Implanting Neural Network Elements in System Dynamics Models to Surrogate Rate and Auxiliary Variables

Mahmood Alborzi, Ph.D.
Islamic Azad University Science & Research Branch
Tehran, Iran

February, 2006

Abstract

Rate variables and auxiliary variables in System Dynamics models are normally constructed using functional equations and or table functions. To construct functions, however, it is imperative to know the underlying relation between the independent variables and the dependent variable. This, we know, is not always an easy task. Indeed, in many differentially non-linear or chaotic situations this may be totally impossible. One may have to resort to less accurate representations if constrained to write relations as equations or tables. Neural Networks has been deployed in many fields to capture the underlying structural relations between variables in such situations through training schemes. When trained, Neural Networks may achieve generalization capabilities though literarily as black boxes. As Neural Networks models when trained can work online like a function, they can be easily implanted within System Dynamics models to surrogate rates or auxiliary variables. The idea in this article is, in situations were it is not possible or it is considerably difficult to construct explicit functions or tables, to deploy Neural Networks to surrogate functions. Neural Network models, here called elements, can be trained on actual data to capture the underlying functional relationships between input output variables and implanted as rates or auxiliary variables to carry out computation on line.

Key Words: *System Dynamics, Neural Networks, Chaos, Non-linearity*

Introduction

In System Dynamics models, while stock variable are integrals of rate variables, the rates and auxiliary variables are given as causal functions of some input variables and constants. This is because the philosophy behind System Dynamics modeling is basically the prevalence of casual relationships (Forrester, 1961). It is consequently imperative to know the causal relationships and have a clear picture of the underlying structures to construct appropriate functional equations or tables for the rates and auxiliary variables. Without knowing the causal relationships and without having a clear picture of the underlying structure between the input variables and the rate or auxiliary variables it is not an easy task to constructing the appropriate functional equations or tables.

Indeed, in many differentially non-linear or chaotic situations this may be totally impossible. We specifically know that chaotic relations though deterministic in nature do not lend themselves to any normal mathematical equation (Stewart, 1990), hence the absence of such equations to start with. One may have to resort to less accurate representations if constrained to write relations as equations or tables. This may adversely affect the validity of the final model.

Neural Networks has been deployed in many fields to capture the underlying structural relations between variables in such situations through training schemes (Haykin, 1994). When trained, Neural Networks may achieve generalization capabilities though literarily as black boxes. As Neural Networks models when trained can work online like a function, they can be easily implanted within System Dynamics models to compute rates or auxiliary variables. The idea in this article is, in situations were it is not possible or it is considerably difficult to construct explicit functions or tables, to deploy Neural Networks to surrogate functions. Neural Network models, here called elements, can be trained on actual data to capture the underlying functional relationships between input output variables and implanted as rates or auxiliary variables to carry out computation on line.

Neural Networks as Function Approximate

Adapting from Aleksander and Morton (1990), as quoted in Haykin (1994), a neural network can be defined as follows:

A neural network is a massively parallel distributed processor that has a natural property for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store knowledge.

A neural network derives its computing power from, first, its massively parallel structure and second its ability to learn and therefore to generalize. In this way it is able to solve complex problems which are currently intractable (Haykin, 1994).

Neural networks are consequently often used to approximate functions. Here inputs to the neural networks are the independent variables and the outputs are the dependents variables. This is specifically constructive when the underlying relations between variables are not known to the researcher or when the relations are differentially nonlinear or chaotic in nature making it impossible to be mathematically captured. Neural networks have often proved useful in such instances in capturing the underlying structures even if non-linear or chaotic and hence able to surrogate for mathematical functions.

