Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

STABILITY ANALYSIS OF THE SUPPLY CHAIN BY USING NEURAL NETWORKS AND GENETIC ALGORITHMS

Alfonso Sarmiento
Luis Rabelo
Ramamoorthy Lakkoju

Industrial Engineering and Management Systems Dept.
University of Central Florida
Orlando, FL 32816, U.S.A.

Reinaldo Moraga

Industrial and Systems Engineering Dept.
Northern Illinois University
De Kalb, IL 60115, U.S.A.

ABSTRACT

Effectively managing a supply chain requires visibility to detect unexpected variations in the dynamics of the supply chain environment at an early stage. This paper proposes a methodology that captures the dynamics of the supply chain, predicts and analyzes future behavior modes, and indicates potentials for modifications in the supply chain parameters in order to avoid or mitigate possible oscillatory behaviors. Neural networks are used to capture the dynamics from the system dynamic models and analyze simulation results in order to predict changes before they take place. Optimization techniques based on genetic algorithms are applied to find the best setting of the supply chain parameters that minimize the oscillations. A case study in the electronics manufacturing industry is used to illustrate the methodology.

1 INTRODUCTION

During the last decade, manufacturing enterprises have been under pressure to competently cope with a market that is rapidly changing due to global competition, shorter product life cycles, dynamic changes of demand patterns and product varieties and environmental standards. In these global markets, competition is ever increasing and companies are widely adopting customer-focused strategies in integrated-system approaches (Shin and Leem 2002). In addition, push manufacturing concepts are being replaced by pull concepts, and notions of quality systems are getting more and more significant.

Globalization of products and services and the rapid changes in technology have also resulted in increasingly dynamic markets and greater uncertainty in customer demand. The enlarged geographic scope of facilities that are consequence of this globalization process has increased the difficulty of managing and controlling supply chains. Moreover, competition has evolved from one company against other companies to one supply chain against other supply chains.

Supply chain management (SCM) is seen as a mechanism that will allow companies to respond to these environmental changes and has become one of the top priorities on the strategic agenda of industrial and service businesses. The objective of SCM activities is to provide right quality of the right product at the right time. The attempt is to improve responsiveness, understand customer demand, control production or service processes, and align together the objectives of all partners in the supply chain. To achieve this goal, companies need to have the ability to predict and control unexpected events taking place in the supply chain (SC).

Effectively managing a supply chain requires visibility to detect unexpected variations at an early stage. This paper introduces a methodology that first uses system dynamics (SD) to model dynamic behavior of the SC. Then neural networks (NNs) are used to capture the knowledge of the SC model and make it available to the enterprise to detect changes and predict the future behavior of the SC. Finally, optimization is applied to make modifications in the SC settings in order to avoid (or mitigate) the undesirable behaviors and performances.

The integration of these techniques to detect and reduce oscillatory behavior of the SC has been proposed in the literature (Rabelo *et al*. 2006; Moraga *et al*. 2007). However, in most cases sensitivity analysis is used to determine the changes in the model parameters required to stabilize the system. We propose a genetic algorithm based approach that minimizes the area under the curve of the state variable of interest in order to achieve stability. This represents an alternative to the concept of using the norm of the state vector in which the control theory relies on (Khalil 1996).

2 STRUCTURE OF THE METHODOLOGY

The proposed approach is a procedure that detects and reduces SC undesired behaviors based on the dynamics of the supply chain environment. The general functioning of the approach is depicted in Figure 1.

The Supply Chain Environment is characterized by exogenous and endogenous factors that make the actual supply chain have certain behavior patterns. From this environment, a supply chain configuration described through the input vector is taken and entered to the Behavior Monitor Module (BMM), which is based on the use of Neural Networks and their pattern recognition capabilities. This monitor system predicts the supply chain behavior that the actual configuration may cause in the future (mid-long range terms). If the predicted behavior is a desired pattern no action is taken over the actual supply chain operations, otherwise the optimization module is used to search for the best configuration of the decision variables. Actions would be required over the actual supply chain operations to apply the best configuration found and get the desired behavior pattern in practice.

2.1 SD Model of the Supply Chain

The SC environment represents the actual participants, structure, strategies, policies, objectives, variables, constraints and parameters that configure different scenarios of the supply chain over time. The SC environment and its dynamics are represented by a SD model. The output of the SD simulation is composed by the SC state variables (state vector), which will be used as inputs to the BMM and the Optimization module. SD modeling is a methodology for studying the dynamics of real-world systems. It was introduced by Jay Forrester (1961) and has its origins in control engineering and management. The essential idea in SD is that all objects in a system interact through causal relationships that form the structure of any system.

