Supply Chain Dynamics and Forecasting

Mu Niu

PhD

2008

Supply Chain Dynamics and Forecasting

Mu Niu

A thesis submitted in partial fulfilment of the requirements of the University of Northumbria at Newcastle for the degree of

Doctor of Philosophy

School of Computing, Engineering and Information Science

University of Northumbria at Newcastle

July 2008

Abstract

Nowadays, the global supply chain system needs to respond promptly to changes in customer demand and adapt quickly to advancements in technology. Supply chain management becomes an integral approach which links together producers, distributors and customers in collaborative management of the whole system. The variability in orders or inventories in supply chain systems is generally thought to be caused by exogenous random factors such as uncertainties in customer demand or lead time. Studies have shown, however, that orders or inventories may exhibit significant variability, even if customer demand and lead time are deterministic. Most researchers have concentrated on the effects of the ordering policy on supply chain behaviour, while not many have paid attention to the influences of applying different forecasting to supply chain planning.

This thesis presents an analysis of the behaviour of a model of a centralised supply chain. The research was conducted within the manufacturing sector and involved the breathing equipment manufacturer Draeger Safety, UK. The modelling process was embedded in the organization and was focused on the client's needs. A simplified model of the Draeger Safety, UK centralised supply chain was developed and validated. The dynamics of the supply chain under the influence of various factors: demand pattern, ordering policy, demand-information sharing, and lead time were observed. Simulation and analysis were performed using system dynamics, non-linear dynamics and control theory. The findings suggest that destructive oscillations of inventory could be generated by internal decision making practices. To reduce the variation in the supply chain system, the adjustment parameters for both inventory and supply line discrepancies should be more comparable in magnitude. Counter-intuitively, in certain fields of decision, sharing demand information can do more harm than good. The linear forecasting *ARMA* (autoregression and moving average) model and the nonlinear forecasting model Wavelet Neural Network were applied as the supply chain forecasting methods. The performance was tested against supply chain

costs. A management microworld was developed, allowing managers to experiment with different decision policies and learn how the supply chain performs.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to acknowledge and thank my supervisor team, Dr Petia Sice, Dr Ian French and Professor Erik Mosekilde. This thesis would not have been possible without their generous support. I would like to express my heartfelt gratitude to my principal supervisor Dr Petia Sice, who has given me intensive supervison throughout my research. I thank her for the time she devoted to a critical reading of this thesis and for her suggestions. I am particularly indebted to Dr Ian French for his support, valuable technical advice and guidance. I appreciate the time he put aside for meetings that always give me new insights. I also owe a debt to Professor Erik Mosekilde for the training in nonlinear dynamic in Denmark.

I would like to thank Vince Smith, Mel Hedley and Lynn Kennedy of Draeger Safety, who gave generously of their time and energy.

I would also like to thank my friends who have supported and helped me, in particular Sujan, who helped me in data processing.

Finally, these acknowledgements would not be complete without thanks to my parents, uncle, sister and my girlfriend for all the support, encouragement and love they have given me.

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or certificate of the university or other institute of higher learning.

Name: Mu Niu

Signature:

Date: 27/02/2009

Chapter 1: Introduction

The contemporary business environment is characterised by an acceleration of the process of change due to the communication and analytical capabilities supported by modern Information and Communication Technologies (Sterman, 2000; Terzi and Cavalieri, 2004; Ortega and Lin, 2004). This poses an ever-increasing demand for periodical reviews of strategies and organisational processes, and for faster and more effective learning to deal with problem situations more quickly. Furthermore, companies increasingly need to react quickly to customer requirements (Hong-Minh, Disney and Naim, 2000). Naylor *et al/.* (1999) suggested the principle of agile manufacturing, i.e. responding to customer requirements with very short lead times, can be applied to supply chains.

1.1 Supply Chain Management Issues

Global companies are realising that the efficiency of their businesses is dependent on collaboration with their suppliers and customers (van der Zee, 2005; Yusuf, Gunasekaran, 2004; Adeleye and Sivayoganatahn, 2004; Cox *et al/.* 2004). The concept *of supply chain management* (SCM) is concerned with the strategic approach of integrating producers, suppliers and customers in a collaborative network, with the objective of improving the overall responsiveness of the network, but at the same time preserving the organisational autonomy of each unit (Terzi and Cavalieri, 2004). More accurate data and the availability of information to all nodes of the supply chain are considered crucial to improving its performance. Thus, companies invest heavily in intra-company information and communication platforms and E-collaboration tools, such as data warehouse or Enterprise Resource Planning systems (ERP) and Manufacturing Resource Planning systems (Lyneis, 2005; Li & Li, 2005; Marquez, Bianchi and Gupta, 2004).

Contemporary Manufacturing Resource Planning (MRP) is a structured approach to manufacturing management in which a suite of integrated computer software covering the main operational business functions is used to assist the planning of materials, capacity and cash flow, in accordance with company policy, to meet customer delivery

requirements (Ackermans & Dellaert, 2005). It is a total approach to managing a business (Li & Li, 2005). It includes inventory control data, purchasing and accounts payable, suppliers' data, master schedules, production schedules, resource capacities, manufacturing orders, etc. MRP systems are developed on the assumption that more data and better information lead to better decisions. The quality of this data is believed to be accurate or at least good enough to be reliable (Ackermans and Dellaert, 2005). However, even with the introduction of resource planning systems, the performance of the supply chain remains problematic (Lyneis, 2005; Fowler, 1999). The weakness of such systems is that when used on their own they fail to address the complexity of management situations for two reasons: firstly, they do not take into account the inherent 'messiness' of situations that contain human decision making within the process that is considered problematic (Maturana & Varela, 1987; Beer, 1979, 1994); secondly, such tools do not promote learning (Senge, 1984, 1996; Venix, 1994; Morecroft, 2007) or effective decision support as they do not include the powerful technique of simulation to allow for what-if analysis of alternative strategies (Terzi and Cavalieri, 2004).

1.2 Draeger Safety, UK. Research Focus

Draeger Safety Ltd. is an international corporation, manufacturing breathing protection and gas detection equipment. It provides innovative products, services and system solutions in such areas as gas measuring technology, personal protection, diving and system technologies. It has a presence in approximately 200 countries on all continents, with around ten thousand employees worldwide. In 2006, the order intake in Draeger Safety was around 1.9 billion Euros. Draeger Safety UK in Blyth is one of the manufacturing plants in the Draeger Group. It is equipped with a sophisticated MRP environment for sharing information and enhancing communication between production and marketing. However, the performance of the supply chain is still perceived was problematic as it was not unusual for high inventories to mount up and backlogs to develop, causing disruption, as production was dealing with the crisis of overcoming them. Forecasts were perceived to play a key role. In the current IT logistics planning system, computer generated forecasting, combined with human decisions, formed the main driver of other functions, such as scheduling, resource planning and marketing. Forecast accuracy was, therefore, an important component in the delivery of an effective supply chain system.

A centralised supply chain system was recently implemented with the purpose of diminishing costs and avoiding backlogs. This made Draeger's planning managers even more worried as it was difficult to predict what the consequences of centralised inventories would be for the manufacturing plant in Blyth. A team consisting of the planning department at Draeger Safety, UK and this researcher was formed to explore the impact of different decision strategies on the behaviour of the supply chain and to provide insights and recommendations for managers. This required the research to focus on:

- Modelling and simulation of the material and information flows, including the decision processes of the centralised supply chain at Draeger Safety, UK;
- Validating the model within the research team and against any available data;
- Analysing of the behaviour of inventories in relation to different decision strategies and characteristics of managers;
- Evaluating the sensitivity of the supply chain to different methods of forecasting;
- Developing a learning environment to enable managers to conduct what-if scenarios and explore the behaviour of the supply chain.

The novelty of the research lies in articulating and applying a synergy of theories to the modelling and analysis of the supply chain, i.e. system dynamics (as an overarching framework), control theory and non-linear and forecast analysis. The application of these theories in practice is validated and evaluated within a real life context at Draeger Safety UK. It is envisaged (a hypothesis is made) that the combined use of system dynamics, control theory and non-linear and forecast analysis will enhance the quality and depth of study of the systems behaviour and will lead to new insights. A brief outline of the modelling and simulation of the supply chain with the application of the suggested theories is given in the following section. The uses and insights of the applied theories are explored and evaluated throughout the thesis.

1.3 Modelling and Simulation of Decision Making in the Supply Chain

After reviewing more than eighty scientific papers on the role and application of simulation in the supply chain, Terzi and Cavalieri (2004) show how simulation is

successfully adopted in different studies of logistic networks. Their findings ascertain that simulation allows evaluation of performance prior to the implementation of the system since it: 1) enables companies to perform powerful what-if analyses leading to learning and improved planning decisions; 2) permits the comparison of alternative strategies in a safe virtual world, without interrupting the real system; 3) permits time compression that allows for timely policy decisions to be made. These findings are also supported by Chang and Makatsoris (2001).

The suitability of simulation as an approach to studying supply chains does not guarantee its adequate decision support (van der Zee, 2005) as simulation relies on a heuristic search for good decision strategies which are led by people. The success of the simulation study depends on the joint work of the analyst and the chain managers and actors, as well as the facilities offered by the modelling and simulation tools. Van der Zee (2005), Sterman (2000) and Senge (1996) suggest that it is important to consider the simulation model as a communicative tool between analysts and chain actors. system dynamics (Forrester, 1961; Sterman 2000) as a modelling and simulation framework has been developed for such a purpose.

This research uses a combination of system dynamics and control theory analyses to address the issues of decision support and learning in the supply chain. It acknowledges the insights of non-linear dynamics and chaos theory in exploring the system's behaviour (Sice & French, 1998; Mosekilde, 2002).

When used in a management context, Systems Dynamics is a framework for thinking about how the operating policies of a company and its customers, competitors and suppliers interact to shape the companies performance over time (Forrester, 1961). It is an approach that allows the explicit modelling of perceived delays, bounded rationality and goal-setting and explicitly to discuss systems'boundaries (Ackermans and Dellaert, 2005). To do this, system dynamics uses metaphors drawn from control theory to provide tools for mapping and simulation, and metaphors from behavioural decision theory to describe decision making processes (Morecroft, 1985, 2007).

Systems dynamics is a powerful framework for thinking and explaining how the structure of systems generates behaviour. Ortega and Lin (2004), after reviewing the critical literature on System dynamics, conclude that this approach is suitable for

modelling complex systems but, when it comes to analysis, it contains insufficient analytical support. If an in-depth mathematical analysis of behaviour needs to be carried out, the appropriate mathematical tools from control theory need to be introduced (Ackermans and Dellaert, 2005). System dynamics was founded in control theory and, thus, is compatible with its tools and techniques (Forester, 1961; Roberts, 1978; Sterman, 2001). Ortega and Lin (2005), after studying the use of control theory in supply chain analyses, suggest that control theory analytical tools are quite suitable. In this research, block diagrams and stability in discrete time systems and eigenvalue analysis have been adopted (Atherton D.P., 1982; Richard R.J., 1993; Jacobs O.L.R., 1993). However, in the supply chain there are asymmetric consequences, for example, negative and positive inventories are asymmetric and thus introduce non-linearity in the system. Recent research suggests that non-linearity in the supply chains may lead to chaotic behaviour (Thomsen *et al/.*, 1992; Mosekilde, 2002; Sice, French *et al/.*, 2000). Control theory is generally linear in nature, and in order to consider effective application in supply chain analysis, it needs to be complemented by non-linear dynamics analyses (Ortega and Lin, 2005; Sice, French *et al/.*, 2000).

Forecasting plays an important role in the decision making process of the supply chains. Typically, sales forecasts are quantitative forecasts based on historical data plus managerial judgment within a forecasting support system (Robert, 2005). The research included in this thesis concentrates on the quantitative forecasting of the market sales series.

Traditionally, the short-term load forecast employs statistical models to analyze and predict the time series. The Autoregressive and Moving Average (*ARMA*) model is one of the most popular dynamic load shape models (AD, 1990). However, linear forecasting techniques have limited ability to capture nonlinearity in the series. Artificial intelligence methods for forecasting have received considerable attention as means to deal with nonlinearity and other difficulties in modelling of the time series (Grundnitski & Osburn, 1993). Artificial neural networks have been widely applied in the area of time series forecasting owing to their flexible data modelling (Ping-Feng Pai, 2006, Chiang, 1996). Combining neural network with wavelet analysis allows the approximating of the time series at different levels of resolution using multiresolution wavelet decomposition (Bai-Ling, 2001). In subsequent chapters, the neural wavelet

forecasting model is built and applied to the forecasting of the Draeger Safety UK sales series. The forecasting performance of *ARMA* model and wavelet neural network model are examined and compared.

Traditionally, system dynamics models of the supply chain focus on identifying the structure that generates behaviour in relation to inventory levels and the supply line (Sterman, 2000). In this research, it is also considered important to evaluate the sensitivity of different methods of forecasting to the performance of the supply chain.

1.4 Outline of the Thesis

An overview of this thesis structure is shown below:

Chapter One examines the current challenges to supply chain management and introduces the background and focus of the research.

Chapter Two explores the current approaches to modelling and simulation of the supply chain. It evaluates existing system dynamics and the contribution of control theory to model development and analysis. The amplification and uncertainty in the multistage supply chain that lead to complex behaviour, are discussed and the decision heuristics introduced. The research approach and the modelling process are developed.

Chapter Three describes the Draeger Safety UK case and develops the simplified model of Draeger centralised supply chain. Dynamic analysis of the behaviour of inventories and backlogs are performed, and the sensitivity of the decision parameters is tested.

Chapter Four introduces an additional two month manufacturing lead time and production constraint to simulate the effect of possible production hold-ups. The primary route to instability in the factory is identified and further explored.

Chapter Five presents the effect of an additional information delay between the hub and the factory. The primary route to instability in the hub is identified and further explored.

Chapter Six uses the *ARMA* (autocorrelation and moving average) model to analyse and forecast Draeger sales. The forecast error is estimated as the absolute difference between forecast and real data.

Chapter Seven applies wavelet neural networks to sales forecasting. This allows the taking into account of the nonlinear characteristics built in the series. The performance of different forecasting techniques is compared.

Chapter Eight develops Management Microworld to explore 'what-if' scenarios in decision strategies. The performance of different forecast strategies is measured as inventory and backlog costs.

Chapter Nine reflects the modelling process and outlines the major findings of the research. It develops recommendations for further investigation.

Chapter 2:

Supply Chain Management and Uncertainty in the Supply Chain

The majority of current single stage production inventory models or multi-stage supply chain systems are based on steady state conditions analysis (Suri and Desiraju 1997). However, the present business situation is quite different. Companies need to respond quickly to changes in customer demand and adapt to changes in technology. This requires a modelling approach that takes into account the dynamic behaviour of the supply chain and the factors influencing the dynamics of its performance.

Recently, such an approach has been developed and applied within the fields of control theory and system dynamics. Existing generic models from these traditions are reviewed in this chapter. The key elements of supply chain management are also examined, including the decision making heuristics. The final section explains the epistemological assumptions and the modelling process of the research approach.

2.1 The Key Components of Supply Chain Management

A supply chain is defined thus by Stevens (Stevens, 1989) as "A supply chain is a system whose constituent parts include material suppliers, production facilities, distribution services and customers linked together via a feedforward flow of materials and the feedback flow of information". The definition emphases are on the material and information flow between each element of the supply chain. Complex interactions occur inside of the supply chain system. The management of the supply chain is

accomplished through logistics. The British Standard Institute defines logistics thus, "Logistics is the planning, execution and control of the movement of people, goods and related support in order to achieve an objective within a system" (British Standards Institute, 1997). The application of logistics optimises the material and information flows and positioning of goods. Jones and Riley (1985) suggest three fundamental elements in supply chain management: first, recognising customer service level requirement; second, defining the position of the inventory along the supply chain and the quantity of the stock at each point; third, developing the appropriate policies for managing the supply chain as a single entity. Jones (1989) argues that by focusing on the end customers' requirement, the improvements made on each echelon will improve the overall performance of the supply chain. Achieving "supply chain synergy" benefits the management of the supply chain as a whole rather than the individual echelon. In summary, supply chain management is characterised by focusing on the end customer and the integration of systems, policies and inventories within the supply chain.

2.2 Control Theory Applications to the Single Stage Production Inventory Model

Control theory is applied to reduce inventory variation, reduce demand amplification and optimize ordering rules. Control tools such as the block diagram, algebra, Mason's gain formula, Laplace transform, Z transform and optimal control have been implemented in the analysis of supply chain dynamics and modelling.

Wikner (1994) has presented a rigorous implementation of control theory applied to the one stage supply chain model. He considered three main activities in the single stage manufacturing system: forecasting, lead times, and inventory replenishment rules. The Laplace transform of the first order exponential smoothing forecasting is adopted. The lead time and inventory replenishment rules are represented in the same way. The representation of Wikner's system is shown in Figure 2.1.

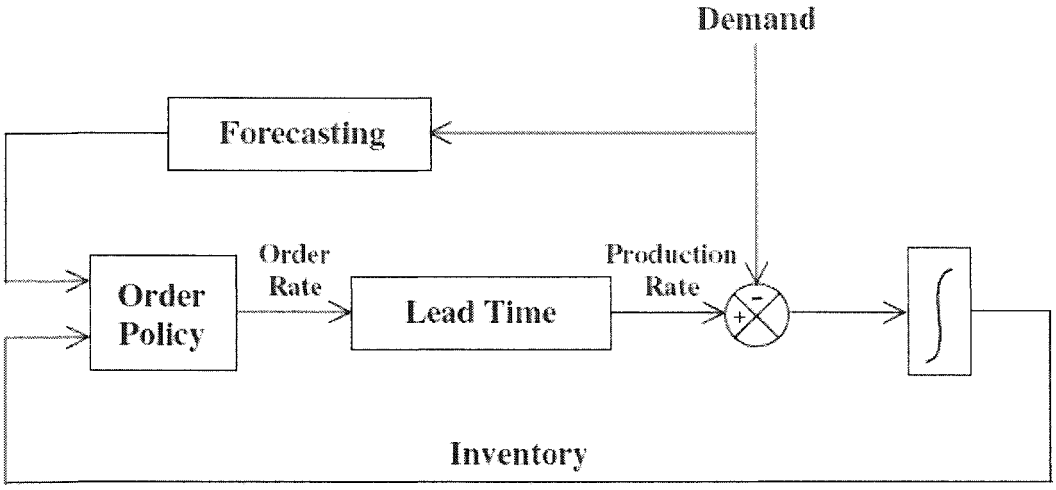


Figure 2.1 *Block diagram representation of a production inventory system (Wikner, 1994)*

According to the simplified block diagram in Figure 2.1, the general format for the order policy transfer function adopted is written as:

$$Order\ policy = U(s) = \left(K_p + K_D s + \frac{K_I}{s} \right) \tag{2.1}$$

Wikner identifies U(s) in equation (2.1) as a proportional, integrative and derivative (PID) controller. The proportional control K_p adjusts the inventory. The derivative controller K_D adjusts the rate of change of the inventory and the deviation of the production rate from the sales rate.

The order decision rules are based on three fundamental information flows: the demand, the current inventory level and the product work in progress. These information flows are classified as feedforward and feedback in control systems. In general, the inventory level and the products work in progress are transmitted as feedback information, whereas the demand has a feedforward character (Grubbstrom 1996). Figure 2.1 shows the feedback information flow from integral to inventory and back to order policy. It carries the information about the effect of previous decisions, coming from within the system, while feedforward conveys information about what to expect in the future, coming from outside the system. It can be concluded that feedforward is preventive and feedback is remedial.

The PID control algorithms are also applied by White (1999) to a simple inventory management with more detail. He illustrates that the PID control could reduce stock levels by 80 per cent. The block diagram of his system is shown in Figure 2.2.

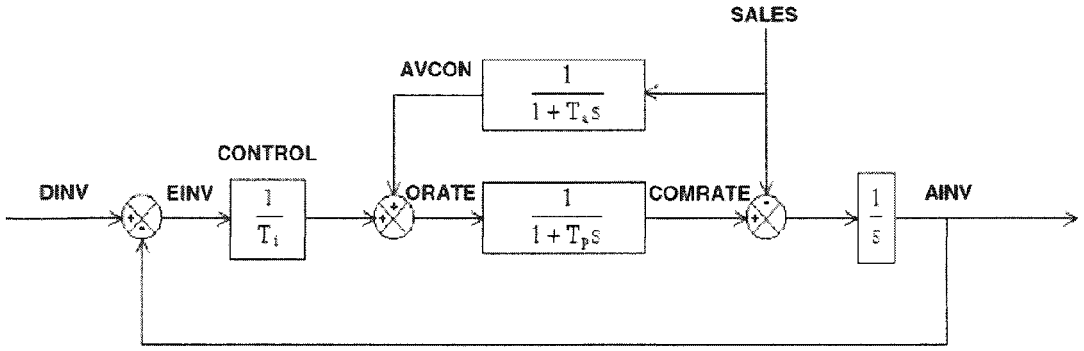


Figure 2.2 *Inventory and order based production control block diagram (White, 1999)*

In the model, the inventory error (EINV) is defined as the difference between the desired inventory level (DINV) and the actual inventory (AINV). Exponential smoothing is used to obtain the forecasting (FORES) as a function of the sales. The forecasts are used to obtain the order rate (ORATE). There is a production delay inherent in the manufacturing process between order rate and completion production rate (COMRATE). It was defined in exponential smoothing in White's model. Sales are regarded as a disturbance with feedforward action in the control system. By using Mason's gain formula, the transfer function between sales and actual inventory is:

$$\frac{AINV}{SALES} = -T_i \left[\frac{T_a T_p s^2 + (T_a + T_p)s}{(T_a s + 1)(T_a T_p s^2 + T_i s + 1)} \right] \qquad (2.2)$$

Equation (2.2) involves three parameters T_a, T_i and T_p. T_a is the smoothing constant used in the exponential smoothing forecasting technique. T_a produces a forecast that 'damps' the effect of demand changes. With regard to T_p, Wikner *et al/.*. (1992) noted that the inverse of T_p represents how fast the production unit adapts to changes in the order rate. In the same literature, the inverse of T_i represents the fraction ordered of the actual inventory deviation from the target level. With equation (2.2), the author quantifies the reduction in stock level achieved with the application of PID.

These models take into account a single stage of supply chain. They represent the process of supply chain management as a set of forecasting activities. Lead time and inventory replenishment rules are described objectively.

Further work builds on this generic model to include multi nodes supply chains and use of decision making heuristics derived from the specific characteristics of the managers. This work is described in the following section.

2.3 Decision Making and Amplification and Uncertainty in the Multistage Supply Chain

Collaboration between the sectors in the supply chain system has been the major field of work in global logistics. Beamon (1998) defined a supply chain as an integrated process where various business entities, such as suppliers, manufactures, distributors and retailers work together. The suppliers provide the raw material to the factory. The factory converts the material into the final product and delivers it to the distributor. The basic character of the multi-sector supply chain is the forward flow of material and feedback flow of planning information.

The most frequent and harmful phenomenon in supply chains is upstream order amplification. This effect is called the 'Bullwhip Effect'. It was defined by Lee (1997) as the phenomenon of orders to suppliers tending to have larger variance than sales to buyers. While customer demand does not vary much, the inventory and backlog levels fluctuate considerably across the supply chain.

The work which forms the foundation of most current research about the amplification and uncertainty of supply chain management is Forrester's original *Industrial Dynamics* (Forrester, 1961). He analyses the behaviour of a four echelon supply chain consisting of retailer, distributor, wholesalers, and factory. His approach is built on the principles of control theory as described in the previous section; however, he introduces terminology suitable for managers. His approach to modeling a feedback system is referred to as system dynamics. Forrester defines a feedback system thus, "An information-feedback system exists when the environment leads to a decision that results in action which affects the environment and thereby influences future decisions"

(Forrester, 1961). There are three key characteristics of an information feedback system: the structure of the system, the delay in the information or material flows, and amplification in the entity of the system. The structure shows the general relationship of different parts. Delay always exists between the available information and the decision making. Both system structure and delay could lead to the amplification of a certain entity in the system. All the effects combine to determine the behaviour of the system.

The information required to create Forrester's supply chain model with feedback information includes organisational structure, delays in decisions and material flow and policy on compiling orders and inventories. The orders compiling principles focuse on: the order to replace goods sold, the order to adjust inventories, and the order to fill supply pipelines with in-process orders and shipments. The sales order goes through each echelon of the supply chain. The retailer receives a sales order and then, after a sales analysis and information lag, orders are placed on the next level echelon. The order placed on the supplier level includes items compensating for sales consumption on the inventory level. A proportion of orders accounts for the items in process; for example, the orders in transportation and backlogs in the suppliers. Forrester's work demonstrates how various types of business policy create disturbances. It has been developed into a management training known as the Beer Game.

Beer Game

The 'Beer Game' is a simplified realization a brewerie's production-distribution system, based on Forrester's model. To allow role-playing experiments to be performed, only a single inventory (beer) is considered. It has been developed by MIT and has been used since the 1960s. Thousands of students of management and experienced managers from major US companies have been asked to operate the four-stage production-distribution beer game consisting of a factory, a distributor, a wholesaler and a retailer (Jarmain 1963). Each stage holds an inventory. Once the retailer receives a demand, it ships the required number of units out of its local inventory. The retailer then subsequently orders beer from the next stage, the wholesaler, who ships out beer on request and submits orders to the distributor. The order is passed through the supply chain to the factory. The object of the game is to

minimize stock holding costs across the entire chain by keeping invetories as low as possible while avoiding out of stock conditions.

The game is started in an equilibrium situation. The order placed is equal to the incoming shipment. The inventories in each level remain stable. The demand for beer remains constant in the first four weeks to allow the participants to become familiar with simulation procedures. From week five, the demand doubled, the participants are required to react to the change in demand.

Sterman analysed the rules used by players and built simulations based on the heuristics used by the players (Sterman, 1989a; Sterman, 1989b). He points out that by virtue of the built in delays and nonlinear constraints, many players found that they were unable to secure a stable operation of the system and that, consequently, large- scale fluctuations developed (Mosekilde,Sterman, 1992). Some typical behaviours have been witnessed in the simulation; large amplitude oscillations in inventories, and the steady amplification of the order placed on the next level supplier from retailer to factory. The players generally try to keep the stock in the initial level. Most of them do not take account of the supply line, which consists of the next level backlog and out shipment.

The players want to maintain the inventory at a particular level. However, the stock can not be changed directly; the manager can only change the stock by controlling the inflow and outflow rate. The inflow rate will compensate the loss of inventory for feeding the demand and recover the inventory to the desired level. Delay always exists between the change of inventory and the realization of the decision maker's action. Therefore, the decision rules within the supply chain focus on three key aspects. First, feeding the expected demand, i.e. forecast; second, reducing the discrepancy between the actual inventory level and desired level; third, maintaining an adequate supply line.

The 'Beer Game' was demonstrated to the Draeger planning staff. Given the complexity of the system, it is not surprising that the participants could not minimize costs. Orders and inventory were dominated by large fluctuations. The amplitude and variance of order rose steadily from retailer to factory. Moreover, these phenomena have also been observed in the results of previous research studies (Sterman, 1988a); and it has been concluded that the majority of players use a common anchoring-adjustment heuristic in placing orders. The mechanism for this anchoring-adjustment

heuristic is articulated by Sterman (1988a): the anchor is the expected demand, ED, the subjects forecast of expected orders. Replacing the incoming orders, ceteris paribus, maintains inventory at its current value. The asymmetric cost function means subjects should maintain their inventory at a non-negative value DS. Thus an adjustment to the anchor arises from the discrepancy between the desired stock DS and the actual stock I. Subjects may also consider a fraction β of the supply line of unfilled orders. The supply line SL is the accumulation of orders placed but not yet received. Finally, orders OP must be non-negative (order cancellations are not allowed). Thus,

$$OP_t = \text{Max}(0, ED_t + \alpha\ (DS_t - I_t - \beta SL_t)) \qquad (2.3)$$

where the stock adjustment parameter α (which lies in the range 0-1) represents the fraction of any discrepancy corrected between the desired stock (DS) and the effective inventory to be ordered each time period.

The adjustment term $\alpha\ (DS_t - I_t - \beta SL_t)$ creates two negative feedback loops which regulate the stock. Discrepancies between the desired and actual stock induce additional orders until the inventory reaches the desired value. Likewise, an insufficient quantity of goods on order induces additional orders until the pipeline is filled, depending on β (again in the range 0-1). It is the fraction of the supply line which is taken into account. If β=0, then orders placed are forgotten, causing overordering and instability. If β=1, then subjects fully recognize the supply line and do not double order.

The expected demand (ED) represents each participant's forecast of incoming orders.It forms the main motive or the anchor in the ordering heuristics. In the equilibrium state, the inventory, and the supply line are at their desired level. The supply chain managers need to order enough to cover future demand. Failure to secure a sufficient supply will cause the reduction of the inventory, which leads to limitation to cope with unexpected change in demand. Adaptive expectations are assumed:

$$ED_t = \theta \cdot IO_{t-1} + (1-\theta) \cdot ED_{t-1} \qquad (2.4)$$

where ED_t and ED_{t-1} are the expected demand at time t and t-1, respectively. IO is the incoming order. θ represents the rate at which the expectation is updated. θ=0

corresponds to a constant expectation, and θ=1 describes a situation where the immediately previous value of incoming orders is used as an estimate of future demand. In addition, participants are found to adjust orders above or below the expected demand in order to keep their inventory and supply line at the desired levels.

If combining the beer game ordering heuristics with control theory, the order policy with parameter α , β and θ is a proportional derivative (PD) controller which is very similar to Wikner's single stage supply chain. Wikner's model is continuous, while the beer game is a discrete time simulation.

The beer game exhibits complex behaviour. Mosekilde& Larsen (1988) investigated whether chaos could be generated from the decision making processes of the system. They demonstrate that stable, periodic, quasi periodic and chaos exist with different values of the planning coefficients in certain regions of the model. A further discussion of chaos in human decision making is carried out in their later research. Their results show that chaos can be produced by the decision making behaviour of real people in simple managerial systems (Mosekilde, Larsen, & Sterman, 1991). Therefore, these types of behaviour could occur in real industrial and management environments. Complex chaos system behaviour is observed when a low safety stock, little consideration of the supply line and rapid correction of discrepancies between the desired inventory and the actual inventory apply. This reflects the over ambitious manager's aggressive planning policy, neglecting the supply line and keeping low safety stock. Zhanybai and Erik (2003) also conclude that the instability observed in the Beer Game is connected with the delays in the system. For a significant increase in demand, the manager needs to wait to find out whether the change is permanent and hesitate to adjust the supply order to the full amount. On top of this, the delivery delay would also lead to a deduction in inventory and increase in backlog. Thus, the original demand has been amplified and delayed. Similar mechanisms can be identified in the real economic world. They are believed to play the major role in forming the general business cycle (Sterman and Mosekilde, 1994).

Sterman (1994) states that oscillations can arise only when there are time delays in at least one of the causal links in a negative feedback loop. The time delays between the correction actions and their effects create the supply line that has been initialised but

not yet had any impact. The managers continue to execute the correction action by perceiving the gap between the desired and the actual state of the model, even through the sufficient correction order they placed to close the gap is coming into the pipeline. The ignored or partially ignored time delay forms the origin of the oscillation. Previous research (Diehl and Sterman, 1995) also shows that the players' behaviour is quite systematic and explained well by the simple ordering heuristic. The tendency to ignore the time delays and the supply line is clear.

One of the key issues in the research is about the cost implications of the chaos. The experiments show that stable solutions are relatively lower cost while chaotic solutions lead to costs 500 times higher. In the management system, apparently random system behaviour could be produced by simple and deterministic structures. It is important to distinguish between internal processes and random external effects. The lack of understanding of the internal process makes the manager attribute unstable phenomena to external causes.

Other researchers, such as Wikner (1991), Jones and Towill (1996), Disney and Towill (2002) have also developed multi echelon supply chain models and make recommendations for improving performance through free information flow between all echelons. However, no suggestions are made about how to select the decision parameters to avoid system oscillation.

To study the influence of forecasting on supply chain performance requires quantitative analysis and detailed examination. Statistical models for time series analysis and forecasting, such as exponential smoothing and auto-regression and moving average (*ARMA*) (Box and Jenkins, 1976), have been widely applied in supply chain management. However many studies have demonstrated that the real financial time series varies in a highly nonlinear dynamic manner (Blank,1991). To deal with the embedded nonlinearity and increase the reliability of forecasting, neural networks have received considerable attention (Hsiao, 2006; Xie, Zhang and Ye, 2007; Grudnitski & Osburn, 1993). Neural networks provide universal function approximations to map nonlinearity, without making prior assumptions about data properties (Haykin,1999). Traditionally, system dynamics models of the supply chain focus on identifying the structure that generates behaviour in relation to inventory levels and the supply line

(Sterman, 2000). In this present research, it has also been considered important to evaluate the sensitivity of different methods of forecasting to the performance of the supply chain.

2.4 The Research Approach

It could be argued that the basic elements of any research approach include process and methods, theoretical perspective(s) and epistemology (Crotty, 1998). Thus, it is essential for any rigorous research attempt to clarify and explore the epistemological assumptions, which will underpin the use of methods and tools, and the design of the process of inquiry (in this case, the modelling process).

2.4.1 Epistemological Assumptions and Research Approach

The most basic cognitive operation we perform as observers is the operation of distinction. It is through the operation of distinction that the observer specifies a unity as an entity distinct from its background (Maturana & Varela, 1987). If the observer applies the operation of distinction recursively and, thus, distinguishes the components within the unity, he redefines it as a 'composite unity', i.e. a system. It is through our human way of being that we perceive the world in terms of systems. Thus, systems are epistemological qualities and not definitions of how things actually are or occur. Different observers perceive or describe systems, and, therefore, their boundary and their structure, differently. The observer has to be accounted for as part of any explanation. Nevertheless, systems have become the means by which we explore and describe the consistency of situational behaviour. Therefore, descriptions of system structures are useful tools and, if some form of agreement can be reached about 'what a system does' then it is possible to communicate about its structure and boundaries with greater (in relative terms) coherence (Winograd & Flores, 1986). Language needs to take account of a systemic vocabulary. The Systems Approach has proven its merits (Flood and Jackson, 1981; Wolstenholme, 1990; Mosekilde, 2002). It is not the purpose of this report to discuss them. What is important to reflect on is that developing systemic language to address a problem situation is likely to be beneficial.

The epistemological perspective outlined above suggests that it is important to consider different world views and perspectives of a situation considered problematic. Thus, the use of communication practices that will allow, at least transiently, the coexistence of different understandings (Bohm, 1999) is important in the process of enquiry (i.e. the modelling process).

In the context of Draeger Safety UK, the enquiry took place at a group level (the Planning Department). The researcher was part of the group. There is a felt need to explore and learn how human decision making could impact on the performance of the supply chain. For an enquiry to be effective, and thus lead to a valid analysis of the situation, it is required that it takes place as a collaborative and participative process, i.e. as action research (Shein, 1996). Action research is an iterative process where research leads to action and action leads to evaluation and further research (Baskerville, 1996). The details of implementation of the action research approach are embedded in the modelling process used by Systems Dynamics (Sterman, 2000).

2.4.2 The Modelling Process

This research follows the discipline of the modelling process of the system dynamics theorists (Sterman, 2000). However, the use and implementation of this process is compatible and complies with the epistemological perspectives outlined above. The outcome of the modelling process is an in-depth case study of the centralised supply-chain at Draeger Safety, UK. The findings of the case study will have some generic validity to the dynamics of the supply chain in other organisations.

Modelling does not take place in isolation. It is embedded in an organisation and in a social context. Before the modelling process begins the researcher must gain access to the organisation. Also, to be effective (enhance insight and learning) the modelling process must be focused on the client's[1] needs. Very often, clients are involved in organisational politics and this needs to be taken into consideration as there may be a danger that clients may choose to use the model as a tool to defend their own position or interests. Modellers have the responsibility to require their clients to justify their

[1] Clients are the people whose behaviour influences the problem situation and, thus, they may need to change (Sterman 2000).

opinions, ground these in data, and be willing to consider alternative views. Thus, the modeller has the responsibility for both rigour and integrity in the modelling process.

As argued in the epistemological perspectives outlined above, research into management systems inevitably involves and relies on human communication. This will require an approach that allows for organisational actors to be engaged as researchers in addressing and improving a problem situation. Thus, there is a need to deviate from the paradigm of action research conducted by a third party (the researcher) to participative action research, i.e. research conducted by the organisational actors themselves, with the researcher as part of the research team (Robson, 2002).

The modelling process requires an understanding of the causes and implications of a problem situation, and a commitment to learning and improving. Alternative views are important in developing understanding and challenging assumptions. Defensive routines may occur and this can be damaging to the success of the modelling project. Successful modelling requires that organisational actors (in this case, the clients) participate in the process as researchers, together with the modeller.

There is no prescription for successful modelling. However, successful modellers follow a disciplined process that involves the following steps (Figure 2.3): articulating the problem to be addressed, and the purpose of the model; formulating a dynamic hypothesis/theory about the causes of the problem; formulating a simulation model to test the dynamic theory; testing the model until satisfied it is suitable for purpose; designing and evaluating policies of improvement (Sterman, 2000). Modelling is iterative; in any modelling project, one will iterate through these steps many times. Modelling is an ongoing process of cycling between the virtual world of the model and the real world.

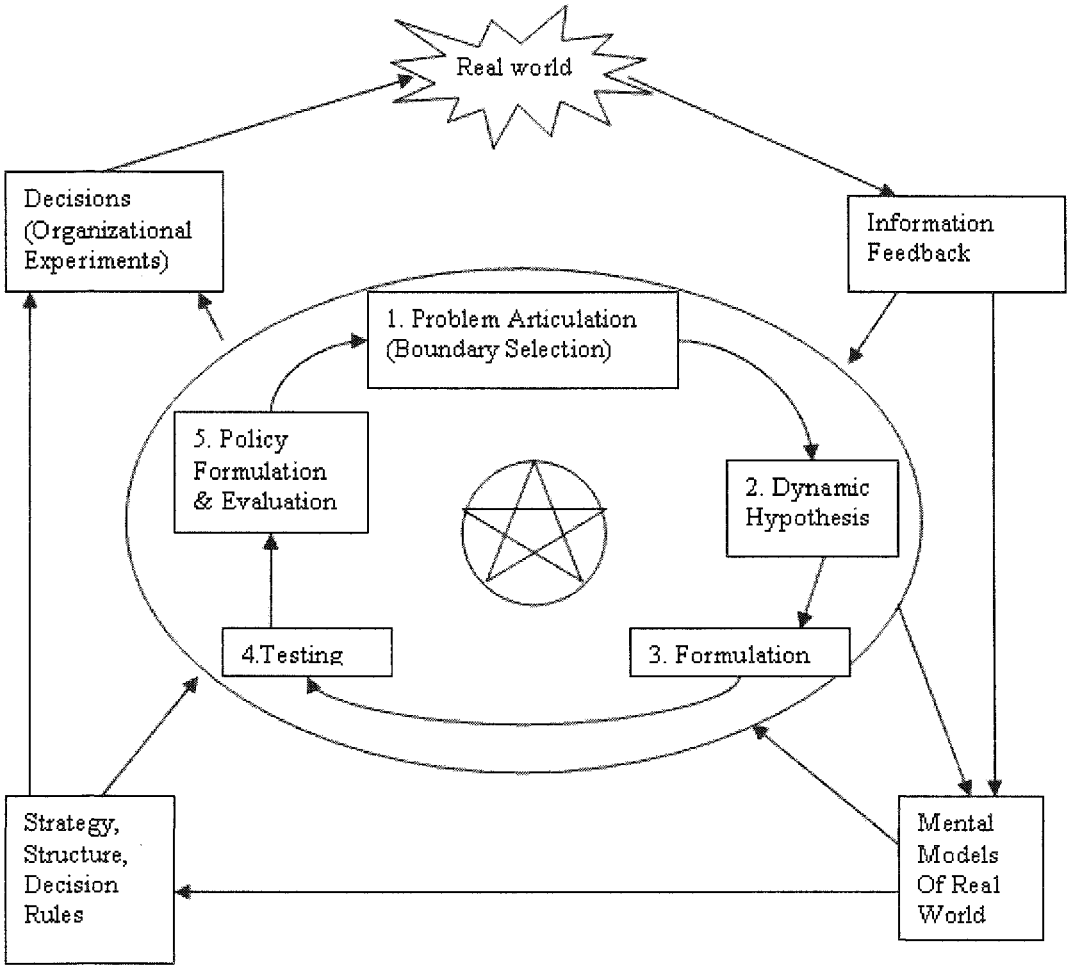


Figure 2.3 *The Modelling Process (adapted from Sterman, 2000, pp. 87)*

Articulating the problem to be addressed and the purpose of the model

Every model is a representation of a group of functionally interrelated elements, i.e. a system. However, for a model to be useful, it must address specific problem issues, this will allow simplification of the complexity of the entire situation and thus allow some meaningful analysis. The model will thereby act as a logical knife (Sterman, 2000; Morecroft, 1994; Morecroft & Sterman, 1994).

The methods that allow the eliciting of information within the research team (the clients and the modeller) include: interviews, discussion, observations, reflection on archival data (referred to as reference mode), and the time horizon[2] of the problem situation.

[2] Modellers must guard against accepting the client's initial assessment of time horizon. Often this is based on milestones, planning issues, etc. that have little to do with the dynamics of the problem. A longer time horizon is likely to reveal the full dynamics.

Formulating a dynamic hypothesis

Once the problem situation has been identified, the research team, and the modeller in particular, must begin to develop an explanation of the problematic behaviour over time, i.e. a dynamic hypothesis. A dynamic hypothesis is a working theory of how the problem arose from within. Each member of the research team is likely to have a different explanation (mental model). The discussion may be painful and damaging if they turn it into a field for defending one mental model against another. It is important that the modeller act as a thoughtful listener and facilitator and create conditions for dialogue (Bohm, 1999) rather than discussion, where the purpose of the enquiry and conversation is to explore different mental models and the assumptions behind them.

Causal loop diagrams, boundary charts, etc. can be used as facilitative tools to allow the team to develop a shared understanding of the problem issues and conceptual model of the problem situation (Sterman, 2000; Wolstenholme, 1990; De Geus, 1988).

Formulating a simulation model to test the dynamic theory

In order to test the dynamic implications of the conceptual model, normally difficult to assess through qualitative analysis only, it is important to develop a simulation model. Developing a simulation model and formalising the relationships between variables often generates important insights even before the full simulation model is ready. It is possible, and in fact desirable, to involve the research team in devising simple equations and links between variables and estimated parameters. However, the modeller should take responsibility for the rigour and integrity of the model.

Testing the model

Part of testing involves comparing simulated behaviour with actual behaviour. However, there is much more to testing than this. Testing begins as soon as the first equation has been written. Every variable corresponds to a meaningful concept in the real world. Equations are checked for dimensional consistency. Extreme conditions tests (conditions unlikely to occur but still possible), etc. reveal inconsistencies in understanding and measurement. The sensitivity of model behaviour is assessed.

Designing and evaluating policies of improvement

Once the research team has developed confidence in the structure and behaviour of the model, it can be used to design and evaluate policies of improvement. The robustness of policies and their sensitivity to uncertainties must be assessed. The interaction of different policies must also be considered. Because management systems are highly non-linear, the impact of combination policies is different from the sum of their impact alone.

The five step modelling process represents an iterative learning cycle. The effective implementation of this cycle requires the intimate involvement of the researcher in the day to day activities of the client in order to gain relevant experience in relation to the problem situation. It also requires communication practices such as dialogue and co-creative discussion (Senge, 1990), allowing for an understanding of the mental models and motivation of the clients. The validity of the model depends on these factors. Thus, equal attention should be paid to the process of building the model and to validating the outcome through comparison with the existing situation.

Chapter 3:

The Centralised Supply Chain at Draeger Safety Ltd

Like many other industries, Draeger makes use of a cascaded distribution system, with inventory holders at several different levels. Draeger's previous model consisted of sales and distribution units in 44 countries around the world, each with its own stock. In the reorganised structure, three main worldwide distribution hubs have been constructed to consolidate finished goods: Lubeck as the German hub, Pittsburgh as the USA hub and Singapore as the Aisa hub. In its simplest form (Figure 3.1), the primary hub, based in Germany, receives stock from the UK factory and ships this to the main EU markets and to the Pacific and US hubs. Each of the hubs acts as a business unit and maintains a warehouse. The hub receives orders from Draeger sales agencies linked to them. The sales agencies do not keep any inventories or receive shipments from the hub. The role of the agencies is to receive orders from the customer and make regional sales forecasts. The orders and forecasts are forwarded to their local hub. The hub ships the goods directly to the customer and communicates with the production units to help establish production requirements.

The basic structure of this supply chain is shown in Figure 3.1.

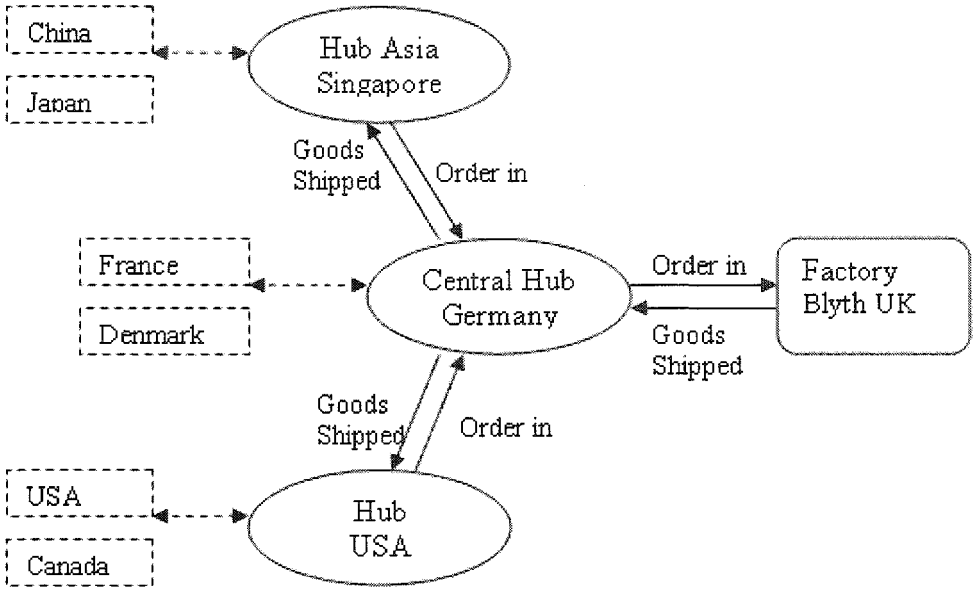


Figure 3.1 *Schematic representation of the centralised supply chain proposed by Draeger Safety Ltd.*

The process is based on a two month ahead forecast. That is, based on the current trends in orders and stock levels, the hub forecasts its requirements for two months ahead. This hub requirement is produced by the factory in the current month so that it can be shipped to the hub next month; in time to meet the expected demand. The relationships that exist between the Central hub and the Secondary hubs are similar to those that exist between the factory and the Central hub. The research was conducted at the Draeger factory in Blyth.

3.1 The Production Planning System at Draeger UK

Draeger uses the ERP software SAP as its enterprise resource planning system. This includes a Material Requirement Planning process in sales, production and inventory planning (this process is shown schematically in Figure 3.2) and a Master Planning System (MPS). The MPS has 4 primary inputs: the forecast, the current sales orders, the time fence and, the current stock.

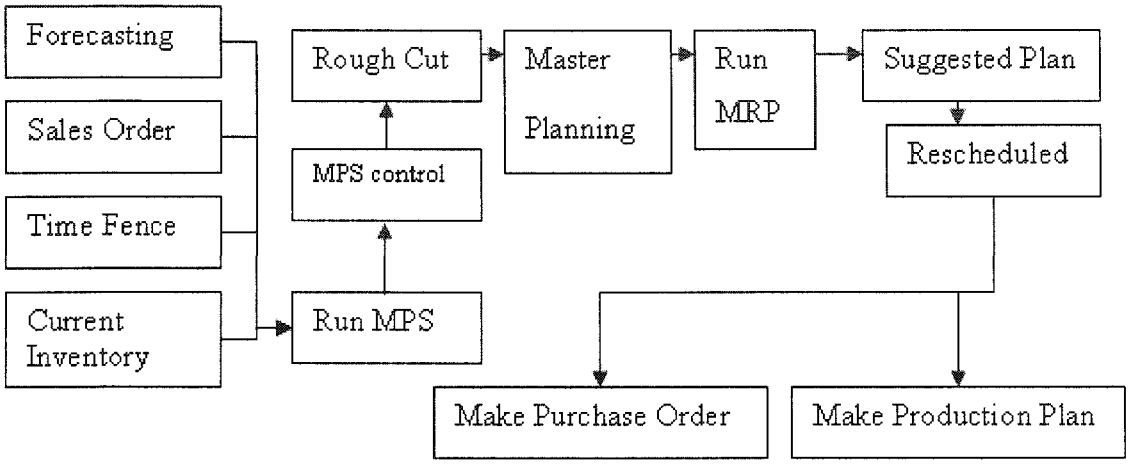


Figure 3.2 *Flow chart of the production planning process.*

3.1.1 Forecast

Product managers make monthly forecasts (based on a 15-month horizon) for the products that they are responsible for, using the GPS (Global Planning System). These monthly forecasts are then uploaded into SAP by the planning managers. The MPS is run to generate the suggested production plan for the coming month. The suggested plan is then subject to 'MPS control'. This is a form of manual override on the requirements planning process, since the MRP itself is considered to be a simple calculation engine. Thus, it is at this point that management influences the response of the production system to firm and anticipated demands. The tentative MPS results are then tested against capacity and a rough-cut capacity test is performed to check availability of resources. If successful, the planning manager can agree the master planning schedule for that particular month and run MRP to generate the detailed production plans and raw materials purchase orders for each product. Alternatively, the planning manager needs to adjust the MPS before running the MRP.

3.1.2 Safety Stock

Within the production/planning process, so-called 'safety stock' is used to smooth the impact of the variability of demand and to allow the manufacturing system to respond within some predefined lead time.

3.1.3 Time Fence

The time fence defines the minimum planning horizon for the MPS for a particular product. It is required because of material constraints in terms of material purchase and production lead times. Since component item orders must be completed before the parent item orders can be started, purchase and production lead times result in a cumulative lead time. Thus, product managers cannot change the forecast within the time fence. On the other hand, to satisfy some urgent (unexpected) demand, the product managers can decrease the master production schedules for some MPS items in order to create space or increase other master production schedules. 'Sometimes, some parts of the products are similar. The components from other part numbers can be used to compensate the stock requirement. All of the orders that cannot be satisfied will be put into the next time fence.'

3.1.4 Problem Situation

After a series of discussions with product managers Lynn Kennedy and Ian Bell, the planning manager Mel Hedley and the purchase manager Peter Kirby, some potential problems were identified within Draeger's supply chain: managers were needed to manually adjust the planned orders quite often. This, then, caused difficulties in the master planning and purchase planning schedules. The perception within the team was that the current forecasting mechanism could not meet the requirements of the sales orders.

The current software system used by Draeger for forecasting (GPS) is based purely on sales data history. Thus, product managers will often come up with different forecasts, based on additional information and their experience. Product managers forecast the sales each month; however, every night SAP downloads transaction data into the GPS. Thus, GPS could compare the actual sales with the forecast and cater for forecasting consumption. However, faults in the software lead to inaccurate information; thus, planning managers often need to intervene manually.

The company is equipped with a sophisticated MRP environment for sharing information and enhancing communication between production and marketing. However, the performance of the supply chain is still perceived as problematic as it is

not unusual that high inventories mount up and backlogs develop, causing disruption, as production hasa to deal with the crisis of overcoming these. A centralised supply chain system was recently implemented with the purpose of diminishing costs and avoiding backlogs. This made Draeger's planning managers even more worried as it was difficult to predict what the consequences of centralised inventories would be for the manufacturing plant in Blyth.

3.2 Developing the Dynamic Hypothesis

In order to investigate the problem situation, as perceived by Draeger, it was decided (in consultation with the planning department) to use a combination of system dynamics, control theory and non-linear analyses in order to model and explore the behaviour of the centralised supply chain at Draeger Safety, Blyth, UK. A team consisting of the planning department at Draeger Safety, UK and researchers from Northumbria University was formed to study the impact of different decision strategies on the behaviour of the supply chain. The boundaries of the model and the dynamic hypothesis emerged as a result of numerous discussions within the research team (primarily between the researcher and the planning manager, but also, within the wider Draeger team). The model includes the customer, the production plant and the hub in Germany. The purpose of the study was to model the material and information flows including the decision processes, to provide explanations and insights into the link between decision strategies and inventory and backlog oscillations, and to develop a simple microworld that managers could use to conduct what-if scenarios and learn about the behaviour of the supply chain.

A considerable body of research in supply chain dynamics has already been conducted at MIT (Massachusetts Institute of Technology). Indeed, The Beer Game (Sterman, 2000) has been used for decades to help managers understand the dynamic interactions that can influence the behaviour of these systems. Consequently, simplified equations have been developed, that reasonably represent the decision-making processes used by the thousands of participants that have played the game. Our work incorporates the generic decision heuristics of the beer game. However, the structure of the supply chain, and the information flows are modelled to represent Draeger Safety UK.

To better understand the operation of the supply chain, a simplified system dynamics model was constructed representing the relationships between the factory and the Central hub. The boundaries of the model were chosen based on discussions with Draeger Safety, UK staff. These determined that the model should include the customer, the production plant, the hub in Germany and the relevant decision making processes. The structure of this portion of the supply chain is illustrated in Figure 3.3.

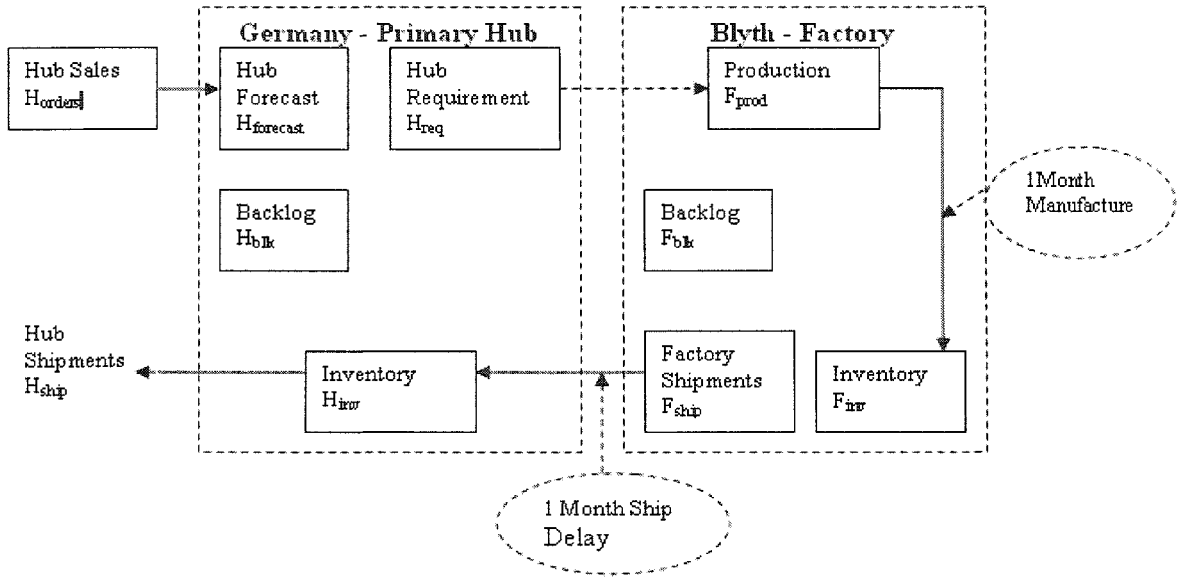


Figure 3.3 *Schematic diagram showing relationships between factory and central hub. The figure represents a 'way it's meant to work' model of the supply chain at Draeger Safety, UK.*

The figure represents a 'way it's meant to work' model of the supply chain at Draeger. However, interviews with the planning and production managers suggested that additional delays and constraints were often present. This was especially true in periods of high demand or when the forecast information was untimely or inaccurate.

Thus, the reality implies that there are at least three scenarios that should be considered, namely:

- 'the way it's meant to work';
- 'the way it's meant to work' plus additional manufacturing delays;
- 'the way it's meant to work' plus additional planning delays.

There are basically two parts in the 'meant to work' model. Those squares that appear in the top of Figure 3.3 are the planning parts, including 'Hub Forecast', 'Hub Requirement', and 'Factory Production'. The lower part of the diagram represents the warehouse and transporting parts. All the planning components work in one time unit. For example, when the customer places a sales order on the hub, it takes one single time unit (one month in the simulation) for the sales input into 'Hub Forecast'. At the same time the production plan is generated as the output from 'Factory Production'. A two month forward prediction strategy was applied in the planning process. Assuming that in the current month of January the hub received the hub order, the hub forecast would make a two month ahead forecast, which is the forecast of the sales in March, based on the incoming order in January and the forecast history. Still in January, the hub generates the 'Hub Requirement' for March and forwards these to the factory. The factory will make their production plan in January and start producing. Following the diagram in Figure 3.3, the work flow will then go through '1 month manufacture'. The production plan will be finished in February. The finished goods are kept in 'inventory' and waiting to be shipped in 'Factory shipment'. The shipping takes one month. It will then be March when the products arrive at hub inventory. The work flow is just in time. In Figure 3.3, all the solid arrow lines represent that a delay existed between the two squares. The broken line indicates that there is data flow between the two squares, without any time delay.

3.2.1 Causal Loop Diagram

The dynamic structure of the centralised supply chain is captured in the causal-loop diagram in Figure 3.4. In this case, the demand is considered as exogenous. The hub requirements are determined by the sales orders (demand) and also the hub effective inventory.

The causal loop diagram suggests numerous feedback loops. Some of the negative and positive feedback loops that are perceived to influence the dynamics are defined below.

Loop A. Hub Requirements-Production-Factory Effective Inventory-Factory Shipment Capacity - Factory Out Shipments- HUB Effective Inventory-HUB Inventory Gap-Hub Requirements, negative feedback loop

Loop B. HUB Requirements- Production-Factory Effective Inventory-Factory Shipment Capacity-Factory Out Shipments- Hub Requirements, negative feedback loop

Loop C. Production-Factory Effective Inventory-Factory Inventory Gap-Production, negative feedback loop

Loop D. Factory Effective Inventory (when negative, i.e. backlog)- HUB Requirements- Production - Factory Effective Inventory, negative feedback loop

Loop E. HUB Requirements - Factory Effective Inventory (when negative , i.e. backlog) – HUB requirements, negative feedback loop

Loop G. HUB Requirements- Desired Factory Shipments- Factory Out Shipments- HUB Requirements, negative feedback loop

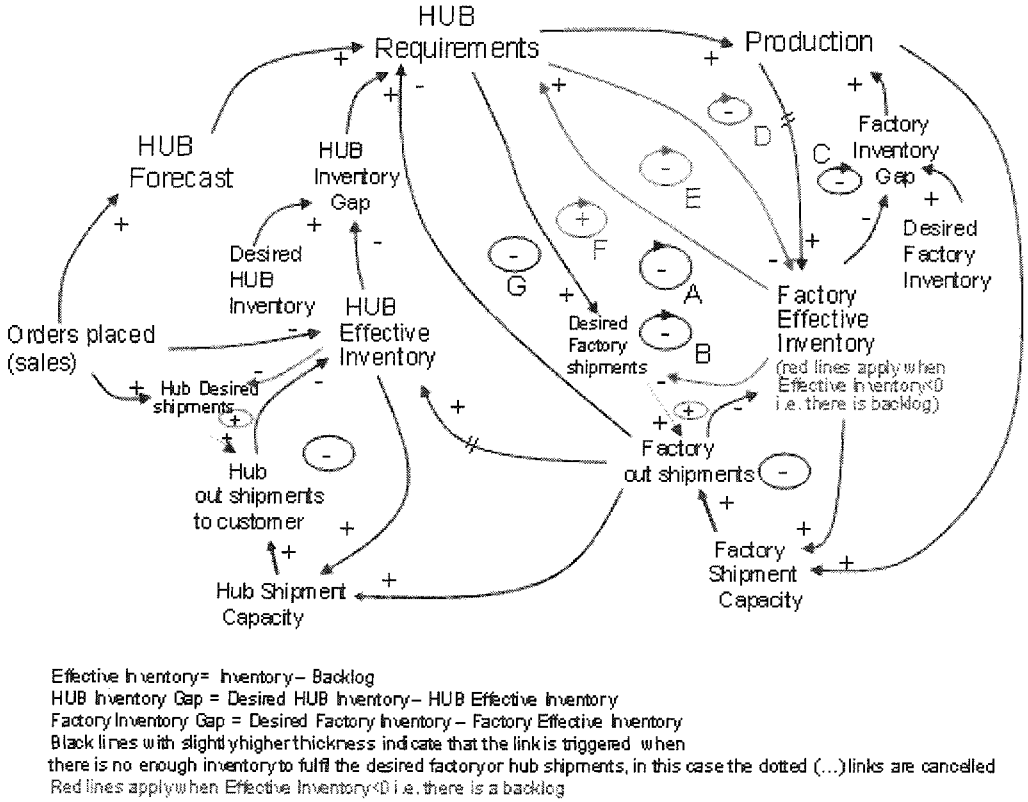


Figure 3.4 *Causal Loop diagram of the centralised supply-chain.*

As can be seen in the figure, management decisions are represented by the three corrective (negative) feedback loops. An initial estimate of factory production is made based on the forecast for the hub sales in two months' time. This estimate is then

corrected (feedback A) based on the discrepancy between the current hub inventory and its desired level multiplied by a 'correction factor', α. This modified estimate is then re-corrected (feedback B) to take into account any previously placed hub orders (based on previous attempts to correct the hub inventory discrepancy) that have not yet arrived and are thus still in the pipeline; in this case, the multiplying factor is β. The final correction to factory production (feedback C), is made based on the discrepancy between current factory inventory and its desired level; once again, this is multiplied by α. In this form, the 'management policy' appears to be, primarily, one of stock control in the presence of external disturbances (actual product sales), and the feedback structure that is in place bears a strong similarity to the classical 'cascade control' structure, which is familiar to many engineers.

3.2.2 The Draeger Model

After in-depth discussions with Draeger representatives, the qualitative relationship between different variables in the causal loop diagram was quantified. The equations that are used to describe the inventories, backlogs and shipments are discrete (using a 1 month sample interval) and based loosely on those of the Beer Game (Sterman 1989; Thomsen *et al/.*, 1992). The hub requirements, facotory production, inventories, backlogs and shipments which are included in loop A and B are formulated from equations (3.1) to (3.9). As in the causal loop diagram, the inventories are updated by adding incoming shipments and subtracting outgoing shipments.

$$H_{inv}(t) = \max(0, H_{inv}(t-1) + F_{ship}(t-1) - H_{ship}(t)) \quad (3.1)$$

$$F_{inv}(t) = \max(0, F_{inv}(t-1) + F_{prod}(t-1) - F_{ship}(t)) \quad (3.2)$$

To the extent that inventory plus incoming shipments are sufficient, outgoing shipments are existing backlog plus incoming orders. Otherwise, outgoing shipments are incoming shipments plus inventory, and the new inventory is empty.

$$H_{ship}(t) = \min(H_{orders}(t) + H_{blk}(t-1), H_{inv}(t-1) + F_{ship}(t-1)); \quad (3.3)$$

$$F_{ship}(t) = \min(H_{req}(t+1) + F_{blk}(t-1), F_{inv}(t-1) + F_{prod}(t-1)); \quad (3.4)$$

In a similar way, backlogs are updated by adding incoming orders and subtracting outgoing shipments. If incoming orders plus backlogs are completely covered by incoming shipments plus existing inventory, the new backlog is empty. From the negtive feedback loop D and E, the factory effective inventory has a negative link to the hub requiment and a positive link to the factory production. Since the factory effective inventory is negative when it becomes factory backlog, the equation (3.6) has a negative sign for $H_{req}(t+1)$ and $F_{prod}(t-1)$.

$$H_{blk}(t) = \max(0, H_{blk}(t-1) + H_{orders}(t) - (H_{inv}(t-1) + F_{ship}(t-1)); \quad (3.5)$$

$$F_{blk}(t) = \max(0, F_{blk}(t-1) + H_{req}(t+1) - (F_{inv}(t-1) + F_{prod}(t-1)); \quad (3.6)$$

The hub requirements and factory production are based on a simplified anchoring and adjustment heuristic (Thomsen *et al/.*, 1992). Expected demand is formed from incoming orders in an adaptive manner based on a simple first-order exponential prediction.

$$H_{forcast}(t+2) = (1 - \theta) H_{orders}(t) + \theta H_{forecast}(t+1) \quad (3.7)$$

Here, θ represents the rate at which the forecast is adapted. $\theta = 1$, corresponds to a constant forecast, and $\theta = 0$ describes a situation where the immediately previous value of hub requirements is used as an estimate of future demand.

Provision for this expected demand is taken to be the cornerstone of the order/production policy. Orders/Production is adjusted above or below this expected demand value to maintain current inventory and supply lines at their desired levels. In the causal loop diagram, the hub requirments are linked and defined by hub forecast, hub inventory gap, factory shipment, and factory effective inventory. On the other hand, the factory productions are linked and defined by factory inventory gap and hub requirements. Thus, the overall ordering policy (hub request & factory production) may be characterised by the expressions:

$$H_{req}(t+2) = \max(0, \alpha (Q - H_{inv}(t) + H_{blk}(t)) - \alpha \beta (F_{blk}(t) + F_{ship}(t)) + H_{forcast}(t+2)); \quad (3.8)$$

$$F_{prod}(t) = \max(0, \alpha (Q - F_{inv}(t) + F_{blk}(t)) + H_{req}(t+2)); \quad (3.9)$$

where, α, (which lies in the range 0-1) defines the fraction of the discrepancy between the desired inventory Q and the effective inventory (Inventory – Backlog) that needs to be ordered each month. β (which, again, lies in the range between 0 – 1) is the fraction of the supply line (requests or production already ordered but not yet received) that is taken into account. If $\beta = 1$, decision makers fully recognise shipments already in the pipeline and do not double order. If $\beta = 0$, requests (orders) are immediately forgotten and new requests to cover the same lack of inventory are placed in the next month.

During simulation, the initial value for hub and factory inventory, hub requirement, production, and shipment is 400 and the initial backlog is zero. Figure 3.5 is a Simulink block diagram representing the system equations. It is clear from Figure 3.5 that the system comprises four primary elements: These are, the production and hub dynamics (with their inherent saturations and constraints), the external disturbances or demands, and the management decision making policies. In the study presented here, the dynamics of the infrastructure (factory and hub) are assumed to be fixed and the analysis concentrates on an assessment of the influence of management decisions on overall system behaviour. The equations are programmed in Appendix A.

