# SUPPLY CHAIN MODELING AND ANALYSIS AT ALTERNATIVE LEVELS OF AGGREGATION

by
Güven Demirel
B.S., Industrial Engineering, Boğaziçi University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2008

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis supervisor, Prof. Yaman Barlas for his guidance, support, and patience not only during this research but also throughout my undergraduate and graduate education. I am thankful to him especially for his critical interventions that have prevented me from getting completely lost in the space of the sub-problems of sub-problems I have encountered during this research.

I am grateful to Prof. Ali Rana Atılgan for everything I have learned from him. He has been a role model for me and his guidance has been one of the most important sources of motivation for my future studies. I am glad that he took part in my thesis committee. I would like to thank Assoc. Prof. Taner Bilgiç for taking part in my thesis committee and providing valuable comments for my further research.

I wish to express my deepest gratitude to my parents, Emine and Muammer Demirel and to my sister, Pelin Demirel for their never ending support and affection.

I wish to thank Ezgi Er for her patience, optimism, and emotional support during the last but never ending months of this research. I am indebted to her for entering in my life and making it more beautiful.

I wish to thank all of my friends for being there whenever I needed it. I am thankful to the members of Sesdyn Research Group -Burak Eskici, Emre Demirezen, Emre Keskin, and Fırat İncioğlu in special- for their camaraderie. I am thankful to Malik Çürük for inspiring talks on this research. I account myself fortunate for the friendships of Buket Avcı, Bulut Aslan, Emin Çiftçi, Engin Durmaz, Hande Küçükaydın, İsmail Arı, Kadir E. Kavak, and Osman Can İçöz.

Finally, I am grateful to all members of the Industrial Engineering department for what I have learned from them and what we have done together.

# ABSTRACT


## SUPPLY CHAIN MODELING AND ANALYSIS AT ALTERNATIVE LEVELS OF AGGREGATION

Agent-based modeling (ABM) and system dynamics (SD) are two methodologies commonly used in the modeling of complex systems, which are conventionally placed at the two ends of the aggregation spectrum. Disaggregated ABM facilitates the analysis of the relationship between individual features, individual dynamics, and aggregate dynamics. Aggregated SD captures the aggregate dynamics without considering the individual scale. Its representative capability thus depends on the system considered.

This research asks the general questions about the level of aggregation in the scope of a three-echelon, multiagent supply chain. First, an agent-based model is built, and the effects of individual firms' features (in terms of demand forecasting, order batching, and dynamic pricing) on aggregate dynamics are analyzed. The analysis focuses especially on inventory/order variability and the bullwhip effect, which is the increase in variability as one moves up in a supply chain. It is shown that the bullwhip effect does not exist if none of these features exist in the agent schemata (Fixed S policy). On the other hand, demand forecasting, order batching, and dynamic pricing are shown to be the causes of variability and bullwhip effect in aggregate orders and inventories. It is demonstrated that these factors at the agent level affect aggregate dynamics through individual dynamics, which systematically depend on the supply chain topology and interactions among individuals. Given the ABM findings, an aggregated SD model is built. SD model is shown to capture the dynamics under Fixed S policy perfectly. It captures the general behavioral pattern under demand forecasting, but not under order batching. The variability amplification characteristics of both demand forecasting and order batching are also captured by the SD model. However, SD fails to capture the variability amplification characteristics of dynamic pricing; because SD does not distinguish individuals and thus does not consider the interactions of agents according to their price levels.

# ÖZET

## DEĞİŞİK MODELLEME DÜZEYLERİNDE TEDARİK ZİNCİRİ ANALİZİ

Etmen-temelli modelleme (ETM) ve sistem dinamiği (SD), karmaşık sistemlerin modellemesinde yaygın olarak kullanılan iki yöntemdir ve geleneksel olarak detaylandırma yelpazesinin ayrı uçlarında konumlanırlar. Dağıtma ve detaylandırma esaslı ETM, bireysel özellik ve dinamikler ile makro dinamikler arasındaki ilişkinin analizini mümkün kılar. Birleştirme esaslı SD ise bireysel ölçeği dikkate almaz ve sadece makro dinamikleri tanımlar; böylece temsil etme yetisi ele alınan sisteme bağlı hale gelir.

Bu çalışma, detaylandırma ve analiz düzeyleriyle ilgili genel soruları, üç-basamaklı ve çok-etmenli bir tedarik zinciri kapsamında ele alır. İlk olarak, etmen-temelli bir model kurulmuş ve (talep tahmini, sipariş birleştirme, ve dinamak fiyatlama cinsinden) bireysel firma özelliklerinin makro dinamikler üzerindeki etkileri analiz edilmiştir. Analiz, envanter ve siparişlerdeki değişkenlik ile bu değişkenliğin tedarik zincirinin üst basamaklarına çıktıkça artması anlamına gelen kamçı etkisi üzerine yoğunlaştırılmıştır. Etmen bünyesindeki bu özelliklerin hiçbirinin mevcut olmadığı durumda (Sabit S politikasında), kamçı etkisinin ortaya çıkmadığı gösterilmiştir. Talep tahmini, sipariş birleştirme ve dinamik fiyatlamanın ise makro düzeydeki sipariş ve envanterlerde değişkenliğe ve kamçı etkisine yol açtığı gösterilmiştir. Bireysel düzeydeki bu özelliklerin, makro dinamikleri, tedarik zincirinin topolojisine sistemik olarak bağlı bireysel davranışlar aracılığıyla etkilediği gösterilmiştir. ETM analizinin ardından, birleştirme esaslı bir SD modeli kurulmuştur. SD modelinin, Sabit S politikasındaki dinamikleri mükemmel şekilde temsil ettiği ve bunun ötesinde talep tahmini altındaki davranış kalıbını ürettiği gösterilmiştir. Ancak, sipariş birleştirme durumundaki davranış kalıbı SD tarafından yakalanamamıştır. Talep tahminiyle sipariş birleştirmeye bağlı değişkenlik artışı SD tarafından açıklanabilirken; dinamik fiyatlama durumunda -bireyler ayrıştırılmadığı ve dolayısıyla fiyata bağlı ilişkilenmeler dikkate alınmadığından- bunun mümkün olmadığı gözlenmiştir.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS / ABBREVIATIONS

| | |
|---|---|
| $B_{M_k}(t)$ | Backlog of manufacturer k at time t |
| $B_{R_i}(t)$ | Backlog of retailer i at time t |
| $B_{W_j}(t)$ | Backlog of wholesaler j at time t |
| $D_{M_k}(t)$ | Total demand from manufacturer k at time t |
| $D_{R_i}(t)$ | Total demand from retailer i at time t |
| $D_{W_j}(t)$ | Total demand from wholesaler j at time t |
| $\hat{D}_{FC}$ | Long run average of aggregate final customer demand |
| $\hat{D}_{M_k}(t)$ | Demand forecast of manufacturer k at time t |
| $\hat{D}_{R_i}(t)$ | Demand forecast of retailer i at time t |
| $\hat{D}_{W_j}(t)$ | Demand forecast of wholesaler j at time t |
| $I_{M_k}(t)$ | Inventory on hand of manufacturer k at time t |
| $I_{R_i}(t)$ | Inventory on hand of retailer i at time t |
| $I_{W_j}(t)$ | Inventory on hand of wholesaler j at time t |
| $IAT_{M_k}$ | Inventory adjustment time of manufacturer k |
| $IAT_{R_i}$ | Inventory adjustment time of retailer i |
| $IAT_{W_j}$ | Inventory adjustment time of wholesaler j |
| $IP_{M_k}(t)$ | Inventory position of manufacturer k at time t |
| $IP_{R_i}(t)$ | Inventory position of retailer i at time t |
| $IP_{W_j}(t)$ | Inventory position of wholesaler j at time t |
| $k_{M_k}$ | Lead time inflation parameter of manufacturer k |
| $k_{R_i}$ | Lead time inflation parameter of retailer i |
| $k_{W_j}$ | Lead time inflation parameter of wholesaler j |

| | |
|---|---|
| $N_{FC}$ | Number of final customers |
| $N_M$ | Number of manufacturers |
| $N_R$ | Number of retailers |
| $N_W$ | Number of wholesalers |
| $NI_M(t)$ | Aggregate net manufacturer inventory at time t |
| $NI_R(t)$ | Aggregate net retailer inventory at time t |
| $NI_W(t)$ | Aggregate net wholesaler inventory at time t |
| $NI_{M_k}(t)$ | Net inventory of manufacturer k at time t |
| $NI_{R_i}(t)$ | Net inventory of retailer i at time t |
| $NI_{W_j}(t)$ | Net inventory of wholesaler j at time t |
| $\mathrm{NI}^*_{M_k}(t)$ | Desired inventory of manufacturer k at time t |
| $\mathrm{NI}^*_{R_i}(t)$ | Desired inventory of retailer i at time t |
| $\mathrm{NI}^*_{W_j}(t)$ | Desired inventory of wholesaler j at time t |
| $O_{FC_l}(t)$ | Total order of final customer l at time t |
| $O_{M_k}(t)$ | Production order of manufacturer k at time t |
| $O_{R_i}(t)$ | Order of retailer i at time t |
| $O_{W_j}(t)$ | Order of wholesaler j at time t |
| $O_{FC_l R_i}(t)$ | Order of final customer l from retailer i at time t |
| $O_{R_i W_j}(t)$ | Order of retailer i from wholesaler j to retailer i at time t |
| $O_{W_j M_k}(t)$ | Order of wholesaler j from manufacturer k at time t |
| $P_T(x)$ | Spectral power threshold for sequence x |
| $s_{M_k}(t)$ | Reorder point of manufacturer k at time t |
| $s_{R_i}(t)$ | Reorder point of retailer i at time t |
| $s_{W_j}(t)$ | Reorder point of wholesaler j at time t |
| $S_{M_k}(t)$ | Order-Up-To-Level of manufacturer k at time t |

| | |
|---|---|
| $S_{R_i}(t)$ | Order-Up-To-Level of retailer i at time t |
| $S_{W_j}(t)$ | Order-Up-To-Level of wholesaler j at time t |
| $\mathrm{SL}_{M_k}(t)$ | Supply line of manufacturer k at time t |
| $\mathrm{SL}_{R_i}(t)$ | Supply line of retailer i at time t |
| $\mathrm{SL}_{W_j}(t)$ | Supply line of wholesaler j at time t |
| $\mathrm{SL}^*_{M_k}(t)$ | Desired supply line of manufacturer k at time t |
| $\mathrm{SL}^*_{R_i}(t)$ | Desired supply line of retailer i at time t |
| $\mathrm{SL}^*_{W_j}(t)$ | Desired supply line of wholesaler j at time t |
| $SLAT_{M_k}$ | Supply line adjustment time of manufacturer k |
| $SLAT_{R_i}$ | Supply line adjustment time of retailer i |
| $SLAT_{W_j}$ | Supply line adjustment time of wholesaler j |
| $SM_{M_kW_j}(t)$ | Material sent by manufacturer k to wholesaler j at time t |
| $SM_{W_jR_i}(t)$ | Material sent by wholesaler j to retailer i at time t |
| $SS_{M_k}$ | Safety stock of manufacturer k |
| $SS_{R_i}$ | Safety stock of retailer i |
| $SS_{W_j}$ | Safety stock of wholesaler j |
| $O_{FC}(t)$ | Aggregate final customer order at time t |
| $O_M(t)$ | Aggregate manufacturer order at time t |
| $O_R(t)$ | Aggregate retailer order at time t |
| $O_W(t)$ | Aggregate wholesaler order at time t |
| $P_{M_k}(t)$ | Actual price of manufacturer k at time t |
| $P_{R_i}(t)$ | Actual price of retailer i at time t |
| $P_{W_j}(t)$ | Actual price of wholesaler j at time t |
| $P^R_{M_k}(t)$ | Reference price of manufacturer k at time t |
| $P^R_{R_i}(t)$ | Reference price of retailer i at time t |

| | |
|---|---|
| $P_{W_j}^{R}(t)$ | Reference price of wholesaler j at time t |
| $PF_{M_k}$ | Pricing function of manufacturer k at time t |
| $PF_{R_i}$ | Pricing function of retailer i at time t |
| $PF_{W_j}$ | Pricing function of wholesaler j at time t |
| $PT_{FC_l}(t)$ | Maximum acceptable price for final customer l at time t |
| $PT_{R_i}(t)$ | Maximum acceptable price for retailer i at time t |
| $PT_{W_j}(t)$ | Maximum acceptable price for wholesaler j at time t |
| $\hat{P}_{FC_l}^{R}(t)$ | Price forecast of final customer l at time t |
| $P_{R_i}^{R}(t)$ | Price forecast of retailer i at time t |
| $\hat{P}_{W_j}^{R}(t)$ | Price forecast of wholesaler j at time t |
| $\alpha_{M_k}$ | Exponential smoothing constant of manufacturer k |
| $\alpha_{R_i}$ | Exponential smoothing constant of retailer i |
| $\alpha_{W_j}$ | Exponential smoothing constant of wholesaler j |
| $\tau_{MW}$ | Lead time of material transit from manufacturer to wholesaler |
| $\tau_{WR}$ | Lead time of material transit from wholesaler to retailer |
| $\tau_{P}$ | Lead time of production for manufacturers |

| | |
|---|---|
| ABM | Agent-based Modeling |
| A&A | Anchor-And-Adjust |
| Fixed S | Fixed Order-Up-To-Level |
| IID | Identically and Independently Distributed |
| PSD | Power Spectral Density |
| SD | System Dynamics |
| (s,S) | Reorder Pont – Order-Up-To-Level |
| Variable S | Variable Order-Up-To-Level |

# 1. INTRODUCTION

The accumulation of scientific knowledge and the development of high performance computers have enabled the emergence of a new way of doing science, which falls under the complexity science heading. Nobel physicist Philip W. Anderson (1972) defines complexity as “more is different” in a nut shell. This motto embodies the non-reductionist philosophy of science that the system as a whole cannot be analyzed or understood by simply decomposing it into its components and taking them separately in isolation.