System dynamics modeling requires the identification of the causal relationships that capture the system feedback mechanisms or loops. The system dynamics arise from the interaction of two types of feedback loops: positive and negative loops. Positive feedbacks tend to amplify disturbances in the system, while negative loops force the system behavior toward a certain goal level. From the causal loops, the stock and flow structure is developed. Stocks are accumulators of information that describe the state of the system at any particular time. Flows are rates that are added to (inflows) or subtracted (outflows) from a stock, and they represent the management policies to control and regulate the state of the system. The stock and flow structure is converted into a system of differential equations, which is numerically solved by simulation.

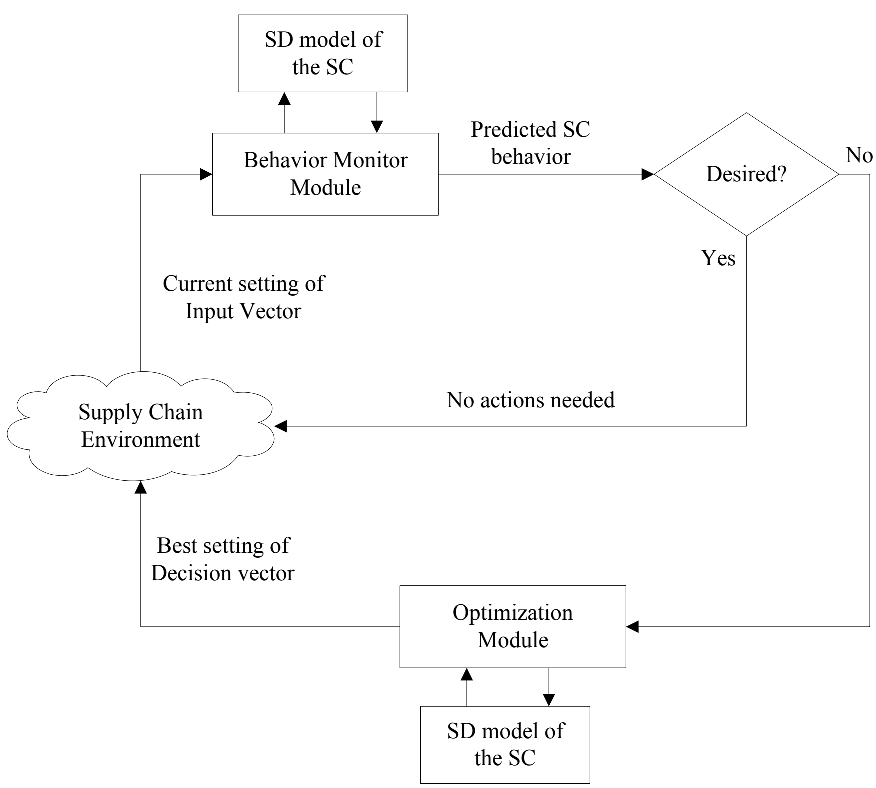

Figure 1: General Procedure of the Methodology

2.2 The Behavior Monitor Module

The BMM is a neural network that is used to capture the knowledge of the SC model and make it available to the enterprise to detect changes and predict the future behavior of the SC. This use of NNs can be very practical, as NNs can be encapsulated in a software agent that can communicate with the enterprise resource planning (ERP) records and business intelligence findings to perform automatic detection of unexpected variations of the SC at an early stage.

NN pattern recognition analysis is conducted using different settings of the input vector in order to map them to the future behavior modes. The input vector (**I**) is a composite vector used by the BMM to predict the behavior mode of the target variables in the state vector. The composition of the input vector is formed by the decision vector (**d**), the current state vector (**s**), and the trend vector (**w**). The decision vector (**d**) is the one that contains the independent variables of the model. The current state vector is the state vector but containing the current values for the state variables. The trend vector contains the values of the state variables from the last two periods. The number of periods selected for trends is not a minor issue because it influences the quality of the prediction.

Before training the NN for the BMM, the behavior modes of the state variables have to be observed and classified. The classification scheme is made on graphs with particular shapes of each simulation scenario over a future horizon in months for each state variable. Fuzzy ART NNs are used to discover similarities in the "graphs", organizing clusters of similar graphs and devising a category scheme based on their shapes and amplitudes. Once the categories for all state variables are obtained and validated, verbalized descriptions of each category are provided.