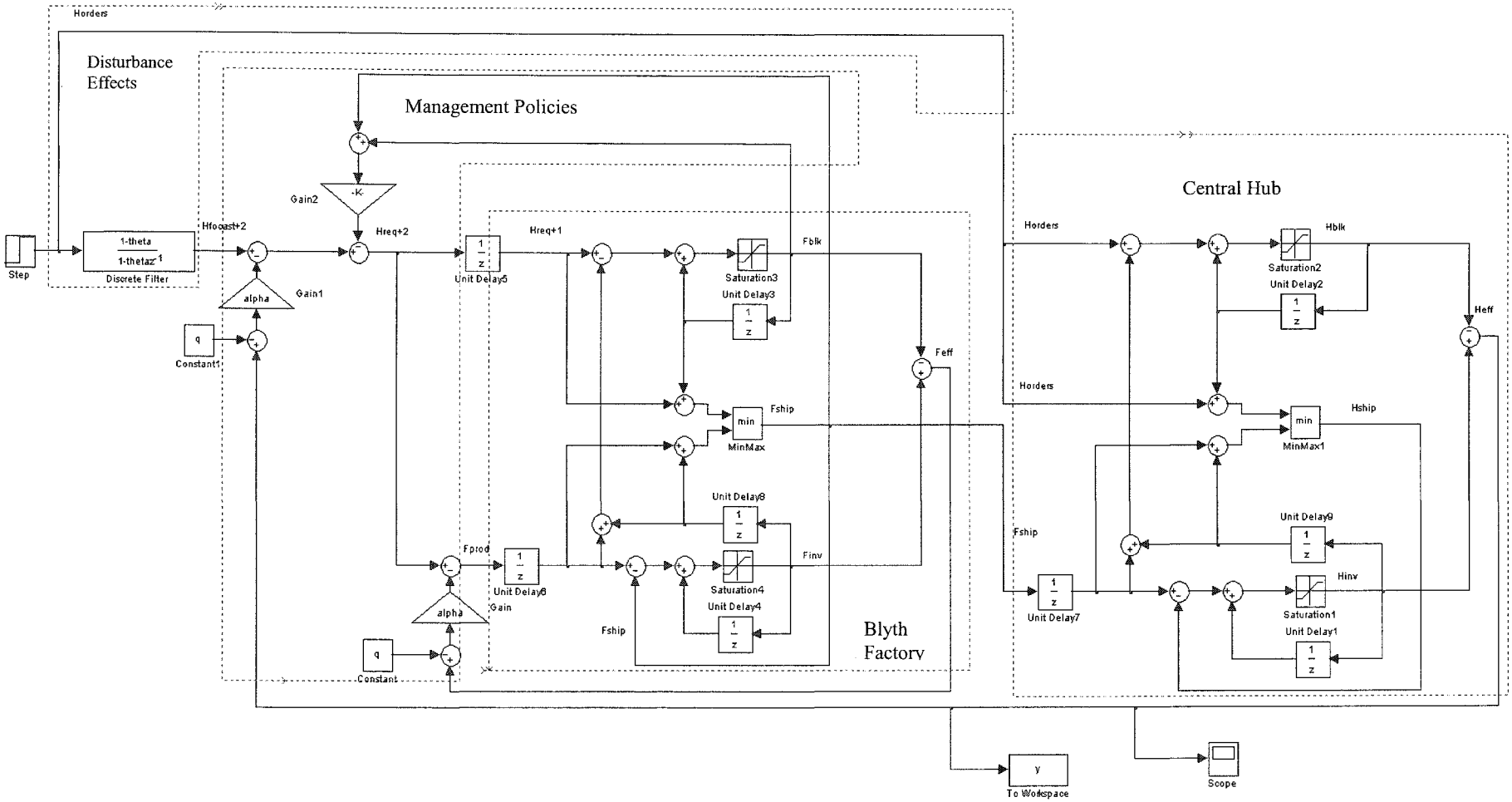


Figure 3.5 *Simulink diagram of the factory - hub model based on equations 1 to 9*

Model Testing

The model testing briefly includes model structure test and equation formulation tests. The boundary adequacy and model structure assessment are examined as the model structure test. After discussions with the project team in Draeger, the model boundary was agreed as including the factory in Blyth and central Europe hub in Germany. The finished model formulated these two business units' actions. The structure assessment checked whether the model consisted with and followed the physical realities and decision rules. Compared with the actual working process in Draeger, the model focused on the planning process, the detailed material requirement planning (MRP) process was out of model boundary. Throough the researcher's working with the planning manager, the model could be confirmed as representing the overall working process at Draeger.

Parameter verification and extreme conditions are the main part in equation formulation tests. The four parameters Q, θ, α and β have real life meaning. Each parameter represents the weight of the factors in decision making. In the simulation, θ was kept as constant most of the time. It was estimated from a previous study (Diehl and Sterman 1995) that most people form forecasts by smoothing the history of past orders. The model should be robust in extreme conditions. Extreme conditions were tested as equilibrium states and enormous sales input. In the equilibrium state, the sales inputs, forecast and production are at the same level. The inventory has no oscillation and is stable at the desired stock level. The enormous sales input was tested with and without production constraint in the model. In the case of no production constraint, factory production can always satisfy the hub demand. Therefore, the factory inventory is stable while the hub inventory oscillates, depending on the planning parameter. If there is a constraint in the production, when a big sales order comes in, the backlog order will accumulate. Detailed figures are shown in further sections.

The model reflects the project team's mental model. However, perfect validation or verification of the model is impossible.

3.3 Hub Analysis

A brief inspection of Equations 3.1 to 3.9 indicates that the primary nonlinearities present in the model are of a saturation form. Because of this, the initial analysis of the model is based on a locally linear representation in which all backlogs are assumed to be zero. Such an analysis will allow the soundness (or otherwise) of the basic feedback (or management) structure to be assessed and will also provide an insight into the various transitions towards instability, without this information being lost in the added complications introduced by nonlinear behaviour. The eigenvalues plot in z plane and block diagram analysis have been applied to the exploration of the hub behaviour (D.P. Atherton, 1982; R.J. Richards, 1993; O.L.R. Jacobs, 1993). Thus, for a constant hub order there is a fixed point at:

$$F_{ship} = F_{prod} = H_{req} = H_{forecast} = H_{ship} = H_{orders} \tag{3.10}$$

$$H_{inv} = Q - \beta H_{orders} \tag{3.11}$$

$$F_{inv} = Q; \tag{3.12}$$

Defining the general discrete state space representation as:

$$X(k) = A\,X(k-1) + B\,U(k) \tag{3.13}$$

$$Y(k) = C\,X(k) \tag{3.14}$$

the state vector X(k) as:

$$X(k) = \begin{bmatrix} H_{ship}(t) \\ H_{inv}(t) \\ H_{req}(t+2) \\ H_{forcast}(t+2) \\ F_{ship}(t) \\ F_{inv}(t) \\ F_{prod}(t) \end{bmatrix} \tag{3.15}$$

and the input U(k) as:

$$U(k) = \begin{bmatrix} H_{orders}(t) \\ Q_H(t) \\ Q_F(t) \end{bmatrix} \tag{3.16}$$

then the system matrices with zero backlogs can be written as:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -\alpha_H & -\alpha_H\beta_H & \theta & -\alpha_H & 0 & 0 \\ 0 & 0 & 0 & \theta & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 \\ 0 & -\alpha_H & \alpha_F - \alpha_H\beta_H & \theta & -\alpha_H & -\alpha_F & -\alpha_F \end{bmatrix} \tag{3.17}$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1+\alpha_H-\theta & \alpha_H & 0 \\ 1-\theta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1+\alpha_H-\theta & \alpha_H & \alpha_F \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.18}$$

Considering the model is a discrete time system, the general state space equation is z transformed (Golten and Verwer, 1991). It becomes

$$X(z) = A\,X(z)z^{-1} + B\,U(z) \tag{3.19}$$

$$\frac{X}{U} = \frac{B \cdot z}{(zI - A)} \tag{3.20}$$

where 'I' is an identity matrix of dimension A. The inverse of a matrix is obtained by dividing the adjoint by the determinant:

$$[zI - A]^{-1} = \frac{adj(zI - A)}{|zI - A|} \tag{3.21}$$

Therefore, the denominator of the transferred function is determined by the determinant of the matrix [zI - A]. The system characteristic equation is obtained by equating this determinant to zero

$$|zI - A| = 0 \tag{3.22}$$

The transient characteristics of the system are thus determined solely by matrix A in the state space equation. The roots of the characteristic equation in this form are the eigenvalues of matrix A. An n^{th} order characteristic equation has n roots, each giving rise to a term in the transient response. The total transient is thus the sum of these terms. In this model, seven eigenvalues are produced in total. The eigenvalues of the characteristic equation will either be real or complex.

$$\lambda = a + bi \tag{3.23}$$

Each root of the discrete time system characteristic equation at $z = \lambda$, gives rise to a transient response term of the form

$$\tilde{y}(j) = C\lambda^j \tag{3.24}$$

where j is discrete time and C is an arbitrary constant. When λ is real, with $\lambda > 1$ the terms in equation (number) become successively larger with increasing 'j' and so the transient increases exponentially without limit. The eigenvalues in this region correspond to an unstable system. When $-1 < \lambda < 1$, the transient is decaying as the increase of 'j' and finally arrive at constant value. These eigenvalues correspond to a stable system. With $\lambda < -1$, the transient no longer dies away but increases in amplitude with time; again, corresponding to an unstable system.

Considering λ as a complex number, with Euler formula the complex eigenvalue can be expressed as a point located at a radius, r, and an angle θ in a complex plane.

$$\lambda = re^{i\theta} \tag{3.25}$$

where $r = \sqrt{a^2 + b^2}$ and $\tan(\theta) = b / a$. Then the transient term can be written as

$$\tilde{y}(j) = C(re^{i\theta})^j \tag{3.26}$$

The exponential terms give rise to a sinusoid of frequency θ rad per sample. The constant C produces a constant phase shift ϕ.

$$\tilde{y}(j) = P \cdot r^{j} \cos(j\theta + \phi) \tag{3.27}$$

The transient response is, therefore, a sinusoid of frequency θ rad per sample which decays, or increases, at a rate governed by the radius r. P is a constant. In order to develop an understanding of relating the roots of the characteristic equation to the transient, a graphical representation of the roots is appropriate. The roots can be plotted on a complex plane, which is z plane in this research. The axes of the z plane are the real and imaginary parts of the eigenvalues.

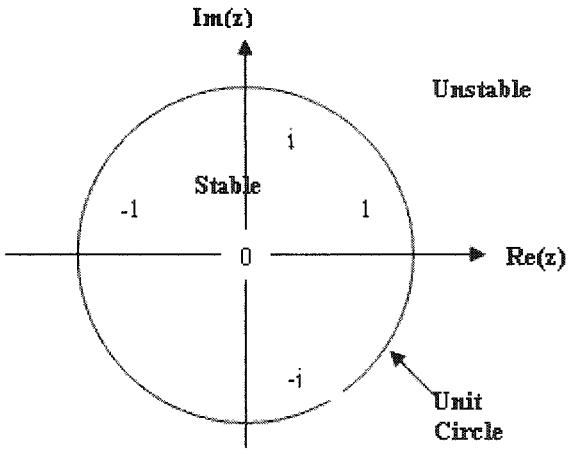


Figure 3.6 *The stability of the z plane*

The oscillatory transient produced by the complex eigenvalues decays in exactly the same way as for the real eigenvalues at radius $r = \lambda$. The eigenvalues at a radius of less than 1.0 will produce a decaying (stable) transient, whereas a radius of greater than 1.0 produces an increasing (unstable) response. The region of stability on the z plane is thus inside a circle of unit diameter (Figure 3.6). For a discrete time system to be stable, all the system poles must lie inside the unit circle $|r| < 1.0$. The eigenvalues which appear on the unit circle correspond to a marginally stable system. The system will oscillate continuously with a frequency of θ radian per sample, where θ is the angle of the eigenvalue around the unit circle.

In analysing system behaviour, two performance related parameters are generally explored, the natural frequency, w_n, which relates to the speed of oscillation and the

damping ratio, ς, which is a dimensionless representation of the amount of damping in the system. Given an eigenvalue $\lambda = re^{i\theta}$, the damping ratio λ and natural frequency w_n can be calculated as

$$r = e^{-\varsigma w_n T} \tag{3.28}$$

$$\theta = w_n \sqrt{1-\varsigma^2}\, T \tag{3.29}$$

where T is the time between each sample of the discrete time system. It is accounted as plus one in the calculation. Lines of the damping factor (ς) and natural frequency (w_n) are drawn within the unit circle in Z plane. These spiral contours with different ς values are called Jury contours (Golten and Verwer, 1991). Jury contours are useful in that they give an idea of the decay rate of the transient. By inspecting the position of a system eigenvalue and the nearest Jury contours, the damping factors can be predicted and the corresponding decaying ratio of the system.

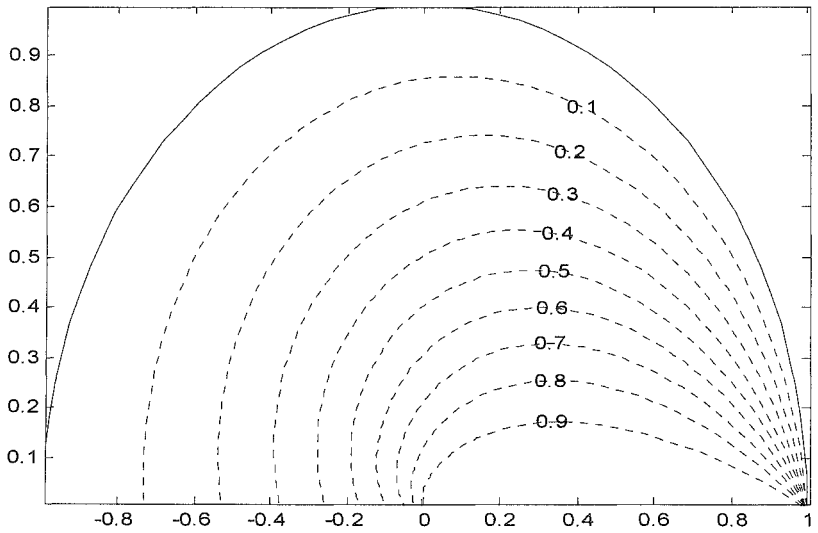


Figure 3.7 *Jury contours- lines of constant damping ratio on z plane*

As shown in Figure 3.7, the ς value on the spiral decreased from 0.9 to 0.1 from the inside of the circle to the edge of the circle. When ς = 0, the system has no damping. Based on the spiral in the figure, the related eigenvalue locates on the unit circle. Then

the transient of the system is a continiuous sinusoid of frequency w_n which neither increases nor decays with time. The natural frequency, w_n, is the frequency at which the system oscillates with no damping. When $0<\varsigma<1.0$, which are those spirals shown in the circle, the eigenvalue locates in the unit circle. The system will oscillate with a damped natural frequency w_d which is always lower than the natural frequency, w_n; the difference becoming more pronounced with larger ς. With $\varsigma = 1$, the system is critically damped. As seen in the figure, the eigenvalue moves to the real axis. There is no oscillation at all.

By applying ς, the percentage overshoot and the decay ratio of the oscillation can be calculated. For the oscillatory systems, the amount of oscillation can be specified by the percentage peak overshoot; it is defined as

$$\%\text{overshoot} = \frac{m_{k+1}}{m_k} \times 100 = \exp\left(\frac{\pi\varsigma}{\sqrt{1-\varsigma^2}}\right) = \exp\left(\frac{-\pi \ln(r)}{\theta}\right) \qquad (3.30)$$

where m_k is the absolute magnitude of any peak. The percentage overshot is the ratio of the wave peak at m_k and the previous wave peak at m_{k+1}. It can be expressed as the function of ς.The decay ratio is given by the ratio of two successive peaks of the oscillation, m_k and m_{k+2}. The ratio of any two peaks separated by 2 half periods of oscillation will thus be

$$\text{Decay ratio} = \frac{m_{k+2}}{m_k} = \exp\left(\frac{2\pi\varsigma}{\sqrt{1-\varsigma^2}}\right) = \exp\left(\frac{-2\pi \ln(r)}{\theta}\right) \qquad (3.31)$$

In linear system analysis, the relationship between behavioural modes and the location of the eigenvalues of A is well established. Consequently, it is clear that a plot of the trajectories of the eigenvalues in response to the variation of a specific parameter is a good indication of the influence that the parameter has on system behaviour. Inspection of equation 3.17 indicates that matrix A contains 3 variable parameters: α, β and θ. In studies of the Beer Game (Thomsen *et al*. 1992), it is generally accepted that for the majority of players (a large proportion of whom are managers) the value of θ tends to remain constant at 0.75. This was confirmed by findings at Draeger Safety, UK, where conversations with managers suggest that forecasts were amended with a

smoothing time of 4 months ($\theta = 0.75$). In addition, inspection of Figure 3.5 indicates that θ is presented only in the disturbance section of the model. Thus, θ only affects the dynamics of the feedforward (or disturbance) signal and can be considered to be outside of the feedback structure. α and β, on the other hand, clearly lie inside the feedback loops and, consequently, will significantly influence the dynamic behaviour generated by the system itself. For these reasons, in the study presented here, θ will remain fixed at 0.75 and the analysis will concentrate only on variations in α and β.

For conditions specified, the 'constellation' of eigenvalues of the matrix A, calculated as the solution to det(zI - A) = 0 (where 'I' is an identity matrix of dimension A), can now be plotted in the z plane (each eigenvalue, represented by an 'x') as both α and β are incremented from 0 to 1 in steps of 0.1. Figure 3.8 shows this constellation plotted on a Jury map.

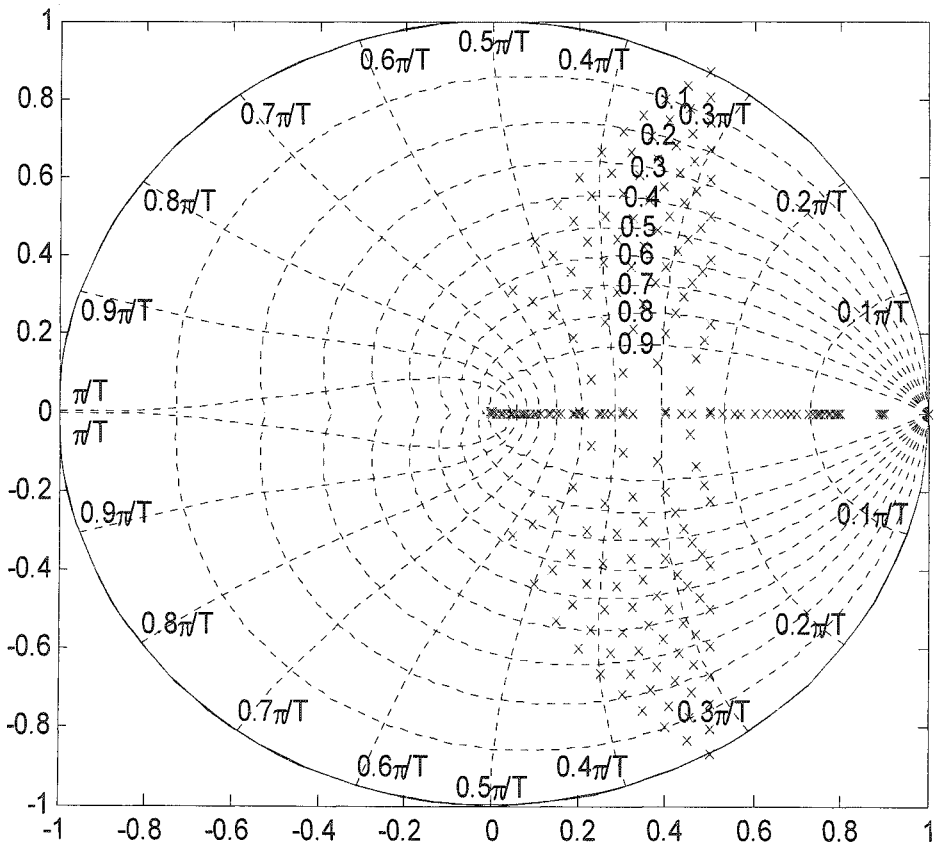


Figure 3.8 *The constellation of eigenvalues plotted on z plane when $\theta = 0.75$, $\alpha = 0$ to 1 in steps of 0.1 & $\beta = 0$ to 1 in steps of 0.1.*

Nine sets of eigenvalues are plotted with unit circle when α and β change from 0 to1. All of these eigenvalues have effects on the dynamics of the system. However, those eigenvalues located on the x axis do not have imaginary part. They do not contribute to the oscillation of the inventory. Furthermore, the damping ratio ς is 1.0 in this case. The system is critically damped. Hence, the research focus was on analysing those plots sitting out of the x axis. It clear from the Figure 3.8 that the plots above the real axis and under the axis are symmetrical. These eigenvalues appear in pairs. Their effects on the dynamics of system are similar.

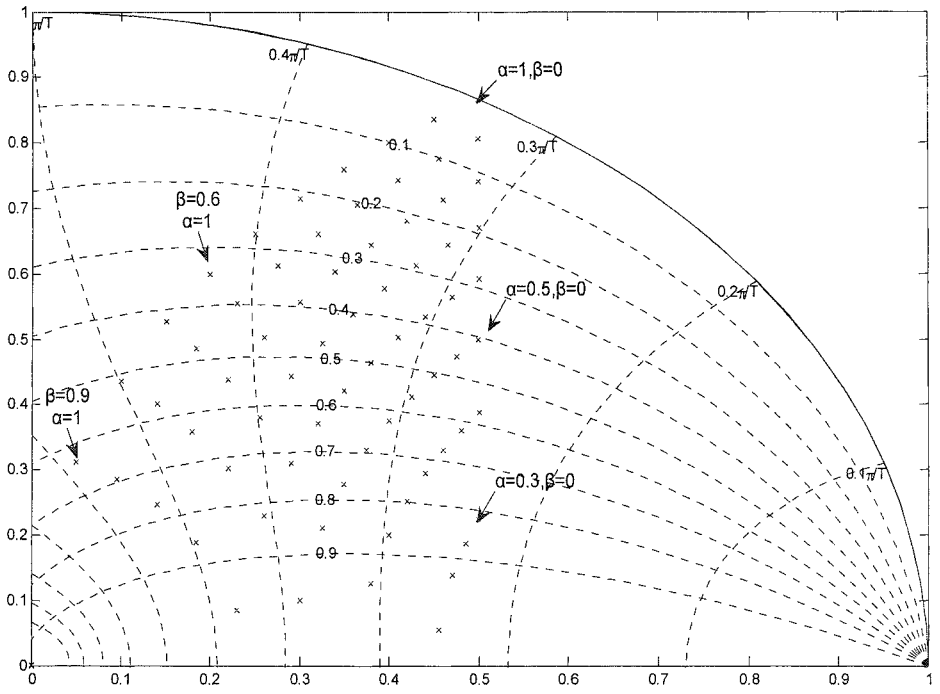


Figure 3.9 *Expanded view of 1st quadrant, ignoring those eigenvalues located along the real axis.*

By expanding the first quadrant of the constellation, the relationship between the eigenvalue movement and parameter chang is explored. Inspection of Figure 3.9 suggests that α has a destabilising influence, whilst β appears to be stabilising. As α decreases from 1 to 0, the corresponding eigenvalues move from the edge toward the inside of the circle. The system becomes heavily damped. In contrast, when β decreases

from 1.0 to 0, the eigenvalues move from the inside to the boundary of the circle. The damping ratio becomes smaller.

The condition $\alpha = 1$, $\beta = 0$, generates a pair of complex eigenvalues that lie on the unit circle at a position 0.5 + i0.866. Damping ratio ς is zero. These are responsible for the 'linear looking' (input dependant amplitude) oscillatory behaviour seen in Figure 3.10. The period of these oscillations is predicted by the angle to be $2\pi/1.047$ (approx 6) distinct samples, where 1.047 is the angle of the eigenvalue. This is confirmed by the inspection of the expanded section, presented in Figure 3.11.

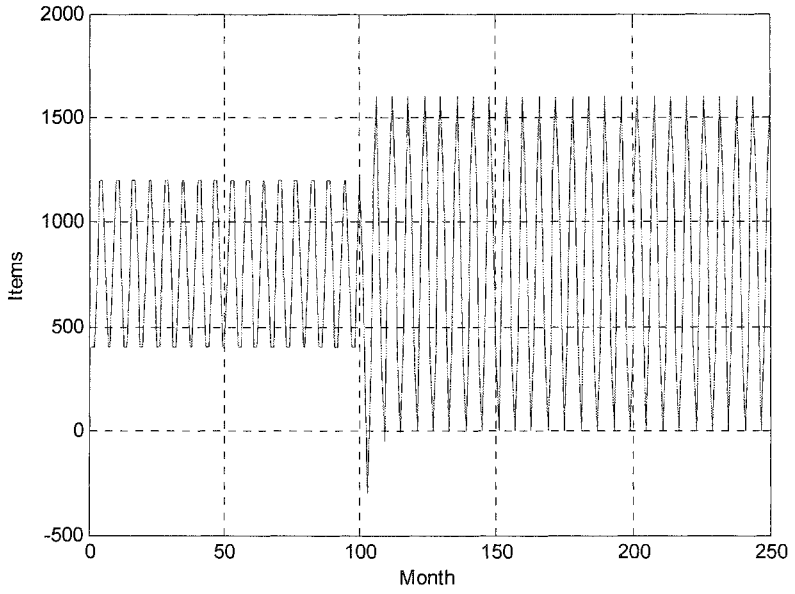


Figure 3.10 *Hub effective inventory for $\theta = 0.75$, $\alpha = 1$, $\beta = 0$ & $Q = 800$. The figure shows the response of the hub effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400 for the first 100 months. Tthese are then increased to 800 for the remainder of the simulation.*

The desired inventory Q is 800 units through the whole simulation. In the first 100 months, the hub inventory oscillates around the desired inventory Q with amplitude at 400 units, which equals the incoming order. The inventory oscillation peaks at 1200 units. From 100 to 250 months, the incoming orders increase to 800 units, and the variation amplitude increases to the same level. The peak of the oscillation becomes 1600 units. When $\alpha = 1$ and $\beta = 0$, the eigenvalus lie on the unit circle. The dynamics of the system are marginally stable. If external disturbances such as the incoming order or the desired inventory are fixed, the hub inventory will oscillate with fixed amplitude.

In real manufacturing, the oscillation of the inventory will cause major problems with regard to fulfilling the demand and managing store costs.

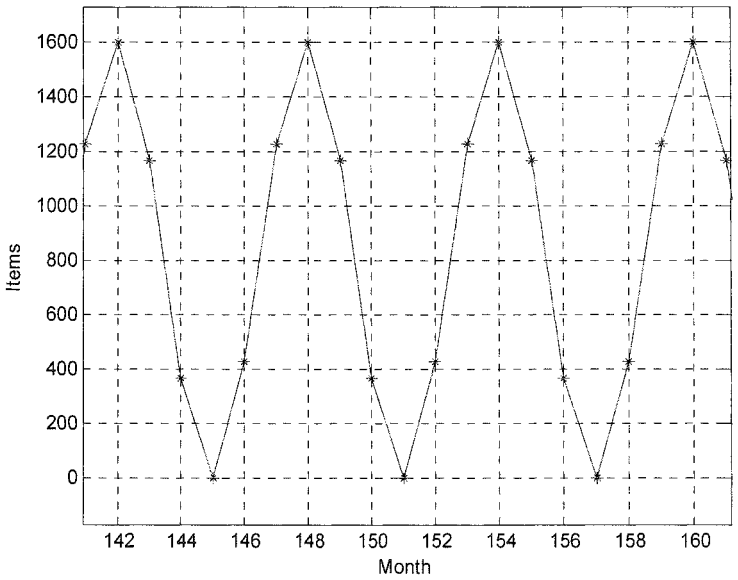


Figure 3.11 *Expanded section of hub effective inventory confirming 6 sample period of oscillation.*

By closer inspecting the hub effective inventory plotting, it is very clear that there are six distinct samples in each wave. The first point of each single wave repeats itself after six plottings. This frequency or period of the oscillation does not vary with changes in the desired inventory or incoming orders. Frequency is only defined by the eigenvalues of the model. The external disturbances can only affect the oscillation amplitude.

Figure 3.12 shows the time response of the hub effective inventory when $\theta = 0.75$, $\alpha = 0.5$, $\beta = 0$ and $Q = 800$ units. It can be seen from Figure 3.9 that the eigenvalues are located inside of the unit circle in this condition.The corresponding damping ratio is around 0.4 (approximately 25% overshoot in the system step response). These clearly dominate the initial (linear) transient seen in Figure 3.12. The second transient, after 100 months, is produced by the 'nonlinear' disturbance caused by an increase in customer orders.

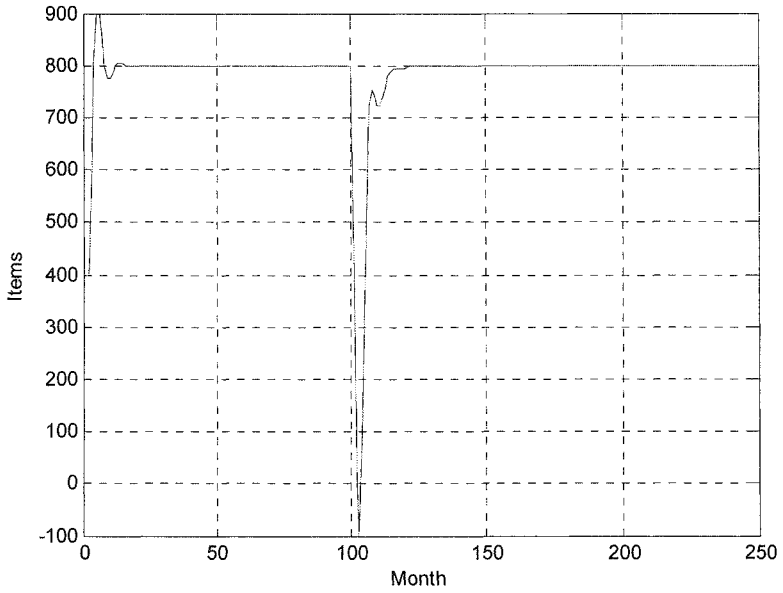


Figure 3.12 *Hub effective inventory for $\theta = 0.75$, $\alpha = 0.5$, $\beta = 0$. The figure shows the response of the hub effective inventory over a 250 month period. All initial values are set to 4 and the desired hub inventory (Q) set to 8. Again customer orders are increased to 8 after 100 months.*

Clearly, when operating linearly the basic hub model tends to be stable. Also, it behaves as expected, since, as can be seen in Equations 3.8 and 3.9, α is a measure of the aggressiveness with which inventory differences are corrected (β, is a measure of the weight with which inventory ordered but still to arrive is taken into account).

3.3.1 The Influence of α

Figure 3.13 shows a plot of the eigenvalue locus when $\beta = 0$ and α is varied from 0 to 1. Inspection of the figure suggests that α has a destabilising influence, since the complex eigenvalue pair moves towards the edge of the unit disc as α increase.

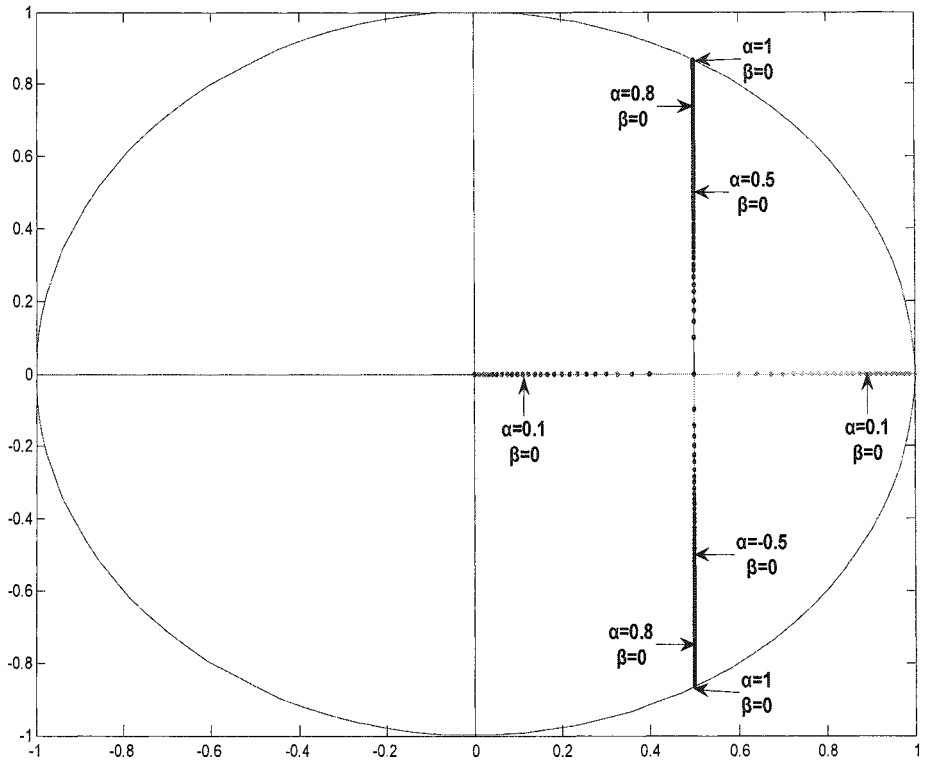
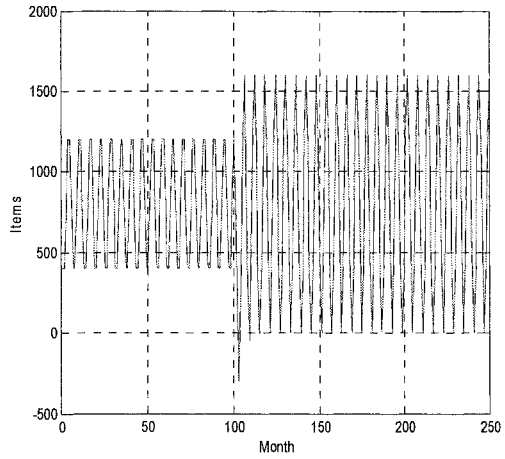
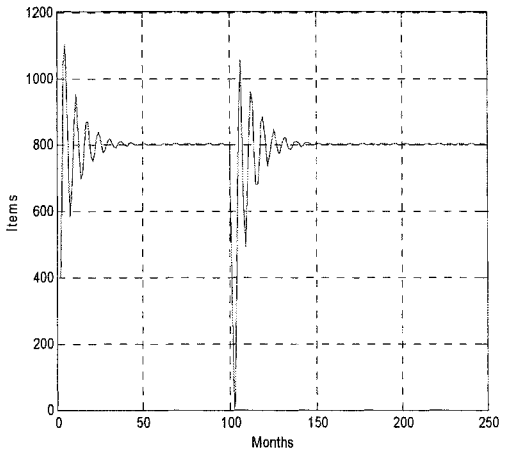


Figure 3.13 *Locus of eigenvalues plotted on the z plane for θ = 0.75, β = 0 and α varied from 0 to 1. Clearly, the complex pair migrate toward the edge of the unit circle as α increases.*

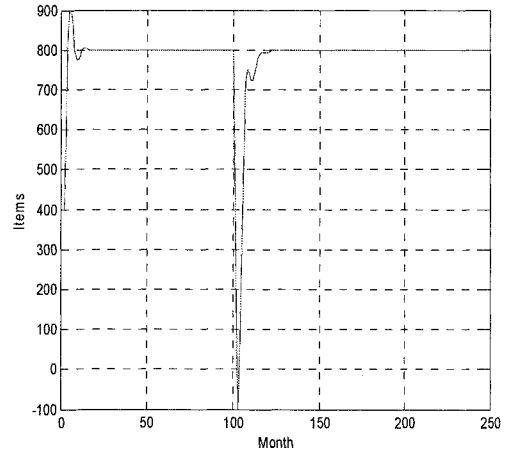
As can be seen, the condition $\alpha = 1$, $\beta = 0$ generates a pair of complex eigenvalues that lie on the unit circle at a position $0.5 \pm j0.866$. These are responsible for the 'linear looking' (input dependant amplitude) oscillatory behaviour seen in Figure 3.14(a). By decreasing α, the oscillations become smaller. For example when $\alpha = 0.8$, $\beta = 0$, the complex eigenvalue pair locate at a position $0.5 \pm j0.745$ and the relevent damping ratio is approximately 0.2. The time response of the hub effective inventory is shown in Figure 3.14(b). When $\alpha = 0.5$, $\beta = 0$, the damping ratio increases to approximately 0.4. The hub inventory oscillates with approximately 25% overshoots and 15 sample settling times in Figure 3.14(c). Finally, when $\alpha = 0.1$, $\beta = 0$ all of the roots lie only on the real axis and the inventory oscillation is purely exponential in Figure 3.14(d).



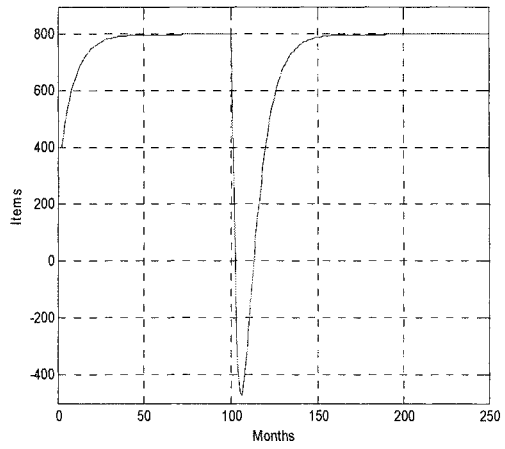
(a) $\alpha = 1, \beta = 0$



(b) $\alpha = 0.8, \beta = 0$



(c) $\alpha = 0.5, \beta = 0$



(d) $\alpha = 0.1, \beta = 0$

Figure 3.14 *Hub effective inventory when $\theta = 0.75$, $\alpha = 1, 0.8, 0.5, 0.1$, $\beta = 0$ and $Q = 800$. The figure shows the response of the hub effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

Clearly (equations 3.8 & 3.9), when $\beta = 0$ the ordering decisions are made based on the current stock levels only. Under these conditions, the management policy is basically integrating and acts to drive inventories to the desired value: in this case Q = 800 units. This is clearly seen in Figure 3.14 (b), (c), (d).

3.3.2 The Influence of β

Unlike the stablising effects of α, β has a destablising effect on the dynamics of the system. Figure 3.15 shows a plot of the eigenvalue locus when α = 1 and β is varied from 0 to 1. It is clear from the figures that the eigenvalues move toward the centre of the unit disc and the oscillations reduce as β is increased. Figure 3.16 shows the corresponding time response when β = 0.2, 0.5 and 1.

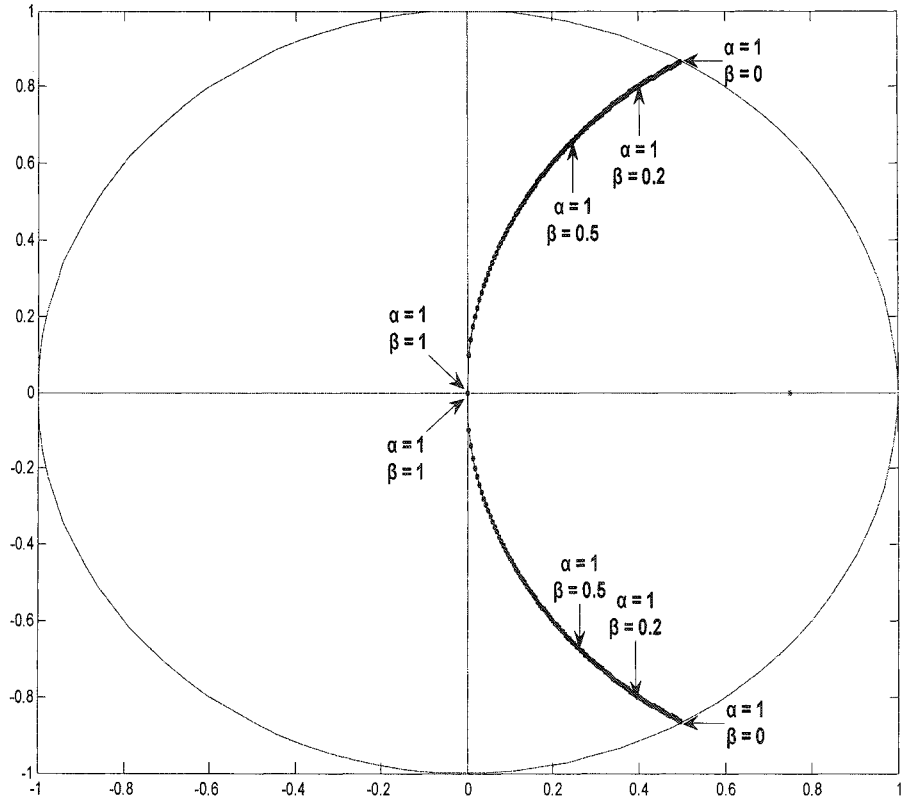
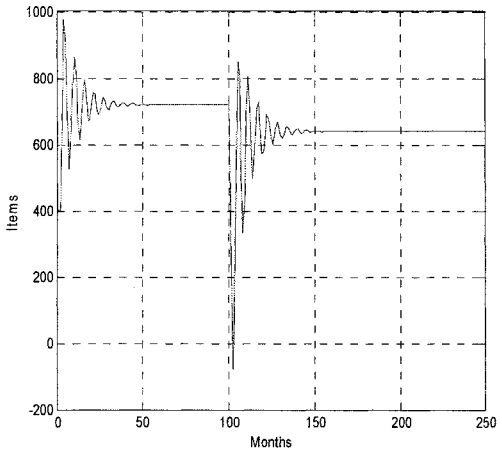
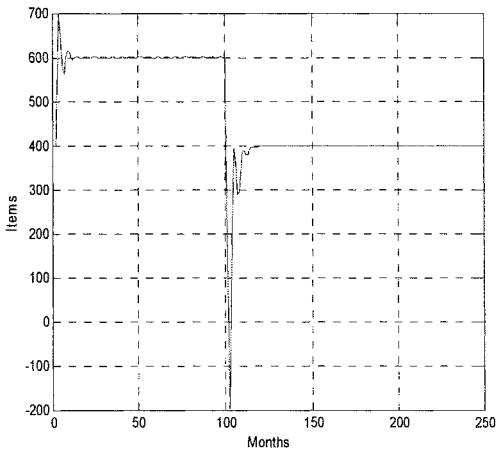


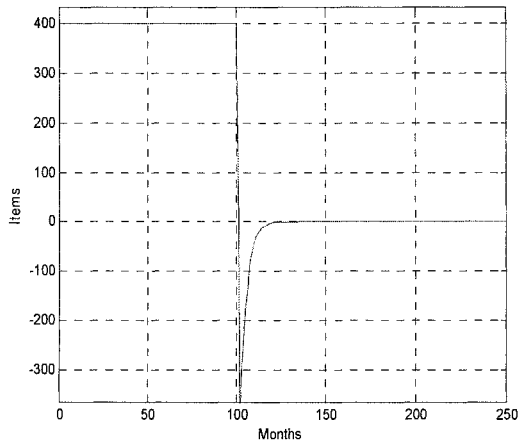
Figure 3.15 *Locus of eigenvalues plotted on the z plane for θ = 0.75, β varied from 0 to 1 and α = 1. Clearly, the complex pair migrate toward the centre of the unit circle as β increases.*



(a) α = 1, β = 0.2



(b) α = 1, β = 0.5



(c) α = 1, β = 1

Figure 3.16 *Hub effective inventory for θ = 0.75, α = 1, β = 0.2, 0.5, 1*

Q = 800. The figure shows the response of the hub effective inventory over a 250 month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.

It is clear from Figure 3.16 that when β is increased the oscillations diminish faster and the hub inventory steady state level decreases as a function of β and the incoming order. Thus, the increase of β enlarge the gap between the desired hub inventory (Q = 800) and the actual hub inventories. For example, in Figure 3.16b β = 0.5, the hub inventory settles at a value of 600 units when the order level is 400 units/month, it then reduces to a steady value at 400 units when the order level increases to 800

units/month. This effect can be easily predicted from Equation 3.11 where the term $\beta x H_{forecast}$ clearly acts as a load disturbance at the input to the previously described integration mechanism (inspection of Equations 3.11 & 3.12 shows that this will not be the case for the factory in which the inventory settles at Q).

3.3.3 The Primary Route to Instability

To obtain a greater understanding of the behaviour observed, the structure of the system block diagram (Figure 3.17) needs to be examined. Inspection of the figure reveals that the factory and the hub have 'isolated' feedback structures. The feedback loops in hub and factory are separate. There is no global feedback loop to link hub and factory. Therefore, by considering their individual characteristic equations, these can be analysed independently.

The system block diagram with zero backlogs in hub and factory has the form:

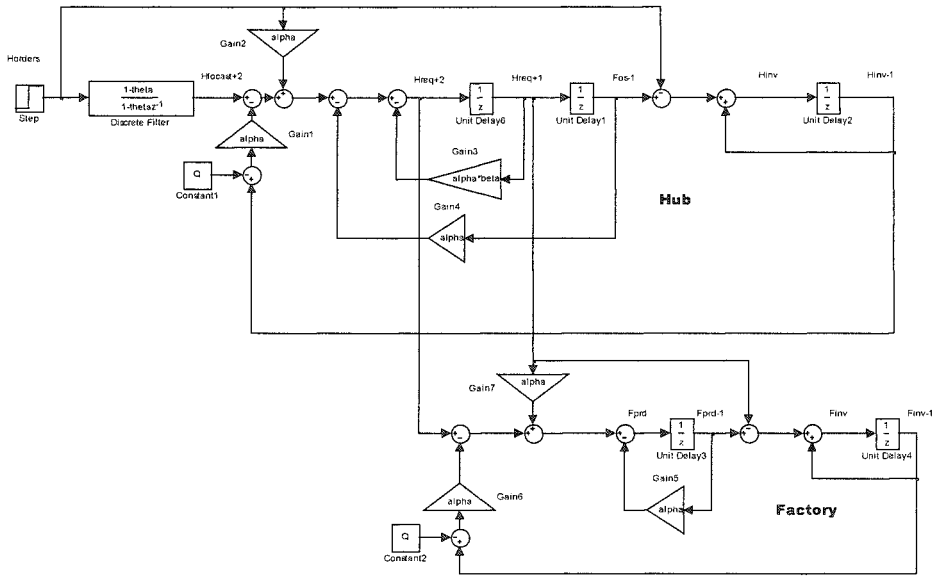


Figure 3.17 *Small signal block diagram representation of the unmodified model of the supply chain at Draeger Safety, UK.*

Thus, for the condition $\beta = 0$ (depicted in Figure 3.13), the factory characteristic equation is:

$$(z + \alpha_F)(z - 1) + \alpha_F = 0 \qquad (3.32)$$

This has two eigenvalues, one at $z = 0$ and a second, which is always real and which lies in the range $z = 1 \rightarrow 0$ as $\alpha = 0 \rightarrow 1$.

The hub characteristic equation is:

$$(z^2 + \alpha_H)(z - 1) + \alpha_H = 0 \qquad (3.33)$$

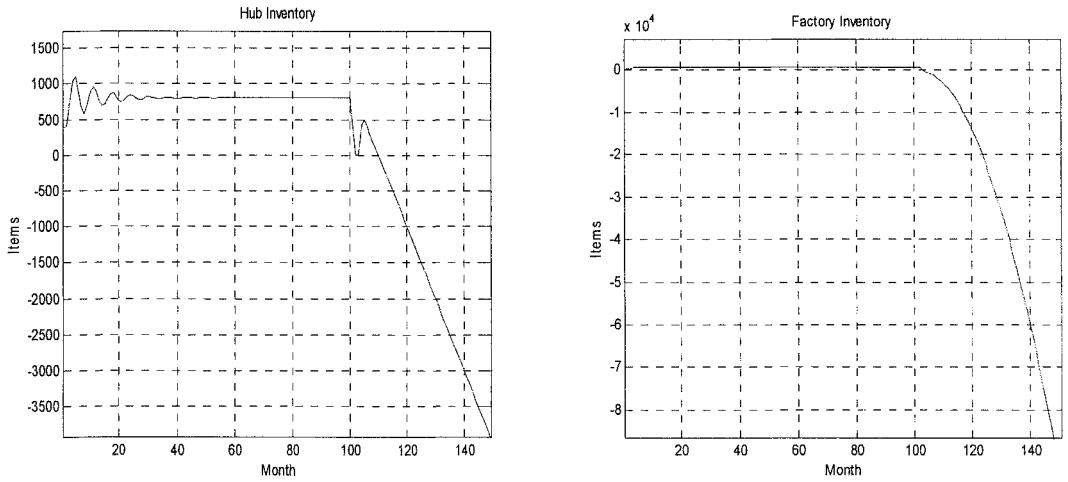
This has three eigenvalues. Again, one of these is at $z = 0$, the other two form a second order pair that become complex when $\alpha > 0.25$. It is this pair that is clearly identified in Figure 3.13. Moreover, it is the hub's dynamics and not the factory's that are the potential source of unstable behaviour.

3.4 The Model with Production Constraint

In real industry, factory production capability is not infinite. Production constraints are introduced into the factory production in this section. Inventory behaviour varies depending on the magnitude of the constraint. With a constraint, the factory production formula changes to

$$F_{prod}(t) = \min(\text{Constraint}, \alpha\,(Q - F_{inv}(t) + F_{blk}(t)) + H_{req}(t+2)); \qquad (3.34)$$

The inventory response is illustrated in three occasions according to the magnitude of the constraint.



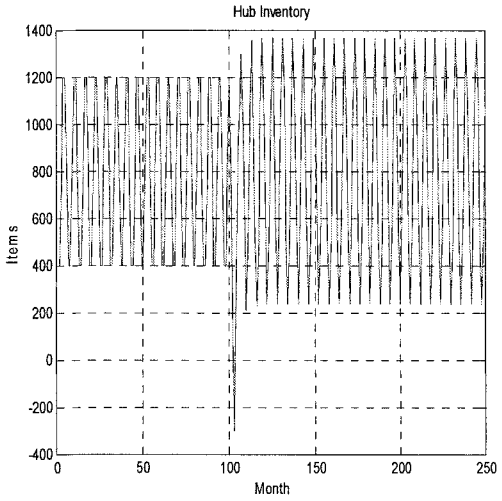
(a) Hub effective inventory *(b) Factory effective inventory*

Figure 3.18 *The response of hub and factory effective inventory over a 150 month periodwhen θ = 0.75, α = 0.8, β = 0, Q = 800, production constraints = 700. The initial hub inventory and factory inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

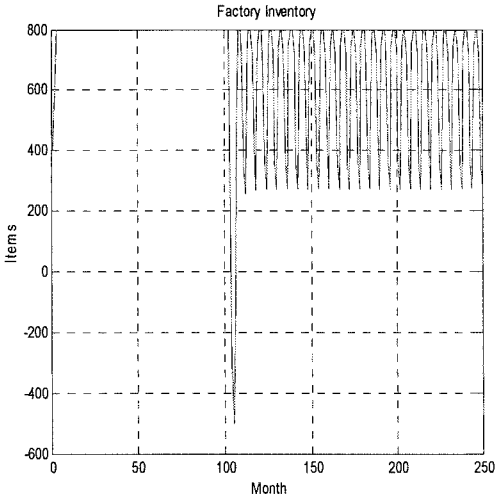
If the constraints are smaller than the incoming order, for example, the production constraints are 700 and incoming orders are 800 in Figure 3.18, the factory production cannot meet the demand. In this case, both hub inventory and factory inventory are exhausted, while the hub backlog and factory backlog keep increasing in each round of the simulation. As shown in Figure 3.18 (a), both hub and factory effective inventory, which are inventory minus backlog, decrease exponentially. The decreasing speed of the hub inventory and the slope of the curve are defined by α and β.

If the constraint is greater than the incoming order, two kinds of behaviour could be generated depending on the size of the constraints. To obtain better understanding of the variation in the inventory, the hub requirements need to be analysed. When α =1, β = 0, the hub requirements only consist of two parts: hub inventory replenishment and hub forecasting. The factory supply line is ignored by multiplying zero β. The hub requirements oscillate around the quantity of the incoming order plus the actual hub inventory. The peak of the oscillation is greater than the incoming order. If there is no production constraint or the constraint is great enough to cover the hub requirement, the hub inventory will swing with great amplitude, while the factory inventory is stable at

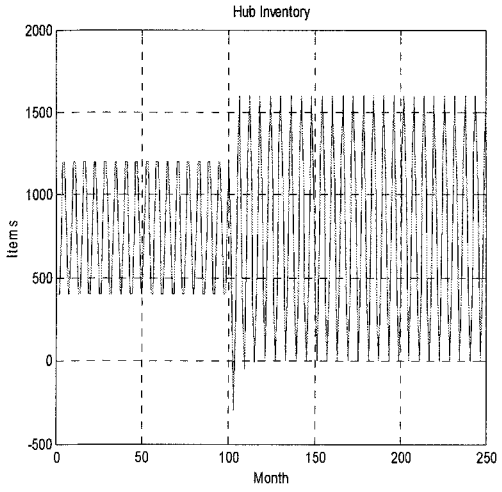
the desired level, as shown in Figure 3.19 (c) and (d). In this case, the constraint has no effect on inventory behaviour.



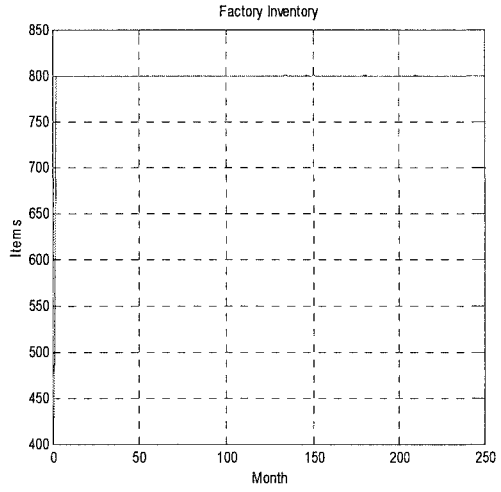
(a) Hub effective inventory with constraint 1000



(b) Factory effective inventory with constraint 1000



(c) Hub effective inventory without constraint



(d) Factory effective inventory without constraint

Figure 3.19 *The response of hub and factory effective inventory over a 250 month period when $\theta = 0.75$, $\alpha = 1$, $\beta = 0$, $Q = 800$, production constraints = 1000 in (a) and (b) and no constraint in (c) and (d). The initial hub inventory and factory inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

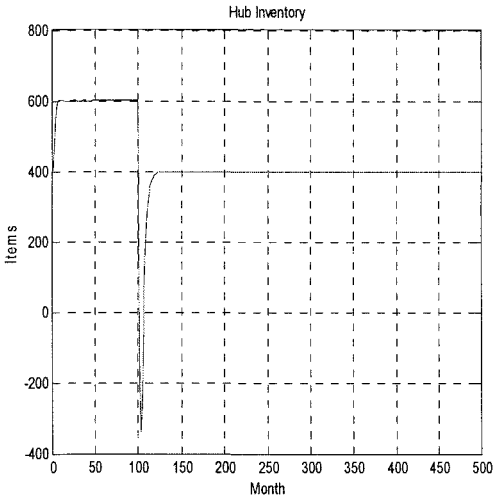
However, if the constraint is not great enough, i.e. greater than the incoming sales order but smaller than the amplitudes of the hub requirement variation, the hub and factory effective inventory is impacted by the constraint. Figure 3.19 (a) and (b) show the inventory behaviour of the model with 1000 production constraints when $\alpha = 1$, $\beta = 0$. Comparing these with Figure 3.19 (c) and (d), the amplitude of the hub effective inventory variation is 1400 units in Figure (a), which is smaller than the amplitude of 1600 units in Figure (c). When the production constraint is smaller than the amplitude of the hub requirement variation, the factory cannot provide enough shipments to make the hub inventory oscillation reach its peak. In this way, the scale of the hub inventory oscillation is reduced. For the same reason, the factory production does not meet the hub requirement. The hub will consume the factory inventory first to meet the hub demand. If the requirement still cannot be satisfied, the additional hub requirement will be backlogged in the factory inventory. Therefore, these factory backlogs caused the oscillation in Figure 3.19 (b).

When the constraints are equal to the incoming order, the situation is more complicated. The inventory behaviour is closely related not only to the production constraints and incoming orders but also to the system parameters α, β. In the follwing, the dynamics of the model are analysed according to different values of β.

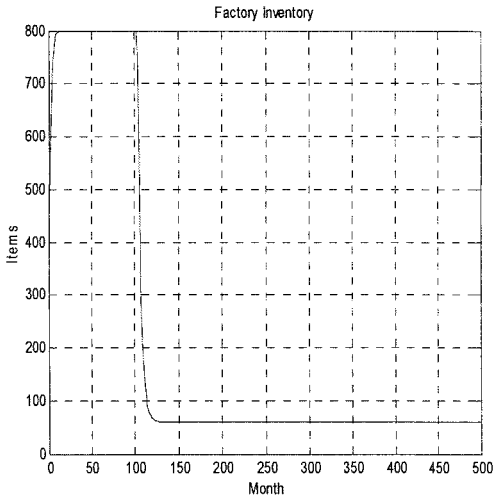
When β is greater than zero, the hub requirement are calculated as

$$H_{req}(t+2) = \max(0, \alpha (Q - H_{inv}(t) + H_{blk}(t)) \qquad (3.35)$$
$$- \alpha \beta (F_{blk}(t) + F_{ship}(t)) + H_{forecast}(t+2));$$

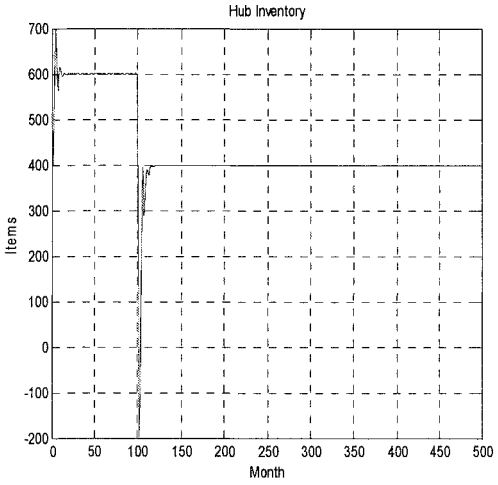
The hub requirements are determined by three factors, '$\alpha (Q - H_{inv}(t) + H_{blk}(t))$', the hub inventory replenishment, '$- \alpha \beta (F_{blk}(t) + F_{ship}(t))$', the factory supply line, and '$H_{forecast}(t+2)$' the hub forecasts. α represents how fast the hub inventory can be recovered to the desired level Q. β defines the weight of the consideration about the previous order sent to the factory. Since the production constraint is equal to the incoming order, factory production is only able to fulfil the sales order, but not be to keep the factory inventory and hub inventory at their desired level. According to the different value of α and β, the hub inventory and factory inventory are stable at different levels.



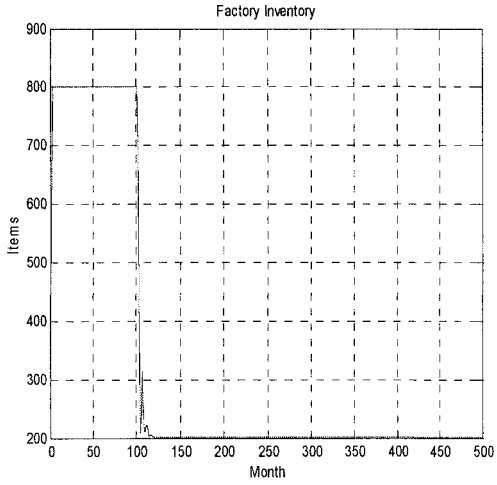
(a) Hub effective inventory
α =0.4, β = 0.5



(b) Factory effective inventory
α =0.4, β = 0.5

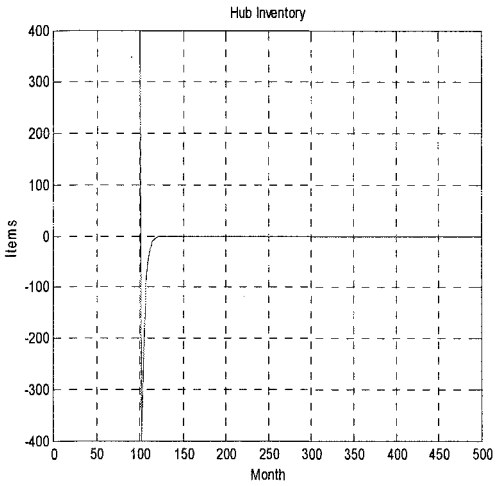


(c) Hub effective inventory
α =1, β = 0.5

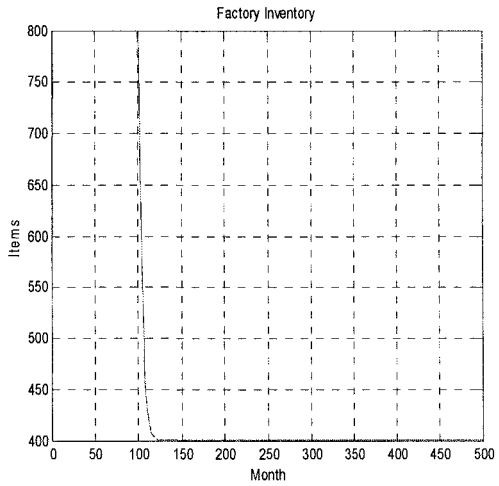


(d) Factory effective inventory
α =1, β = 0.5

Figure *3.20 Hub and factory effective inventory*



(e) Hub effective inventory
α =1, β = 1



(f) Factory effective inventory
α =1, β = 1

Figure 3.20 *The response of hub and factory effective inventory over a 250 month period for θ = 0.75, Q = 800, production constraints = 800. The initial hub inventory and factory inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

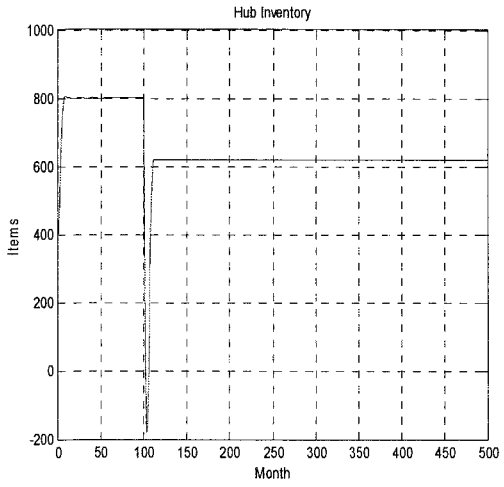
In Figure 3.20 (a) and (b), the hub inventory is stable at 400 units and the factory inventory is stable at 75 units when α =0.4, β = 0.5. Increasing α to 1 and keeping β at 0.5 in Figure 3.20 (c) and (d), the hub inventory is stable at the same level; however, the factory inventory's stable level increases to 200 units. Fixing α and increasing β to 1 in Figure 3.20 (e) and (f), the hub inventory stable level decreases to 0 and the factory inventory level increases to 400 units. It can be concluded that when β is increased, the factory inventory stable level is increases, while the hub inventory stable level is decreased. When β = 1, the hub management policy take into full consideration previous decisions, which are the requirements the hub placed before. In the hub requirement formula, the previous requirements are expressed as '- α ($F_{blk}(t) + F_{ship}(t)$)' corresponding to the factory shipment and factory backlog. When the hub inventory is consumed, the hub requirement are calculated as

$$H_{req}(t+2) = \alpha Q - \alpha (F_{blk}(t) + F_{ship}(t)) + H_{forecast}(t+2); \quad (3.36)$$

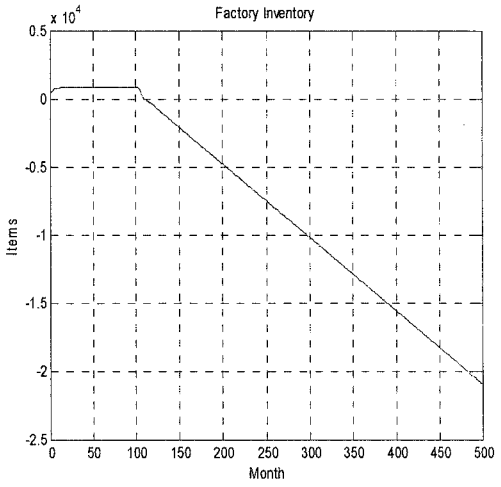
The hub inventory replenishment is α Q. The factory shipment is the production constraint 800 units, which equals the hub desired inventory Q. Hence, the new Hub

requirement is the hub forecast $H_{forecast}(t+2)$, which is the same as the incoming sales order. In this case, the model keeps the factory inventory at the current level and consumes the hub inventory, which is shown in Figure 3.20 (e) and (f).

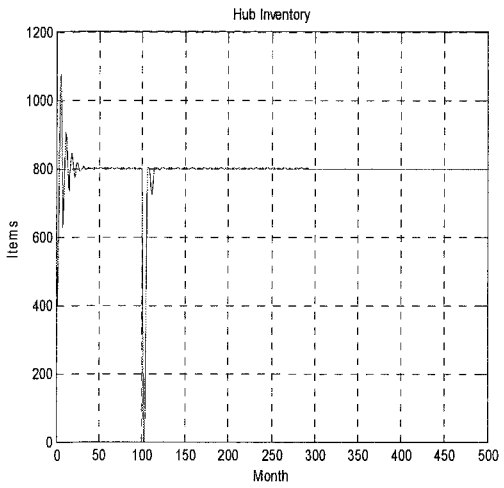
When β is 0, the heuristics are simulated as that the hub planning manager never considers the supply line in the factory and ignores their previous hub requirements and the backlog in the factory. In this case, the production constraint has a huge influence on both hub and factory inventory.



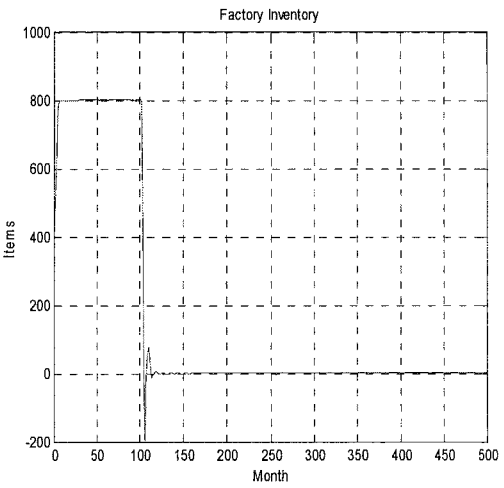
(a) α = 0.3, β = 0



(b) α = 0.3, β = 0

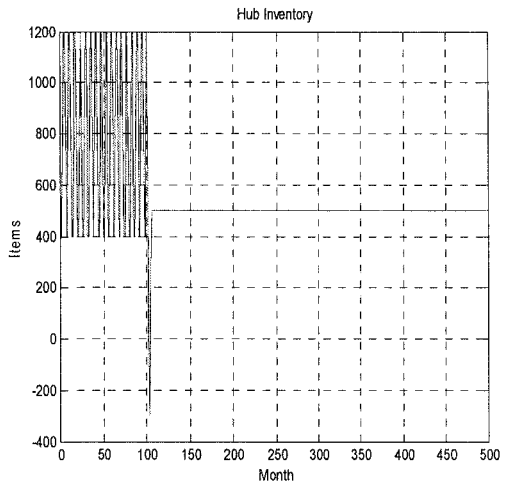


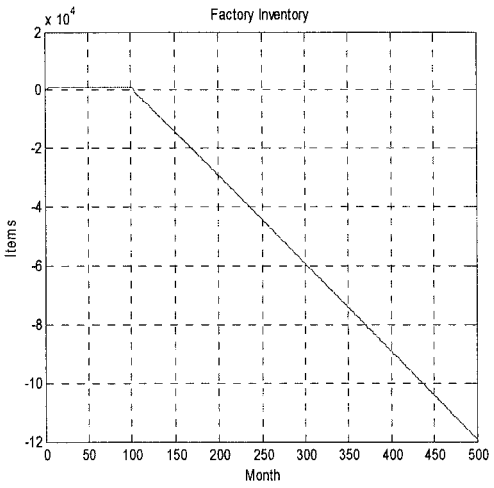
(c) α = 0.75, β = 0



(d) α = 0.75, β = 0

Figure 3.21 *Hub and factory effective inventory*





(e) $\alpha = 1, \beta = 0$ *(f) $\alpha = 1, \beta = 0$*

Figure 3.21 *The response of hub and factory effective inventory over a 250 month period for $\theta = 0.75$, $Q = 800$, production constraints = 800. The initial hub inventory and factory inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

When $\beta = 0$, the hub requirement formula changes to

$$H_{req}(t+2) = \alpha (Q - H_{inv}(t) + H_{blk}(t)) + H_{forecast}(t+2)); \qquad (3.37)$$

The minus part '- α ($F_{blk}(t) + F_{ship}(t)$)' in equation 3.35 has disappeared in equation 3.37. The hub requirements are dependent on the sum of hub inventory replenishments and hub forecast, which equal the incoming sales order. The production constraint allows the factory to fulfil the incoming sales order only. In this case, the hub starts to consume the factory inventory to recover its own inventory. The recovering speed is relevant to α. The extra requirements are backlogged. The factory production formula is redefined as

$$F_{prod}(t) = \alpha (Q - F_{inv}(t) + F_{blk}(t)) + H_{req}(t+2); \qquad (3.38)$$

Integrating formula (3.37) and (3.38), it is clear that as α is increasing, the factory accelerates increasing their production to the constraint level. When α is smaller than 0.75, factory production cannot be improved from 400 units to 800 units within one month. Both hub and factory inventory have been consumed. In this case, the hub keeps

sending requirements to satisfy the sales demand and recover its own inventory to the desired level. However, the factory cannot afford to replenish the hub inventory and have to backlog the extra order. The hub does not take account of the factory backlog and keeps sending requirements to replenish its own inventory. The requirements double, then triple.... Both hub and factory are got caught in a vicious circle. The inventory performance is shown in Figure 3.21 (a) and (b). The hub inventory cannot be replenished to the desired level of 800 units, but is stable at lower level 600 units. The factory inventory decreases exponentially as the simulation is runing.