Modeling of complex systems, mostly using computational techniques, has been an important line in this research. Several simulation methodologies have been applied to study complex economic, social, ecological, managerial, and demographic -or so called socioeconomic- systems. Although agent-based modeling is perceived to be the natural medium of complex systems modeling; system dynamics is another important methodology to model and analyze complex socioeconomic systems.

There have been studies that compare and also integrate system dynamics and agent-based modeling in the last decade (Akkermans, 2001; Demirel, 2006; Parunak *et. al.*, 1998; Rahmandad, 2004; Schieritz and Grossler, 2003; Scholl, 2001a,b). One of the questions that have been put forth in these studies is beyond the “agent-based vs. system dynamics” dichotomy and targets the very basics of the systems science. The issue is about selecting the level of aggregation at the modeling phase to answer questions like: “How much does the level of aggregation affect the dynamics of the model? To what extent can an aggregated model represent the dynamics generated by a disaggregated model with more detail? Can some conclusions be generalized?” Although the level of aggregation in model building is a problem of degree and is not strictly dependent on the modeling methodology; it has been significantly higher in system dynamics models when compared to agent-based models.

Another research question is about the relationship between individual decision making processes, individual level dynamics and the aggregate system behavior. Although aggregated system dynamics models are analyzed at aggregate variables level; there are basically two different levels of analysis for agent-based models: (disaggregate) individual agent level and (aggregate) system level. This research analyzes how aggregate dynamics are formed by the individual dynamics and the interactions among individual agents, and also how aggregate dynamics are affected from individual agents' features. It is important to note that the relationship between individual agents' features, interactions among individuals, individual agent behavior and the system behavior are important questions of the complexity science regarding the sources of the emergence phenomenon.

This research asks the general questions above in the scope of a generic three-echelon multiagent supply chain. Inventory management policies of firms are modeled and inventory/order dynamics are analyzed with a special focus on inventory/order oscillations and the bullwhip effect. Bullwhip effect is the increase in demand/inventory variability and oscillations as one moves up in a supply chain. As far as supply chain dynamics are concerned, contributions of several factors to inventory/order variability and to the bullwhip effect are identified. These structural factors consist of demand forecasting, order batching, and dynamic pricing. The interplays between agent decision making processes, agent dynamics, supply chain topology -which regulates the interactions among individuals-, and aggregate dynamics have been investigated in the analysis of these factors. The representative power of the aggregated system dynamics model is examined.

The outline of the thesis is as follows. Section 2 reviews the modeling methodologies and the supply chain management literature. In Section 3, problem definition and methodology are explained. In Section 4, model structures are introduced. Section 5 analyses the agent-based model dynamics. In Section 6, aggregate system dynamics models of different ordering policies are built and resulting dynamics are compared with the aggregate dynamics of the disaggregated model. In Section 7, general conclusions are drawn and future research questions are proposed.

# 2. LITERATURE REVIEW

Complex systems research has been constructed and is being developed in a multitude of disciplines ranging from physics, mathematics, computer science, and engineering sciences to economy, sociology, ecology, and information science. System, complexity, emergence, adaptiveness, and nonlinearity concepts are the most important concepts to understand complex systems in this wide spectrum. Although system and nonlinearity terms have conventional definitions; complexity, adaptiveness and emergence are a matter of degree in general and it is hard to develop formal definitions and generic quantitative measures for these concepts.

As far as complexity is concerned, it is hard to make a clear cut distinction between simple and complex systems; and it is better to define complexity as “a partial ordering, rather than a quantitative measure” (Gershenson and Heylingen, 2004). Several measures of complexity have been proposed in the literature for different contexts (Edmons, 2000); however there is not a universal measure of complexity. It can still be stated that the complexity increases with the number of distinct components, the number of connections between them, the complexities of components, the complexities of the connections, the existence of delays and nonlinearities, and the richness of the feedback structure.

Adaptiveness, in nontechnical terms, is the system capability to adjust itself to get suited to the changes in the environment. Since, there are numerous adaptation possibilities, it is not viable in general to measure and compare alternative structures in terms of adaptiveness. Miller and Page (2007) propose to generalize equivalence classes in order to classify adaptiveness characteristics.

“Emergence” concept is used to point to the relationship between the components and the whole system. Macro-level properties are said to emerge from the interactions between the components. Emergent properties display identifiable characteristics and are insensitive to non-dramatic modifications at individual level. A most intuitive example of emergent phenomenon is the central limit theorem, which basically states that a Normal distribution emerges as the sample size increases, irrespective of the initial distribution

(Miller and Page, 2007). To give some more examples, temperature is a macro-level phenomenon emerging from the movements of molecules and bullwhip effect is an emergent system level phenomenon in supply chains emerging from the interactions of firms.

According to Surana *et. al.* (2005), supply chains are complex adaptive systems with a highly interconnected network of organizations including high numbers of entities, processes, interactions, and delays. The adaptiveness of supply chains come from their ability to adapt to the changes in the market environment. Inventory management in supply chains is a hard task due to its complex structure. A ubiquitous emergent phenomenon in supply chains is the bullwhip effect, which is the amplification of inventory/order oscillations and variability as one moves up in the chain (Sterman, 2000; Chen *et. al.*, 2000). Occurring inventory oscillations incur additional costs to the supply chain in terms of excessive inventories, poor customer service, and inefficient capacity utilization (Gündüz, 2003).

There have been several studies that detect the causes of the bullwhip effect and develop managerial insights in order to improve system performance in supply chains. According to Lee *et. al.* (1997), fundamental sources of the bullwhip effect are demand forecasting, order batching, lead times, rationing game, and price variations. Chen *et. al.* (2000) analytically shows and discusses how these factors contribute to the bullwhip effect in a two-echelon supply chain with one agent at each level. Gündüz (2003) demonstrates that demand forecasting and delays are the most important sources of the bullwhip effect as a result of the analysis based on a three-echelon system dynamics model. As far as system performance improvement is concerned, Chen *et. al.* (2000) proposes to use smoother demand forecasts in order to reduce the oscillations. Centralizing or sharing the demand and forecast information through the supply chain has also been shown to be a most effective strategy to deaden the oscillations for different supply chain structures, ordering policies, demand patterns, and information sharing types (Gündüz, 2003; Lee and Whang , 1998; Chen *et. al.*, 2000; Xu *et. al.*, 2001; Gavirneni *et. al.*, 1999).

System dynamics and agent-based modeling are among the major simulation tools to model and analyze supply chain dynamics. The history of system dynamics goes back to the early works of Jay W. Forrester in 1950's. System dynamics theory has been structured by application of the ideas from control engineering, cybernetics, and general systems theory to the field of socioeconomic systems. System dynamics is based on the idea of feedback loops. The structure of system dynamics models is formed by a set of positive and negative feedback loops. An increase in one factor may cause its further increase (positive feedback loops) or reversely it can be balanced by the circular relationships (negative feedback loops). The overall behavior is determined according to the strengths of these positive and negative feedback loops.

System dynamics conceptualizes the systems in terms of accumulations (stocks) and their rates (flows) to construct the basis of feedback loops. Stock-flow diagrams are representations of ordinary differential or difference equation systems. Stocks are the states of the system, which can be thought as the snapshots of the system. The flows are the rates of changes of stocks. Since system dynamics models include highly nonlinear equation formulations, they are numerically simulated. Vensim, Stella, Powersim, and Dynamo are simulation platforms specifically designed for the construction, simulation, and analysis of system dynamics models.

Supply chain management and modeling in the field of system dynamics dates almost back to the foundation of the field. Jay W. Forrester constructed a system dynamics model of a production-distribution system in 1961 and discussed several issues regarding supply chain management in Industrial Dynamics (Forrester, 1961). There have been numerous applications of system dynamics modeling to supply chain systems since that date. For taxonomy of the supply chain research and specific examples in the system dynamics field, see Angerhofer and Angelides (2000).

On the other hand, agent-based modeling has been a major tool to build simulation models of complex adaptive systems since 1970s. Agent-based modeling has been applied to social problems in different fields such as sociology, demography, economics, finance, political science, and ecology. Agent-based modeling has a bottom-up modeling perspective which focuses on the actions of heterogeneous individuals. The system behavior emerges from the local dynamics of individuals.

There are some general characteristics of agent-based models: there are multiple -possibly heterogeneous- agents in the system which interact with one and other and their localized environment; agents are autonomous in their decisions and actions, there is no global control over the system; and individual agents act according to simple sets of rules (Gilbert, 1995). A further property that is observed in many contexts is the adaptiveness of agents.

Software agents are used to model the individuals in the real system. Agent technology was first developed in the artificial intelligence (AI) field and later applied to the area of social simulation in the form of agent-based modeling. The most general definition of agency is the capability to perceive, reason, and act. According to Wooldridge and Jennings (1995), the defining characteristics of agents as software-based computer systems are autonomy, social ability, reactivity, and pro-activeness. Autonomy is the ability to operate without control of other agents or humans. Social ability is the capability to interact with the other agents through agent-communication languages. Reactivity is the ability to perceive the environment and react to it. Pro-activeness is the ability to exhibit goal-directed behavior.

Doran (2006) discusses that the following decisions should be made for the design of an agent in agent-based modeling. First, it should be decided whether the model will be a generic model or a specific model. Second, real actors in the system should be identified and a decision regarding which of these will be represented in the model as agents should be given. Third, in parallel to the first two decisions the level of model abstraction should be determined. Fourth, cognition and structure (or schemata) to be built into the agents – with which beliefs and cognitive capabilities - should be designed. Fifth, a decision regarding the choice of agent architecture should be made. The most common among these architectures is production systems. For other architectures like predictive planner and BDI agents, see Doran (2006). Production systems have three components: a set of rules, a working memory, and a rule interpreter. Each rule has two parts: condition part and action part. The condition part specifies when the rule will be fired; and action part specifies what happens when the rule is fired. Rule interpreter learns the state of the system by looking at the working memory and specifies which rule will be fired by checking the condition parts of all rules. Most agent-based models are constructed using a combination of object orientation and production systems (Gilbert and Troitzsch, 2003). Once these design decisions are made, the next step is the programming of agents. There are several simulation platforms and toolkits designed for agent-based modeling: StarLogo, NetLogo, Swarm, Repast, AnyLogic, MASON, CORMAS, SDML, Sim_Agent, Echo, and XRaptor. NetLogo and AnyLogic provide tools for system dynamics modeling also. General purpose programming languages like Java, C++, and Matlab are also widely used for agent-based modeling.

There have been several studies that use multiagent systems in the management of supply chains. Most of the research in this area focus on problem-solving where multiagent systems serves as a new information technology for better management of supply chains. Moyaux *et. al*. (2006) gives a list of multiagent systems applications of supply chain management. For further information on the application of multiagent systems to supply chain management, design, and coordination; see Moyaux *et. al.* (2006) and Panti *et. al.* (2005).

There have been several studies that compare –or integrate- system dynamics and agent-based modeling (Akkermans, 2001; Demirel, 2006; Parunak *et. al.*, 1998; Rahmandad, 2004; Schieritz and Grossler, 2003; Scholl, 2001a,b). Based on this type of research, some general similarities and differences regarding the methodologies, designs, and computational performances have been identified.

First difference is related to the focus of analysis. System dynamics is more policy oriented which is called confirmatory analysis. Agent-based modeling rather puts emphasis on exploratory analysis seeking only to explore the simple rules how agents behave that will form the emergent system behavior (Phelan, 1999; Scholl, 2001a).

Second, two approaches differ in the level of aggregation they use (Parunak *et. al.*, 1998). In general, system dynamics models are highly aggregated models. The modeling and analysis unit is the observable variables at system level. It is assumed that the heterogeneity among individuals does not affect the system behavior significantly. On the other hand, agent-based modeling focuses on the level of individuals; the source of emergent behavior is the nonlinear interactions between individuals. It has been shown that these nonlinear interactions may result in heterogeneous behavior of agents and aggregate behavior may be affected by this heterogeneity; contrary to the assumption of aggregated modeling (Demirel, 2006). However, system dynamics and agent-based modeling can also be used to construct models at different levels of aggregation (Rahmandad, 2004). For example, system dynamics have been used to program agents where the decision-making processes of individual agents are seen as systems themselves (Akkermans, 2001; Schieritz and Grossler, 2003). Therefore, the level of aggregation problem is rather a general discussion; where most system dynamics models are closer to the aggregated end of spectrum and agent-based models to the disaggregated end.

Third, there is a difference in the entities -observables or individuals- they focus their attention on (Parunak *et. al.*, 1998). System dynamics focuses attention on the relationships between observables, which are measurable characteristics. These observables may belong to individuals –as in the modeling of human decision making using system dynamics- or the system as a whole. On the other hand, agent-based modeling focuses on the interactions between individuals and observables emerge from the dynamics of individuals themselves.

Fourth, agent-based modeling has the capability of modeling adaptive agents using the techniques of machine learning. System dynamics models are structurally fixed; however modeling of learning using system dynamics through the idea of feedback is very common (Scholl, 2001a).

Fifth, validation in system dynamics modeling is done through system level structural and behavioral validation tests (Barlas, 1996). However, validation in agent-based modeling requires an additional level: validation of behavioral rules of individual agents and also their corresponding behavior (Parunak *et. al.*, 1998). The lack of unit consistency concept in agent-based modeling is seen as a problem for validation of these models (Rahmandad, 2004).

Sixth, system dynamics and agent-based models differ in their requirements of computational resources, computation time and the complicatedness of model building process. Computational resources required for system dynamics simulation are limited and easily satisfied. However, computation time and resources required for agent-based modeling are high; and these requirements increase significantly as the number and complexity of agents increase.

There have been several comparative studies regarding the capabilities of agent-based and system dynamics modeling based on specific problems (Demirel, 2006; Parunak *et. al.,* 1998; Rahmandad, 2004; Wilson, 1998). The conclusions regarding the capabilities of agent-based and system dynamics modeling are mostly specific to the systems investigated. However, there is evidence that as the behavioral complexity –regarding the decision making processes- of individuals increases; as the network of relationships between individuals differ from a completely randomized network; and as the heterogeneity of individuals increases, aggregated system dynamics models in general cannot capture emergent system behavior.