The supervised backpropagation (Werbos 1994) approach is used for the training of the NN, using different data sets for validation, testing and training. Several epochs are run for each training set in order to obtain the minimum training error for the different NN architectures and learning algorithms. The architecture with the minimum validation error is used for the testing purpose and the testing error is calculated (Morgan and Bourlard 1989). To search for a suitable learning algorithm, several algorithms such as those involving gradient descent optimization, regularization parameters, Bayesian, Levenberg-Marquardt (using second order derivatives), and conjugate gradient-based schemes may be necessary to compare (Hagan *et al*. 1995).

To implement this methodology in actual operational conditions, a database that continuously stores the behavior patterns occurring in the SC will be needed. A computer system utilizing this methodology (and integrated with the ERP system) should be able to detect any changes and provide predictions such that corrective actions or needed decisions could be made to adjust the behavior. The optimization method is then used to eliminate the undesired behaviors.

2.3 The Optimization Module

This module uses optimization techniques based on GAs to find the best setting of decision variables to keep the supply chain stable over time. Because the stability of the state variables considered in the SD model of the SC is an optimization problem that requires a continuous search space then a real-coded genetic algorithm (RCGA) is used. The basic structure and genetic operators of this algorithm have been adopted from the algorithm developed by Deb (2001).

The algorithm considers the optimization of one state variable (which is called variable of interest) and consists in three main steps: (1) selecting the state variable of interest and the decision variables to be optimized, (2) initializing the population, and (3) fitness evaluation and new population creation. Step (3) is iteratively performed on each generation of the population until a number of generations are performed. After all the specified number of runs (one run includes several generations) of the algorithm are performed, the individual having the best fitness among all the runs is designed as the solution to the problem.

The variable of interest is selected from the state variables that will show oscillations according to the predictions of the BMM. The criterion used to minimize these oscillations is to minimize the area under the curve of the state variable of interest. The optimization problem should include initially all decision variables of the model. However, if the user can get some insights about specific decision variables that are responsible for the fluctuations of the state variable of interest (for example using sensitivity analysis) then the optimization problem will focus on finding the values of these decision variables that would lessen the oscillations of the supply chain.

The next step in the algorithm is to randomly create an initial population. The population contains several individuals or solutions (population size). Each individual is a set of values, one value for each of the decision variables, where each value is generated between the lower and upper bounds of these variables. The fitness of an individual is nothing but the absolute value of the area under the curve of the state variable of interest generated by simulating the supply chain model with the values of the variables corresponding to that individual.

A new population of individuals is created by applying three genetic operators: (i) selection (ii) crossover, and (iii) mutation. The genetic operators are applied to the individuals in the population chosen with a probability based on their fitness.

The main objective of the selection operator is to make duplicates of good solutions and eliminate bad solutions in a population, while keeping the population size constant. The proposed algorithm uses Stochastic Remainder Roulette-Wheel (SRRW) selection operator to create a new population. The implementation of this selection operator can be thought of as a roulette-wheel mechanism, where the wheel is divided into N (population size) divisions, where the size of each is marked in proportion to the integer part of the expected number of copies of each solution. This expected number is calculated by multiplying the probability of selecting a solution and the population size. Thereafter, the wheel is spun N times, each time choosing the solution indicated by the pointer of the roulette-wheel. The process is repeated until the desired number of individuals is obtained (called mating pool).

A crossover operator called Simulated Binary Crossover (SBX) is applied next to the solutions of the mating pool in order to create new individuals by combining genetic material randomly selected from two parents. The procedure computes the offspring $X_i^{(1,\ t+1)}$ and $X_i^{(2,\ t+1)}$ from parent solutions $X_i^{(1,\ t)}$ and $X_i^{(2,\ t)}$ using a spread factor (β_i) defined as the ratio of the absolute difference in offspring values to that of the parents:

$$\beta_i = \left| \frac{X_i^{(2,\ t+1)} - X_i^{(1,\ t+1)}}{X_i^{(2,\ t)} - X_i^{(1,\ t)}} \right|$$

Firstly, a random number $u_i \in [0, 1)$ is chosen. Secondly, from a specified probability distribution function, the ordinate β_{qi} is found so that the area under the probability curve from 0 to β_{qi} is equal to the chosen random number u_i. The probability distribution used to create the offspring is as follows:

$$P(\beta_i) = 0.5(\eta + 1)(\beta_i)^{\eta}, \text{ if } \beta_i \le 1,$$
$$= 0.5(\eta + 1)(1 / \beta_i^{\ \eta+2}), \text{ otherwise;}$$

where η is a non-negative real number. A large value of η gives a higher probability for creating 'near parents' solutions and a small value of η allows distant solutions to be selected as offspring. After equating the area under the above probability curve to u_i, the value of β_{qi} is given as follows:

$$\beta_{q\,i} = (2u_i)^{1/(\eta+1)}, \text{ if } u_i \le 0.5,$$
$$= [2(1 - u_i)]^{-1/(\eta+1)}, \text{ otherwise.}$$