When α is greater than or equal to 0.75 and less than 0.99, factory production can be improved to the constraint level in one month. In this case, the factory still has the opportunity to compensate the hub inventory to the desired level Q with improved production and its own inventory. In this situation, the hub does not send extra orders and does not generate the factory backlog. However, the factory cannot recover its own inventory owing to the production constraint. The inventory behaviour is shown in Figure 3.21 (c) and (d). The hub inventory is stable at the desired level of 800 units, while factory inventory is stable at zero level.

When α is greater than or equal to 0.99, as in Figure 3.21(e) and (f), the whole system becomes unstable. When the incoming orders increase to 800 units in month 101, the hub effective inventory oscillates at an amplitude over 800 units. Considering the formula 3.37, the large hub inventory leads to a small hub requirement. In this way, even if α is large enough compared with the case in Figure 3.21 (a) and (b), the hub requirement is still relatively small. factory production still cannot be improved to the constraint level within a month. As a result of this, the same vicious circle emerges between hub and factory. As shown in Figure 3.21 (e) and (f), the hub inventory is stable at 500 and the factory inventory decreases exponentially. The inventory performance in this case is similar to that in Figure 3.21(a) and (b) when α is smaller than 0.75.

3.5 Discussion

Within the context of this specified 'management policy', the two correction factors α and β, can be thought of as representing the psyche of the decision maker: $\alpha = 1$ and $\beta = 0$, represents one extreme; an aggressive manager thinking only within the confines of his area of control and not recognising the impact of previous decisions (no account of history). In the simulations, this resulted in persistent oscillations (Figure 3.14a), generated purely by the decision making strategy and not as the result of some external influence (as is often claimed by managers with such tendencies). Such oscillations, while not impacting on customers (as inventories are, in general, maintained at positive levels) would require larger than necessary warehouse space. Decreasing α, and thus being less rigorous in terms of stock maintenance, produces a stabilising effect: Figures 3.14b & 3.14c. However, if α is too small (representing a very cautious manager) the response becomes too slow and can thus lead to significant backlogs and the potential for customer dissatisfaction: Figure 3.14d.

$\beta > 0$, represents a manager that thinks outside of the immediate area of control and takes some or full account of history. Taking into account the impact of previous decisions has a stabilising effect: Figures 3.16a, 3.16b & 3.16c. $\alpha = 1$, $\beta = 1$ would seem to represent the ideal case (just in time, or deadbeat, management – with the eigenvalues lying at the origin of the z plane: Figure 3.16c). It is well known, however, that such control policies often prove unreliable as they tend to be intolerant of unexpected events, such as unforeseen production hold-ups.

By introducing production constraints to the factory, the oscillation becomes more complicated. The dynamics of the system are not only impacted by the parameter α and β, but also depend on the degree of constraint and the quantity incoming orders. The vicious circle could occur at a particular range of α and β when the production constraints equal the sales order. Since the hub order can never be fulfilled, the factory effective inventory is backlogged forever: Figure 3.21 (e) & (f). However, by selecting the right value of decision parameters, these destructive vicious circle could be avoided: Figure 3.21 (c) & (d).

From the analysis given here, it is clear that the dynamics of the supply chain are intimately dependent on the decision making strategy of the manager. A purely aggressive strategy without considering the history of the system, will usually lead to undesired behaviour. Indeed, this message can be extended towards management decision making in a more general context: effective management decisions should take into account the previous history of the situation and the impact of previous decisions.

What is more interesting, however, in terms of the analysis presented in this chapter is that it is possible to establish a relationship between the two major feedback loops (the factory loop representing factory management policies and the hub loop representing hub management policies) and potential system behaviours. For example, in the 'as it is' arrangement (represented by Equations 3.32 & 3.33) the route toward instability is via the complex eigenvalue pair which are generated via the hub management policy. More importantly, however, because the systems are 'isolated', poor management decisions in the hub cannot be corrected by good decisions in the factory. On the other hand, a good manager in the hub may be able to mitigate poor management decisions in the factory.

Chapter 4:

Hub Model with Two Additional Production Delays

To help better understand the factory/hub interaction, in this section the effects of adding an additional two months' delay into the production process are investigated. This additional delay brings the model into line with the classical Beer Game representation. However, unlike in the Beer Game, where the delay occurs naturally owing to the nature of the brewing process, here, such an additional delay would tend to be the result of some unforeseen external influence: perhaps a lack of material supply (due to industrial action by a supplier) or a prolonged upturn in business that outstrips the ability to produce.

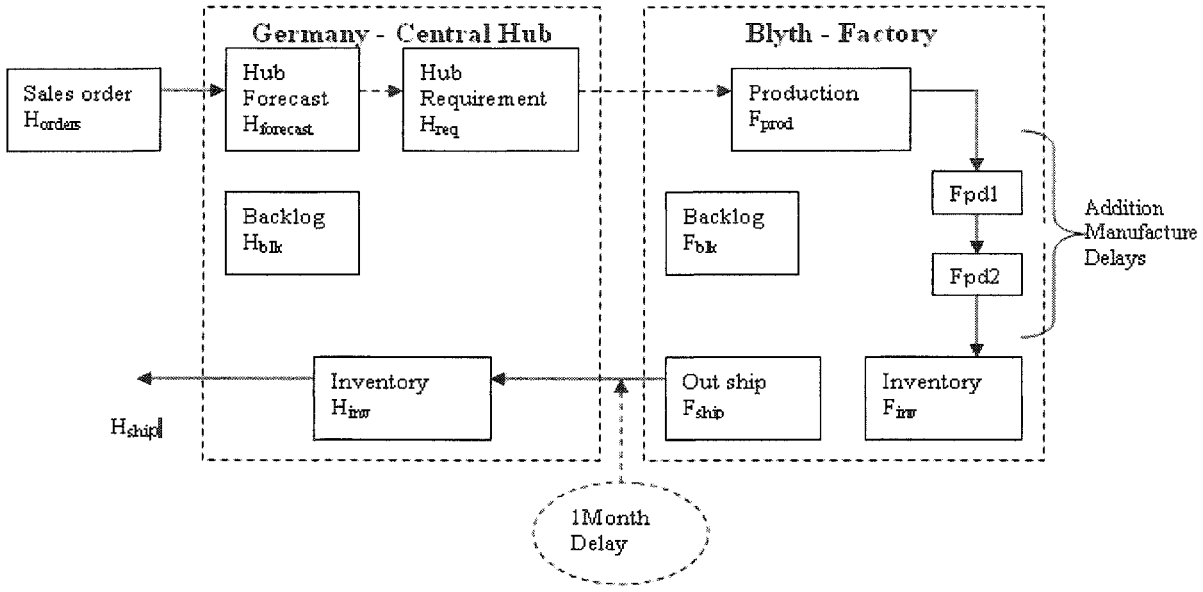


Figure 4.1 *Schematic diagram showing relationships between factory and central hub. The figure represents a 'Production Delay' model of the supply chain at Draeger Safety, UK.*

The basic model operates on a one month sample rate, therefore, the two month additional delay in production is represented by the inclusion of two additional factory states, F_{pd1} and F_{pd2}. To clarify the working progress of this delayed model, it is assumed that the hub receives the sales order and makes a forecast for March in January. As discussed in the previous chapter, the factory makes the production plan and starts manufacturing in January. The work flow moves into F_{pd1} in February and F_{pd2} in March. The production is finished in April and the finished goods are placed in the factory inventory ready to ship. After one month's transport, the finished goods arrive at the hub inventory in May. In other words, the hub requirements for March have been delayed for two months.

4.1 The Draeger Model

The equations for the factory part have been modified by the two extra production delays. The rest of the formulas remain the same as for the original model. The equations representing the factory inventory thus become:

$$F_{inv}(t) = \max(0, F_{inv}(t-1) + F_{pd2}(t-1) - F_{ship}(t)) \qquad (4.1)$$

The factory inventory is updated by the incoming shipment from the second stage of production $F_{pd2}(t-1)$, and subtracting from the outgoing shipment $F_{ship}(t)$.

$$F_{ship}(t) = \min(H_{req}(t+1) + F_{blk}(t-1), F_{inv}(t-1) + F_{pd2}(t-1)) \qquad (4.2)$$

If the factory inventory plus the incoming shipment of $F_{pd2}(t-1)$ are sufficient, the outgoing shipment is the hub requirements plus the factory backlog. Otherwise, the outgoing shipment is the factory inventory plus the incoming shipment. The updated factory inventory becomes empty.

$$F_{blk}(t) = \max(0, F_{blk}(t-1) + H_{req}(t+1) - (F_{inv}(t-1) + F_{pd2}(t-1)) \qquad (4.3)$$

The factory backlog is calculated in the same way as the factory out shipment. If the hub requirements plus backlogs are completely covered by the incoming shipments plus existing inventories, the new backlog is empty, otherwise the extra requirements are backlogged.

$$F_{prod}(t) = \max(0, \alpha (Q - F_{inv}(t) + F_{blk}(t)) + H_{req}(t+2) - \alpha \beta (F_{pd1}(t) + F_{pd2}(t))) \quad (4.4)$$

factory production is adjusted above or below the expected demand to maintain the current inventory, $\alpha (Q - F_{inv}(t) + F_{blk}(t))$, and supply lines, $-\alpha \beta (F_{pd1}(t) + F_{pd2}(t))$, at the desired levels. In this production delayed model, the supply line has been extended, since the two extra production delays $F_{pd1}$and F_{pd2} have been introduced. The factory needs to consider the supply line by employing β as '$-\alpha \beta (F_{pd1}(t) + F_{pd2}(t))$' when they make decisions on the production plan.

$$F_{pd1}(t) = F_{prod}(t-1) \quad (4.5)$$

$$F_{pd2}(t) = F_{pd1}(t-1) \quad (4.6)$$

The first delay term F_{pd1} equals the last step production. The second term F_{pd1} equals the last step of the first delay term. The production process is delayed for two months. The equations are programmed in Appendix A.

4.2 Hub Analysis

The nonlinearity existing in the equations of the delayed model are of a saturation form. The initial analysis of the model is based on a locally linear representation in which all backlogs are assumed to be zero. The same discrete system analysis methods as with the original model are applied. The general discrete state space representation is defined as

$$X(k) = A\,X(k-1) + B\,U(k) \quad (4.7)$$

$$Y(k) = C\,X(k) \quad (4.8)$$

Therefore, the new state vector X(t) is:

$$X(t)=\begin{bmatrix} H_{ship}(t) \\ H_{inv}(t) \\ H_{req}(t+2) \\ H_{forcast}(t+2) \\ F_{ship}(t) \\ F_{inv}(t) \\ F_{prod}(t) \\ F_{pd1}(t) \\ F_{pd2}(t) \end{bmatrix} \tag{4.9}$$

and the input U(k) is:

$$U(k)=\begin{bmatrix} H_{orders}(t) \\ Q_H(t) \\ Q_F(t) \end{bmatrix} \tag{4.10}$$

The system matrices are:

$$A=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\alpha_H & -\alpha_H\beta_H & \theta & -\alpha_H & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & -\alpha_H & \alpha_F-\alpha_H\beta_H & \theta & -\alpha_H & -\alpha_F & -\alpha_F\beta_F & -\alpha_F\beta_F & -\alpha_F \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.11}$$

$$B=\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1+\alpha_H-\theta & \alpha & 0 \\ 1-\theta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1+\alpha_H-\theta & \alpha_H\alpha_F & \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad C=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.12}$$

Inspection of equation (4.9) indicates that matrix X(t) consists of nine elements. The system matrix A is a nine times nine matrix. There are two extra elements in matrix X(t) and two additional rows in system matrix A (compared with the original model with a seven times seven system matrix). Those two additional elements and rows express the extra delay F_{pd1} and F_{pd2}. After Z transform of the state space equation (4.11), the eigenvalues of the characteristic matrix A are calculated. These extra elements and rows produce two more eigenvalues. As discussed in the last chapter, in the study presented here, θ will remain fixed at 0.75 and the analysis will concentrate only on the variations in α and β.

For the conditions specified, the 'constellation' of eigenvalues of the A matrix, calculated as the solution to det(zI - A) = 0 (where 'I' is an identity matrix of dimension A), can now be plotted on the z plane (each eigenvalue, represented by an 'x') as both α and β are incremented from 0 to 1 in steps of 0.1. Figure 4.2 shows this constellation plotted on a Jury map.

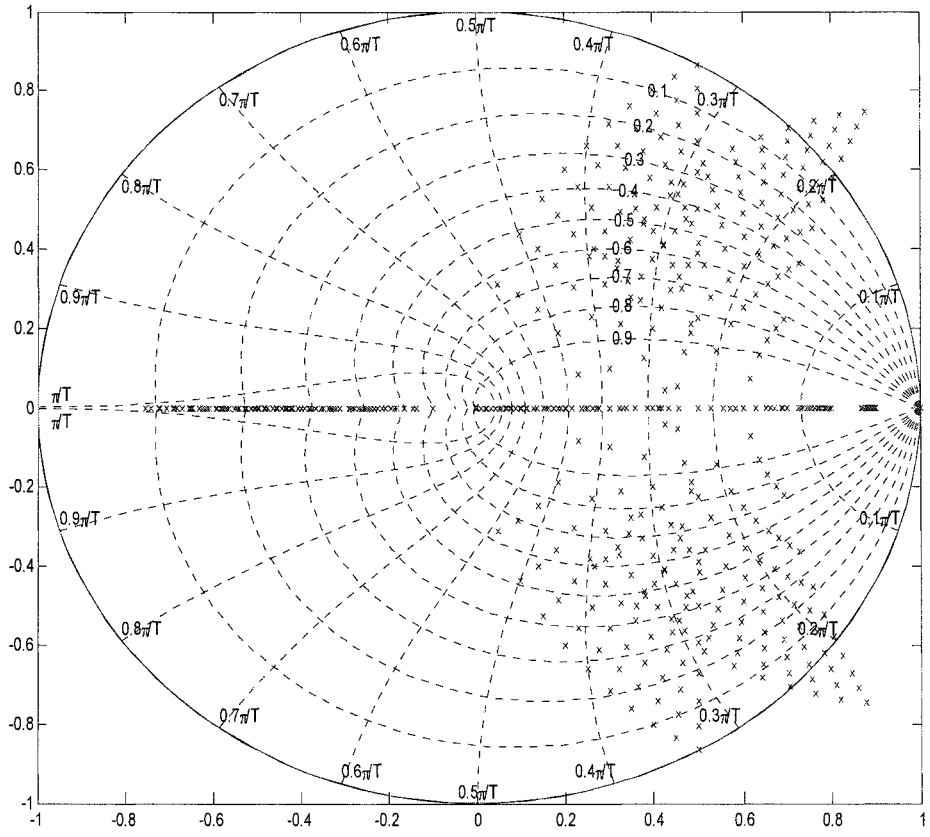


Figure 4.2 *The constellation of eigenvalues of A plotted on the z-plane for θ = 0.75, α = 0 to 1 in steps of 0.1 & β = 0 to 1 in steps of 0.1.*

Nine sets of eigenvalues have been plotted together in constellation as changing α and β from 0 to 1. The damping ratio curve ς and natural frequency w_n are marked within the constellation so that the details of the eigenvalue related variations become easier to predict. Inspection of Figure 4.2 shows the overall shape of the eigenvalue constellation is symmetrical. Compared with the original model, there is one more 'shark fin' that appears on both sides of the real axis. Ignoring the effects of the eigenvalues located on the real axis, which have no effects on the oscillation these four fins contribute to the dynamics of the production delayed model. Given a set of α and β, four complex eigenvalues are produced to represent the state of the system. The dynamics of the inventory become more complicated. As the eigenvalues plots are symmetrical, the analysis concentrates on the upper half of the quadrant.

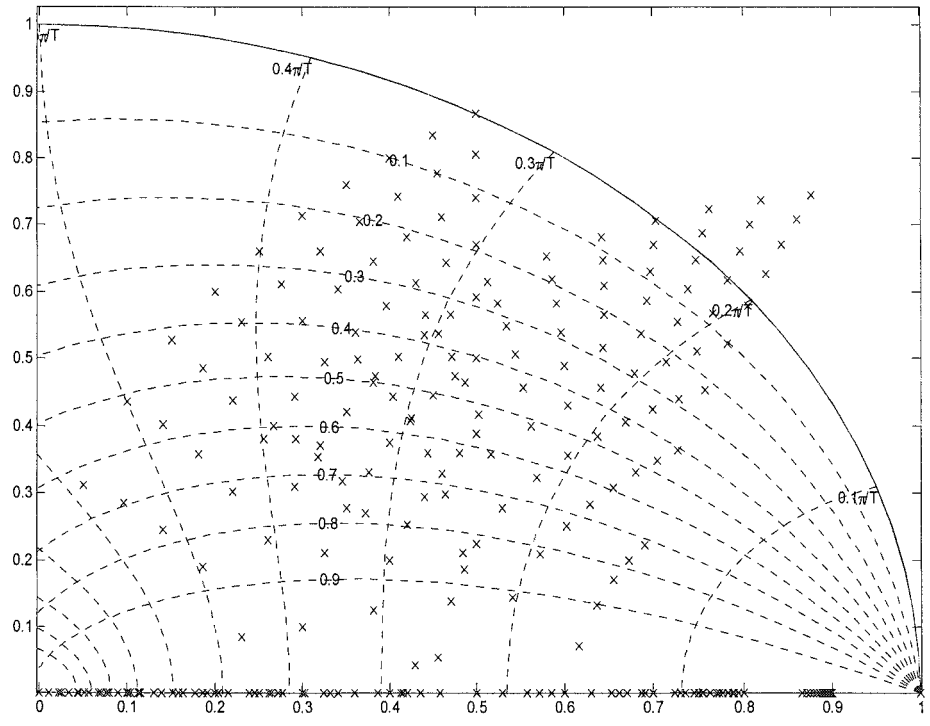
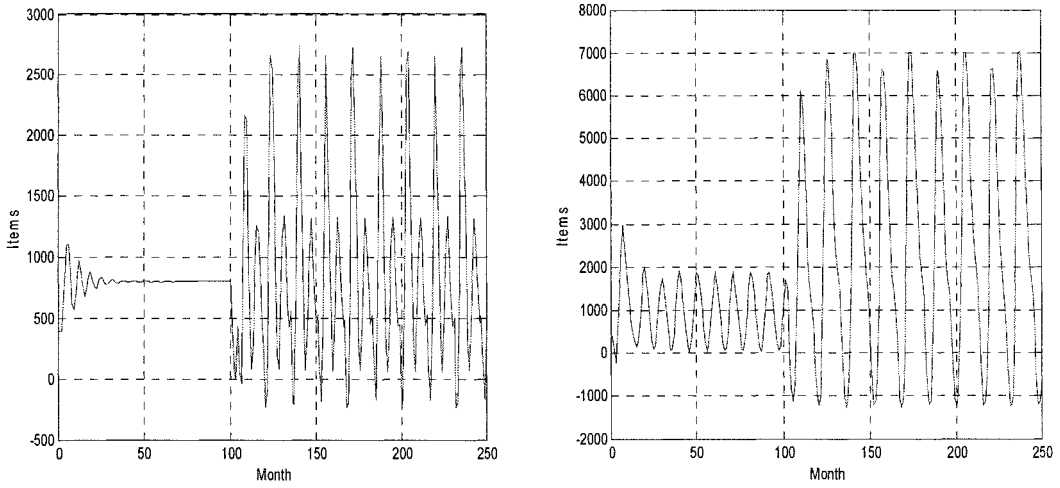


Figure 4.3 *Expanded view of 1st quadrant*

By expanding the first quadrant of the constellation, the correlation between eigenvalue movement and changing of parameters is explored. Fixing $\beta = 0$ and increasing α, the eigenvalues on the left fin move toward the corner on the edge of the unit circle, while the eigenvalues on the right fin move to the corner located outside of the unit circle. It can be concluded that the parameter α has a destabilising effect. On the other hand, keeping α at one and increasing β, both eigenvalues on the left fin and right fin move toward the inside of the unit circle. Therefore β has a stabilising effect on the delayed model. Given $\alpha = 0.8$, $\beta = 0$, two eigenvlaues are produced on the map. One on the left fin locates inside of the circle; the other on the right fin sits outside of the circle. Once there is an eigenvalue locating outside of the circle, the system will be unstable and the inventory starts oscillating. Hence, even though $\alpha = 0.8$, $\beta = 0$ indicates stable system behaviour in the original model, the same management policy still produce oscillations in the inventory in the production delayed model as shown in Figure 4.4.

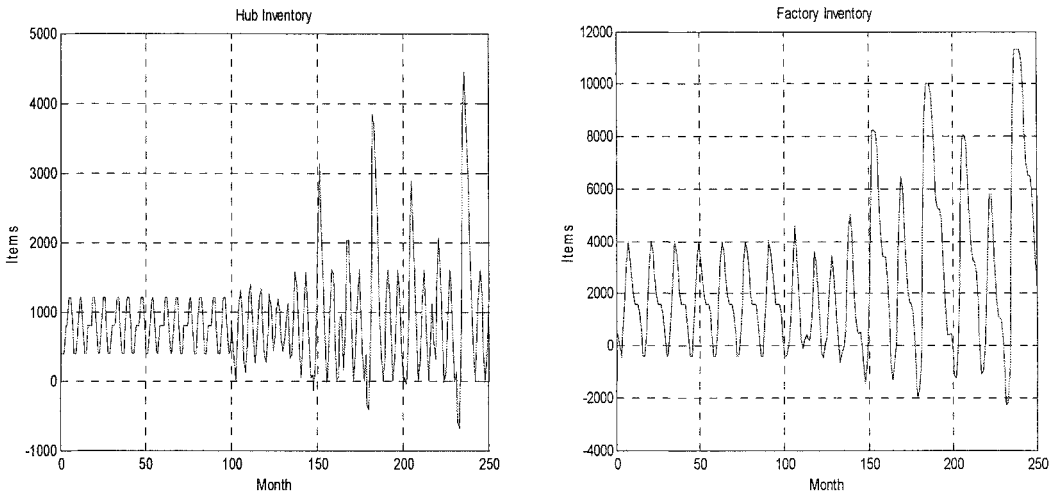


(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.4 *Hub effective inventory and factory effective inventory for $\theta = 0.75$, $\alpha = 0.8$, $\beta = 0$ and $Q = 800$. The figure shows the response of the hub effective inventory and factory effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

In Figure 4.4, the hub inventory is stable in the first 100 months, while the factory inventory starts oscillating from the beginning of the simulation. The factory inventory level is positive. After the first 100 months, the sales order is doubled. The amplitude of the factory inventory oscillation becomes much bigger and starts to generate backlogs, as seen in Figure 4.4 (b). The factory backlog leads to shortage of the outgoing shipment to the hub and makes the hub inventory oscillate. In Figure 4.4 (a), the hub inventory becomes unstable and oscillating after the first 100 months.

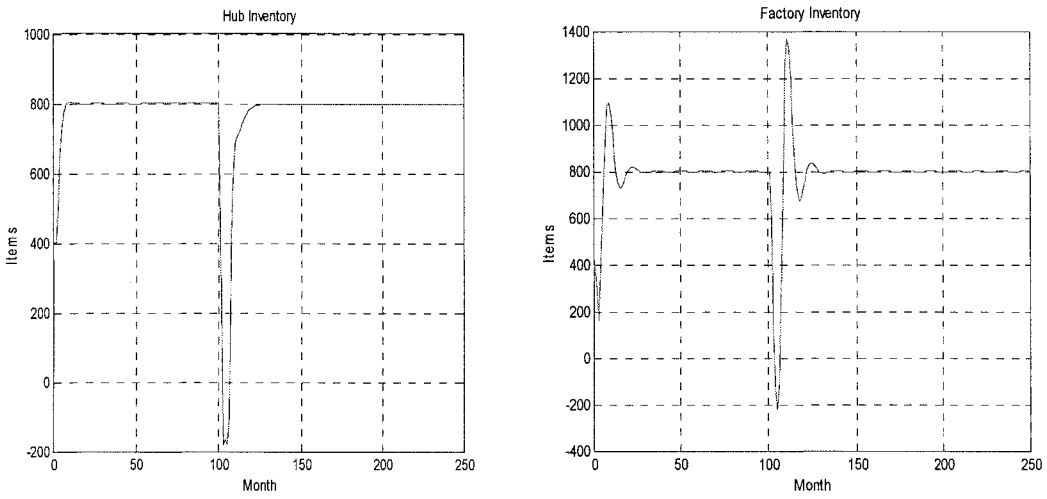
When the eigenvalues on both fins locate outside of the circle, the inventory displays huge variations. For example, when $\alpha = 1$ and $\beta = 0$, the eigenvalues locate on the two corners of the two fins in Figure 4.3. The inventory behaviour resulting from an extremely aggressive policy is illustrated in Figure 4.5. Both hub and factory inventories are oscillate from the begining of the simulation. The amplitudes of the oscillations keep expanding after the incoming order doubles after 100 months. Closer inspection of the Figure 4.5 (a) and Figure 4.4 (a), the hub inventory behaviour in the first 100 months is essentially different. A detailed analysis is carried out below.



(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.5 *Hub effective inventory and factory effective inventory for $\theta = 0.75$, $\alpha = 1$, $\beta = 0$ and $Q = 800$. The figure shows the response of the hub effective inventory and factory effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

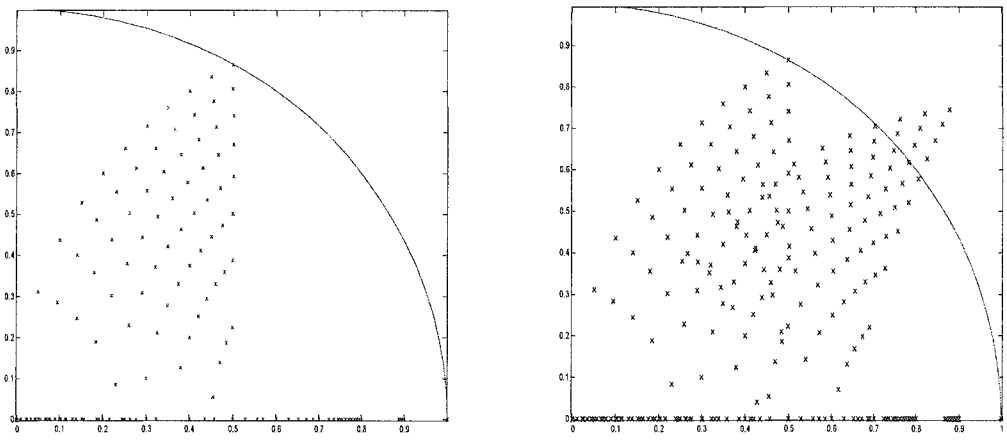
When the eigenvalues both locate within the unit circle, for example $\alpha = 0.5$ and $\beta = 0$, the inventory behaviour is stable, as in Figure 4.6. The hub inventory increases to the desired level and remains at the stable state in the first 100 months. After the incoming order increases, the hub inventory is consumed and backlogged. Then the hub recovers its inventory to the desired level again, with an exponent rise. The factory inventory behaves in a similar way and gains little oscillation.



(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.6 *Hub effective inventory and factory effective inventory when $\theta = 0.75$, $\alpha = 0.3, \beta = 0$ and $Q = 800$. The figure shows the response of the hub effective inventory and factory effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

To clarify the difference between the unmodified original model and the production delayed model, the constellation plots in the first quadrants for both models are listed.



(a)Original model without delay *(b) Model with two steps production delays*

Figure 4.7 *Eigenvalues of model without delay and with production delays*

Inspection of the figures indicates that the most obvious difference between Figures 4.7 (a) & (b) is the appearance of a second 'shark fin' of potentially unstable, complex

eigenvalues. After closer inspection, the fin in Figure 4.7 (a) is revealed as exactly the same as the left fin in Figure 4.7(b). It can be deduced that the original set of complex eigenvalues (seen in Figure 4.7(a) and as the right 'shark fin' in Figure 4.7(b)) must be attributable to the hub only, since their behaviour and locations are totally unaffected by the introduction of the two additional factory states. Moreover, it must be the factory, with its two additional delays, that is the source of the potentially unstable behaviour.

As evidence of the statement above, the eigenvalues constellation for the hub and factory have been plotted separately. At first, the parameters in the factory are fixed at α=0.8 and β=0. The production equation becomes

$$F_{prod}(t) = \max(0, 0.8\,(Q - F_{inv}(t) + F_{blk}(t)) + H_{req}(t+2)\,) \qquad (4.13)$$

The system eigenvalues are then calculated as both α and β incremented from 0 to 1 in steps of 0.1 for the hub planning equation. The system matrix A has changed.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\alpha & -\alpha\beta & \theta & -\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & -0.8 & 0.8 & \theta & -0.8 & -0.8 & 0 & 0 & -0.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (4.14)$$

The hub eigenvalues are plotted and compared to the original model in Figure 4.8. The eigenvalues are plotted with a unit circle on Z-plane (each eigenvalue, represented by an 'x').

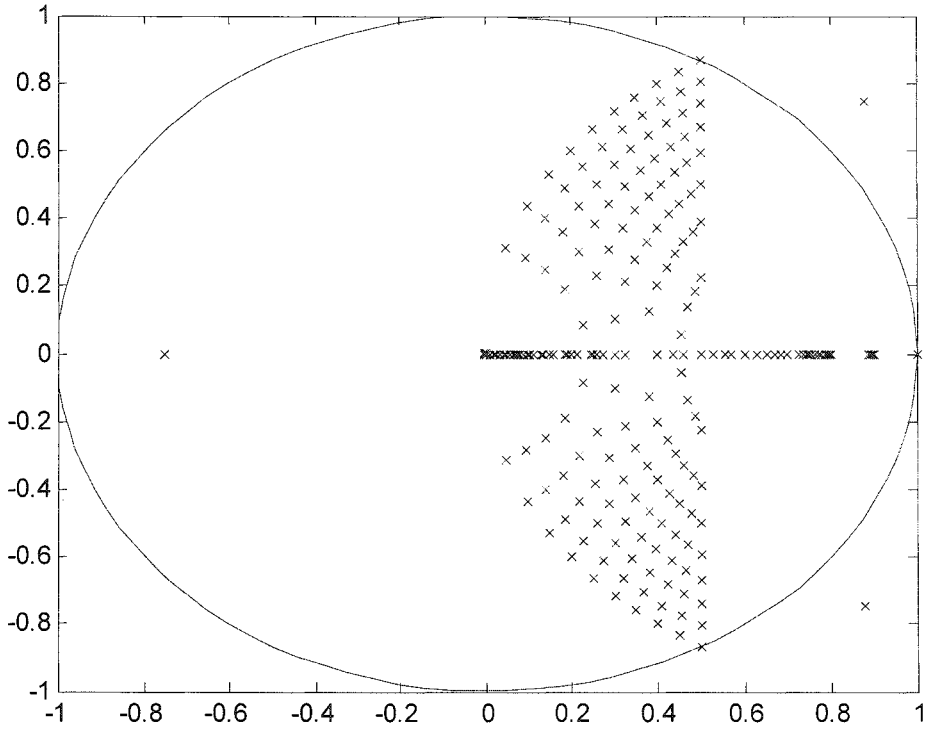


Figure 4.8 *The constellation of eigenvalues of A plotted on the z-plane for θ = 0.75, α = 0 to 1 in steps of 0.1 & β = 0 to 1 in steps of 0.1 for the hub planning equation and fixed at α =0.8, β=0 for factory production.*

In Figure 4.8, the eigenvalues constellation is represented by a stable 'shark fin', which appears in the same location as the 'left fin' in Figure 4.3. If the crosses located outside of the circle and those on the negative x-axis are ignored, the constellation of Figure 4.8 is exactly the same as the constellation for the unmodified original model in Figure 4.7(a). Those three crosses seperate from the 'fin' are the three eigenvalues generated by the factory. Therefore, it can be concluded that the stable fin represents the dynamics of the hub only. According to the above analysis, the hub in the production delayed model could achieve a stable inventory performance through the same decision policy as in the original model. In other words, hub dynamics are not influenced by factory production delay. This sounds contradictory. However, further analysis supports this finding.

The factory eigenvalue constellation is produced in a similar way to which α =0.8, β=0 are fixed in the hub planning. The hub requirement equation becomes

$$H_{req}(t+2) = \max(0, 0.8\,(Q - H_{inv}(t) + H_{blk}(t)) + H_{forecast}(t+2)); \qquad (4.15)$$

The eigenvalues are plotted with a unit circle on the Z-plane (each eigenvalue, represented by an 'x') as both α and β are incremented from 0 to 1 in steps of 0.1 in the factory production equation. The system matrix A changes to

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.8 & 0 & \theta & -0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & -\alpha & \alpha - \alpha\beta & \theta & -\alpha & -\alpha & -\alpha\beta & -\alpha\beta & -\alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (4.16)$$

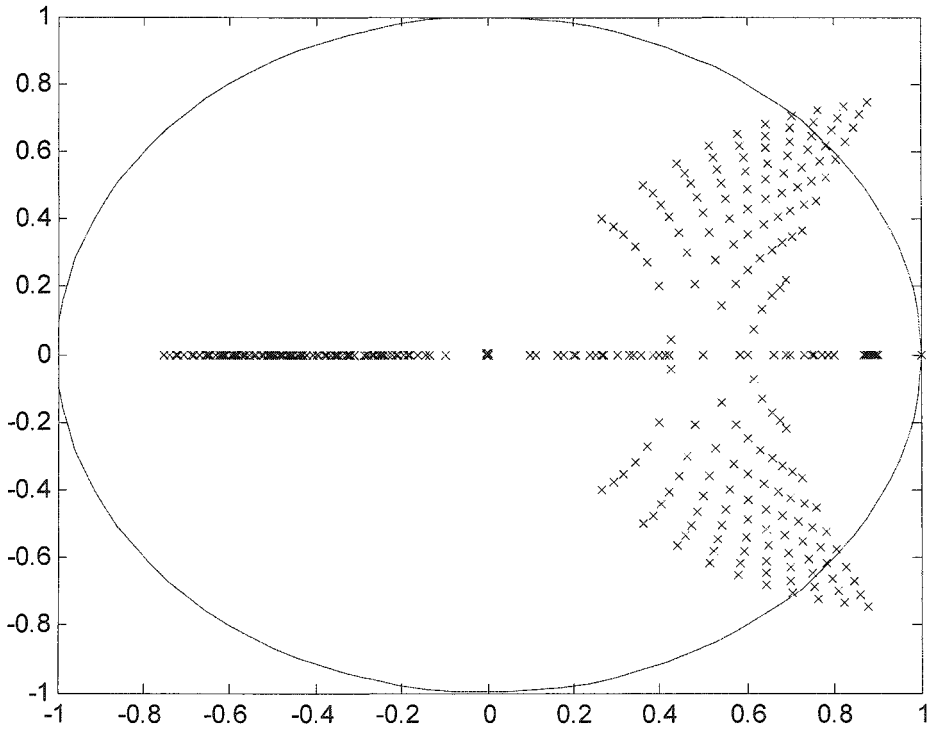
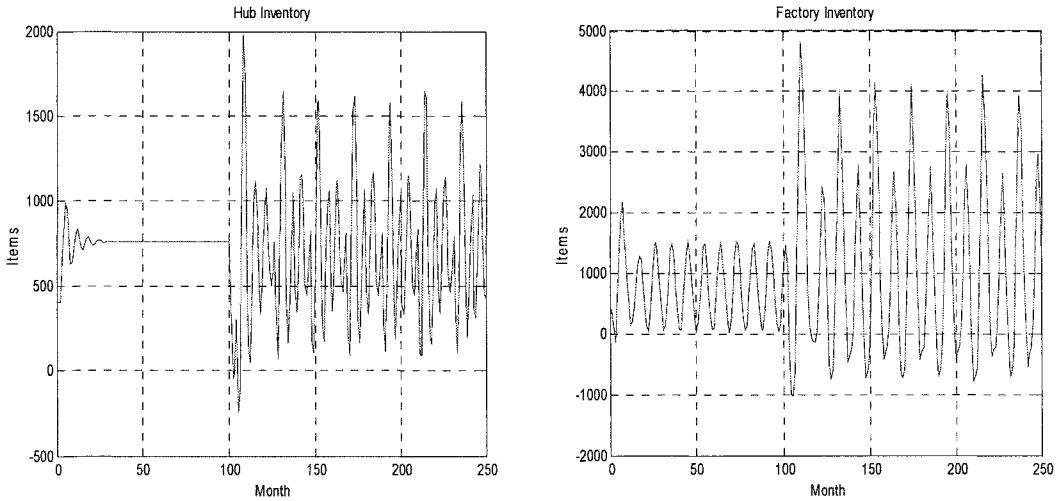


Figure 4.9 *The constellation of eigenvalues of A plotted on the z-plane for θ = 0.75, α = 0 to 1 in steps of 0.1 & β = 0 to 1 in steps of 0.1 for the factory production equation and fixed at α =0.8, β=0 for hub requirement*

In Figure 4.9, the eigenvalues constellation is the same as the 'unstable fin' or 'right fin' in Figure 4.3. As the parameters for the hub planning have been fixed, the figure represents the dynamics of the factory part only. The hub eigenvalues at α =0.8, β=0 have been plotted within the circle but covered by the factory eigenvalues.

4.2.1 The Influence of 'Saturation' Effects

Since Figure 4.3 predicts an unstable factory eigenvalue and a stable hub eigenvalue for the condition, $\theta = 0.75$, $\alpha = 0.8$, $\beta = 0.1$ & $Q = 800$, the 'unstable' transient phenomenon observed after the first 100 months of the response in Figure 4.10(a) is somewhat puzzling. However, as it has been suggested that the unstable eigenvalue pair can be related to the factory section of the model (and not the hub whose response is seen in Figure 4.10(a)), then it is in the factory's responses that we need to look for an answer.

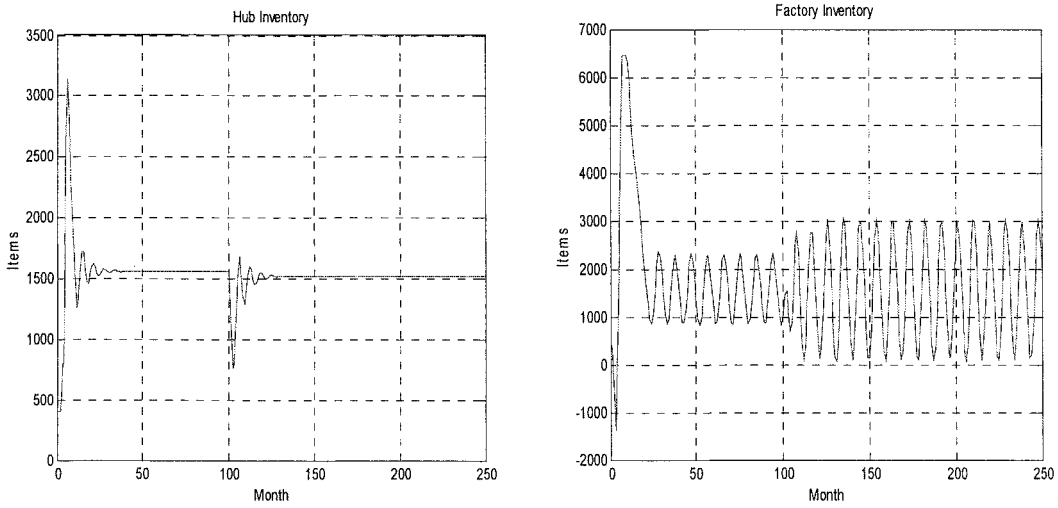


(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.10 *Hub effective inventory and factory effective inventory for $\theta = 0.75$, $\alpha = 0.8, \beta = 0.1$ and $Q = 800$. The figure shows the response of the hub effective inventory and factory effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

As can be seen in Figure 4.10(b) the factory effective inventory response is oscillatory over the whole timeframe. This oscillation is the result of the large signal (saturation

type) nonlinearities defined in Equations 4.1 to 4.4 limiting the growth of the small signal instability defined by the eigenvalues. What is interesting, however, is that prior to the increase in customer orders (from 400 units to 800 units) the factory effective inventory is always positive and, therefore, the hub demand can be met. After the increase in hub orders the factory is no longer able to meet the required demand when the desired safety stock (Q) is 800 units. Hence, the oscillations seen in the factory response cascade down to the hub. Figures 4.11(a) and (b) show the equivalent response when Q is doubled to 1600 units.

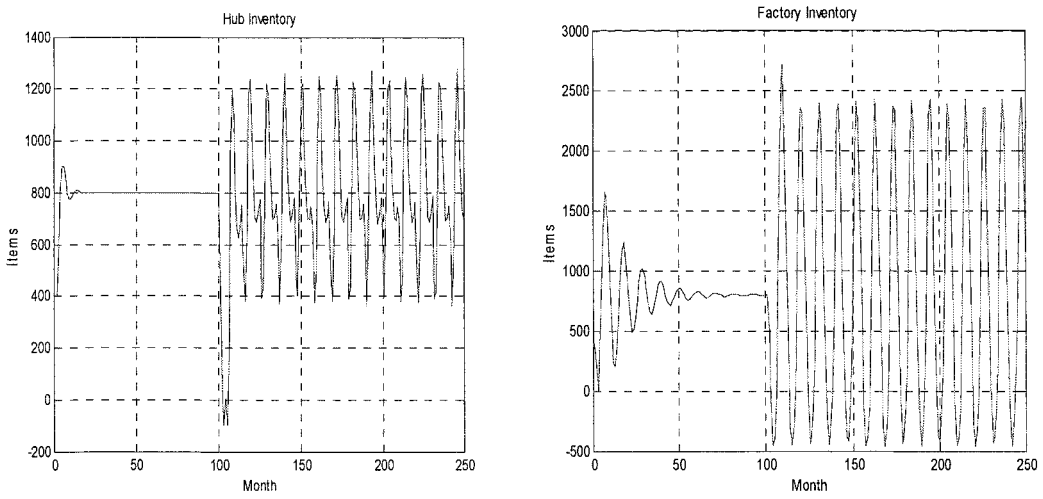


(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.11 *Hub effective inventory and factory effective inventory for θ = 0.75, α = 0.8, β = 0.1 and Q = 1600. The figure shows the response of the hub effective inventory and factory effective inventory over a 250 month period. The initial hub inventory is 400 and the hub orders are held constant at400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

As can be seen in Figure 4.11, when Q = 1600 the factory effective inventory remains positive over the whole time frame. Hence, when viewed from the 'isolation' of the hub, the system appears perfectly stable and hub stock is able to settle to a steady value.

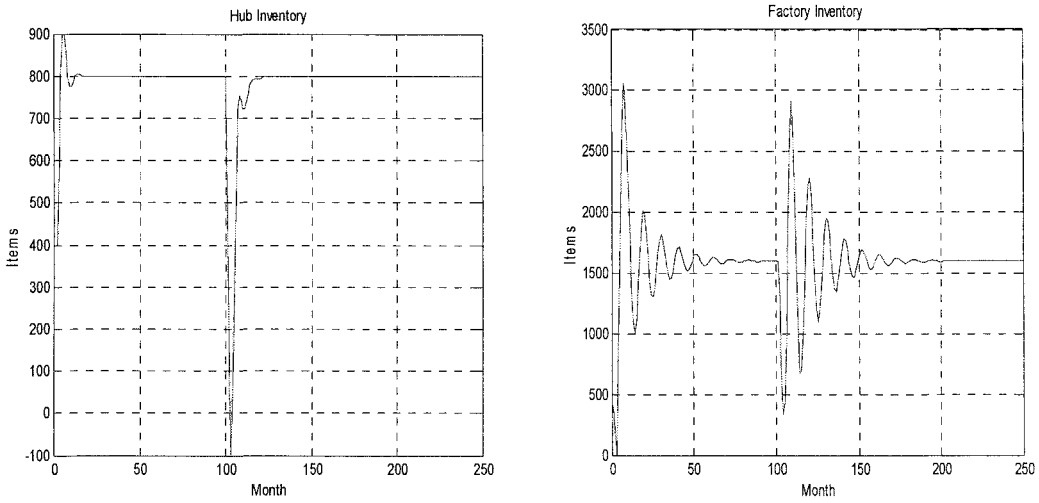
Furthermore, according to the eigenvalues analysis, both the factory and hub should display a stable inventory behaviour when θ = 0.75, α = 0.5, β = 0 & Q = 800. However in Figure 4.12, the plots of the inventory response show unstable behaviour.



(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.12 *Hub effective inventory and factory effective inventory for $\theta = 0.75$, $\alpha = 0.5$, $\beta = 0$ and $Q = 800$. The figure shows the response of the hub effective inventory and factory effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

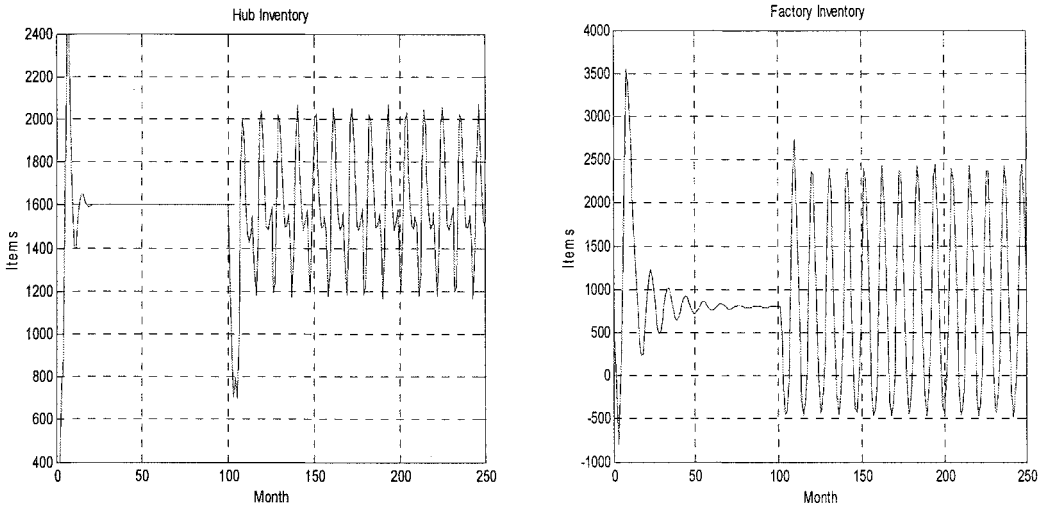
Both the hub and factory inventories are stable at the desired level of 800 units in the first 100 months and start oscillating thereafter. Closer inspection of the factory inventory in Figure 4.12 (b) reveals that the oscillation goes down to a negative value, implying backlog. For the safety stock Q at 800 units, the factory cannot satisfy the hub demand when the sales orders are doubled. To clarify this saturation effects in different safety stock scenario, keep the hub inventory safety stock (or desired inventory) at 800 units and double the factory inventory safety stock to 1600 units. In this case, the hub will place the same amount of requirement on the factory as before, but the factory inventory has improved its safety stock level. When the hub demand increased, the factory inventory gained more inventory space to respond to this increase. Production is also improved to compensate the factory inventory for this increased safety stock level. Increased factory inventory level and production have the effect of diminishing facotry backlog.



(a) Hub effective inventory Q=800 *(b) Factory effective inventory Q=1600*

Figure 4.13 Hub *effective inventory and factory effective inventory for* $\theta = 0.75$, $\alpha = 0.5$, $\beta = 0$. *The figure shows the response of the hub effective inventory and factory effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

The factory effective inventory remains positive over the whole simulation in Figure 4.13(b) when Q =1600 units. The hub requirement could be fully satisfied. The hub inventory becomes stable at its desired level of 800 units as it should be based on the eigenvalue analysis. Conversely, increase the hub safety stock to 1600 units and keep the factory safety stock at 800 units and the hub will require more incoming shipments to compensate its inventory on the safety level and feed the increased sales order. In this case, the factory may get into an even worse situation. They have to react very quickly to feed the increasing sales demand and the increased hub inventory replenishments.



(a) Hub effective inventory Q=1600 *(b) Factory effective inventory Q=800*

Figure 4.14 *Hub effective inventory and factory effective inventory for $\theta = 0.75$, $\alpha = 0.5$, $\beta = 0$. The figure shows the response of the hub effective inventory and factory effective inventory over a 250-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months these are then increased to 800/month for the remainder of the simulation*

On the other hand, if we keep the factory desired inventory is kept at 800 units and the hub safety stock increased to 1600 units, both the factory and hub inventory as shown in Figure 4.14 oscillate in a similar pattern to that in Figure 4.12. After the sales order increases, the factory does not have enough inventories to cover the demand. The system produces a negative factory effective inventory. As a result of the factory backlog, the hub inventory starts oscillating around the safety level of 1600 units, as in Figure 4.14 (a).

To sum up, the backlogs in the factory are expressions of saturation effects. They are the key to illustrating the oscillations, which cannot be explained by eigenvalue constellation. To avoid such saturation effects, the safety stock can be increased and more inventory space created in the factory in order to remove oscillations and backlogs.

4.2.2 The Influence of α

With two extra production delays, the system generates two more eigenvalues and more complicated eigenvalue movements. Figure 4.15 shows the eigenvalue locus when β = 0 and varying α varies from 0 to 1. There are three sets of eigenvalues in the figure, which are represented by green, black and red. When α=0 and β = 0, the corresponding green eigenvalues start from the original. Both of the red and black eigenvlaues have two points. One is on the original (0, 0) and the other with a coordinate (1, 0) at the end of the positive real axis.

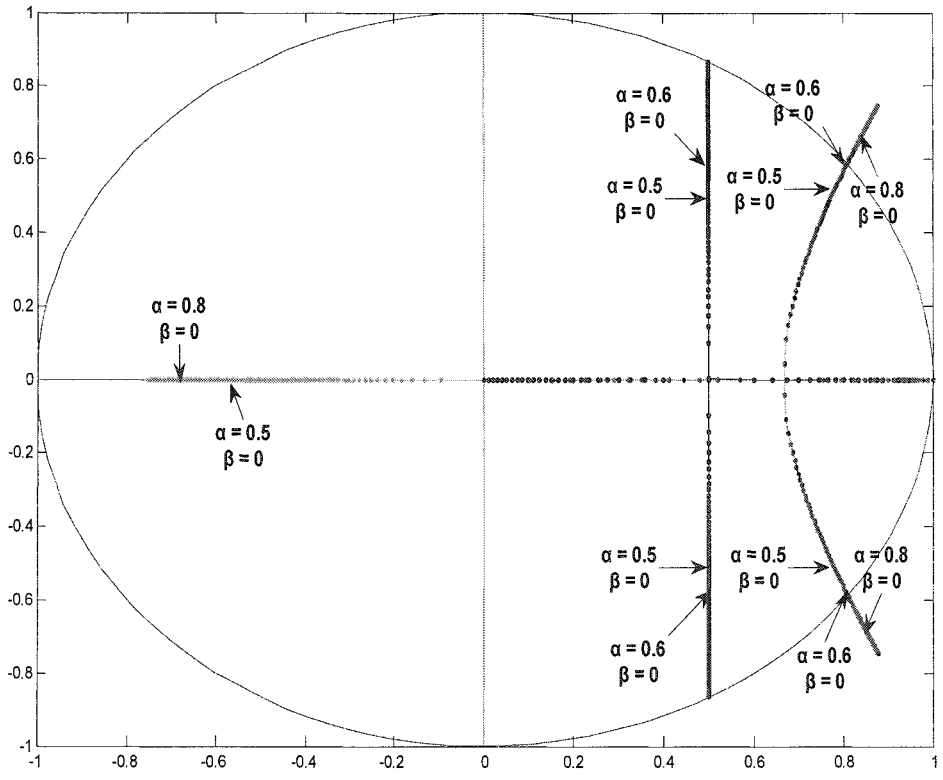


Figure 4.15 *Locus of eigenvalues plotted on z-plane for θ = 0.75, β = 0 and α varied from 0 to 1. Clearly, the complex pairs migrate toward the edge of the unit circle as α increases.*

As α increases, the green points move left along the negative real axis; the red points start moving toward each other from both sides of the positive axis. These two red points meet each other when α = 0.14. Then with little increment of α, the eigenvalues

jump out of the real axis and dive into the first and fourth quadrants above and below the real axis. As discussed, the red eigenvalues represent the factory inventory dynamics. Since the complex value of red point gain an imaginary part when α is bigger than 0.14, the factory effective inventory starts oscillating with decreasing amplitude and finally becomes steady. Keep increasing α to 0.6 and the red points follow the red curve in the figure toward the edge of the unit circle. When the points locate on the circle, the factory inventory oscillates with a fixed rate of amplitude. When α=1, the points arrive at the end of the red curve. The oscillation amplitude of the factory inventory is expanding. The oscillation hit the saturation effects boundary. The inventory oscillates with different amplitude thereafter. On the other hand, the black points representing the hub dynamics follow the same movement pattern as the red points. The two black points meet each other on the positive real axis when α = 0.25. They reach the edge of the unit circle when α = 1. To achhieve a better understanding of eigenvalue and saturation related nonlinear dynamics, phase plots of the hub and factory inventory are used to analyse the inventory dynamics.

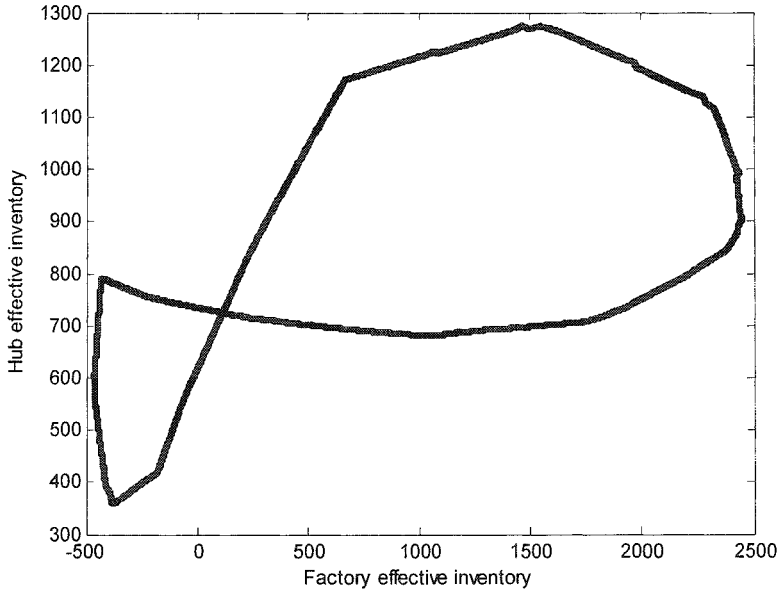


Figure 4.16 *Quasi-periodic solution obtained for α = 0.5, β =0, θ = 0.75, Q =800.The phase plot shows corresponding values of hub and factory effective inventories afterr ten thousand iterations of the production delayed model*

Figure 4.16 shows the phase plots obtained at $\alpha = 0.5$, $\beta = 0$.The eigenvalues locate within the unit circle. The hub effective inventory is plotted as a function of the factory effective inventory over ten thousand simulation iterations. The points in the phase plot fall along a two-dimensional curve and are indicative of quasi-periodic behaviour on the surface of a torus. What can be seen in the figure is a cross section of this torus. In the case of α and β, the hub and factory effective inventory always follow around this cross torus during the simulation and are never stable at any point. By calculating the natural frequency for this set of eigenvalues, there are ten points in one period of the oscillation for both the hub and factory inventory. The ten pairs of the inventory quantity locate on the cross torus in different positions. After ten iterations, the hub and factory inventory move back to the old value with a tiny shift. After a very large number of iterations, the cross torus is filled by enormous points projecting the hub and factory effective inventory.

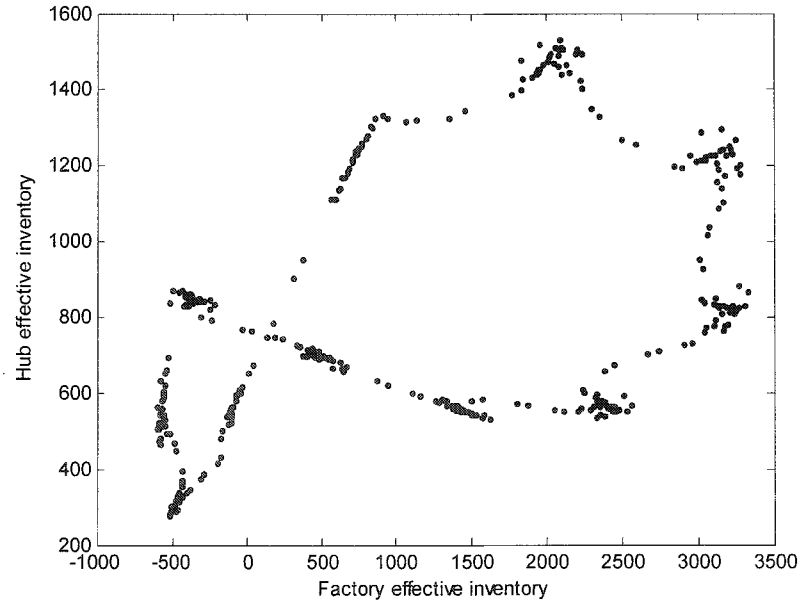


Figure 4.17 *Periodic solution obtained for $\alpha = 0.6$, $\beta = 0$, $\theta = 0.75$, $Q = 800$.The phase plot shows corresponding values of hub and factory effective inventories after ten thousand iterations of the production delayed model*

By increasing α to 0.6 and fixing β at zero, the eigenvalues are plotted on the edge of the unit circle, as shown in Figure 4.15. The phase plots of the hub effective inventory against the factory effective inventory are shown as a periodic map in Figure 4.17. In

this case of α and β, each period of oscillation for the hub and factory inventory includes 11 points. Compared with quasi-periodic behaviour, the inventory points shift in a larger distance after one period. After a number of iterations, the simulation repeats the same quantity of hub inventory and factory inventory. Therefore, this periodic map is built up by many disconnected points. However, the overall shape of the periodic map is still similar to the quasi-periodic behaviour map in Figure 4.16.

Keep increasing α to 1, and the eigenvalues locate on both corners of the two fins. The corresponding phase plot is shown in Figure 4.18 as chaotic behaviour. Compared with Figure 4.17 and 4.16, the behaviour in Figure 4.18 is obviously not periodic or quasi-periodic. The points no longer follow a two dimensional curve, but the iterations produce a folded band, somewhat like a bird.

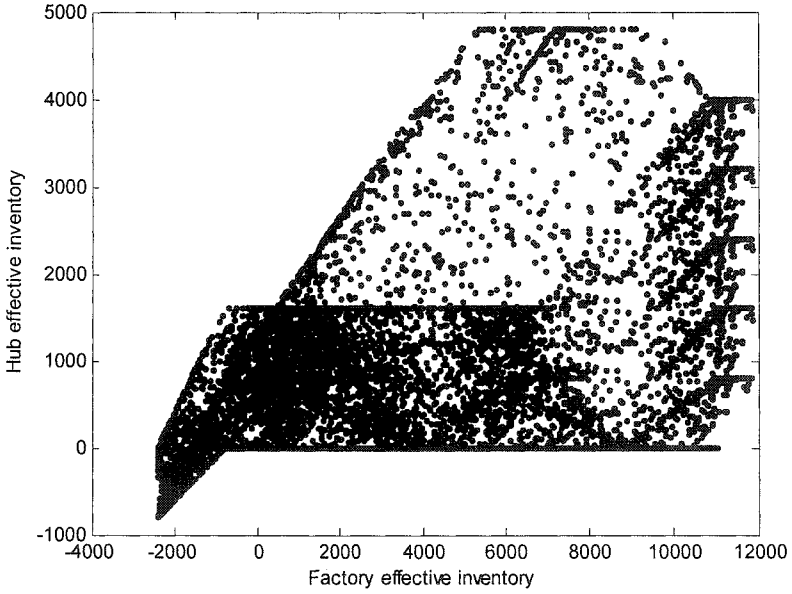


Figure 4.18 *Chaotic behaviour observed in the production delay model for $\alpha = 1\ \beta = 0$, $\theta = 0.75$, $Q = 800$. The phase plot shows corresponding values of hub and factory effective inventories after ten thousand iterations of the production delayed model. More detailed simulations show that the Lliapunov exponent is positive.*

The chaos in physics is defined as "aperiodic long term behaviour in a deterministic system that exhibits sensitive dependence on initial conditions." (Steven, 2001). "Aperiodic long term behaviour" means that there are trajectories, which do not settle down to fixed points, periodic orbits, or quasi-periodic orbits. "Deterministic" means

that the system has no random or noisy inputs or parameters. For example, the input matrix in equation (4.10) is a unit step input and parameters α , β, θ and Q are constant. There are no noisy inputs. Hence, the irregular behaviour arises from the system's nonlinearity, rather than from noisy driving forces. Finally, "sensitive dependence on initial conditions" means that nearby trajectories separate exponentially fast. To prove this statement, the initial value of the state factors in equation 4.9 are changed with very small excursions and the simulation then run for ten thousand iterations. The hub inventory level is recorded in each round. The logarithm hub inventory difference between the model with old initialization and the model with changed initialization in each iteration is plotted in Figure 4.19.

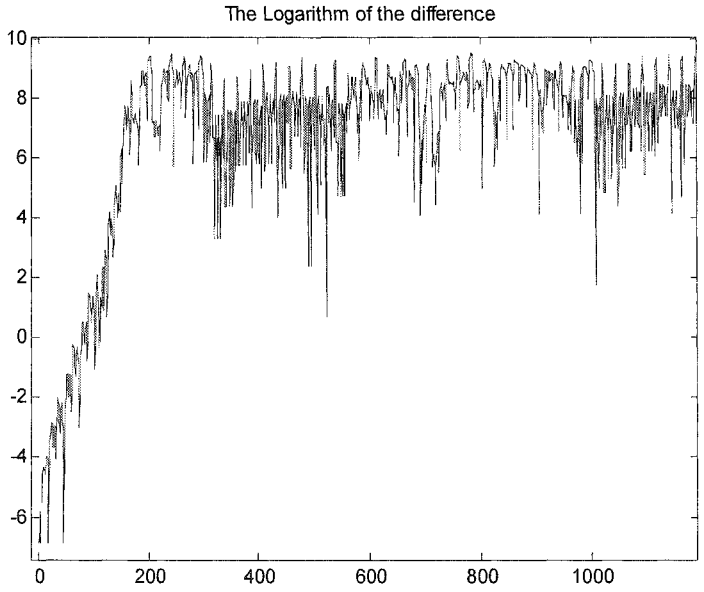


Figure 4.19 *Logarithm difference of hub effective inventory between the production delayed model starting with initial value and the model starting with very small excursions of the initial value for* $\alpha = 1$, $\beta = 0$, $\theta = 0.75$, $Q = 800$.

The curve is never exactly straight. It has wiggles because the strength of the exponential divergence varies along the chaotic map. The divergence between two models with slightly different initialization increases linearly in the first 200 time units, oscillating around 9 thereafter. The oscillation is never over 10 because the trajectories obviously cannot become any further apart than the diameter of the chaos map. The slope of the first 200 time units is called the Liapunov exponent (λ). According to the statement "sensitive dependence on initial conditions", the chaos system should have a

positive Liapunov exponent. Suppose the state starting with unchanged initialization is hinv (t) and the state starting with excursion initialization is hinv(t)+$\delta(t)$, for a chaotic behaviour, $\delta(t)$ grows like $\delta(t)=\delta_0\, e^{\lambda t}$. Then the λ can be calculated as

$$\lambda \approx \frac{1}{n} \ln \left| \frac{\delta_n}{\delta_0} \right| \tag{4.17}$$

The Liapunov exponent λ for $\alpha = 1$, $\beta = 0$, $\theta = 0.75$, $Q = 800$ in the production delayed model is 8.18×10^{-2},which is positive and conforms to the definition of chaos. This chaotic behaviour is the result of eigenvalues and saturation effects, corresponding to α, β and Q. By removing any of the nonlinearity effects such as changing α and β to make the eigenvalues locate inside the circle, or increasing Q to eliminate the saturation effects, the system will not behave as shown in this section.

4.2.3 The Influence of β

Figure 4.20 shows a plot of the eigenvalue locus when $\beta = 0$ and α varies from 0 to 1. Similar to the effect of α, three sets of eigenvalues are plotted. However, the movement of the eigenvalues with changing β is different. All of these three eigenvalues start from the original in coordinate (0, 0). With little decrement of β, both of the black points and red points jump into the first and fourth quadrant immediately. Keep decreasing β to 0.32, the red point arrive at the unit circle. The factory effective inventory oscillates with fixed amplitude. When $\beta = 0$, the eigenvalue locates at the end of curves outside of the unit circle. The green points' movements follow the same pattern as introduced in the last section with decreasing β.

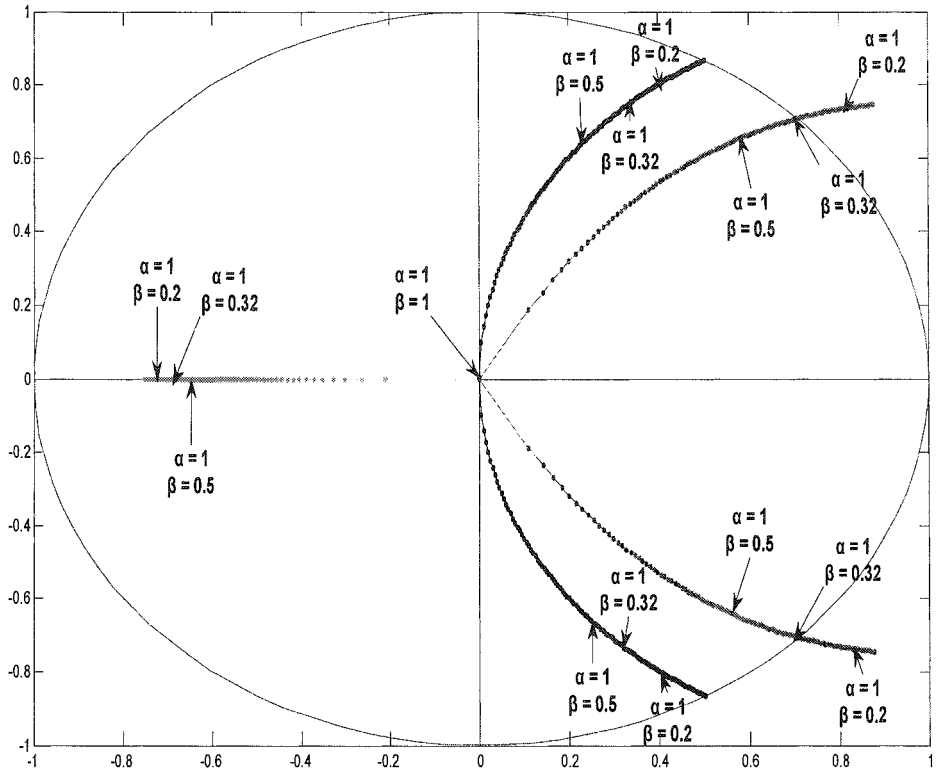


Figure 4.20 *Locus of eigenvalues plotted on z-plane for $\theta = 0.75$, β varied from 0 to 1 and $\alpha = 1$. Clearly, the complex pair migrate toward the centre of the unit circle as β increases.*

It is clear from analysis of the figure that by keeping $\alpha = 1$ and β smaller than one, both the hub and factory effective inventory have oscillations. With big β, the oscillations decay and the model finally arrives at a stable level. With small β, the inventory oscillates with great amplitude. Managers cannot avoid oscillation of the inventory if they want to replenish to the desired inventory level fast. However, they can control the speed and amplitude of oscillation by considering their supply line, which is hub incoming shipment and factory backlog and production delay.

