ANALYSIS OF SUPPLY CHAINS USING SYSTEM DYNAMICS, NEURAL NETS, AND EIGENVALUES

Luis Rabelo
Magdy Helal

Industrial Engineering and Management Systems Dept.
University of Central Florida
Orlando, FL, U.S.A.

Chalermmon Lertpattarapong

Engineering Systems Division
Massachusetts Institute of Technology
Cambridge, MA, U.S.A.

ABSTRACT

Supply chain management is a critically significant strategy that enterprises depend on in meeting the challenges of today's highly competitive and dynamic business environments. An important aspect of supply chain management is how enterprises can detect the supply chain behavioral changes due to endogenous and/or exogenous influences and to predict such changes and their impacts in the short and long term horizons. A methodology for addressing this problem that combines system dynamics and neural networks analysis is proposed in this paper. We use neural networks' pattern recognition abilities to capture a system dynamics model and analyze simulation results to predict changes before they take place. We also describe how eigenvalue analysis can be used to enhance the understanding of the problematic behaviors. A case study in the electronics manufacturing industry is used to illustrate the methodology.

1 INTRODUCTION

The use of system dynamics (SD) simulation in supply chain management (SCM) started with Jay Forrester's Industrial Dynamics (Forrester 1965). Forrester described a production-distribution system that consisted of the flows of information, materials, orders, money, manpower, and capital equipment across a supply chain (SC). Since then, SD has been widely applicable to SC applications to address various issues. One of the serious SCM problems is the changes in the SC behavior due to external market factors and/or internal system and managerial factors. For example, a change in federal policy to allow private companies to sell warfare materials can cause a sudden increase in demand at weapons manufacturing companies. If not detected early enough, companies would not be able to respond to the increasing demand and would lose potential business to competitors. In the service sector, a sudden decrease in interest rates can cause sudden increase in demand for loans, which a loan company might not be able to satisfy. Both manufacturing and service sectors need to equip themselves with tools to detect changes in their SC and be prepared to counteract any undesirable consequences. What makes it a significantly serious problem is that SC behavior is dynamic and controlled by nonlinear interrelationships and interactions among its components. Small variations in demand, for example, can simply cause disproportional major fluctuations and oscillating reactions along the SC.

This paper introduces a methodology for detecting and predicting SC behavior changes based on the dynamics of the supply chain environment. This methodology has three phases: (1) capture the dynamics of the SC, (2) detect changes and predict the behavior based on them, (3) make modifications to avoid (or mitigate) the unwanted behavior in performance. System dynamics is a good methodology to model the SC system and the impact of changes in the business environment. Neural networks capture the dynamics from the system dynamics models and historical data. Then, neural networks can be used to detect and identify the consequences of internal and/or external changes in the behavior of the various SC key parameters. Afterward, linearization and eigenvalue analysis are used to analyze and locate the cause of oscillations, time responses, and support the development of new policies and safety stock levels in the SC. These new policies and safety stock levels can be used to avoid/mitigate the unwanted behavior.

2 BUSINESS ENVIRONMENT – CASE STUDY

A system dynamics model of the SC of an actual electronics manufacturing company (based on the work of Lertpattarapong 2002) is used in this paper to demonstrate the proposed methodology. This company was facing a problem of persistent oscillations in its finished goods inventory and desired capacity. Even though the company has maintained its market share, it has experienced serious competition and demand fluctuations, which in turn impacted its work strategies. The company has been implementing the following SC strategies:

- Utilization of several supplier companies to minimize their bargaining power (Porter 1979).
- Utilization of supply relationship management to guarantee that suppliers (1) provide excellent product quality, (2) meet due dates, and (3) offer competitive prices.

As a market leader, this company supplies its products to major Original Equipment Manufacturers (OEMs) such as Dell and Hewlett-Packard. Many OEMs have changed their strategies by adopting Build-To-Order (BTO) and Just-In-Time (JIT) processes. These changes in the PC in addition to its short life cycles have amplified the coordination problems, which in turn have caused excess inventories and sometimes difficulties to keep up with demandAnother main factor is the intense competition from other companies. The competition has forced the company to introduce more product varieties at lower prices into the market to protect its existing and potential market share. Production capacity is another factor that adds to SC complexity because of its long delays, huge investments, and new products with more complex manufacturing processes than previous generations. In addition, these complementary PC products are at the upstream of the SC for PCs and their resulting fluctuations are higher.

3 SD SIMULATION MODELING

Building the SD model of the company's SC in the previous section followed the steps of Hines (2000). The first step defining the problem. The following steps are understanding the formulations, developing the causal loop diagrams, developing the stocks and flows diagrams, validation, and testing.