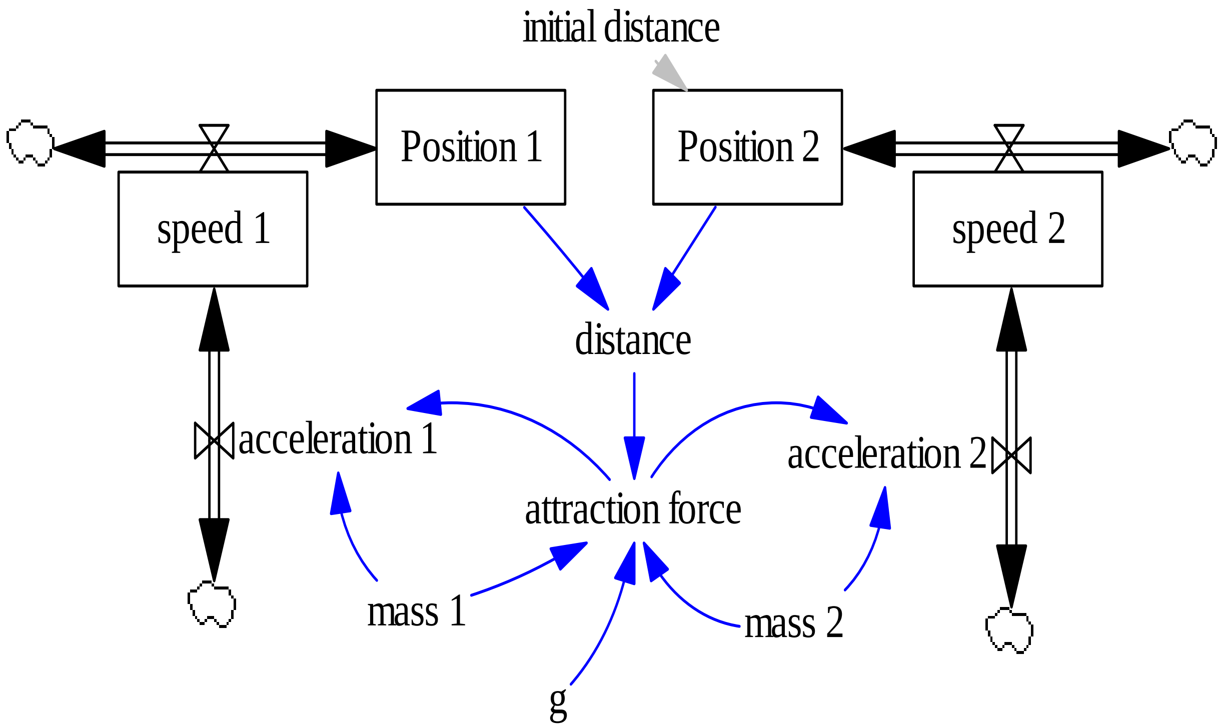
This is because neurons which make up the neural network model are basically nonlinear devices. As a result the network which is made of the neurons is nonlinear itself. Further more the nonlinearity of the neural network is of a special kind in the sense that it is distributed throughout the network (Haykin, 1994). This renders the model an extremely important property to handle nonlinear cases where the underlying physical structural relationship between the input and output variables is nonlinear in nature.

Dynamic systems, by nature, are often complex. Some variables in the system may have nonlinear or chaotic relationships to each other. The prevalence of nonlinearity or chaos may make it difficult to construct highly valid System Dynamics models, as such relations would not lend themselves to normal mathematical formulas. The practice in System Dynamics modeling, to represent rates and auxiliary variables as functional equations or table functions may inherit such difficulties. Neural Network models may be deployed, in such instances, to surrogate the rates and auxiliary variables. Individual Neural Networks models, here called elements, may be trained on actual data and inserted within the model to work on line to compute rates and auxiliary variables.