As far as supply chain dynamics are concerned, there have been studies that use system dynamics and agent-based modeling by integrating these methodologies and applying to supply chain modeling. Schieritz and Grossler (2003) formulate inventory management decisions of firm agents by a system dynamics model, and develop an agent-based supply network connecting these internal models. Schieritz and Grossler (2003) show that the use of relationship based selection stabilizes the network and leads to construction of long-term relationships between agents; which are in agreement with the conclusions of both Akkermans(2001) and Demirel(2006).

Akkermans's (2001) similar research develops an agent-based supply chain model in a system dynamics simulation environment using several ideas from agent-based modeling. The schemata of agents are constructed by system dynamics models which are very similar to those of Schieritz and Grossler (2003). Each agent holds mental models of its potential customers and suppliers based on the performance of them. Akkermans (2001) shows that stable supply networks emerge and giving more weight to short term is more advantageous for firms.

# 3. PROBLEM DEFINITION AND METHODOLOGY

Determining the aggregation level is one of the most important decisions in the model building phase. Aggregation level affects the representative capability of a model. This issue brings forward important questions like "How much does the level of aggregation affect the dynamics of the model? To what extent can an aggregated model represent the dynamics generated by a more detailed disaggregated model?" One purpose of this research is to determine the representation capability of an aggregated model by comparing it with a more detailed agent-based model in the scope of supply chain dynamics.

A second research question is about the relationship between individual decision making processes, individual agent dynamics, interactions among individuals, and aggregate system behavior. It is important to note that the relationships between individual agents' features, agent dynamics, interactions among individuals, and the system behavior are important questions of the complexity science regarding the sources of the emergence phenomenon. Although aggregated system dynamics models are analyzed at aggregate variables level; there can be two different levels of analysis for agent-based models: (disaggregate) individual agent level and (aggregate) system level. This research analyzes the micro-macro data relationship and the sources of emergent phenomenon in the scope of the agent-based supply chain models.

As far as supply chain dynamics are concerned, this research puts a special emphasis on the emergence of inventory and order oscillations, and the bullwhip effect. Inventory/order dynamics of the agent-based model are analyzed under different agent features. The systematic relationship between the individual features and the aggregate system behavior is inquired by including and analyzing the effects of features like demand forecasting, order batching, and adaptive pricing. The ordering policies considered include Fixed and Variable Order-Up-To-Level policies, Reorder Point – Order-Up-To-Level policy, and Anchor-And-Adjust policy.

# 4. SUPPLY CHAIN MODEL STRUCTURE

In this research, a generic model of a three-echelon multiagent manufacturing supply chain is built. Figure 4.1 shows the user interface of the agent-based model. The chain is composed of retailers, wholesalers, and manufacturers. There are $N_{FC}$ final customers, $N_R$ retailers, $N_W$ wholesalers, and $N_M$ manufacturers in the chain. There are two types of flows in the system: (i) information flow, which is the transmission of order information in the upward direction, and (ii) material flow as a response, which is the basically the dispatching of goods to meet incoming orders. Figure 4.2 illustrates the supply chain agents and the flows in the supply chain.

Supply chain is triggered by final customers' demand. Inventory of retailers are increased by the arrival of products sent by wholesalers, and decreased by sales to final customers. Inventory of wholesalers are increased by the arrival of products sent by manufacturers, and decreased by sales to retailers. Inventory of manufacturers are increased by the completion of its own production orders, and decreased by sales to wholesalers. Unmet demand is not lost; but its record is kept as backlog, and backlogged demand is met whenever there is sufficient inventory. There are material delays between retailers and wholesalers, wholesalers and manufacturers, and within the manufacturing process; respectively with lead times $\tau_{WR}$, $\tau_{MW}$, and $\tau_P$. $\tau_{WR}$, $\tau_{MW}$, and $\tau_P$ are set as three, three, and five respectively.

Irrespective of ordering policies, the general structure of the model is fixed. The common structure is explained below for each echelon.

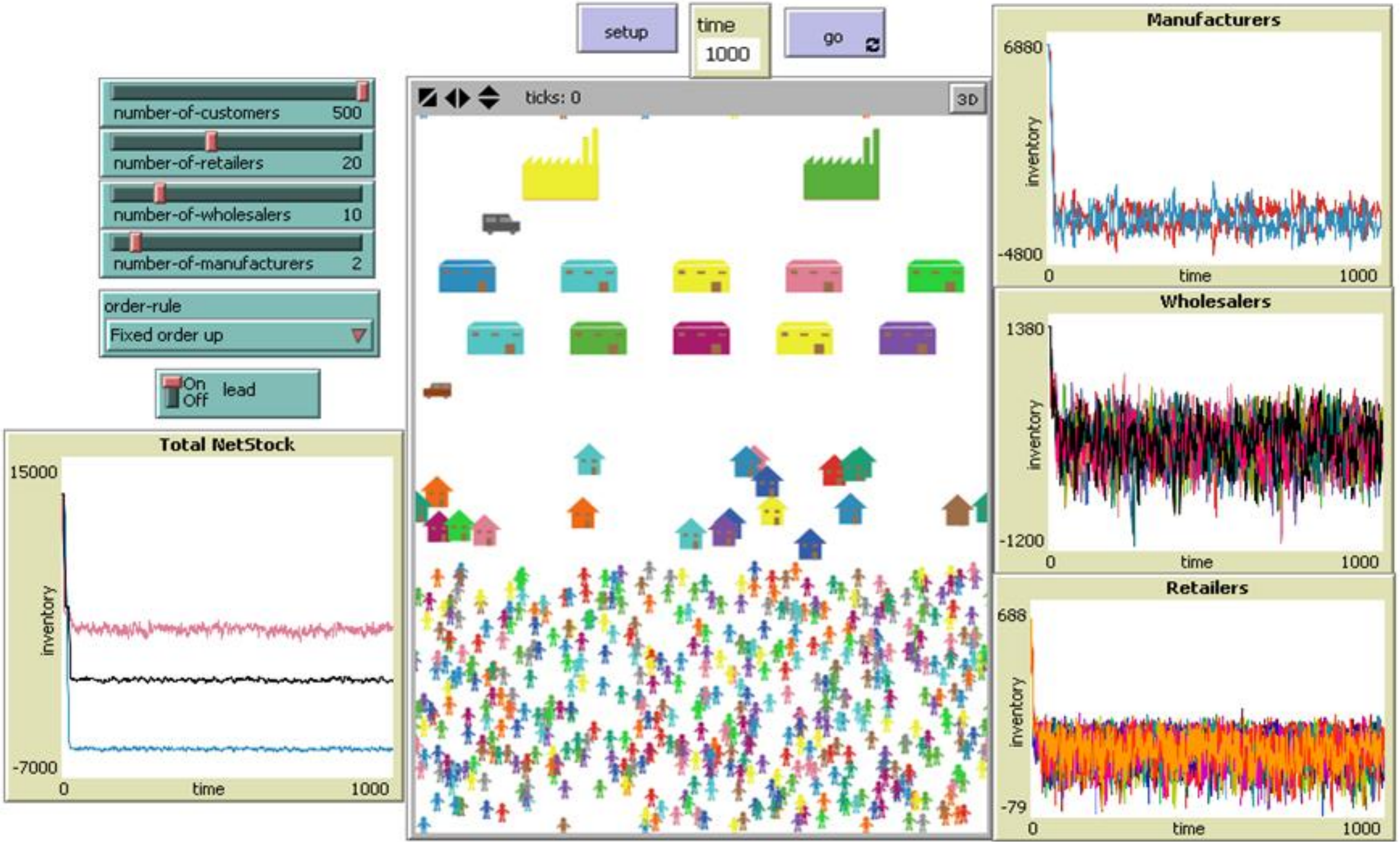


Figure 4.1. Agent-Based Supply Chain Model User Interface

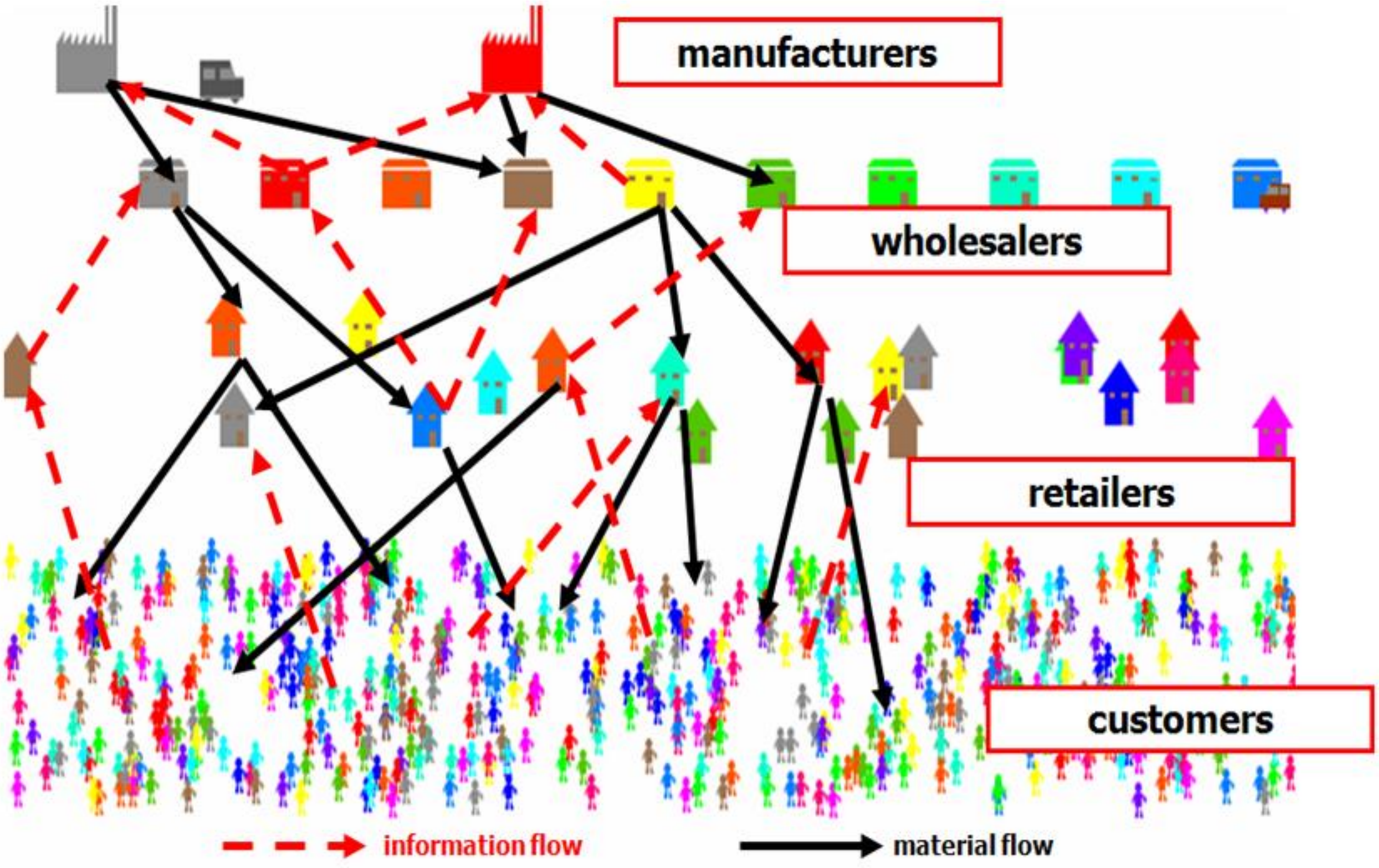


Figure 4.2. Supply Chain Agents and Flows

## 4.1. Supply Chain Structure

Final customer demand is independent of supply chain dynamics. A final customer agent orders $O_{FC_i}$ items from a selected supplier. $O_{FC_i}$ values are iid Uniform(0,5) . In the basic scenarios, supplier selection is made in a totally random fashion. However, in other scenarios supplier selection can be made by looking at some rationality criterion, like supplier inventory position and supplier price. The complete agent-based model is given in Appendix A.

Retailer inventory is increased by the arrival of products sent by wholesalers, and decreased by sales to final customers. Sales may take two forms: meeting incoming demand of the current period or meeting backlog. Equation 4.1 gives inventory balance equation for retailers.

$$NI_{R_i}(t) = NI_{R_i}(t-1) + \sum_{j=1}^{N_W} SM_{W_j R_i}(t-\tau_{WR}) - D_{R_i}(t), \text{ for } i = 1,2,...,N_R \tag{4.1}$$

$NI_{R_i}(t)$ is the net inventory of retailer i at time t. Agents are assumed to satisfy maximum possible amount of incoming demand. Thus, they cannot have inventory and backlog at the same time:

$$I_{R_i}(t) = \max(0, NI_{R_i}(t)\ ) \text{ and } B_{R_i}(t) = \max(0, -NI_{R_i}(t)\ ), \text{ for } i = 1,2,...,N_R \tag{4.2}$$

$D_{R_i}(t)$ is the sum of goods demanded from retailer i, at time t. $O_{FC_l R_i}(t)$ is the order of final customer l from retailer i at time t . Then, equation for $D_{R_i}(t)$ is as follows:

$$D_{R_i}(t) = \sum_{l=1}^{N_C} O_{FC_l R_i}(t), \text{ for } i = 1,2,...,N_R \tag{4.3}$$

$SM_{W_j R_i}(t-\tau_{WR})$ is the material sent by wholesaler j, at time $(t-\tau_{WR})$; which is received by retailer i, exactly $\tau_{WR}$ days after that shipment. Equation 4.4 shows the relationship between the order of retailer i and goods sent by wholesaler j, both at time $(t-\tau_{WR})$.

$$SM_{W_jR_i}(t-\tau_{WR}) = \min(O_{R_iW_j}(t-\tau_{WR}), I_{W_j}(t-\tau_{WR})), \text{ for } i=1,2,...,N_R \tag{4.4}$$

$O_{R_iW_j}(t-\tau_{WR})$ is the order of retailer i from wholesaler j at time $(t-\tau_{WR})$. Notice that if $SM_{W_jR_i}(t)$ is less than $O_{R_iW_j}(t)$, then the difference between the two is kept as backlog and processed in the following step by wholesaler j as new demand information. Backlogged demand is processed before regular demand at each time step. $I_{W_j}(t-\tau_{WR})$ is the inventory of wholesaler j at time $(t-\tau_{WR})$. If wholesaler j has sufficient inventory at that point, it satisfies the whole demand. Otherwise, it satisfies as much as possible. Note that the sequence at which the order of retailer i is processed by wholesaler j is important. Suppliers process demand information in a random sequence with FIFO principle.