3.1 Problem Definition

Problem definition includes identifying the relevant parameters and their respective perceived trends, and stating the problem statement. Several participants (at different levels of the managerial hierarchy) from various departments (e.g., information technology, strategic planning, supply chain, manufacturing) were interviewed in addition the analysis of historical data in order to define the problem. The relevant parameters are listed below (Lertpattarapong 2002):

1. Product Life Cycle
2. Actual Capacity Relative to Desired Capacity
3. Change in Customer Orders
4. Raw Materials Inventory Write-0ff
5. Average OEM Margin
6. Pre-Assembly Component Inventory
7. Throughput Time and Work in Process Inventory
8. OEMs' Inventory
9. Product Inventory

Two problematic issues were then stated as:

- The fluctuations and oscillations in the finished goods inventories with relatively large amplitudes, and
- The growing oscillations in actual capacity relative to desired capacity.

3.2 Causal Loop Diagrams

Causal loop diagrams are the basis on which the SD model is built. They depict, graphically, the interactions and cause-and-effect relationships among the different system parameters. The causal loop diagrams in our case consisted of several segmental loops. Figure 1 depicts the complete causal loop diagram of the current case study. The principal causal loops included are described below:

- **Market share needs production capacity**: Capacity increase means more orders and this translates into more future demand. But if demand increases beyond capacity customers will become unsatisfied because of delivery delays and might shift to competitors.
- **Investments in production capacity depend on market share**: Higher revenues are realized with increases in market share and then more investments can be made on production capacity.
- **Competition increases with the increase in profits and vice versa**: Higher profits creates an environment that attracts new entrants to the market and hence competition, and lower profits creates less motivation for new entrants. In addition, the market growth increases the competition.
- **Growth, new product development and product life cycle**: Higher revenues are realized with increases in market share and then more investments can be made on new product developments. The increase in new product development could lead to the obsolescence of old products.

3.3 Stocks and Flows Analysis

The following step in building a SD model is converting the causal loop diagrams into stocks and flows diagrams and defining the mathematical formulation. The basic SD model in this paper follows the generic models of Sterman (2000). The model is composed of the three connected stocks and flows models that are described below.

3.3.1 The Production Model

This company runs a push-pull manufacturing process: a push process from the pre-assembly processes to the assembly process and a pull process from the assembly process to the packaging and shipping. Main state variables

