Neural Network Heuristics for Controlling System Dynamics Models

Dong Hwan Kim
Research Center for Public Administration
Korea University
Seoul, Korea

Ik Jae Chung
Department of Public Administration
Seoul National Polytechnic University
Seoul, Korea

Abstract

Many System Dynamics researchers have found that decision-makers have difficulties in controlling System Dynamics models which represent complex social reality. This means that heuristics employed by decision-makers are not appropriate for controlling dynamic social problems. As alternative ways for understanding and controlling System Dynamics models, various mathematical mehtods have been suggested.

Some simulation-based experiments demonstrated the possiblity of decision-makers' learing ability. For instance, the experiment performed by Sterman showed that game players' performance was improved slowly as their experiences are accumulated. The slow learning process is often regarded as indicating the limitation of human intelligence. On the contrary, it may be interpreted as indicating a potential power of human intelligence or heuristics.

In previous studies, decision-makers' heuristics are formulated in simple decision rules. Such decision rules have failed to incorporate the learngin ability of decision-makers. To experiment the learing ability of decision-makers, this study replaces decision- maker with a neural network model. The neural networks are recognized as a representative of human intelligence by many students in artificial intelligence. In this study, neural network heuristics are applied to two System Dynamics models; Meadows's commodity cycle (1969) and Sterman's model of the Kondratiev cycle, or long wave (1985). Neural networks model have deomonstrated a surprising performance in learning and pattern recognition.

In addition to neural network applications, this study demonstrated technical feasibilities in IBM environments using Smalltalk.

Neural Network Heuristics for Controlling System Dynamics Models

Recently, System Dynamics researchers have found that human decision makers ha difficulties in controlling System Dynamics models which represent complex social reality. Th means that heuristics employed by decision-makers are not appropriate for controlling understanding dynamic social problems. As alternative ways for understanding and controllir System Dynamics models, various mathematical methods have been suggested (Coyle 1985; Macec 1989; Ozveren & Sterman 1989; Franco 1990).

Simulation-based experiments, however, demonstrated the possibility of decision-make learning ability. For instance, the experiment performed by Sterman showed that game playe performance was improved slowly as their experiences are acumulated. The slow learning process often regarded as indicating the limitation of human intelligence. On the contrary, it may interpreted as indicating a potential power of human intelligence or heuristics. The learning proce of decision-maker, even very slow, is worth studying.

In previous studies, decision makers' heuristics are formulated in simple decision rul (Sterman 1989; Morecroft 1988). Such decision rules have failed to incorporate the learning ability decision-makers. As Sterman pointed, "the primary difference between the model and t experimental results is the fact that many of the players began to learn how to control the system the game progressed (Sterman 1989)."

To experiment the learning ability of decision-makers, this study replaces decision-makers wi a neural network model. Neural networks are recognized as a representative of human intelligence many students in artificial intelligence. That is why some researchers call nerual networks as natu intelligence (Caudill & Butler 1990). Neural network models have shown a suprising performance learning and pattern recognition problems (Rumelhart & McClelland 1986; Burke & Ignizio 19S Salchenberger et al.1992).

In this study, neural network heuristics are applied to two System Dynamics models; Meadov commodity cycle (1970) and Sterman's model of the Kondratieff cycle, or long wave (Rasmussen al.1985). The main purpose of this study is to apply neural network heuristics for controlling Syste Dynamics models, and to discuss model behavior. Furthermore, this study was done within IB environments using Smalltalk. That is, this study demonstrated technical feasibilities which is r available in STELLA.

1. Policy making process and neural networks model

In this study, it is assumed that a task of a policy maker can be divided into two objects. First policy maker should maintain the value of target variables within a desirable level. Second, he shou eliminate the fluctuating behavior of the target variables.

Previous studies in S.D society have paid more attention on the second object. We want focus on the first object in this paper, because it provides a rather straight problem for poicy mak to solve and it seems to be more consistent with the real task to which most policy makers accustomed.

