dynamic system modeling and simulation, which is directed at K-12 and beyond. With STELLA students design, analyze, and explore the behavior of systems, and thus both learn about particular systems, and develops system-thinking skills (Forrester, 1994).

Jay Forrester, one of the pioneers of system dynamics theory and the 'father' of STELLA, believes that most systems can be described by a small set of "generic structures" (Forrester, 1989). We hypothesize that the CU-OU model belongs to this group of structures, and is especially suitable to cases where technological, natural and social subsystems are mixed. One of the goals of our research is to show that it can be constructively modeled in several subject matters, as demonstrated in examples presented in Table 1. Another is to evaluate the pedagogical advantages of the model.

Table 1. Control and operation units in various contexts

Subject matter	**Control Unit**	**Operation Unit**
Economics	Government and central bank fiscal policy	Market mechanism
Geography	City council decisions	Urban development
Ecology	Hunting regulations	Prey- predator equilibrium
Medicine	Medication prescription	Sub-systems of human body.
Mechantronics	Computer program	Mechanical elements
Transportation	Rules of traffic	Traffic flow
Applied ethics	Negative discrimination	Mobility of minorities in society.

Pedagogical Motivation

Forrester (1994) argues that dynamic systems simulation develops skills of clear thinking and communication besides panoramic inter-disciplinary outlook over science and technology. We believe that introducing the CU-OU model to system simulation in high school and beyond may further achieve the following goals:

1. *Overcoming complexity of systems.* In complex systems, the number of connections between elements becomes so big, that the model is filled with lines (the Spaghetti affect). The separation of the control unit – both physically and as a difference type of representation - is a way to simplify the model, without damaging it's fidelity.

2. *Introducing fundamental digital control principles and techniques*. Digital control is already playing and important role in computer embedded systems, and may be even more prominent in the near future intelligent machines. Not only engineers, but also users and the general public affected by technology should acquire it as part of their technological literacy.

3. *Widening students' conception of technology*. Similar control principles may be applied to both physical and social technologies. Thus students may develop a wide perspective on technology, from the mechanical through the biological to social engineering (Bunge, 1985).

4. *Emphasizing the importance of control components in different technological contexts.* Besides helping pupils to understand intelligent systems such as robots, it may provide an interesting perspective on environmental, economical and political problems, which are in many cases problems of control (Agassi, 1985).

Example: hunting regulations

The Prey-Predator model is a famous example of system dynamics simulation, which also happens to be one of the examples supplied by STELLA. The model (*Figure 3, right side the dashed line*) describes the mutual dependence of hares, lynx and their environment, and encourages students to investigate how equilibrium is reached in a natural environment (*Figure 4*).

The natural behavior of the system comes to an end once hunters begin to act (Figure 3, left side of dashed line). Not only is the total number of animals diminished, but also the equilibrium between the two groups is lost. The hunting effect is demonstrated on the graph describing the change of the populations over time (Figure 5).

Let us assume that forbidding all hunting is not a political option, since hunters demand their right to hunt. Our challenge is to determine hunting regulations so that both human needs are met, and the natural system survives satisfactory. These regulations should be adjusted to the changing behavior of the system over time.

We suggest to look at the problem as one of designing a digital controller to the system, and to use digital control techniques to solve it. The control unit will monitor data from the system, and send back commands for action based on a computation of the input. This controller is a finite state machine that can be designed with no prior programming knowledge, as a truth table, a flow chart or a states-diagram (Mioduser, 1995).