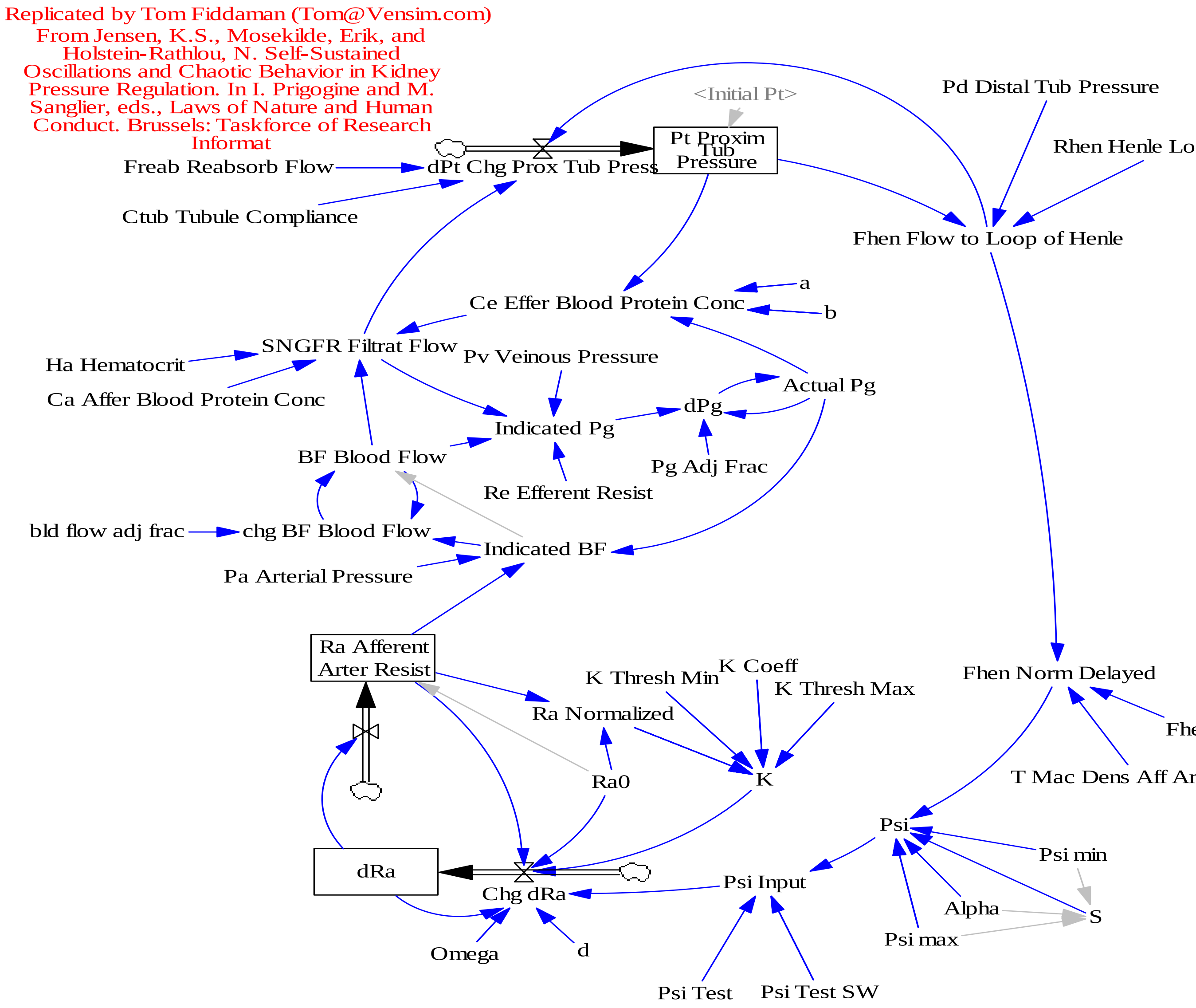
Some Hypothetical Examples

In the simple model of gravity shown below, adopted from the Vensim software demo, the auxiliary variable "attraction force" is computed by an equation i.e. (g*mass1*mass2/distance*distance)**.** This is fortunate because we know the underlying functional relationship from physics. In the absence of this knowledge we may have not been able to make an accurate representation of the relation ship with an equation. We could alternatively do an experiment whereby we could record the actual attraction forces given the two masses with different distance measurements. The data so obtained could be used to train a neural network model to surrogate the equation for the attraction force.

Same argument applies to the rate equations, accelerations 1 and 2, as well. Of course we know from physics that acceleration is attraction force over mass. Yet if we did not know this from physics we would have been at loss again.



In another more complicated example which involves chaotic behavior, from the field of medicine given below, again adopted from the Vensim software demo, it might have been possible to use neural network elements to surrogate some of the empirical equations e.g. the rate variable Chg. dRa i.e (-2*d*Omega*(K)^0.5*dRa-Omega*Omega*K*(Ra Afferent Arter Resist-Psi Input*Ra0) if that was found to be more appropriate.



Conclusions

There are situations where the underlying structure between variables are not known to the researcher or where the underlying structure does not lend itself to close mathematical formulas, e.g. when there is chaos. It may therefore be impossible or not accurate enough, when modeling, to construct formulas or table functions to represent some rates or auxiliary variables. It may be more appropriate, in these cases, to resort to Neural Networks to capture the underlying structures and surrogate for these rates or auxiliary variables. Neural Networks has been deployed in many fields to capture the underlying structural relations between variables in such situations through training schemes. When trained, Neural Networks may achieve generalization capabilities though literarily as black boxes. As Neural Networks models when trained can work online like a function, they can be easily implanted within System Dynamics models to surrogate rates or auxiliary variables.

References

Aleksander, I. and H. Morton (1990). *An Introducation to Neural Computing*. London: Chapman & Hall.

Forrester, J. W. (1961). *Industrial Dynamics.* Massachusetts: The MIT Press.

Haykin, S. (1994). *Neural Networks- A Comprehensive Foundation*. New York: Macmillan.

Stewart, I. (1990). *Does God Play Dice? : The Mathematics of Chaos*. Oxford: Blackwell Publishers.

Ventana Systems (2005). *Vensim-PLE for Windows Version 5.5 Demo*. Ventana. Systems, Inc.