In general, a policy maker cannot control the target variable directly. For example, econon policy makers cannot change the amount of private investment. A policy maker must change value of a target variable indirectly; by increasing or decreasing a tax rate. From the viewpoint o policy maker, variables in S.D. model may be classified as signal variables, control variables a target variables.

Policy makers are expected to maintain a value of *a target variable* at a desired level. *control variable* is a place where a policy maker interrupt to achieve his policy goal. He decides value of a control variable by watching *signal variables* which provide him informations ab changing states of the policy context. The bounded rationality of a policy maker permits onl limited number of variables to be processed for determining policies (Simon 1978). In this study have used only a limited number of variables as signal variables.

Of course, a policy maker may be unable to decide the value of a control variable, because a control variable may be dependent on lots of another variables which are beyond his control. In theses situations, a policy maker cannot determine the value of a control variable, he can only change it toward his intention.

The process used for determining or affecting the value of a control variable by watching signal variables can be defined as a decision function (Forrester 1961). A traditional trial-and-error heuristics for reducing a gap between the current state and the desired state could be classified as two kinds.

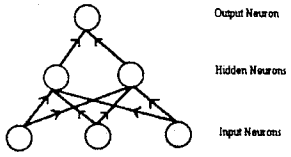
First, a gap between the current state and the dsired state can be reduced by changing the output value of a decision function with feedbacks about the error (Richardson 1991). This kind of trial-and-error method is usually imbeded within the decision function itself. Changing only the output of a decision function according to errors, a decision function itself is not modified and so learning in its strict sense cannot be occurred.

Second, the trial-and-error principles may be operated on the decision function itself. That is, the parameter or structure of a decision function is changed according to the previous error. This process means the learning ability of decision-makers. In fact, these kinds of trial-and-error are what neural network models are expected to perform. In particular, the trial-and-error method is used extensively by the backpropagation models which are one of the most frequently used neural network models. For this reason, we use a backpropagation network as our experimental policy makers.

2. Backpropagation Networks and Its Heuristics

A common backpropagation network is composed of input neurons, hidden neurons and output neurons. These three layers of neurons are hierachically related with each other by weighted connections which represent strengths of relations. Each neuron has its activation level. Each neuron's activation level is propagated toward other neurons in proportion to the weight attachted to each connections (Freeman & Skapura 1992).

Figure 1 shows a common backpropagation model composed of three input neurons, two hidden neurons and one output neuron. Activation levels of input neurons are determined by input patterns provided from the environment. In S.D. model, activation levels of input neurons are determined by signal variables. In other words, values of signal variables are inputs to the backpropagation network. Activation levels of input neurons are propagated to hidden neurons, and the activation levels of hidden neurons are propagated to an output neuron. In this study, the activation level of an output neuron means the amount by which a policy maker intends to change the value of a control variable.



< Figure 1> Backpropagation Model

Backpropagation network learns from the error of an output neuron. The output error is a difference between the activation level of an output neuron and the desired level of an output neuron. Learning in neural networks means the change of connection weights according to the error (Rumelhart, Hinton & Williams 1986; Stone 1986). At first, connection weights between hidden neurons and an output neuron are changed according to the output error. And the output error is back propagated to determine the error of hidden neurons. The error of hidden neurons are computed by considering the activation level of hidden neurons and connection weights between hidden neurons and the output neuron. After determining all errors of hidden neurons, connection weights between

hidden neurons and input neurons are changed. Repeating these process, a backpropagation networl learns complex and nonlinear input-output patterns.