Thereafter, offspring are computed using the following equations proposed by Deb (2001):

$$X_i^{(1,\,t+1)} = 0.5\ \{(1 + \beta_{q\,i})\, X_i^{(1,\,t)} + (1 - \beta_{q\,i})\, X_i^{(2,\,t)}\},$$
$$X_i^{(2,\,t+1)} = 0.5\ \{(1 - \beta_{q\,i})\, X_i^{(1,\,t)} + (1 + \beta_{q\,i})\, X_i^{(2,\,t)}\}.$$

The need for mutation is to keep diversity in the population. After applying selection and crossover, polynomial mutation operator is used in order to induce some diversity in the population. A mutated solution $Y_i^{(1,\ t+1)}$ using polynomial mutation is obtained as follows:

$$Y_i^{(1,\,t+1)} = X_i^{(1,\,t+1)} + \{X_i^{(U)} - X_i^{(L)}\}\delta_i\ ;$$

where δ_i is calculated from the polynomial probability distribution $P(\delta) = 0.5\ (\eta+1)\ (1 - |\delta|)^{\eta}$ in the following way:

$$\delta_i = (2\, r_i)^{1/(\eta+1)} - 1, \text{ if } r_i < 0.5,$$
$$= 1 - [2(1 - r_i)]^{1/(\eta+1)}, \text{ if } r_i \ge 0.5;$$

where $r_i \in [0, 1)$ is a random number, $X_i^{(U)}$ and $X_i^{(L)}$ are the upper and lower bounds of solution X_i.

After every single run of the RCGA algorithm, the individual that is identified to be having the best fitness (among all the generations in this run) is designated as the result of the algorithm for that run. After all the specified number of runs of the algorithm are performed, the individual having the best fitness among all the runs is designated as final result of the algorithm, which represents a solution to the problem.

3 CASE STUDY

A model of the SC of an actual electronics manufacturing company (LSMC) is used to demonstrate the use of the proposed analysis methodology (Lertpattarapong 2002). The LSMC name is used to respect confidentiality. LSMC products are technological gadgets and personal computer (PC) complementary products. As a market leader, LSMC supplies its products to Original Equipment Manufacturers (OEMs) like Dell, Gateway, and Hewlett-Packard. LSMC was facing a problem of persistent oscillations in its finished goods inventory and desired capacity. Even though LSMC has maintained its market share, it experienced increasing competitive pressures and demand fluctuations, which have impacted its SC performance.

Since 1998, led by Dell, many OEMs have changed their strategies by aggressively eliminating slack in their inventories through a Build-To-Order (BTO) and Just-In-Time (JIT). Further, because of fast dynamic changes in the PC market, the short lifecycle of PCs and other complementary products has also amplified coordination problems, which in turn have often caused excess inventory and sometimes difficulties to keep up with demand. Moreover, 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 supply chain complexity because 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 supply chain for PCs and their resulting fluctuations are higher.

In the following lines we present the analysis conducted for the LSMC supply chain using the proposed methodology to detect and mitigate oscillations due to an unexpected change in demand. For additional information on this case study, the reader is referred to Lertpattarapong (2002).

3.1 SD Model of the LSMC Supply Chain

During the study of LSMC supply chain, 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, available historical data was analyzed in order to identify the relevant parameters and variables of the company's supply chain operations.

From the list of variables and the interviews with the participants, the causal loop diagram that explains the dynamic behaviors of LSMC supply chain was developed. After that, the causal loop was converted into stock and flow diagrams, like the one shown in Fig. 2, and the mathematical formulations were defined. The complete model has more than 91 equations, including differential and auxiliary equations, and it is comprised of three connected stock and flow submodels: (1) the production model, (2) the market share and shipment model, and (3) the demand forecast and capacity model. The validation of the model to represent the operations and polices of LSMC was done by verifying the historical and projected behavior (reference modes) of variables that the team of managers, engineers and planners considered important (Lertpattarapong 2002).

3.2 The Behavior Monitor Module

3.2.1 Category Classification and NN Training

NN pattern recognition analysis was conducted on the SD simulation results. To generate the required data the three components of the input vector were defined, i.e., the decision vector, the state vector and the trend vector. The decision vector **d** contains seventeen parameters; the state vector **s** has seven variables, and the trend vector **w**, fourteen variables. The complete definition of the vectors is shown next.