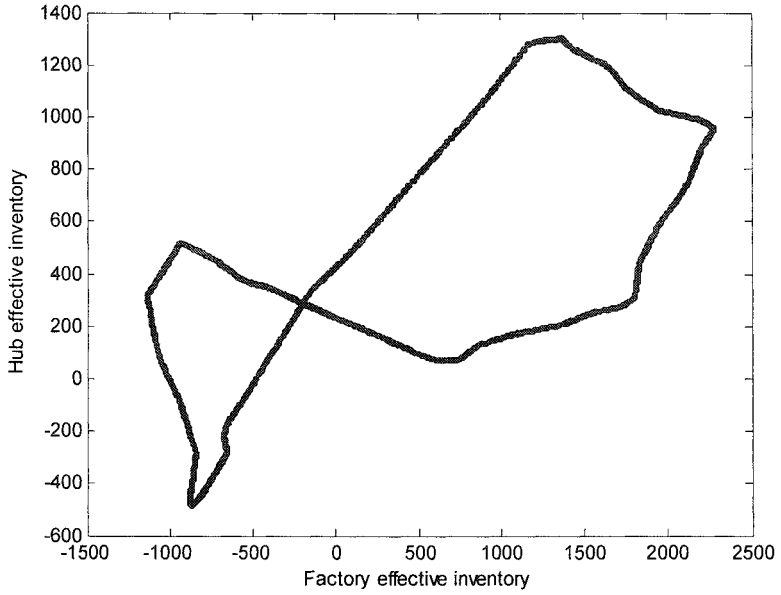


Figure 4.21 *Periodic solution obtained for* $\alpha = 1$, $\beta = 0.4$, $\theta = 0.75$, $Q = 800$. *The phase plot shows corresponding values of hub and factory effective inventories after ten thousand iterations of the production delayed model*

Figure 4.21 shows the phase plot of the hub inventory against the factory inventory. It is a quasi-periodic solution for the eigenvalues locating within the unit circle at $\alpha = 1$ and $\beta = 0.4$. The overall shape of the cross torus in Figure 4.21 is quite similar to that in Figure 4.16. However, as α is one, the model will generate backlogs in both hub and factory. Since both the hub and factory effective inventory have negative value in the plane, the bottom left part of the plane is negative.

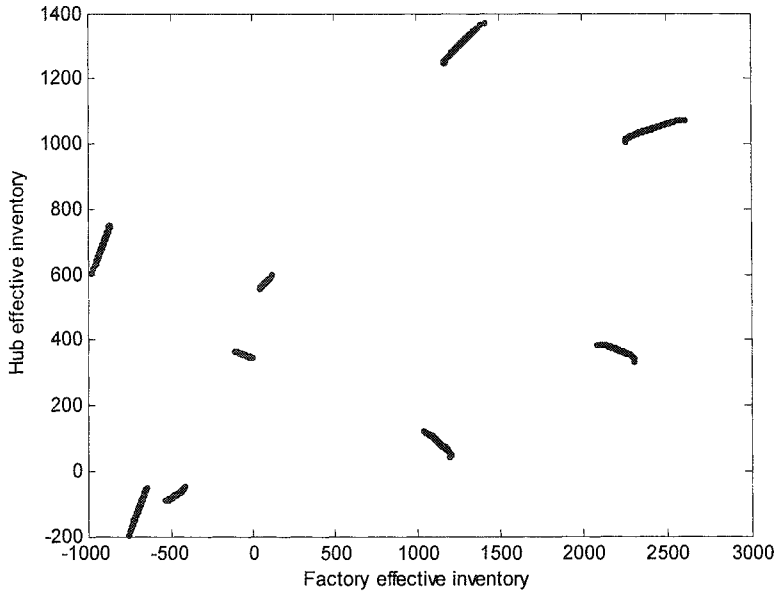


Figure 4.22 *Quas-iperiodic solution obtained for $\theta = 0.75$, $\alpha = 1$, $\beta = 0.32$, $Q = 800$. The phase plot shows corresponding values of hub and factory effective inventories after ten thousand iterations of the production delayed model*

When the eigenvalues locate on the unit circle at $\alpha = 1$, $\beta = 0.32$, a periodic phase plot is produced for the hub and factory effective inventory. Compared with the periodic plots in Figure 4.17, these short strings in the plots are more dispersive. The distance between each string is very great. The corresponding inventory behaviour is unstable. The inventory oscillation switches its amplitude and frequency in a large ratio within a short time.

4.2.4 The Primary Route to Instability

To better understand the behaviour observed, the structure of the system block diagram will be examined (Figure 4.23). The system block diagram has the form:

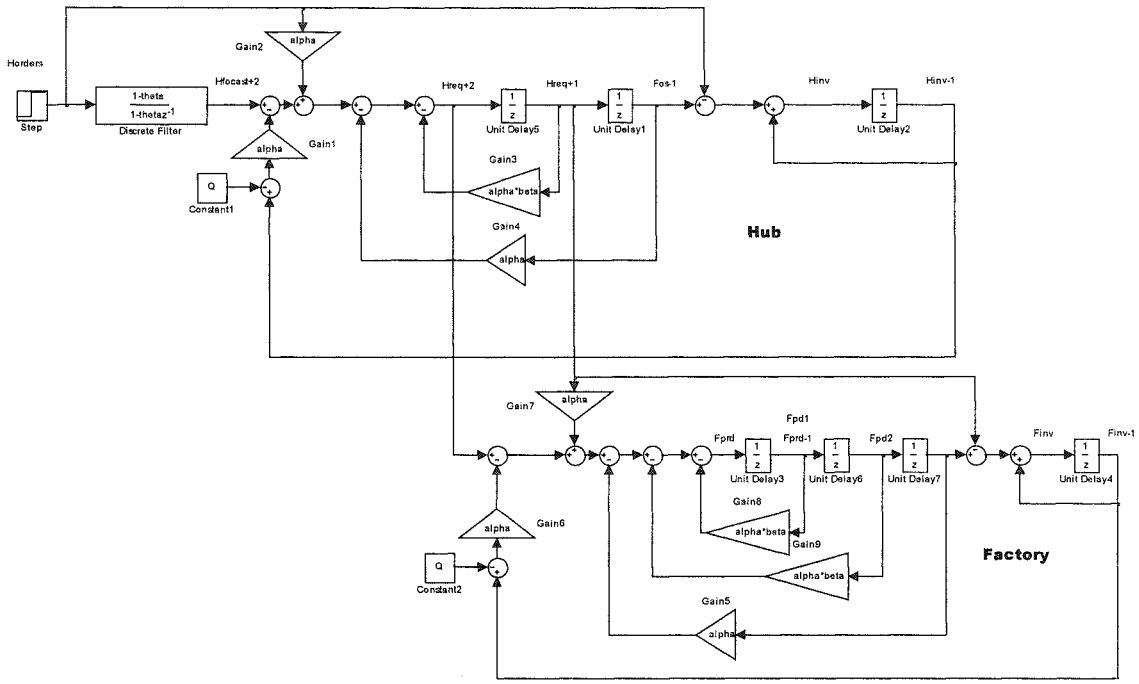


Figure 4.23 *Small signal block diagram representation of the model of the supply chain at Draeger Safety UK, including additional production delays.*

From the figure, it is clear that the factory and the hub still retain their 'isolated' feedback structures. Moreover, closer inspection reveals that the additional production delays only have an effect on the factory 'loop'. There are more feedback loops generated in the factory part, which might produce more complex eigenvalues and complex behaviour compared with the original model. Thus, the complex pairs of eigenvalues previously identified as being associated with the hub remains unaffected and are still clearly visible in Figure 4.15 and Figure 4.20.

Turning our attention to the factory, the characteristic equation for the factory part of the system with additional production delays, for the condition $\beta = 0$, is:

$$(z^3 + \alpha_F)(z - 1) + \alpha_F = 0 \qquad (4.18)$$

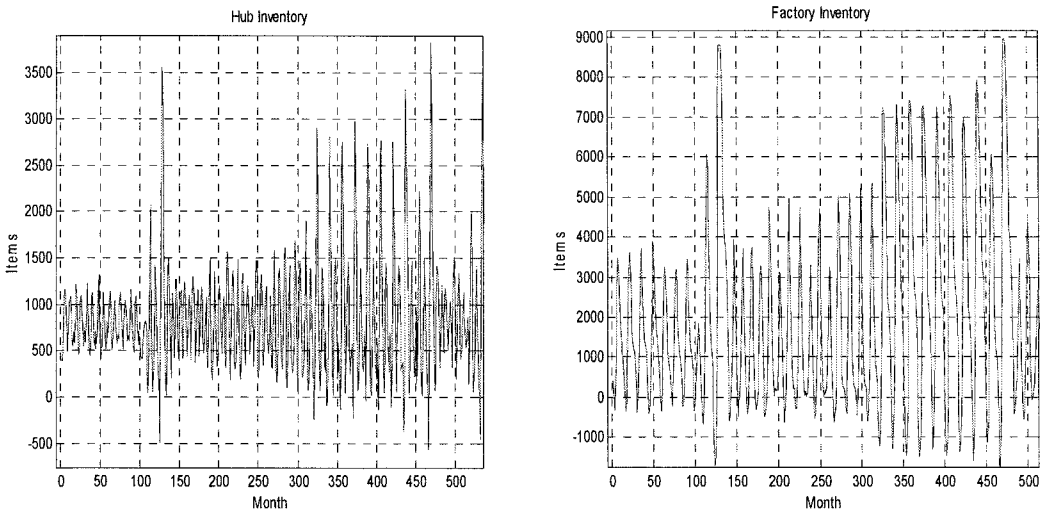
Clearly, Equation 4.18 has four eigenvalues. One of these is at $z = 0$, one exists on the negative real axis and lies in the range $z = 1 \rightarrow -0.755$ as $\alpha = 0 \rightarrow 1$ and the other two form the second complex pair seen in Figure 4.15 and Figure 4.20 (red curve). These become complex when $\alpha > 0.148$ and intersect with the unit circle at a position $0.809 \pm$

j0.5878, for the condition $\alpha = 0.618$. It is obvious, therefore, that it is now the factory that has become the primary route to instability.

4.3 Production constraint on production delayed model

The dynamics of the model with production delay and production constraint are very complicated. To clarify the effects of production constraint on system behaviour, five different quantities of constraints are applied in the model with fixed planning parameters $\alpha = 0.9$, $\beta = 0$. The corresponding system eigenvalues locate outside of the unit circle. The relevant inventory behaviour with no constraint is chaotic.

Applying 5000 units as the production constraint on the factory, the time response of hub and factory effective inventories are plotted respectively (Figure 4.24). The oscillations in both hub and factory inventory appear very random. Figure 4.24 (a) shows that the peak of the oscillation in hub effective inventory is around 3800, which is smaller than the production constraint. Therefore, the constraint has no effect on the dynamics of the system. Figure 4.25 shows the chaotic solution.



(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.24 *Hub effective inventory and factory effective inventory for production constraint = 5000, $\theta = 0.75$, $\alpha = 0.9, \beta = 0$, $Q = 800$. The figure shows the response of the hub effective inventory and factory effective inventory over a 500-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

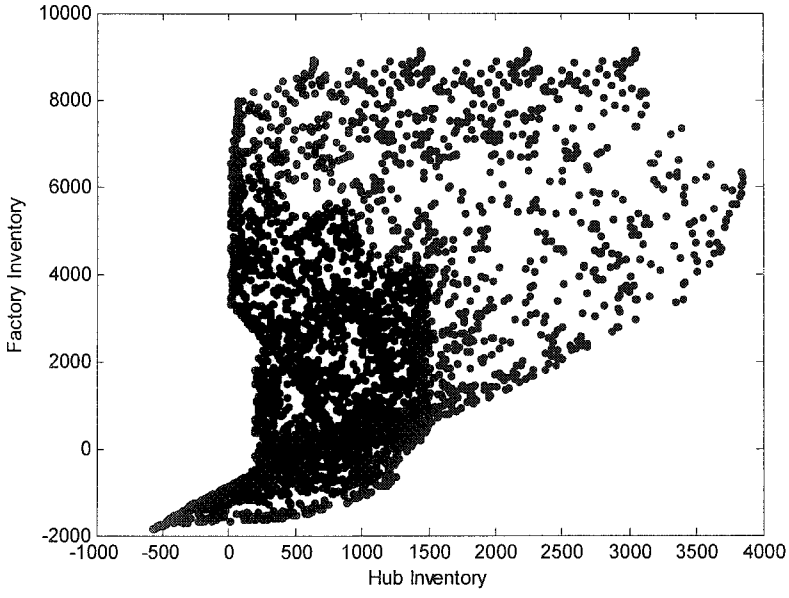
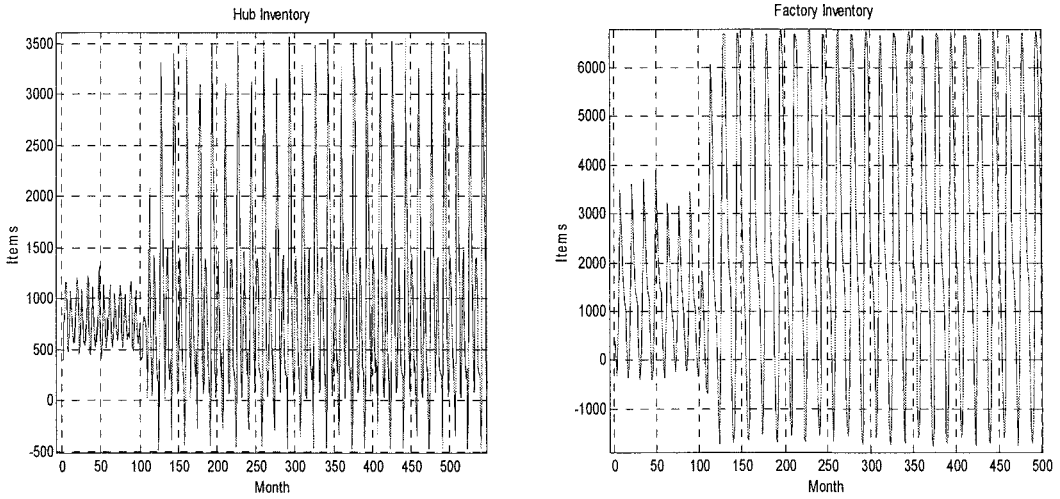


Figure 4.25 *Chaotic solution obtained for Production Constraint = 5000, $\theta = 0.75$, $\alpha = 0.9$, $\beta = 0$, $Q = 800$. The phase plot shows corresponding values of hub and factory effective inventories after five thousands iterations of the production delayed model*

If the production constraint is decreased to 3000, which is smaller than the peak of hub oscillation amplitude in Figure 4.24(a), the hub and factory inventories oscillate with regular amplitude.



(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.26 *Hub effective inventory and factory effective inventory for production constraint = 3000, $\theta = 0.75$, $\alpha = 0.9, \beta = 0$, $Q = 800$. The figure shows the response of the hub effective inventory and factory effective inventory over a 500-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

Figure 4.27 shows the periodic solution of the system by plotting the hub effective inventory against factory effective inventory. These discrete points in the plots represent the different state of the hub and factory inventories. After a certain period of simulation, the hub and factory inventory move back to the same state and repeat a similar oscillation.

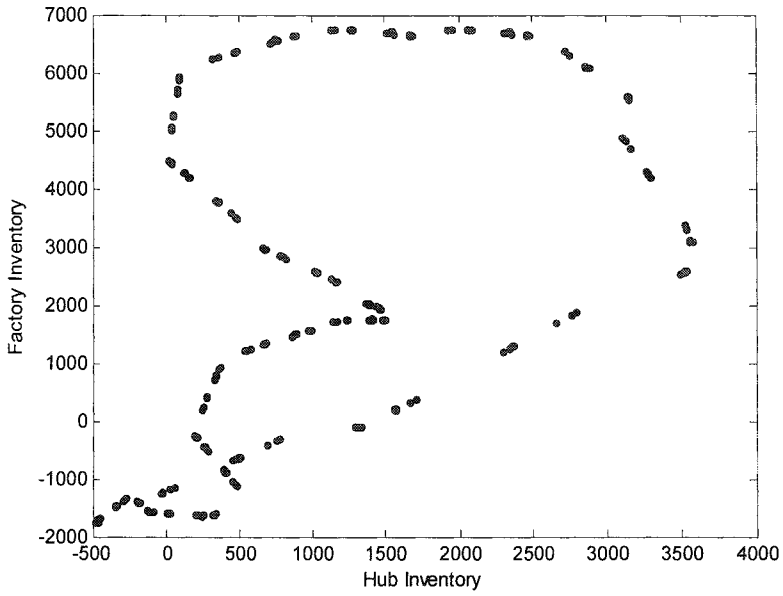


Figure 4.27 *Quasi-periodic solution obtained for Production Constraint = 3000, θ = 0.75, α = 0.9, β =0, Q =800.The phase plot shows corresponding values of hub and factory effective inventories after five thousand iterations of the production delayed model*

By continuing to decrease the production constraint to 1000 units and 900 units, a quasi-periodic solution is produced which looks like a 'bird's wing' in both cases (Figure 4.28, Figure 4.30).

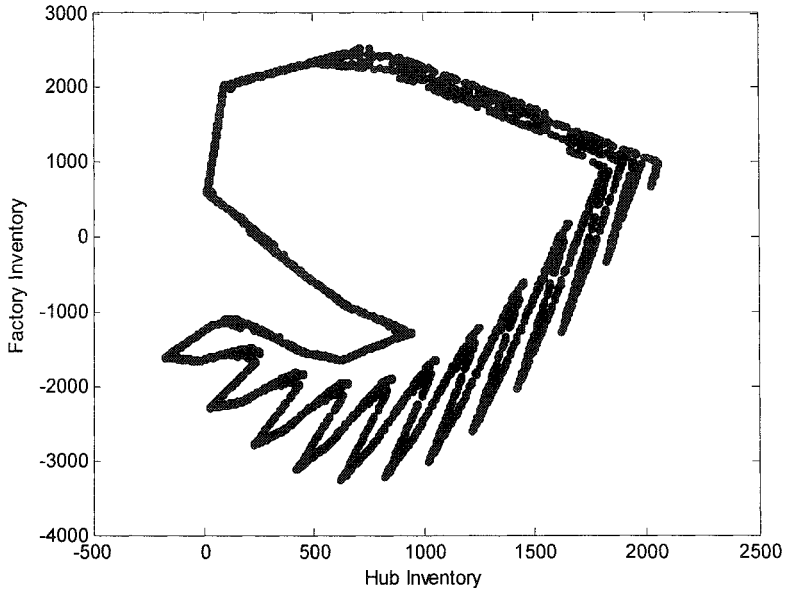
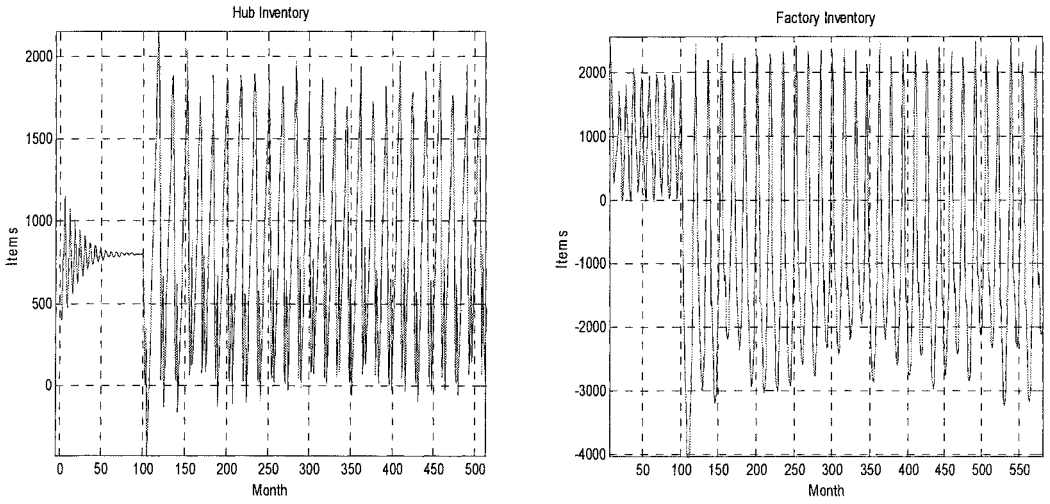


Figure 4.28 *Periodic solution obtained for production constraint = 1000, θ = 0.75, α = 0.9, β =0, Q =800.The phase plot shows corresponding values of hub and factory effective inventories after five thousand iterations of the production delayed model*



(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.29 *Hub effective inventory and factory effective inventory for production constraint = 1000, θ = 0.75, α = 0.9,β =0, Q =800. The figure shows the response of the hub effective inventory and factory effective inventory over a 500-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

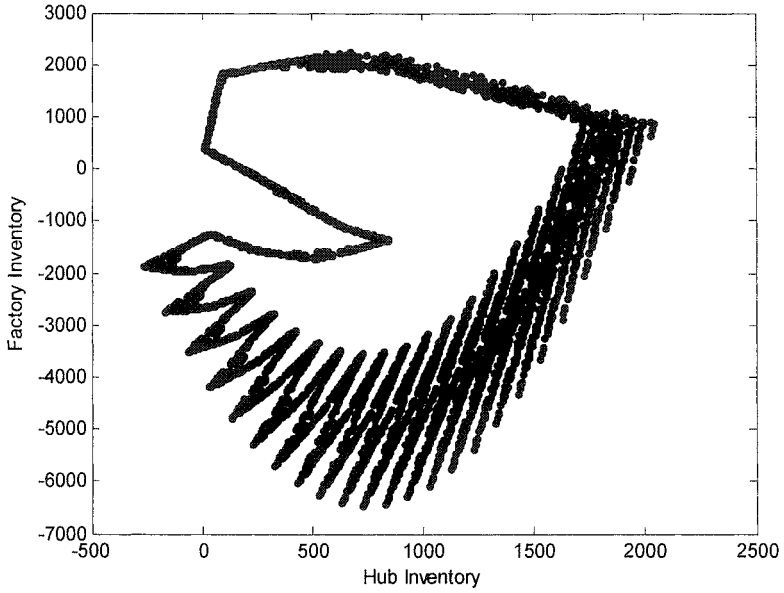
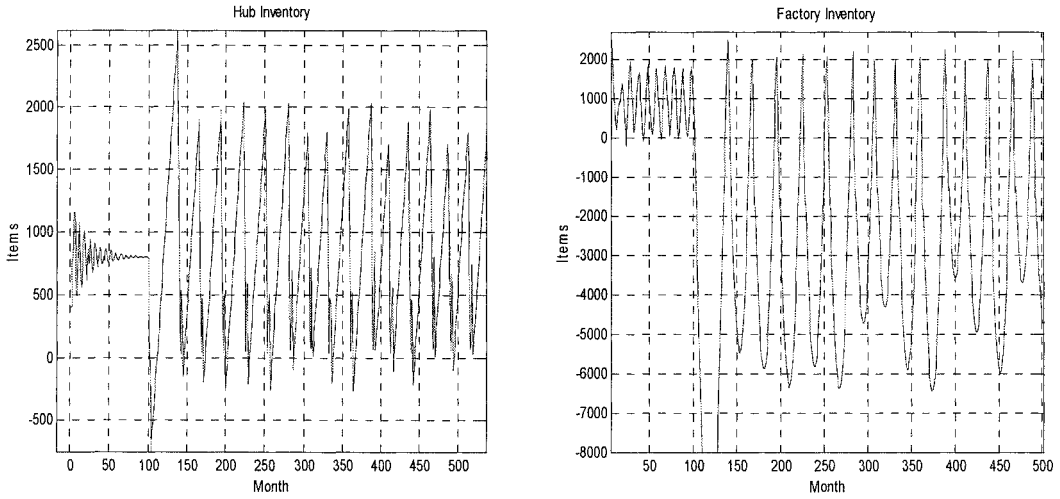


Figure 4.30 *Periodic solution obtained for Production Constraint = 900, θ = 0.75, α = 0.9, β =0, Q =800.The phase plot shows corresponding values of hub and factory effective inventories after five thousand iterations of the production delayed model*



(a) Hub effective inventory *(b) Factory effective inventory*

Figure 4.31 *Hub effective inventory and factory effective inventory for production constraint = 900, θ = 0.75, α = 0.9, β =0, Q =800. The figure shows the response of the hub effective inventory and factory effective inventory over a 500-month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. Tthese are then increased to 800/month for the remainder of the simulation*

Close inspections of two figures show that the 'wing' in Figure 4.28 has 12 'fingers' while the 'wing' in Figure 4.30 has 24 'fingers'. The effects of the 'fingers' are illustrated in Figure 4.29 and Figure 4.31. The inventory oscillations in Figure 4.29 are denser compared with those in Figure 4.31. There are 14 and 18 points in a single wave of hub and factory oscillations in Figure 4.29, while there are 25 and 28 points in a single wave of hub and factory oscillations in Figure 4.31. The oscillation frequency had become higher in this case.

If the production constraint equals the sales order, the factory can only supply the sales demand. The hub inventory can never be compensated to the desired level. In this situation, the hub keeps sending the hub requirements to replenish the hub inventory. The hub requirements are backlogged in the factory. The factory effective inventory become negative and keeps decreasing as the simulation runs (Figure 4.32).

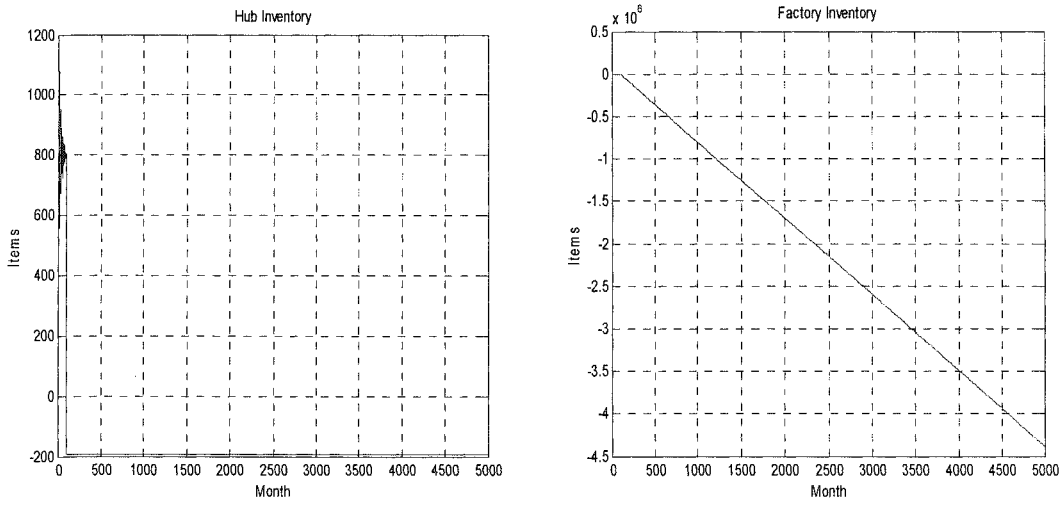


Figure 4.32 *Chaotic solution obtained for Production Constraint = 800, θ = 0.75, α = 0.9, β =0, Q =800.The phase plot shows corresponding values of hub and factory effective inventories after five thousands iterations of the production delayed model.*

4.4 Conclusion

Based on the analysis given above, it is clear that the replenishing inventory rate α has a destabilising effect, while consideration of the past decision rate β has a stabilising effect on the dynamics of this production delayed supply chain model. The extra production delay has made the system more sensitive to management decisions. Compared with the original model, the production delayed model could be unstable owing to the influence of the saturation effect even the local eigenvalues locating inside the unit circle. Instead of employing conservative management policy and sacrificing the customer service and on time shipping to eliminate oscillation, managers have the flexible option of improving the safety stock Q to stabilize the supply chain and achieve on time delivery. However, the warehouse has then to pay higher costs for holding the extra mount of safety stock. It is a trade off for planning managers: risk on time delivery or have a high volume of warehouse stock. Decisions can be made which reflect different occasions and objections in order to maximise the benefits and minimise the costs.

Production constraint impacts can be explored when the model is unstable. With a different degree of constraint, the model can display chaotic, quasi-periodic and periodic behaviour. With the introduction of the two additional lead time states, it is the factory which provides the primary route toward instablility. In this situation, the hub can do little against poor management decisions in the factory.

Chapter 5:

Hub Model with One Additional Delay in Planning

In this chapter, the effects of introducing an additional information delay into the communication between hub and factory are investigated. In terms of the present analysis, the delay represents two likely scenarios of 'getting the forecast wrong' and 'compatibility problem with different planning systems'. In the case of Draeger, the factory in the UK and the hub in Germany use different versions of SAP as their planning tool. The compatibility problem of the planning software causes communication problems so that they cannot fully share planning information. Discovering and sorting out these compatibility problems will delay the planning processing.

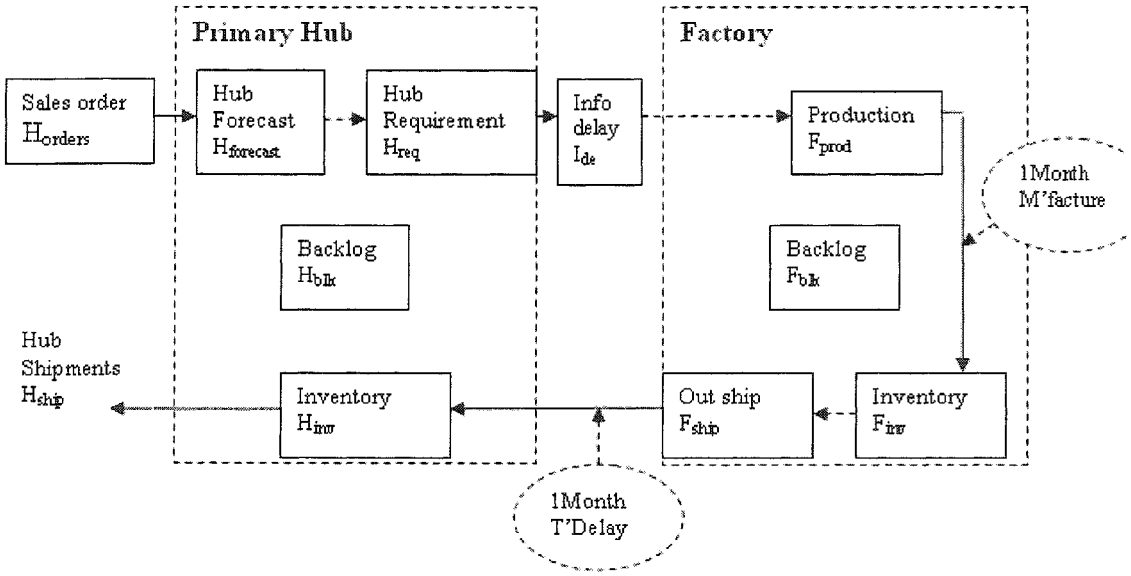


Figure 5.1 *Schematic diagram showing relationships between factory and hub including 1 month additional communication delay.*

In this modification, an additional information delay state, Info delay $I_{de}(t)$, is introduced between the hub requirement and the factory production. Figure 5.1 shows the schematic diagram of the planning delay model. By introducing this one step planning delay, hub planning factory production planning are not working in synchronization. The factory production planning is delayed. Assuming the hub receives sales orders in January and generates two month ahead forecasting; two month ahead hub requirements are produced. The info delay is introduced after generating hub requirements. Therefore, the hub requirements for March are kept in the $I_{de}(t)$ for a month. When the hub requirements for March move into the factory, the simulation has moved to February already. In this way, the factory makes a production plan and starts producing the March hub requirements in February. Now it can be seen that the two month ahead planning in the old model has been changed to one month ahead planning. After one month production, the finished goods stored in the factory are ready to be shipped in March. The transportation from the factory to the hub consumes another month. Therefore, when the hub requirement for March arrives at the hub, the simulation calendar has moved to April. The factory fails to achieve the hub demand on time.

As can be observed in the factory in Figure 5.1, the information delayed state is actually last month hub requirement. Hence, the factory part is exactly the same as the factory in the original model without any delay. The hub supply line is the same as in the original model. The only effect is that the whole factory works one month behind the hub. Therefore, the factory inventory behaviour should be similar to the original model.

5.1 The Draeger Model

With the modification, the additional information delay state, $I_{de}(t)$ is defined as the last month hub requirements.

$$I_{de}(t) = H_{req}(t+1) \tag{5.1}$$

To the extent that the inventory plus incoming shipments are sufficient, the factory outgoing shipments can satisfactorily deal with the existing backlog and the incoming

demand. If the demand is too great to satisfy, factory outgoing shipments becomes incoming shipments plus inventory.

$$F_{ship}(t) = \min (I_{de}(t-1) + F_{blk}(t-1), F_{inv}(t-1) + F_{prod}(t-1)) \quad (5.2)$$

The factory backlogs are updated by adding the information delay state and subtracting outgoing shipments. If the information delay state plus backlogs are covered by the incoming shipments plus existing inventory the new backlog is empty.

$$F_{blk}(t) = \max(0, F_{blk}(t-1) + I_{de}(t-1) - (F_{inv}(t-1) + F_{prod}(t-1))) \quad (5.3)$$

Factory production is calculated based on the current information delay state, following the same anchoring and adjusting heuristics as in the original model.

$$F_{prod}(t) = \max(0, \alpha (Q - F_{inv}(t) + F_{blk}(t)) + I_{de}(t)) \quad (5.4)$$

Unlike the production delay model, there is no factory supply line in the planning delay model. Most equations are identical to the original model. Since there are no production limits and production delays, the factory can always satisfy the hub demand and does not have any backlog. The factory inventory is always stable at the desired level. The delayed shipments are only due to inaccurate hub requirements information. The equations are programmed in Appendix A.

5.2 Hub Analysis

A similar dynamics analysis to that in previous model is applied. All backlogs are assumed to be zero. To construct the system characteristic matrix and calculate the eigenvalues, all of the system equations are grouped together and expressed as the general discrete state space representation

$$X(k) = A\,X(k-1) + B\,U(k) \quad (5.5)$$

$$Y(k) = C\,X(k) \quad (5.6)$$

Thus, the state vector X(k) becomes:

$$X(t) = \begin{bmatrix} H_{ship}(t) \\ H_{inv}(t) \\ I_{de}(t) \\ H_{req}(t+2) \\ H_{forecast}(t+2) \\ F_{ship}(t) \\ F_{inv}(t) \\ F_{prod}(t) \end{bmatrix} \tag{5.7}$$

the input U(k) as:

$$U(k) = \begin{bmatrix} H_{orders}(t) \\ Q_H(t) \\ Q_F(t) \end{bmatrix} \tag{5.8}$$

The system matrices are:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\alpha_H & -\alpha_H\beta_H & 0 & \theta & -\alpha_H & 0 & 0 \\ 0 & 0 & 0 & 0 & \theta & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & \alpha_F & 1 & 0 & 0 & -\alpha_F & -\alpha_F \end{bmatrix} \tag{5.9}$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \\ 1+\alpha_H-\theta & \alpha_H & 0 \\ 1-\theta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \alpha_F \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.10}$$

Inspection of equation (5.7) and (5.9) indicates that the extra information delay unit $I_{de}(t)$ has been added to the state vector X(t) and the system matrix A. An extra

eigenvalue is generated, compared to the original model. In the research presented here, θ will remain fixed at 0.75 and the analysis of dynamics will concentrate only on variations in α and β.

With the state vector and system matrix, the eigenvalues of the A matrix are calculated as the solution to det(zI - A) = 0. The constellation is plotted on z-plane (each eigenvalue is represented by an 'x') as both α and β are incremented from 0 to 1 in steps of 0.1. Figure 5.2 shows this constellation plotted on a Jury map.

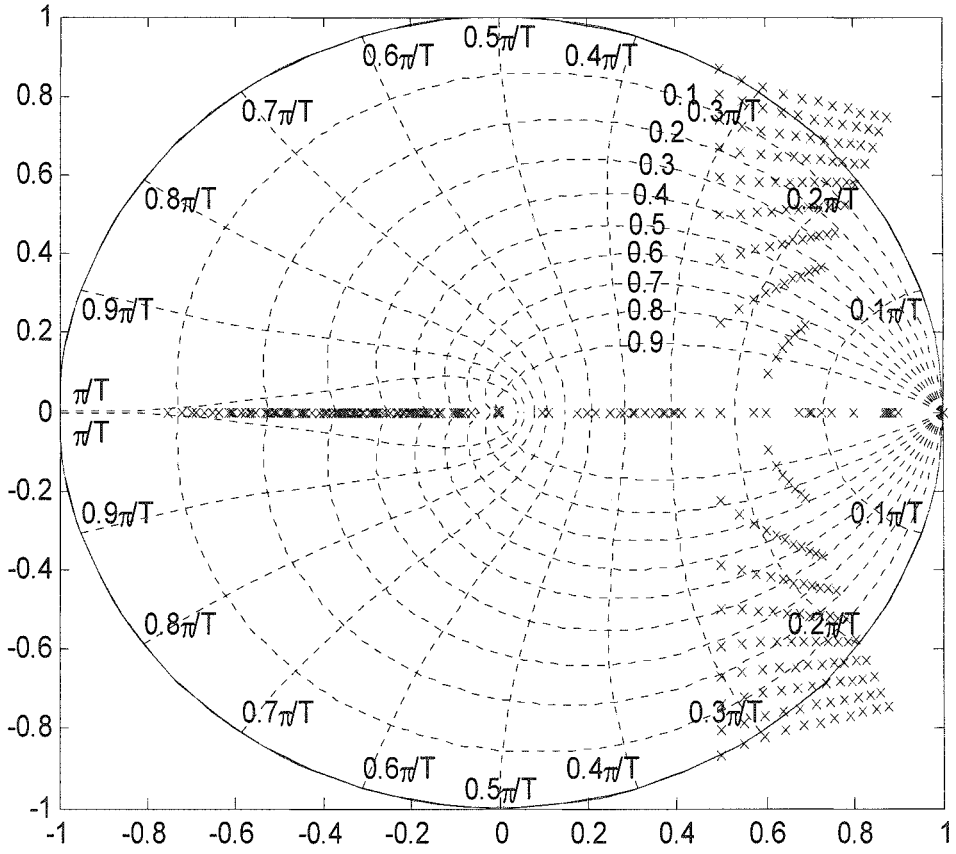
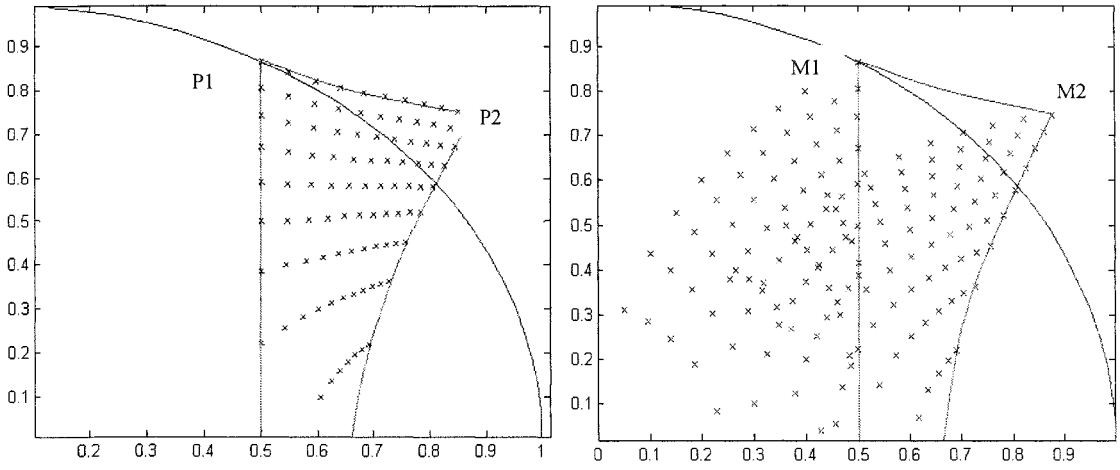


Figure 5.2 *The constellation of eigenvalues of A plotted on the z-plane for θ = 0.75, α = 0 to 1 in steps of 0.1 & β = 0 to 1 in steps of 0.1.*

Eight sets of eigenvalues have been plotted within the constellation, changing α and β from 0 to 1. Inspection of Figure 5.2 shows that the overall shape of the planning delay constellation is symmetrical. It is neither similar to the original model without any delay nor the model with production delay.



(a)Model with planning delay *(b) Model with two step production delays*

Figure 5.3 *Eigenvalues of model without delay and with production delays*

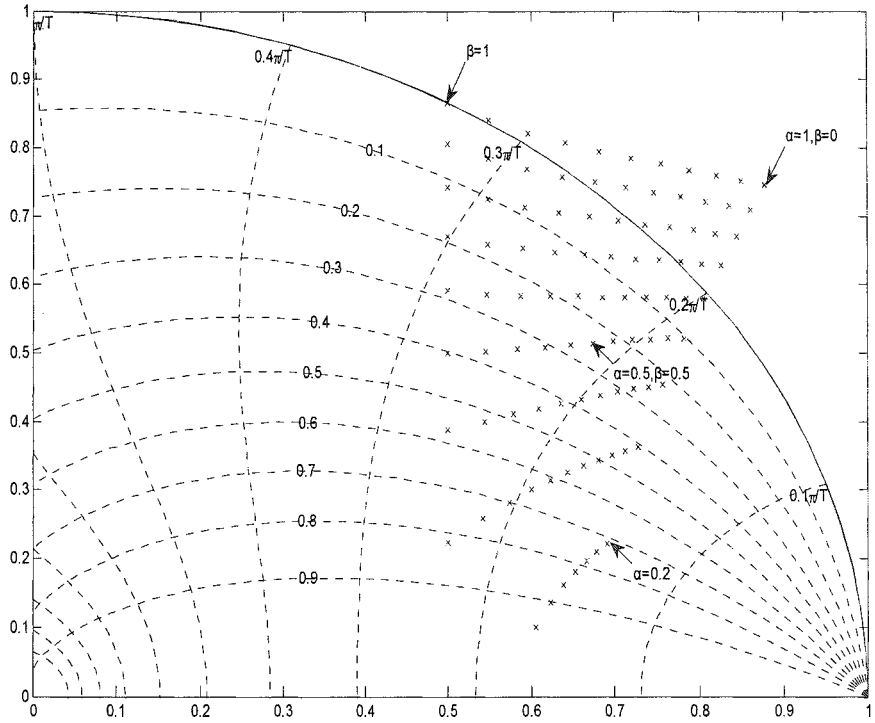


Figure 5.4 *Expanded view of 1st quadrant, ignoring eigenvalues located on the real axis.*

However, the expanding of the first quadrant in Figure 5.3 shows that the shape of eigenvalues plot for planning delay model in Figure 5.3 a is the same as the red quadrangle in Figure 5.3 b. The points on the corner of the two red quadrangles in Figure 5.3 (a) and (b) P1, M1 and P2, M2 have the same complex value.

The symmetry allows the analysis focusing on the upper half of the eigenvalues constellation to explore the detail of the dynamics of the whole model. Figure 5.4 indicates the existence of pairs of potentially unstable complex eigenvalues. The damping ratio curve ς (curve 0.9 to 0.1) is marked within the constellation. As α decreases from one to zero, the eigenvalues move vertically from outside of the circle to the inside of the circle. They cross the damping ratio curve from $\varsigma = 0.1$ near the edge of the circle to the curve at $\varsigma = 0.9$ near the real axis. As a result of the increasing damping ratio, the system becomes more damped and the inventory oscillations become smaller. On the other hand, decreasing β, the corresponding eigenvalues migrate from inside to outside of the unit circle. For example, by fixing α at1 and decreasing β from 1 to 0, the eigenvalues start moving from the edge of the circle toward the outside of the circle horizontally. Therefore as long as α =1, the corresponding eigenvalues are always located outside of the unit circle. By changing β, the eigenvalues do not cross as many damping ratio curves as α does. However, the corresponding natural frequency w_n plotted from $0.1/\pi$ T to $0.4/\pi$ T along the edge of the circle, has been changed significantly by this relative horizontal eigenvalues movement. As discussed before, the natural frequency w_n represents the oscillation frequency. In decreasing β, the relevant w_n becomes smaller and leads to small frequency oscillation. It is clear that, just as in the two previous cases, β has a stabilising influence, whilst α is destabilising.

5.2.1 The Influence of α

Figure 5.5 shows a plot of the eigenvalue locus when β = 0 and α is varied from 0 to 1. There are two sets of eigenvalues in the figure, represented by green and black points. The green point locate at the origin of Z-plane with coordinates (0, 0) when α =0 and β=0.

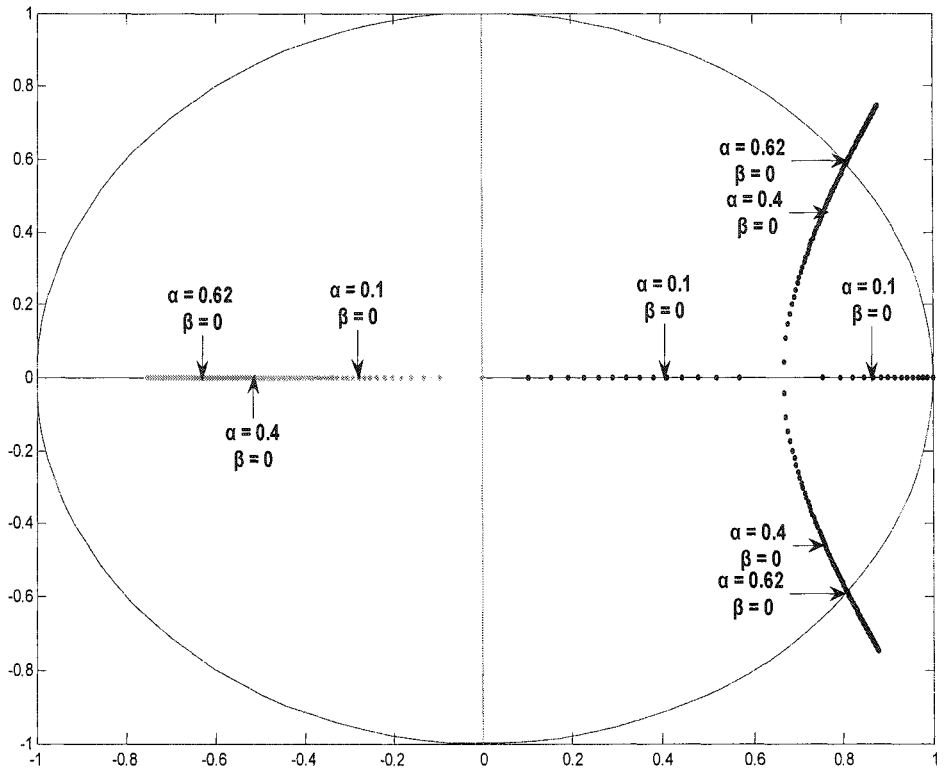
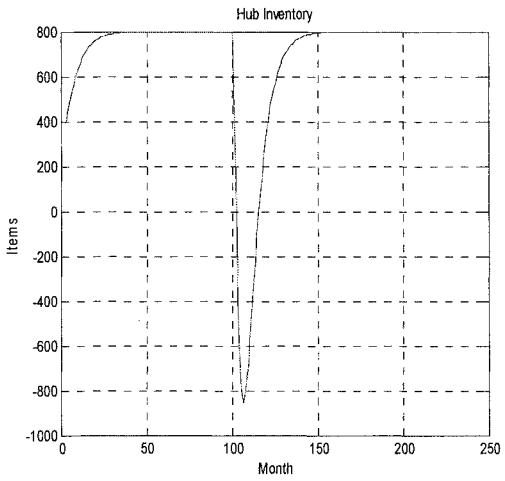
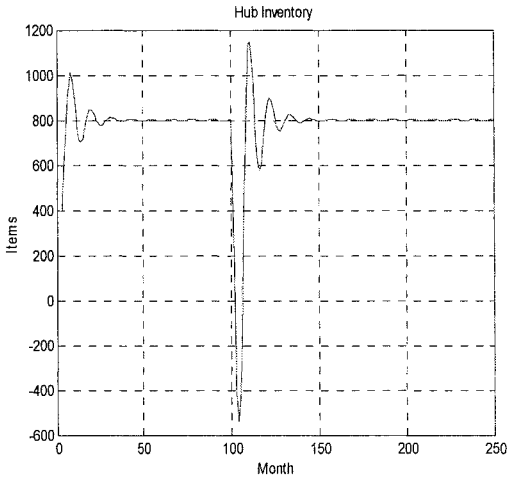


Figure 5.5 *Locus of eigenvalues of A plotted on the z-plane for θ = 0.75, β = 0 and α varied from 0 to 1. Clearly, the complex pair migrate toward the edge of the unit circle as α increases.*

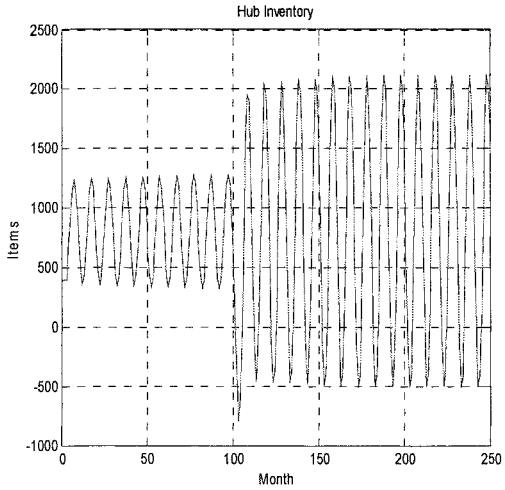
Increasing α, the eigenvalues (represented by the green point) move left along the negative real axis to -0.75. The black eigenvalues z = 0 & z = 1 locate on the real axis with coordinates (0, 0) and (0, 1) when α =0 and β=0. Increasing α, these two black points move toward each other on the real x axis. When α = 0.14, they collide in the real axis. With further increments of α, the black eigenvalues jump into the first and fourth quadrant of the plane. Once the eigenvalue obtain the imaginary part, the inventory starts oscillating, as shown in Figure 5.6 (b). The black curve crosses the circle at α = 0.62. If α continues to increase, the black point moves out of the unit circle. The inventory behaviour becomes unstable in Figure 5.6 (d).



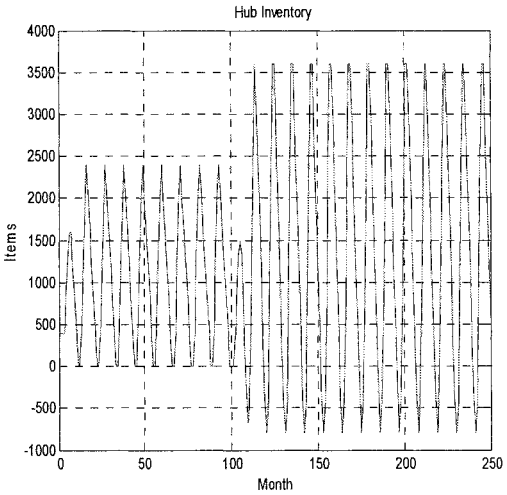
(a) α = 0.1, β = 0



(b) α = 0.4, β = 0



(c) α = 0.62, β = 0



(d) α = 1, β = 0

Figure 5.6 *Hub effective inventory for θ = 0.75, α = 0.1, 0.4, 0.62, 1, β = 0 and Q = 800. The figure shows the response of the Hub effective inventory over a 250 month period. The initial Hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

It is clear from Figure 5.6 that as α increases the inventory responses become more oscillatory. When α is greater or equal to 0.62, the response exhibits continuous oscillation. The ratio between the two successive wave peaks becomes greater. This is defined as $m_{k+1}/m_k \times 100$, where m_k is the absolute value of the wave peak at time k and m_{k+1} is the peak of the neighbour wave at time k+1. In Figure 5.6 (a), there is no

oscillation at all. The ratio is considered as 0. Increasing α to 0.4 as shown in Figure 5.6 (b), the inventory oscillates with decreasing amplitude. The ratio can be calculated from the relevant eigenvalues as 48.2%. In Figure 5.6 (c) with α equal to 0.62, the ratio is equal to 99.8%. This means the hub inventory oscillates with nearly fixed amplitude. When the eigenvalues move to the outside of the unit circle, the system becomes unstable. The calculation of the eigenvalues for α equal to 1 shows the ratio should be 187%, which indicates the oscillation amplitude is expanding. However, Figure 5.6 (d) shows that the inventory maintains constant oscillation amplitude. The hub inventory follows the same oscillation pattern as Figure 5.6 (c) and greater amplitude. When the oscillation amplitude is great enough to meet the saturation form boundary that is defined in the model equations, the oscillation cannot expand over the limits. Therefore, the eigenvalues for α equal to 1 do not indicate the inventory behaviour properly as in Figure 5.6(d).

5.2.2 The Influence of β

Figure 5.7 shows a plot of the eigenvalue locus when α = 1 and varying β from 0 to 1. The eigenvalues have been plotted in two groups in green and red in the plane. The green points move left along the negative real axis as β decreases. The red points locate on the edge of the circle when α=1 and β=1. Decreasing β, the red point moves further outside of the unit circle (red curve). As long as α=1, the whole red curve always locates outside of the circle. It is clear that the stabilising effect of β is not as strong as the destabilising effect of α, which has been strengthened by the information delay.

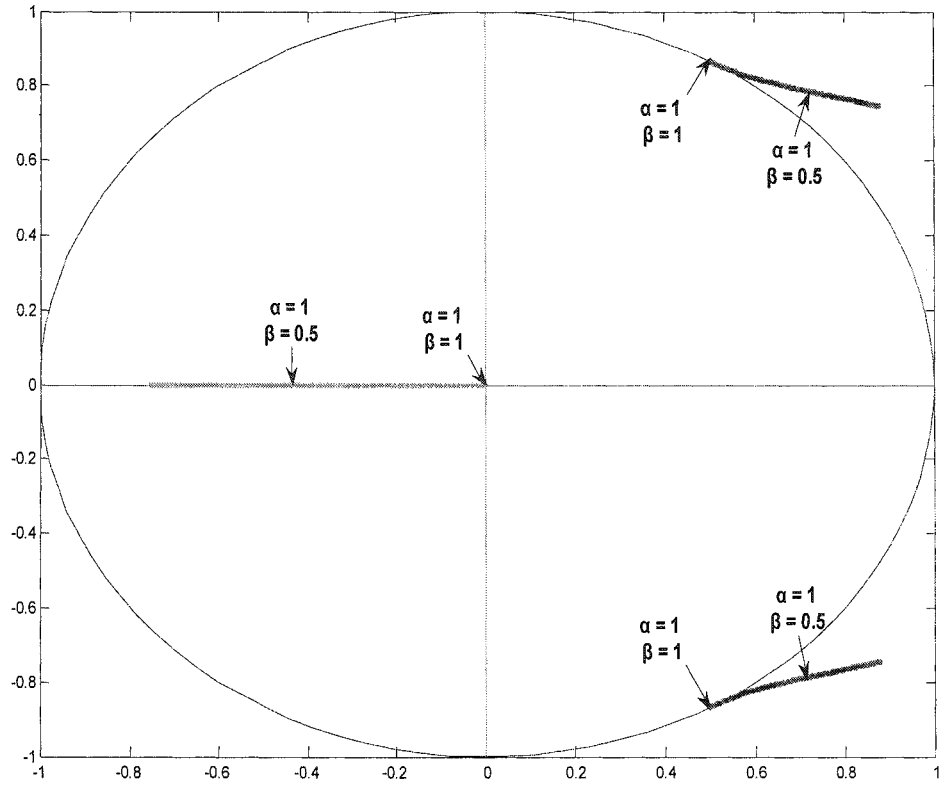
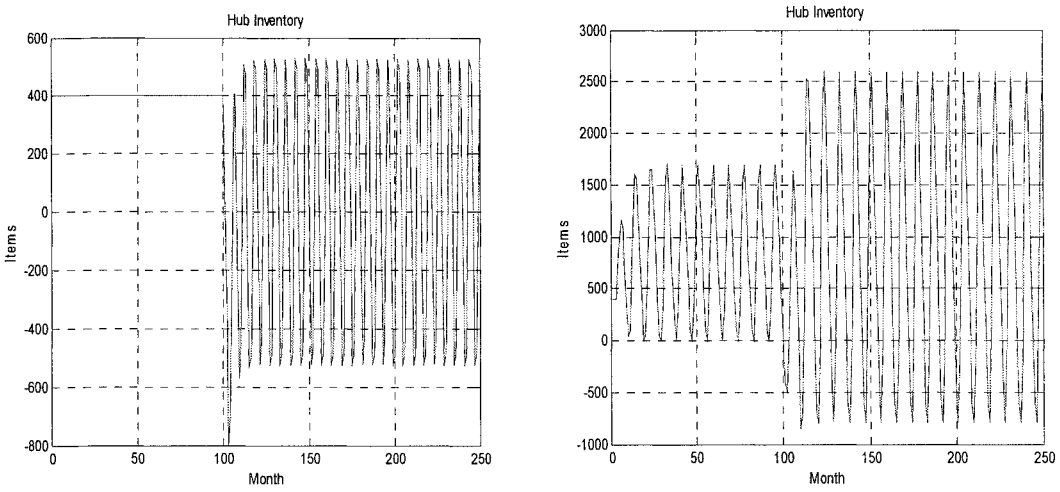


Figure 5.7 *Locus of eigenvalues plotted on z-plane for $\theta = 0.75$, β varied from 0 to 1 and $\alpha = 1$. Clearly, the complex pair migrates toward the centre of the unit circle as β increases.*

Figure 5.8 shows hub inventory behaviour when $\alpha = 1$, $\beta = 1$ and $\alpha = 1$, $\beta = 0.3$. After 100 months the overall hub inventory behaviour of the two cases is very similar. However, in the first 100 months, Figure 5.8 (a) is stable at 400 units, while Figure 5.8 (b) is oscillating with fixed amplitude. In both cases, the eigenvalues locate outside of the unit circle. The relevant system behaviour is expected to be unstable. Therefore, the stable inventory behaviour in the first 100 month is not expected to be seen in Figure5.8 (a).



(a) $\alpha = 1, \beta = 1$ *(b) $\alpha = 1, \beta = 0.3$*

Figure 5.8 *Hub effective inventory for $\theta = 0.75$, $\alpha = 1$, $\beta = 1$, 0.3 and $Q = 800$. The figure shows the response of the hub effective inventory over a 250 month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

To explain this puzzling stable system behaviour, the hub requirement equations need to be reviewed.

$$H_{req}(t+2) = \max(0, \alpha (Q - H_{inv}(t) + H_{blk}(t)) \\ - \alpha \beta (F_{blk}(t) + F_{ship}(t)) + H_{forecast}(t+2)); \quad (5.11)$$

When $\alpha = 1$, $\beta = 1$, the equation changes to

$$H_{req}(t+2) = Q - H_{inv}(t) + H_{blk}(t) - F_{blk}(t) - F_{ship}(t)) + H_{forecast}(t+2) ; \quad (5.12)$$

If there is no backlog in both hub and factory, the initial hub inventory and factory incoming shipments are 400 units, while Q is 800 units; the equation can be written as

$$H_{req}(t+2) = H_{forecast}(t+2) ; \quad (5.13)$$

In the first 100 months, the incoming order is 400 units, which is equal to the initial value of the inventory and shipment. In this situation of equilibrium, a backlog does not exist. The hub requirement is always equal to the forecast whose initial value is the

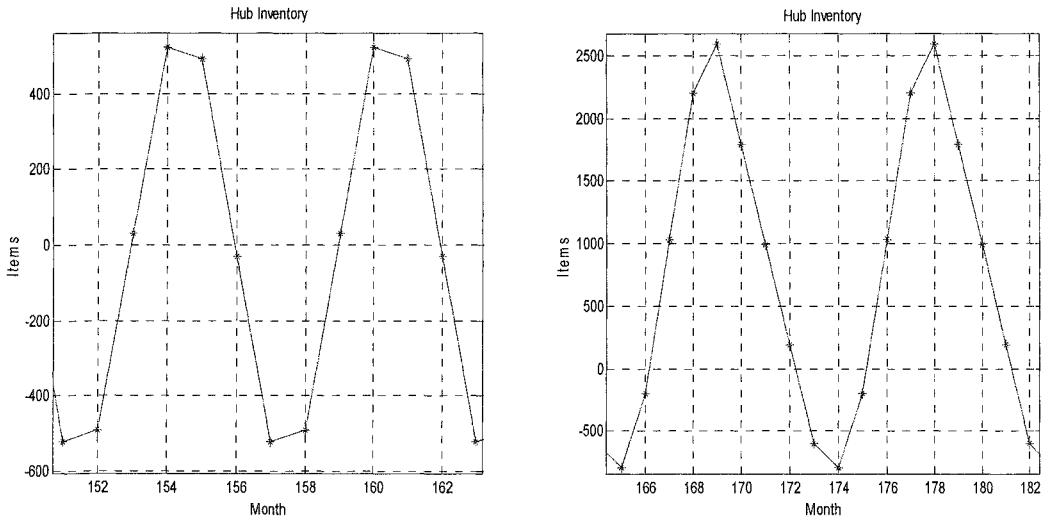
same as the order. The planning delay cannot have any effect on this equilibrium model. Therefore the hub inventory is stable at the initial level. However, after 100 months the orders increase to 800 units. An inventory backlog is produced in the hub. The planning delay starts to impact the system, as the hub cannot receive the shipment on time and the hub inventory oscillates.

On the other hand, if β is smaller than 1, the hub requirement is

$$H_{req}(t+2) = 400-\beta F_{ship}(t)+ H_{forecast}(t+2)\ ; \qquad (5.14)$$

The hub requirement is not the same as the initial value. The planning delay keeps the right information for a month and forwards the last month hub requirements to the factory. In this way, the factory shipment cannot be delivered to the hub on time. The hub inventory starts oscillating from the beginning of the simulation in Figure 5.8 (b).

After the first 100 months, the inventory shown in Figure 5.8 (a) and (b) behaves in a similar way. However, closer observation shows that the oscillation details are still different.



(a) α = 1, β = 1 *(b) α =1, β = 0.3*

Figure 5.9 *Expanded section of hub effective inventory for two periods, where θ = 0.75, α = 1, β = 1, 0.3 and Q = 800.*

Figure 5.9 shows that the oscillation has been expanded into two waves when α = 1, β = 1 and α = 1, β = 0.3 respectively. The eigenvalue is 0.7203 +i0.7848 when α = 1, β = 1,

where the period of the oscillation can be predicted by the angle to be 2 π/0.84 (approximate 7). It has been confirmed in Figure 5.8 (a), one period of the wave include seven distinct points. On the other hand, the eigenvalues for $\alpha = 1$, and $\beta = 0.3$ are 0.7889 + i0.7676, where the oscillation period can be predicted by the angle to be $2\pi/0.7$(approximately 9). The result has been confirmed, as shown in Figure 5.8 (b): there are nine distinct points in the wave when $\alpha = 1$, $\beta = 0.3$. It is clear that with smaller values of β, the period of the oscillation is extended, while the frequency becomes lower.

5.2.3 The Primary Route to Instability

To understand fully this behaviour, the structure of the system block diagram needs to be examined; which, in this case, is shown in Figure 5.10. Here it can be seen that additional planning state modifies only the hub loop, the factory loop reverting to the structure originally seen in the original model.

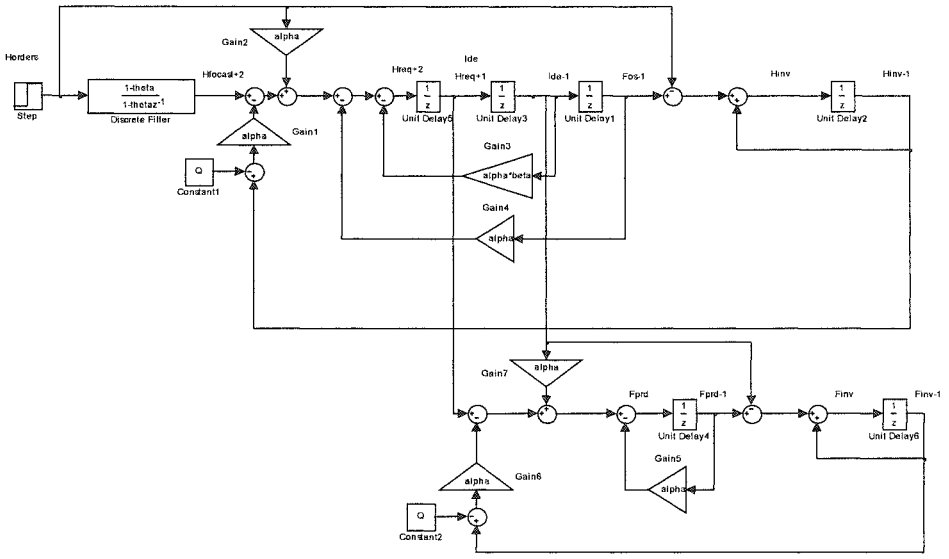


Figure 5.10 *Small signal block diagram representation of the model of the supply chain at Draeger Safety UK, including additional planning delay.*

Thus, the factory characteristic equation is:

$$(z + \alpha)(z - 1) + \alpha = 0 \qquad (5.15)$$

It has only real eigenvalues.

However, for the condition when $\beta = 0$, the hub characteristic equation becomes:

$$(z^3 + \alpha_H)(z - 1) + \alpha_H = 0 \qquad (5.16)$$

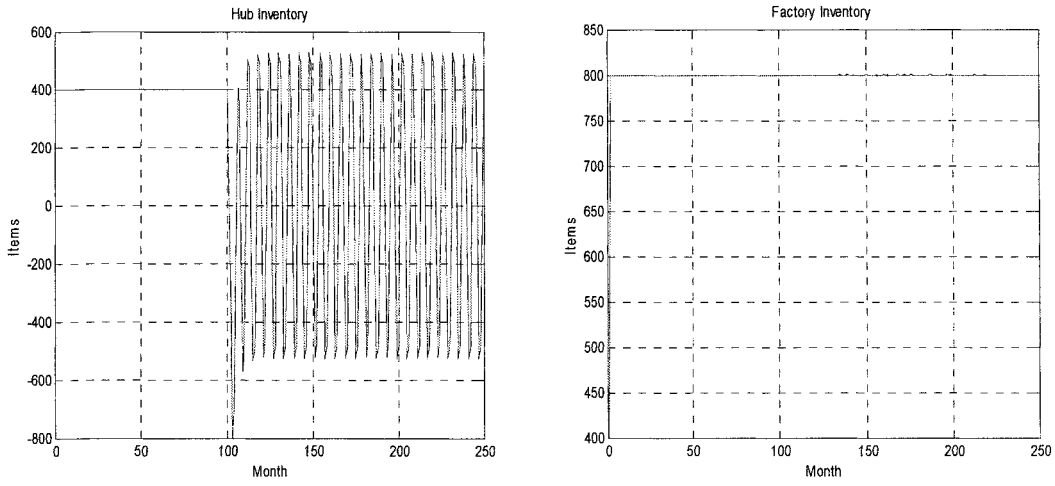
This is identical to the characteristic equation for the system with additional production delays, for the condition $\beta = 0$ (Equation 4.18). Thus, the complex eigenvalue pair seen in Figure 5.5 lie at identical locations to the 'similar' pair seen in the eigenvalue plots for the production delay model when $\beta = 0$ (Figure 4.15). Further, in the case when $\beta = 1$, the hub characteristic equation becomes:

$$(z^3 + \alpha_H z + \alpha)(z - 1) + \alpha_H = 0 \qquad (5.17)$$

This has two eigenvalues at the origin, together with a complex pair whose locations are identical to those given by the hub characteristic equation in the original model when $\beta = 0$ (Equation 3.33). Thus, it becomes clear that the locations of the eigenvalues seen in Figure 5.7 and Figure 5.5 are bounded by the two pairs of complex eigenvalues for the production delay model when $\beta = 0$ (Figure 4.15) . Consequently, it is the hub that is once again the primary route to instability, the structure of the hub feedback loop, under the influence of the planning delay, becoming identical to that seen in the factory with additional production delays, when $\beta = 1$.