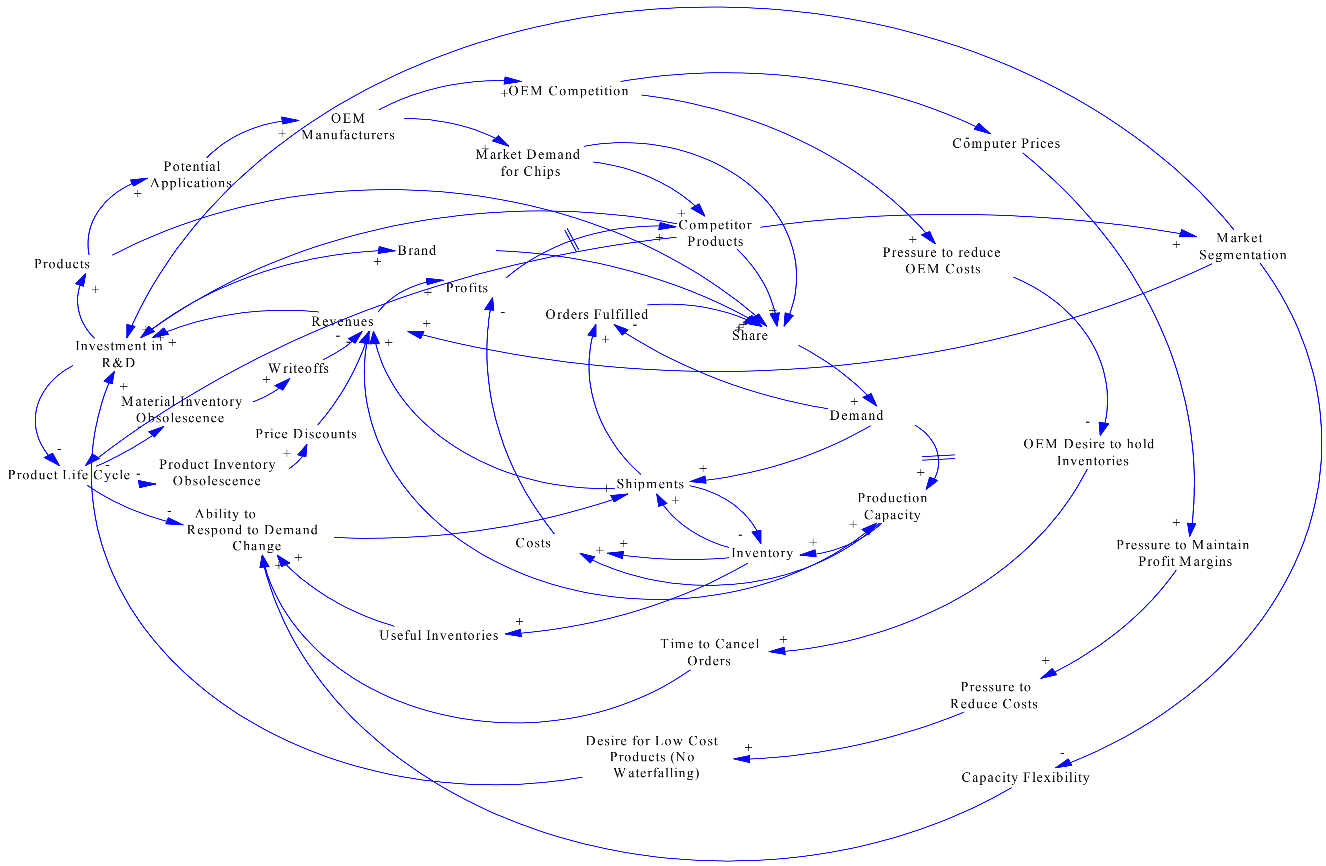


Figure 1: Causal Loop Diagram for the SD Supply Chain Model

in this production model are the inventories. Three types of inventory, which also represented the sequence of the production process, were included in this production model. These are the pre-assembly inventory (PAI), the assembly inventory (AI), and the finished goods inventory (FGI). In the system dynamics notations these state variables are represented as stocks and mathematically as integrations as follows:

$$PAI = \int (PSR - NPR - PR)$$

where PSR represents the production start rate, NPR the net start rate, and PR the production rejection rate.

$$AI = \int (ASR - NAC - AR)$$

where ASR represents the assembly start rate, NAC the net assembly completion rate, and AR the assembly rejection rate.

$$FGI = \int (NAC - S)$$

where S represents the sales rate of the final product.

3.3.2 The Market Share and Shipment Model

This consists of two sub-models: the market share sub-model and the inventory, backlog and shipping sub-model. The market share sub-model mainly represents the causal relationship involving demand, orders filled and company market share. The inventory, backlog, and shipping sub-model represents the links of inventories and shipment orders filled from the finished goods inventory to customers. Main state variables of interest are the finished goods inventory (given above) and the channel order backlog level (COB), which is represented mathematically as follows:

$$COB = \int (CD - OFR)$$

where CD is the channel demand for the company's product, and OFR is the order fulfillment rate.

3.3.3 The Demand Forecast and Capacity Model

This model represents the link from demand to production capacity. The state variables of interest are the perceived present demand (PPD) on the company's product, which is mathematically the integration of the change in the perceived demand over time (this change is a function of historical demand and present channel demand), and the available capacity (AC), which is represented mathematically as follows:

$$AC = \int (CA - CO)$$

where CA represents the rate of capacity acquisition, and CO represents the rate of capacity obsolescence.

The SD model was built in Vensim SD simulation environment. For the details of the development of the SD model and its mathematical structure and validation results refer to (Lertpattarapong 2002).

4 USING THE NEURAL NETWORK ANALYSIS

Neural network (NN) analysis is used to detect changes in the SC behavior and map them into the future. NN, with their pattern recognition capability, are effective mechanisms for that use. This can be very practical, as neural networks can be encapsulated in a software agent that can in turn use the company's ERP records and business intelligence data to perform this task routinely in real-time in an actual system. Using NN to detect changes in the SC will empower companies to detect any changes occurring in the business environment that can affect their SC and hence give the company enough time to adjust its business strategies in order to counteract the impact of these changes (Bruzzone and Orsoni 2003).