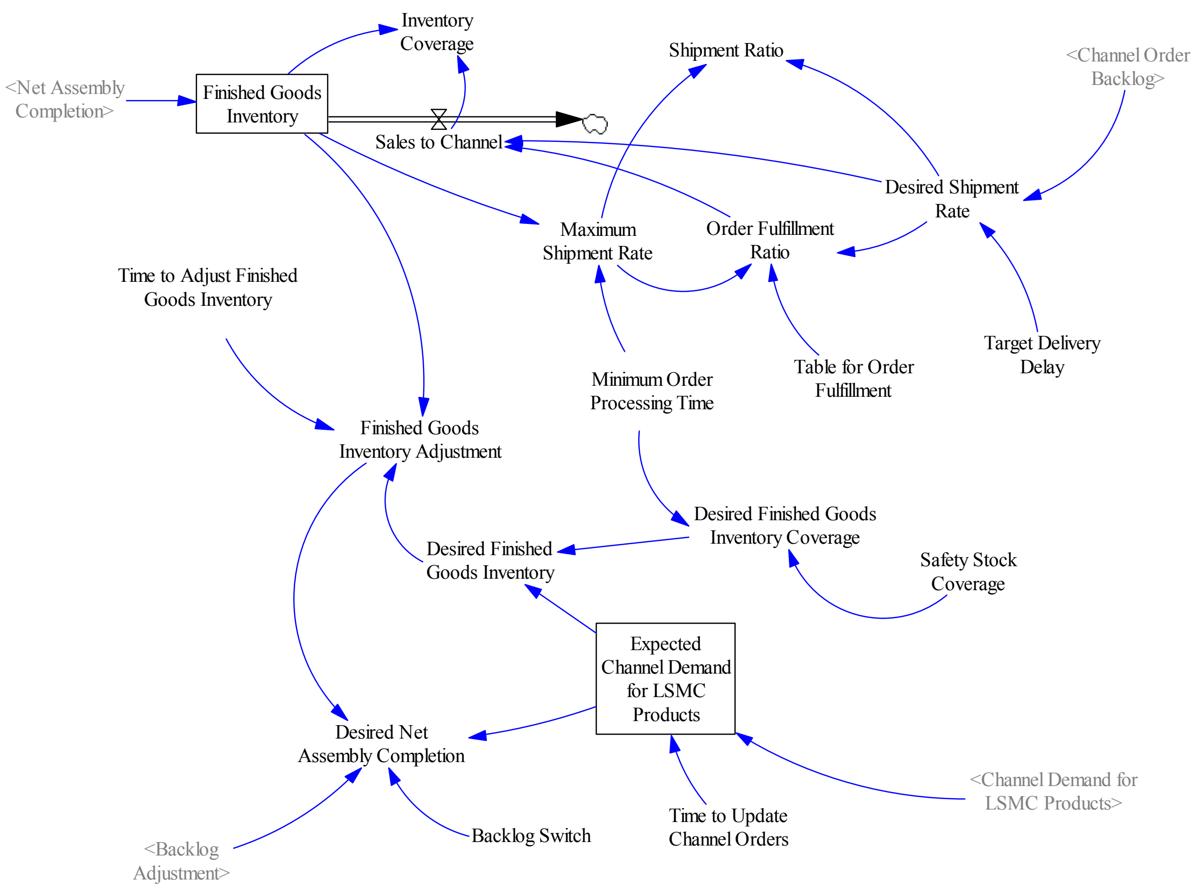

Figure 2: Stock and flow diagram of the Finished Goods Inventory sub-model

$$\mathbf{d} = \begin{bmatrix} \text{Manufacturing Cycle Time} \\ \text{Minimum Order Processing Time} \\ \text{Time to Complete Assembly} \\ \text{Time to Adjust Backlog} \\ \text{Time to Perceive Present Demand} \\ \text{Capacity Acquisition Delay} \\ \text{Safety Stock Coverage} \\ \text{Forecast Horizon} \\ \text{Backlog Switch} \\ \text{Line Yield} \\ \text{Component per Lot Yield} \\ \text{Time to Adjust Assembly Inventory} \\ \text{Pre - assembly Inventory Adjustment Time} \\ \text{Time to Adjust Finished Goods Inventory} \\ \text{Time to Update Channel Orders} \\ \text{Competitor's Attractiveness} \\ \text{Channel Demand} \end{bmatrix} \quad \mathbf{w}(t) = \begin{bmatrix} \text{Historical Demand}(t-1) \\ \text{Available Capacity}(t-1) \\ \text{Desired Capacity}(t-1) \\ \text{Pre - assembly Inventory}(t-1) \\ \text{Assembly Inventory}(t-1) \\ \text{Finished Goods Inventory}(t-1) \\ \text{Channel Order Backlog}(t-1) \\ \text{Historical Demand}(t-2) \\ \text{Available Capacity}(t-2) \\ \text{Desired Capacity}(t-2) \\ \text{Pre - assembly Inventory}(t-2) \\ \text{Assembly Inventory}(t-2) \\ \text{Finished Goods Inventory}(t-2) \\ \text{Channel Order Backlog}(t-2) \end{bmatrix} \quad \mathbf{s}(t) = \begin{bmatrix} \text{Historical Demand}(t) \\ \text{Available Capacity}(t) \\ \text{Desired Capacity}(t) \\ \text{Pre - assembly Inventory}(t) \\ \text{Assembly Inventory}(t) \\ \text{Finished Goods Inventory}(t) \\ \text{Channel Order Backlog}(t) \end{bmatrix}$$

The behaviors of each state variable were simulated during the future 24 months, observed and classified in categories. Five sets each with 800 different scenarios were generated from the simulations. 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).

The 800 graphs of each state variable, obtained from the first set of simulation scenarios, were exposed to the Fuzzy ART NNs. The first Fuzzy ART NN was used for the Historical Demand, which was able to develop nine stable and different categories of behavior. A second Fuzzy ART NN was used for the Available Capacity, generating eleven stable and different categories. Similarly, four categories for the Desired Capacity, eight categories for the Pre-assembly Inventory, nine categories for the Assembly Inventory, six categories for the Finished Goods Inventory, and six categories for the Channel Order Backlog were generated.