5.2.4 The Influence of 'Saturation' Effects

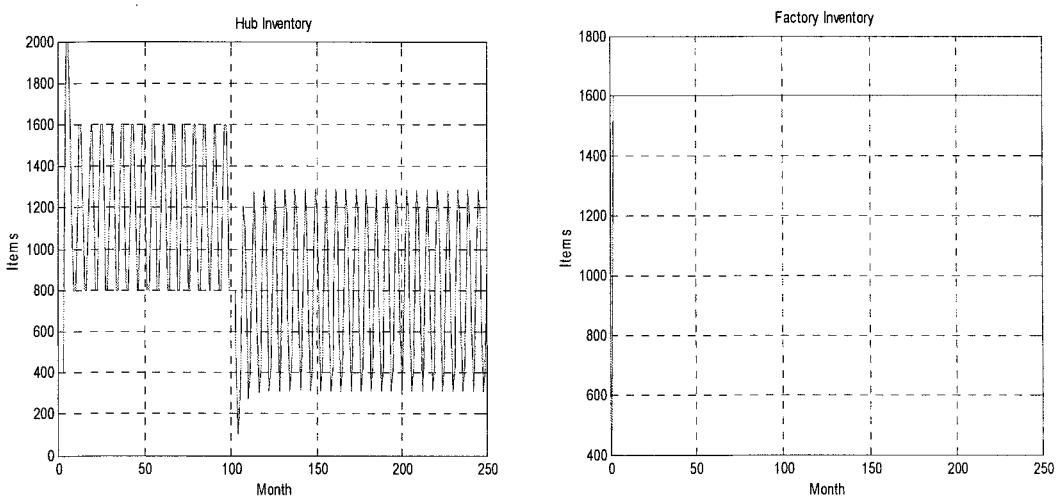
The saturation effects which are the impact of desired inventory Q is measured in this section. Figure 5.11 shows inventory behaviour when $Q = 800$, $\alpha = 1$ and $\beta = 1$. The hub inventory is stable in the first 100 months and starts oscillating from -500 to 500 units after. The factory inventory is always stable at the desired level of 800 units.



(a) Hub effective inventory Q=800 *(b) Factory effective inventory Q=800*

Figure 5.11 *Simulation when $\theta = 0.75$, $\alpha = 1$, $\beta = 1$ and $Q = 800$. The figure shows the response of the Hub and factory effective inventory over a 250 month period. The initial hub and factory inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

If the desired inventory Q is doubled to 1600 units, the hub inventory shown in Figure 5.12 (a) oscillates from 800 units to 1600 units in the first 100 months. After the incoming sales order increase, the hub inventory oscillates at the lower level from 300 units to 1300 units. The factory inventory is still stable at the desired level of 1600 units in Figure 5.12 (b). The increase of Q does not eliminate the oscillation. However, the increase of Q leads to oscillation in the first 100 months, as in Figure 5.12 (a).



(a) Hub effective inventory Q=1600 *(b) Factory effective inventory Q=1600*

Figure 5.12 *Simulation when $\theta = 0.75$, $\alpha = 1$, $\beta = 1$ and $Q = 1600$. The figure shows the response of the hub and factory effective inventory over a 250 month period. The initial hub and factory inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation*

To explain the oscillation in the first 100 months, the hub requirement equation (5.12) is recalled. When Q =1600, the equation changes to

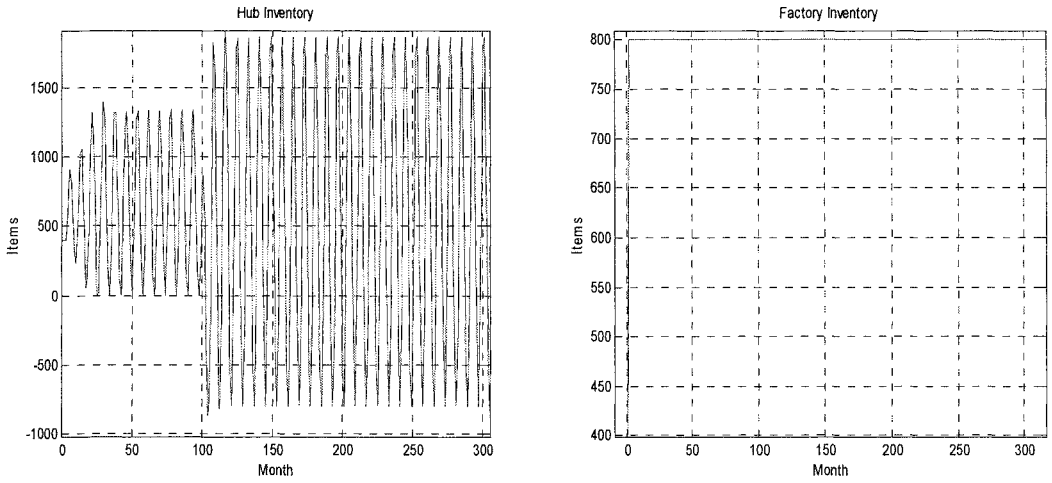
$$H_{req}(t+2) = 800 + H_{forecast}(t+2) ;$$

The order is constant at 400 in the first 100 months. The hub requirement is 1200 units when $\alpha = 1$, $\beta = 1$ and Q =1600 units. It is greater than the initial value of the information delay state of 400 untis. Since the planning delay postpones the transfer of the hub demand, the factory shipment arrives at the hub one month later. Then the hub inventory oscillates around 1200 with an amplitude of 400. The oscillations in Figure 5.12 (a) after 100 months are at a positive level (300 to 1300), while the oscillations shown in Figure 5.11(a) generate backlogs (-500 to 500). It can be concluded that changing safety stock Q in the planning delayed model cannot eliminate oscillation. However, it will influence the backlog level of the inventory. The oscillation analysis in this section relates to the disturbance which in our case, is incoming order. Therefore, if the initial situation or incoming order is changed, the system behaviour will be different.

5.3 The Influence of Production Constraints

To measure the influence of production constraints on the dynamics of the planning delay model, the planning parameters α and β are fixed at 1 and 0.5 so that the relevant eigenvalues sit outside of the unit circle. Hence, according to the previous eigenvalue analysis the model should be unstable. However, by applying different degrees of production constraints in the factory, the model exhibits variety dynamics behaviours.

Applying the production constraint as 2000 units on the planning delay model, factory inventory response remains stable, while the hub inventory oscillates, with a peak at 1800 (Figure 5.13).



(a) Hub inventory *(b) Factory inventory*

Figure 5.13 Simulation when production constraint=2000, θ = 0.75, α = 1, β = 0.5 and Q = 800. The figure shows the response of the hub effective inventory over a 250 month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.

Since the factory production constraint is greater than the Hub inventory oscillation amplitude, the factory can always produce enough inventories to meet hub demand. To clarify the dynamics of the inventory, Figure 5.14 plots the hub inventory against the factory inventory after the transient state of the first 100 months.

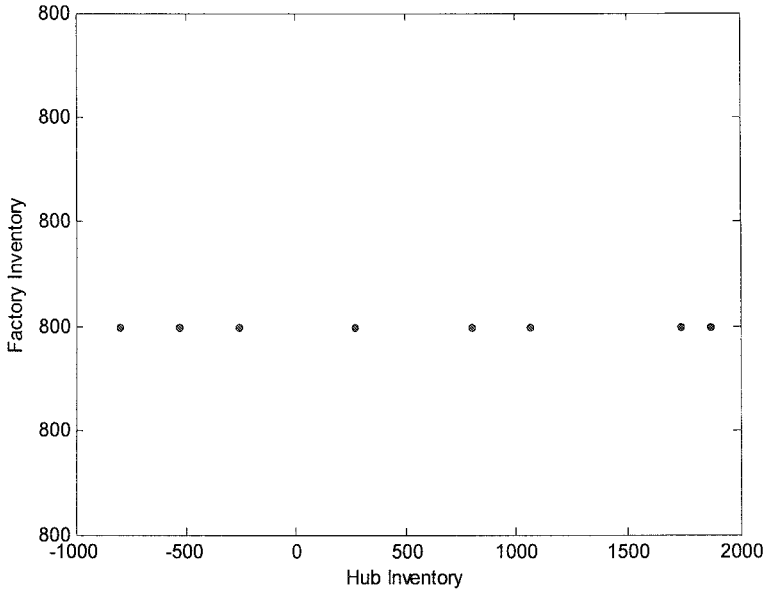
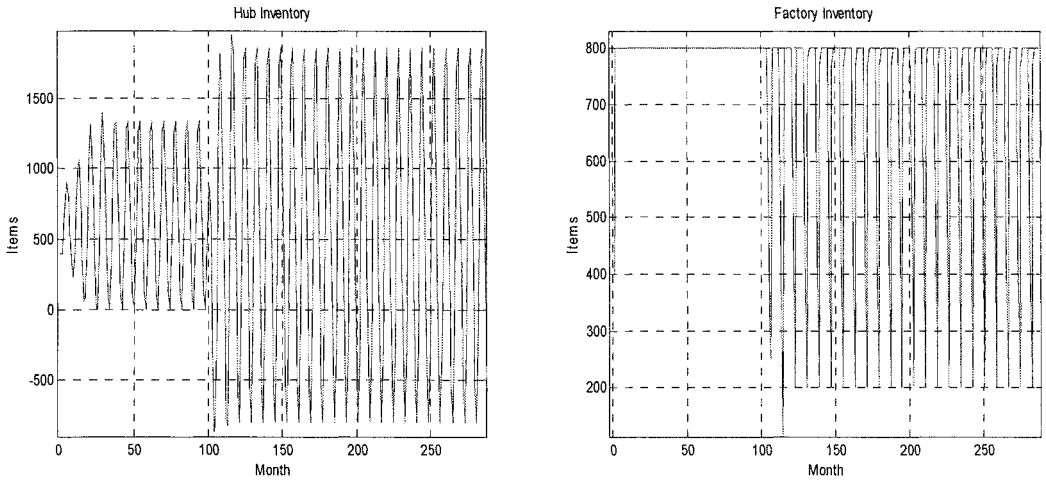


Figure 5.14 *Periodic solution obtained for Production constraint=2000, θ = 0.75, α = 1,β =0.5, θ = 0.75, Q =800.The phase plot shows corresponding values of hub and Factory effective inventories after ten thousand iterations of the planning delayed model with production constraint.*

The factory inventory remains constant, while the hub inventory oscillates between eight levels. After one period of oscillation, the hub inventory moves back to the start point.

Decreasing the production constraint to 1500 units, the factory inventory starts oscillating after 100 months. Inspection of Figure 5.15 (a) shows that hub inventory oscillation is below 1500 in the first 100 months. Factory production can cover hub requirements and keep its own inventory stable. However, after 100 months, the hub inventory generates backlogs and the amplitude of the oscillations is larger than the production constraint, which leads to oscillations in the factory (Figure 5.15 (b)).



(a) Hub inventory *(b) Factory inventory*

Figure 5.15 *Simulation when Production constraint=1500, $\theta = 0.75$, $\alpha = 1$, $\beta = 0.5$ and $Q = 800$. The figure shows the response of the hub effective inventory over a 250 month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

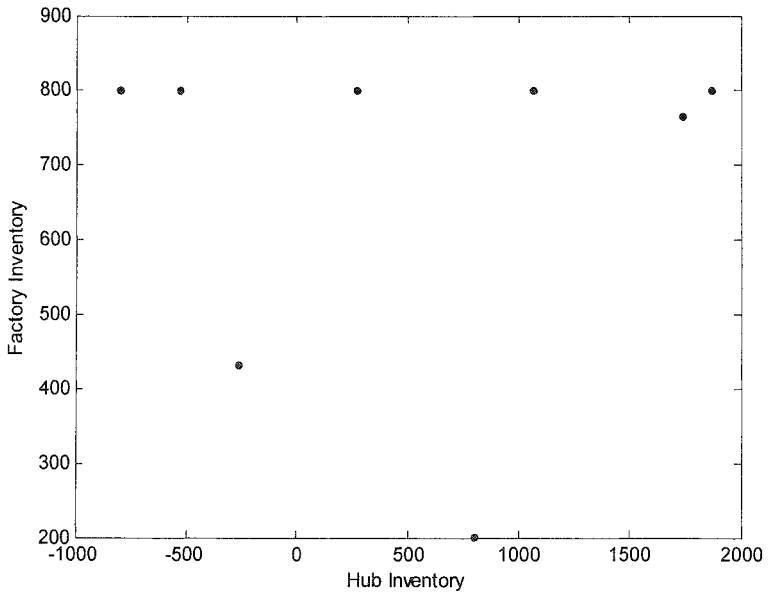
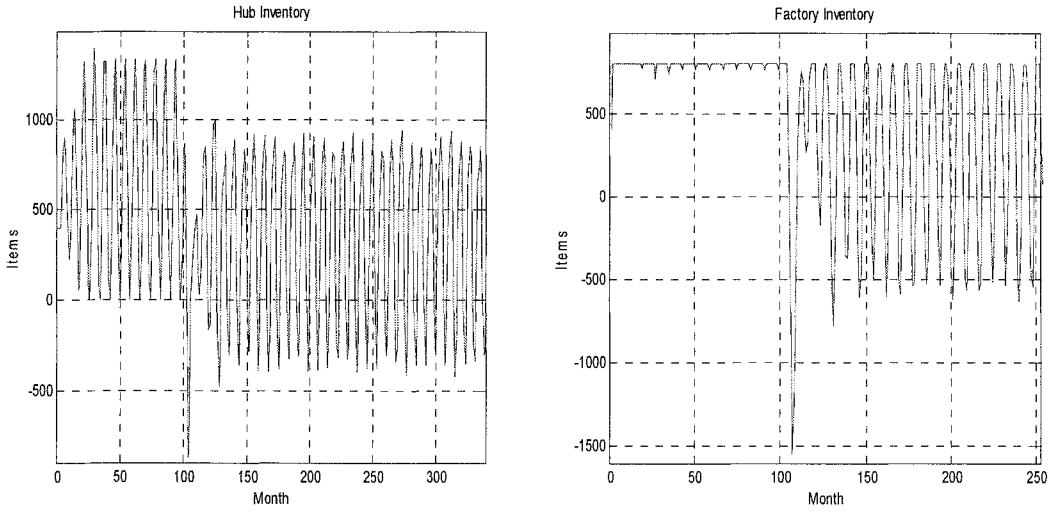


Figure 5.16 *Periodic solution obtained when production constraint=1500, $\theta = 0.75$, $\alpha = 1$,$\beta = 0.5$, $\theta = 0.75$, $Q = 800$. The phase plot shows corresponding values of hub and factory effective inventories after ten thousands iterations of the planning delayed model with production constraints.*

Figure 5.16 represents the phase plots of the factory effective inventory against the hub effective inventory with production constraint at 1500 units. Since the factory inventory start oscillating after the first 100 months, those points in the phase plots locate at different factory inventory levels.

Keep decreasing the production constraints to 900 units, the hub inventory and factory inventory behaviour are then similar, as shown in Figure 5.16, in the first 100 months. However, after the incoming order doubles the hub inventory oscillation produces a backlog. As a result, the factory inventory starts oscillating (Figure 5.17(b). When the production constraints become smaller, the hub inventory is oscillates in a smaller amplitude. The hub inventory against factory inventory phase plot is a quasi-periodic solution, as in Figure 5.18, when the production constraint is 900 units.



(a) Hub inventory *(b) Factory inventory*

Figure 5.17 *Simulation at Production constraint=900, $\theta = 0.75$, $\alpha = 1$, $\beta = 0.5$ and $Q = 800$. The figure shows the response of the hub effective inventory over a 250 month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

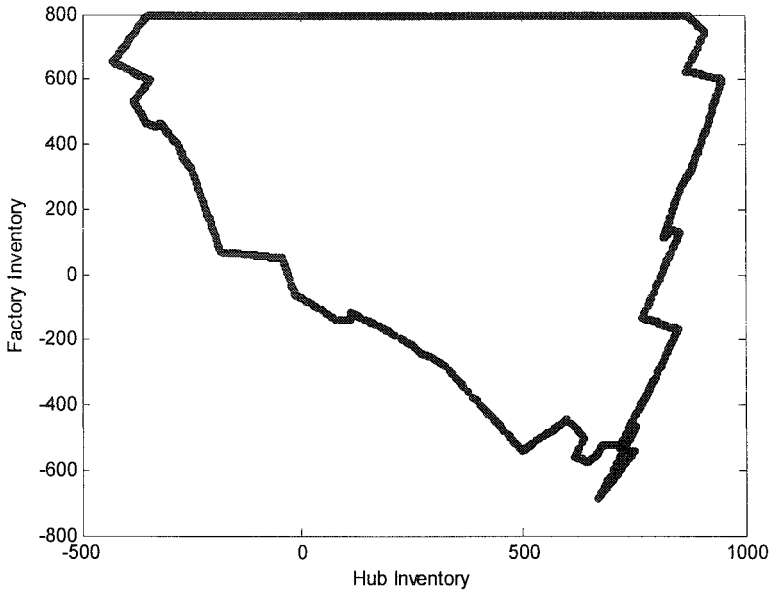
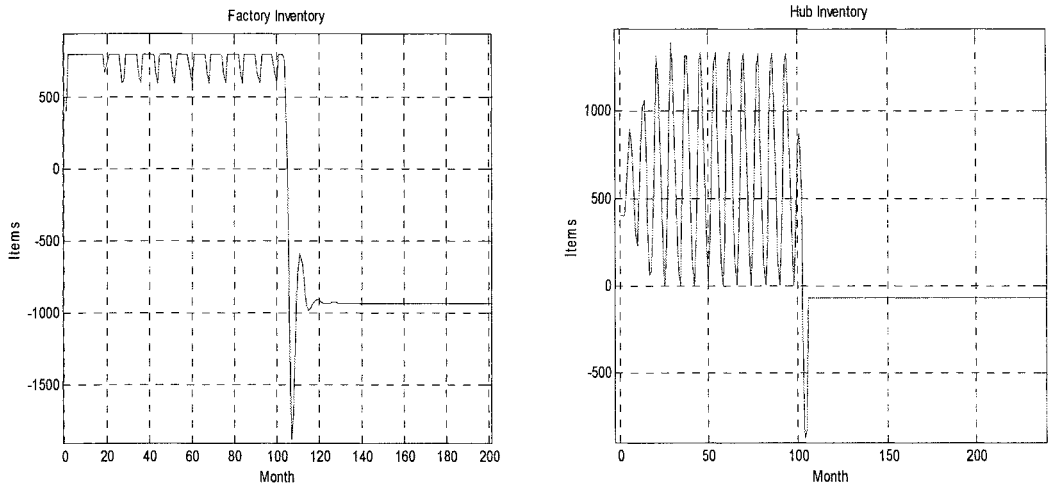


Figure 5.18 *Quasi-periodic solution obtained at Production constraint=900, θ = 0.75, α = 1, β =0.5, θ = 0.75, Q =800.The phase plot shows corresponding values of hub and factory effective inventories after ten thousand iterations of the planning delayed model with production constraints.*

Decreasing the constraint to 800 units (the level of sales demand), the production can only feed the demand and cannot compensate for the hub and factory inventories. In Figure 5.19 (a) and (b), the hub and factory effective inventories are stable at backlog level. Since both the hub and factory are in their steady state, there is only one dot in the phase plot in Figure 5.18.



(a) Hub inventory *(b) Factory inventory*

Figure 5.19 *Simulation at Production constraint=800, $\theta = 0.75$, $\alpha = 1$, $\beta = 0.5$ and $Q = 800$. The figure shows the response of the hub effective inventory over a 250 month period. The initial hub inventory is 400 and the hub orders are held constant at 400/month for the first 100 months. These are then increased to 800/month for the remainder of the simulation.*

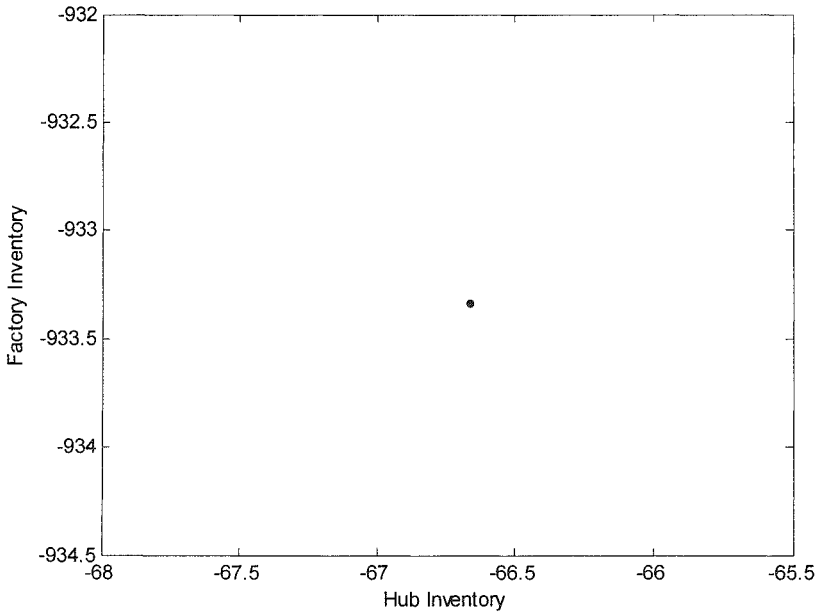


Figure 5.20 *Periodic solution obtained at Production constraint=800, $\theta = 0.75$, $\alpha = 1$, $\beta = 0.5$, $\theta = 0.75$, $Q = 800$. The phase plot shows corresponding values of hub and factory effective inventories after ten thousand iterations of the planning delayed model with production constraints.*

5.3 Conclusion

The third scenario presented looks at the effect of an additional one-month information delay between the hub and the factory. In this situation, it is, again, hub management policy that is the primary route to instability. Also, just as in the two previous cases, α has a destabilising influence, whilst β is stabilising. However, with the additional information delay, the hub's route to instability now follows the more severe path defined by Equation 5.16. Moreover, in the presence of the one month information delay, even the stabilising influence of β only lessens the severity of the route to instability; in this case, to the route defined in Equation 5.17 (for the case when $\beta = 1$). Change of the safety stock Q which measures the saturation effects cannot influence the stability of the system, but change the scale of the variation to reduce the costs. Thus, in this situation, good management and management policies are crucial if significant problems are to be avoided. Therefore, accurate forecasting is essential to improve supply chain performance.

Chapter 6:

Forecasting Implemented with the ARMA Model

Forecasts play a key role in supply chain management. Forecast accuracy is an important component in the delivery of an effective supply chain. The basic principle of time series forecasting is to use a model to predict future data, based on known past data. This chapter introduces the forecasting methods implemented with the *ARMA* (auto-regression and moving average) model. The auto-regression and moving average model is a kind of technique for time series analysis and forecasting which has been adopted by many economists (Duc, 2008, A.D., 1990, Suri, 1988).

However, forecasting with *ARMA* is a linear method that is only suitable for the stationary time series. Preprocessing is required to remove some of the variation built within the original data and analyze the interrelationship between the observed values. The structure of the *ARMA* model can then be constructed and applied to make one step forward forecasting. The model parameters are optimized by using recursive least square algorithm during each round of forecasting. After processing the whole time series, the *ARMA* model structure is changed and the same procedure is applied again. These processes involve five steps:

1) Data preprocessing;
2) Constructing the structure of the *ARMA* model;
3) Applying the model to the original data, to make forecasts the model parameters are updated and optimised during the prediction;
4) Changing the structure of the model and going back to step three;
5) Selecting the model with minimum forecasting error.

A performance function, which measures the absolute difference between forecast and real data, was employed to record the forecasting error for each different structured

ARMA model. The model with best performance was selected. The corresponding forecasting results are compared in the later sections.

The original data for making the forecast is sixty-four months of sales history data of a valuable product from Draeger Safety, UK Ltd. Draeger managers authorized the use of data in this research. It can be seen from Figure 6.1, that the overall trend is of growth and the oscillation intervals are not regular.

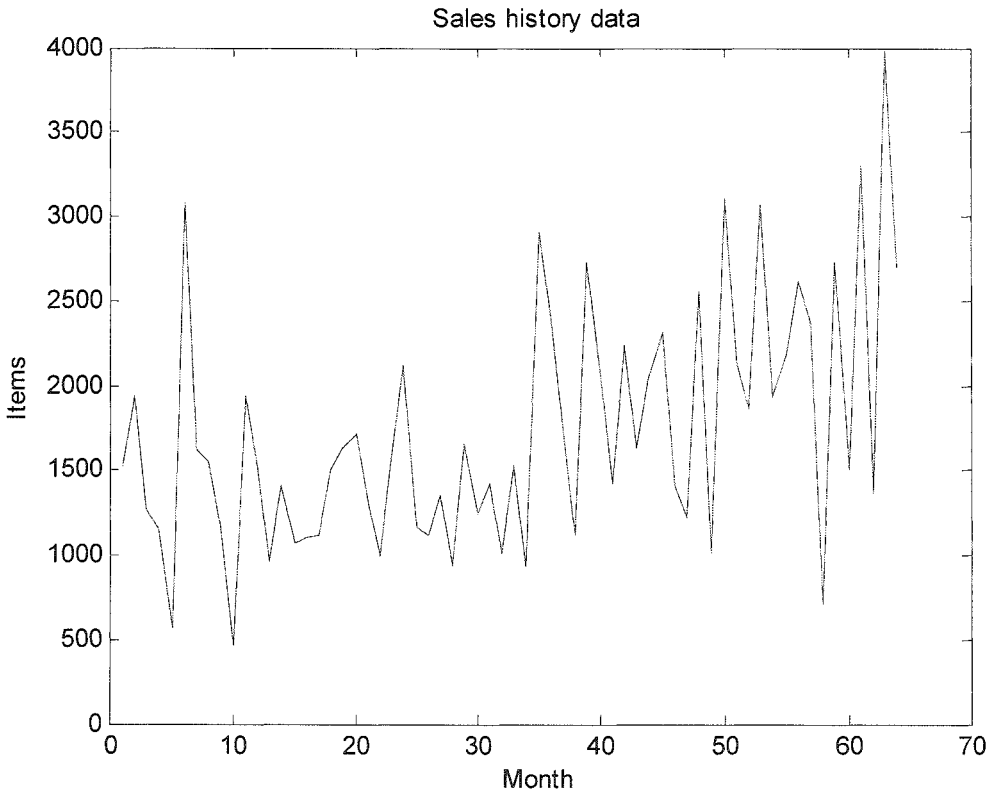


Figure 6.1 *The 64 months sales history*

6.1 Autocorrelation

The foundation of our time series analysis is built on the interrelationship between the observed values of a series at different points in time. Calculating the autocorrelation coefficients is the first step in time series research. The formula of autocorrelation coefficients is listed below:

$$\hat{\rho}_k = \frac{\sum_t^{T-k}(y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum_t^{T}(y_t - \bar{y})^2}; \tag{6.1}$$

$$\bar{y} = \frac{1}{T}\sum_t y_t ; \tag{6.2}$$

where ' y_t ' is the sample data. ' $\bar{y}$ ' is the mean of samples. ' k ' is the lag and ' $\hat{\rho}_k$ ' is the autocorrelation coefficient at lag k. The correlation coefficients need to be tested for their statistical significance before any reasonable conclusion can be drawn. The following hypotheses were tested:

$H_0 : \rho = 0$ the null hypothesis, the correlation is not significant;

$H_1 : \rho \neq 0$ the alternative hypothesis, the correlation is significant.

The standard error is:

$$S.E.(\hat{\rho}_k) = \sqrt{\frac{1-\hat{\rho}_k}{n-2}}; \tag{6.3}$$

Thus, a z test would be appropriate to test the hypothesis with respect to significance of such statistics. The test statistic is given as:

$$z = \frac{\hat{\rho}_k}{S.E.(\hat{\rho}_k)} . \tag{6.4}$$

The table values of z distribution at a 5% level of significance are approximately -1.96 and +1.96. If the computed z is numerically larger than this, the null hypothesis is rejected and concludes the relationship is significant, and vice versa. In the program, the process of the hypothesis has been executed inversely. The 5% level of significance related autocorrelation coefficient is 0.25. The coefficient has been calculated from lag1 to lag20. The correlation values have been plotted in Figure 6.2 and compared with the significance level at 0.25. If the coefficient is bigger than 0.25, the null hypothesis is rejected and concludes the autocorrelation is significant.

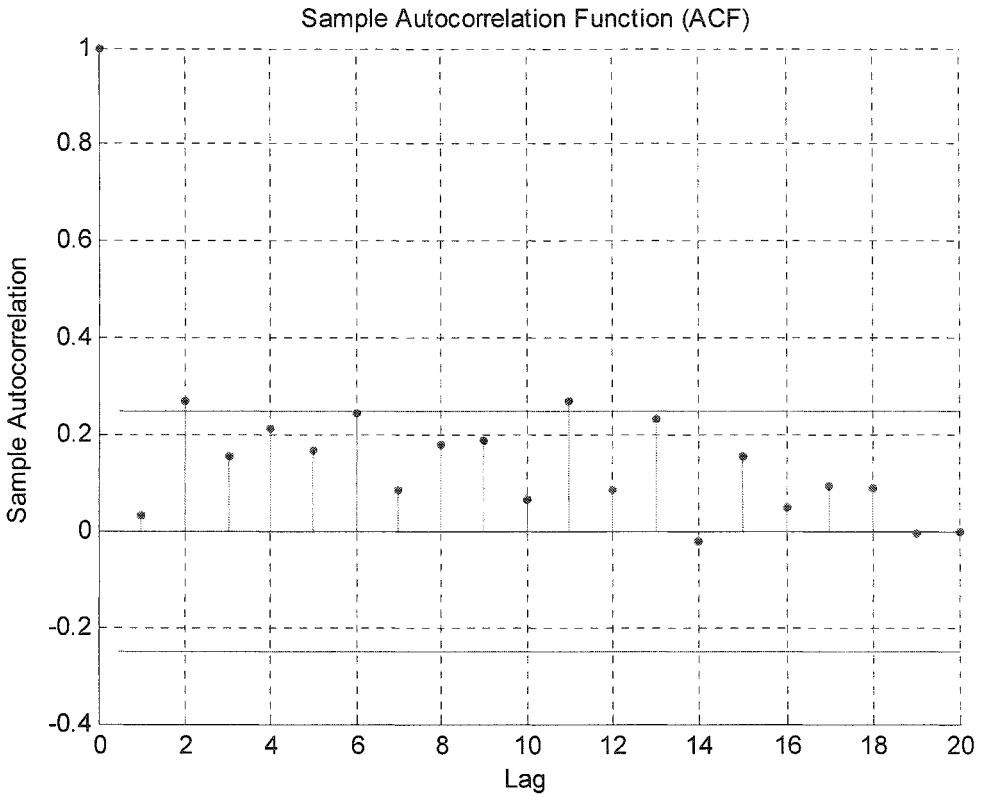


Figure 6.2 *Autocorrelation coefficients for lag 1 to 20 plotted in red node; the significance level is plotted in two blue horizontal lines*

Most of the autocorrelation coefficients from lag one to lag twenty are located between the two significance lines. The coefficient at lag two and lag eleven is a little greater than 0.25. However, the difference is not very significant. Therefore, it can be concluded that there is no significant autocorrelation in the original data. Based on the test result, most of the seasonal forecasting models are not suitable for the original data. Alternatively, the general auto-regression integral moving average model (*ARMA*) is applied in the research.

6.2 ARMA Model

Box and Jenkins (1976) developed the heuristics of the *ARMA* model, typically applied to time series data in statistics. Given a time series y_t, *ARMA* models are developed to predict future values. The model consists of two parts, the autoregressive (*AR*) part and the moving average (*MA*) part. The model is referred to as $ARMA(p,q)$ where p is the order of the autoregressive part and q is the order of the moving average part.

For example, the equation (6.5) shows the $ARMA(1,1)$ model. $\hat{y}_t$ is the forecast of the series at time period t. '$\phi_1 \cdot y_{t-1}$' is the AR part, ϕ_1 is the first order autoregressive parameter, ε_t is the stochastic error at time t. '$\theta_1 \cdot \varepsilon_{t-1}$' is the ($MA$) part. θ_1 is the first-order moving average parameter. δ is the constant term.

$$\hat{y}_t = \phi_1 \cdot y_{t-1} + \delta - \theta_1 \cdot \varepsilon_{t-1} + \varepsilon_t \; ; \tag{6.5}$$

$$\varepsilon_{t-1} = y_{t-1} - \hat{y}_{t-1} ; \tag{6.6}$$

However, one step back regression is not sufficient in most circumstances when dealing with realistic data sets. Hence, the higher order $ARMA$ model is required to improve performance.

$$y_t = \phi_8 y_{t-8} + \cdots + \phi_2 y_{t-2} + \phi_1 y_{t-1} + \delta - \theta_7 \varepsilon_{t-7} - \cdots - \theta_1 \varepsilon_{t-1} + \varepsilon_t ; \tag{6.7}$$

Equation (6.7) represents an $ARMA(8,7)$ model which regress eight steps back to y_{t-8} and includes seven moving average sectors. The parameters are calculated and optimized by the least square algorithm program to make the minimum sum of the squared differences between forecast and actual series value. Detailed calculations are presented in the next section.

Differently structured $ARMA$ models were tested against the original data. The best combination of autoregressive part and moving average part were selected. There are two ways to change model structure: first, varying the order of AR and MA; second, for the same order model, different regressed sectors are chosen to form the AR and MA part. For example, compared with $ARMA(8,7)$ in equation (6.7), in the second case, the model only picks y_{t-8}, y_{t-2} as the AR part and $\theta_1 \varepsilon_{t-1}$ as the MA part. Then, the expression changes to equation (6.8):

$$y_t = \phi_8 y_{t-8} + \phi_2 y_{t-2} + \delta - \theta_1 \varepsilon_{t-1} + \varepsilon_t \tag{6.8}$$

To choose the best structure of $ARMA$ model, the autocorrelation of the time series need to be calculated. Those data sets which have high correlation with the forecast target are chosen to construct the model. However, as shown in Figure 6.2, no

significant correlations existed in the data sets. In such circumstances, all combinations of *AR* and *MA* sectors need to be tested to construct different models. The forecasting error of each model, which is the accumulative absolute difference between forecast and real data, is recorded and compared. The model with minimum forecasting error is applied to carry out the prediction. This process is time consuming. A huge loop program was developed to perform this task. The program regresses eight steps back to form the autoregression part and seven steps back to form the moving average part.

It is important to note that whatever the model coefficients may be, the forecasting value y_t must be finite for the time series model. This leads to two conditions of the *ARMA* model, the stationarity condition for the autoregression coefficients and invertibility condition for moving average coefficients (Sufi, 1988). Considering an *ARMA(1,1)* model in equation (6.9)

$$y_t = \phi_1 \cdot y_{t-1} + \delta - \theta_1 \cdot \varepsilon_{t-1} + \varepsilon_t \tag{6.9}$$

Letting $t = t+1$, the equation can be expressed as

$$y_{t-1} = \phi_1 \cdot y_{t-2} + \delta - \theta_1 \cdot \varepsilon_{t-2} + \varepsilon_{t-1} \tag{6.10}$$

Substituting y_{t-1} in y_t, y_t is rewritten as

$$y_t = \phi_1^{\,2} y_{t-2} + (1+\phi_1)\delta + (\phi_1 - \theta_1)\varepsilon_{t-1} - \phi_1\theta_1\varepsilon_{t-2} + \varepsilon_t \tag{6.11}$$

The successive elimination of $y_{t-2}, y_{t-3} \ldots y_{t-n}$ will result in the removal of the autoregressive term and produces the final form of the model. Finally, by summing the infinite series and eliminating the negligible term, y_t can be expressed as below:

$$\begin{aligned} y_t = {} & \frac{\delta}{1-\phi_1} + (\phi_1 - \theta_1)\varepsilon_{t-1} + \phi_1(\phi_1 - \theta_1)\varepsilon_{t-2} + \phi_1^{\,2}(\phi_1 - \theta_1)\varepsilon_{t-3} + \cdots \\ & + \phi_1^{\,i}(\phi_1 - \theta_1)\varepsilon_{t-i+1} + \cdots + \varepsilon_t \end{aligned} \tag{6.12}$$

The expression is a pure moving average model with a constant term $\delta/(1-\phi_1)$ and moving average coefficients of $(\phi_1 - \theta_1), \phi_1(\phi_1 - \theta_1), \phi_1^{\,2}(\phi_1 - \theta_1) \ldots$ of first , second , and

third order respectively. It can be noted from the coefficient that y_t can only be finite if both the constant term and the different orders moving average coefficients $\sum_{i=0}^{\infty} \phi_1^{\,i}(\phi_1 - \theta_1)\phi_1$ are finite. This finite condition can only be met when $|\phi_1|<1$. Finally, it can be concluded that $|\phi_1|<1$ is the stationarity condition for the regressive term.

The invertibiltiy condition works on the moving average coefficients θ. Starting with the same *ARMA(1,1)* model, the error term ε_t can be expressed as:

$$\varepsilon_t = -\phi_1 y_{t-1} - \delta + y_t + \theta_1 \varepsilon_{t-1} \tag{6.13}$$

And, by analogy,

$$\varepsilon_{t-1} = -\phi_1 y_{t-2} - \delta + y_{t-1} + \theta_1 \varepsilon_{t-2} \tag{6.14}$$

Similarly, by substituting ε_{t-1}, and through successive substitution of $\varepsilon_{t-2}, \varepsilon_{t-3} \ldots y_t$, this can be written as:

$$\begin{aligned} y_t &= (1 + \theta_1 + \theta_1^{\,2} + \cdots)\delta + (\phi_1 - \theta_1) y_{t-1} + \theta_1(\phi_1 - \theta_1) y_{t-2} \\ &+ \theta_1^{\,2}(\phi_1 - \theta_1) y_{t-3} + \cdots + \theta_1^{\,i}(\phi_1 - \theta_1) y_{t-i-1} + \cdots + \varepsilon_t \end{aligned} \tag{6.15}$$

The *ARMA(1,1)* model has changed into being purely autoregressive, containing only the past observations y_{t-i} and the current error term ε_t. The model includes autoregressive parameters $(\phi_1 - \theta_1), \theta_1(\phi_1 - \theta_1), \theta_1^{\,2}(\phi_1 - \theta_1) \ldots$ providing relative weights. The model y_t can be finite only if both the sum of the parameter $\sum_{i=0}^{\infty} \theta_1^{\,i}(\phi_1 - \theta_1)\phi_1$ and the constant term are finite. To satisfy these conditions, the numerical value of θ_1 must be less than unity or $|\theta_1|<1$, the invertibility condition is $|\theta_1|<1$ for a time series process. Conforming to the invariability and stationarity condition, the time series need to be stationary. However, the time series observed did not meet the necessary condition of stationarity, and in such circumstances mathematical transformations must be performed to change the series to fulfil this requirement before further analysis can be undertaken. Without such transformation, further analysis would be inappropriate.

6.3 Stationary and Nonstationary Time Series

The concept of stationarity is best explained in terms of probability distribution. In a stationary time series, the probability distribution does not change over time. However, in the real world, there are as many nonstationary series as there are stationary ones (Gould, 2008, Suri, 1988). Most economic and business time series are, in fact, nonstationary, which is the nature of economic phenomena, containing growth and decay.

ARMA time series models are estimated only if the series is stationary and conforms to stationary and (inevitability) conditions. Nonstationary series needs to be processed to be stationary time series.

A series could be nonstationary in two ways, its mean and variance. The most common form of nonstationarity exists in the mean of a series, which is usually referred to as "trend in the mean". A series of this nature can easily be converted to a stationary series simply differencing the original series. The observed series, $y_1, y_2 \ldots y_t$ is sampled from a nonstationary time series process. If this series is differenced successively, a new series is then created, $w_1, w_2 \ldots w_{t-1}$, which is expressed as

$$\begin{aligned} w_1 &= y_2 - y_1 \\ w_2 &= y_3 - y_2 \\ &\ldots \\ w_{t-1} &= y_t - y_{t-1} \end{aligned} \tag{6.16}$$

By differencing the original series, the trend has been removed and the mean of new series w is around zero.

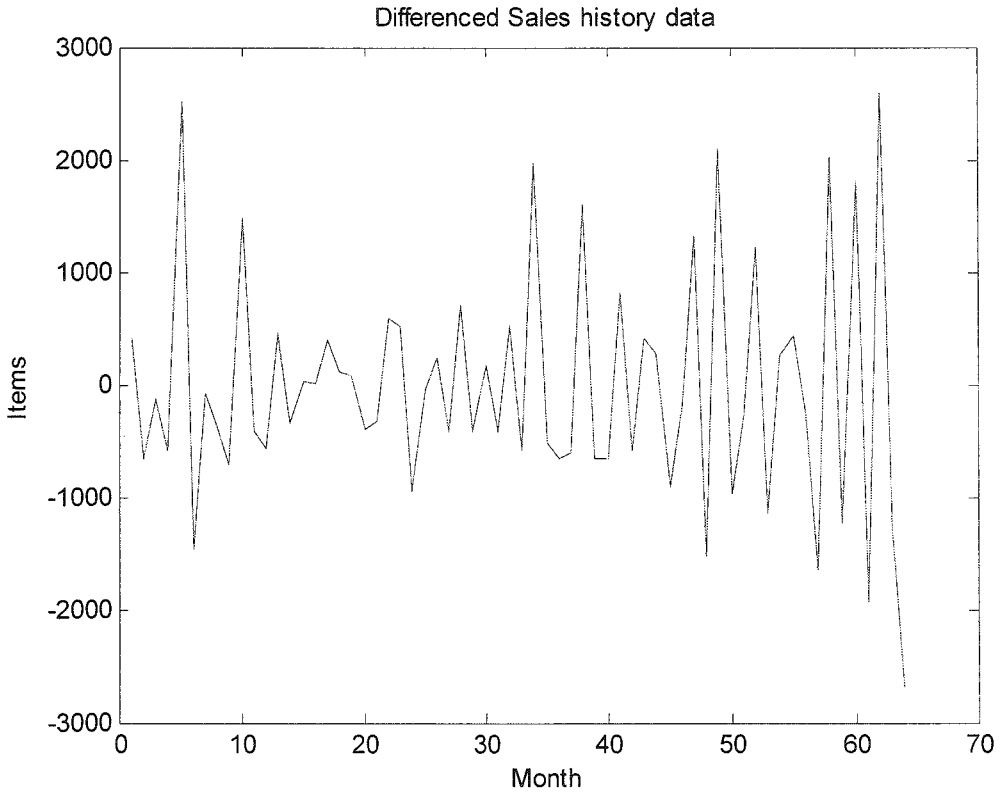


Figure 6.3 *Differenced sales history data*

As shown in Figure 6.3, the growing trend in the original data has been removed. However, the oscillations are even bigger. The negative values in the differenced series have no physical meaning. After making the forecast based on the processed data, the final result will be post processed to the original scale with de-differencing.

On the other hand, the logarithmic transformation is applied to remove the variability from the series. The processing is shown in equation (6.17). However, normally the variability cannot be completely removed. Such a transformation generally removes a substantial part of the variability but rarely removes it completely.

$$w_t = Ln(y_t); \tag{6.17}$$

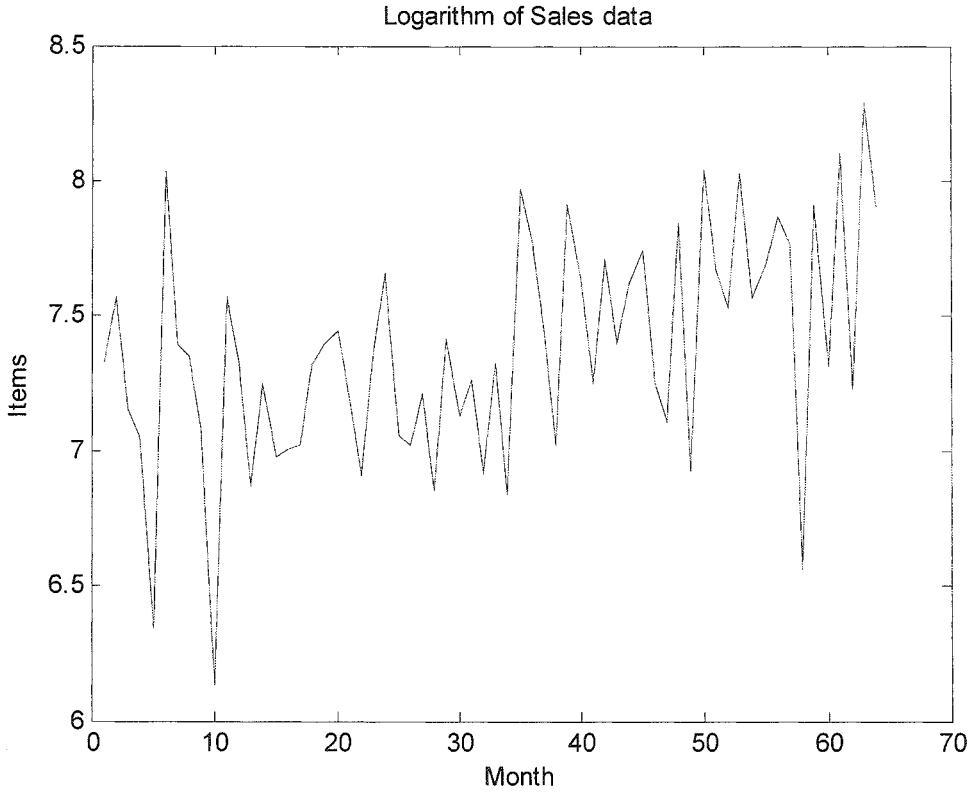


Figure 6.4 *Logarithm of sales history data*

Figure 6.4 show the logarithm processed time series data. The variation scale in Figure 6.4 has become smaller. The range of the oscillation is from 6 to 7.5. The overall shape of the logarithm data remains the same as the original. The growth trend still exists in the data and the mean is not zero. After logarithm processing, the processed data is still not stationary.

Applying the logarithm first and then differencing, both the trend and variation can be partially removed. Figure 6.5 shows the preprocessed data. The oscillation range becomes much smaller, and the growth trend is removed. The mean of the processed data is nearly zero. The series has become the stationary time series data. Forecasting methods were then applied to these data. The results are compared in later sections.

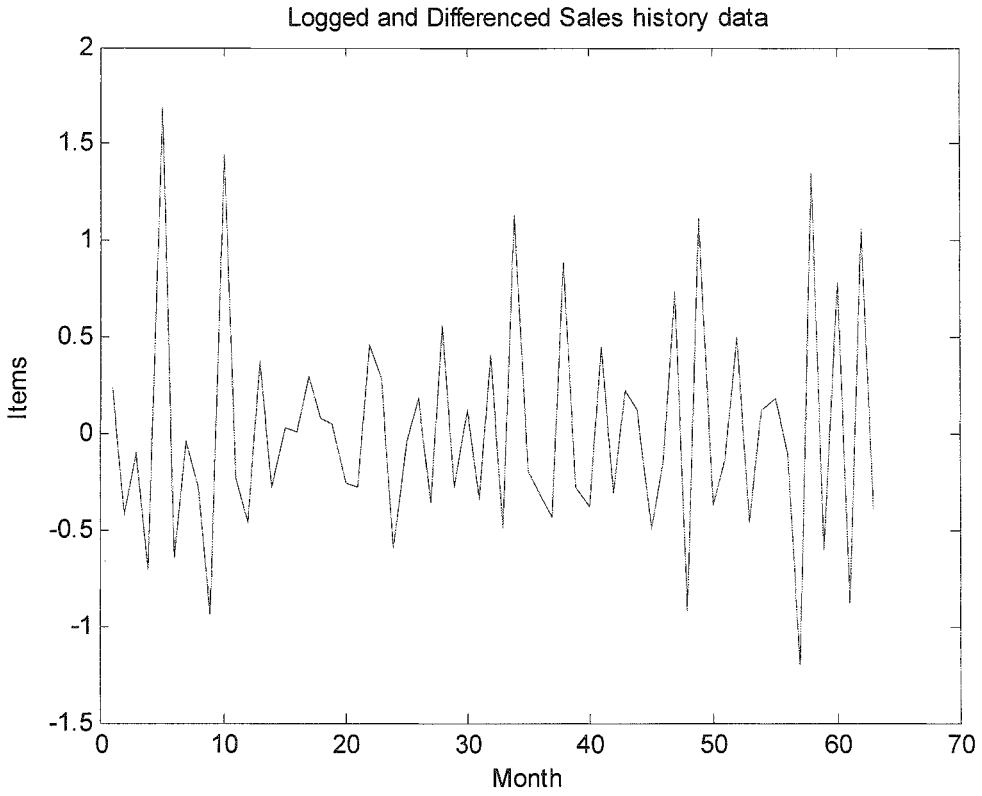


Figure 6.5 *Logarithm and differenced sales history data*

Such pre-processing makes the time series data have zero mean and smaller oscillation. However, in some cases, the data might still be very nonstationary. It needs ofdifferencing and logarithm processing further steps. However, after too much stationary processing, the data may loose much of its information. When the data is converted back to the original scale, the result could be a nonsensical forecast. Generally, two or three times of stationary processing is the maximum. In our case, one step stationary processing was performed.

6.4 Least Square

The optimization algorithm Least Square is applied to calculate and optimise the parameters of the *ARMA* model. The least square technique was originally developed by Gauss and Legendra in the early 19^{th} century. It is thus defined by Gauss: the most probable value of the unknown quantities will be that in which the sum of the squares of the differences between the actually observed and the computed values multiplied by numbers, which measure the degree of precision is a minimum.

Suppose our original data Y_t can be produced from a stochastic process for which the mean value is a linear function of a parameter vector θ, i.e.

$$E[Y_t] = x_t^T \theta \tag{6.18}$$

where x_t is a known vector for t = 1, … ,r. The aim is to obtain a good estimation of θ denoted by $\hat{\theta}$. Values of $\hat{\theta}$ are calculated which minimise the sum of the squares of the deviations between the observation, y_t, and the mean of the predicted model, namely, $x_t^T \hat{\theta}$, we attempt to minimise

$$s = \sum_{t=1}^{r} (y_t - x_t^T \hat{\theta})^2 \; ; \tag{6.19}$$

s is the difference between real data y_t and prediction $x_t^T \hat{\theta}$. Differentiating the function of s with respect to $\hat{\theta}$ gives

$$\frac{\partial s}{\partial \hat{\theta}} = -2x^T y + x^T x\theta + (x^T x)^T \theta \; ; \tag{6.20}$$

Equating $\partial s / \partial \hat{\theta}$ to zero yields

$$\hat{\theta} = [x^T x]^{-1} x^T y \; ; \tag{6.21}$$

When $\partial s / \partial \hat{\theta}$ is zero, s in equation (6.19) reaches the minimum value and the solution of $\hat{\theta}$ in equation (6.20) is the parameter to make this least squared difference in equation (6.19). In our case, y_t and x_t are both the original time series data with different lags. $\hat{\theta}$ is the parameter matrix for ϕ, δ and θ, which is used in the *ARMA* model. The *ARMA*(2,1) model is shown in equation (6.24). $\hat{y}_t$ is the forecast of y_t

$$x = \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix}; \quad y = [y_t]; \tag{6.22}$$

$$\hat{\theta} = [x^T x]^{-1} x^T y = [\phi_1 \quad \phi_2 \quad \delta \quad \theta_1]; \tag{6.23}$$

$$\bar{y}_t = x^T \hat{\theta} = \phi_1 \cdot y_{t-1} + \phi_2 \cdot y_{t-2} + \delta - \theta_1 \cdot \varepsilon_{t-1} + \varepsilon_t; \qquad (6.24)$$

The $\hat{\theta}$, generated from the equation (6.23), can only minimise the square difference of $\bar{y}_t$. If the forecasting target changes to be y_{t+1} or y_{t-1}, the parameter $\hat{\theta}$ is not suitable to produce the forecast. The logic of our forecasting with the *ARMA* model was using least square methods to generate the best parameters which could minimise the squared error for all the history data, and then applying these parameters to the forecasting model to predict one step forward future data. The hypothesis is, if the parameters $\hat{\theta}$ can minimise the square error for all the history, it might work for the immediate future. $\hat{\theta}$ needs to be updated for each round of calculations. The classic least square, discussed previously, is inefficient to perform this task. If the estimated parameter $\hat{\theta}$ is made from *r* observations, then it is not possible to make use of this prior knowledge subsequently. It is necessary to update the estimates having obtained one more observation, making a total of *r+1* observations. Suppose the inputs matrix and target matrixes are as below

$$X = \begin{bmatrix} y_{r-1} & y_{r-2} \\ y_r & y_{r-1} \\ \cdots & \\ y_{r+n-1} & y_{r+n-2} \end{bmatrix}; \quad Y = \begin{bmatrix} y_r \\ y_{r+1} \\ \cdots \\ y_{r+n} \end{bmatrix}; \qquad (6.25)$$

Setting $x_r = \begin{bmatrix} y_{r-1} & y_{r-2} \end{bmatrix}$, then the inputs matrix becomes

$$X = \begin{bmatrix} x_r \\ x_{r+1} \\ \cdots \\ x_{r+n} \end{bmatrix}; \qquad (6.26)$$

The forecast is y_{r+n+1}. The estimated parameter $\hat{\theta}_n$ is calculated recursively based on the history data and previous $\hat{\theta}$. The recursive least square is employed as the optimisation method to produce $\hat{\theta}$. The algorithm is explained in the equations below:

$$\hat{\theta}_{r+1} = \hat{\theta}_r + k_{r+1}\left(y_{r+1} - x_{r+1}^T \hat{\theta}_r\right); \qquad (6.27)$$

$$k_{r+1} = \frac{p_r x_{r+1}}{1 + x_{r+1}^T p_r x_{r+1}}; \tag{6.28}$$

Furthermore, p_r can be computed recursively from:

$$p_{r+1} = \left[I - \frac{x_{r+1}}{1 + x_{r+1}^T p_r x_{r+1}} \right] p_r; \tag{6.29}$$

p_r is defined as $p_r = \left[x_r^T x_r \right]^{-1}$. After executing the recursive process, the parameter $\hat{\theta}_n$ is generated and applied to the forecast.

$$\hat{y}_{r+n+1} = x_{r+n+1}^T \hat{\theta}_{r+n} = \phi_1 \cdot y_{r+n} + \phi_2 \cdot y_{r+n-1} + \delta - \theta_1 \cdot \varepsilon_{r+n} + \varepsilon_{r+n+1}; \tag{6.30}$$

This theoretical procedure was programmed and implemented in Matlab as follows:

Make forecast for t month sales y_t *and process till the end of original series at* y_{64}

Construct the ARMA(p,q) model, with $p<t$ *and* $q<t$

While t < size of original data

Construct input matrix X and target matrix Y

$X = \left[y_{t-2} \quad \cdots \quad y_{t-p-1} \right]; \quad Y = \left[y_{t-1} \right];$

Calculate P_t, K_t *according to recursive least square algorithm*

Update $\hat{\theta}_{t-1}$ *as* $\hat{\theta}_{t-1} = \hat{\theta}_{t-2} + k_{t-1}\left(y_{t-1} - x_{t-1}^T \hat{\theta}_{t-2} \right)$

Constructing the forecast matrix X_f *as*

$X_f = \left[y_{t-1} \quad \cdots \quad y_{t-p} \right];$

Produce the forecast $\hat{y}_t$ *as* $\hat{y}_t = X_f^T \hat{\theta}_{t-1}$

Calculate the difference between $\hat{y}_t$ *and* y_t, ε_t

Add the absolute difference to the performance function

Update t as $t+1$

After setting p and q, the frame of the regression part and the moving average part are constructed. The forecast produced by this *ARMA(p,q)* model will be $\hat{y}_{p+2}$ or $\hat{y}_{q+2}$,depending on the value of p and q. Assuming $p > q$, the first forecast generated in the first loop will be $\hat{y}_{p+2}$, which is the $\hat{y}_t$ in the program. The data before t, i.e. y_p, y_{p-1}, y_{p-2} and successive terms become components of the autoregression terms in the model. These data also form the input matrix $X \lfloor y_p \quad \ldots \quad y_1 \rfloor$. The data y_{p+1} which is y_{t-1} becomes the result or target of the *ARMA(p,q)* equation. It is on the left side of the model equations (6.31).

$$y_{p+1} = \phi_1 \cdot y_p + \cdots + \phi_p y_1 + \delta - \theta_1 \cdot \varepsilon_p - \cdots - \theta_q \varepsilon_{p-q+1} + \varepsilon_{p+1} \tag{6.31}$$

Then the variables of the recursive least square can be calculated from the equation. The coefficient matrix $\hat{\theta}_t$ is computed from P_{p+2}, K_{p+2}.

$$\hat{\theta}_{p+1} = \hat{\theta}_p + k_{p+1}\left(y_{p+1} - x_{p+1}^T \hat{\theta}_p\right) \tag{6.32}$$

There are two parts in the calculation (6.32) for $\hat{\theta}_{p+1}$. The first factor is the last step $\hat{\theta}_p$. The other is a polynomial of y_{p+1}. The last step $\hat{\theta}$ is inherent from the last round of the loop. In the initial situation, $\hat{\theta}$ is a zero matrix. In the second part, K_{p+2} can be calculated from equation (6.28). The x_{p+1}^T is the same as the transpose of the input matrix $X \lfloor y_{p+1} \quad \cdots \quad y_1 \rfloor$. It is quite clear that the second part considers $y_{p+1} - x_{p+1}^T \hat{\theta}_p$, the difference of the last step forecast and real value, where $x_{p+1}^T \hat{\theta}_p$ is the expression of $\hat{y}_{p+1}$.

Using $\hat{\theta}_{p+1}$, the forecasting for the time series before p+2 achieved minimum squared error. It can be assumed that this parameter could be suitable for one step ahead future y_{p+2}. With these parameters $\hat{\theta}_{p+1}$, combining with the forecast matrix X_f $\lfloor y_{p+1} \quad y_{p+1} \cdots \quad y_1 \rfloor$, the next step forecast $\hat{y}_{p+2}$ is calculated as:

$$\hat{y}_{p+2} = x_{p+2}^T \hat{\theta}_{p+1} \tag{6.33}$$

The difference between $\hat{y}_{p+2}$ and real value y_{p+2} will be recorded to use as the component of the future *ARMA* model and count as the accumulative forecasting error in performance function. the *ARMA(p,q)*'s forecasts arrive at the minimum square error for $\hat{y}_{p+1}$. Finally, the program moves back to the start to run the whole process for the prediction $\hat{y}_{p+3}$. The program loops until generating $\hat{y}_{64}$. The program code is provided in Appendix B.

6.5 Forecasting Program and Results

Based on the analysis of the data preprocessing, four different forms of preprocessing are illustrated. Our forecasting program tested the performance of the forecasting in the following sequence: data without any preprocessing; applying logarithm to the data; differencing the original data; applying both logarithm and differencing to the data. Since the differenced series had some minus numerical value, which could not be processed with logarithm, the differencing and logarithm preprocessing was not applied. For each of these four branches, the same procedure was followed.

Since there was no significant correlation within the original data, during the constructing of the *ARMA(p,q)* model, all the combinations of the autoregression terms $[y_{t-8} \ldots y_{t-1}]$ and moving average terms $[\varepsilon_{t-7} \ldots \varepsilon_{t-1}]$ were tested to form the structure of *ARMA(p,q)*. The absolute value of the difference between forecast and real data for each particular *(p,q)*structured model were summed. This summed difference was treated as the forecasting error in order to measure their forecasting performance. Forecasting with minimum forecasting error was selected and the results are documented in the following sections.

6.5.1 Data without Any Preprocessing

Without any preprocessing, the operational data for forecasting is the original data. The program generated the best *ARMA* model with autoregression $[y_{t-2}\ \ y_{t-4}\ \ y_{t-5}\ \ y_{t-6}]$ and no moving average. It can be written as:

$$\hat{y}_t = \phi_1 \cdot y_{t-2} + \phi_2 \cdot y_{t-4} + \phi_3 \cdot y_{t-5} + \phi_4 \cdot y_{t-6} + \delta + \varepsilon_t \qquad (6.34)$$

The coefficient matrix $\hat{\theta}\,[\phi_1\ \phi_2\ \phi_3\ \phi_4\ \delta\,]$ is produced and updated after the eighth sample in the original data. The forecast is plotted in green while the original sales history data is in blue in Figure 6.6.

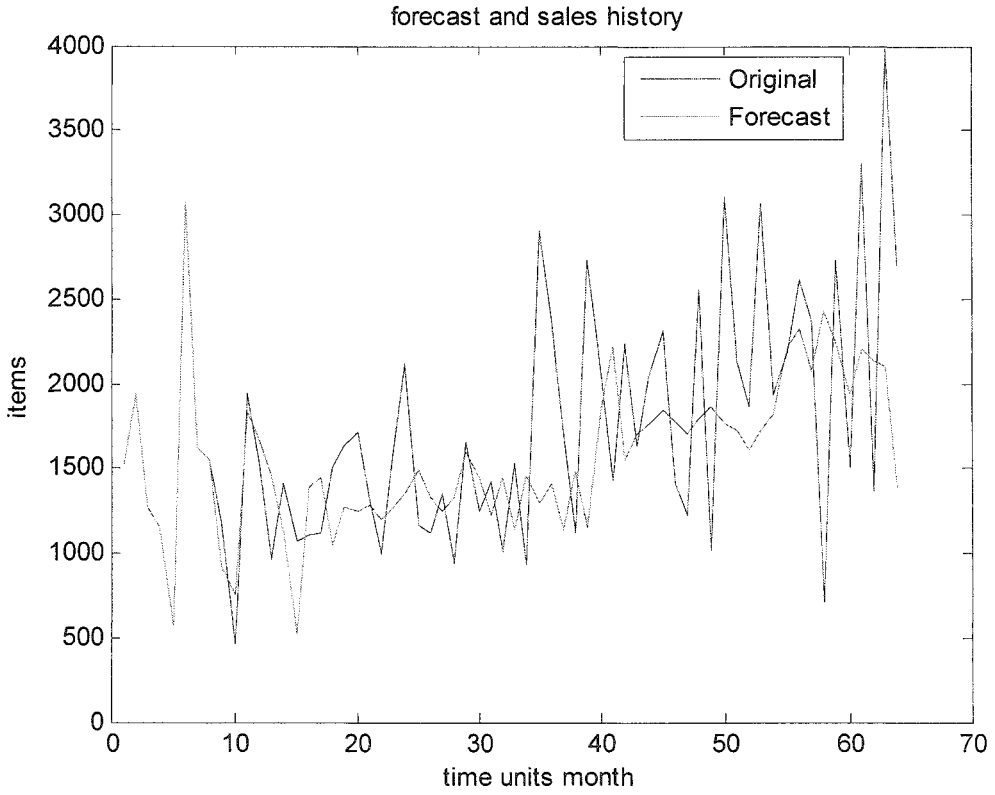


Figure 6.6 *The plots of forecast accompany with original sales*

The green and blue curves are perfectly matched in the first eight months. This is because the first eight months' data were required to construct the model and calculate the model parameters. The forecast in this period is set to be the same as the original. The coefficient matrix $\hat{\theta}$ is updated recursively to fit the data pattern. The programs need to learn the data pattern from the previous data history. Hence, forecasting error were calculated for the last 15 months' forecasts only, which are the data sample in the range of 50 - 64 (Figure 6.6). The overall forecast performance is poor. The green curve reluctantly follows the overall trend of the blue curve. There is significant difference in the details of the variation. The accumulative absolute difference is 11806 units. The average forecasting error for these 15 months is 787 units. In the next step, more testing was done to improve the forecasting performance.

6.5.2 Logarithm Preprocessing

In this section, logarithm processing is applied to the original data. The preprocessed series are programmed to generate the forecasting model. After producing the forecast result, exponent processing is required to counteract the effects of logarithm preprocessing. The program that generates the best *ARMA* model for logarithmic data has three autoregression factors at $[\, w_{t-2} \;\; w_{t-3} \;\; w_{t-8} \,]$ and a single moving average factor at $[\, \varepsilon_{t-3} \,]$. It can be written as:

$$\hat{w}_t = \phi_1 \cdot w_{t-2} + \phi_2 \cdot w_{t-3} + \phi_3 \cdot w_{t-8} + \delta - \theta_1 \varepsilon_{t-3} + \varepsilon_t \tag{6.35}$$

where

$$w_t = Ln(y_t) \tag{6.36}$$

Logarithm processing diminishes the variation scale, but does not remove the growth trend from the original. Since the regression parts of the model constitute the term w_{t-8}, the coefficient matrix $\hat{\theta}\,[\phi_1 \;\; \phi_2 \;\; \phi_3 \;\; \delta \;\; \theta_1]$ is computed and updated from month 10 onwards. The first nine months' data are used as initializing and building the structure of the model. The forecasting results are plotted in green against the processed original data in blue in Figure 6.7.

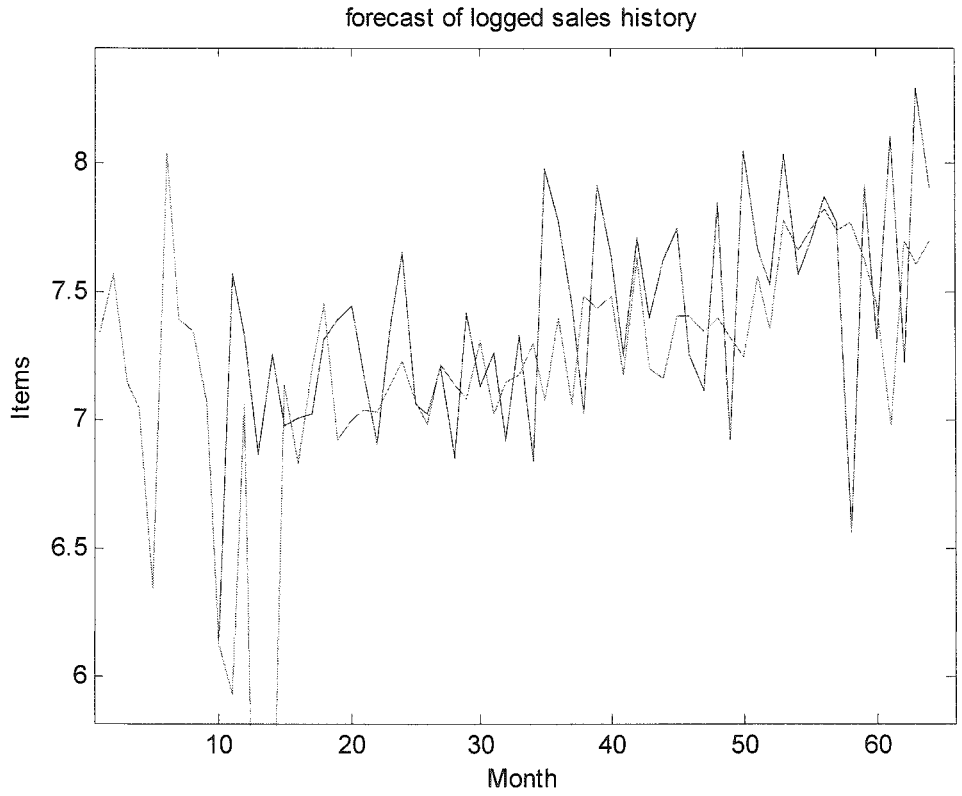


Figure 6.7 *Plots of forecast against logarithmic data*

It is obvious that there is a huge discrepancy between forecasting and processed original data from the beginning of month 10 to month 20, when the model is initializing the parameters. It is wise to ignore the forecasting error of this large oscillation when the program is still accommodating the coefficient matrix $\hat{\theta}$. Therefore, concentration is still on the last 15 months' forecasts. From month 50 to month 60, the forecast increases and follows the logarithmic data pattern; the difference of the detail variation is still great. However, after month 60, the forecast cannot track the movement of the logarithmic data. Neither the trend nor the variation pattern can be followed.