To analyze the SC and generate the data that will be used for NN analysis, 17 variables were identified as independent and used as inputs to the SD model (to generate different scenarios). Some of them are variables that the company can control. Others are controlled by the market and level of competition. Beside these variables, an aggregate representation of the current state (i.e., the current values of the state variables: Historical Demand, Available Capacity, Desired Capacity, Pre-assembly Inventory, Assembly Inventory, Finished Goods Inventory, and Channel Order Backlog.) and trends (i.e., recent trend of the state variables) was used as well. Changing some of these variables would require the company to change production policies as for the desired inventory levels. On the other hand changes in others would require the company to consider significant financial investments (i.e. buying new equipment or implementing corporate productivity and quality initiatives such as six-sigma). The selected input variables are the following:

1. Manufacturing Cycle Time
2. Minimum Order Processing Time
3. Time to Complete Assembly
4. Time to Adjust Backlog
5. Time to Perceive Present Demand
6. Capacity Acquisition Delay
7. Safety Stock Coverage
8. Forecast Horizon
9. Backlog Switch
10. Line Yield
11. Component per Lot Yield
12. Time to Adjust Assembly Inventory
13. Pre-assembly Inventory Adjustment Time
14. Time to Adjust Finished Goods Inventory
15. Time to Update Channel Orders
16. Competitors' Attractiveness
17. Channel Demand

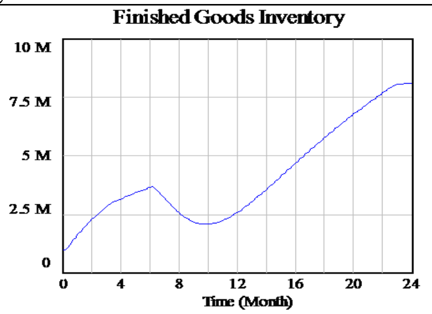
Different values were given to each of the inputs based on the feasible ranges/distributions. The number of possible combinations of the different inputs was enough for a number of simulation runs/scenarios in the order of trillions. However, we wanted to capitalize on the generalization capabilities of NN so that we only needed to generate five different sets each with 800 different combinations, for these input variables. The number of 800 different combinations was provided by following an estimate of the prediction risk as provided by Akaike's final prediction error (Akaike 1970, Moody 1994, Moody and Utans 1994).

The simulation of the SD model was started in equilibrium before changing (i.e., perturbing) the input variables. The simulation was run for 24 months (after the changes) with each combination. It is important to note that when some of the changes were made, the resulting behavior appeared several months later and it was not initially evident. The main focus was to study the future behavior of demand, capacity, and inventory levels during the 24 months. The future behaviors of the state variables were of interest. These were the Historical Demand, Available Capacity, Desired Capacity, Pre-assembly Inventory, Assembly Inventory, Finished Goods Inventory, and Channel Order Backlog.

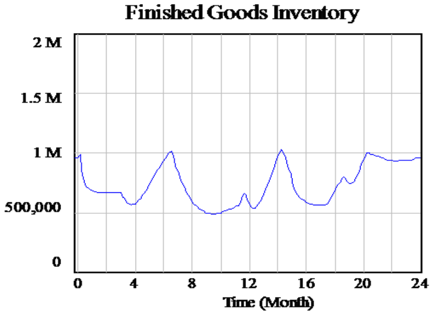
4.1 Data Classification

The behaviors of each SD stock (i.e., state variable) were observed during the future 24 months and classified in categories. The first set of 800 simulation runs was utilized to classify the future behavior of the stocks. We studied the simulation output graphs. Clustering techniques were used to develop and verify the different categories of behaviors. Fuzzy Adaptive Reasoning Theory (ART) neural networks were utilized for this clustering (Carpenter et. al. 1991). Fuzzy ART is a class of neural networks which self-organize categories in response to arbitrary sequences of input patterns in real time for pattern recognition. Fuzzy ART has fuzzy operators which are added in order to handle analog patterns. Behaviors of each of the seven state variables were studied. The classification performed by the Fuzzy ART neural networks can be explained on a qualitative basis. For example, a behavior can be described as initially increasing, then decreasing with fluctuations, or decreasing then oscillating and so on.

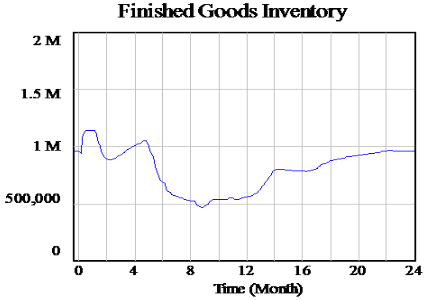
Table 1: Some Behavior Categories of Finished Goods Inventory



Increasing initially, having fluctuations and distortions and ending up at the level above the initial level



Decreasing initially, having fluctuations and distortions and ending up at the level same as the initial level



Increasing initially, having fluctuations and distortions and ending up at the level same as the initial level

The different categories of each stock were validated using the second set of 800 samples. The validation of the categories using this data set was 100% correct. Figure 2 shows some of the possible behaviors of the finished goods inventory level.