Structure of wholesalers is similar to that of retailers. Wholesaler inventory is increased by the arrival of products sent by manufacturers, and decreased by sales to retailers. Sales may take two forms: meeting incoming demand of current period or satisfying backlog. Inventory balance equation for wholesalers is:

$$NI_{W_j}(t) = NI_{W_j}(t-1) + \sum_{k=1}^{N_M} SM_{M_kW_j}(t-\tau_{MW}) - D_{W_j}(t), \text{ for } j=1,2,...,N_W \tag{4.5}$$

$NI_{W_j}(t)$ is the net inventory of wholesaler j at time t. Wholesalers cannot have inventory and backlog at the same time. The relationship between wholesaler inventory and backlog is:

$$I_{W_j}(t) = \max(0, NI_{W_j}(t)) \text{ and } B_{W_j}(t) = \max(0, -NI_{W_j}(t)), \text{ for } j=1,2,...,N_W \tag{4.6}$$

$D_{W_j}(t)$ is the sum of goods demanded from wholesaler j at time t. $O_{R_iW_j}(t)$ is the order of retailer i from wholesaler j at time t . Then, $D_{W_j}(t)$ is:

$$D_{W_j}(t) = \sum_{i=1}^{N_R} O_{R_iW_j}(t), \text{ for } j = 1, 2, ..., N_W \tag{4.7}$$

$SM_{M_kW_j}(t-\tau_{MW})$ is the material sent by manufacturer k, at time $(t-\tau_{MW})$; which is received by wholesaler j, exactly $\tau_{MW}$ days after that shipment. Equation 4.8 shows the relationship between the order of wholesaler j and goods sent by manufacturer k, both at time $(t-\tau_{MW})$.

$$SM_{M_kW_j}(t-\tau_{MW}) = \min(O_{W_jM_k}(t-\tau_{MW}), I_{M_k}(t-\tau_{MW})), \text{ for } j = 1, 2, ..., N_W \tag{4.8}$$

$O_{W_jM_k}(t-\tau_{MW})$ is the order of wholesaler j from manufacturer k at time ($t$-$\tau_{MW}$). Notice that if $SM_{M_kW_j}(t)$ is less than $O_{W_jM_k}(t)$, then the difference between the two is kept as backlog and processed in the following step by manufacturer k as new demand information. Backlogged demand is processed before regular demand at each time step. $I_{M_k}(t-\tau_{MW})$ is the inventory of manufacturer k at time ($t$-$\tau_{MW}$). If manufacturer k has sufficient inventory at that point, it satisfies the whole demand. Otherwise, it satisfies as much as possible.

Manufacturer structure is similar, with the difference that manufacturers make production themselves ($O_{M_k}(t-\tau_P)$) instead of giving orders to other firms. Manufacturer inventory is increased by the completion of production, and decreased by sales to wholesalers. Inventory balance equation for manufacturers is:

$$NI_{M_k}(t) = NI_{M_k}(t-1) + O_{M_k}(t-\tau_P) - D_{M_k}(t), \text{ for } k = 1, 2, ..., N_M \tag{4.9}$$

$NI_{M_k}(t)$ is the net inventory of manufacturer k at time t. $O_{M_k}(t-\tau_P)$ is the production order of manufacturer k, at time $(t-\tau_P)$; which is completed in $\tau_P$ days.

Production capacity is assumed to be unlimited. Manufacturers cannot have inventory and backlog at the same time. The relationship between manufacturer inventory and backlog is:

$$I_{M_k}(t) = \max(0, NI_{M_k}(t)) \text{ and } B_{M_k}(t) = \max(0, -NI_{M_k}(t)), \text{ for } k = 1, 2, ..., N_M \quad (4.10)$$

$D_{M_k}(t)$ is the sum of goods demanded from manufacturer k, at time t. $O_{W_j M_k}(t)$ is the order of wholesaler j from manufacturer k at time t . Then, $D_{M_k}(t)$ is:

$$D_{M_k}(t) = \sum_{j=1}^{N_W} O_{W_j M_k}(t), \text{ for } k = 1, 2, ..., N_M \quad (4.11)$$

## 4.2. Ordering Policies

All agents in a given echelon are assumed to use the same ordering policy. Order-Up-To-Level policies with both Fixed and Variable Order-Up-To-Levels (Fixed S and Variable S), Reorder Point - Order-Up-To-Level Policy (s,S), and Anchor-And-Adjust (A&A) policy are considered. All are continuous review policies, where inventory level is checked and order is given, if necessary, at each time point. Fixed S policy is different from other three in terms of the utilization of demand forecast information. In Fixed S policy, it is assumed that a reasonable Order-Up-To-Level is available from past data and demand information is not used, namely S is not updated. Other policies make use of the demand information through forecasting in order to update policy parameters.

### 4.2.1. Fixed Order-Up-To-Level (Fixed S) Policy

Fixed S policy is the base stock policy, where a firm gives orders at each time point in an amount equal to the discrepancy between the current inventory position and the Order-Up-To-Level. The Order-Up-To-Level is predetermined and is not updated according to the incoming demand. Although analytical formulas for optimal Order-Up-To-Levels exist; this is beyond the scope and aim of this research. Hence, we set the Order-Up-To-Level by trial and error. Its value is determined by considering the long-run average of final customer demand in such a way that only negligible backlogs occur, but inventories stay at reasonable levels. Order-Up-To-Level for each echelon is given below:

$$S_{R_i} = k_1 * (\hat{D}_{FC} / N_R), \text{for i} = 1, 2, ..., N_R \tag{4.12}$$

$$S_{W_j} = k_2 * (\hat{D}_{FC} / N_W), \text{for j} = 1, 2, ..., N_W \tag{4.13}$$

$$S_{M_k} = k_3 * (\hat{D}_{FC} / N_M), \text{for k} = 1, 2, ..., N_M \tag{4.14}$$

$S_{R_i}$, $S_{W_j}$, and $S_{M_k}$ are Order-Up-To-Levels for retailer i, wholesaler j, and manufacturer k respectively. $\hat{D}_{FC}$ is the long-run average of final customer demand. The parameters $k_1$, $k_2$, and $k_3$ are set as five, 11, and 19 respectively, by trial-and-error.

In Fixed S policy, retailers give order decisions as follows:

$$O_{R_i}(t) = \begin{cases} S_{R_i} - IP_{R_i}(t), \text{ if } IP_{R_i}(\text{t}) < S_{R_i} \\ 0 \qquad\qquad\qquad , \text{otherwise} \end{cases} \text{, for } i = 1, 2, ..., N_R \tag{4.15}$$

$O_{R_i}(t)$ is the order given by retailer i at time t. Retailer agent selects one of the wholesaler agents as its supplier. $O_{R_iW_j}(t)$ is the order of retailer i from wholesaler j at time t. $O_{R_iW_j}(t) = O_{R_i}(t)$ for one and only one wholesaler, and $O_{R_iW_j}(t) = 0$ for the remaining.

$IP_{R_i}(t)$ is the inventory position of retailer i at time t. Inventory position is net inventory plus supply line ($SL_{R_i}(t)$). Inventory position and supply line for retailers are respectively:

$$IP_{R_i}(t) = NI_{R_i}(t) + SL_{R_i}(t), \text{ for } i = 1, 2, ..., N_R \tag{4.16}$$

$$SL_{R_i}(t) = SL_{R_i}(t-1) + O_{R_i}(t-1) + \sum_{j=1}^{N_W} SM_{W_jR_i}(t - \tau_{WR}), \text{ for } i = 1, 2, ..., N_R \tag{4.17}$$

$SL_{R_i}(t)$ increases with the order at the previous time step and decreases with the arrival of products sent by wholesalers at time $(t-\tau_{WR})$.

Wholesalers give order decisions similarly:

$$O_{W_j}(t) = \begin{cases} S_{W_j} - IP_{W_j}(t), \text{ if } IP_{W_j}(\mathrm{t}) < S_{W_j} \\ 0 \qquad\qquad\quad , \text{otherwise} \end{cases}, \text{ for } j = 1, 2, ..., N_W \tag{4.18}$$

$O_{W_j}(t)$ is the order given by wholesaler j at time t. Wholesaler agent selects one of the manufacturer agents as its supplier. $O_{W_jM_k}(t)$ is the order of wholesaler j from manufacturer k at time t. $O_{W_jM_k}(t) = O_{W_j}(t)$ for one and only one manufacturer, and $O_{W_jM_k}(t) = 0$ for the remaining.

$IP_{W_j}(t)$ is the inventory position of wholesaler j at time t. Inventory position is net inventory plus supply line ($SL_{W_j}(t)$). Inventory position and supply line for wholesalers are respectively:

$$IP_{W_j}(t) = NI_{W_j}(t) + SL_{W_j}(t), \text{ for } j = 1, 2, ..., N_W \tag{4.19}$$

$$SL_{W_j}(t) = SL_{W_j}(t-1) + O_{W_j}(t-1) + \sum_{k=1}^{N_M} SM_{M_kW_j}(t-\tau_{MW}), \text{ for } j = 1, 2, ..., N_W \tag{4.20}$$

$SL_{W_j}(t)$ increases with the order at the previous time step and decreases with the arrival of products sent by manufacturers at time $(t-\tau_{MW})$.

Manufacturers give production order decisions similarly:

$$O_{M_k}(t) = \begin{cases} S_{M_k} - IP_{M_k}(t), & \text{if } IP_{M_k}(\mathrm{t}) < S_{M_k} \\ 0 & \text{, otherwise} \end{cases} \text{, for } k = 1,2,...,N_M \tag{4.21}$$

$O_{M_k}(t)$ is the production order given by manufacturer k at time t.

$IP_{M_k}(t)$ is the inventory position of manufacturer k at time t. Inventory position is net inventory plus in transit orders. Inventory position of manufacturers is:

$$IP_{M_k}(t) = NI_{M_k}(t) + \sum_{m=1}^{\tau_p} O_{M_k}(t-m), \text{ for } k = 1,2,...,N_M \tag{4.22}$$

$\sum_{m=1}^{\tau_p - 1} O_{M_k}(t-m)$ term includes materials that are in the manufacturing line. All other orders, which are older than $\tau_P$ days, have already been received by time t. This is due to unlimited capacity assumption.

### 4.2.2. Variable Order-Up-To-Level (Variable S) Policy

Variable Order-Up-To-Level (Variable S) policy is the base stock policy where Order-Up-To-Level is adjusted according to the incoming demand. The remaining ordering structure is the same. Hence, equations 4.15-4.22 are valid for Variable S policy, only with the change that Order-Up-To-Level is a function of time. Variable S is an inflated lead time base stock policy with safety stock.

Retailers give order decisions as follows:

$$O_{R_i}(t) = \begin{cases} S_{R_i}(t) - IP_{R_i}(t), & \text{if } IP_{R_i}(\mathrm{t}) < S_{R_i}(t) \\ 0 & \text{, otherwise} \end{cases} \text{, for } i = 1,2,...,N_R \tag{4.23}$$

Order-Up-To-Level for retailers is:

$$S_{R_i}(t) = SS_{R_i} + (k_{R_i} + \tau_{WR}) * \hat{D}_{R_i}(t), \text{ for i} = 1,2,..., N_R \tag{4.24}$$

$S_{R_i}(t)$ is the Order-Up-To-Level of retailer i at time t. $SS_{R_i}$ and $k_{R_i}$ are constant safety stock level and lead time inflation parameter for retailer i, respectively. Their values are set simply by trial-and-error, searching for the minimal values that will cause neither backlogs nor excessive inventories. $k_{R_i}$ is set as three and value of $SS_{R_i}$ is as follows:

$$SS_{R_i} = 1.6 * (\hat{D}_{FC} / N_R), \text{ for i} = 1,2,..., N_R \tag{4.25}$$

$\hat{D}_{R_i}(t)$ is the demand forecast of retailer i. Forecasting is done through simple exponential smoothing with the smoothing constant $\alpha_{R_i}$. Smoothing equation is as follows:

$$\hat{D}_{R_i}(t) = \alpha_{R_i} * \hat{D}_{R_i}(t-1) + (1-\alpha_{R_i}) * D_{R_i}(t-1), \text{ for i} = 1,2,..., N_R \tag{4.26}$$

Wholesalers give order decisions similarly:

$$O_{W_j}(t) = \begin{cases} S_{W_j}(t) - IP_{W_j}(t), \text{ if } IP_{W_j}(\mathrm{t}) < S_{W_j}(t) \\ 0 \qquad\qquad\qquad\quad , \text{otherwise} \end{cases}, \text{ for } j = 1,2,..., N_W \tag{4.27}$$

Order-Up-To-Level for wholesalers is as follows:

$$S_{w_j}(t) = SS_{W_j} + (k_{W_j} + \tau_{MW}) * \hat{D}_{W_j}(t), \text{ for j} = 1,2,..., N_W \tag{4.28}$$

$S_{W_j}(t)$ is the Order-Up-To-Level of wholesaler j at time t. $SS_{W_j}$ and $k_{W_j}$ are constant safety stock level and lead time inflation parameter for wholesaler j, respectively. $k_{W_j}$ is set as three and $SS_{W_j}$ is as follows:

$$SS_{W_i} = 1.6 * (\hat{D}_{FC} / N_W), \text{ for j} = 1, 2, ..., N_W \tag{4.29}$$

$\hat{D}_{W_j}(t)$ is the demand forecast of wholesaler j. Demand expectation formulation is as follows:

$$\hat{D}_{W_j}(t) = \alpha_{W_j} * \hat{D}_{W_j}(t-1) + (1 - \alpha_{W_j}) * D_{W_j}(t-1), \text{ for j} = 1, 2, ..., N_W \tag{4.30}$$

Manufacturers give production order decisions similarly:

$$O_{M_k}(t) = \begin{cases} S_{M_k} - IP_{M_k}(t), \text{ if } IP_{M_k}(\text{t}) < S_{M_k} \\ 0 \qquad\qquad\qquad , \text{otherwise} \end{cases}, \text{ for } k = 1, 2, ..., N_M \tag{4.31}$$

Order-Up-To-Level for manufacturers is as follows:

$$S_{M_k}(t) = SS_{M_k} + (k_{M_k} + \tau_P) * \hat{D}_{M_k}(t), \text{ for k} = 1, 2, ..., N_M \tag{4.32}$$

$S_{M_k}(t)$ is the Order-Up-To-Level of manufacturer k at time t. $SS_{M_k}$ and $k_{M_k}$ are constant safety stock level and lead time inflation parameter for manufacturer k, respectively. $k_{M_k}$ is set as three and $SS_{M_k}$ is as follows:

$$SS_{M_k} = 1.6 * (\hat{D}_{FC} / N_M), \text{ for k} = 1, 2, ..., N_M \tag{4.33}$$

$\hat{D}_{M_k}(t)$ is the demand forecast of manufacturer k. Demand expectation formulation is as follows:

$$\hat{D}_{M_k}(t) = \alpha_{M_k} * \hat{D}_{M_k}(t-1) + (1 - \alpha_{M_k}) * D_{M_k}(t-1), \text{ for k} = 1, 2, ..., N_M \tag{4.34}$$

#### 4.2.3. Reorder Point – Order-Up-To-Level (s,S) Policy

Reorder Point – Order-Up-To-Level (s,S) policy is a continuous inventory replenishment policy, where an order is placed to raise the inventory position to Order-Up-To-Level (S) whenever inventory position drops to or below reorder point (s).