The second set of 800 samples was used to validate the different categories of each state variable. The validation of the categories using this data set was 100% correct.

The last three sets of 800 samples each were used for training, validation, and testing. The backpropagation NN was trained using different architectures and learning algorithms (as the ones mentioned in the methodology). The Levenberg-Marquardt algorithm, which provided the most reliable and fast training option, was used to select the best architecture. Different architectures from 2 to 40 hidden neurons were evaluated. The architecture with five hidden neurons showed the minimum validation error and was selected for further analysis. This architecture was then tested using the testing data set and the final testing error was considerable smaller.

3.2.2 Undesired Behavior Detection

For the purpose of illustration, a particular scenario with specific settings was used. With these settings, while the system was in equilibrium with no oscillations, the Channel Demand experienced a sudden increase of 10% of its value at the six month. The resulting behavior of the different inventories was oscillatory, as shown in Figure 3 for the Finished Goods Inventory. The effect of the disturbances starts at the eighth month, i.e. two months after the 10% increase.

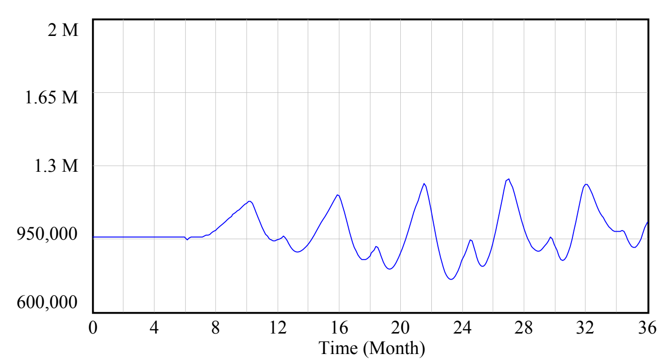

Figure 3: Finished Goods Inventory oscillations

NNs were applied at the seventh month, setting the current state vector and the trend vector for t=7. These NNs were able to predict that the behavior of the Finished

Goods Inventory would be of the "oscillatory with undesired amplitudes" category.

3.3 The Optimization Module

Before running the optimization algorithm, some important observations about the dynamic behavior of the LSMC supply chain were found through simulations. One of the observations is that varying time to adjust inventories has impacts on the oscillatory behavior of the product inventories. Another important observation is that the fluctuation in the Finished Goods Inventory (FGI) oscillates and the amplitude is large compared to demand and capacity (Lertpattarapong 2002). This fact is utilized to select the state variable FGI as the variable of interest.