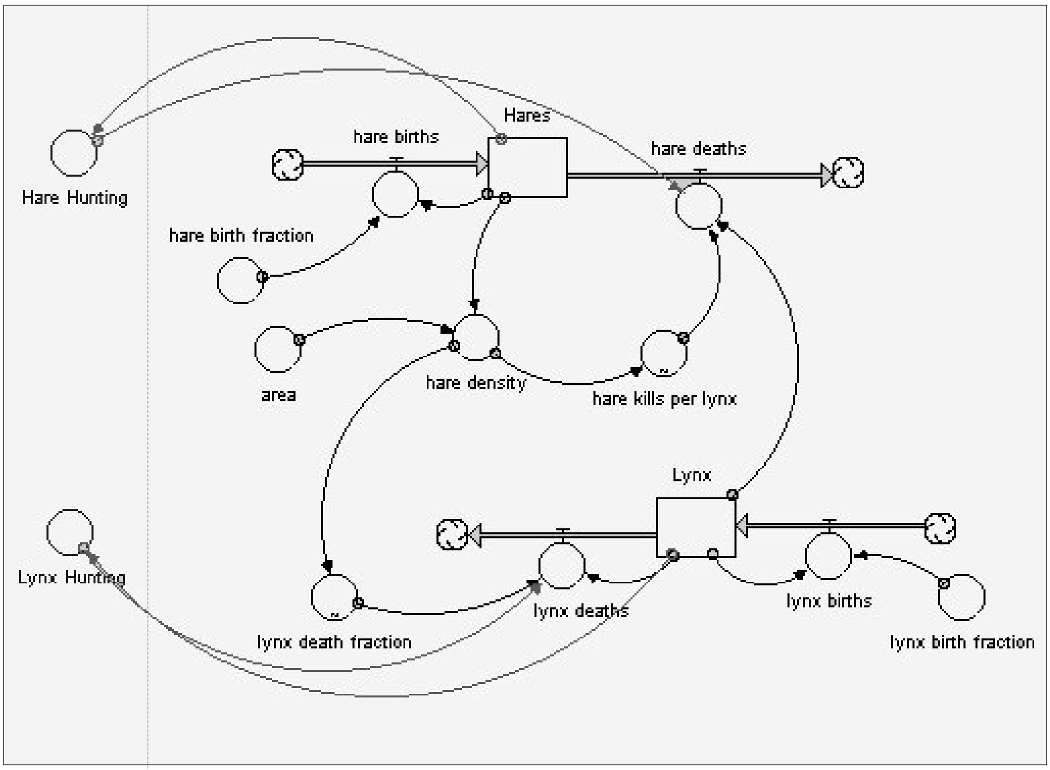

Figure 3: The Prey-Predator model

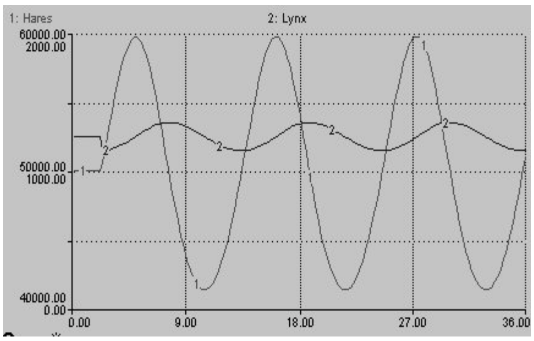

Figure 4:The Prey-Predator populations graph

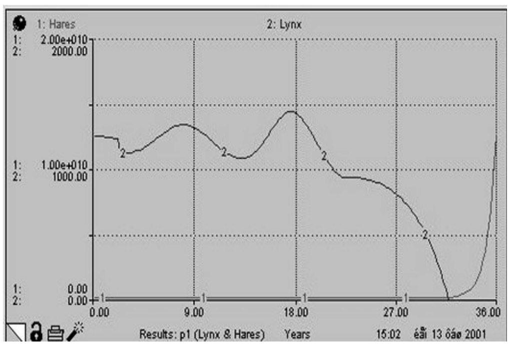

Figure 5: The Prey-Predator-Hunter populations graph

For example, we present a simple design of such a control unit, which regulates the hunting quota according to the animals' population size. The hares-hunting quota is 3,000 per year as long as the hare's population is beyond 50,000, and it drops to 1,000 when the population is below the number. The lynx-hunting quota is 15 per year as long as there are more than one thousand of them, and 2 otherwise (Table 2).