Retailers give order decisions as follows:

$$O_{R_i}(t) = \begin{cases} S_{R_i}(t) - IP_{R_i}(t), \text{ if } IP_{R_i}(\text{t}) \le s_{R_i}(t) \\ 0 \qquad\qquad\qquad\quad , \text{otherwise} \end{cases}, \text{ for } i = 1, 2, ..., N_R \tag{4.35}$$

Reorder point for retailers is defined as in Equation 4.36.

$$s_{R_i}(t) = SS_{R_i} + \tau_{WR} * \hat{D}_{R_i}(t), \text{ for i} = 1, 2, ..., N_R \tag{4.36}$$

$s_{R_i}(t)$ is the reorder point for retailer i at time t. $SS_{R_i}$ is constant safety stock level for retailer i. Safety stock for retailers is determined as follows:

$$SS_{R_i} = 1.6 * (\hat{D}_{FC} / N_R), \text{ for i} = 1, 2, ..., N_R \tag{4.37}$$

Order-Up-To-Level for retailers is given below:

$$S_{R_i}(t) = s_{R_i}(t) + k_{R_i} * \hat{D}_{R_i}(t), \text{ for i} = 1, 2, ..., N_R \tag{4.38}$$

$S_{R_i}(t)$ is the Order-Up-To-Level of retailer i at time t. $k_{R_i}$ is lead time inflation parameter for retailer i. $k_{R_i}$ is set as three. $\hat{D}_{R_i}(t)$ is the demand forecast of retailer i, and Equation 4.26 is used for forecasting.

Wholesalers give order decisions similarly:

$$O_{W_j}(t) = \begin{cases} S_{W_j}(t) - IP_{W_j}(t), \text{ if } IP_{W_j}(\text{t}) \le s_{W_j}(t) \\ 0 \qquad\qquad\qquad\quad , \text{otherwise} \end{cases}, \text{ for } j = 1, 2, ..., N_W \tag{4.39}$$

Reorder point for wholesalers is defined as in Equation 4.40.

$$s_{W_j}(t) = SS_{W_j} + \tau_{MW} * \hat{D}_{W_j}(t), \text{ for j} = 1, 2, ..., N_W \tag{4.40}$$

$s_{W_j}(t)$ is the reorder point for wholesaler j at time t. $SS_{W_j}$ is constant safety stock level for wholesaler j. Safety stock for wholesalers is specified as follows:

$$SS_{W_j} = 1.6 * (\hat{D}_{FC} / N_W), \text{ for j} = 1, 2, ..., N_W \tag{4.41}$$

Order-Up-To-Level for wholesalers is given below:

$$S_{W_j}(t) = s_{W_j}(t) + k_{W_j} * \hat{D}_{W_j}(t), \text{ for j} = 1, 2, ..., N_W \tag{4.42}$$

$S_{W_j}(t)$ is the Order-Up-To-Level of wholesaler j at time t. $k_{W_j}$ is lead time inflation parameter for wholesaler j. $k_{W_j}$ is set as three. $\hat{D}_{W_j}(t)$ is the demand forecast of wholesaler j, and Equation 4.30 is used for forecasting.

Manufacturers give production orders similarly:

$$O_{M_k}(t) = \begin{cases} S_{M_k}(t) - IP_{M_k}(t), \text{ if } IP_{M_k}(\text{t}) \leq s_{M_k}(t) \\ 0 \qquad\qquad\qquad, \text{ otherwise} \end{cases}, \text{ for } k = 1, 2, ..., N_M \tag{4.43}$$

Reorder point for manufacturers is defined as follows:

$$s_{M_k}(t) = SS_{M_k} + \tau_P * \hat{D}_{M_k}(t), \text{ for k} = 1, 2, ..., N_M \tag{4.44}$$

$s_{M_k}(t)$ is the reorder point for manufacturer k at time t. $SS_{M_k}$ is constant safety stock level for manufacturer k. Safety stock for manufacturers is specified as in Equation 4.45:

$$SS_{M_k} = 1.6 * (\hat{D}_{FC} / N_M), \text{ for j} = 1, 2, ..., N_M \quad (4.45)$$

Order-Up-To-Level for manufacturers is given below:

$$S_{M_k}(t) = s_{M_k}(t) + k_{M_k} * \hat{D}_{M_k}(t), \text{ for k} = 1, 2, ..., N_M \quad (4.46)$$

$S_{M_k}(t)$ is the Order-Up-To-Level of manufacturer k at time t. $k_{M_k}$ is lead time inflation parameter for manufacturer k. $k_{M_k}$ is set as three. $\hat{D}_{M_k}(t)$ is the demand forecast of manufacturer k, and Equation 4.34 is used for forecasting.

### 4.2.4. Anchor-And-Adjust (A&A) Policy

Anchor-And-Adjust (A&A) policy is a widely used stock management policy in the system dynamics literature. It aims to keep inventory and supply line at their desired levels, which are functions of demand forecasts and lead times (Sterman, 2000; Gündüz, 2003).

Retailers make order decisions as follows:

$$O_{R_i}(t) = \max\left(\frac{\mathrm{NI}^*_{R_i}(t) - NI_{R_i}(t)}{IAT_{R_i}} + \frac{\mathrm{SL}^*_{R_i}(t) - SL_{R_i}(t)}{SLAT_{R_i}} + \hat{D}_{R_i}(t), 0\right), \text{ for } i = 1, 2, ..., N_R \quad (4.47)$$

$\mathrm{NI}_{R_i}(t)$, $\mathrm{NI}^*_{R_i}(t)$, and $IAT_{R_i}$ are net inventory, desired inventory, and inventory adjustment time, respectively. $\mathrm{SL}_{R_i}(t)$, $\mathrm{SL}^*_{R_i}(t)$, and $SLAT_{R_i}$ are supply line inventory, desired supply line inventory, and supply line inventory adjustment time, respectively. Both $IAT_{R_i}$ and $SLAT_{R_i}$ are set as three.

The desired inventory, $\mathrm{NI}^*_{R_i}(t)$, is formulated as a multiplier, $m_{R_i}(t) = 5$, times demand forecast:

$$NI^{*}_{R_i}\text{(t)=m}_{R_i}\text{*}\hat{\text{D}}_{R_i}\text{(t), for i} = 1,2,...,N_R \tag{4.48}$$

Supply line of retailer i is given in Equation 4.17. Desired supply line of retailer i is a function of both the demand forecast and the transit lead time:

$$SL^{*}_{R_i}\text{(t)=}\tau_{WR}\text{*}\hat{\text{D}}_{R_i}\text{(t), for i} = 1,2,...,N_R \tag{4.49}$$

Wholesalers determine order quantities similarly:

$$O_{W_j}(t) = \max(\frac{\text{NI}^{*}_{W_j}(t) - NI_{W_j}(t)}{IAT_{W_j}} + \frac{\text{SL}^{*}_{W_j}(t) - SL_{W_j}(t)}{SLAT_{W_j}} + \hat{D}_{W_j}(t), 0),\text{ for } j = 1,2,...,N_W \tag{4.50}$$

$\text{NI}_{W_j}(t)$, $\text{NI}^{*}_{W_j}(t)$, and $IAT_{W_j}$ are net inventory, desired inventory, and inventory adjustment time, respectively. $\text{SL}_{W_j}(t)$, $\text{SL}^{*}_{W_j}(t)$, and $SLAT_{W_j}$ are supply line inventory, desired supply line inventory, and supply line inventory adjustment time, respectively. Both $IAT_{W_j}$ and $SLAT_{W_j}$ are set as three.

The desired inventory, $\text{NI}^{*}_{W_j}(t)$, is formulated as a multiplier, $m_{W_j}(t) = 5$, times demand forecast:

$$NI^{*}_{W_j}\text{(t)=m}_{W_j}\text{*}\hat{\text{D}}_{W_j}\text{(t), for j} = 1,2,...,N_W \tag{4.51}$$

Supply line of retailer i is given in Equation 4.20. Desired supply line of wholesaler j is a function of both the demand forecast and the transit lead time:

$$SL^{*}_{W_j}\text{(t)=}\tau_{MW}\text{*}\hat{\text{D}}_{W_j}\text{(t), for j} = 1,2,...,N_W \tag{4.52}$$

Manufacturers determine production order quantities as follows:

$$O_{M_k}(t) = \max(\frac{\mathrm{NI}^*_{M_k}(t) - NI_{M_k}(t)}{IAT_{M_k}} + \frac{\mathrm{SL}^*_{M_k}(t) - SL_{M_k}(t)}{SLAT_{M_k}} + \hat{D}_{M_k}(t), 0), \text{ for } k = 1, 2, ..., N_M \quad (4.53)$$

$\mathrm{NI}_{M_k}(t)$, $\mathrm{NI}^*_{M_k}(t)$, and $IAT_{M_k}$ are net inventory, desired inventory, and inventory adjustment time, respectively. $\mathrm{SL}_{M_k}(t)$, $\mathrm{SL}^*_{M_k}(t)$, and $SLAT_{M_k}$ are supply line inventory, desired supply line inventory, and supply line inventory adjustment time, respectively.

The desired inventory, $\mathrm{NI}^*_{M_k}(t)$, is formulated as a multiplier, $m_{M_k}(t)$, times demand forecast:

$$NI^*_{M_k}(\mathrm{t}) = \mathrm{m}_{M_k} * \hat{\mathrm{D}}_{M_k}(\mathrm{t}), \text{ for k} = 1, 2, ..., N_M \quad (4.54)$$

Supply line of manufacturer i is the production orders that have not been completed yet:

$$SL_{M_k}(t) = \sum_{m=1}^{\tau_P - 1} O_{M_k}(t-m), \; for \; k = 1, 2, ..., N_M \quad (4.55)$$

Desired supply line of manufacturer k is a function of both the demand forecast and the production lead time. Demand expectation formulation is as in Equation 4.34.

$$SL^*_{M_k}(\mathrm{t}) = \tau_P * \hat{\mathrm{D}}_{M_k}(\mathrm{t}), \text{ for k} = 1, 2, ..., N_M \quad (4.56)$$

The next chapter includes the analysis of the agent-based model under the ordering policies explained in this section and with modifications according to the adaptive pricing formulation.

# 5. DISAGGREGATED AGENT-BASED MODEL DYNAMICS

The disaggregated agent-based model is analyzed in this section under different agent features. The relationships between agent features, individual level agent dynamics and system level aggregate dynamics are systematically investigated. The formation of aggregate dynamics by individual dynamics is illustrated under different ordering policies and supply chain settings. The mechanism that governs the sensitivity of individual and aggregate dynamics to the supply chain topology is explained. The effects of agent features on aggregate supply chain dynamics are inquired by introducing components to agent schemata one at a time and analyzing the contribution of these components to the aggregate dynamics.

We first explain the procedure to analyze the periodicity of time series. Then, aggregate dynamics of the agent-based model are analyzed under basic ordering policies. The relationship between individual dynamics and aggregate dynamics is analyzed for each ordering policy. The dependency of aggregate dynamics on the supply chain topology is systematically investigated. Then, the effects of adaptive pricing on aggregate supply chain dynamics are analyzed.

## 5.1. The Analysis of Periodicity in Aggregate Inventory and Order Dynamics

Since this research puts a special emphasis on the variability and oscillatory dynamics of inventory and orders; periodicity analysis of time series is utilized in order to detect and quantify oscillatory components.

Experiments with the model have shown that the order and inventory signals include multiple periodicities. Spectral analysis is a widely used technique to quantify the strengths of the corresponding oscillatory dynamics in multi-periodic signals. We select the dominant periods in the data and report the spectral power of these periods as representations of the strength of oscillatory dynamics.

In spectral analysis, the signal is first transformed into frequency domain from time domain. Frequency, amplitude and phase information are examined in this domain rather than the time domain. Generally, phase information is irrelevant to the analysis; frequency and amplitude information are enough. Power spectrum gives the power information at each frequency, and includes amplitude and frequency information.

Spectral representation theorem states that all mean-zero stationary time-series can be represented as the superposition of sinusoidal waveforms. Based on this theorem, Fourier spectral analysis first transforms the data into frequency domain and represents the signal as superposition of orthogonal sinusoidals.