Moreover, the following five decision variables that are independent and are in control of LSMC were chosen for the purpose of minimizing the oscillations in FGI: Pre-assembly Adjustment Time (PAT), Time to Adjust Assembly Inventory (TAAI), Time to Adjust Finished Goods Inventory (TAFGI), Manufacturing Cycle Time (MCTime) and Minimum Order Processing Time (MOPTime). The graph of FGI for the original LSMC model looks as is shown in Figure 4. The oscillations in the curve can be seen very clearly.

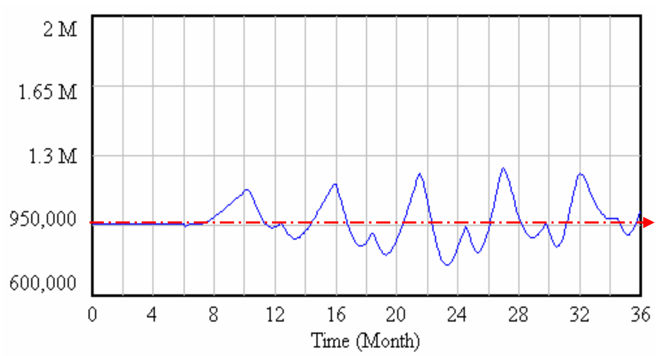

Figure 4: Finished Goods Inventory of LSMC Supply Chain

The criterion used to minimize these oscillations in FGI is to minimize the area under this curve. An imaginary axis (shown as a discontinuous line in Figure 4) is drawn at the initial condition and the absolute value of the area under the curve about this imaginary axis is minimized. When the area under the curve is zero, the curve is just a straight line meaning that FGI remains constant and stable over time. The RCGA takes as input the lower and upper bounds of the five variables mentioned above, and attempts to find the values (for these five variables) that would give rise to a FGI curve with the minimum area possible.

3.4 Results and Analysis of the LSMC Model

The genetic algorithm for LSMC model was run at the eighth month using the settings mentioned below:

1. Number of Generations = 20
2. Population Size = 30
3. Probability of Crossover = 0.7
4. Probability of Mutation = 0.15
5. Lower and Upper Limits for MCTime = (1, 3)
6. Lower and Upper Limits for MOPTime = (0.1, 1)
7. Lower and Upper Limits for TAAI = (0.1, 8)
8. Lower and Upper Limits for PAT = (0.5, 10)
9. Lower and Upper Limits for TAFGI = (0.5, 10)
10. Number of Runs = 10
11. Selection Strategy = SRRW selection
12. Crossover Strategy = Simulated Binary Crossover
13. Exponent (η) for Crossover ($\eta = 2$ is used)
14. Exponent (η) for Mutation ($\eta = 20$ is used)
15. Random seed = 0.123

Table 1 shows the values of the five decision variables obtained from the RCGA by using the above settings in comparison with the original values of these variables.

Table 1: Comparison of new variable values with the original values

	New Values from GAs	Original Values from LSMC Model
Manufacturing Cycle Time	1.163297 months	2 months
Minimum Order Processing Time	0.227299 months	0.25 months
Time to Adjust Assembly Inventory	3.240612 weeks	0.5 weeks
Pre Assembly Adjustment Time	2.266047 weeks	2 weeks
Time to Adjust Finished Goods Inventory	2.672158 weeks	2 weeks

The FGI curve obtained after simulating the LSMC model with these new decision variable values is shown in Figure 5. The new curve obtained using the proposed algorithm is relatively better than the original one in terms of the oscillations occurring in the Finished Goods Inventory. Hence, the proposed methodology, though in its preliminary stage, is quite capable of minimizing the oscillations in the Finished Goods Inventory.

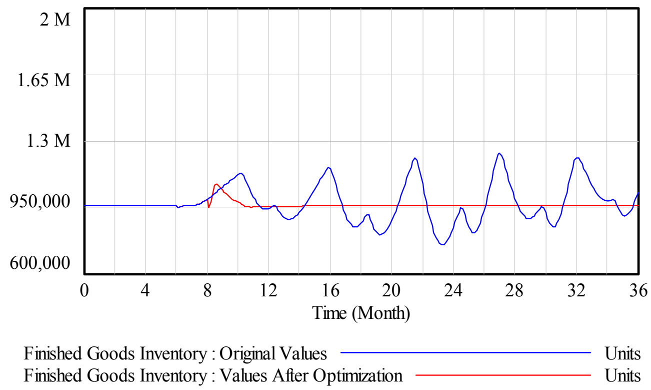

Figure 5: Finished Goods Inventory curve with reduced oscillations

From the comparison of the values (Table 1), it is evident that LSMC has to reduce the Manufacturing Cycle time and Minimum Order Processing time and needs more time to adjust inventories (TAAI, PAT and TAFGI) in order to be able to have fewer oscillations in the Finished Goods Inventory.