Table 2: The control unit

<table>
<tr><td colspan="6">Inputs</td></tr>
<tr><td>X1</td><td colspan="5">Hares population >50,000</td></tr>
<tr><td>X2</td><td colspan="5">Lynx Population >1,000</td></tr>
<tr><td colspan="6">Commands</td></tr>
<tr><td>Y1</td><td colspan="5">Set_Hunting_hares_Max_Qouta= 50,000</td></tr>
<tr><td>Y2</td><td colspan="5">Set_Hunting_hares_Max_Qouta2= 1,000</td></tr>
<tr><td>Y3</td><td colspan="5">Set_Hunting_lynx_Max_Qouta= 15</td></tr>
<tr><td>Y4</td><td colspan="5">Set_Hunting_lynx_Max_Qouta2= 2</td></tr>
<tr><td colspan="6">Controller table</td></tr>
<tr><td>X1</td><td>X2</td><td>Y1</td><td>Y2</td><td>Y3</td><td>Y4</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
</table>

The goal of controlling the system can be either specified as part of the problem, or be left to the students to decide. In both cases it may serve as a starting point for a discussion on hybrid ecological systems, in which the man and nature intertwines. While exploring the functioning of the designed controller, student may gain surprising insight concerning the system. In our example, they might discover that a constant quota of hunting – which does not take into account the changes in the population size over time – is the best policy. Can we generalize this rule to other cases of ecological problems?

Using the Web for Controlling Remote Systems

Being the universal channel of communication, the Web is an ideal environment for remote control. Therefore the model of separating a control unit from the rest of the system presents interesting opportunities. This potential has already been discovered by commercial manufactures of robots that offer the Web-based control to home robots. An example to the potential of Tele-robotics can be found in Irobot's site (http://www.irobot.com/ir/index.asp).

A web-based learning environment of this kind should be designed in client-server architecture. The simulation of the operation unit will be located on the server, and students from remote computer will design digital units to control its behavior. This architecture enables collaboration between students from remote geographical locations in the process of control design. While doing so, students will also have an opportunity to explore the concept of decentralized control in which many independent control units coexists, as opposed to a hierarchy with one control unit at the top (Wilensky & Stroup).

Summary

We proposed a new approach to system dynamics learning based on the CU-OU design concept. Through designing digital controllers to computer models of systems, students learn basic digital control methods, and develop system-thinking skills. Our future research will focus on the applicability of this model in specific technological contexts.

Bibliography

Bunge Mario (1985). Philosophy of Science and Technology. Dordrecht: Reidel Publishing.

De Vries, M. J. (1996), *Technology Education: Beyond the "Technology is applied Science" Paradigm*. Journal of Technology Education 8(1), 7-15

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.

Levin Ilya. *The State Machine Paradigm and the Spreadsheet Learning Environment*. In: A.J. Smith (ed.) Engineering Education, Increasing Students Participation Sheffield Hallam University, Sheffield, UK, pp.351-355, 1994.

Maier Frank H, Grobler Andreas (2000). What are we talking about? – Taxonomy of computer simulations to support learning. System Dynamics Review 16 (2) p. 135-148.

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

Martin, Fred G. (1994). Circuits to control: Learning engineering by designing LEGO robots. Ph.D. thesis, Massachusetts Institute of Technology.

Mioduser David, Levin Ilya, Talis Vadim. *Cognitive-Conceptual Models for Defining Robot Control.* Artificial Intelligence in Education, Association for Advancement of Computers in Education, Washington, USA, pp.131-134, 1995.

Paul H. Lewis. *Introducing Discrete-Event Control Concepts and State Transitions Methodology into Control Systems Curricula (*1994). IEEE Transactions on Education. Vol. 37 (1), pages 65-70.

Resnick M. & Ocko S. (1991). Lego-Logo: Learning through and about design. In I.Harel & S. Pappert (Eds.), Constructionism. Norwood, NJ: Ablex Publishing Corporation. http://www.ed.uiuc.edu/courses/edpsy490az-sp96/k12-technology/Resnick-Ocko/

Simon Herbert A. *The Science of the Artificial (Third edition). Cambridge: MIT Press.*

Wiener, Norbert (1965). Cybernetics. Cambridge: MIT Press.

Wilensky Uri, Stroup Walter (1999). Learning through Participatory Simulations: Network based Design for Systems Learning in Classrooms. Proceedings of CSCL ’99, Computer-Supported Collaborative Learning, Stanford University