Normalized discrete Fourier transform of a sequence x(n), n=0,1,…, N-1, is a sequence of complex numbers X(f):

$$X(f_{k/N}) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} x(t) e^{\frac{i2\pi kt}{N}},\ k = 0,1,...,N-1 \tag{5.1}$$

X(f) represents x(n) as a linear combination of complex sinusoids $\frac{e^{\frac{i2\pi ft}{N}}}{\sqrt{N}}$, in frequency domain, and records the amplitude information of the original signal. Frequency, $f_{k/N}$, is the number of complete cycles (k/N) in unit time.

Power spectral density (PSD) is used to discover periodicities in a signal. PSD reveals how much power is concentrated at each frequency. PSD is the squared length of discrete Fourier transform:

$$P_F(f_{k/N}) = \left\| X(f_{k/N}) \right\|^2,\ k = 0,1,...,\frac{N-1}{2} \tag{5.2}$$

There are advanced issues in Fourier spectral analysis like windowing, finite-size data handling, PSD smoothing, and etc. However, numerically sensitive PSD estimation is beyond the scope of this research, and we basically use Tukey-Hanning estimate of PSD:

$$P_F(f_{k/N}) \ = \ \frac{1}{2\pi}\sum_{j=1}^{N/2}(1+\cos(2\pi j/N))*\mathrm{cov}(j)*\cos(2\pi jk/N),\ k=1,..,N/2 \qquad (5.3)$$

where Cov(j) is the covariance of time series x(t) at lag j.

Once PSD is calculated, one needs to distinguish the dominant periodicities from the insignificant ones. Vlachos *et. al.* (2005) develop an algorithm that finds a power threshold only over which the spectral power of a corresponding frequency can be considered significant –or under which no significant periodicity exists. Therefore, the spectral power over this threshold is a measure of the oscillatory dynamics. The significant period estimation algorithm is explained below.

Vlachos *et. al.* (2005) states that the significant periodicity is formed by the autocorrelation at lags other than zero. Therefore, a random process can be formed by removing the autocorrelation in the signal. Let's call the original signal x. Assume that we shuffle the elements of x and call this new permutation $\tilde{x}$. $\tilde{x}(t)$ will retain first order statistics of $x(t)$, but will not exhibit any periodicity. A periodicity in x can be considered significant only if it has power more than any value in the power spectrum of this random process. Therefore, we should compare individual power values of frequencies in PSD of x with the maximum power in PSD of $\tilde{x}$, $P_{\max}(\tilde{x})$:

$$P_{\max}(\tilde{x}) = \arg\max_k (P_{f_{k/N}}(\tilde{x})) \qquad (5.4)$$

where $P_{f_{k/N}}(\tilde{x})$ is the power of permutation $\tilde{x}$ at frequency (k/N).

We can formulate the power threshold, $P_T(x)$ by repeating this experiment for 100 times, and constructing a 95 percent confidence interval. Calculate $P_{\max}(\tilde{x})$ for 100 different permutations of x(t), 95$^{th}$ largest $P_{\max}$ is the estimate of $P_T(x)$. Then, an aggregate measure of the strength of oscillation is the total spectral power over the threshold, $P_D(x)$.

$$P_D(x) = \sum_{\{k: P_{f_{k/N}}(x) \geq P_T(x),\ k=0,1,2,..,N/2\}} P_{f_{k/N}}(x) \tag{5.5}$$

Appendix C illustrates the use of spectral analysis and dominant periodicity on several artificial data sets.

### 5.2. Inventory and Order Dynamics under Basic Ordering Policies

In this section, inventory and demand data at each supply chain echelon is analyzed under basic ordering policies. The aggregation is done by simple summation of the individual firm inventory/orders at each echelon. The following aggregation equations apply for all ordering policies and scenarios:

$$NI_R(t) = \sum_{i=1}^{N_R} NI_{R_i}(t) \tag{5.6}$$

$$NI_W(t) = \sum_{j=1}^{N_W} NI_{W_j}(t) \tag{5.7}$$

$$NI_M(t) = \sum_{k=1}^{N_M} NI_{M_k}(t) \tag{5.8}$$

$$O_{FC}(t) = \sum_{l=1}^{N_C} O_{FC_l}(t) = \sum_{l=1}^{N_C}\sum_{i=1}^{N_R} O_{FC_l R_i}(t) \tag{5.9}$$

$$O_R(t) = \sum_{i=1}^{N_R} O_{R_i}(t) = \sum_{i=1}^{N_R}\sum_{j=1}^{N_W} O_{R_i W_j}(t) \tag{5.10}$$

$$O_W(t) = \sum_{j=1}^{N_W} O_{W_j}(t) = \sum_{j=1}^{N_W}\sum_{k=1}^{N_M} O_{W_j M_k}(t) \tag{5.11}$$

$$O_M(t) = \sum_{k=1}^{N_M} O_{M_k}(t) \tag{5.12}$$

$NI_R(t)$, $NI_W(t)$, and $NI_M(t)$ are aggregate inventory for retailer, wholesaler, and manufacturer echelons, respectively. $O_{FC}(t)$, $O_R(t)$, $O_W(t)$, and $O_M(t)$ are aggregate echelon orders for final customers, retailers, wholesalers and manufacturers respectively.

It is important to note that the individual final customer demand is distributed uniformly between 0 and 5. The number of final customers is 500, otherwise stated.

The time horizon in the experiments of this section is 1000 days; and the simulation experiments are repeated 20 times and the statistics reported are the averages of the statistics from each run of that scenario. However, plotting time horizon is selected as 300 in order to demonstrate the behavioral patterns more adequately.

### 5.2.1. Inventory and Order Dynamics for Fixed Order-Up-To-Level Policy

Inventory balance equation for individual retailers is given in Equation 4.1. It is possible to make analytical derivations, once backlog assumption is made. This is a common assumption in inventory management literature for analytical purposes, as stated by Chen *et. al.* (2000). It is important to note that simulation experiments with the model have shown that Order-Up-To-Levels (which can be specified such that backlogs are prevented) affect only the aggregate inventory average in terms of pattern components. It follows from no backlog assumption that firms satisfy the whole demand at each period. Therefore, Equations 5.14 and 5.15 are substituted for equations 4.4 and 4.8, respectively. Equation 5.13 follows with the same reasoning.

$$SM_{R_i FC_l}(t) = O_{FC_l R_i}(t), \text{ for } l = 1,2,...,N_C; i = 1,2,...,N_R \tag{5.13}$$

$$SM_{W_j R_i}(t) = O_{R_i W_j}(t), \text{ for } i = 1,2,...,N_R; j = 1,2,...,N_W \tag{5.14}$$

$$SM_{M_k W_j}(t) = O_{W_j M_k}(t), \text{ for } j = 1,2,...,N_W; k = 1,2,...,N_M \tag{5.15}$$

Plugging Equation 5.13 in the inventory balance equation (Equation 4.1) yields:

$$NI_{R_i}(t) = NI_{R_i}(t-1) + \sum_{j=1}^{N_W} O_{R_iW_j}(t-\tau_{WR}) - D_{R_i}(t), \text{ for } i = 1,2,...,N_R \tag{5.16}$$

Summation of individual inventory balance equations and plugging in Equation 5.6 and Equation 5.10 gives the aggregate balance equation for the retailer echelon:

$$NI_R(t) = NI_R(t-1) + O_R(t-\tau_{WR}) - D_R(t) \tag{5.17}$$

Aggregate inventory balance equations for wholesalers and manufacturers follow with the same reasoning:

$$NI_W(t) = NI_W(t-1) + O_W(t-\tau_{MR}) - D_W(t) \tag{5.18}$$

$$NI_M(t) = NI_M(t-1) + O_M(t-\tau_P) - D_M(t) \tag{5.19}$$

Ordering policy equation for individual retailers under Fixed S policy is given in Equation 4.15. The constancy of Order-Up-To-Level guarantees that $IP_{R_i}$ (t) ≤ $S_{R_i}$ . Therefore, Equation 5.20 is substituted for Equation 4.15.

$$O_{R_i}(t) = S_{R_i} - IP_{R_i}(t), \text{ for } i = 1,2,...,N_R \tag{5.20}$$

Plugging Equation 4.16 and Equation 4.17 in Equation 5.20 results in:

$$O_{R_i}(t) = S_{R_i} - NI_{R_i}(t) - \sum_{m=1}^{t} O_{R_i}(t-m) + \sum_{j=1}^{N_W} \sum_{m=\tau_{WR}}^{t} SM_{W_jR_i}(t-m), \text{ for } i = 1,2,...,N_R \tag{5.21}$$

Plugging in $O_{R_i}(t) = \sum_{j=1}^{N_W} O_{R_iW_j}(t)$ and Equation 5.14 in Equation 5.21 yields:

$$O_{R_i}(t) = S_{R_i} - NI_{R_i}(t) - \sum_{m=1}^{t} O_{R_i}(t-m) + \sum_{m=\tau_{WR}}^{t} O_{R_i}(t-m), \text{ for } i = 1,2,...,N_R \tag{5.22}$$

Equation 5.23 follows directly from Equation 5.22:

$$O_{R_i}(t) = S_{R_i} - NI_{R_i}(t) - \sum_{m=1}^{\tau_{WR}-1} O_{R_i}(t-m), \text{ for } i = 1,2,...,N_R \tag{5.23}$$

Summation of individual ordering equations and plugging in Equations 5.6 and 5.10 gives the aggregate ordering equation for the retailer echelon:

$$O_R(t) = \sum_{i=1}^{N_R} S_{R_i} - NI_R(t) - \sum_{m=1}^{\tau_{WR}-1} O_R(t-m) \tag{5.24}$$

Equation 5.25 follows from Equation 5.24:

$$NI_R(t) - NI_R(t-1) = O_R(t-\tau_{WR}) - O_R(t) \tag{5.25}$$

From Equations 5.17 and Equation 5.25:

$$O_R(t) = D_R(t) \tag{5.26}$$

The same procedure applies to wholesaler and manufacturer echelons. The result is that demand at retailer echelon is transmitted through the supply chain without any modification. Therefore, there is no bullwhip effect in aggregate echelon orders in Fixed S policy irrespective of the number of agents and Order-Up-To-Levels.

$$O_M(t) = O_W(t) = O_R(t) = D_R(t) \tag{5.27}$$

From equation 5.24, aggregate retailer echelon inventory can be represented in terms of safety stocks and orders at the last $\tau_{WR}$ periods:

$$NI_R(t) = \sum_{i=1}^{N_R} S_{R_i} - \sum_{m=0}^{\tau_{WR}-1} O_R(t-m) \tag{5.28}$$

Similarly:

$$NI_W(t) = \sum_{j=1}^{N_W} S_{W_j} - \sum_{m=0}^{\tau_{MW}-1} O_W(t-m) \tag{5.29}$$

$$NI_M(t) = \sum_{k=1}^{N_M} S_{R_k} - \sum_{m=0}^{\tau_P-1} O_M(t-m) \tag{5.30}$$

Then, the variance at each echelon is a function of final customer demand and the corresponding lead time:

$$Var[NI_R(t)] = \tau_{WR}\sigma_{FC}^2 \tag{5.31}$$

where $\sigma_{FC}$ is the variance of uncorrelated aggregate final customer demand.

Similarly:

$$Var[NI_W(t)] = \tau_{MW}\sigma_{FC}^2 \tag{5.32}$$

$$Var[NI_M(t)] = \tau_P\sigma_{FC}^2 \tag{5.33}$$

Therefore, there is bullwhip effect at aggregate echelon inventories only due to the increasing lead times of. The conclusions that under Fixed S policy bullwhip effect is not observed in aggregate echelon orders and that it exists in aggregate echelon inventories only if the lead times are rising in the upward direction are generalizations of the findings by Chen *et. al.* (2000). Chen *et. al.* (2000) shows that there is no bullwhip effect in a two-echelon supply chain if there is a single firm at each echelon and if the firms are using Fixed S policy.

Simulation experiments below illustrate the analytical findings. Experiments are made for several settings with different number of agents. Figure 5.1 shows aggregate order dynamics at each echelon for the case of 20 retailers, 10 wholesalers and two manufacturers. Outputs for the remaining experiments can be found in Appendix D. Figure 5.1 reveals that aggregate order remains the same throughout the supply chain and it changes randomly around a fixed level without any autocorrelation. In other words, there is no bullwhip effect in aggregate orders. Table 5.1 shows the statistics for order data for some sample cases with different number of agents. Since aggregate order remains the same; standard deviation, mean and range do not change at different echelons. Order statistics are not influenced by the number of agents at any echelon as long as the number of final customers and their individual demand distributions remain the same, which is an observation in line with Equation 5.27.