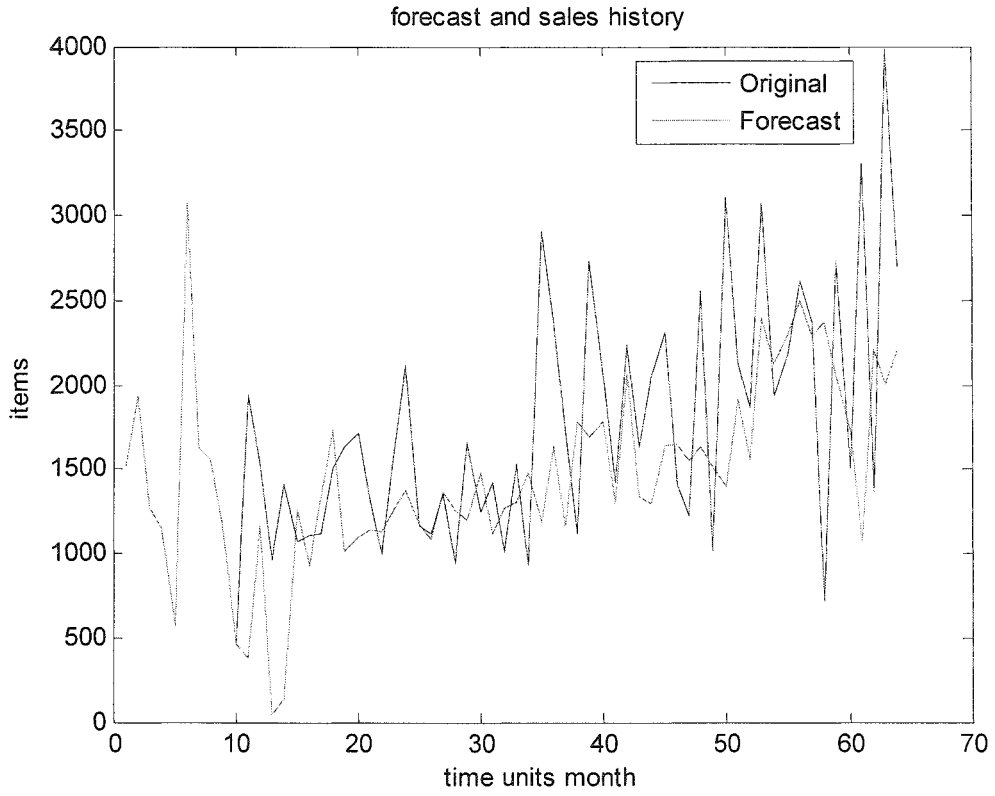


Figure 6.8 *Plots of exponent forecast and original data*

The exponential function is applied to the forecasting results (equation (6.37)) as postprocessing. The exponential forecasting result is plotted against the original data in Figure 6.8.

$$\hat{y}_t = \exp(\hat{w}_t) \tag{6.37}$$

The exponential forecast results do not match the overall shape of the original data, and show a greater variation scale. The accumulative absolute difference for the last 15 months is 11535 units. The average forecasting error is 769 units. Compared with the forecasting without any preprocessing, logarithm preprocessing improved the forecasting performance on a small scale. The accumulative forecasting error becomes smaller. However, the overall performance is still not good.

6.5.1 Differencing Preprocessing

Differencing processing is the successive subtraction between two continued series. The model generates minimum forecasting error when the autoregression part is $[w_{t-1}\ w_{t-2}\ \ w_{t-4}\ w_{t-5}\ \ w_{t-8}]$, and the moving average part is $[\varepsilon_{t-3}]$. It can be written as:

$$\hat{w}_t = \phi_1 \cdot w_{t-1} + \phi_2 \cdot w_{t-2} + \phi_3 \cdot w_{t-4} + \phi_4 \cdot w_{t-5} + \phi_5 \cdot w_{t-8} + \delta - \theta_1 \varepsilon_{t-3} + \varepsilon_t \qquad (6.38)$$

where

$$w_t = y_{t+1} - y_t \qquad (6.39)$$

The differencing has changed the overall shape of the original. The mean of the new series is around zero. The forecasting results in green are plotted against differenced original data in blue in Figure 6.9.

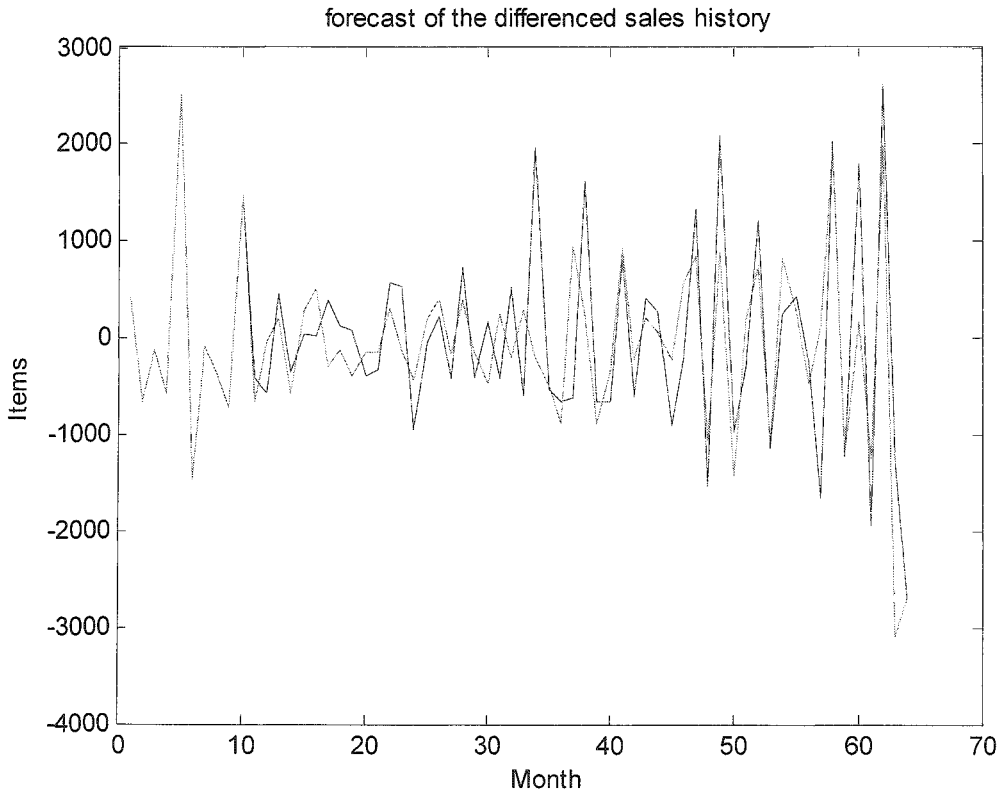


Figure 6.9 *Plots of differenced data and forecast*

Since the furthest regressive part is w_{t-8}, the coefficient matrix $\hat{\theta}\,[\phi_1\ \phi_2\ \phi_3\ \phi_4\ \phi_5\ \delta\ \theta_1]$ is initialized in the first 10 months data. Ignoring the starting stage and the last 15 months, forecasts are selected to measure the performance. It can be seen from Figure 6.9 that the forecast essentially matches the variation pattern in the last 15 months. However, an enormous gap still exists in some details of oscillation amplitudes.

During the postprocessing illustrated in equation (6.40), the first original data y_1 is kept as the origin. The forecast for the second original data y_2 is the sum of the forecast of the first differencing data and this origin y_1. The rest of the postprocessed data can be calculated recursively according to the formula

$$\hat{y}_2 = y_1 + \hat{w}_1 \tag{6.40}$$

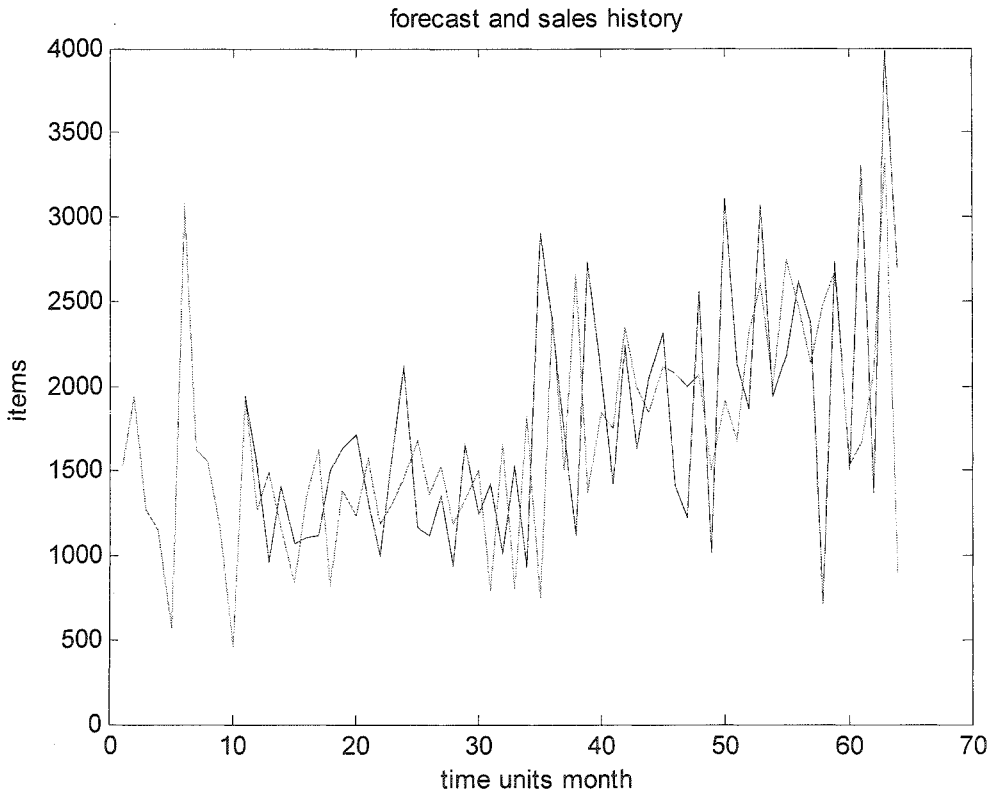


Figure 6.10 *Plots of forecast and original*

The postprocessed forecast is plotted against original data in Figure 6.10. Comparing Figure 6.10 to Figure 6.9, the postprocessed forecast does not perform as well as the forecasting for the differenced original data. The discrepancy in the amplitude has been enlarged so that this forecast is not reliable. The accumulative differences are 10219

units from month 50 to 64. The average forecasting error is 681 units. The error is smaller than both preprocessing methods discussed before. However, the forecasting performance has not improved significantly.

6.5.3 Logarithm and Differencing Preprocessing

In this section, the original series are processed by the logarithm function first to diminish the variation, and then the logarithmic data are differenced to remove the trend. In this case, the model generating the minimum forecasting error has the autoregression part as $[\, y_{t-1}\ y_{t-2}\ y_{t-5}\ \ y_{t-8}\,]$ and $[\, \varepsilon_{t-5}\,]$ for the moving average part. The equation expression is

$$\hat{w}_t = \phi_1 \cdot w_{t-1} + \phi_2 \cdot w_{t-2} + \phi_3 \cdot w_{t-5} + \phi_4 \cdot w_{t-8} + \delta - \theta_1 \varepsilon_{t-5} + \varepsilon_t \qquad (6.41)$$

where

$$z_t = y_{t+1} - y_t\,; \qquad w_t = Ln(z_t) \qquad (6.42)$$

Figure 6.11 plots the preprocessed data against forecasting. The data scale becomes smaller (from -1.5 to 2). The mean of the processed data is around zero. In the last 15 months, from month 50 to month 64, the forecast can basically match the variation pattern of the original data. However, the differences between the oscillation amplitudes are still very great.

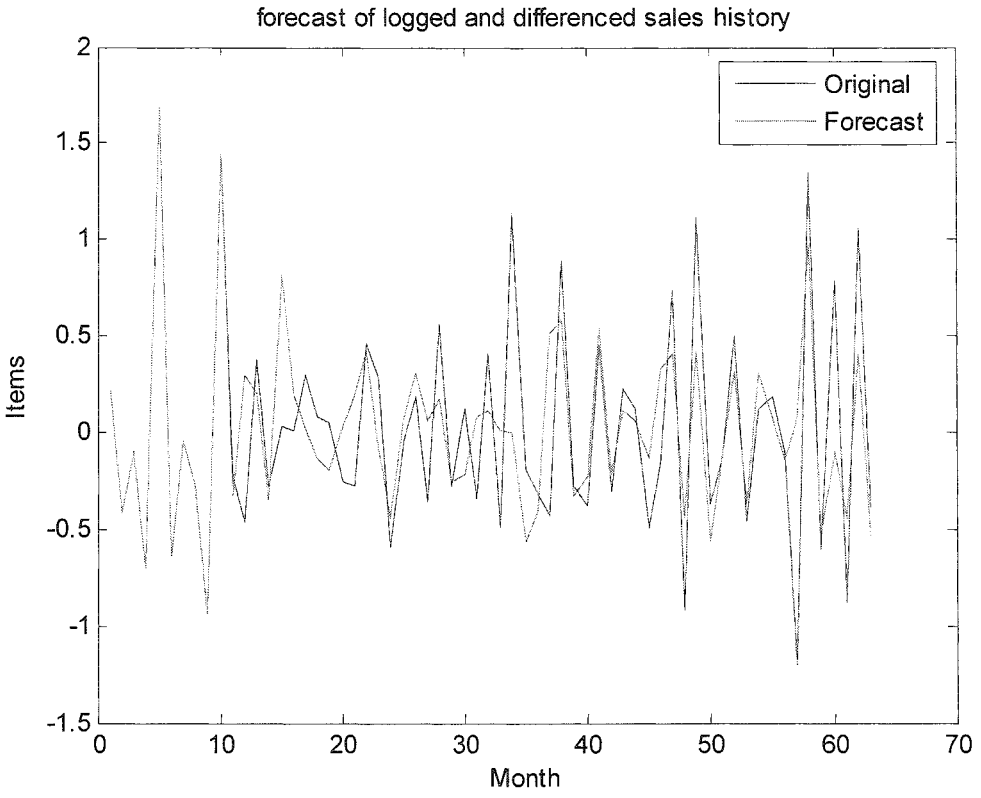


Figure 6.11 *Plots of logarithm and differenced original data and forecast*

The postprocessing in this case includes two steps, adding the origin to the forecast value and then calculating the exponent of the result. The postprocessing results are plotted against the original data in Figure 6.11. The forecast predicts neither the trend nor the variation pattern. The *ARMA* model works well only for the stationary preprocessed data with a mean of zero and small variation. But if the results are converted back to the original scale, the forecast results are not acceptable.

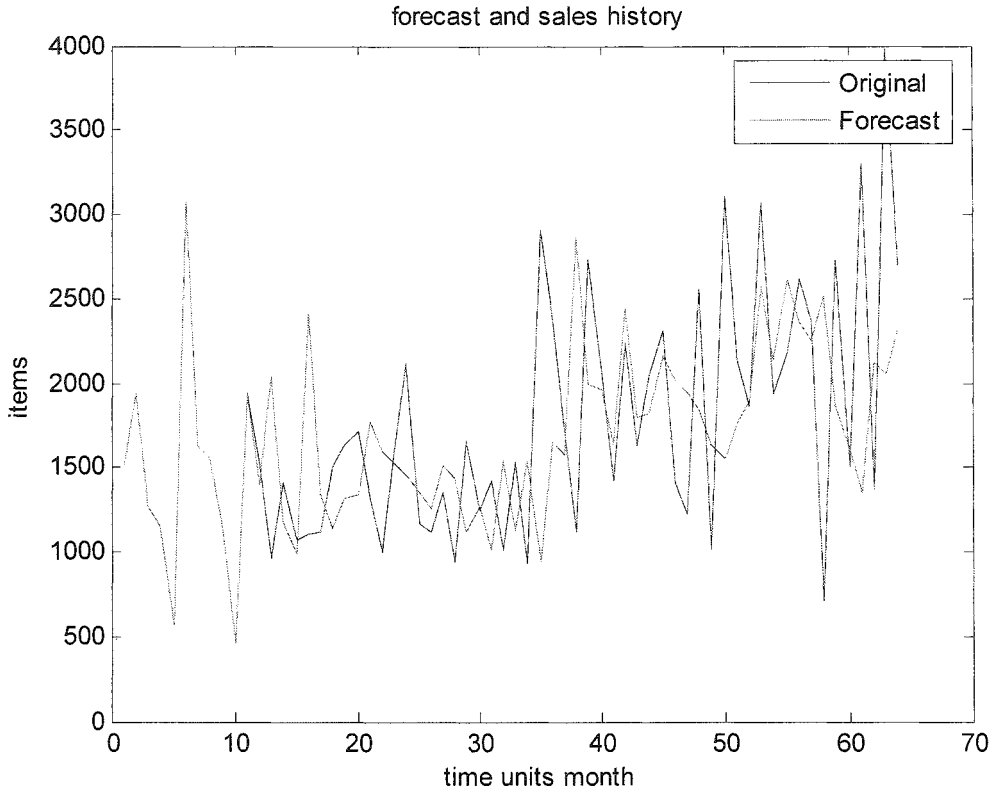


Figure 6.12 *Plots of forecast and original data*

In the logarithm and differencing model, the accumulative difference is 11258 units and the average forecasting error is 750 units. The forecasting error is greater than difference preprocessing but still smaller than the model without any preprocessing. It can be concluded that forecasting with logarithm and difference preprocessing is not reliable.

6.6 Effect of Different Forecasting Methods on the Supply Chain Model

In this section, the forecasts generated by the first-order exponential prediction formula (the forecasting equation with Θ) and *ARMA* are compared to the hub forecasting in the original supply chain model. The performance of each forecasting method is measured by the cost of holding inventory and backlog orders. It is calculated in equation (6.43).

$$\text{Cost} = F_{inv}(t) + H_{inv}(t) + 2 \times H_{blk}(t) \qquad (6.43)$$

The 64 months' sales history of the lung demand valve in Draeger is applied as the sales demand series. The safety stock is fixed at 150 units (the same as the real safety stock in Draeger). Different forecasting methods need different training periods to track and learn the data pattern of the time series. In order to avoid the effects of training periods, the last 15 months' forecasting data are chosen to measure their performance. The last 15 months' sales demands are shown in Figure 6.13

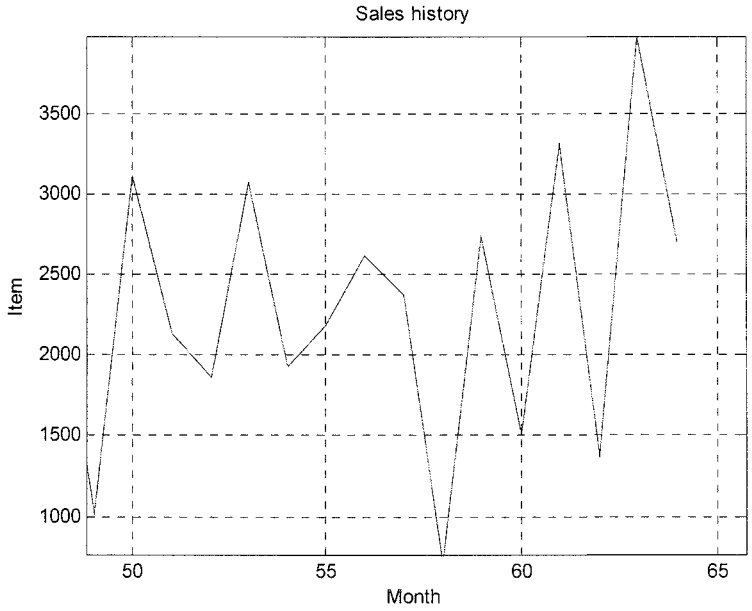


Figure 6.13 *The Last 15 Month Sales History*

In the ideal case, the forecast perfectly matches the incoming sales orders. The factory production can always deliver the right amount of inventories to satisfy the demand without any backlog order. Both factory and hub inventory are stable at the safety stock level. In this situation, holding safety stock in the hub and factory inventory becomes the major cost.

$$\text{Costs} = 150 \times 15 + 150 \times 15 = 4500$$

However, as perfect forecasting does not exist, the costs are always higher than 4500 units. In the first test, the first-order exponential forecasting formula was applied to minimise the cost of holding inventory and backlog order. The forecasting formula is

$$H_{forecast}(t+2) = (1 - \theta) H_{orders}(t) + \theta H_{forecast}(t+1) \qquad (6.44)$$

Changing α, β and θ from 0 to 1 in 0.1 step, the supply chain model achieved the minimum cost at 22427 units when α=0.2, β= 0 and θ= 0. The forecasts are plotted in red against sales inputs in blue in Figure 6.14.

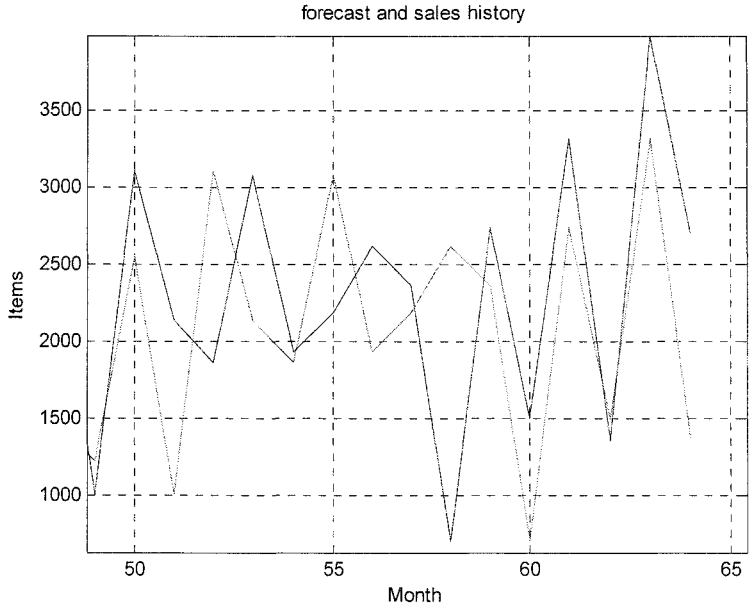


Figure 6.14 *First-order exponential forecasting result (red) and original sales data (blue)*

When θ= 0, the forecast in formula (6.44) is equal to the actual sales of two months ago. Inspection of Figure 6.14 shows the forecast (in red) is delayed by two months compared to the sales curve (in blue). Since there is no production limitation, the factory production can meet any hub requirements. The factory inventory in Figure 6.15 is stable at the safety stock level. However, the hub effective inventory displays great oscillations, with a peak of 1500 units and a low of -2000 units.

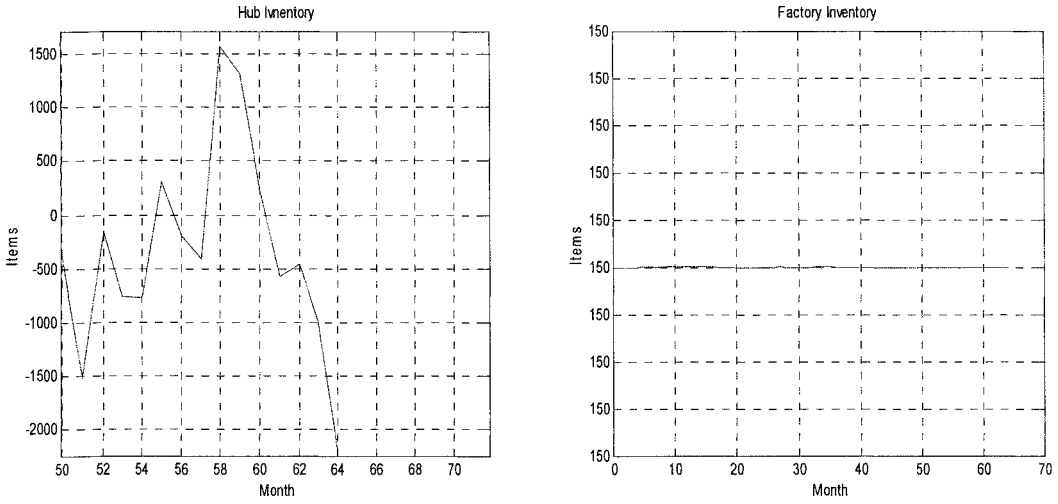


Figure 6.15 *Inventory response with first order exponential forecasting*

In the second test, the *ARMA* model with differencing pre-processing forecasting result was applied in the model. The minimum supply chain costs are 18524 units when α=0.1 and β= 0. The 15 months' forecast in green and sales in blue are plotted in Figure 6.16. The green forecast curve does not follow the oscillation of the blue sales. However, compared with the first-order exponential forecasting, the absolute difference is relatively small.

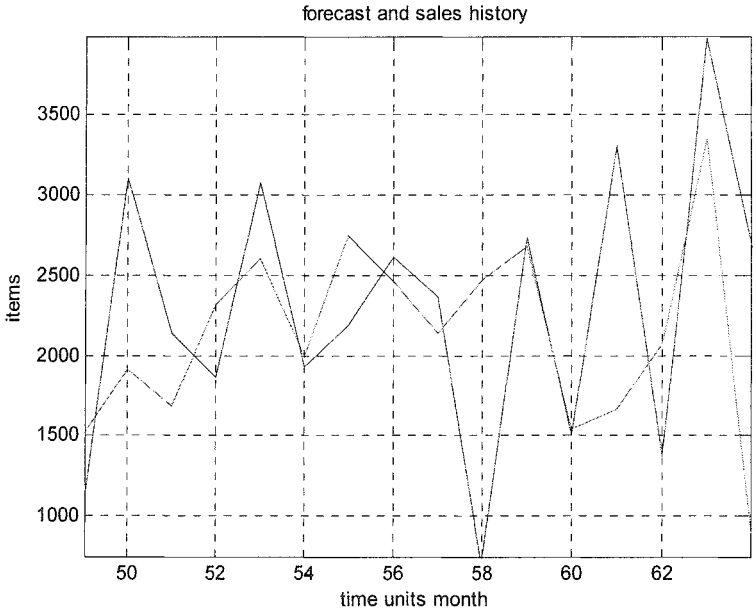


Figure 6.16 *ARMA forecasting result (green) and original sales data (blue)*

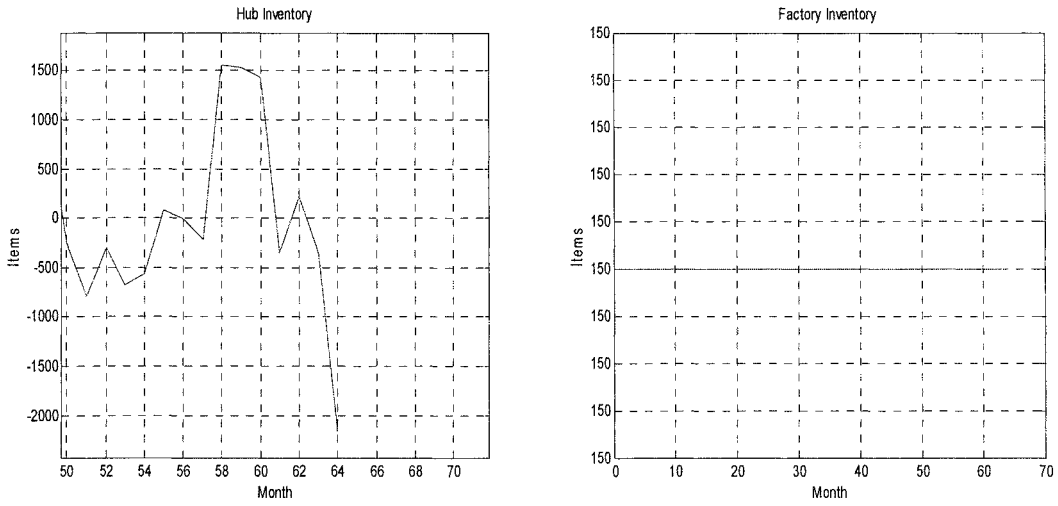


Figure 6.17 *Inventory response with ARMA forecasting*

The hub inventory plotted in Figure 6.17 still displays large scale oscillations, with the lowest backlog at –2000 units and the inventory peak at 1600 units. The hub inventory varies in a similar range to that in Figure 6.15. The system performance has apparently not been improved. The relevant supply chain costs are 18524 units, which is much bigger than the ideal cost.

After testing the *ARMA* model with four type of preprocessing, it can be concluded that the overall forecasting performance is not reliable. The absolute accumulative and average error of these four forecasting methods has been listed in Table 6.1. The minimum cost is generated by the *ARMA* $[\phi_1\ \phi_2\ \phi_3\ \phi_4\ \phi_5\ \delta\ \theta_1]$ model with differencing preprocessing. However, the cost is still too high to be acceptable.

	No Preprocessing	Logarithm	Differencing	Logarithm and Differencing
Accumulative error	11806	11535	10219	11258
Average error	787	769	681	750
Supply chain costs	24313	23176	18524	22109

Table 6.1 *The costs of the forecast*

From the analysis given here, the *ARMA* model could produce a good forecast on the stationary preprocessed data which has a mean of zero and small variation (Figure 6.11 and Figure 6.9). However if the forecasts were converted back to the original series scale, the post processed data could not track the data pattern and predict the trend or variations (Figure 6.10 and Figure 6.12). The *ARMA* model is a linear forecast model which is suitable for analyzing the stationary series only. However the research sample data is quite nonstationary and has inbuilt nonlinearity. Thus, other nonlinear forecast techniques are required to handle this problem.

Chapter 7:

Neural – Wavelet Forecasting

Financial time series forecasting has become one of the biggest challenges in the application of forecasting. Francis and Cao (2001) have explained that the financial time series is nonstaitionay and inherently noisy. Previous studies have demonstrated that market price and sales movements vary in a highly nonlinear, dynamic manner, rather than purely randomly (Blank,1991). To deal with the embedded nonlinearity and increase the reliability of forecasting, neural networks have received considerable attention (Hsiao, 2006; Xie, Zhang and Ye, 2007; Grudnitski & Osburn, 1993). Neural networks provide universal function approximations to map nonlinearity, without making prior assumptions about data properties (Haykin,1999).

Discrete wavelet decomposition is applied to divide the original financial time series into multi level approximate and detailed coefficients so that the neural network can then be applied to the denoised data and improve prediction accuracy. The main idea of wavelet decomposition is to transform the original data into a different wavelet basis, where the large coefficients mainly describe useful information, such as trends, and the small coefficients represent noise and variations. By modifying the coefficients in the new basis and suitable threshold reconstructing back to the original data domain, noise can be removed and different data characteristics can be achieved. Wavelet decomposition and reconstruction methodology developed by Donoho and Johnstone (1998) for estimation. It has been implemented in many applications, e.g. wavelet toolbox in Matlab (Gao,1997).

The idea of combining the neural networks with multiscale wavelet discrete decomposition has been proposed by previous research (Bai, 2001, S.J.Yao, 2000, A.M.Azoff, 1994).The properties of wavelet transforms emerging from the multiscale decomposition of time series allow the study of both stationary and non-stationary time series. Moreover, the simple wavelet neural network requires shorter computing time to

process the data as compared to the multi layer feedforward neural network (T.Yamakawa, 1994). Hence combining wavelet and neural network is a powerful data analysis method to solve nonstaionary time series forecasting. The proposed neural wavelet model for time series prediction is shown in Figure 7.1. Given the limited time series $g(t)$, the aim is to predict one sample forward, $g(t+1)$. The first oval in the figure represents the wavelet processing. The circle 'NN' stands for neural networks. This hybrid scheme includes three stages. In the first stage, the time series were decomposed with a wavelet function into three sets of coefficients. After the threshold reconstruction of the decomposition coefficients, three new time series were reconstructed. In the second stage, each of these three new time series were predicted by a separate NN. The prediction results were used as inputs for the third stage, where the next sample of $g(t)$ was derived by NN4.

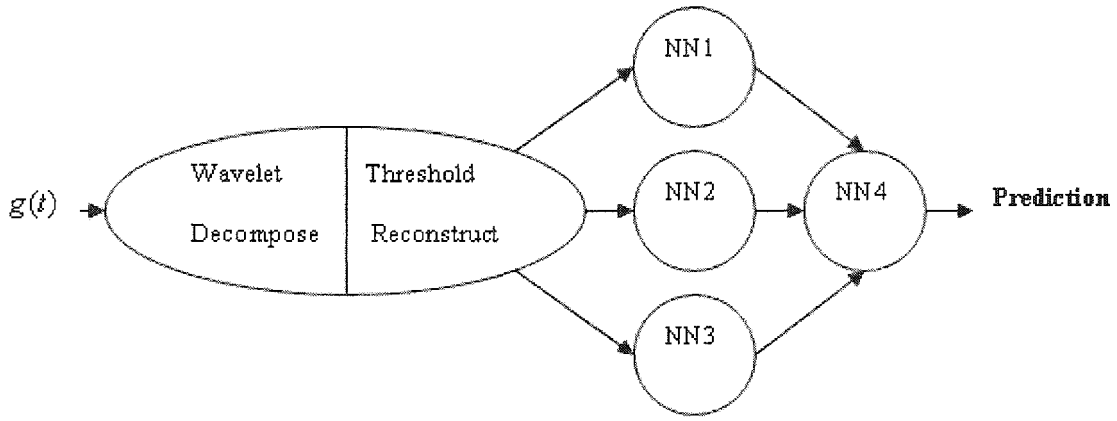


Figure 7.1 *Overview of the neural-wavelet forecasting model*

7.1 Data Preprocessing

To make the neural networks more efficient, some preprocessing needs to be done on the inputs and targets of the neural network. In the research, the target was the sales data g(t), and the inputs were the wavelet decomposition of g(t). The sales g(t) needed to be normalized and fall into the range between minus one to plus one. The algorithm used in this conversion is below.

$$s = 2 \times \frac{s - \min(g(t))}{\max(g(t)) - \min(g(t))} - 1, \; s \in (-1,1); \qquad (7.1)$$

where s is the data element in $g(t)$, min(g(t)) is the minimum data in g(t) and max(g(t)) is the maximum data in g(t). It is important for the preprocessing to avoid losing any information built in the old time series. After preprocessing, the new time series keeps the same shape as before (Figure 7.2). The detail characteristics such as trend and oscillation in the old series have been preserved.

The outputs of neural networks have the same scale as the input, which is the preprocessed g(t). Therefore, post processing is required to convert the output forecast back into the scale of sale series.

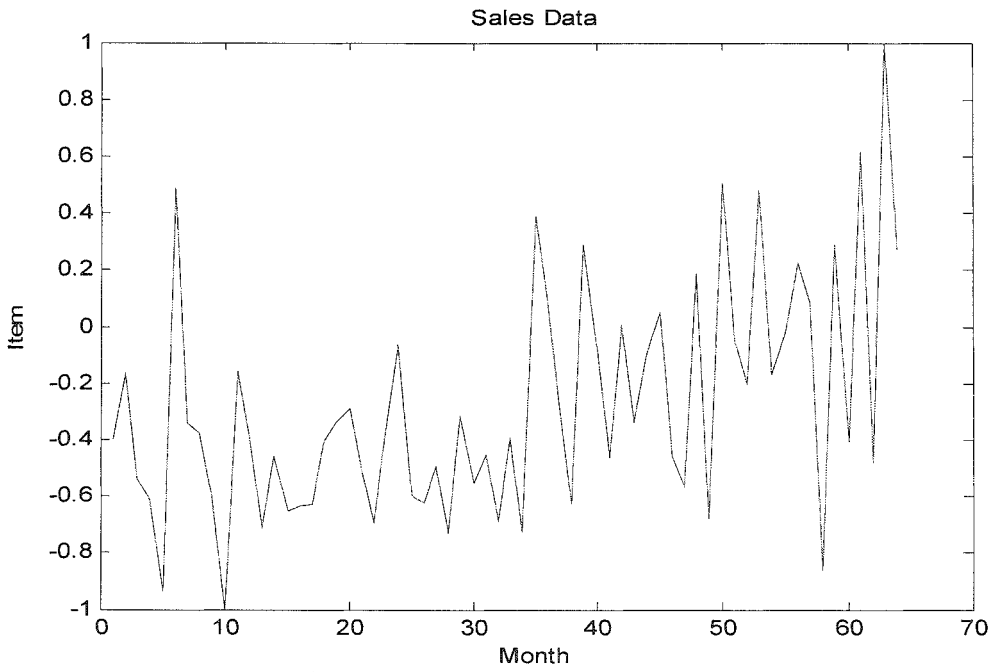


Figure 7.2 *Preprocessed sales data*

7.2 Wavelet Analysis

The preprocessed original series need to be transformed to remove the noise and reduce the complexity. There are many ways to decompose or transform the data. Wavelet transform and Fourier transform are two of the most famous. The Fourier transform is very useful, as the frequency content can be efficiently expressed. However, the drawback is that it has only frequency resolution and no time resolution. The time series samples contain numerous nonstationary or transitory characteristics: drift,

trends, abrupt changes, and beginnings and ends of events. Fourier analysis is not suited to detecting them (C.Sidney, 1998).

Alternatively, wavelet analysis is a kind of time-frequency joint representation. It can provide the frequency characteristics and time information at the same time. By decomposing the original data into many levels with wavelet function, different levels of approximation and variation detail can be extracted from the original data. Wavelets provide the way to estimate the underlying function of the data.

7.2.1 Discrete Wavelet Transform

Discrete wavelet transform (DWT) was chosen to perform the decomposition. For a set of time series data $g(t) \in L^2(R)$ [3], it can be expanded as below:

$$g(t) = \sum_{k=-\infty}^{\infty} c(k)\varphi_k(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \qquad (7.2)$$

In this expansion, the first summation gives the coarse approximation of $g(t)$. For the second summation, as the increasing of index j, a higher resolution function is added. In equation (7.2) $\varphi_k(t)$ is the scaling function, $\psi_{j,k}(t)$ is the wavelet function. The scaling function is defined in terms of integral translates of the basic scaling function by

$$\varphi_k(t) = \varphi(t-k), \qquad k \in Z \qquad (7.3)$$

A two dimensional family of functions is generated by scaling and translating the basic scaling function (equation (7.4)).

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k), \qquad (7.4)$$

[3] This is the space of all functions with a well-defined integral of the square of the modulus of the function. 'L' signifies a Lebesque integral, the '2'denotes the integral of the square of the modulus of the function, 'R' states that the independent variable of integration t is a number over the whole real line.

By the definition, the scaling function can be expressed in terms of a weighted sum of shifted $\varphi_k(2t)$ as:

$$\varphi_k(t) = \sum_n h(n)\sqrt{2}\varphi(2t-n), \quad n \in Z; \tag{7.5}$$

while the wavelet function can be represented by a weighted sum of shifted scaling function $\varphi_k(2t)$

$$\psi(t) = \sum_n h_1(n)\sqrt{2}\varphi(2t-n), \quad n \in Z; \tag{7.6}$$

the function generated by (7.6) gives the prototype wavelet $\psi(t)$ for a class of expansion functions:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), \tag{7.7}$$

where 2^j is the scaling of t, $2^{-j}k$ is the translation in t, and $2^{j/2}$ maintains the norm of the wavelet at different scales. The wavelet coefficient $h_1(n)$ is required by orthogonality to be related to the scaling function coefficients $h(n)$ by

$$h_1(n) = (-1)^n h(1-n). \tag{7.8}$$

Once given the prototype or mother wavelet $\psi(t)$, the scaling function and wavelet functions can be generated and summed to form the original data. The 'db2' wavelet has been applied as mother wavelet (Figure 7.3), which is the Daubechies family wavelet. '2' is the order of the wavelet (Raghuveer, 1998). These wavelets have no explicit expression except 'db1', which is Haar wavelet.

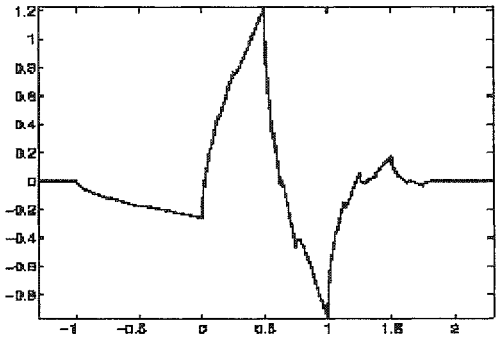


Figure 7.3 *'db2' prototype wavelet*

The coefficients $c(k)$ and $d(j,k)$ in the wavelet expansion equation (7.2) are called the discrete wavelet transform (DWT) of data $g(t)$. These wavelet coefficients can be used as description, approximation, and filtering of the original series. $c(k)$ and $d(j,k)$ can be calculated by the inner products of original data $g(t)$ and wavelet scaling function $\varphi_{j,k}(t)$ (equation (7.9) & equation (7.10))

$$c_j(k) = \langle g(t), \varphi_{j,k}(t) \rangle = \int g(t) \varphi_{j,k}(t) dt \tag{7.9}$$

and

$$d_j(k) = \langle g(t), \psi_{j,k}(t) \rangle = \int g(t) \psi_{j,k}(t) dt \tag{7.10}$$

7.2.2 Wavelet Decomposition

Based on the recursion equation from (7.5), the time variable was scaled and translated to give

$$\varphi(2^j t - k) = \sum_n h(n) \sqrt{2} \varphi(2^{j+1} t - 2k - n) \tag{7.11}$$

After changing $m = 2k + n$, this becomes

$$\varphi(2^j t - k) = \sum_m h(m - 2k) \sqrt{2} \varphi(2^{j+1} t - m) \tag{7.12}$$

Now, by considering the discrete wavelet transform of the $g(t)$, which is equations (7.9) and (7.10),

$$c_j(k) = \langle g(t), \varphi_{j,k}(t) \rangle = \int g(t) 2^{j/2} \varphi_{j,k}(t) dt \qquad (7.13)$$

by using (7.12) and interchanging the sum and integral, this can be rewritten as

$$c_j(k) = \sum_m h(m-2k) \int g(t) 2^{(j+1)/2} \varphi(2^{j+1}t - m) dt \qquad (7.14)$$

In this formula, the integral is the inner product with the scaling function at a scale of $j+1$, giving

$$c_j(k) = \sum_m h(m-2k) c_{j+1}(m) \qquad (7.15)$$

The corresponding relationship for the wavelet coefficient is

$$d_j(k) = \sum_m h_1(m-2k) c_{j+1}(m) \qquad (7.16)$$

Equation (7.15) and (7.16) show that the scaling and wavelet coefficients at different levels of scale ($c_j(k)$ and $d_j(k)$) can be obtained by convolving the expansion coefficient at scale $j+1$ ($c_{j+1}(m)$), the coefficient $c_{j+1}(m)$ time $h(-n)$, $h_1(-n)$ and down sampling to give the expansion coefficients at the next level of j. In other words, as Figure 7.4makes clear, $c_{j+1}(k)$ are filtered by two filters with coefficient $h(-n)$ and $h_1(-n)$, after which, down sampling gives the next coarser scaling coefficient c_j and wavelet coefficient d_j.

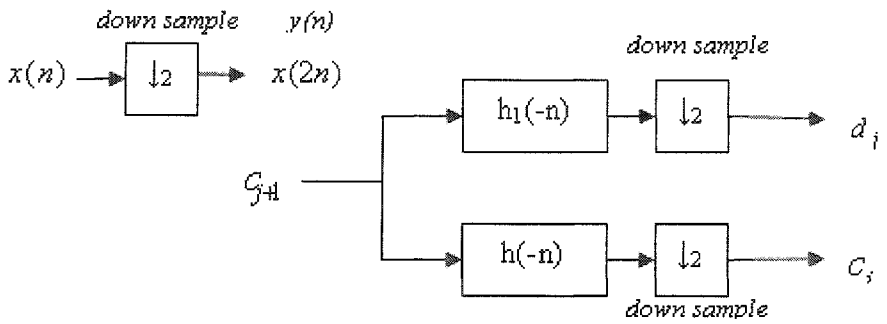


Figure 7.4 *Discrete wavelet decomposition*

In the top right of Figure 7.4, the down sampler takes $x(n)$ as input and produces output y(n) as $x(2n)$. In our case, the output y(n) is the odd index term $x(2n+1)$. This is defined by the filtering process. When convolving the input coefficients with filters, the convolution 'smears' the inputs and introduce some extra samples into the results (Jeruchim, 2000).

The filter implemented by $h(-n)$ is a low-pass filter (i.e. low frequency passed) and the one implemented by $h_1(-n)$ is a high-pass filters (i.e. high frequency passed). Those high pass filter coefficients, which are wavelet coefficients, consist mainly of high frequency noise and represent a detailed variation of the original. These low pass filter coefficients, which are scaling coefficients, are called approximation coefficients. They are the identity of the data, and express the overall shape and trend of the original. They contain less noise information, compared with the original data. In Figure 7.5, 'db2' wavelet is used as the mother wavelet to decompose the data *s* into two levels. Two detail coefficients d_1 and d_2 and one approximation coefficient a_2 are produced.

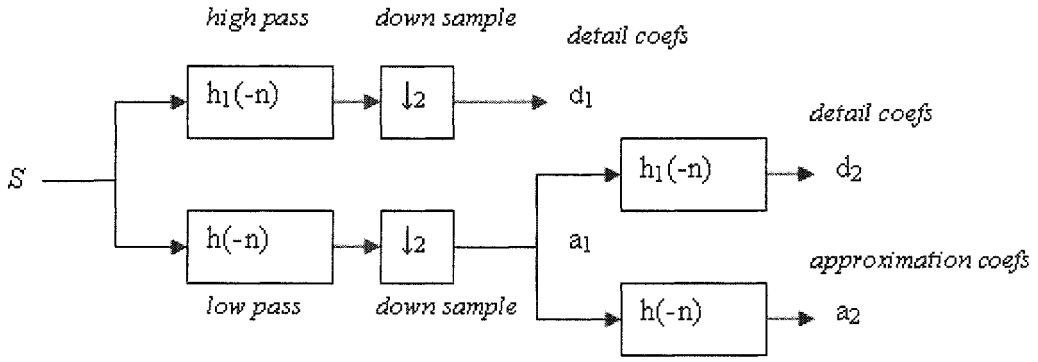


Figure 7.5 *Two stages two bands decomposition*

As shown in the schematic figure above, in the first stage, the two filters divide the original data into a low pass and high pass bands, and produce detail coefficients d_1 and approximation coefficients a_1. In the second stage, the approximation coefficients a_1 is decomposed again to generate the second detail coefficient d_2 and second level approximation coefficient a_2. The first stage divides the original data into two equal parts; the second stage divides the lower parts into quarters. In the program, the original data'*s*' comprises 64 sets. After first stage decomposition, the detail coefficient d_1 consists of 33 sets. In the second level approximation a_2 has 18 sets, the same as detail d_2. Figure 7.6 shows the original 's'.

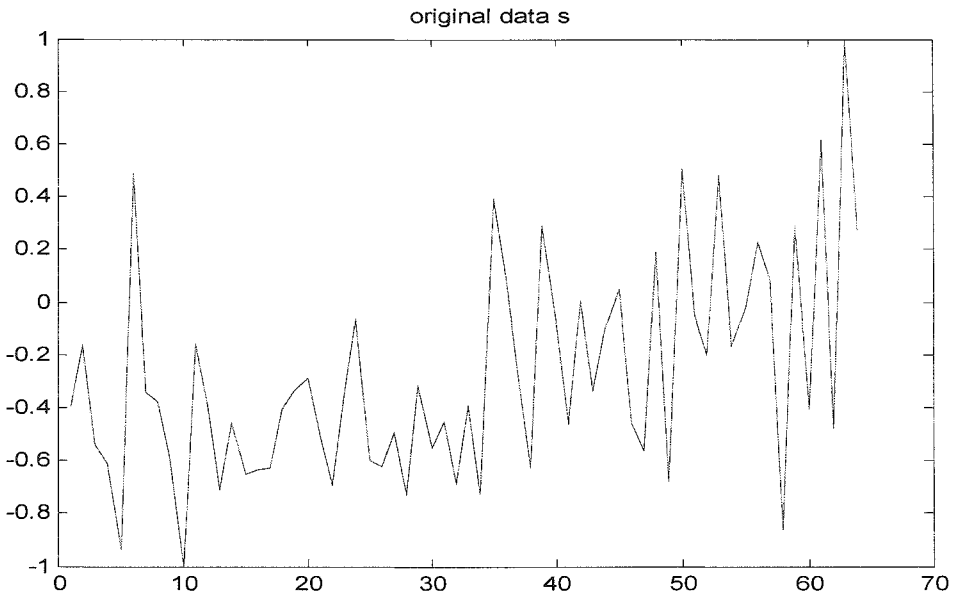


Figure 7.6 *Preprocessed original data S*

The results of decomposition a_2, d_2, d_1 are plotted on Figure 7.7. The overall trend of the series can be observed from a_2, which is the 'identity' of s. However, by observing d_2 and d_1, it is still hard to clarify the detail of the variation or frequency of s. On the other hand, the quantities of the samples for each of the coefficients are so small (a_2 has only 18 sets, d_2 has 18 sets and d_1 has 35 sets). It is hard to make a forecast based on this short history data. To solve this problem, the wavelet reconstruction was implemented and is introduced in the next section.

Decomposition at Level 2 $g(t) = a_2 + d_2 + d_1$

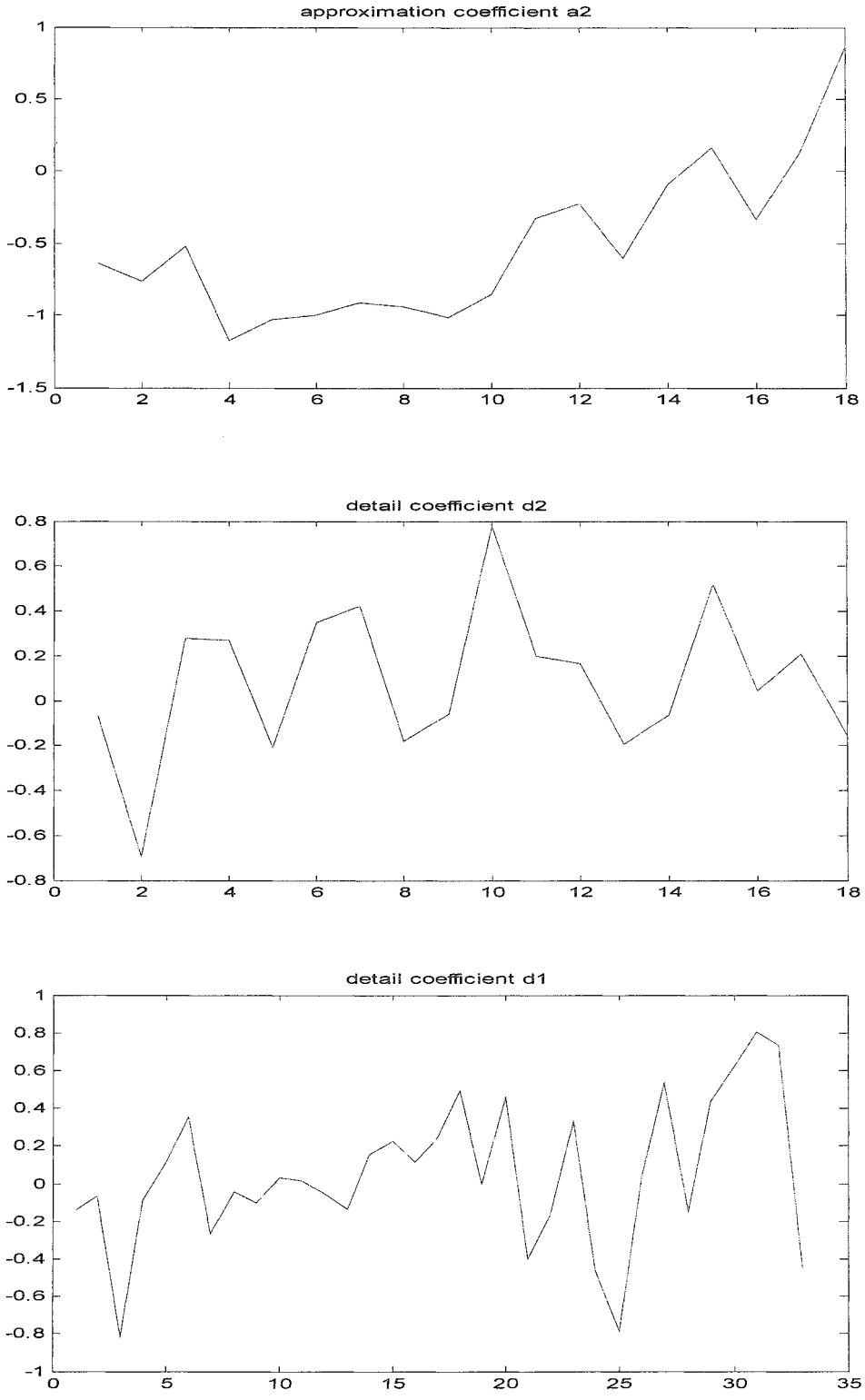


Figure 7.7 *Decomposition of original data*

7.2.3 Wavelet Reconstruction

Wavelet decomposition breaks input series data into different levels of detail and approximation coefficients. It is then important to assemble these components into a new series, which is suitable for forecasting. This process is called wavelet reconstruction. Since it is the postprocess of decomposition, the reconstruction can be made from a combination of the scaling function and wavelet coefficients. Considering a set of data $g(t)$ in the $j+1$ scaling function, this can be written in terms of the scaling function as:

$$g(t) = \sum_k c_{j+1}(k) 2^{(j+1)/2} \varphi(2^{j+1} t - k) \tag{7.17}$$

and

$$c_{j+1}(k) = \sum_m c_j(m) h(k-2m) + \sum_m d_j(m) h_1(k-2m) \tag{7.18}$$

where c_j is the scaling or approximation coefficient and d_j is the wavelet or details coefficient. This combining process can be carried out at any level as long as there are appropriate scale and wavelet coefficients. In Figure 7.7, the inputs a_2, d_2 and d_1 are the coefficient generated from wavelet decomposition in the previous section. A sequence of first up sampling followed by the filtering process was applied in the reconstruction. As shown in the top left of Figure 7.8, the up sampling means that the input to the filter has zeros inserted between each of the original terms. The output $y(2n)$ is $x(n)$ and $y(2n+1)$ is zero, where the input data is stretched to twice its original length and zeros are inserted.

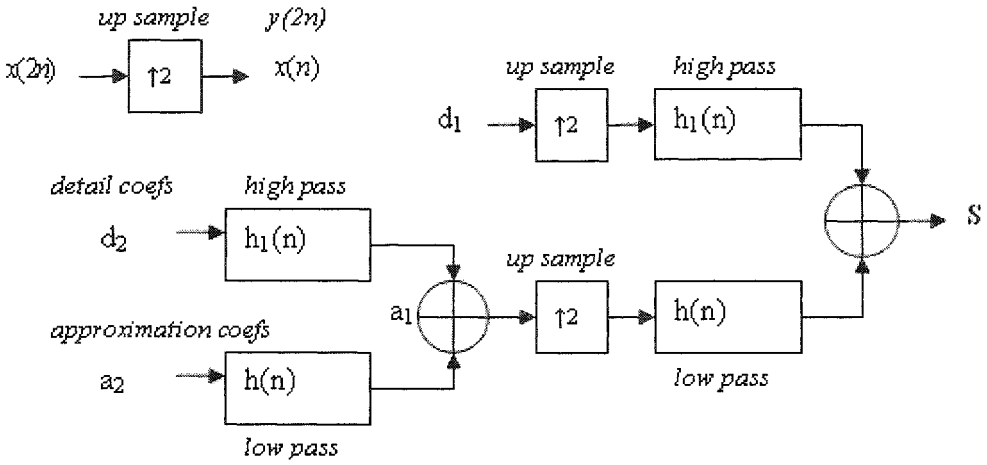


Figure 7.8 *Two stages two band reconstruction*

By up sampling and combining different coefficients, the output *s* will be the same as the original data. However, it does not make any sense to decompose the original dataset into a_2, d_2, d_1 and then construct back to itself again. Alternatively, the inputs series are simply 'denoise' or threshold reconstructed. A two stage reconstruction is carried out on a_2 and zero coefficients for d_1, d_2. The reconstruction result is shown below.

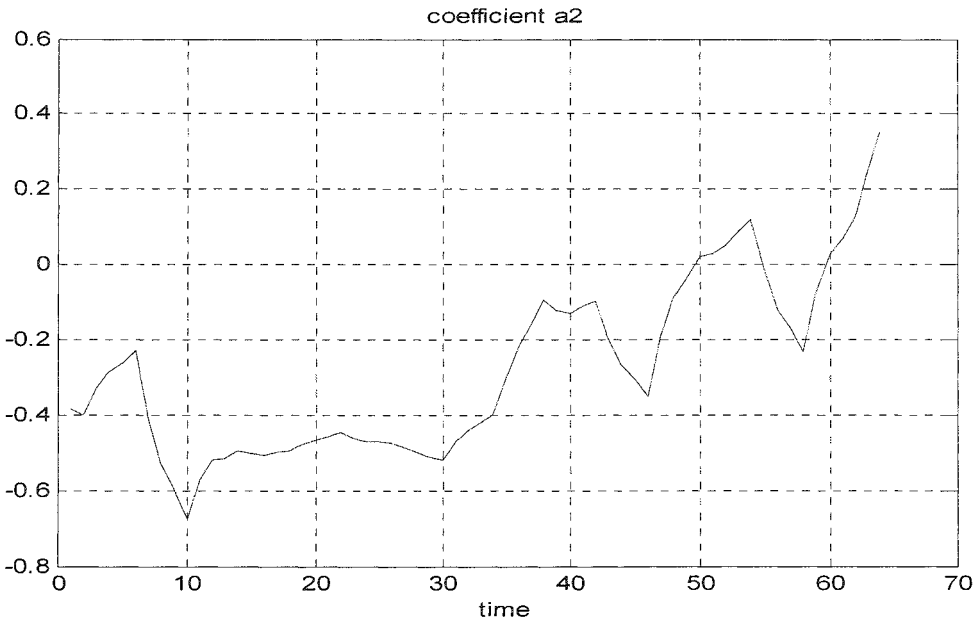


Figure 7.9 *Threshold reconstruction with* a_2

The overall pattern in Figure 7.9 is similar to the pattern of the approximation coefficient a_2 in Figure 7.7. However, the size of the data is still 64, which is the same as the original. Reconstruction increases the size of the data set and makes forecasting more feasible. The threshold reconstruction based on d_1 and d_2 is shown in Figure 7.10 and Figure 7.11. The oscillations of the reconstructions are more frequent than the coefficients oscillations in Figure 7.7. Thus, it is easier for neural networks to pick up the oscillation characteristics.

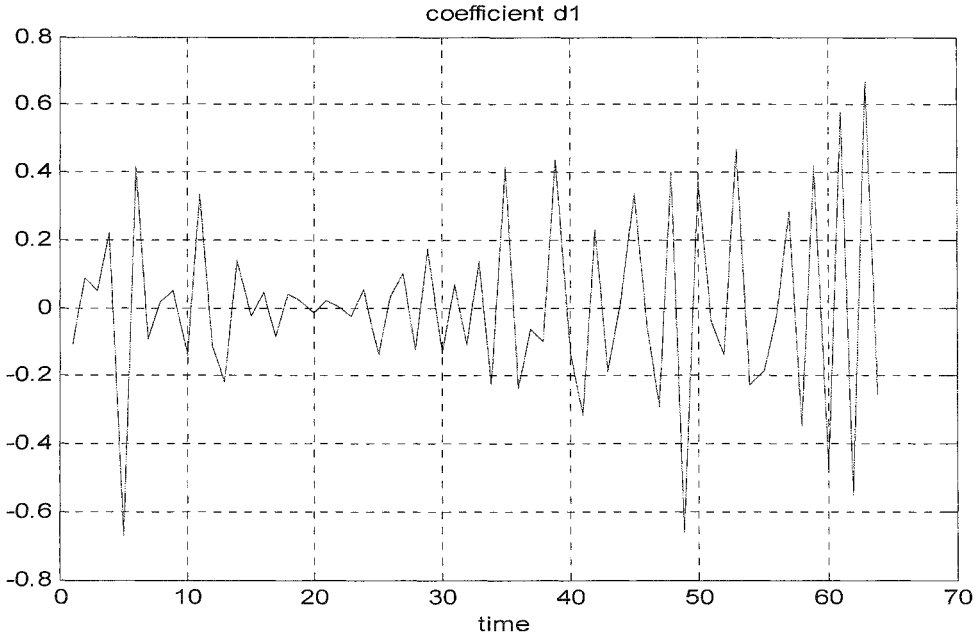


Figure 7.10 *Threshold reconstruction with d_1*

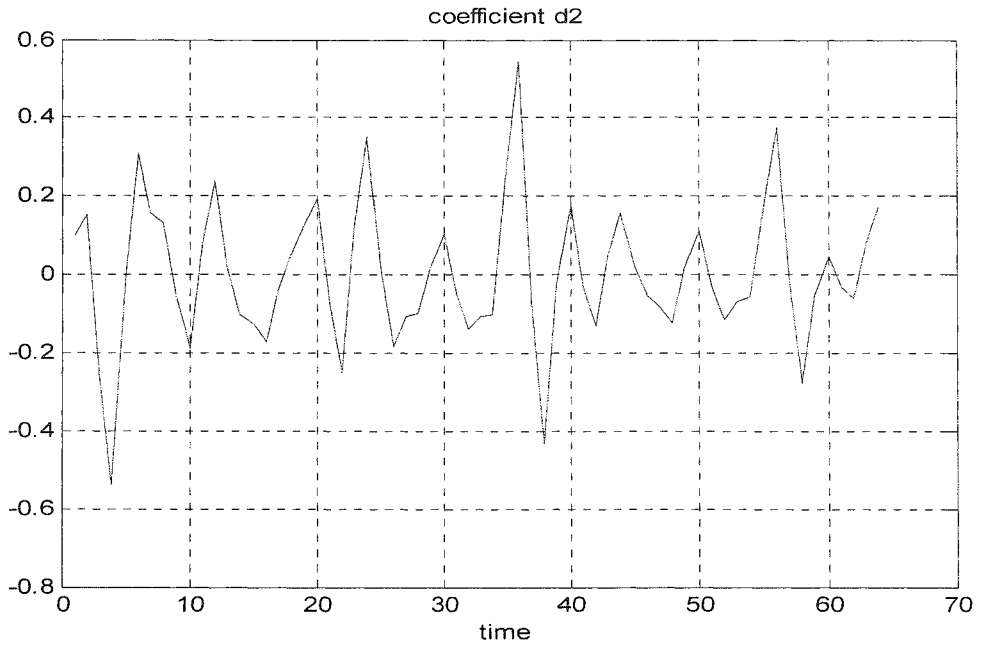


Figure 7.11 *Threshold reconstruction with d_2*

Some researchers (Bai, 2001, S.J. Yao, 2000, Xie, 2006) have used decomposition coefficients directly to make forecasts. However, this is only suitable for large size data sets, which can generate numerous wavelets coefficients. In our case, the size of the series was not large enough.

7.3 Neural Networks

A neural network is an interconnected assembly of simple processing elements, units or neurons, whose functionality is loosely based on the animal neuron (Robert, 1999). The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns. The network function is largely determined by the connection between their elements. The neural network can be trained or the weights of the connection between each element can be adjusted to perform a particular function. In Figure 7.12, the neural networks are trained to generate particular outputs based on inputs and targets.

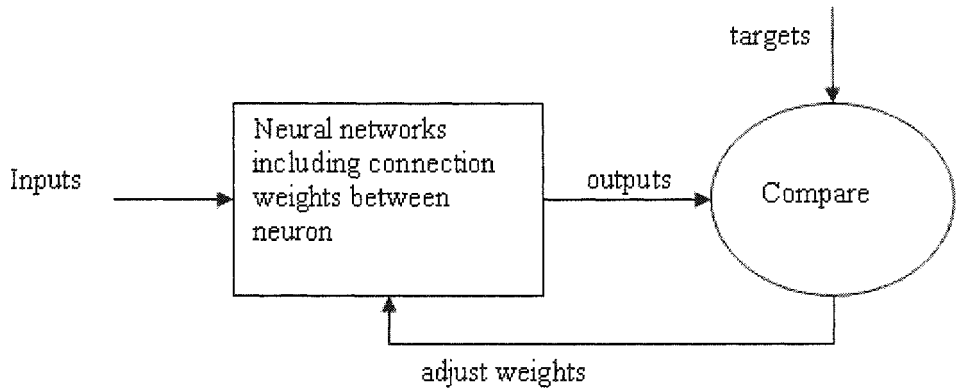


Figure 7.12 *Neural networks training*

The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. There are generally four steps in applying and training neural networks:

- Assemble the training and testing data;
- Create the network object;
- Train the network;
- Simulate the network response to the testing inputs.

In the first step, the training and testing data is assembled as the wavelet processed sales series. In the next step, the network object or the architecture of the network needs to be designed. After building up the network, an efficient training algorithm is chosen to train the network and update the network weights. If the training result is not acceptable, the network needs to be redesigned and trained again. Finally, the trained neural network is tested by the testing data which is not used in training. This process is also called network simulation. After examining the testing results, it is possible to decide whether to use this network or redesign it again.

7.3.1 Feed Forward Networks

In designing network architecture, three basic rules are considered, the pattern of connectivity which is the structure and levels of the network, the rule for combining inputs, and the rule for calculating the outputs.

The pattern of connectivity refers to the way in which the units are connected. The neural network used in the research is a feed forward network. The feed forward

network structure is illustrated in Figure 7.13. The inputs series is always fed forward and no information feedback is presented.

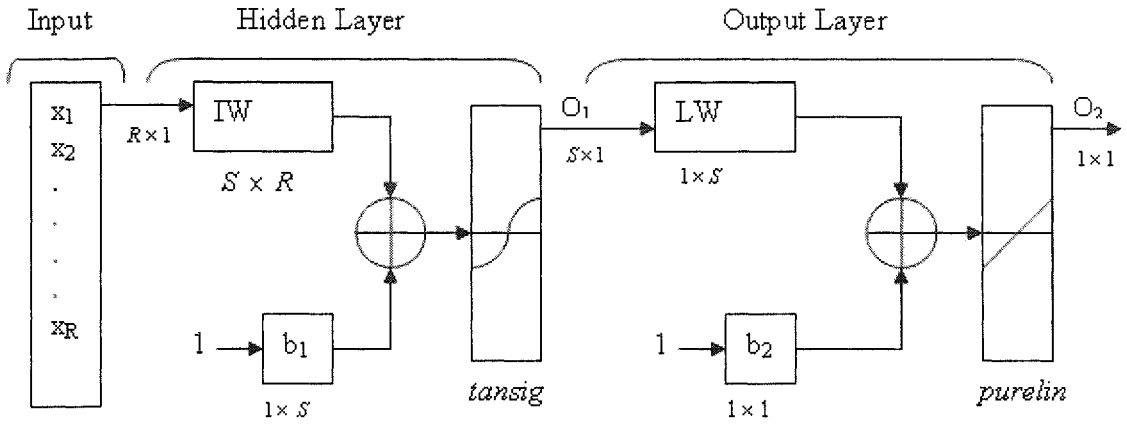


Figure 7.13 *Feedforward neural networks*

There are two layers in our feed forward neural networks: hidden layer and output layer. The series $x_{1...}$ x_R are the inputs to the hidden layer. Here 'R' is the size of the input; 'IW' represents the input neurons and their weights in the hidden layer; 'S' is the amount of the neurons; 'b_1' determines the bias weights. The nonlinear transfer function '*tansig*'(Figure 7.15) in the hidden layer allows the networks to learn both the linear and nonlinear characteristics of the inputs. 'O_1' is the output of the hidden layer and input for the output layer. 'LW' presents the neurons and their weights in the output layer; 'b_2' is the bias weight; '*purelin*' (Figure 7.16) calculates the neuron outputs 'O_2'. In the program, 'O_2' is a single value which indicates the one step forward forecast.