4 CONCLUSIONS AND FUTURE WORK

This paper proposed a methodology to detect changes in the SC environment and eliminate possible oscillatory behaviors. Our approach utilizes the modeling flexibilities of system dynamics to model complex systems, the pattern recognition capabilities of neural networks to detect structural changes in a dynamic environment, and the potential of genetic algorithms to scan complex and nonlinear search spaces in order to minimize the oscillatory behavior of the supply chain.

This methodology can contribute to assist in implementing Six-Sigma programs, improve forecasts, and other management initiatives as well. Most important, it will allow the analysis of planning strategies to design stable supply chains that can effectively cope with significant changes and disturbances, with the corresponding cost savings to the companies.

Currently, the proposed methodology takes into consideration only the objective of minimizing the oscillations of one state variable. For future work, we suggest extending the analysis to cover more than one variable of interest and applying stability conditions simultaneously to these variables when optimizing the control parameters of the supply chain.

Because the ability of the searching algorithm to avoid premature convergence in a local optimum is critical for the optimization problem, we will be experimenting with other type of algorithms, such as Particle Swarm Optimization (Kennedy and Eberhart 1975) or hybrid algorithms that combine the ideas of GA and Particle Swarm Optimization models (Engelbrecht 2005). Finally, it is also required the development of a computerized framework that integrates the modules described in this paper in order to perform an automatic model analysis.

REFERENCES

Akaike, H. 1970. Statistical predictor identification, *Annals of the Institute of Statistical Mathematics*, 22(2), 203-217.

Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, Ltd., West Sussex, England.

Engelbrecht, A. P. 2005. *Fundamentals of Swarn Intelligence*, John Wiley & Sons, Ltd., West Sussex, England.

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

Hagan, M., H. Demuth, and M. Beale. 1995. *Neural network design*, PWS Publishing Company, Boston, MA.

Kennedy, J. and R. Eberhart. 1975. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia.

Khalil, H. K. 1996. *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ.

Lertpattarapong, C. 2002. Applying system dynamics approach to the supply chain management problem. Master Thesis, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.

Moraga, R., L. Rabelo, and A. Sarmiento. 2007. Towards a methodology for monitoring and analyzing the supply chain behavior, to appear in *Handbook on Computational Intelligence in Manufacturing and Production Management*.

Morgan, N. and H. Bourlard. 1989. Generalization and parameter estimation in feed forward nets: some experiments. *International Computer Science Institute*, Technical Report, TR-89-017.

Rabelo, L., M. Helal, C. Lertpattarapong, R. Moraga, and A. Sarmiento. 2006. Using system dynamics, neural nets, and eigenvalues to analyse supply chain behaviour – A case study, to appear in *International Journal of Production Research*.

Shin, K., and C. Leem. 2002. A reference system for internet based inter-enterprise electronic commerce. *The Journal of Systems and Software*, 60(3), 195–209.

Werbos, P. 1994. *The roots of backpropagation*, John Wiley & Sons, Inc., New York, NY.

AUTHOR BIOGRAPHIES

ALFONSO SARMIENTO is a Ph.D. candidate at the Industrial Engineering and Management Systems Department at the University of Central Florida. He has a

M.S. degree in Industrial Engineering at the University of Florida, and a B.S. in Industrial Engineering at the University of Lima, Peru. Currently he is working as a Graduate Research Associate at the Engineering Technology Department at the University of Central Florida. His research interests include Supply Chain Management, Operations Management, and Decision Support Systems. His e-mail address is <sarmient@mail.ucf.edu>.

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. His e-mail address is <lrabelo@mail.ucf.edu>.

REINALDO MORAGA is an Assistant Professor in the Department of Industrial Engineering at the Northern Illinois University. He received two B.Sc. degrees in Wood Engineering in 1988 and Industrial Engineering in 1993 from Universidad del Bio Bio, in Chile, and has degrees of M.S. and Ph.D. in Industrial Engineering from the University of Central Florida. His research interests are Simulation, Operations Research, and Supply Chain Management. His e-mail address is < moraga@ceet.niu.edu>.

RAMAMOORTHY LAKKOJU has received his M.S. degree in Industrial Engineering (Operations Research) from the Department of Industrial Engineering and Management Systems at the University of Central Florida. He has a Bachelors degree in Mechanical Engineering from S.V.U. College of Engineering, Tirupati, India. He is currently working in Santander Consumer USA Inc. His e-mail address is < ram_msie@yahoo.com>.