4.2 Neural Networks Calculations

Input data and behavior classifications (output) were properly scaled and pre-processed. In training the supervised neural networks the backpropagation approach was used. We used the last three data sets. One data set was for training, another for validation, and the last one was for testing (each data set had 800 samples).

Many epochs were run through each training set, and the possible minimum training error was obtained for different architectures and learning algorithms. The architecture with the minimum validation error was used for the testing purpose. Several learning algorithms were tested. These algorithms were those involving gradient descent optimization, regularization parameters, Bayesian, Levenberg-Marquardt (using second order derivatives), and conjugate gradient-based schemes (Hagan et. al. 1995). The different backpropagation algorithms (Werbos 1994) were trained and compared to search for a suitable algorithm for the problem. The execution of a trained network was on the order of microseconds for all algorithms. Levenberg Marquardt provided the most reliable and fast training option.

NN are chaotic systems and the final training error is a function of the starting point such that every time a unique result can be obtained. The experiment was run many times (25 times) in order to get the best result. NN architectures with 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 17, 18, 19, 20, 30, and 40 hidden neurons were tested. We then calculated the validation error, which signifies the amount of error of the validation data set.

The architecture with five hidden neurons showed the minimum validation error and was selected for further evaluation and optimization using the training data set. It was then tested using the testing data set and the final testing error obtained was considerably smaller (~ 0.025 RMS). The qualitative result based on the category selected, not the root mean squared error indicated a very high performance in testing (~99% accuracy). Table 2 shows training and validation errors for the different arch-

Table 2: Training and Validation Errors for Different NN Architectures Using Levenberg-Marquardt

Architecture	Training Error	Validation Error
2	0.0555	0.0611
3	0.0518	0.0602
4	0.0497	0.0651
5	0.0462	0.0595
6	0.0461	0.0684
7	0.0432	0.0742
8	0.0414	0.0643
9	0.0411	0.0719
10	0.0398	0.0811
11	0.0389	0.0797
12	0.0371	0.0834
13	0.0345	0.0813
14	0.0323	0.0799
15	0.0313	0.0725
16	0.0300	0.0808
17	0.0295	0.0901
18	0.0276	0.0962
19	0.0268	0.0942
20	0.0258	0.0929
30	0.0118	0.128
40	0.0106	0.1325

itectures. Notice that the five-hidden neurons architecture shows the minimum validation error.

To implement this methodology, a database, which continuously stores the behavior patterns occurring in the supply chain needs to be evaluated in real-time. A computer system utilizing this methodology should be able to detect any change and provide predictions such that correction actions or needed decision could be made to adjust the SC behavior. Software agents can be used for this purpose.

5 EIGENVALUE ANALYSIS OF SC BEHAVIOR

The third phase of the analysis methodology is the analysis of the causes of the unwanted SC behavior. This analysis phase provides better understanding of all the different classes of the SC behaviors, as well as how they are linked to particular causal relationships. The SD simulation model for our case study was developed as a system of nonlinear differential equations. To analyze the SC model and study its dynamic behavior, we used several techniques including decomposition, linearization, and eigenvalue analysis.

As the behavior of the model is non-linear, it does not comply with the properties of superposition and homogeneity. In addition, not only is the emergent behavior of the stocks non-linear, but the individual static relationships between certain variables are as well. Therefore, we decomposed the original model by "linearizing" the differentnonlinear static relationships. The resulting models (the decomposed model) applied at specific points in time with the same settings reproduced the exact behavior of the original model.

For the purpose of verification, a particular scenario with specific settings was used. With these settings and the system in equilibrium with no oscillations, the variable channel demand experienced a sudden increase of 10% of its value at the sixth month. The resulting behavior of the different inventories was oscillatory, as shown in Figure 2, for the finished good inventory, starting at the eighth month (two months after the 10% increase). This was the same behavior of the original SC model before decomposition. Decomposition is discussed in the following section.