These two layers have different amounts of neurons, related weights and rules for calculating outputs. During this research, the quantity of the neurons in the output layer was kept as one and the numbers of neurons in the hidden layer were changed to improve the forecasting performance for different inputs. The details of hidden layer are shown in Figure 7.14. Figure 7.14 (a) is the expansion of Figure 7.14 (b).

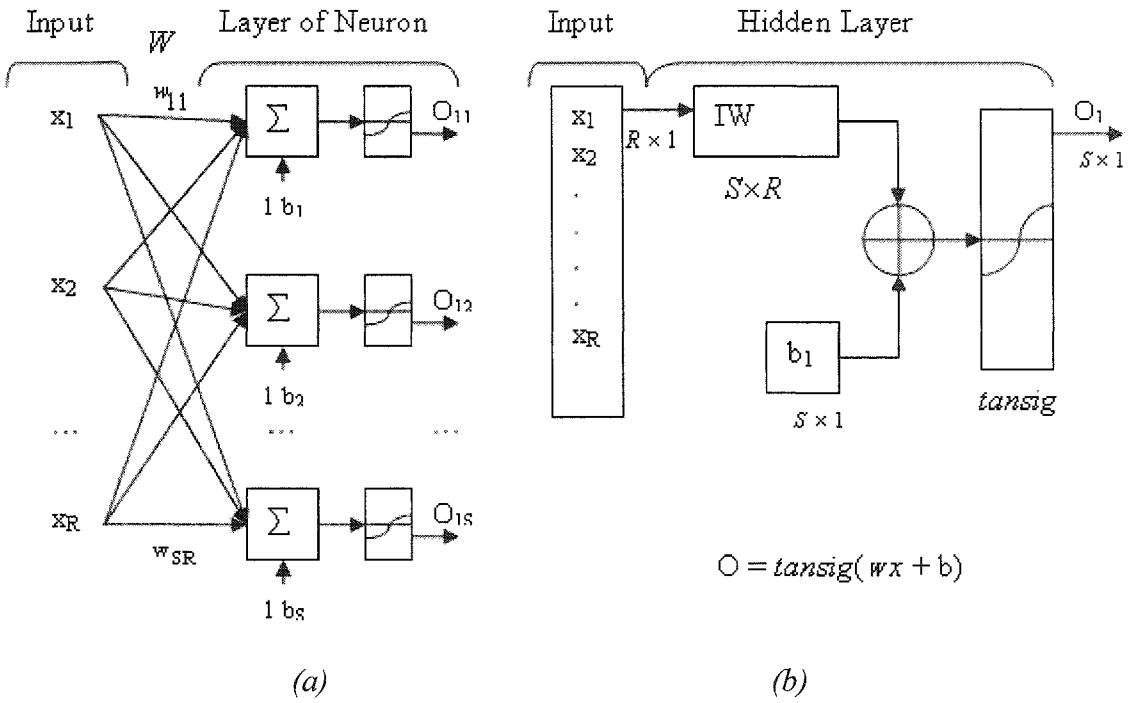


Figure 7.14 *Expansion of the layer diagram*

In the input layer, each input vector 'x' is connected to each hidden layer neuron through the weight matrix 'W'. The pattern of the connectivity is described in the weight matrix 'W'. The weight 'w_{SR}' is specified by three parameters: the first subscript 'S' shows the neuron S where the weight connects to; second subscript 'R' shows the input R where the weight connects from; and the value of the weight 'w_{SR}'. A negative weight value will inhibit the activity of the neurons connected to, while a positive weight value will serve to excite the neurons connected to. The absolute magnitude of the weight specifies the strength of the connection. The weight matrix is the network's memory that stores the knowledge about how to solve a problem.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1R} \\ w_{21} & w_{22} & \cdots & w_{2R} \\ w_{S1} & w_{S2} & \cdots & w_{SR} \end{bmatrix}$$

The way of combining the inputs is summation, as shown in Figure 7.14.The inputs are combined by summing their weight values and bias.

Two functions are applied as the rules for calculating output value in the neurons: '*tansig*' in the hidden layer and '*purelin*' in the output layer. These are known as

activation functions. The '*tansig*' function is a tangent sigmoid transfer function (Vogl, 1988). It maps the input to the interval $(-1,1)$ with the algorithm below:

$$a = \frac{2}{1+e^{-2n}} - 1 \tag{7.19}$$

'n' is the input to the function and 'a' is the result.

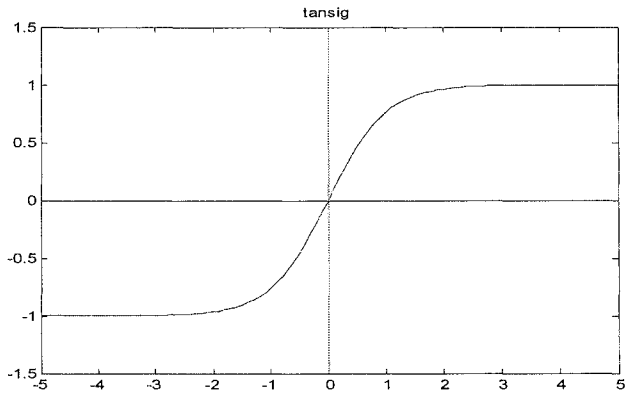


Figure 7.15 *Schematic graph of 'tansig'*

The '*purelin*' function is the linear transfer function that calculates the neuron output by simply returning the value passing to it.

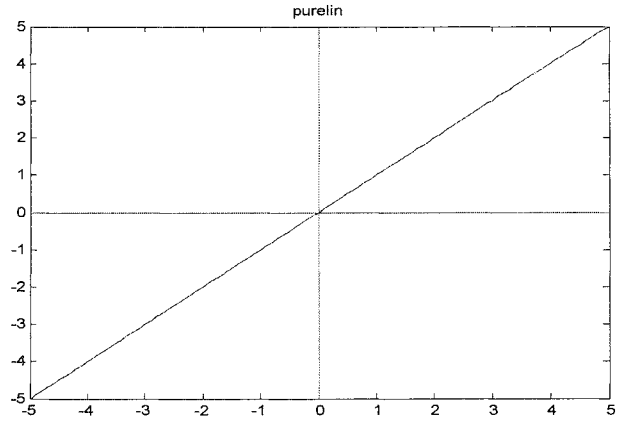


Figure 7.16 *Schematic graph of 'purelin'*

7.3.2 Weights Adapting Rule

Widrow-Hoff Law or Delta Rule

Once the network structure is constructed, the neural network needs to be trained to adapt the weights. At the start of the training, the initial weights are set at a small random number. When the inputs are presented for the first time, it is unlikely that the network could produce the correct output. The discrepancy between what the network actually outputs and what it is required to output constitutes an error. This error can be used to adapt the weights. One of the error correcting rules for adapting weights is the Widrow-Hoff law or delta rule (Kevin, 2003). Consider Figure 7.17, the output neuron has an activation output of o_j and a target output t_j.

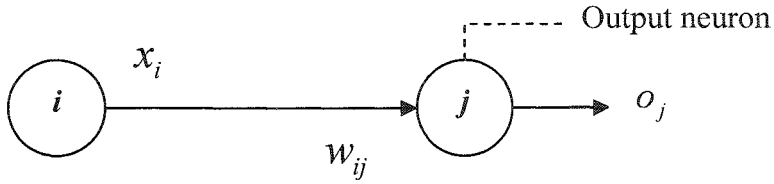


Figure 7.17 *Single weight connecting two neurons: the signal* x_i *is multiplied by the weight* w_j.

The error, δ_j, is given by

$$\delta_j = t_j - o_j \tag{7.20}$$

$$o_j = \sum_j x_i w_{ij} \tag{7.21}$$

X_i is the input to the output neuron j. The delta rule states that the adjustment to be made Δw_{ij} connecting neuron i and j, is

$$\Delta w_{ij} = \eta \delta_j x_i \tag{7.22}$$

where η is a real number that stands for the learning rate (by how much to adapt the weight).

The new weight is the sum of the adjustment and the old weight

$$w_{ij} = \Delta w_{ij} + w_{ij} \tag{7.23}$$

During the training, the weights and biases of the network are iteratively adjusted to optimize the network performance function. This performance function in our feedforward network was a mean square error; the average squared error between the networks outputs o_j and the target outputs t_j. The overall squared error can be defined as:

$$E = \frac{1}{2}\sum_j (t_j - o_j)^2 \tag{7.24}$$

The chain rule can be used to express the derivative of the error with respect to a weight as a product that reflects how the error changes with a neuron's output and how the output changes with an impinging weight.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} \tag{7.25}$$

From (7.24) and definition of o_j in (7.21),

$$\frac{\partial E}{\partial o_j} = -\delta_j \tag{7.26}$$

$$\frac{\partial o_j}{\partial w_{ij}} = x_i \tag{7.27}$$

Substituting (7.26) and (7.27) back into (7.25) obtained

$$\frac{\partial E}{\partial w_{ij}} = -\delta_j x_i \tag{7.28}$$

Considering the equation (7.28), to minimize the error the weights need to change in a direction that is opposite to the gradient vector.

Backpropagation

The weight adaptation rule in feedforward networks is backpropagation. This was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks. Backpropagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights move along the negative of the gradient of the performance function.

Backpropagation defines two sweeps of the network. The forward sweep of the feedforward network is from input layer to output layer. The backward sweep is similar to the forward sweep, except the error values are propagated back through the network to determine how the weights are to be changed during training. During the backward sweep, data pass along the weighted connections in the reverse direction. The neurons will receive error signals from every neuron in the output layer. This double sweep process is illustrated in Figure 7.18.

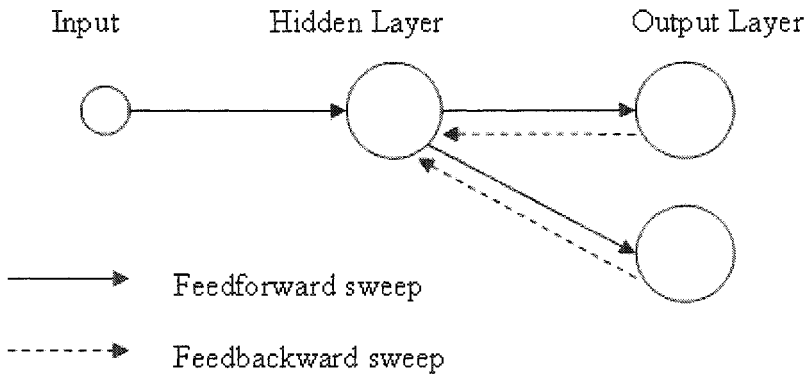


Figure 7.18 *Feedforward and feedbackward sweep*

The hidden neuron sends activation to the output neuron (solid line). In the backward sweep this hidden neuron receive error signal from the output neuron (dash line).

During the training process, each input pattern has an associated target pattern, once all outputs fall within a specified tolerance with respect to their target values, the training is at an end. This tolerance value can be changed for different circumstances. Backpropagation uses a generalization of the delta rule. The simplest implementation

of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. One iteration of this algorithm can be written below as

$$\Delta w_{ij}(n+1) = \Delta w_{ij}(n) + \eta g(n) \qquad (7.29)$$

where $\Delta w_{ij}(n)$ is the current weight change or bias change, η is the learning rate, $g(n)$ is the current gradient. The procedure for backpropagation training is as follows:

for each input vector associate a target output vector

while not STOP

STOP =TRUE

for each input vector

perform a forward sweep to find the actual output

obtain an error vector by comparing the actual and target output

if the actual output is not within tolerance set STOP = FALSE

perform a backward sweep of the error vector

use the backward sweep and training algorithm to determine gradient and weight changes

update weights

When an input is presented to the network, the performance function which is the mean square error is calculated. The gradient vector is computed by the training algorithm. Then the neural networks determine how the weights should change. The process is then repeated for each input pattern. An epoch is a complete cycle through each input. The inputs are continually presented to the network, epoch after epoch, until during one epoch all actual outputs for each input are within tolerance.

In the program, the Levenberg-Marquardt algorithm was adopted to calculate the gradient of the performance function and determine how to adjust the weights to minimize performance. It is a fast algorithm to calculate the adjusting weight. The gradient $g(n)$ and adjusting weight is shown in equation (7.30) and (7.31)

$$g(n) = J^T e \tag{7.30}$$

$$\Delta w_{ij}(n+1) = \Delta w_{ij}(n) - (J^T J + \mu I)^{-1} g(n) \tag{7.31}$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through backpropagation. The performance function which is the squared error will be reduced after each iteration of the algorithm (Kevin, 1997).

7.3.3 Training and Simulation

Approximation Level Two (a2) Reconstruction Prediction

After wavelet decomposition, the original time series $g(t)$ were decomposed into three coefficients at two levels. Then the time series were reconstructed with coefficient a_2 and generated a_2 reconstruction series. The neural networks 'NN1' in Figure 7.19 are developed to predict the a_2 reconstruction series.

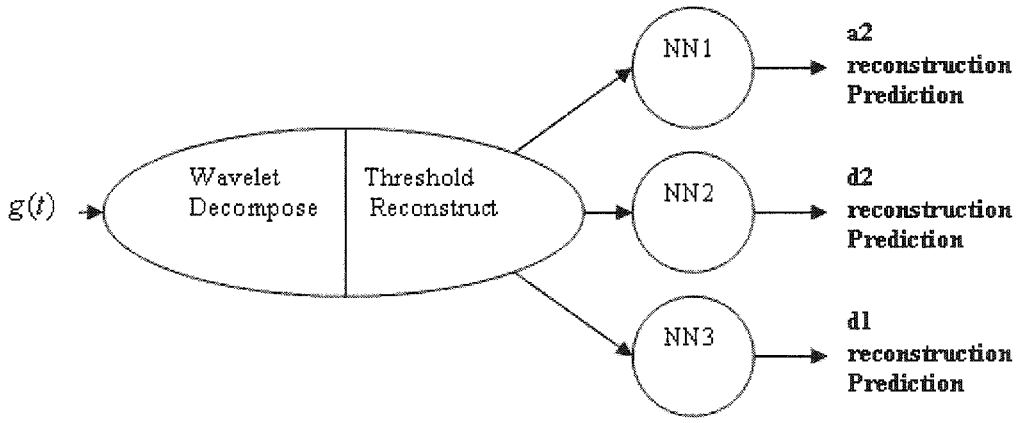


Figure 7.19 *Coefficient reconstruction prediction*

'NN1' is a two layer feedforward network, as introduced in previous sections. The structure is detailed in Figure 7.20. It has 12 input vectors and 24 hidden neurons with '*tansig*' transfer function in the hidden layer and a single output neuron with '*purelin*' transfer function in the output layer. The output is the one step ahead prediction of the a2 reconstruction series.

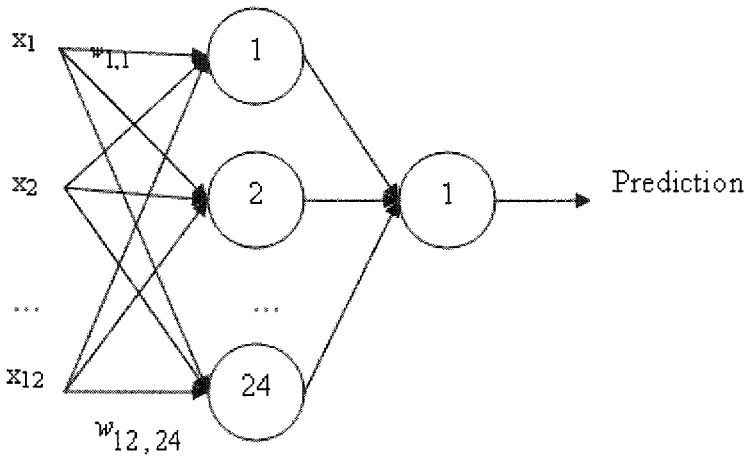


Figure 7.20 *Expansion of NN1*

'NN1' was trained with the backpropagation Levenberg-Marquardt algorithm. The whole series of a2 has 64 data sets. The first 49 data sets were used as training data and the last 15 data sets as testing data. The input matrix X and target matrix T can then be written as below

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_{38} \\ x_2 & x_3 & & x_{39} \\ \dots & & & \\ x_{12} & x_{13} & & x_{49} \end{bmatrix} \qquad T = \begin{bmatrix} x_{13} & x_{14} & \dots & x_{50} \end{bmatrix}; \tag{7.32}$$

After training, this trained network was applied to simulate the one step ahead forecast. The simulation input matrix was X_1.The forecast was $\hat{x}_{51}$ distinguished with real data x_{51}.

$$X_1 = \begin{bmatrix} x_{39} \\ x_{40} \\ \dots \\ x_{50} \end{bmatrix} \qquad forecast\hat{X} = \begin{bmatrix} \hat{x}_{51} \end{bmatrix} \tag{7.33}$$

After the prediction, x_1 and x_{13} were moved out from the input matrix X and target matrix T. The new series x_{50} and x_{51} were inserted into the inputs and targets matrix. The network was trained again. The matrices X and target T changed to be:

$$X = \begin{bmatrix} x_2 & x_3 & \cdots & x_{39} \\ x_3 & x_4 & & x_{40} \\ \cdots & & & \\ x_{13} & x_{14} & & x_{50} \end{bmatrix} \qquad T = [x_{14} \quad x_{15} \quad \cdots \quad x_{51}]; \tag{7.34}$$

Based on the updated input matrix X and target T, the forecast for x_{52} was produced. This training-simulating process was iterated until predicting the last data x_{64} in a_2 reconstruction series. The forecasting result is shown in Figure 7.21. The blue curve stands for the real data of the a_2 reconstruction series. The red curve is the forecasting result $\hat{X}$ generated by neural network NN1. As discussed in section 7.2.2, the coefficient a_2 represent the overall trend of the original data. It is very obvious that the trend is going up with regular variation from month 30 to the end. The shape of the two waves from month 30 to 45 and month 45 to 60 are very similar. The red curve follows the trend and the movement of real data (Figure 7.21).

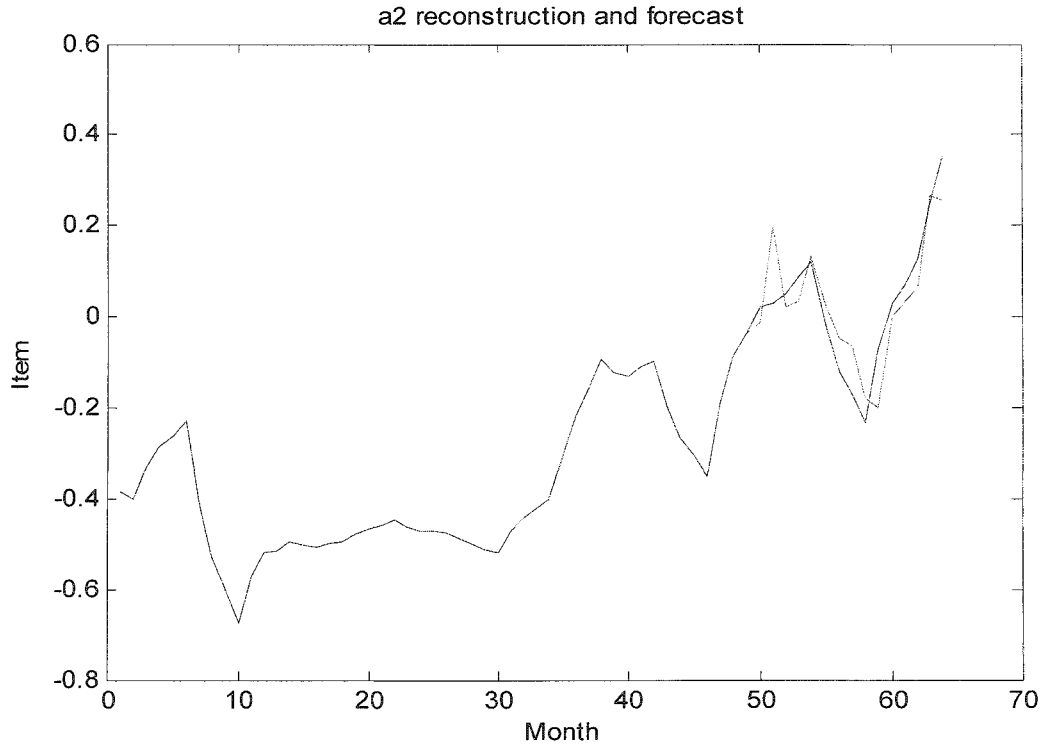


Figure 7.21 *Comparison between forecast and a_2 reconstruction series. The blue curve shows the a2 series, the red curve shows the forecast for the period 51 to 64 months.*

However the forecast cannot be entirely perfect. The neural network learns to follow and repeat the pattern of history data which could result in deviations in the prediction, such as the 'red horn' in month 51.

The neural network training and simulating program generates different results at each round of simulation owing to random initial conditions. To minimize this effect, the program was run fifty times and calculated the mean for each ten rounds. The results are given in Table 7.1 and 7.2.

Real value at month 50	1st 10 round	2nd 10 round	3rd 10 round	4th 10 round	5th 10 round
-0.026	0.031	0.03	0.021	-0.001	-0.033

Table 7.1 *Mean forecast value for each ten rounds. The difference between the means is always less than 0.06.*

Real value at month 51	1st 10 round	2nd 10 round	3rd 10 round	4th 10 round	5th 10 round
0.085	0.031	0.18	0.179	0.046	0.03

Table 7.2 *Mean forecast value for each ten rounds. The difference between the means of the second and third ten rounds and real value is less than 0.15. The difference between the means of the first, fourth and fifth rounds and real value is less than 0.05.*

Regarding the results in Table 7.1, the mean forecast value is relatively determined. The difference between the different rounds is very slight. The mean zero was chosen as the forecast. However, regarding the results in Table 7.2, the mean forecasts are clustered around two numbers: 0.04 and 0.18. The decision has to be made by considering other factors.

Detail Level Two (d2) Reconstruction Predictions

The prediction of detail coefficient at level two d_2 follows the same procedure as the prediction of a_2. While a_2 represents the trend of the data, d_2 shows oscillation details. The character of the data pattern in a_2 and d_2 is different. Therefore a different structured neural network is required. Based on a large number of training experiments,

the feedforward neural network NN2, which has 9 input vectors, 18 neurons in the hidden layer and single output neuron in the output layer, achieves a better performance. The detail structure of NN2 is explained in Figure 7.22.

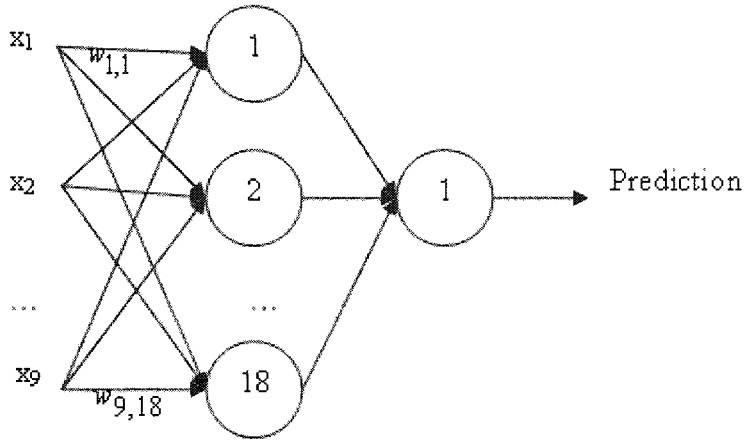


Figure 7.22 *Expansion of NN2*

The forecasting result is shown in Figure 7.23. The blue curve is real data of d_2 reconstruction series. The red curve is the forecasting result $\hat{X}$ generated by neural network NN2. The prediction approximately varies on the same pattern as the d2 reconstruction series, but there is a big discrepancy in the amplitude existing in some of the wave.

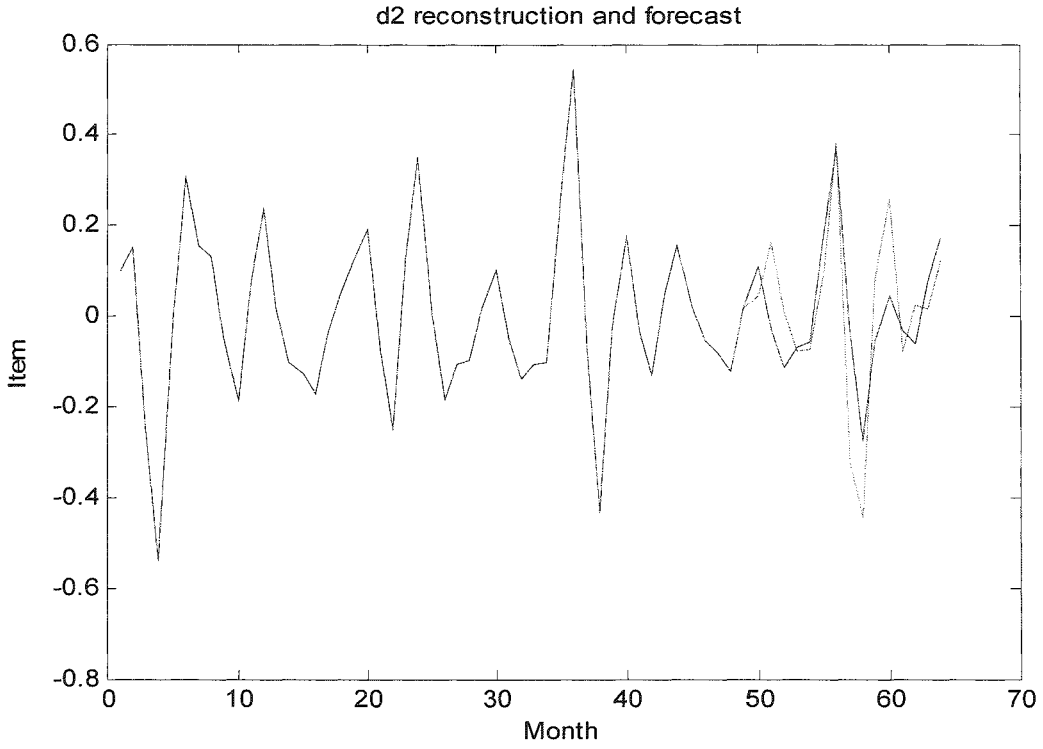


Figure 7.23 *Comparison between forecast and d_2 reconstruction series. Blue curve shows the d_2 series. Red curve shows the forecast for the period 51 to 64 months.*

Detail Level One (d_1) Reconstruction Prediction

The reconstruction series of detail coefficient d_1 represents the oscillation details in the original sales series. The neural network NN3 was applied to predict the d_1 reconstruction series.

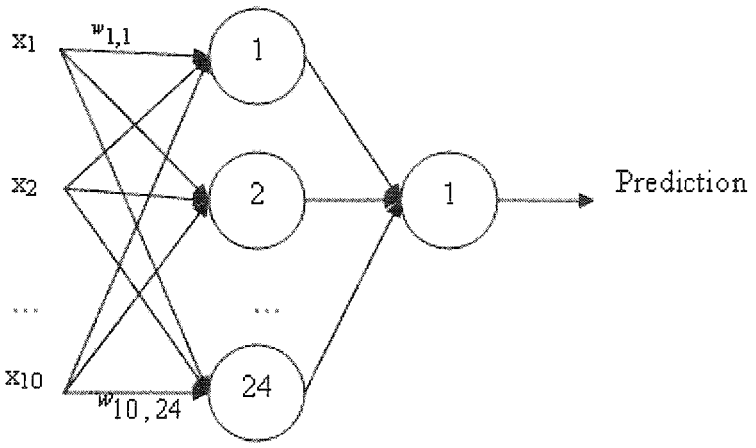


Figure 7.24 *Expansion of NN3*

In Figure 7.24, NN3 has 10 input vectors, 24 neurons in the hidden layer and one output neuron in output layer. During the testing of trained networks, experiment showed that training and simulation can achieve better performance when the size of input vector is less than half of the number of the neurons.

The forecast performance is shown in Figure 7.25. The last ten month forecast follows the pattern of the real series. Closer observation shows that the inaccurate prediction occurs between month 51 and 54. However, the real data in this period also exhibits a strange pattern, compared with neighbor data sets.

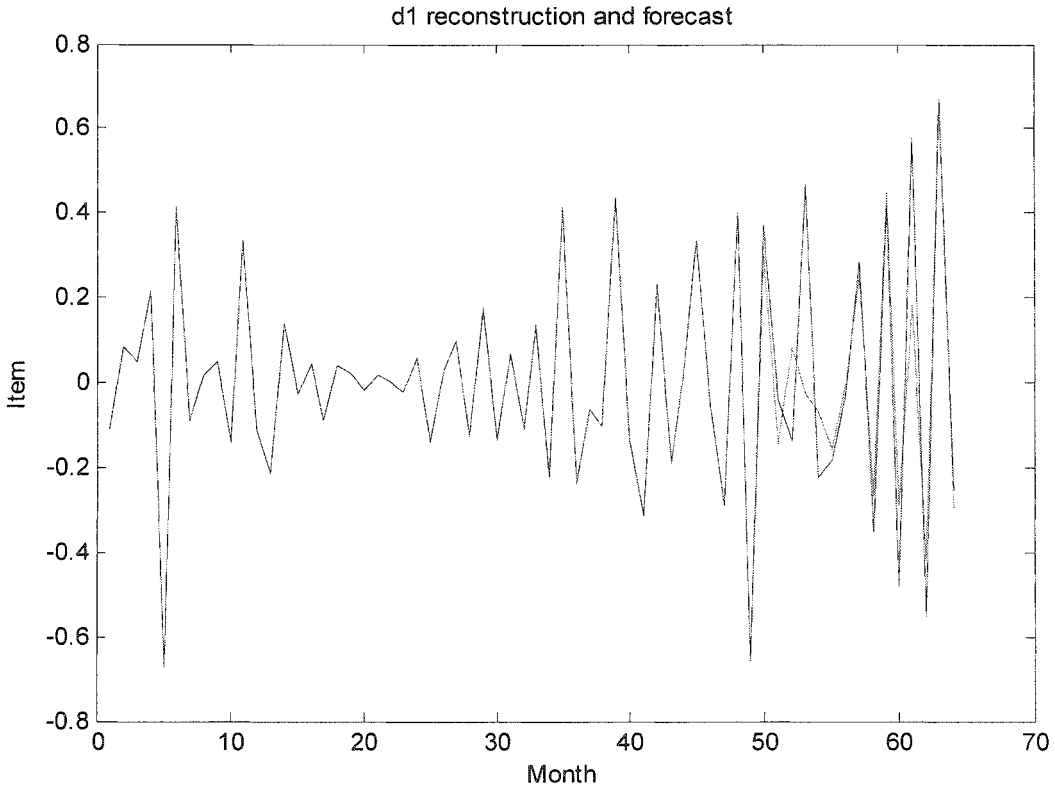


Figure 7.25 *Comparison between forecast and d_1 reconstruction series. Blue curve shows the d_1 series. Red curve shows the forecast for the period 51 to 64 months.*

Forecasting of the original series

In order to produce a final forecast, the three predictions of the reconstruction coefficients were grouped together by NN4.

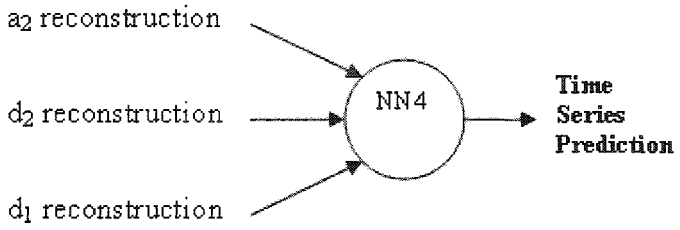


Figure 7.26 *Neural network for final forecast*

The feedforward neural network NN4 was designed to perform the task. There were only three input vectors in NN4, 12 neurons in the hidden layer and one neuron in the output layer. The output is a one step forward forecast of the original time series $g(t)$. The detail structure of NN4 is shown in Figure 7.27.

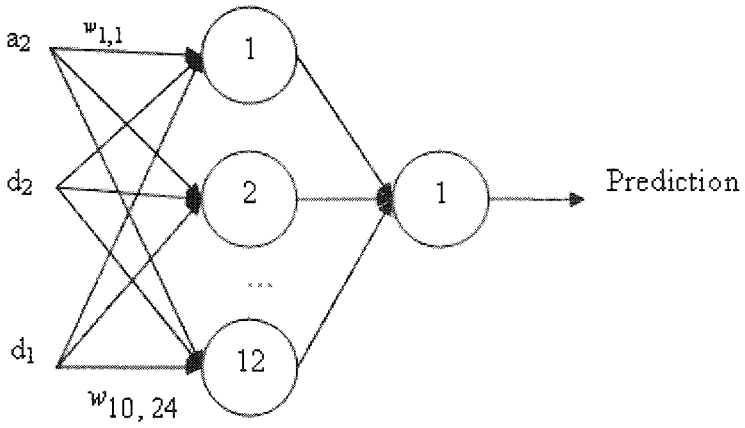


Figure 7.27 *Expansion of NN4*

The coefficient reconstruction series was used as the input vectors of training NN4. The input vector matrix and the target matrix are shown in equation (7.35).

$$X = \begin{bmatrix} a2_1 & a2_2 & \dots & a2_{50} \\ d2_1 & d2_2 & & d2_{50} \\ dl_1 & dl_2 & & dl_{50} \end{bmatrix} \qquad T = \begin{bmatrix} s_1 & s_2 & \dots & s_{50} \end{bmatrix}; \tag{7.35}$$

During the training process, 4000 epochs (iterations) of training were executed to minimize the mean square error between target and forecasting. As seen in Figure 7.28, the error diminishes to 10^{-12}. The exact error value is 7.6161 times 10^{-13}, which indicates a good performance of the neural network.

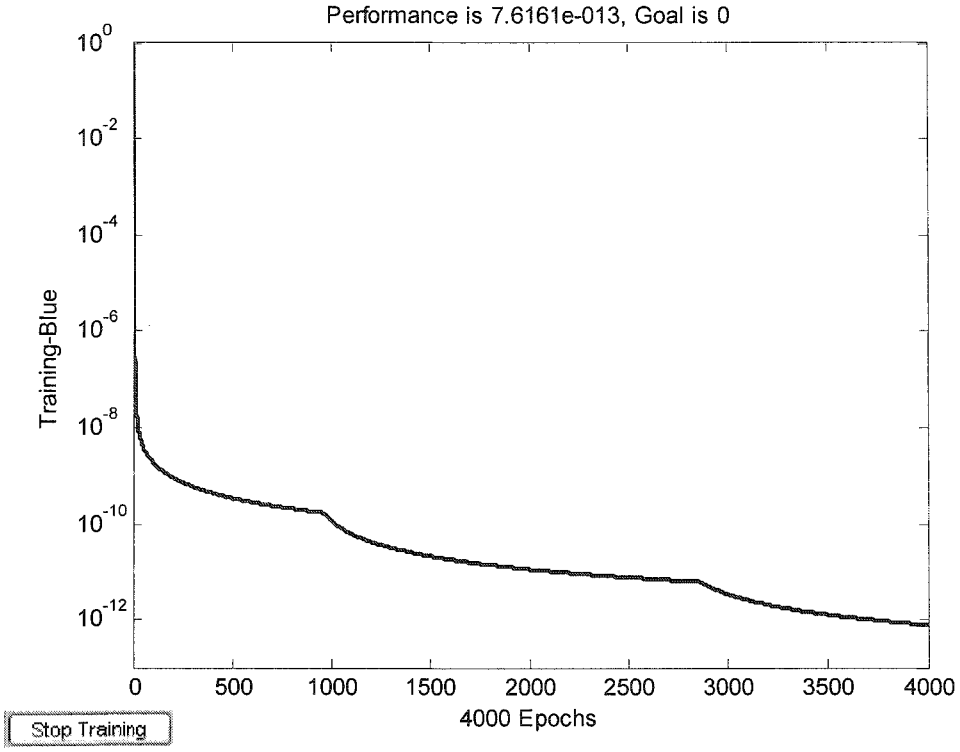


Figure 7.28 *Training results for mean square error*

After training, the mean square error performance was acceptable; the trained network was tested with a_2, d_2 and d_1 reconstruction series to produce the forecast of the original time series $g(t)$. The input vectors matrix X is:

$$X = \begin{bmatrix} a2_{51} & a2_{52} & \dots & a2_{64} \\ d2_{51} & d2_{52} & & d2_{64} \\ d1_{51} & d1_{52} & & d1_{64} \end{bmatrix} \tag{7.36}$$

The output is forecast of $g(t)$

$$forecast \cdot g(t) = \begin{bmatrix} \hat{s}_{51} & \hat{s}_{52} & \dots & \hat{s}_{64} \end{bmatrix}; \tag{7.37}$$

After generating the output forecast, post processing is required in order to fit the forecasts into the original sales series data scale. The post processed data is plotted in Figure 7.29.

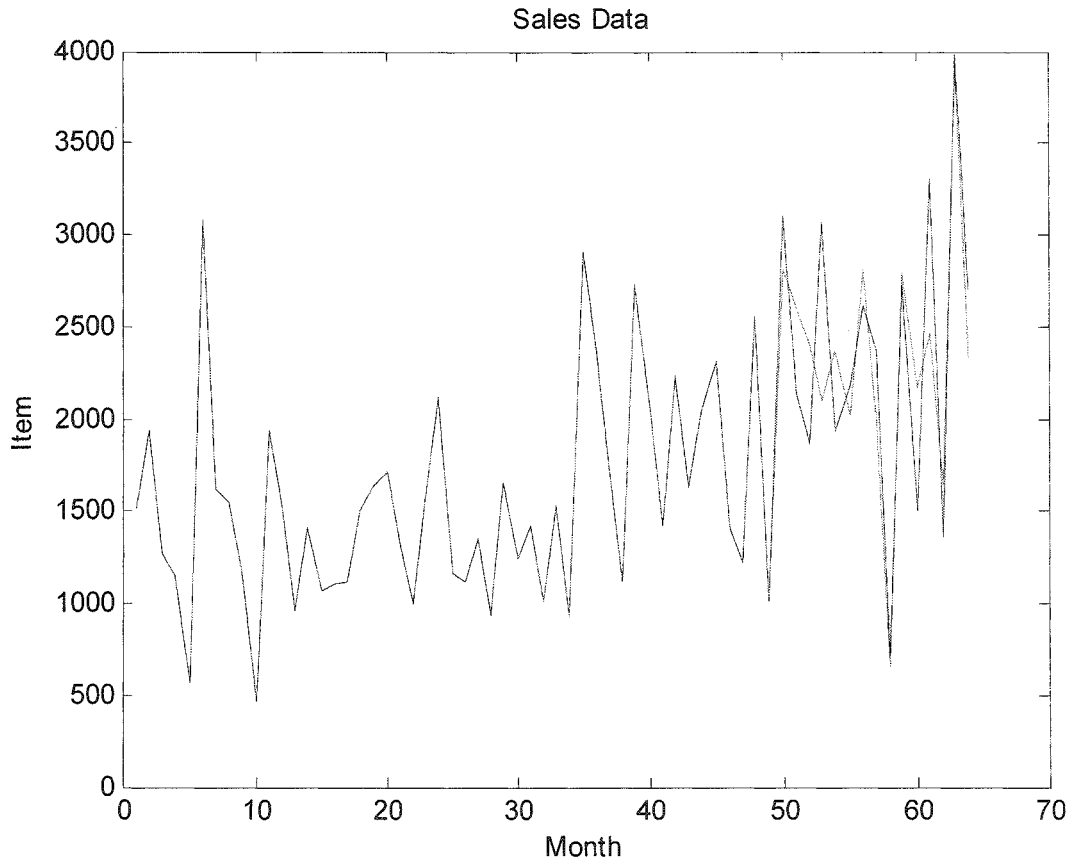


Figure 7.29 *Comparison between forecast and original time series $g(t)$. Blue curve shows the original time series $g(t)$. Red curve shows the forecast for the period: 51 to 64 months.*

The accumulative absolute difference of the forecast and real data from month 50 to 64 is 5822 units. The average absolute difference is 388 units per month. The forecasting performance has been greatly improved.

7.4 Application of Wavelet Neural Network Forecasting to the Supply Chain Model

In this section, wavelet neural network forecasting is adopted to produce forecast results for the supply chain model. The best performance was achieved at cost = 6354 units, when α=0.4 and β= 0. The forecasts are plotted in red, against sales inputs in blue, in Figure 7.30.

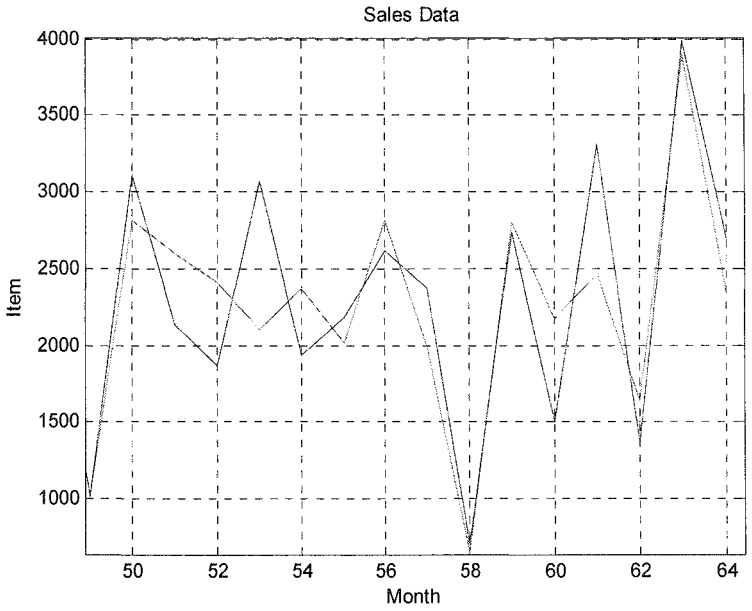


Figure 7.30 *Wavelet neural network forecasting result (red) and original sales data (blue)*

In Figure 7.30, the red forecast curve is very close to the blue sales curve over a period of many months. These accurate forecasts resulted in better inventory performance in the simulation. The hub inventory varies from 500 units to -100 units throughout the time period in Figure 7.31. The system performance has obviously improved. The simulation costs of 6354 units are quite close to the ideal costs of 4500 units, as discussed in equation (6.44).

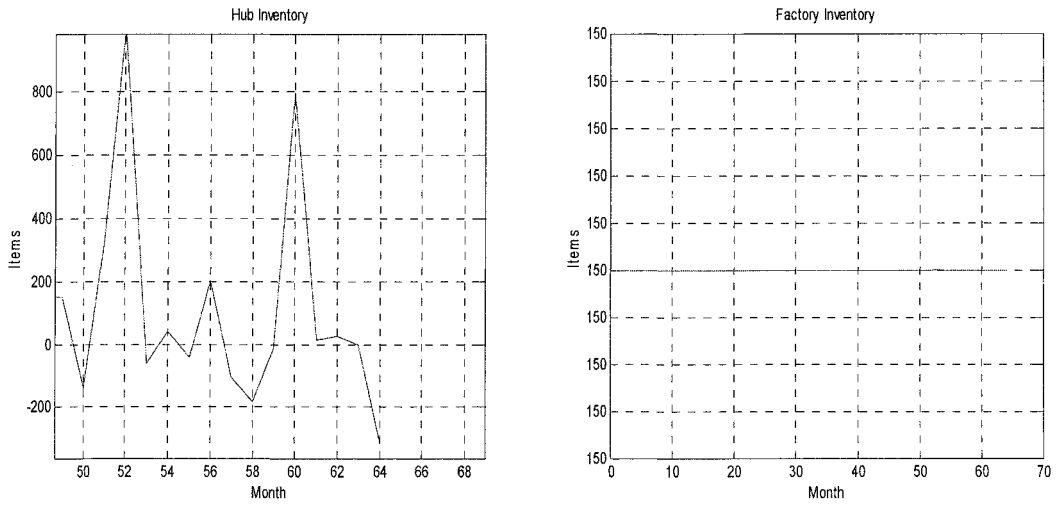


Figure 7.31 *Inventory response with wavelet neural network forecasting*

In comparison with the *ARMA* forecasting model, the wavelet neural network could pick up the nonlinear characteristics, such as trend and varation frequency, and produce better forecasts. The forecasting error for wavelet neural network forecasting and *ARMA* methods with different preprocessing are listed in Table 7.3. The error is calculated as the absolute accumulative difference between the forecast and real data in periods of month 51 to month 64. The corresponding supply chain costs are also provided in the last row. It is clear that the neural wavelet forecast exhibits a significantly better performance. The neural wavelet average error is only 388 units compared to 681 units for the best *ARMA* model. 388 units, are 9% of the upper boundary and 25% of the lower bound for the original data range from 1500 to 4000 units. This forcasting error is within the tolerance range. The minimum supply chain costs of 6354 units generated by neural wavelet forecasting are much less than the 18524 units produced by the differencing *ARMA* model. It can be concluded from the above analysis that wavelet neural network forecasting brought a remarkable improvement to the overall supply chain performance.

	No Preprocessing	Logarithm	Differencing	Logarithm and Differencing	Neural Wavelet
Accumulative error (units)	11806	11535	10219	11258	5822
Average error (units)	787	769	681	750	388
Supply chain Costs (units)	24313	23176	18524	22109	6354

Table 7.3 *The costs of the forecast using different methods*

Chapter 8: Learning Microworld

The supply chain microworld developed in this chapter enables the planning manager to test planning strategies in a virtual environment. Different forecasting methods, first-order exponential prediction (the forecasting equation with Θ), *ARMA* (autoregressive and moving average) and wavelet neural networks are applied as the forecasting function in the original supply chain model to evaluate the impact of forecasting on supply chain performance.

The term "microworld" is defined as a computer-based learning environment for children, in which they could program the environment, see how it responded, and draw out their own understanding of the principles of mathematical relationships (Seymour Papert, 1972). Gradually, it has come to mean any simulation in which people can run experiments, test different strategies and build a better understanding of the real world which the microworld depicts. It is the explorative learning environments that allow a free combination of possible and impossible conditions and concepts.

Simulation as defined by Roberts *et all.* (1983) or Pidd (1997) has been used on computer for many years. It imitates the real world by using a model which simplifies, but maintains the essential characteristics of the situation. The models may take different forms i.e. computer or mathematical models. They will facilitate experiment or risk free play. The experiment may take the form of enacting scenarios which may assist the user to develop new mental models (Senge *et all.*,1994).

The heart of the learning microworld will be a case depicting a problematic business situation. The business situation is represented by a dynamic model enabling the user to test different strategy policies to measure business performance. These models may also enable users to create their own simulations. Feedback is provided on performance. Finally, the Internet potentially provides an opportunity to encourage communication between learners.

The computer based training and simulations microworld was developed with Visual Basic for Draeger internal use. A subsequent Java version microworld was also built for wide usage on the Internet for research on the supply chain.

8.1 The Design of the Microworld

The microworld is constructed based on the discrete time supply chain model described in the previous chapters. Since the components in the discrete mathematics model are assumed to update at regular intervals, the corresponding microworld is developed using the next-event technique (Preece Jenny, 1994). In this case, the system is only examined when it is known that a state change is due. These state changes are the events. Therefore, the time in the simulation is moved from event to event. This makes possible to clearly and accurately identify when significant events occurred in the simulation. With the implementation of the next event technique, all the elements of the supply chain in our microworld are updated at regular intervals, which is one month in the simulation.

The microworld is developed for clients' direct use, rather than use by an analyst on behalf of clients. The client will take his or her own control over the analysis. Thus, it is necessary to adopt a user friendly graphic display of analysis and results. The supply chain structure in the microworld has been programmed into a dynamic work flow. The display clearly shows the product moving from element to element. This graphic design gives a representation of the logical flow of materials and information. The user can quickly gain an idea about whether the model is right or wrong. This logical clarity also provides the modeller with an opportunity to develop the model with the client. The dynamic graphic interaction can easily reveal the errors of logic when entities are seen to change state wrongly.

8.2 User interface

The completed software is programmed as an .exe extension file that can be run by users. The user interface is programmed as a window form, as shown in Figure 8.1.The user interface consists of six parts. The menu at the top of the window provides some options for different models and basic controls of the software. The 'Work Flow Panel'

in the centre shows the structure and details for different supply chain models. The dynamics of each component can be perceived from this panel. Users can apply different planning policies by operating the sliders and text box to change the planning parameters in 'Strategy Panel' in the bottom right. The hub inventory and incoming order can be visualised as a dynamic curve in 'Inventory Graphic Panel', bottom left. The 'Index Panel' shown on the top right offers users flexible control. Users can observe any details at any time unit point during the simulation. The costs label on the right of the window shows the accumulative costs of the current simulation. More details of each of the panels and menus are provided in the following sections.

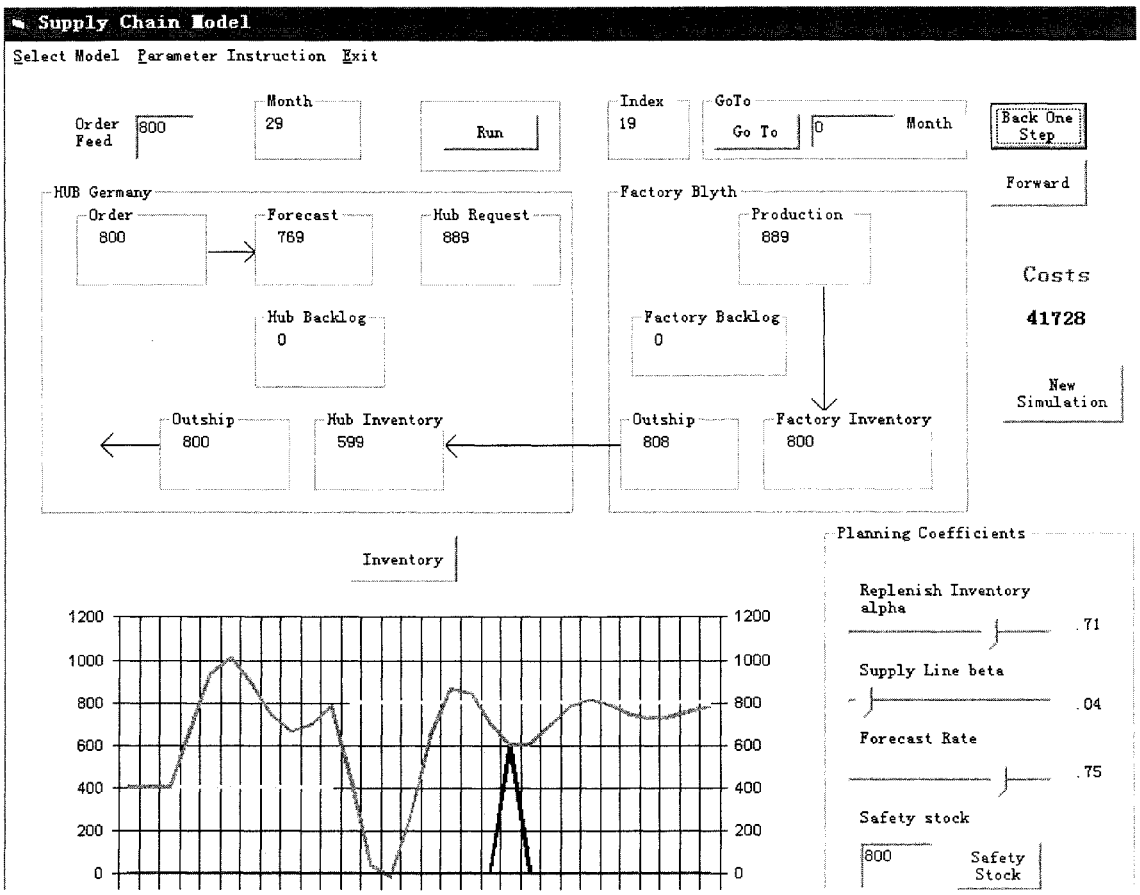


Figure 8.1 *The main interface for learning microworld*

8.2.1 Menu

There are three items in the menu bar. The drop down menu 'Select Model' provides three options for the user to choose a differently structured supply chain; the default is the model without delay. The model with production delay and the model with planning delay are optional. By clicking the options in the menu, the corresponding

supply chain model will turn up in the work flow panel. For the new user to our supply chain system, it is necessary to list the meaning of each control parameter. The 'Parameter Instruction' gives further details to explain parameters α, β, Θ and Q. As defined in previous chapters, α is the fraction of the discrepancy between desired and actual inventory. β is the fraction of the supply line taken into account. Θ is the demand expectation update rate. Q is the desired inventory or safety stock. The last menu item is 'Exit'. It will terminate the program and close all the relevant windows in the supply chain system.

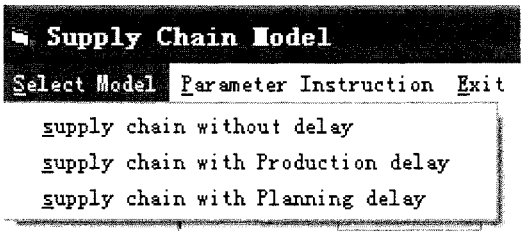


Figure 8.2 *The menu and corresponding items in the Microworld*

8.2.2 Work Flow Panel

The overall structure of the model has been programmed in the working flow panel (Figure 8.3). Each work unit in the supply chain, such as forecasting, production, outshipment and inventory has been represented by a square box with a title on the top. The content in the square is the quantity of that title unit. The text box 'Order Feed' on the top left corner provides users with the option of making an incoming sales order. The default value is 400. The user can change this number according to his sales assumption. After entering the sales quantity for the next month, clicking the button 'Run' next to the text box will send this customized order into the box labelled 'Order' in the upper left of the 'HUB Germany' frame, and the calculation for the whole model is executed once. The number in each component is updated. The calendar of the simulation goes forward for a month. The box labelled 'Month', next to the button 'Run', shows the time (current month) of the simulation.

The structure of the supply chain model in this microworld is similar to that discussed in the previous chapter. It comprises two main frames, 'HUB Germany' and 'Factory UK'. The 'HUB Germany' frame consists of six components, 'Order', 'Forecast', 'Hub Requirement', 'Hub Backlog', 'Hub Outship' and 'Hub Inventory'. The 'Factory UK'

frame consists of four components, 'Production', 'Factory Backlog', 'Factory Inventory' and 'Factory Outship'. Each of the corresponding supply chain working units can be found in the previous model specification chapters. The observer can track the movement of the sales order from the beginning of hub planning, forecasting, and generating requirements to factory production and shipment. The whole working process of the supply chain has been clearly represented and is easy to follow.

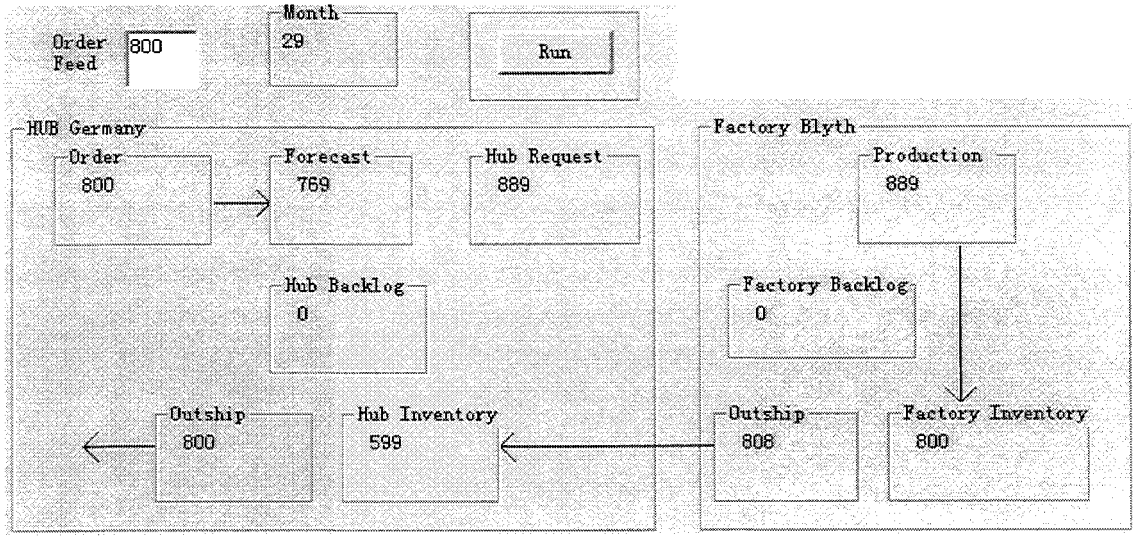


Figure 8.3 *The work flow panel for the model without delay in the Microworld*

For the model with additional production delay, the working flow panel changes (Figure 8.4). It is clear that two additional boxes 'Delay1' and 'Delay2' representing the production delays have been inserted between 'Production' and 'Factory Inventory'. By pressing the button 'Run', the user can observe that the number representing the products moves from 'Production' down to 'Delay1', 'Delay2' and 'Factory Inventory' step by step.

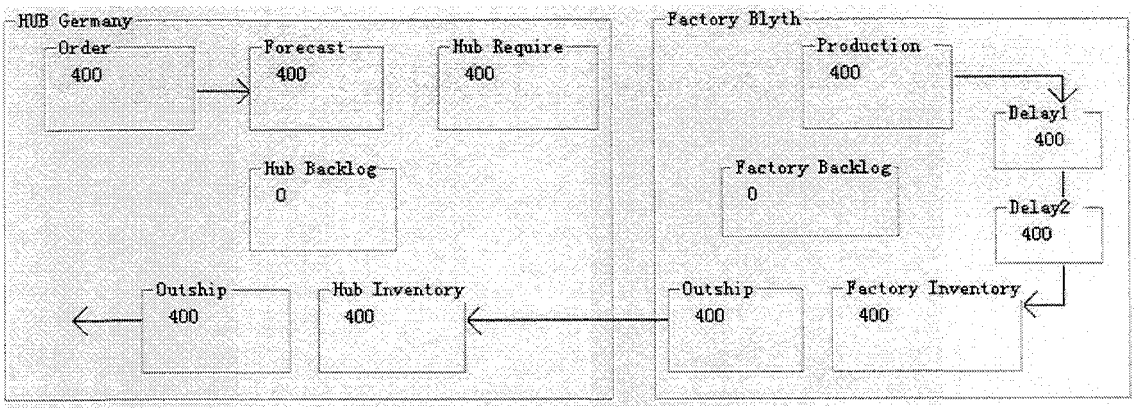


Figure 8.4 *The work flow panel for the model with two additional production delays in the microwrold*

The model with planning delay introduced another 'Info Delay' in the planning process in Figure 8.5. A solid arrow line representing shipping delay appears between hub and factory.

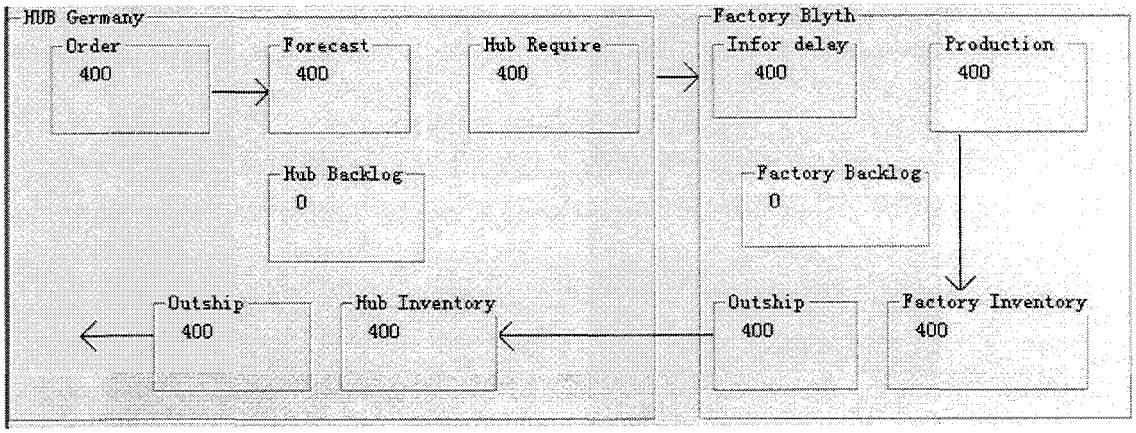


Figure 8.5 *The work flow panel for the model with planning delay in the Microworld*

Given different sales feeding, the user can perceive every change in detail within the supply chain. Some of the strange behaviour of the inventory plot curve can be explained by inspecting the dynamics in the flow panel. Furthermore, potential solutions can be found to fix the planning problem.

8.2.3 Strategy Panel

In the 'Strategy Panel', an aggressive or conservative planning policy can be simulated by changing the system parameters α, β, Θ and Q. The meaning of these four parameters has been given in Parameter Instruction. The changing of α, β, and Θ is

implemented with three sliders. By moving the cursor on the slider, the values of the parameters are varied from zero to one. The initial value for α is 0.26 and 0 for β. Θ is initialised as 0.75. Instead of using the slider, the text box can be employed to edit the safety stock. The default desired inventory is 800. Clicking the button 'Safety Stock' will change the configuration of the desired inventory in the model.

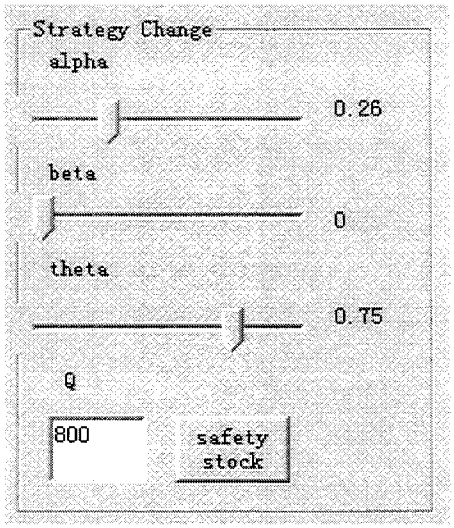


Figure 8.6 *The strategy panel in the Microworld*

8.2.4 Inventory Graphic Panel

Once the user presses the button 'Run', the supply chain model is actived. The detailed information is calculated and all supply chain components are updated. The hub inventory data and incoming order is recorded and plotted in this graphic panel (Figure 8.7). The x-axis in the chart represents the time units in a month. The y-axis denotes the quantity of the inventory level. The zero inventory level is highlighted in red. The inventory point above the red line is the positive inventory. The inventory below the red line is the hub backlog. To clarify the influence of the incoming order on the inventory dynamics, the hub inventory is plotted in green against the incoming order in yellow in the Figure 8.7.

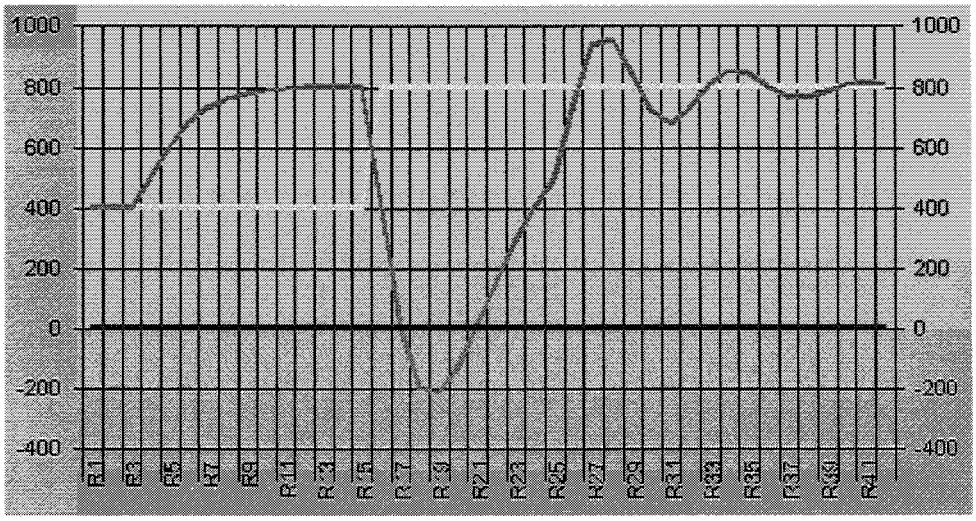


Figure 8.7 *The graphic panel in the microworld*

This graphic expression of the hub inventory and sales orders allow the user to observe the trend or variation pattern from long-term inventory history plots. Comparing with the zero inventory level in red, the user can easily recognize the hub backlog and positive inventory.

8.2.5 Index Panel

After a long-term simulation, the user may need to go back to a particular time slot to inspect the change of detail in the supply chain. The index panel has been introduced to manage the model more flexiblely. By clicking the button 'Back One Step' shown in Figure 8.8, the whole system state moves one step backward and each supply chain component shows the last month's quantity in the unit box. At the same time, a blue triangle is plotted as in Figure 8.9 to indicate the current inventory level. Keep pressing the back button, and the blue triangle pointer will move left until the start of the simulation. The box labelled 'Index' shows the month which the user is observing. The default Index label value is the same as the box labelled 'Month'. Each times the user presses the 'Back' button, the Index box shows the observed month.

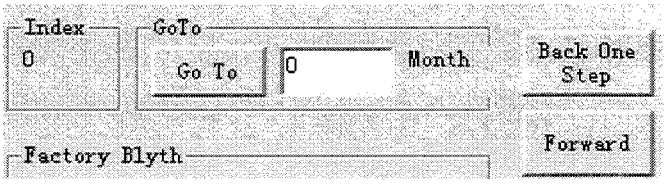


Figure 8.8 *The Index panel in the Microworld*

Conversely, by clicking the 'Forward' button the model will move one step forward. One step is added in the Index label. The blue triangle in Figure 8.9 moves forward to the month shown in the Index box. The details for each component are updated until the Index arrives at the end of the simulation.

In some cases, the user needs to move to one particular month from the current state. It may take a lot of clicking, using the 'Back' or 'Forward' button to achieve the target state. Alternatively, the 'Go To' button and relevant text box shown in Figure 8.8, are provided to simplify this task. The user can enter the month which he/she wants to observe and press the 'Go To' button next to the text box. The model will backward or forward to that particular month. All the relevant supply chain detail information is updated to match that month. Simultaneously, the blue triangle pointer appears in the graphic panel and points at the corresponding hub inventory.

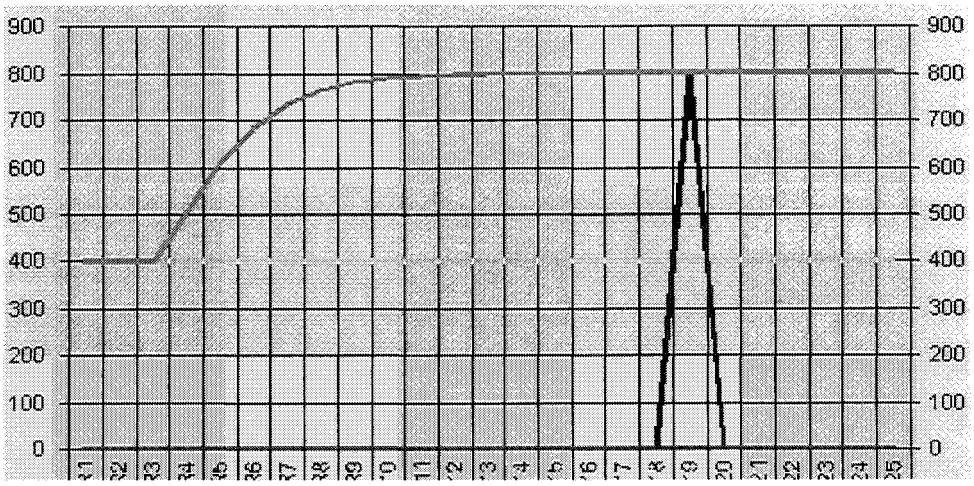


Figure 8.9 *Plots of hub inventory with index pointer*

8.2.6 Performance

After the discussion with the planning manager, the performance of the simulation is examined considering two factors. From the customer service aspect, failure to deliver to the customer on time should be counted as a major cost in measuring system performance. This delay in shipping may lead to the lose of potential customers. In the simulation model, those sales orders that cannot be shipped on time are backlogged in hub. Therefore, the hub backlogs become one part of the cost function. Each hub backlog is accounted as two units in cost function.