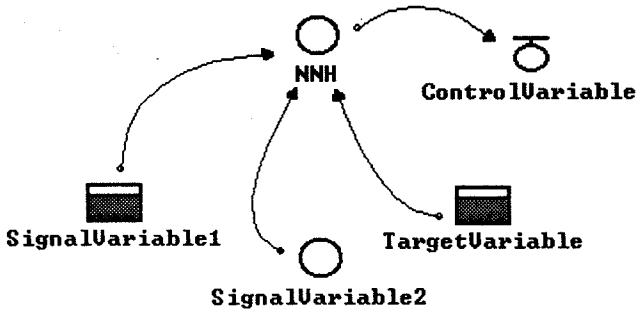
Unfortunately however, a backpropagation network cannot be directly used within a S.D model. The activation level of an output neuron represents the amount of a policy that a neura network as a policy maker exerts on a control variable. For taking place a learning process, a desire activation level of an output neuron which means the desired value of a control variable should b provided each time. But generally we do not know the desired value of a control variable in S.D model. Without the desired value of a control variable, we cannot determine the error of an outpu neuron.

Although we do not know the desired values of control variables, we know the desired level o a target variable. The desired value of a target variable is imposed on the policy maker as a polic goal. In this study, we have replaced the error of an output neuron with the error of a target variable This replacement is justified from our common senses that our decision is usually guided by the erro of consequences, not by the error of decision itself. This analogy adds another heuristic property t our nerual network so that our neural network policy maker may not even find local optimality.

3. Interface between neural networks and S.D. model: Object oriented approach

Combining a neural network with a S.D. model requires a rather complex interactions betwee them. On each simulation times, a neural network must receive values of signal variables, and control variable in S.D. model should receive the activaiton level of an output neuron of the neura network.

We have made an objected oriented simulation environment for system dynamics to solve thi problem. We have named it as '*Equations as Graphic Objects* (EGO)'. EGO is constructed withi Smalltalk language which runs on MS-Windows in IBM-PC. In EGO, all equations in S.D. mode are represented as objects. In fact EGO itself is an object. An object in Smalltlak means a independent entity which can receive or send messages from/to other objects (Goldberg & Robso 1989). In Smalltalk and EGO, all objects can interact with each other as peoples talk each other.



<Figure 2> Interface between neural network and S.D. model

Because all algorithms and models are reprsented as objects in Smalltalk, a neural network ca be regarded as an independent object and thus it can be inserted into S.D. model with ease. In o study, a neural netowrk is inserted into an auxiliary variable equations as demonstrated typically figure 2.

As displayed in Figure 2, Figure 3 and in Figure 5, EGO provides users various icons f representing different kinds of variables. Furthermore, in EGO, the icons for table functions a automatically designed to represent their time trends. In particular, users can change the color arrows to mark important causal loops. (EGO is a public software. But the source code of EGO w

be open to the public next year, because optimization and documentation about the source code of EGO is not completed.)

In figure 2, a neural network observes three variables for deciding the value or amount of its policy. Here a target variable also serves as a signal variable. In EGO, the equation for a neural network can be defined as follows.

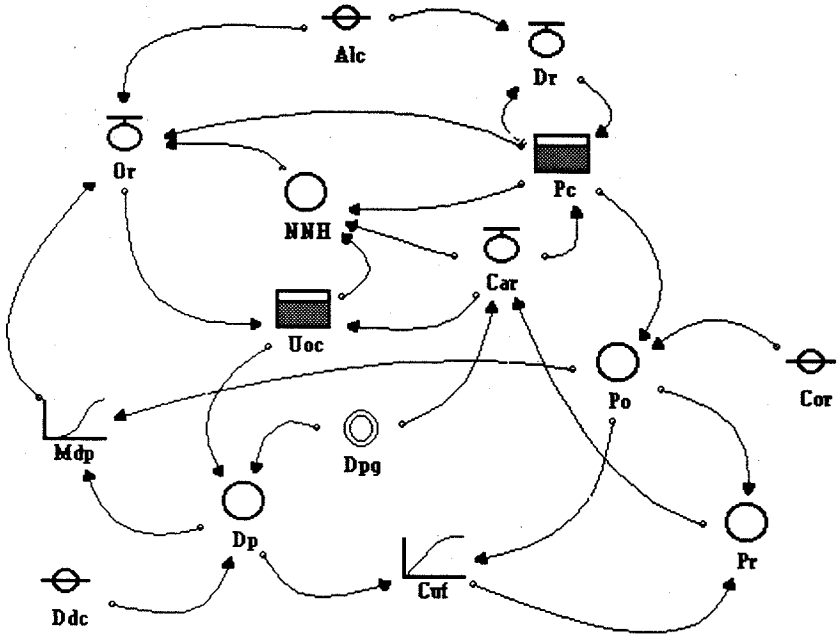
NNH = NeuralNetwork in: s1 in: s2 in: s3 goal: TargetVariable