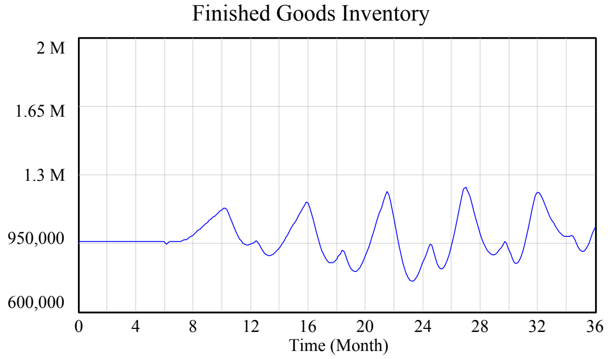


Figure 2: Finished Goods Inventory Oscillations Due to 10% Increase in Demand at Time = 6

5.1 Model Decomposition

We decomposed the original model into several models to linearize individual non-linear static relationships. A system of nonlinear differential equations can be represented as $\dot{x} = g(x)$, where x is the vector of state variables, $g(x)$ is a nonlinear function of x, and $\dot{x}$ is the time derivative of x. The following state variables are considered in decomposing the original model:

1. Available Capacity (AC)
2. Channel Order Backlog (COB)
3. Expected Channel Demand (ECDLSMC)
4. Historical Demand (HD)
5. Perceived Present Demand (PPD)
6. Available Inventory (AI)
7. Pre-Assembly Inventory (PAI)
8. Finished Goods Inventory (FGI)
9. Perceived Fraction Orders Filled (PFOF)
10. Perceived Fraction Orders Filled 1 (LV1)
11. Perceived Fraction Orders Filled 2 (LV2)

The last three variables; the three versions of the perceived fraction order filled, in the list were required to represent the smooth function (the 3rd order exponential smooth function) that was used to capture the perceptions of the electronics company for the fraction of orders filled (See Vensim help information related to the function SMOOTH3 (Ventana Systems Inc. 2004)).

Decomposition was needed due to the nonlinearities at the function level (static relationships). The original model was decomposed into 12 non-linear models. The 12 models superposed in time reproduced exactly the behavior of the original model.

5.2 Linearization and Eigenvalue Analysis

If NNs have detected that the current changes in the SC settings would produce undesirable behaviors (i.e., NN analysis was applied after the 10% sudden increase in Channel Demand), then we would like to investigate more of these current settings and detect the loops and/or variables and/or causal relationships that are (or will be) causing the negative behavior. Ultimately, we would like to see how we can mitigate these effects. Eigenvalue analysis can support these investigations. The different behavior modes of a linear dynamic system are driven by the eigenvalues of the dynamic matrix. Therefore, eigenvalue analysis can tell us about the causes of the oscillatory behavior and the stability of the supply chain.

5.2.1 Linearization

The decomposed models are very non-linear. Therefore, we decided to linearize them by using Taylor expansion for the current settings of the SC. The slope of a non linear curve at a certain point is a good approximation to the curve over a small range around the current point. Thus a nonlinear system can be treated as linear for a small variation about an operating point $\tilde{x}$ and hence can be represented by $\dot{\tilde{x}} = A\tilde{x} + B$. If $\tilde{x}$ depends upon several excitation variables x_1, x_2, ..., x_n, then the functional relationship is written as $\tilde{x} = g(x_1, x_2, \ldots, x_n)$. Taylor expansion about a certain operating point is useful for a linear approximation to the nonlinear function. When the higher-order terms are neglected, the linear approximation of the different components of $\dot{\tilde{x}}$ can be written as follows, where x_0 is the operating point.

$$g(\bar{x}_{10}, \bar{x}_{20}, \ldots, \bar{x}_{n0}) + \left.\frac{\partial g}{\partial x_1}\right|_{x=\bar{x}} (x_1 - \bar{x}_{10}) +$$

$$\left.\frac{\partial g}{\partial x_2}\right|_{x=\bar{x}} (x_2 - \bar{x}_{20}) + \ldots + \left.\frac{\partial g}{\partial x_n}\right|_{x=\bar{x}} (x_n - \bar{x}_{n0})$$

5.2.2 Eigenvalues

Taking a particular model and the current SC settings and its period of time of application, the model is approximated at a specific time using Taylor expansion. Matrix A can then be found for the modified models. For the square matrix A, if there exists a nonzero vector v and a scalar λ such that Av = λv and λI – Av = 0, then the values of λ = λi for which the second equation is satisfied are the eigenvalues of A, and v is the eigenvector. The equations have a solution if and only if |λI – A| = 0.

Figure 3 plots the eigenvalues for 11 state variables with respect to time, using at each specific time the appropriated model. The figure shows the behaviors over a 36-month period (the first 12 months the model was in equilibrium and the perturbations were applied in the 6th month). It clearly shows the oscillations in the systems behavior due to its settings (and the oscillatory behavior was predicted by the neural networks phase). The eigenvalues identify the stability of the system; being in equilibrium is not a guarantee of robustness. The eigenvalue analysis can support the analysis of those conditions and provide even a quantification of the oscillatory behavior.