Table 5.1. Aggregate Order Statistics for Fixed S Policy
(R2W2M2 denotes 2 retailers, 2 wholesalers, 2 Manufacturers)

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 |
|---|---|---|---|---|---|
| RTL. | Mean | 1249.528 | 1250.273 | 1250.164 | 1249.694 |
| | std. dev. | 32.1771 | 32.2639 | 32.2224 | 32.0516 |
| | Range | 235.5001 | 252.9065 | 245.2582 | 249.5351 |
| WHL. | Mean | 1249.528 | 1250.273 | 1250.164 | 1249.694 |
| | std. dev. | 32.1771 | 32.2639 | 32.2224 | 32.0516 |
| | Range | 235.5001 | 252.9065 | 245.2582 | 249.5351 |
| MNF. | Mean | 1249.528 | 1250.273 | 1250.164 | 1249.694 |
| | std. dev. | 32.1771 | 32.2639 | 32.2224 | 32.0516 |
| | Range | 235.5001 | 252.9065 | 245.2582 | 249.5351 |

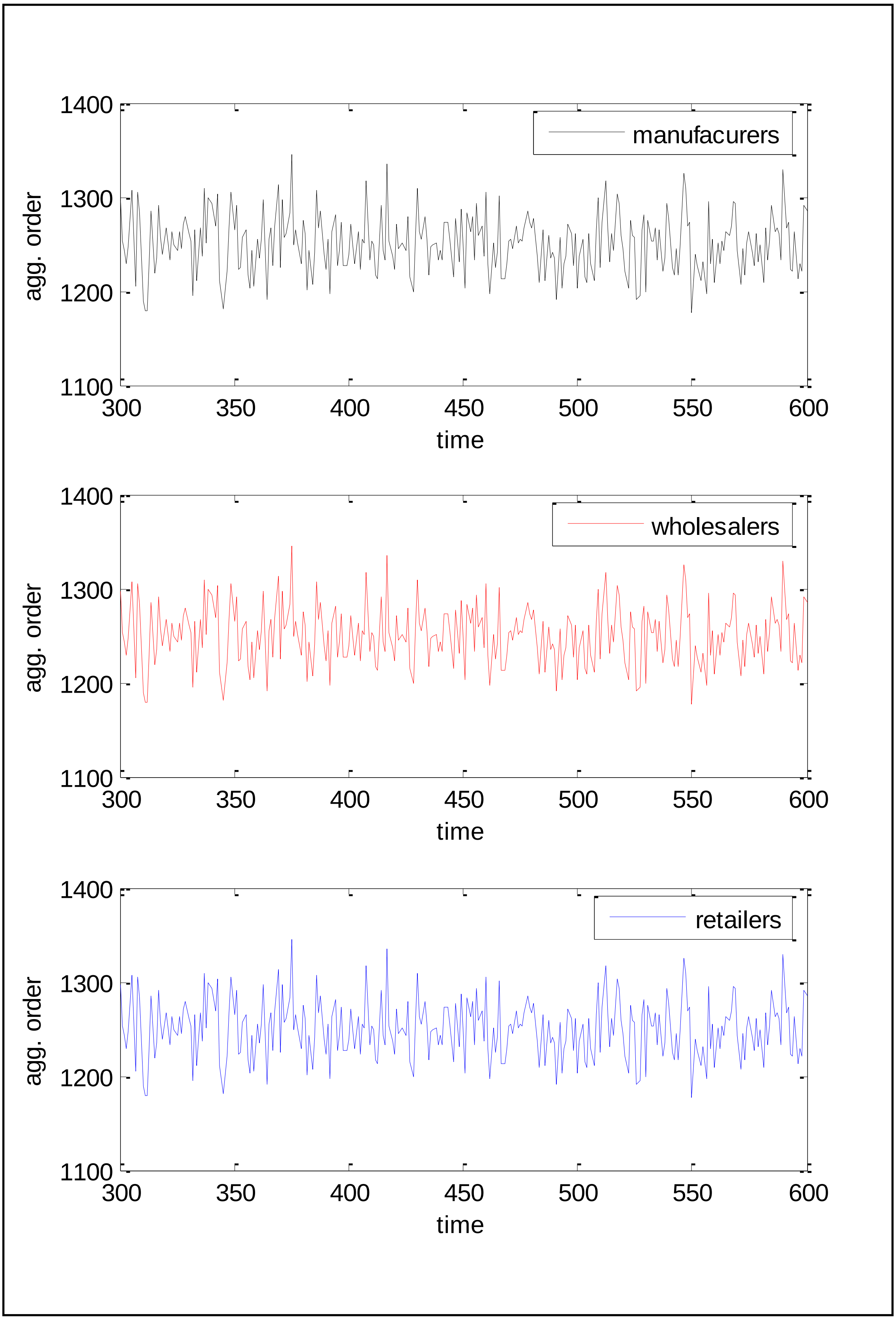


Figure 5.1. Aggregate Order Dynamics for Fixed S Policy

(20 Retailers, 10 Wholesalers, 2 Manufacturers)

The observation that the bullwhip effect does not exist at aggregate orders under Fixed S policy is valid independent of the individual Order-Up-To-Levels. Figure 5.2 and Figure 5.3 demonstrate the aggregate order time series for two cases with different Order-Up-To-Levels. Figure 5.2 and Figure 5.3 correspond to the scenarios if S values of all agents are selected as 100 and 1000, respectively. It can be observed that aggregate order dynamics are independent of the Order-Up-To-Levels and bullwhip effect does not occur in any case.

Figure 5.4 shows the power spectral densities for aggregate order time series in Figure 5.1. Figures for the remaining experiments with Fixed S policy can be found in Appendix D. Figure 5.4 shows that the spectral power is spread over the whole spectrum for aggregate orders; i.e. there is no significant periodicity. Table D.1 in Appendix D shows the spectral analysis results for aggregate order data under various settings of the number of agents. Total spectral power remains the same at each echelon irrespective of the number of agents as long as the number of final customers and their individual demand distributions remain the same. Dominant spectral power is zero at all echelons. These facts demonstrate that there is no periodicity and bullwhip effect in aggregate order time series.

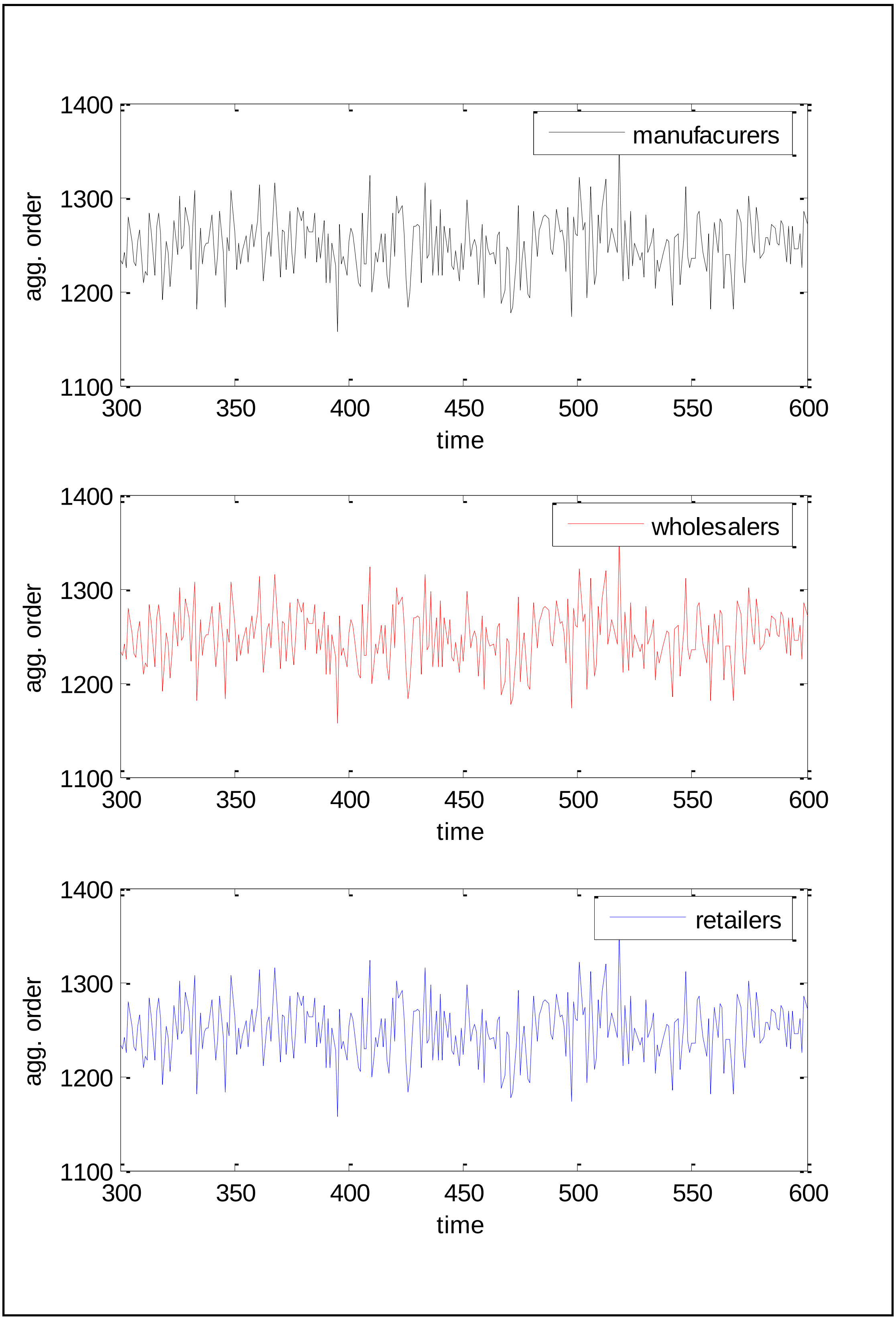


Figure 5.2. Aggregate Order Dynamics for Fixed S Policy when S=100

(20 Rtl., 10 Whl., 2 Mnf.)

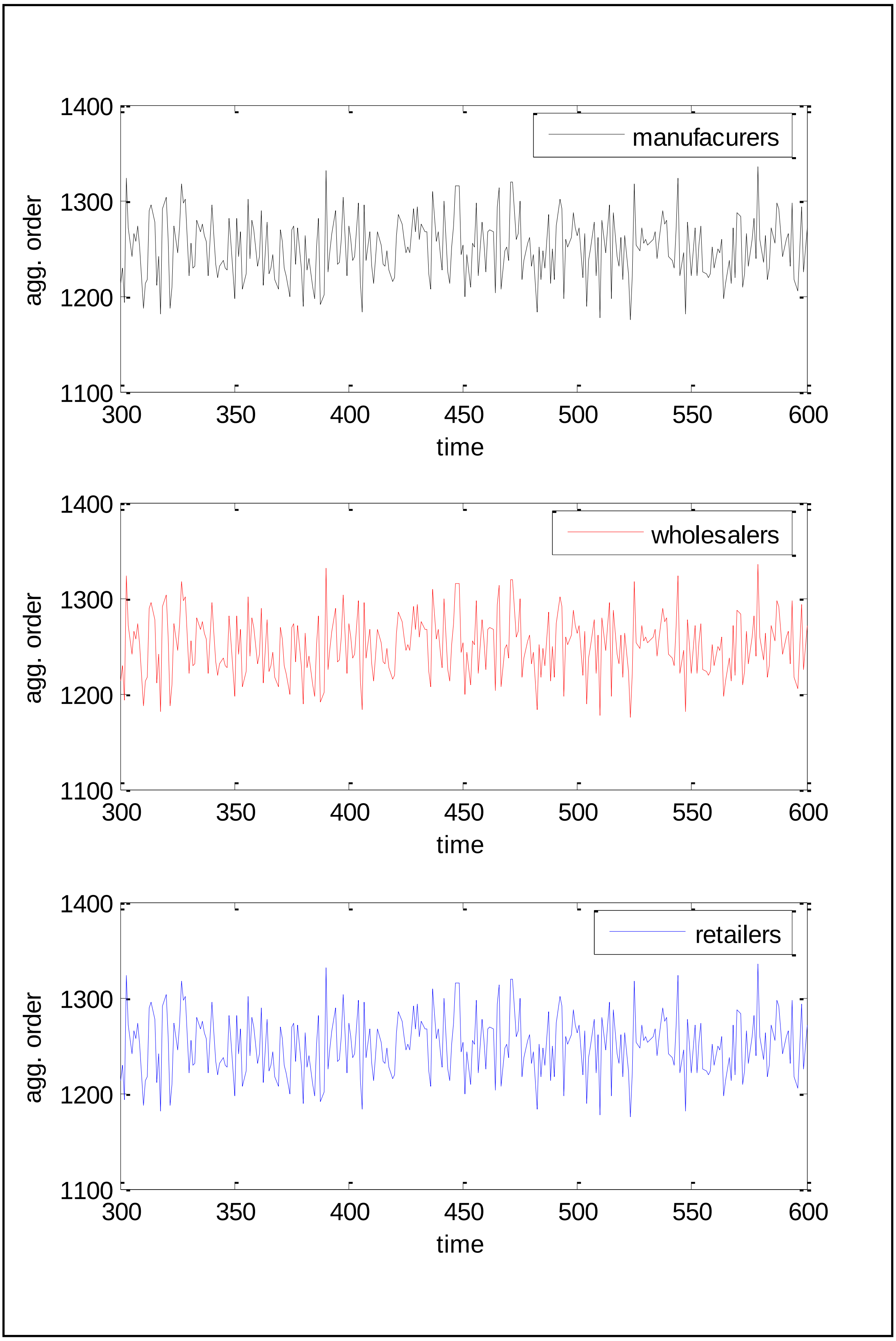


Figure 5.3. Aggregate Order Dynamics for Fixed S Policy when S=1000

(20 Rtl., 10 Whl., 2 Mnf.)