'NNH' is an auxiliary variable which receive a value of an output neuron from neural network. NNH is a communication channel between a neural network and a control variable in S.D. model. 'NeuralNetwork' is an object which is an instance of neural network class. 'TargetVariable' is simply a name of a target variable. S1, S2, S3 is signal variables. If a target variable is also used as a signal variable, the name of a target variable will appear twice in above equation.

A neural network receives informations about current value of a target variable and determine an error of a previous policy; the error resulted from the previously intended change of a control variable. And connection weights are updated to reduce errors. Informations about signal variables are used to determine a new policy for a control variable. A new policy is represented by NNH, and NNH can be included in any equations which define control variables.

4. Neural network heuristics for Kondratieff Wave Model

Figure 3 shows a System Dynamics flow diagram for a simplified version of the Kondratieff model developed by Sterman (Rasmussen et al. 1985) into which our neural network heuristics (NNH) is inserted. Figure 3 is a flow diagram displayed in EGO. The Kondratieff model shows chaotic behaviors as well as limit cycles. In this paper we present our experiments with the version of chaotic Kondratieff model. NNH has shown identical performances with the case of limit cyle model of Kondratieff wave. To produce a chaotic behavior a small sinusoidal variation is introduced in 'Desired production of goods (Dpg)' as in Rasmussen et al.(1985).

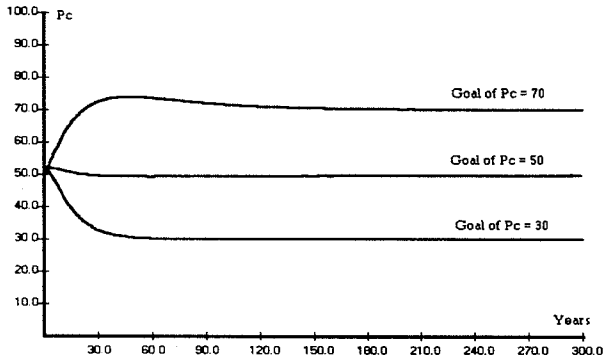


<Figure 3> Kondratieff Model with Neural Network Heuristics

In figure 3, NNH observes current states of an unfilled orders for capital (Uoc), a capit acquisition rate (Car) and a production capital (Pc) to make a policy affecting an order rate for capit (Or). The value of a policy determined by NNH is introduced to the original equation of Or. The go of NNH is to maintain the value of Pc at a specified level.

We have simulated the model by specifying the desired level of Pc at 30, 50 and 7 respectively. Given a specific goal, NNH has been trained for five or six times. Each traning continued until 300 simulation time. After training of NNH is completed, a Kondratieff model wi the control of NNH was simulated for each specified goals. Results of simulations are displayed i figure 4.

NNH has performed successfully in maintaining the target variable at a specified goal leve ..ll the more, NNH has successfully stabilized the inherent fluctuations of Kondratieff model. W were surprised at these results, because NNH has only incomplete informations about the model, ar in particular NNH has been completely ignorant of the structure of the Kondratieff model.
A direct implication of these results is that a human-like intelligent mechanism with limite informations can control a complex system as well as various complicated algorithms with comple informations.



<Figure 4> Performances of NNH in Controlling Kondratieff Model

From the performance of NNH in Kondratieff model, we have got a strong impression that neural network can be a promising tool for controllng a S.D. models. We think that neural netwo model is essentially a new controlling mechanism in system dynamics studies for several reasons.