5.2.3 Sensitivity Analysis

Now, that we have decomposed and linearized the model, we can now experiment by changing some of the input control parameters to see if we can avoid or mitigate the undesired behavior. We can experiment with the control parameters in order of "cost" and implementation time. For example, changing the time to adjust inventory policies is less expensive than changing the manufacturing cycle time. Changing the policies for Time to Adjust Assembly Inventory can be done with training and a series of meetings to coordinate these changes company-wide. On the other hand, changing the manufacturing cycle time can be obtained by "buying more equipment," and/or implementing an aggressive Six-Sigma program which can require more investments and time.

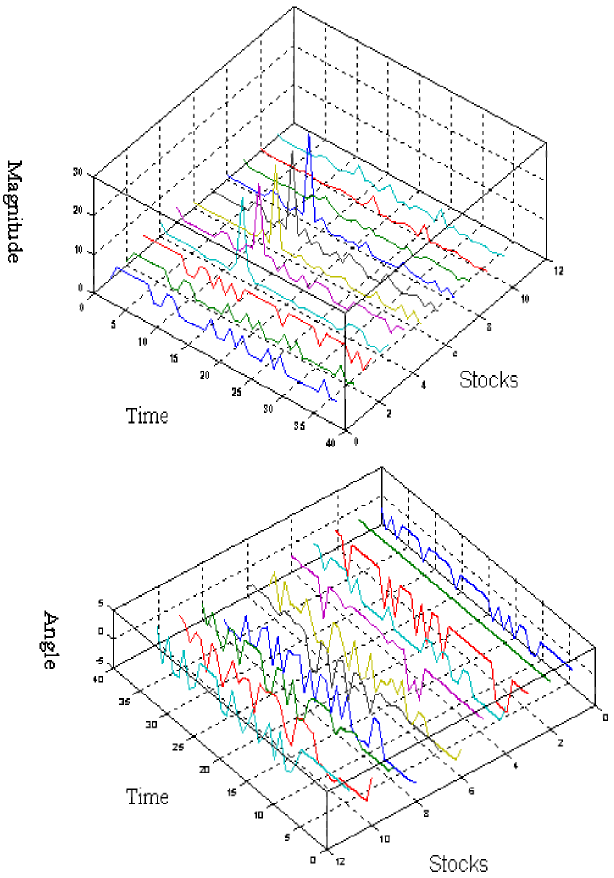


Figure 3: Eigenvalues over 36 Months; 30 Months after the NN Predicted the Undesirable Behavior, for the 11 State Variables

As an example, the variables of Time to Adjust Assembly Inventory (TAAI), Time to Adjust Finished Goods Inventory (TAFGI), and Pre-Assembly Adjustment Time (PAT) were selected. Several experiments using different levels of these variables and changing them (design of experiments) were performed.

There were changes in the output of the stocks and the way they behaved. Some of the combinations made the stocks oscillate more, and some made the oscillations totally damp and become constant. Hence, we conclude that if there is a change of time to adjust inventories in the manufacturing, it will have an effect on the oscillation in the system. The best combination that mitigated the unwanted behavior was to set TAAI to four Weeks, TAFGI to 8 Weeks, and PAT to four Weeks. These changes do not only mitigate the oscillations, but also make the system more robust to perturbations.

For example Figure 4 shows that oscillations were eliminated in the finished goods inventory behavior by carefully setting these times. The oscillating behavior shown is due to current settings when the demand was increased 10% on month six. After adjusting the times to adjust inventories the performance reached stability after six months after the demand increase. Compare the finished goods inventory behavior in Figure 4 to its behavior in Figure 2 (which is also shown in Figure 4 as well)

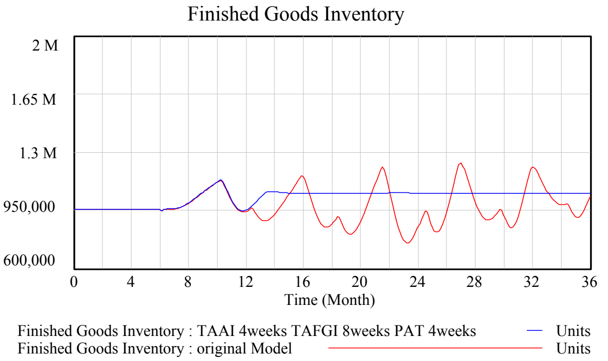


Figure 4: Sample Results for a Good Combination of TAAI, TAFGI, and PAT (in this Particular Figure for Finished Goods Inventory).

It is concluded that the oscillatory behavior is endogenous. Internal actions – including adjusting the times to update production inventories such as PAT, TAAI, and TAFIG could eliminate or minimize the oscillations. Lertpattarapong (2002) reported that this result was not expected by management, who believed that the main causes for their problems were exogenous and not under the company's control.