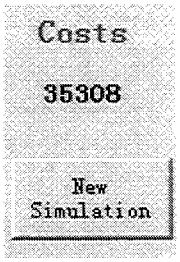


Figure 8.10 *Numeric expression of the cost for measuring supply chain performance*

On the other hand, the warehouse managers may concentrate on the cost of storing the product in the warehouse. In the simulation model, each unit of inventory scored one point per month in the cost function. Hence, the total costs of the supply chain are the sum of hub backlog cost and inventory costs (equation (8.1)). Decision making that produces lower costs is regarded as achieving better performance.

$$\text{Cost} = F_{inv}(t) + H_{inv}(t) + 2 \times H_{blk}(t) \qquad (8.1)$$

The user can click the 'New Simulation' button to clear the current result. The system moves back to the initial situation and is ready for a new simulation.

8.3 Applications of the Microworld

The microwrold was tested with different types of sales input. The simulation results with the wave shaped sales inputs are illustrated in the graphic panel in Figure 8.11. The simulation lasts for 40 months. The sales order is 400 for the first 10 months and then oscillates around 400, 600 and 800 until the end of the simulation. The planning manager decides planning policy in terms of $\alpha = 0.3$, $\beta=0.25$, $\Theta=0.75$ and $Q=800$. The yellow curve in the Figure 8.11 shows the variation of the incoming order. The green curve represents the hub effective inventory.

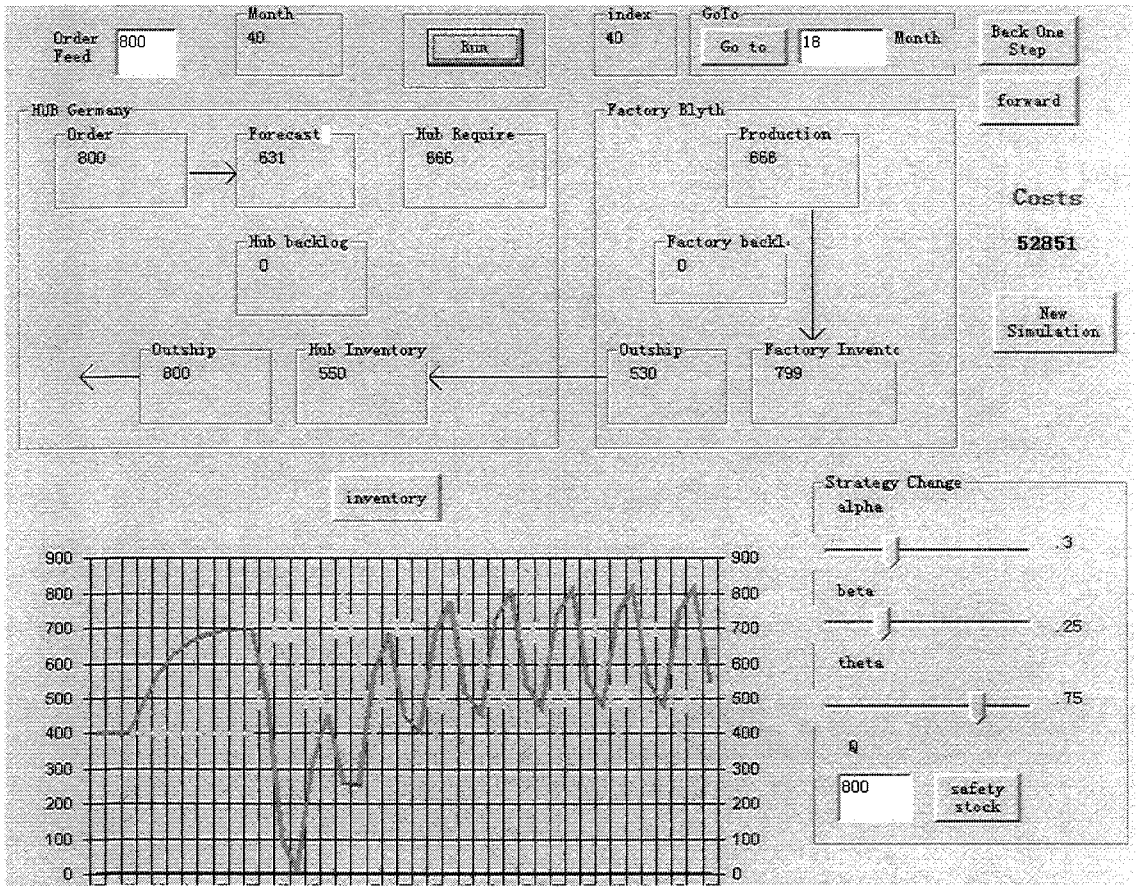


Figure 8.11 *Simulation results for $\alpha = 0.3$, $\beta = 0.25$ with wave sales inputs around 400,600 and 800.*

From Figure 8.11, it is clear that with the variation demand the hub inventory cannot be stabilised at the desired level. The hub inventory follows the oscillating pattern of the sales input, but the variation of curves does not match. The peak of the green wave is nearly of the same level as the yellow wave. The cost is calculated, as discussed in the performance section. The results are accumulated after each round of the simulation. The final costs were 52851 units in 40 months' simulation.

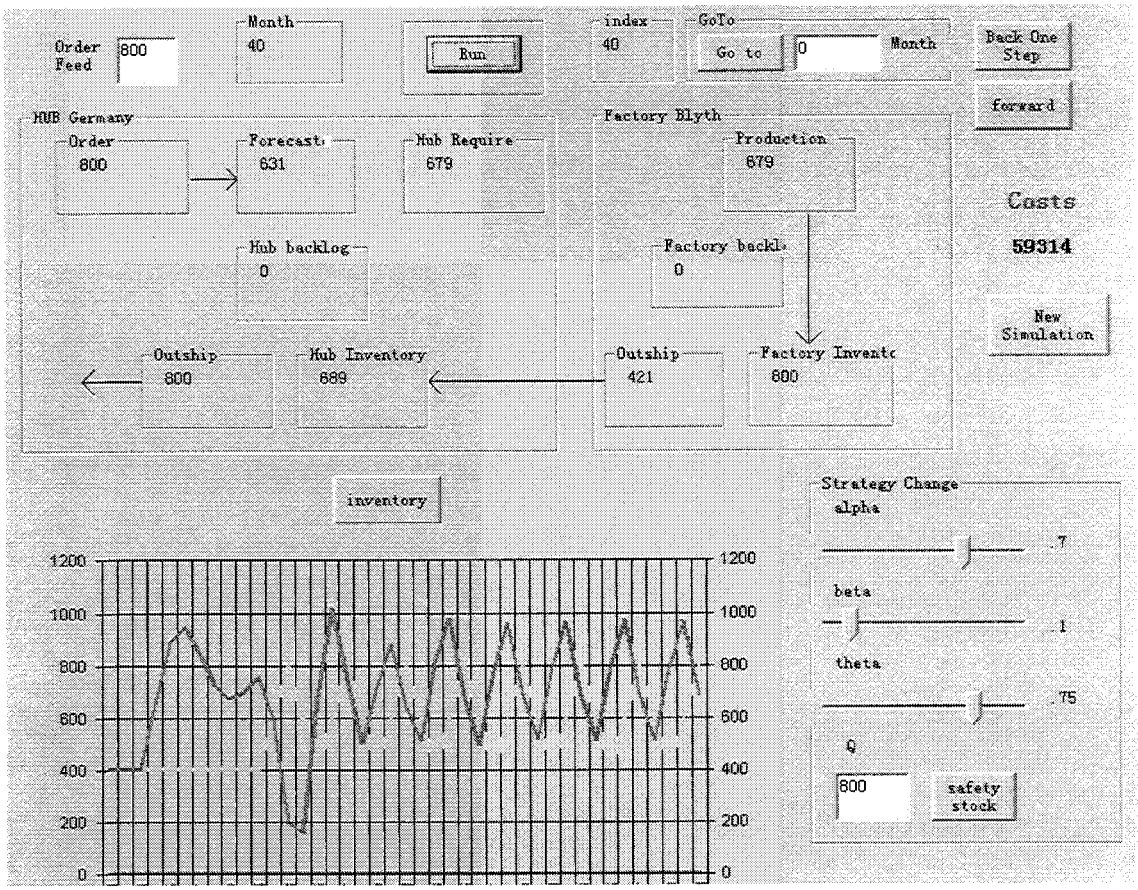


Figure 8.12 *Simulation results for α = 0.7, β = 0.1 with a wave sales inputs around 400,600 and 800.*

Increasing α to 0.7 and decreasing β to 0.1, a more aggressive planning policy is applied to hub and factory planning. Both the hub and factory want to replenish their local inventory to the desired level or safety stock level in very short periods. In the oscillating sales inputs scenario, the hub inventory in green, in Figure 8.12, varies in a similar pattern but with greater amplitude compared to the inventory performance under the conservative policy in Figure 8.11. The peak of the green wave reaches 1000 units, which is bigger than the peak of the sales at 800 units. The large amount of inventory cost more for the warehouse. The final score for this simulation was 59314 units, which is higher than the previous case. Therefore, it can be concluded that the conservative planning policy with $\alpha = 0.3$, $\beta = 0.25$ achieved a better supply chain performance than the aggressive planning policy with $\alpha = 0.7$, $\beta = 0.1$.

8.4 The Java Version of Microworld on the Internet

The simplified three sector supply chain model has been programmed for the web Microworld on the Internet and for university research. It was developed by Java Applet and embedded in a webpage which is accessible from any web explorer. The web microworld is open to any kind of users on the Internet. By working with this web microworld, users can observe and experience the effects of decision making in supply chain management. The three sector model is shown in Figure 8.13.

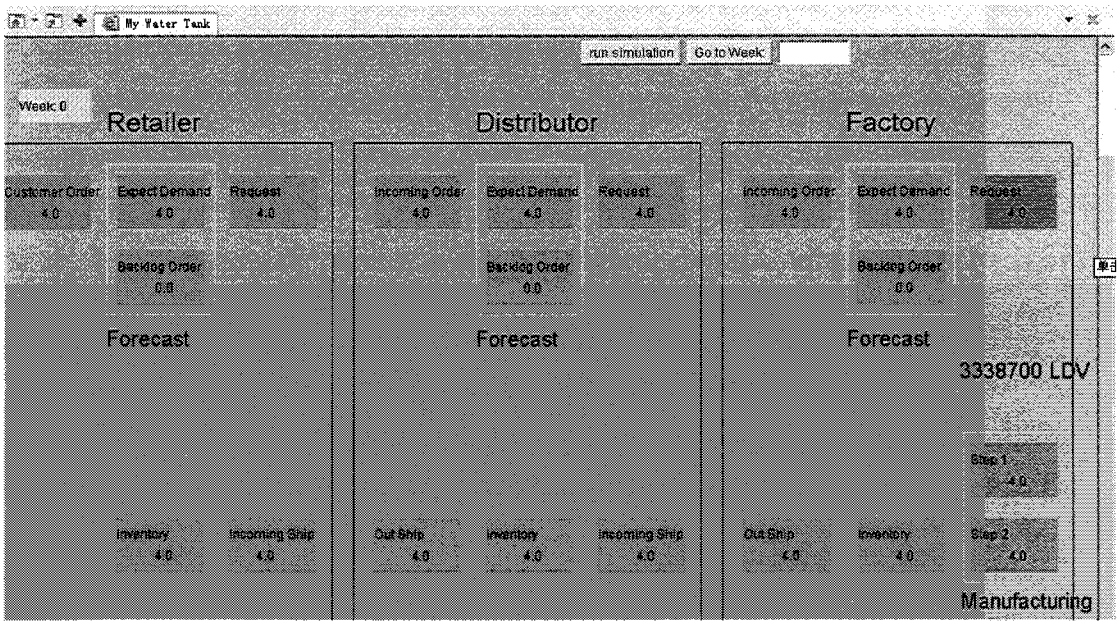


Figure 8.13 *The interface of the web Microworld*

The user can compare the different behaviour of the three sector web microworld and the two sector offline microworld. The similarities are that both models behave in an unstable manner with aggressive planning policies. The three sector model gains greater inventory oscillations and needs a longer time to remove the oscillation. This web microworld is based on the Beer Game equations. It does not represent the behaviour of the Draeger centralised supply chain. However, it is still very helpful for the users and developers to explore further research. It was a preparation and rehearsal for constructing the three stages Draeger model and a good way to communicate with the client. The program code is provided in Appendix C.

8.5 Reflections

Once this prototype of the software is available, a user based assessment can be carried out. This is a very important development activity, which can confirm the extent to which user and organizational objectives have been met, as well as provide further information for refining the design. It is important to consider the characteristics of the users (or predicted users) of the microworld who take part in the testing. As the microworld was developed based on Draeger heuristics, the Draeger planning department manager Mel Hedley, who has many years of supply chain management experience, was invited to be involved in the pilot testing following heuristic aspects of usability:

(1) Visibility of system status;

(2) Match between system and the real world;

(3) User control and freedom;

(4) Recognition rather than recall,

(5) Flexibility and efficiency of use;

(6) Aesthetic and minimalist design;

(7) Help users recognize, diagnose and recover from errors;

(8) Help and documents.

The state of the system is made visible on the graphic panel that displays the inventories. The work flow panel shows the information and material movement. The Draeger planning manager confirmed that the behaviour of the model met his expectations. The behaviour is influenced by the variety of planning parameters: α (the fraction of the discrepancy corrected between the desired stock and the effective inventory), β (the fraction of the supply line taken into account), and θ (the rate at which the expectation is updated). The user can observe and learn the dynamics of different scenarios. The planning parameters are clearly explained. However, the description of the supply chain costs need to be further developed in the help documents. The controllability of the system is complicated for the new user. The

Draeger manager suggested an extra tabular format of supply chain components to help the user who familiar with Excel operation.

By working with the Learning Microworld, managers could have a better understanding about the working principles of their supply chain system. Practically, they can test different planning strategies in different kinds of sales inputs by examining the inventory performance. Compared to traditional tools, like Excel spreadsheets and critical path management, that focus on the solutions, the microworld allows for developing the managers' understanding and intuition about the problem situation (Robert, 1999). By varying the demand, the manager can explore the impact of the planning parameters on the model. However, the success of the microworld depends on the validity of the model and on the suitability of the user interface. Feedback from the clients suggests the user found the current user interface rather complex. Future developments could focus on research implementation of a user friendly interface. Table 8.1 lists the advantages and disadvantages of the microworld.

Advantages	Disadvantages
Builds a dynamic understanding of supply chain management;	The interface is not very user friendly; Too many inputs in one screen;
Experiments with the supply chain without making changes to the real system;	The explanation of supply chain costs are not clear;
Predicts system behavior under conditions not previously experienced;	Introducing an extra tabular format of supply chain components which is similar to Excel will be more helpful;
Allows observation of the dynamics of the supply chain from different perspectives.	Users need supply chain background knowledge to understand the system.

Table 8.1 *Advantages and disadvantages of the microworld as perceived by the user*

Chapter 9: Conclusion

The overall aim of the research was to model and analyse the behaviour of a centralised supply chain and provide explanations and insights into the link between decision strategies and performance. It was also envisaged that the application of a combination of theories, i.e. system dynamics, control theory, nonlinear dynamics and forecasting analysis would lead to a more effective exploration and new insights into the behaviour of the supply chain with regard to decision policies. This conclusion will reflect on the modelling aspects and summarise the findings in relation to the use of the combined theories. These insighst could not have been achieved through the use of a single theoretical approach.

The context of the study was the breathing equipment manufacturer Draeger Safety, Blyth. Thus, the modelling process was embedded in the organisation and was focused on the client's needs. The client was an active participant in developing, testing and validating the model. It is, however, important to emphasise that the purpose of the approach undertaken in this study was to use the model as a vehicle for understanding and learning about the problem situation, i.e. supply chain management at Draeger Safety UK. This required the effectiveness of communication with the client to be facilitated and monitored throughout the process in order to ensure that the meanings, assumptions and insights were shared and validated within the research team (researcher and members of the planning department). This was achieved not just through conversations with individual members of the team but through numerous team discussions. All discussions included achieving consensus with regard to the issues that were considered within each stage of the modelling process. This included shared understanding and learning about:

- The dynamic behaviour of the intended (ideal) supply chain model at Draeger Safety UK;
- The impact of manufacturing delays on the performance of the supply chain;
- The impact of information delay and inaccuracies in planning on the supply chain performance and the effect of different forecasting methods.

The simplified model of Draeger centralised supply chain was developed based on the Beer Game heuristics (Sterman, 2000). The modelling approach followed the system dynamics process of model development and validation. The findings suggest that management decisions are based on two parameters: α is the correction factor of the discrepancy between the current inventory and its desired level; β takes into account any previously placed orders that have not yet arrived and are thus 'still in the pipeline'. α and β determine the corrective feedback within the hub and within the factory. The different psyches of decision makers could be simulated by different values of α and β. Big α and small β represent the thinking of an aggressive manager focusing on his local sector, ignoring the impact of previous decisions. Analysis shows that a purely aggressive strategy, without considering the history of the system, will usually lead to unwanted behaviour, i.e. large oscillations of inventories. Decreasing α in terms of slow stock recovery produces a stabilising effect. Small α and big β represent a manager that thinks outside of the immediate area of control and takes some or full consideration of the history. Taking into account the impact of previous decisions by increasing β has a stabilising effect. These findings were validated within the research group and the labelling and meaning of 'aggressive' and the 'full consideration' strategy were derived as a shared understanding in group discussions with the planning department.

The use of control theory analysis and, in particular, block diagrams and eigenvalue z-plane plots, leads to the understanding that the two systems, hub and factory, are isolated as there is no corrective feedback loops that cover both sectors. Moreover, since the systems are 'isolated', the complex eigenvalues generating oscillations are produced from hub management decisions only. Therefore, poor management decisions in the hub cannot be corrected by good decisions in the factory. On the other hand, a good manager in the hub may be able to mitigate poor management decisions in the factory. The hub becomes the primary route to instability.

The production delay model was built by adding two production delays in the factory within the original model. In this situation, α and β have similar overall effects on the performance. The system becomes more sensitive to management decisions. The same values of α and β which led to stable behaviour in the original model could produce big oscillations in the production delay model. The extreme conservative management

policy with small α and big β could stabilise the system. However, as pointed out in the discussions of the research group, this will delay on time delivery and harm customer satisfaction. Alternatively, increasing the safety stock could eliminate the oscillation. However, holding a large amount of safety stock will increase inventory costs. With the extra production delay state, the previous roles, which lead to the instability, are changed. Control theory analysis and the existence of non-linearities due to the accumulation of backlog, suggest that the factory becomes the main route toward instability. Chaotic behaviour occurs from aggressive strategies taking no account of the supply line. It is the hub that can do little about poor management decisions in the factory.

The planning delay model introduces a planning delay between the hub and the factory to mimic the scenario of 'getting the forecast wrong' and 'compatibility problem in different planning software'. Control theory and non-linear analysis suggest that the hub management policy is the primary route to instability and follows a more severe path. The dynamics of the system are mostly dependent on α, the stabilising influence of β only lessens the severity of the route to instability. Thus, as discussed in the research team, good management and management policies are crucial in this situation. Planning delay and the scenario of 'getting the forecast wrong' and 'compatibility problem in different planning software' could be minimised in reality. The forecast accuracy needs to be improved to avoid these circumstances. The need to improve the forecast was emphasized wihin the research group.

In order to improve forecasting accuracy, the classic statistical forecasting model *ARMA* was applied to predict the sales series data collected from Draeger Safety, UK. The autocorrelation of the original data was calculated to explore the interrelationship within the time series. However, no significant autocorrelation was found; up to 8 auto-regression and moving average parts were compiled in different combinations to form different structured models. Three kinds of data preprocessing of logarithm of original series, differencing series and logarithm differencing series were applied to preprocess the time series. Part of the original series was used in training and in constructing the best structure of the model. Analysis shows that the model with two regression parts and one moving average part achieved a better performance when the original series

was processed with differencing. However, the forecasting result is not good enough; it cannot follow the basic trend of the original series.

Like most of the financial time series in the real world, Draeger sales data is unstationary. The traditional statistical forecasting methods cannot track the data characteristics from these nonlinear time series. Therefore, wavelet neural network forecasting was adopted to analyze and forecast the unstationary time series. The 'Daubechies two' wavelet was chosen as the mother wavelet to perform the discrete wavelet decomposing and reconstructing of the original time series. After wavelet processing, the data characters as overall trend and oscillation period has been seperated in different level of time series. Forecasting with neural networks was carried out on each level of time series. The trend, variation prediction were grouped together to produce the final forecast for the original time series. This led to better forecasting accuracy. However, it is very time consuming to find out the optimized neural networks structure for different data sets. No fixed procedure for optimizing the neuron quantities and connectivity was developed.

The microworld was programmed for supply chain managers to explore the effects of different management policies. Different planning parameters, such as safety stock, Q, recovering inventory speed, α, and consideration of incoming orders, β, have been programmed as sliders and editable components for the user to simulate different planning heuristics. By working with the microworld, managers could have a better understanding of Draeger's supply chain structure and planning policy and test different management policies and not worry about the impact in the real world. The feedback from Draeger managers shows that the mircorworld was helpful for their decision making; however, the interface is a little confusing. Users could get lost among too many buttons and sliders.

Different forecasting methods were evaluated in the supply chain model. Performance was measured in the cost of holding inventories and backlog. The research shows that wavelet neural network forecasting achieved the best performance. The better forecasting accuracy led to better supply chain performance.

9.1 Contribution to Knowledge and Significance of Study

The contribution to knowledge of this research can be identified as:

- Development and application of a combined approach to modelling and simulation of the supply chain, including system dynamics, control theory and non-linear and forecasting analysis;
- Modelling and simulation of a the supply chain in a real life context at the breathing equipment manufacturer Draeger Safety UK;
- Innovative insights into the behaviour of the supply chain that identify the route to instability and the impact of the forecasting on the efficiency of the system;
- Development and evaluation of a learning microworld allowing for the exploration of different decision strategies.

The findings have been published in two journal papers, two conference papers and one invited poster presentation, as listed below:

Niu M.,Sice P.,French I., Mosekilde E., (2007): The Dynamics Analysis of Simplified Centralised Supply Chain, The Systemist – Journal of the UK Systems Society, Oxford, UK, Nov.2007.

Niu M.,Sice P.,French I., Mosekilde E., (2008): Exploring the Behaviour of Centralised Supply Chain at Draeger Safety UK, International Journal of Information system and Supply Chain Management, USA, Jan. 2008.

French I., Sice P., Niu M., Mosekilde E.,(2008): The Dynamic Analysis of a Simplified Centralised Supply Chain and Delay Effects, System Dynamic Conference, Athens, July.2008.

Sice P., French I., Niu M., Mosekilde E., (2008): The Delay Impacts on a Simplified Centralised Supply Chain, UK Systems Society Conference, Oxford, UK, Sep.2008.

Niu M, Sice P., French I., (2008): Non-linear Forecasting Model, Northumbria Research Forum 2008, Newcastle upon Tyne, UK, Apr.2008.

9.2 Future Research

The research presented here represents only the initial stages of an investigation into Draeger's supply chain. Moreover, in the results presented decision policies have been applied and maintained over an extended (and, in real world terms, unrealistic) period of time. Never the less this analysis has found favour within Dreager Safety, UK and has proven itself to be a powerful decision support tool. It has become obvious that in our investigations so far we have only just scratched the surface of the rich variety of possible behaviours that can emerge. For forecasting research, combined multivariate linear models, such as dynamic regression and neural networks, could be a fruitful area in the future. It is the intention, therefore, to continue this research and to further investigate:

- The effect of using independent decision strategies (having different values of α and β at the factory and the hub) at different levels in the supply chain.
- The impact of the secondary hubs.
- Potential nonlinear or chaotic behaviour analyzed using in-depth tools such as the bifurcation diagram and Liapunov exponent.
- The modification of decision strategies (perhaps to a full state variable feedback form or production control in response to forecast) to improve supply chain performance.
- A systematic way to choose a suitable length of window for training the neural networks for forecasting.

Appendix A: Program for Three Models

Model Without Delay

```
clear
alpha = 1;
beta = 0;
q = 800;
theta = 0.75;
tstop = 250;

heff = [];
feff = [];

% Hub
horders_n = 400;
horders_n_1 = 400;

hblk_n_1 = 0;
hinv_n_1 = 400;
hforecast_f_1 = 400;
hreq_f_1 = 400;
hreq_n = 400;
hship_n_1 = 400;

hship_n = 400;

% Factory
fblk_n_1 = 0;
finv_n_1 = 400;
fprod_n_1 = 400;

fship_n_1 = 400;

 % Loop

for i = 1:tstop
if i > 100
    horders_n = 800;
 end % if

% Inventories

% Hub
hship_n = min(horders_n + hblk_n_1, hinv_n_1 + fship_n_1);
hinv_n=max(0,hinv_n_1 + fship_n_1 - hship_n);
hblk_n =max(0,hblk_n_1 + horders_n - (hinv_n_1 + fship_n_1));
heff(i) = hinv_n - hblk_n;
% Factory
fship_n = min(hreq_f_1 + fblk_n_1, finv_n_1 + fprod_n_1);
finv_n = max(0,finv_n_1 + fprod_n_1 - fship_n);
fblk_n =max(0,fblk_n_1  + hreq_f_1 - (finv_n_1 + fprod_n_1));
feff(i) = finv_n - fblk_n;

% Forecast & Production
% Hub
hforecast_f_2 = (1 - theta)*horders_n + (theta)*hforecast_f_1;
hreq_f_2 = max(0, alpha*(q - hinv_n + hblk_n) - alpha*beta*(fblk_n + fship_n)
+ hforecast_f_2);
% Factory i
fprod_n = max(0, alpha*(q - finv_n + fblk_n) + hreq_f_2);
```

```
% Time Step

% Hub
hblk_n_1 = hblk_n;
hinv_n_1 = hinv_n;
horders_n_1 = horders_n;

hreq_n = hreq_f_1;
hreq_f_1 = hreq_f_2;
hship_n_1 = hship_n;
hforecast_f_1=hforecast_f_2;

% Factory
fblk_n_1 = fblk_n;
finv_n_1 = finv_n;
fprod_n_1 = fprod_n;

fship_n_1 = fship_n;

end % for Loop
plot(heff,'-*');
xlabel('Month');
ylabel('Items');
%title('Hub Inventory');
grid;
```

Model with Production Delays

```
alpha = 0.8;
beta = 0.1;
q = 800;
theta = 0.75;
tstop = 250;

% Define store variables
heff = [];
feff = [];

% Define Initial Values
% Hub
hblk_n_1 = 0;
hinship_n_1 = 400;
hinv_n_1 = 400;
horders_n = 400;
horders_n_1 = 400;
hforecast_f_1 = 400;
hreq_f_1 = 400;
hreq_n = 400;
hship_n = 400;
hship_n_1 = 400;
% Factory
fblk_n_1 = 0;
finv_n_1 = 400;
fprod_n_1 = 400;
frequest_n_1 =400;
fship_n_1 = 400;
fpd1_n = 400;
fpd1_n_1 = 400;
fpd2_n = 400;
fpd2_n_1 = 400;
 % Loop

for i = 1:tstop
if i > 100
    horders_n = 800;
```

```
end % if

% Inventories

% Hub
hship_n = min(horders_n + hblk_n_1, hinv_n_1 + fship_n_1);
hinv_n=max(0,hinv_n_1 + fship_n_1 - hship_n);
hblk_n =max(0,hblk_n_1 + horders_n - (hinv_n_1 + fship_n_1));
heff(i) = hinv_n - hblk_n;
% Factory
fship_n = min(hreq_f_1 + fblk_n_1, finv_n_1 + fpd1_n_1);
finv_n = max(0,finv_n_1 + fpd1_n_1 - fship_n);
fblk_n =max(0,fblk_n_1  + hreq_f_1 - (finv_n_1 + fpd1_n_1));
feff(i) = finv_n - fblk_n;

% Forcast & Production
% Hub
fpd2_n=fprod_n_1;

fpd1_n=fpd2_n_1;

hforecast_f_2 = (1 - theta)*horders_n + (theta)*hforecast_f_1;
hreq_f_2 = max(0, alpha*(q - hinv_n + hblk_n) - alpha*beta*(fblk_n + fship_n)
+ hforecast_f_2);

% Factory i
fprod_n = max(0, alpha*(q - finv_n + fblk_n) + hreq_f_2 - alpha*beta*(fpd1_n +
fpd2_n));

% Time Step
% Hub
hblk_n_1 = hblk_n;
hinv_n_1 = hinv_n;
horders_n_1 = horders_n;

hreq_n = hreq_f_1;
hreq_f_1 = hreq_f_2;
hship_n_1 = hship_n;
hforecast_f_1=hforecast_f_2;

fpd2_n_1=fpd2_n;
fpd1_n_1=fpd1_n;
% Factory
fblk_n_1 = fblk_n;
finv_n_1 = finv_n;
fprod_n_1 = fprod_n;
fship_n_1 = fship_n;

end % for Loop

figure(1);
plot(heff);
xlabel('Month');
ylabel('Items');
title('Hub Inventory');
grid;

figure(2)
plot(feff);
xlabel('Month');
ylabel('Items');
title('Factory Inventory');
grid;

figure(3)
plot(feff(200:end),heff(200:end),'.');
xlabel('Factory effective inventory');
ylabel('Hub effective inventory');
```

Model with Planning Delay

```
alpha = 0.62;
beta = 0;
q = 1600;
theta = 0.75;
tstop = 250;

% Define store variables

heff = [];
feff = [];

% Define Initial Values
% Hub
hblk_n_1 = 0;
hinship_n_1 = 400;
hinv_n_1 = 400;
horders_n = 400;
horders_n_1 = 400;
hforecast_f_1 = 400;
hreq_f_1 = 400;
hreq_n = 400;
hship_n = 400;
hship_n_1 = 400;

% Factory
fblk_n_1 = 0;
finv_n_1 = 400;
fprod_n_1 = 400;
frequest_n_1 =400;
fship_n_1 = 400;
 % Loop

for i = 1:tstop
if i > 100
horders_n = 800;
end % if

% Inventories

% Hub
hship_n = min(horders_n + hblk_n_1, hinv_n_1 + fship_n_1);
hinv_n=max(0,hinv_n_1 + fship_n_1 - hship_n);
hblk_n =max(0,hblk_n_1 + horders_n - (hinv_n_1 + fship_n_1));
heff(i) = hinv_n - hblk_n;

% Factory
fship_n = min(hreq_n + fblk_n_1, finv_n_1 + fprod_n_1);
finv_n = max(0,finv_n_1 + fprod_n_1 - fship_n);
fblk_n =max(0,fblk_n_1  + hreq_n - (finv_n_1 + fprod_n_1));
feff(i) = finv_n - fblk_n;

% Forcast & Production
% Hub
hforecast_f_2 = (1 - theta)*horders_n + (theta)*hforecast_f_1;
hreq_f_2 = max(0, alpha*(q - hinv_n + hblk_n) - alpha*beta*(fblk_n + fship_n)
+ hforecast_f_2);

% Factory i
fprod_n = max(0, alpha*(q - finv_n + fblk_n) + hreq_f_1);
% Time Step
% Hub
hblk_n_1 = hblk_n;
```

```
hinv_n_1 = hinv_n;
horders_n_1 = horders_n;
hreq_n = hreq_f_1;
hreq_f_1 = hreq_f_2;
hship_n_1 = hship_n;
hforecast_f_1=hforecast_f_2;
% Factory
fblk_n_1 = fblk_n;
finv_n_1 = finv_n;
fprod_n_1 = fprod_n;
fship_n_1 = fship_n;
end % for Loop
```

Appendix B: ARMA Program

```
function [items costs] = call2( lg,diff );

lg = lg;
diff = diff;
span = 8;
erro=7;
pas=2;

for nu = 1:erro
    ne = nu-1;
for s =1:span
    if s ==1;
       for i =1:span
           cost (i+ne*255,1) = ex3(i,ne,lg,diff);
           po(i+ne*255,1:s+pas)=[i 99 ne];
       end
    end

    if s==2;
        m=i+ne*255+1;
        for i1 = 1:span
            for j1 = 1:span
                if i1<j1
                    cost(m)=ex3([i1 j1],ne,lg,diff);
                    po(m,1:s+pas)=[i1 j1 99 ne];
                    m=m+1;
                end
            end
        end
    end

   if s==3;
      m1=m;
      for i2 = 1:span
          for j2 = 1:span
              for k2 = 1:span
                  if i2<j2&j2<k2
                      cost(m1)=ex3([i2 j2 k2],ne,lg,diff);
                      po(m1,1:s+pas)=[i2 j2 k2 99 ne];
                      m1=m1+1;
                  end
              end
          end
      end
   end
   if s==4;
       m2=m1;
       for i3 =1:span
           for j3=1:span
               for k3=1:span
                   for l3=1:span
                       if i3<j3&j3<k3&k3<l3
                           cost(m2)=ex3([i3 j3 k3 l3],ne,lg,diff);
                           po(m2,1:s+pas)=[i3 j3 k3 l3 99 ne];
                           m2=m2+1;
                       end
                   end
               end
           end
       end
   end
```

```
    if s==5;
        m3=m2;
        for i4=1:span
            for j4=1:span
                for k4=1:span
                    for l4 =1:span
                        for o4 = 1:span
                            if i4<j4&j4<k4&k4<l4&l4<o4
                                cost(m3)=ex3([i4 j4 k4 l4 o4],ne,lg,diff);
                                po(m3,1:s+pas)=[i4 j4 k4 l4 o4 99 ne];
                                m3=m3+1;
                            end
                        end
                    end
                end
            end
        end
    end
    if s==6;
        m4=m3;
        for i5=1:span
            for j5=1:span
                for k5=1:span
                    for l5=1:span
                        for o5=1:span
                            for p5=1:span
                                if i5<j5&j5<k5&k5<l5&l5<o5&o5<p5
                                    cost(m4)=ex3([i5 j5 k5 l5 o5
p5],ne,lg,diff);
                                    po(m4,1:s+pas)=[i5 j5 k5 l5 o5 p5 99 ne];
                                    m4=m4+1;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
    if s==7;
        m5=m4;
        for i6=1:span
            for j6=1:span
                for k6=1:span
                    for l6=1:span
                        for o6=1:span
                            for p6=1:span
                                for q6=1:span
                                if i6<j6&j6<k6&k6<l6&l6<o6&o6<p6&p6<q6
                                    cost(m5)=ex3([i6 j6 k6 l6 o6 p6
q6],ne,lg,diff);
                                    po(m5,1:s+pas)=[i6 j6 k6 l6 o6 p6 q6 99
ne];
                                    m5=m5+1;
                                  end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
    if s==8;
        m6=m5;
        for i7=1:span
            for j7=1:span
                for k7=1:span
```

```
                    for l7=1:span
                        for o7=1:span
                            for p7=1:span
                                for q7=1:span
                                    for r7=1:span
                                if i7<j7&j7<k7&k7<l7&l7<o7&o7<p7&p7<q7&q7<r7
                                    cost(m6)=ex3([i7 j7 k7 l7 o7 p7 q7
r7],ne,lg,diff);
                                    po(m6,1:s+pas)=[i7 j7 k7 l7 o7 p7 q7 r7 99
ne];
                                    m6=m6+1;
                                end
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
end
[y,i] = min(cost);
po(i,:);
items = po(i,:)
costs = y


function[cost] =ex3(co,ne,loog,diff);
co = co;
ne = ne;
alpha = 0.99;
loog=loog;
diff=diff;
itm =max( co);
num = max(ne,itm)+1;  %regressed from which point

sz1 = max(size(co));
sz = sz1 + ne +1;     %how many the parameters

d =[  1519  1930    1272    1149    568 3085    1623    1552    1177    459
1937    1526    964 1407    1069    1104    1118    1505    1626    1711
1320    1002    1582    2106    1161    1119    1348    943 1656    1250
1419    1011    1524    933 2902    2376    1720    1117    2727    2069
1416    2227    1632    2040    2306    1411    1225    2551    1017    3108
2138    1862    3074    1932    2184    2614    2366    706 2735    1500
3308    1368    3982    2699 0  0
];
cu= max(size(d));
if loog ==1;
z =log(d);
else
    z=d;
end

if diff ==1
    w = z;
    for i=1:max(size(w))-1
        s(i) = w(i+1) - w(i);
    end
    z=s;
else
end

l = max(size(z));
```

```
for i=1:num
ee(i) = 0;
end

for i=1:num
yp(i) = z(i);
end
yp(l-1) = z(l-1);
yp(l) = z(l);

for i =1:num+1
yp1(i) = z(i);
end
yp1(l) = z(l);

yp2(1) = z(1);
yp2(2) = z(2);
yp2(3) = z(3);
yp2(4) = z(4);
yp2(5) = z(5);
yp2(6) = z(6);
yp2(7) = z(7);

j = num;
theta = zeros(sz,1);
I = eye(sz);
p = 1000000*I;

while j < l-2

j = j + 1;
for i=1:sz1
    x(i,1)=z(j-co(i));
end
for i =1:ne
    x(i+sz1,1)=ee(j-i);
end
    x(sz1+ne+1,1)=1;

k = p*x/(alpha + x'*p*x);
p = (I-k*x')*p/alpha ;
theta = theta + k*(z(j) - x'*theta);
yp(j) = x'*theta;
ee(j) = z(j) - yp(j);


for i=1:sz1
   x1(i,1)=z(j+1-co(i));
end
    x1(sz1+1,1)=1;
yp1(j+1) = x1'*theta([1:sz1 sz],1);
end

if diff==1&loog==1
    es(1) = w(1);
    for i=1:l
        es(i+1)=w(i)+yp1(i);
    end
    c = exp(es');
else
    if loog==0&diff==1
        es(1) = w(1);
    for i=1:l
        es(i+1)=w(i)+yp1(i);
    end
    c = es;
    end
end
```

```
if loog ==1&diff==0
    c = exp(yp1');

else
    if diff==0&loog==0
        c = yp1;
    end
end
cost = 0;
for i = 1:cu-2
cost = cost + abs(c(i) - d(i));
end

cost;
```

Appendix C: Web Based Java Microworld

Factory

```
public class Factory
{

double dop;
double dopt;
double fio;
double fiot;
double fi;
double fit;
double fpd2;
double fpd2t;
double fpd1;
double fpd1t;
double fb;
double fbt;
double fpr;
double fprt;
double fos;
double fost;
double fed;
double fedt;
double as=0.45;
double B=0.075;
double H=0.25;
double Q;
//double [] dis={8,8,8,8,8,8,8,8,8,8};
 public Factory(double iOrder ,double delay2 ,double delay1 ,
                                double oShip, double back, double expect, double
request, double inventory )
 {
  fio=iOrder;
  fpd2=delay2;
  fpd1=delay1;
  fos=oShip;
  fb=back;
  fed=expect;
  fpr=request;
  fi=inventory;
 }
 public void Exchange()
 {
  fiot=fio;
  fit=fi;
  fpd2t=fpd2;
  fpd1t=fpd1;
  fost=fos;
  fbt=fb;
  fedt=fed;
  fprt=fpr;
  dopt=dop;
  }
//first step
 public double fosMin(double a, double b)
  {
```

```
    double s;
    if(a>b)
       s=b;
     else
       s=a;
    return s;
  }
 public void adOrder(double a)
  {
  dop=a;
  }
 public double Inventory()
  {
   double x=fit+fpd2t-fbt-fiot;
   if(x>0)
     {
      fi=x;
       }
   else
       fi=0;
  return fi;
   }

 public void forwardFpd()
  {
   fpd2=fpd1t;
   fpd1=fprt;
  }
 public double Fos()
  {
   double a= fit+fpd2t;
   double b= fbt+fiot;
   fos=fosMin(a,b);
  return fos;
  }
//second step
  public void adFio()
 {
  fio=dopt;
  }
  public double Backlog()
  {
   double x=fbt+fiot-fit-fpd2t;
  if(x>0)
   fb=x;
  else
   fb=0;
   return fb;
  }
  public double Expect()
  {
   fed=0.25*fiot+0.75*fedt;
   int c=(int)fed;
   if(fed>c)
    fed=c+1;
   else fed=c;
   return fed;
   }
//fourth step
  public double Request()
  {
   Q=17;
System.out.println(""+Q);
   double t=as*(Q-fi+fb)-as*B*(fpd2+fpd1)+fed;

   if(t>0)
    {
     int p=(int)t;
```

```
     if(t>p)
      t=p+1;
     //else
      //t=p;
     fpr=t;
     }
   else fpr=0;
   return fpr;
  }
//fifth step
}


class FactoryModel
{
       public FactoryModel()
       {
              int c;
              }
  public double[] runFactory(double iOrder ,double delay2 ,double delay1 ,
                                                          double oShip, double
back, double expect, double request, double inventory, double distributor)
  {
     double fio=iOrder;
    double  fpd2=delay2;
      double fpd1=delay1;
      double fos=oShip;
      double fb=back;
      double fed=expect;
      double fpr=request;
      double fi=inventory;
      double dr=distributor;
   Factory f= new Factory(fio,fpd2,fpd1,fos,fb,fed,fpr,fi);
   int i;
   double [] Inv=new double[40];
   double [] Back=new double[40];
   double [] Request=new double[40];
   for(i=0;i<1;i++)
  {
   //first step
   f.adOrder(dr);
   f.Exchange();
   double e=f.Inventory();
   f.forwardFpd();
System.out.println("the fit is:"+f.fit);
   double x=f.Fos();
  System.out.println("the fos is:"+f.fos);
   System.out.println("the inventory is:"+f.fi);
  //second step
   f.adFio();
   double y=f.Backlog();
   System.out.println("the backlog order is:"+f.fb);
   double z=f.Expect();
System.out.println("the fiot is:"+f.fiot);
   System.out.println("the fed is:"+z);
 //third step
   Inv[i]=f.fi;
   Back[i]=f.fb;
//fourth step
   double k=f.Request();
//fifth step
   Request[i]=f.fpr;
   System.out.println("the request is:"+Request[i]+"\n");
System.out.println(f.fio+" "+f.fpd2+" "+f.fpd1+" "+f.fos+" "+f.fb+" "+f.fed+"
"+f.fpr+" "+f.fi+"\n");
  }
double[] a={f.fio,f.fpd2,f.fpd1,f.fos,f.fb,f.fed,f.fpr,f.fi};
return a;
```

```
   }}
```

Distributor

```
public class Distributor
{
double wop;
double wopt;
double dio;
double diot;
double di;
double dit;
double dis;
double dist;
double fos;
double fost;
double db;
double dbt;
double dop;
double dopt;
double dos;
double dost;
double ded;
double dedt;
double as=0.45;
double B=0.075;
double H=0.25;
double Q;
 public Distributor(double iOrder ,double incomingS ,double FoutSt ,
                                  double oShip, double back, double expect, double
request, double inventory,double FoutS )
 {
  dio=iOrder;
  dis=incomingS;
  fos=FoutS;
  fost=FoutSt;
  dos=oShip;
  db=back;
  ded=expect;
  dop=request;
  di=inventory;
 }
 public void Exchange()
 {
  diot=dio;
  dit=di;
  dist=dis;
  //fost=fos;
  dost=dos;
  dbt=db;
  dedt=ded;
  dopt=dop;
  wopt=wop;
  }
//first step
 public double dosMin(double a, double b)
  {
    double s;
    if(a>b)
       s=b;
     else
       s=a;
    return s;
  }
 public void adOrder(double a)
```

```
  {
  wop=a;
  }
 public double Inventory()
  {
   double x=dit+dist-dbt-diot;
   if(x>0)
     {
      di=x;
        }
   else
       di=0;
  return di;
   }
 public void forwardDpd()
  {
   dis=fost;
   //fos=dopt;////////
  }
 public double dos()
  {
   double a= dit+dist;
   double b= dbt+diot;
   dos=dosMin(a,b);
  return dos;
  }
//second step
  public void adDio()
 {
  dio=wopt;
  }
  public double Backlog()
  {
   double x=dbt+diot-dit-dist;
  if(x>0)
   db=x;
  else
   db=0;
   return db;
  }
  public double Expect()
  {
   ded=0.25*diot+0.75*dedt;
   int c=(int)ded;
   if(ded>c)
    ded=c+1;
   else ded=c;
   return ded;
   }
//fourth step
  public double Request()
  {
   Q=17;
System.out.println(""+Q);
   double t=as*(Q-di+db)-as*B*(dis+fos)+ded;
   if(t>0)
    {
     int p=(int)t;
     if(t>p)
      t=p+1;
     //else
      //t=p;
     dop=t;
     }
   else dop=0;
   return dop;
  }
//fifth step
```

```
}

class DistributorModel
{
       public DistributorModel()
       {
              int c;
              }
  public double[] runDistributor(double iOrder ,double delay2 ,double delay1 ,
                                                          double oShip, double
back, double expect, double request, double inventory, double wrequest,double
FoutS)
  {
     double dio=iOrder;
    double  dis=delay2;
      double fos=delay1;
      double dost=oShip;
      double db=back;
      double ded=expect;
      double dop=request;
      double di=inventory;
      double dos=FoutS;
      double wh=wrequest;
   Distributor d= new Distributor(dio,dis,fos,dost,db,ded,dop,di,dos);
   int i;
   double [] Inv=new double[40];
   double [] Back=new double[40];
   double [] Request=new double[40];
   for(i=0;i<1;i++)
  {
   //first step
   d.adOrder(wh);
   d.Exchange();
   double e=d.Inventory();
   d.forwardDpd();
System.out.println("the dit is:"+d.dit);
   double x=d.dos();
   System.out.println("the inventory is:"+d.di);
  //second step
   d.adDio();
   double y=d.Backlog();
   System.out.println("the backlog order is:"+d.db);
    System.out.println("the dist is:"+d.dist);
    System.out.println("the dis is:"+d.dis);
    System.out.println("the fost is:"+d.fost);
   double z=d.Expect();
System.out.println("the diot is:"+d.diot);
   System.out.println("the ded is:"+z);
 //third step
   Inv[i]=d.di;
   Back[i]=d.db;
//fourth step
   double k=d.Request();
//fifth step
   Request[i]=d.dop;
   System.out.println("the request is:"+Request[i]+"\n");
System.out.println(d.dio+" "+d.dis+" "+d.fos+" "+d.dos+" "+d.db+" "+d.ded+"
"+d.dop+" "+d.di+"\n");
  }
double[] a={d.dio,d.dis,d.fos,d.dos,d.db,d.ded,d.dop,d.di};
return a;
   }
}
```

Wholsalor

```
public class Wholesalor
{
double rop;
double ropt;
double wio;
double wiot;
double wi;
double wit;
double wis;
double wist;
double dos;
double dost;
double wb;
double wbt;
double wop;
double wopt;
double wos;
double wost;
double wed;
double wedt;
double as=0.45;
double B=0.075;
double H=0.25;
double Q;
 public Wholesalor(double iOrder ,double incomingS ,double DoS ,
                                double oShip, double back, double expect, double
request, double inventory,double DoSt)
 {
  wio=iOrder;
  wis=incomingS;
  dos=DoS;
  dost=DoSt;
  wos=oShip;
  wb=back;
  wed=expect;
  wop=request;
  wi=inventory;
 }
 public void Exchange()
 {
  wiot=wio;
  wit=wi;
  wist=wis;
  //dost=dos;
  wost=wos;
  wbt=wb;
  wedt=wed;
  wopt=wop;
  ropt=rop;
  }
//first step
 public double dosMin(double a, double b)
  {
    double s;
    if(a>b)
       s=b;
     else
       s=a;
    return s;
  }
 public void adOrder(double a)
  {
  rop=a;
```

```
  }
 public double Inventory()
  {
   double x=wit+wist-wbt-wiot;
   if(x>0)
     {
      wi=x;
        }
   else
       wi=0;
  return wi;
   }
 public void forwardDpd()
  {
   wis=dost;
   //fos=dopt;////////
  }
 public double dos()
  {
   double a= wit+wist;
   double b= wbt+wiot;
   wos=dosMin(a,b);
  return wos;
  }
//second step
  public void adDio()
 {
  wio=ropt;
  }
  public double Backlog()
  {
   double x=wbt+wiot-wit-wist;
  if(x>0)
   wb=x;
  else
   wb=0;
   return wb;
  }
  public double Expect()
  {
   wed=0.25*wiot+0.75*wedt;
   int c=(int)wed;
   if(wed>c)
    wed=c+1;
   else wed=c;
   return wed;
   }
//fourth step
  public double Request()
  {
   Q=17;
System.out.println(""+Q);
   double t=as*(Q-wi+wb)-as*B*(wis+dos)+wed;
   if(t>0)
    {
     int p=(int)t;
     if(t>p)
      t=p+1;
     //else
      //t=p;
     wop=t;
     }
   else wop=0;
   return wop;
  }
//fifth step
}
```

```
class WholesalorModel
{
       public WholesalorModel()
       {
              int c;
       }
  public double[] runWholesalor(double iOrder ,double delay2 ,double delay1 ,
                                                          double  oShip,  double
back, double expect, double request, double inventory, double retailer,double
delay1t)
  {
      double wio=iOrder;
      double  wis=delay2;
      double dos=delay1;
      double dost=delay1t;
      double wos=oShip;
      double wb=back;
      double wed=expect;
      double wop=request;
      double wi=inventory;
      double re=retailer;
   Wholesalor d= new Wholesalor(wio,wis,dos,wos,wb,wed,wop,wi,dost);
   int i;
   double [] Inv=new double[40];
   double [] Back=new double[40];
   double [] Request=new double[40];
   for(i=0;i<1;i++)
  {
   //first step
   d.adOrder(re);
   d.Exchange();
   double e=d.Inventory();
   d.forwardDpd();
System.out.println("the wit is:"+d.wit);
   double x=d.dos();
   System.out.println("the inventory is:"+d.wi);
  //second step
   d.adDio();
   double y=d.Backlog();
   System.out.println("the backlog order is:"+d.wb);
   System.out.println("the wis is:"+d.wis);
   System.out.println("the wit is:"+d.wit);
   System.out.println("the wiot is:"+d.wiot);
   double z=d.Expect();
System.out.println("the wiot is:"+d.wiot);
   System.out.println("the wed is:"+z);
 //third step
   Inv[i]=d.wi;
   Back[i]=d.wb;
//fourth step
   double k=d.Request();
//fifth step
   Request[i]=d.wop;
   System.out.println("the w request is:"+Request[i]+"\n");
System.out.println(d.wio+"  "+d.wis+"  "+d.dos+"  "+d.wos+"  "+d.wb+"  "+d.wed+"
"+d.wop+" "+d.wi+"\n");
  }
double[] a={d.wio,d.wis,d.dos,d.wos,d.wb,d.wed,d.wop,d.wi};
return a;
   }
}
```

Main

```
class linkTest1
{
	public static void main(String args[] )
	{
		//for factory
		double iOrder =4;
				double delay2 =4;
				double delay1=4;
			double back=0;
			double expect=4;
			double request=4;
			double inventory=4;
			double b[]= new double[8];
		FactoryModel f=new FactoryModel();
		//link point
		double FoutS=4;
		double FoutSt;
		double drequest=4;
		double diOrder =4;
		double dincomingS =4;
		double dback=0;
		 double dexpect=4;
		 double dinventory=4;
		 DistributorModel t=new DistributorModel();
		   //for link point
		    double wrequest=4;
		    double doShip=4;
		    double doShipt;
		   //for wholesalor
		    double wiOrder=4;
		     double wincomingS=4;
		     double wback=0;
		     double wexpect=4;
		     double winventory=4;
		     double woShip=4;
		     double retailer=8;
		   WholesalorModel h=new WholesalorModel();
	double
orderList[]={8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8
,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8};
		double d[]= new double[8];
		   double w[]=new double[8];
		for(int j=0;j<8;j++)
	{
		//for factory
		   FoutSt=FoutS;
	b=f.runFactory(iOrder,delay2,delay1,FoutS,back,expect,request,inventory
,drequest);//shallow copy
				iOrder=b[0];
				delay2=b[1];
				delay1=b[2];
				FoutS=b[3];
				back=b[4];
				expect=b[5];
				request=b[6];
				inventory=b[7];
				//distributor=orderList[j+1];
				//for distributor
					doShipt=doShip;
	d=t.runDistributor(diOrder,dincomingS,FoutSt,doShip,dback,dexpect,drequ
est,dinventory,wrequest,FoutS);//shallow copy
						diOrder=d[0];
						dincomingS=d[1];
						FoutS=d[2];
						doShip=d[3];
```

```
                                        dback=d[4];
                                        dexpect=d[5];
                                        drequest=d[6];
                                        dinventory=d[7];

                  //forwholesalor
w=h.runWholesalor(wiOrder,wincomingS,doShip,woShip,wback,wexpect,wrequest,winv
entory,retailer,doShipt);
                   wiOrder=w[0];
                   wincomingS=w[1];
                   doShip=w[2];
                   woShip=w[3];
                   wback=w[4];
                   wexpect=w[5];
                   wrequest=w[6];
                   winventory=w[7];
                   retailer=orderList[j+1];
//end
       for(int i=0;i<6;i++)
         {
                 System.out.print("the value is"+b[i] +" ");
                }
     }
}
}
```

Reference

A.D. Papalexopoulos, T.C. Hesterberg, (1990): A regression-based approach to short-term system load forecasting, IEEE Trans. Power System 5(4) 1535-1547.

Ackermans H., Dellaert N., (2005): The Rediscovery of Industrial Dynamics: the contribution of system dynamics to supply chain management in a dynamic and fragmented world, System Dynamics Review, Volume 21 Number 3 Fall, p173 – 189.

A.M.Axoff, (1994): Neural Network Time Series Forecasting of Financial Markets, Wiley.

Atherton D.P., (1982): Nonlinear Control Engineering, Van Nostrand Reinhold, London.

Balle Michael, (1994): Managing with Systems Thinking, McGraw-Hill, London.

Bai-Ling Zhang, Zhao-Yang Dong, (2001): An adaptive neural-wavelet model for short term load forecasting, Electric Power Systems Research 59 121-129.

Baskerville R.L., Wood-Harper A.T., (1996): A critical perspective on action research as a method for information systems research, Journal of Information Technology, Volume11,Number 3.

Beamon, B. M., (1998): Supply chain design and analysis: models and methods. International Journal of Production Economics, 55, 281-294.

Beer, S., (1979): The Heart of Enterprise, John Wiley & Sons, Chichester.

Beer, S., (1994): Decision and Control, John Wiley & Sons, Chichester.

Bohm, D., (1999): On Dialogue, Routledge, London.

Box, G.P., Jenkins, G.M., (1976): Time series analysis: Forecasting and control. Holden-Day, San Francisco.

Blank, S.C, (1991): Chaos in futures market? A nonlinear dynamics; analysis, Journal of Futures Markets, Vol.11, pp.711 – 728.

British Standards Institute, 1997. EN 12777: Logistics – structure, basic terms and definitions in logistics. Draft for public comment, BSi, London.

C.Sidney Burrus, Ramesh A.Gopinath and Haitao Guo, (1998): Introductino to Wavelets and Wavelet Transforms, Upper Saddle River, N.J. Prentice Hall.

Chang, Y., Makastsoris, H., (2001): Supply chain modelling using simulation, International Journal of Simulatin,1.

Checkland Peter, (1981): System thinking, System practise, Wiley, Chichester

Chiang, W.C., Urban, T.L., and Baildridge, G., (1996): A neural network approach to mutual fund net asset value forecasting, Omega: International Journal of Management Science, Vol.24, pp.205-215.

Cox, A., Watson, G., Lonsdale, C., Sanderson, J. (2004): Managing Appropriately in Power Regimes: Relationship and Performance Management in 12 Supply Chain Cases, Supply Chain Management: an International Journal, Vol. 9, No 5, pp357-371.

Crotty, M., (1998): The Foundations of Social Research, Sage Publications, London.

De Geus, A.P., (1988): Planning as Learning, Harvard Business Review, March-April, p 70-74.

Diehl, E. and J.Sterman, (1995): Effects of feedback complexity on dynamic decision making, Organizational Behavior and Human Dcision Processes 62(2), 198-215.

Disney, S. M., and Towill, D. R., (2002): A discrete transfer function model to determine the dynamic stability of a vendor managed inventory supply chain, International Journal of Production Research, 40, 179-204.

D.J.C MacKay, Bayesian non-linear modeling for the energy prediction competition, in: G.Heidbreder(Ed.), Maximum Entropy and Bayesian Methods, Santa Barbara 1993, Kluwer, Dordrecht, 1995.

Donoho.D.L, Johnstone.I.M, (1998): Minimax estimation via wavelet shrinkage, Annals of Statistics, 26(3): 879-921.

Duc, Truong Ton Hien, Kyibg, Huynh Trung, Kim, Yeong-Dae, (2008): A measure of bullwhip effect in supply chains with a mixed autoregressive-moving aaverage demand process, European journal of Operational Research, Vol 187, Issue 1 243(14).

Forrester, J.W., (1961): Industrial Dynamics, Portland, Oregon, Productivity Press.

Fowler, A., (1999): Feedback and Feedforward as systemic frameworks for Operations Control, International Journal of Operations and Production Management, pp. 182-204, MCB University Press.

Francis E.H.Tay, Lijuan Cao, (2006): Application of support vector machines in financial time series forecasting, The International Journal of Management Science, 309-317.

Gao.H.-Y, (1997): Threshold selection in WaveShrink, theory for matlab wavelet toolbox on denoising.

Golten Jack and Verwer Andy, (1991): Control System Design and Simulation, McGraw-Hill.

Gould, Phillip G., Koehler, Anne B., Ord, J.Keith, Snyder, Ralph D., Hyndman, Rob J., Vahid-Araghi, Farshid, (2008): Forecasting Time Series with Multiple Seasonal Patterns, European Journal of Operational Research, Vol 191, Issue 1, 205(16).

Grubbstrom, R. W. and Wikner, J., (1996): Inventory trigger control policies developed in terms of control theory. International Journal of Production Economics, 45, 397-406.

Grudnitski,G., Osburn, L., (1993): Forecasting S&P and gold futures prices: an application of neural networks, Journal of Futures Markets, Vol.13, pp.631-643.

Haykin, S., (1999): Neural networks: a comprehensive foundation, Pretice Hall. Englewood Cliffs, NJ.

Hong-Minh, S.M., Disney, S.M., and Naim, M.M., (2000): The dynamics of emergency transhipment supply chains. International Journal of Physical Distribution and Logistics Management, 30(9), 788-815.

Hsiao-Tien Pao, (2006): Forecasting electricity market pricing using artificial neural networks, Energy Conversion and Management, 907-912.

Jacobs O.L.R., (1993): Introduction to Control Theory, Clarendon Press, Oxford.

Jarmain WE, (1963): Problems in industrial dynamics. Cambridge, MA: MIT press.

J.D.Sterman and E.Mosekilde, (1994): Business Cycles and Long Waves: A Behavioral Disequilibrium Perspective, in Business Cycles: Theory and Empirical Methods, edited by W.Semmler (Kluwer Academic Publisher), pp.13-51.

Jeruchim, Michel C., Philip Balaban, and K.Sam Shanmugan, (2000): Simulation of Communication Systems, Second Edition, New York, Kluwer Academic/Plenum.

Jones, D.T., and Towill, D.R., (1996): New Directions in Logistics Research. Logistics Academic Network Inaugural Workshop Institute of Logistics, UK

Jones, T.C., and Riley, D.W., (1985): Using Inventory for Competitive Advantage through Supply Chain management. International Journal of Physical Distribution and Material management, Vol. 15, Issue 5, pp. 16-26.

Jones, C., (1989): Supply Chain Mangement – the key issues. BPICS Control, Issue October/November, pp.23-27.

Kevin Gurney, (1997): Computers and Symbols versus Nets and Neurons, University College London Press.

Kevin Gurney, (2003): An Introduction to Neural Networks, CRC Press.

Lee, H. L., Padmanabhan, V., Whang, S., (1997): The bullwhip effect in supply chains. Sloan management Review, 1, 93-102.

Lyneis, J.,(2000): System Dynamics for Market Forecasting and Structural Analysis, System Dynamics Review, Volume 16 Issue 1, Pages 3-25.

Li, H., Li, L., (2005): Integrating Systems Concepts into Manufacturing Information Systems, System Dynamics Review Volume 21 Number 3 Fall, 135–147.

Marquez, A.C., Bianchi, Carmine, Gupta, Jatinder, N.D., (2004): Operational and financial effectiveness of e-collaboration tools in supply chain integration. European Jounal of Operational Research, 159(2), 348(16).

Maturana, H., Varela, F.J., (1987): The Tree of Knowledge, New Science Library, Boston, Mass.

Morecroft, J.D.W., (1985): The Feedback View of Business Policy and Strategy, System Dynamics Review, No 1, p 4-19.

Morecroft, J.D.W., and Sterman, J.D., (1994): *Modelling for Learning Organizations*, System Dynamics Series, Productivity Press, Portland.

Morecroft, J.D.W, (2007): Strategic Modelling and Business Dynamics: A Feedback Systems Approach, Wiley.

Mosekilde, E.,(2002): Topics in Nonlinear Dynamics – Applications to Physics, Biology and Economic Systems, World Scientific, Singapour, 2002.

Mosekilde, E., and Larsen, E.R.,(1988): Deterministic chaos in the beer production distribution model. System Dynamics Review, Vol. 4, Issue 1, pp.131-47.

Mosekilde, E., Larsen, E.R., Sterman, J.D., and Thomsen, J.S., (1992): Nonlinear model- interaction in the macroeconomy. Annals of Operations Research Vol. 37, pp.185-215.

Mosekilde, E., Larsen, E.R., Sterman, J.D., (1991): Coping with complexity: deterministic chaos in human decision making behaviour. Beyond Belief: randomness, prediction and explanation in science. Editors J. L. Casti, and A. Karlqvist CRC press.

Naylor, J., Naim, M. and Berry, D., (1999): Leagility: integrating the lean and agile manufacturing paradigms in the total supply chain, International Journal of Production Economics, 62,107-18.

Ortega, M., and Lin, L., (2004): Control theory applications to the production-inventory problem: a review. International Journal of Production Research, 42(11), 2303-2322.

Pidd Michael, (1997), computer simulation in management science, wiley, Chichester

Pinf-Feng Pai, Yen-change Chen, (2006): A Comparative Study of Three Time Series Models in Grain Future Forecasting, International Journal of Management, Vol 23 No.3.

Preece Jenny, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, Torn Carey, (1994): Human-Computer Interaction, Addison-Wesley, Wokingham

Raghuveer M.Rao, Ajit S.Bopardikar, (1998): Wavelet Transforms Introduction to theory and Applications, Addison-Wesley.

Richards R.J., (1993): Solving Problems in Control, Longman Scientific & Technical, Harlow.

Robert Callan, (1999): The Essence of Neural Networks, Prentice Hall Europe.

Robert Fildes, Paul Goodwin, Michael Lawrence, (2005): The design features of forecasting support systems and their effectiveness, Decision Support Systems 2005.

Robert Marsden Whalley, (1999): A study of the contribution of learning Environments to Business Performance, Leeds.

Roberts Edward B., (1978): Managerial Applications of System Dynamics. Cambridge: MIT Press.

Roberts Nancy, David Andersen, Ralph Deal, Michael Garet, William Shaffer, (1983): Introduction to Computer Simulation – A System Dynamics Modelling Approach, Addison-Wesley, Reading Mass.

Robson, C., (2002): Real World Research, Blackwell Publishing.

S.J.Yao, Y.H.Song, L.Z.Zhang, X.Y.Cheng, (2000): Wavelet Transform and Neural Networks for Short-term Electrical Load Forecasting, Energy Conversion & Management, 41 1975-1988.

Senge, P.M., (1984): System Dynamics as an Artefact for the Systems Age, Dynamica, Vol.10, Part II.

Senge, P.M., (1990): The Fifth Discipline: the Theory Of the Learning Organization, John Wiley & Sons.

Senge, P.M., (1996): Systems Thinking, Executive Excellence, January.

Seymour Papert, (1972): Computers in education - some novel approaches: On making a theorem for a child. Proceedings of the ACM annual conference - Volume 1 ACM'72 ; ACM Press1972-08.

Shein EH, (1996): Kurt Lewin's Change Theory in the Field and in the Classroom, Systems Practice, 9(1), 27-47.

Sice, P.V. and French I.G., (1998): Control of Chaos in Dynamic Decision Making, International Journal of Knowledge Transfer, London, 49-55.

Sice, P.V. and French I.G., (2000): Using system dynamics to analyse interactions in duopoly competition. System Dynamics Review, 16(2), 113-33.

Sterman, J. D., (2001): "System dynamics modeling: Tools for learning in a complex world", California management review 43 (1): 8–25.

Sterman, J.D., (2000): Business Dynamics, Systems Thinking and Modeling for a Complex World, McGraw Hill, Boston.

Sterman, J.D., (1989a): Modelling Managerial Behaviour: misperceptions of feedback in a dynamic decision making experiment. Management Science [MCI], Vol. 35, Issue 3.

Sterman, J.D., (1989b): Misperceptions of feedback in dynamic decision making. Organizational Behaviour & Human Decision Making Processes, Vol. 43, pp.301-35

Steven H. Strogatz, (2001): Nonlinear Dynamic and Chaos,

Stevens, G.C. (1989): Integrating the supply chain. International Journal of Physical Distribution & Logistics Management [IPD], Vol. 9, Issue 8.

Sufi M. Nazem, (1988): Applied Time Series Analysis for Business and Ecomomic Forecasting, Marcel Dekker, INC.

Suri, R. and Desiraju, R., (1997): Performance analysis of flexible manufacturing systems with a single discrete material- handling device. International Journal of Flexible Manufacturing Systems, 9, 223-249.

T.Yamakawa, E.Uchino, and T.Samatsu, (1994): Wavelet Neural Network Employing Over-complete Number of Compactly Supported Non-orthogonal Wavelets and Their Applications, In Proc. Of 1994 IEEE International Conference Joint on Neural Networks, Vollume 3, Pages 1391-1396.

Thomsen, J. S., Mosekilde, E. & Sterman, J. D. (1992): Hyperchaotic Phenomena in Dynamic Decision Making. *Journal of Systems Analysis and Modelling Simulation (SAMS),* 9, 137-156.

Terzi, S., &Cavalieri, S.(2004): Simulation in the supply chain context: A survey. Computers in Industry, 53, 3-16.

Van der Zee, D.J., Van der Vorst, J.G.A.J., (2005): A modelling Framework for Supply Chain Simulation: Opportunities for Improved Decision Making. Decision Sciences, 36(1).

Venix, J., (1994): Group Model Building: Facilitating Team Learning Using System Dynamics, Wiley.

Vogl, T. P., J.K. Mangis, A.K. Rigler, W.T. Zink, and D.L. Alkon, (1988): Accelerating the convergence of the backpropagation method, Biological Cybernetics, vol. 59, pp. 257-263.

Wikner, J., (1994): Dynamic Modeling and Analysis of Information Flows in Production Inventory and Supply Chain Systems (Linkoping: Linkoping Institute of Technology).

Wikner, J., Naim, M. M. And Towill, D. R., (1992): The system simplification approach in understanding the dynamic behaviour of a manufacturing supply chain. Journal of Systems Engineering, 2,164-178.

White, A., S., (1999): Management of inventory using control theory. International Journal of Technology management, 17, 847-860.

Winograd, T., Flores, F., (1986): Understanding Computers and Cognition: a New Foundation for Design, Ablex Press, New Jersey.

Wolstenholme, E., (1990): Systems Enquiry: a System Dynamics Approach, Wiley, Chichester.

Xie Yuanchang, Zhang Yunlong, Ye Zhirui, (2007): Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition, Computer-Aided Civil and Infrastructure Engineering, 326-334.

Yusuf,Y.Y., Gunasekaran, A., Adeleye, E. O., Sivayoganathan, K.(2004): Agile supply chain capabilities: Determinants of competitive objectives. European Journal of Operational Research, 159(2), 379(14).

Zhanybai T.Zhusubailiyev, Erik Mosekilde, (2003): Bifurcations and Chaos in Piecewise-smooth Dynamical Systems, Word Scientific Series on Nonlinear Science, Series A – Vol. 44.