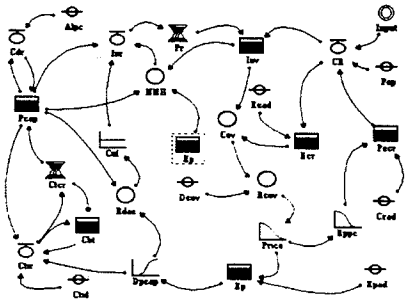
First, it is a self-learning control device. Unlike traditional algorithms for controlling syste dynamics model, a neural network operates with natural principles of intelligence; '*learning by doin* or '*learning by repeat*'. It can be regarded as an expert who has a long experience in controlling specific policy model. Moreover, if social networks of institutions can be modelled through tl metaphor of neural networks models, NNH may be used to find the optimal forms of institution (organizational) networks for resolving social problems.

Second, a neural network does work well without knowing details of system dynamics mod Many algorithms developed for controlling system dynamics model should know all equatio specified within a model. Neural networks don't require any knoweldge about equations. It requir only a period of times to learn about dynamic behavior of some important variables.

Third, a neural network can be tested and used by those who have little knowledge about mathematical control theory. A neural network is a general and easy-to-use control heuristics. Poli optimization in S.D. models may be performed by a broader class of people than before.

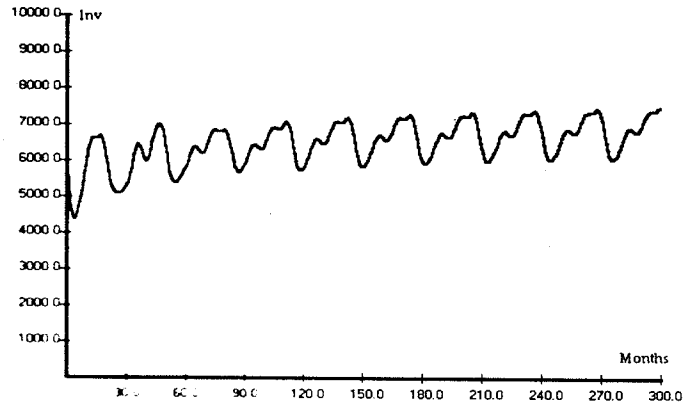
5. Neural Networks heuristics for Commodity Cycles

A famous commodity cycle model developed by Meadow (1970) is a second model in which we have experimented the behavior and performance of NNH. We have conjectured that commodity cycle model provides a more difficult task on NNH than Kondratieff model, because it has many time lags among variables. Time lags in realizing the effect of a policy may confuse NNH so that it may fail to learn from experience.



<Figure 5> Commodity cycle model and NNH

Fig 5 shows System Dynamics flow diagram for commodity cycles. As in Kondratieff wave model, a small sinusoidal variation is introduced into external variable of 'Input' to produce a cyclic behavior all the times as shown in Figure 6.

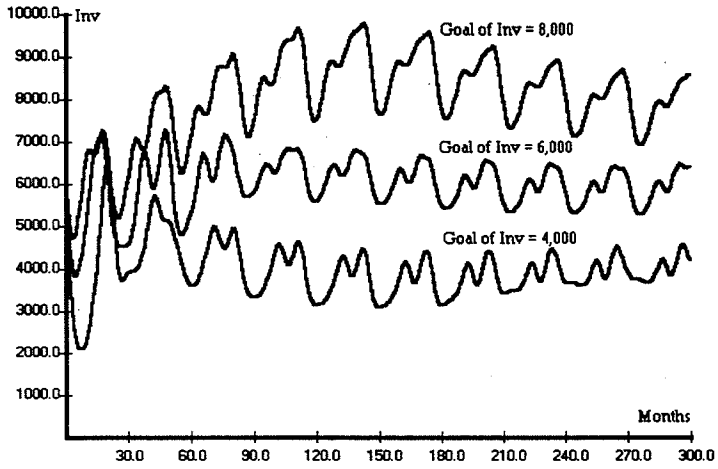


<Figure 6> Cyclic behavior of Inventory (Inv) without control of NNH

NNH receives informations about current values of an inventory of commodity (Inv), a production capacity (Pcap) and a price expected by producers (Ep). The goal of NNH is to maintain an amount of inventory (Inv) at a specified level as before. NNH controls the value of commodity initiation rate (Inr) for achieving a specified goal.