6 CONCLUSIONS AND FUTURE WORK

This paper proposed and discussed the use of a methodology to detect changes in the SC behavior due to external and/or internal factors. Supply chain management solution methods available today do not have the capability to detect changes taking place in the business environment and hence are not able to provide the companies with accurate predictions for the effects of these changes. The proposed methodology used systems dynamics simulation and historical data, neural networks, and eigenvalue analysis in a three-phase procedure. System dynamics simulation was used to capture the dynamics of the supply chain behavior. Neural networks and eigenvalue analyses used the simulation results to detect and study the behavioral changes and oscillations as well as to locate the causes of them.

It is proposed, for future work, that optimization techniques be used with the decomposed, linearized SC models. In addition, the trained neural networks can be used to perform similar analysis to the one developed using eigenvalues. The methodology suggested here can be applied to assist in implementing six sigma programs and other lean enterprise initiatives as well.

ACKNOWLEDGMENTS

We are grateful for all the excellent teachings about System Dynamics provided by Dr. Jim Hines (MIT Sloan School of Management).

REFERENCES

Akaike, A. 1970. Statistical predictor identification, *Annals of the Institute of Statistics and Mathematics* 22: 203-217

Bruzzone, A. and Orsoni, A. 2003. AI and simulation-based techniques for the assessment of supply chain logistic performance. *The 36th Annual Simulation Symposium*, March 30 – 2 Apr, Orlando FL, 151-161

Carpenter, G., Grossberg, S., and Rosen, D. 1991. Fuzzy ART. Poster paper presented at the *Neural Networks for Vision and Image Processing Conference*, Wang Institute of Boston University, May 10-12

Forrester, J. 1965. *Industrial Dynamics*. 4th Ed. MIT Press.

Hagan, M., Demuth, H., and Beale, M. 1995. *Neural Network Design*. 1st Ed. PWS Publishing Company, USA.

Hines, J. 2000. *Notes on Model Analysis*. System Dynamics Group, Sloan School of Management. Massachusetts Institute of Technology.

Lertpattarapong C. 2002. Applying system dynamics approach to the supply chain management problem. Master thesis. System Design and Management Program. Sloan School of Management. Massachusetts Institute of Technology.

Moody, J. 1994. The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems. *Advances in Neural Information Processing System 4*, ed. J. Moody, S. Hanson, and R. Lippmann, Morgan Kaufman Publishers. 847-854

Moody, J. and Utans, J. 1994. Principled Architecture Selection for Neural Networks: Application to Corporate Bond rating Prediction. *Advances in Neural Information Processing System 4*, ed. J. Moody, S. Hanson, and R. Lippmann, Morgan Kaufman Publishers. 683-690

Porter, M. 1979. *How Competitive Forces Shape Strategy*. Harvard Business School Press.

Sterman., J. 2000. *Business Dynamics – Systems Thinking and Modeling for a Complex World*, McGraw Hill, New York, USA.

Ventana Systems Inc. 2004. Vensim user guide. Available online via `<http://www.vensim.com>` (Accessed July 10, 2004).

Werbos, P. 1994. *The Roots of Backpropagation*. John Wiley & Sons. USA.

AUTHOR BIOGRAPHIES

LUIS RABELO is an Associate Professor in the Department of Industrial Engineering and Management Systems at the University of Central Florida and a NASA Fellow. He holds M.S. degrees in Electrical Engineering, and Engineering Management, and a Ph.D. in Engineering Management. In addition, he has a joint MS in Engineering and Business Management from MIT. Luis has worked as a program manager for several fortune 500 companies. He has three international patents and over 140 publications.

MAGDY HELAL received an MS in Industrial Engineering in 1999, and a Bachelor in Manufacturing Technology in 1993, both from Benha Higher Institute of Technology (BHIT) in Egypt. Helal has been working as a full time researcher and teaching assistant in BHIT since 1993 until he moved to the US in 2001 to pursue a Ph.D degree in industrial engineering. Currently he is a Ph.D. candidate at the Industrial Engineering Dept. at the University of Central Florida.

CHALERMMON LERTPATTARAPONG is pursuing his Ph.D. in Engineering Systems Division at Massachusetts Institute of Technology (MIT). He has an MS in Engineering and Business Management from MIT, an MS in Systems Science and Mathematics from Washington University, an MS in Electrical Engineering from Illinois Institute of Technology, and a BE in Electrical Engineering from Chulalongkorn University in Bangkok, Thailand.