In pilot experiments NNH showed a poor performance. So we give a more flexible policy tool to NNH. That is, we simply multiply the value of NNH by 2 and subtracted by 1. This operation makes the range of values produced by NNH from -1 to +1. Whereas NNH in Kondratieff model has only a positive policy tool, NNH in commodity cycle model has a policy tool that can be positive or negative.

NNH has been trained like as in Kondratieff model. We have experimented with three goals c Inv; 4,000, 6,000 and 8,000. The results are displayed in figure 7.

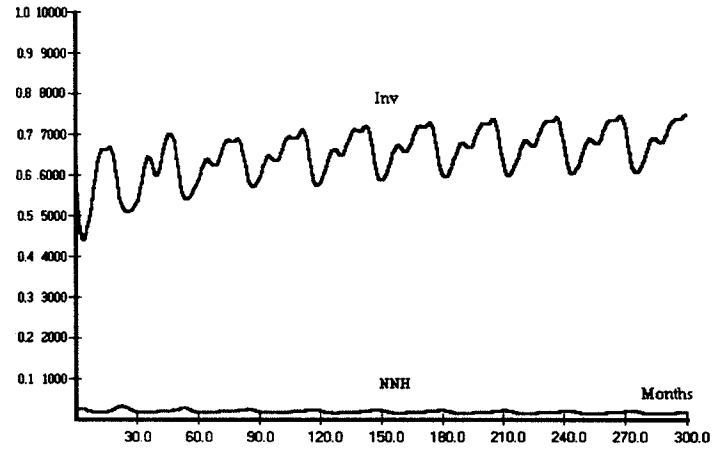


<Figure 7> Performances of NNH in Controlling Commodity Cycle Model

Figure 7 says that NNH has failed to control fluctuations but has succeeded in directing target variable toward a specified goal. In particular, for the goal of 4,000 Inv, NNH has reduced tl magnitude of a fluctuation by significant amounts compared with the fluctuation in figure 6. Th result shows some possiblity that the control ability of NNH can be enhanced with appropriate polic goals.

Although NNH has not shown satisfactory performance in controlling the fluctuations, NN has revealed a rather interesting behavior during our experiments. When a difficult goal was impose on NNH, NNH has shown *a helplessness behavior* like human decision makers.

Now we give NNH only a positive policy tool. When the goal is set to maintain Inv at 4000 ar the policy value of NNH is restricted as positive, it is almost impossible for NNH to achieve a goa Without the interruption of NNH, Inv of commodity cycle model shows 6000 amount at averag Since a policy value produced by NNH is alwayse positive, simply adding it to the control variab (Inr) means that the value of Inv shall increase higher than orginal model. As a result, NNH cann find the satisficing solutions. Figure 8 shows how NNH behaves in hard situations.



<Figure 8> Helplessness behavior of NNH

Before a training period a value of NNH were 0.54. In figure 8, we can see that values of NNH approach toward zero, with the range of 0.003 to 0.001. As a result, the behavior of Inv is identical with that of original model in figure 6. NNH has learned to avoid its responsibility.

An implication of NNH in commodity cycle model is that policy makers faced with unfavorible situation may show the helplessness behavior which means non-decision making. Helplessness is the psychological state that frequently results when events are uncontrollable (Seligman 1975). If a policy goal is set too high, a policy maker may learn to give up any efforts for achieving it.

6. Conclusions and Future Research

In this study we have experimented the behaivors and performances of neural network heuristics within S.D. models. Results of our experients have two major implications.

First, a neural network can be used for controlling S.D. model. We believe that more studies on combining neural networks and S.D. models can reveal a new way of controlling S.D. model. And this way of control may be more easy-to-use than any other control algorithms.

Second, our study shows that the stabilization of fluctuations may be occurred through implementing a policy which is oriented toward maintaining a value of target variables within a desired level.

Our experiments have only concerned with the backpropagation model of neural networks and with two simple S.D. models. Other models of neural networks may improve the performance of NNH. The controlling ability of NNH may be enhanced with the time series informations about important variables. Genetic algorithms may be another promising heuristics for controlling S.D. models (Goldberg 1989). Experiments with complex S.D.models may show different results with ours. Our experiments have produced more questions than answers. We think that studies on the natural intelligences or heuristics in system dynamics will provide great insights on handling complex social systems.

References

Burke L.I & J.P. Ignizio, 1992, "Neural Networks and Operations Research: An Overview *Computers & Operations Research*, Vol. 19, No. 3.

Caudill M. & C. Butler, 1990, *Naturally Intelligent Systems*, The MIT Press.

Coyle R.G., 1985, The use of optimization methods for policy design in a system dynamics mode *System Dynamics Review*, Vol. 1, Num 1.

Forrester J., 1961, *Industrial Dynamics*, Cambridge MIT Press.

Freeman J.A. & D.M. Skapura, 1992, *Neural Networks: Algorithms, Applications, ar Programming Techniques*, Addison Wesley.

Frnaco D., 1990, Policy Design In Oscillating Systems, *1990 conference of System Dynamics*.

Goldberg A. & David Robson, 1989, *Smalltalk-80: the language*, Addison-Wesley.

Goldberg D.E., 1989, *Genetic Algorithms in Search*, Optimization, and Machine Learnin Addison-Wesley.

Macedo J., 1989, A reference approach for policy optimization in system dynamics models, *Syste Dynamics Review*, Vol 5, Num. 2.

Meadows, D.L., 1970, *Dynamics of commodity production cycles*, Cambridge: MIT Press.

Morecroft J.D.W., 1988, System dynamics and mircoworlds for policymakers, *Euporean Journal Operational Research*, Vol. 35, pp. 301-320.

Ozveren C.M. and J.D. Sterman, 1989, Control theory heuristics for improving the behavior economic models, *System Dynamics Review*, Vol 5, Num. 2.

Rasmussen, S., E. Mosekilde and J.D. Sterman, 1985, Bifurcations and chaotic behavior in a simp model of the economic long wave, *System Dynamics Review*, Vol.1, Num 1.

Richardson G.P., 1991, *Feedback Thought in Social Science and System Theory*, University Pennsylvania Press.

Rumelhart D.E., J.L. McClelland & PDP Research Group, 1986, *Parallel Distributed Processin Explorations in the Microstucture of Cognition*, Volume 1: Foundation, MIT Pre Cambridge, MA.

Rumelhart D.E., G.E.Hinton, R.J.Williams, 1986, "Learning Internal representations by Err Propagation", in *PDP Vol. 1*, The MIT Press.

Salchenberger L.M. E.M. Cinar & N.A. Lash, 1992, "Neural Networks: A New Tool for Predicti Thrift Failures", *Decision Sciences*, Vol.23.

Seligman M.E.P., 1975, *Helplessness: on depression, development, and death*, W.H. Freeman a company.

Simon, H.A., 1978, Rationality as Process and as Product of Thought, *American Economic Revie* Vol. 68, No. 2, pp.1-16.

Sterman J.D., 1989, Modeling Managerial Behavior: Misperceptions of Feedback in a Dynar Decision Making Experiment, *Management Science*, Vol. 35, No. 3, pp.321-339.

Stone, G.O., 1986, "An Analysis of the Delta Rule and the Learning of Statistical Associations", *PDP Vol. 1*, The MIT Press.