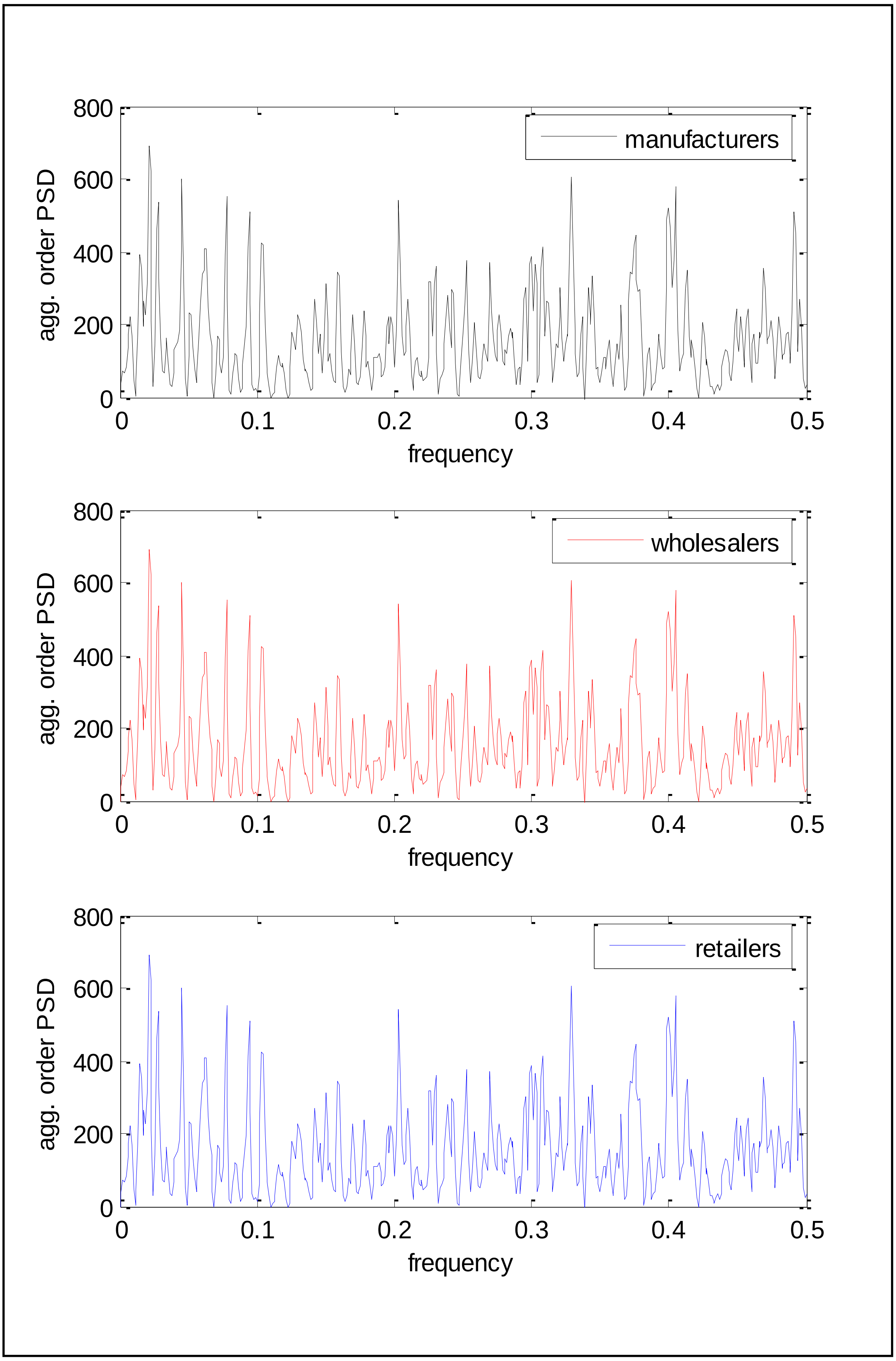


Figure 5.4. Spectral Power Density of Aggregate Orders for Fixed S Policy

(20 Rtl., 10 Whl., 2 Mnf.)

Table 5.2 reports the comparison of the system level aggregate dynamics with the individual firm dynamics. The aim in this comparison is to identify the degree to which aggregate dynamics differ from the individual agent dynamics in terms of variability. This statistics is informative about the capability of the aggregate analysis to capture the variability dynamics at individual scale. For this purpose, time series of order average at an echelon (which is the aggregate orders divided by the number of agents at that echelon) is compared with the time series of a "representative agent" at the same echelon, in terms of standard deviation statistics. Notice that averaging provides normalization. Although it is intuitive that averaging reduces the variability, it is not clear to what extent; because central limit theorem is not applicable to our case due to the presence of correlations among individual order time series. The ratio of average echelon order standard deviation to representative agent order standard deviation quantifies how much reduction is made in terms of variability. Let $X_1(t), X_2(t), ..., X_n(t)$ be individual time series. Then, $\hat{X}(t) = \sum_{i=1}^{N} X_i(t)/n$ is the average time series. The actual ratio is calculated as follows:

$$Actual\ Ratio(X) = \sqrt{\frac{\sigma_{\hat{X}(t)}}{\sigma_{X_i(t)}}} \tag{5.34}$$

Table 5.2. Fixed S Policy, Aggregate-Individual Order Variance Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.443892 | 0.27758 | 0.115142 | 0.071021 | 0.035707 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.036508 | 0.029554 | 0.03755 | 0.039492 | 0.011655 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.029659 | 0.022536 | 0.067688 | 0.055277 | 0.012234 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

Table 5.2 gives the variability ratios for all supply chain echelons under settings with different number of agents. Furthermore, reference ratios are given in order to be able to conceive the intensity of reduction in variability. A reference ratio corresponds to the case where averaging does not introduce any variability reduction other than the number of samples' effect. Reference ratio can be calculated by a procedure which constructs average echelon order time series by using phase-shifted individual agent order time series in order to remove the characteristics peculiar to the system considered. The algorithm for reference ratio determination is given and explained in Appendix E. Standard deviation ratios found by the algorithm converge to $1/\sqrt{N}$, where N is the number of agents in the echelon, which is the variability reduction due to the central limit theorem. The intuition that the reference reduction should be according to the central limit theorem is confirmed by this algorithm. The reference ratios given throughout Section 5 are simply equal to $1/\sqrt{N}$.

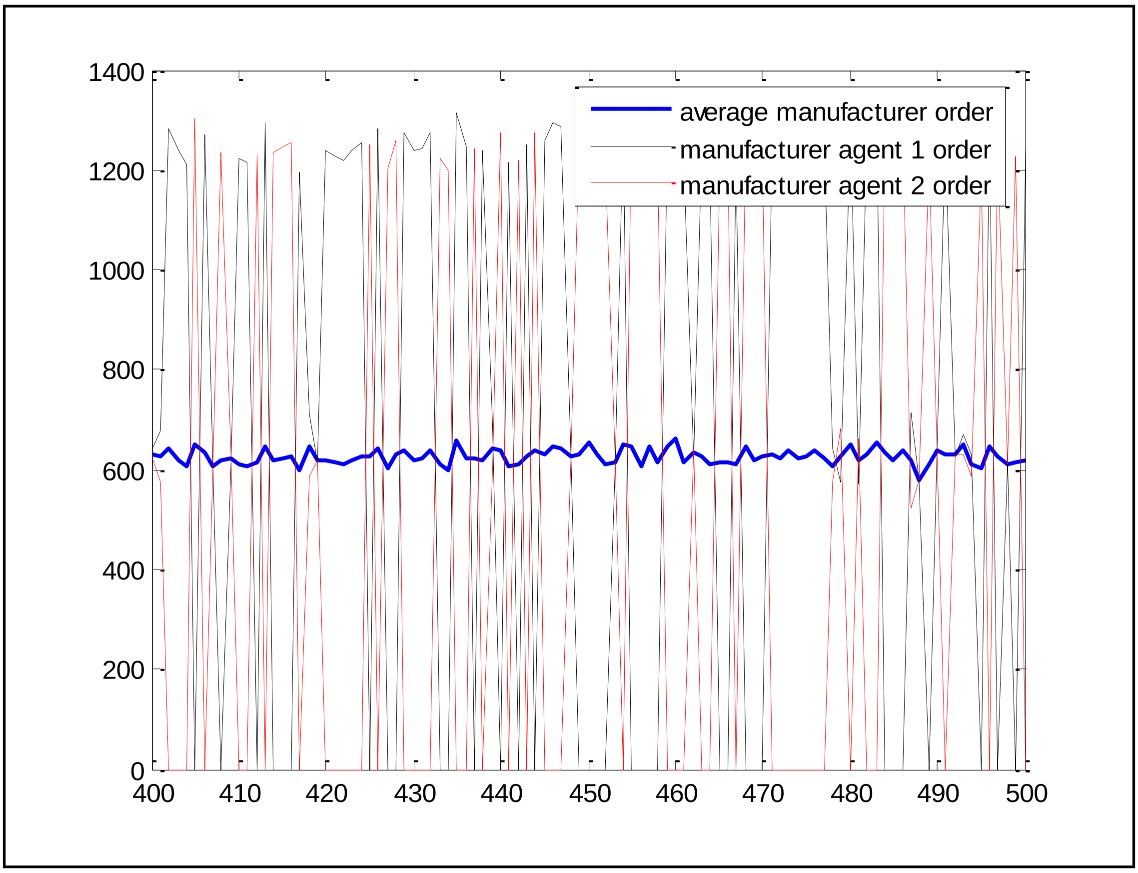


Figure 5.5. Comparison of Individual Manufacturer Orders with Average Manufacturer Echelon Order for Fixed S Policy (20 Rtl., 10 Whl., 2 Mnf.)

It can be observed from Table 5.2 that the variability reduction in retailer, wholesaler, and manufacturer orders are higher than the reference reduction. The reduction in wholesalers and manufacturers is very intense. This is caused by the fact that wholesaler and manufacturer aggregate order time series have phenomenally low variability, when compared to individual order time series. This observation points to the fact that aggregation -of the individual outputs- removes most of the variability existing in individual order dynamics at higher echelons, under Fixed S policy. Figure 5.5 exemplifies “the dramatically high variability of individual orders with respect to the average order” from the manufacturer echelon for the case of 20 retailers, 10 wholesalers, and two manufacturers. Even though the individual order time series exhibit relatively strong variability, the echelon as a whole adjusts itself such that the variability is significantly removed by the cross-correlations and phase lags among individual order time series. Therefore, analyzing aggregate orders underestimates the variability existing in individual orders at high echelons. Under Fixed S policy, there is loss of degrees of freedom at any echelon by its complete dependence on the aggregate order at the lowest echelon. This causes extensive reduction in the variability. This fact reveals that the systemic characteristics of Fixed Order-Up-To-Level policy results in the non-modified transmission of demand information through the supply chain which provides the reduction of individual variability strongly by aggregation, especially at higher echelons.

Figure 5.6 and Figure 5.7 show individual and aggregate echelon inventory dynamics for the case of 20 retailers, 10 wholesalers, and two manufacturers. They demonstrate that aggregate inventories fluctuate around levels which depend on lead times and fixed Order-Up-To-Levels, as in Equations 5.28-5.30, and that individual inventory variability is very high compared to aggregate inventory variability at wholesaler and manufacturer echelons. Aggregate inventory average is increasing in the upward direction as a result of using higher Order-Up-To-Levels. The fluctuation is quite low when compared to the other policies in this section. The variability is the same at retailer and wholesaler echelons, and higher at the manufacturer echelon due to increasing lead times at the manufacturer echelon. Figure 5.8 reveals that the spectral power is concentrated at low to medium frequencies, and that multiple periodicities can be detected in aggregate inventory data. There is almost no power at high frequencies. The condensation of spectral power and existence of periodicity is the result of autocorrelation in inventory. The spectral power is the same for retailers and wholesalers and higher for manufacturers, which is an indicator of the bullwhip effect. Table D.2 in Appendix D shows the results of spectral analysis for inventory data under various settings of number of agents. It can be inferred that total spectral power remains the same at retailer and wholesaler echelons and is higher at the manufacturer echelon irrespective of the number of agents. Similar observations are available for the dominant spectral powers. These facts indicate that there is bullwhip effect in terms of oscillation amplitude amplification at the manufacturer echelon. It is also important to note that there are multiple periodicities in inventory data and the dominance of any frequency at an echelon is preserved at upper echelons.

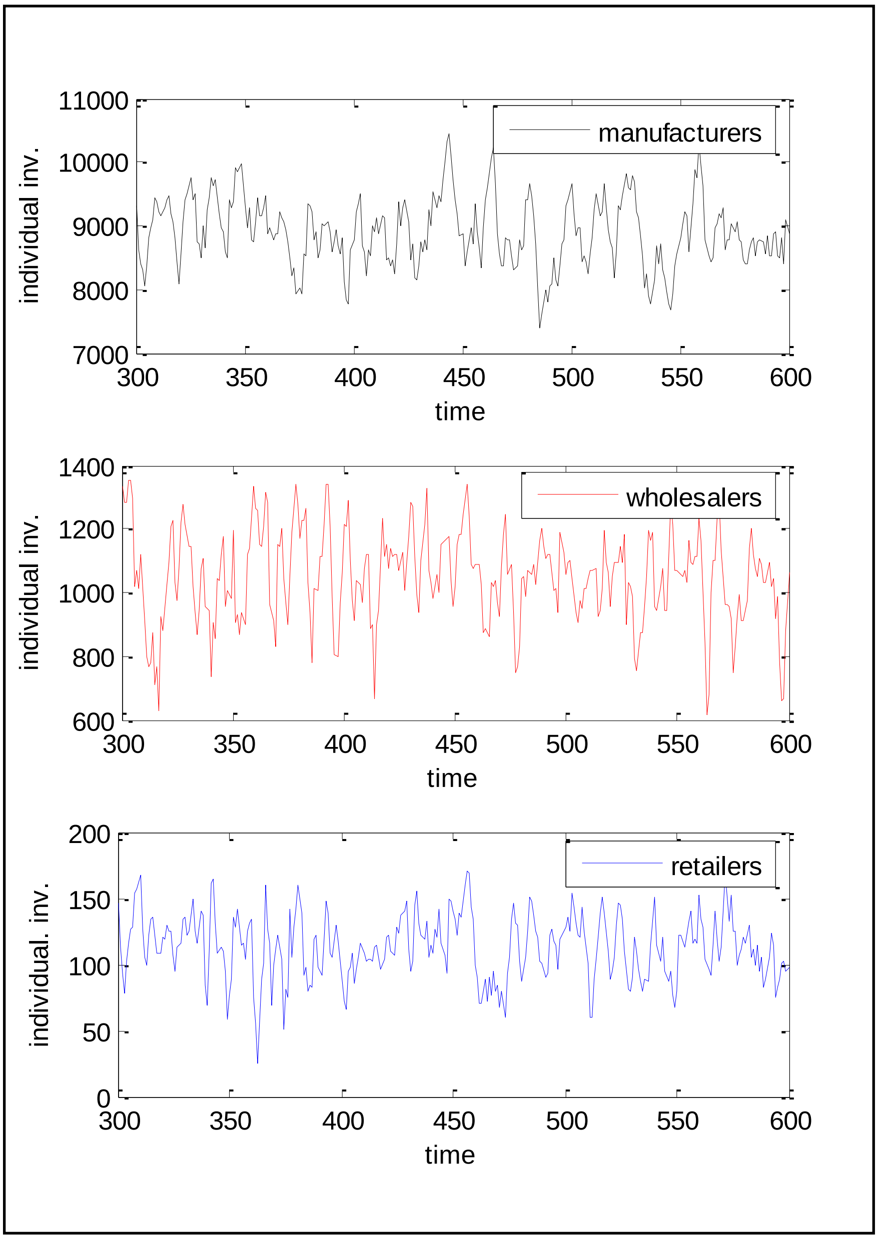


Figure 5.6. Individual Inventory Dynamics for Fixed S Policy

(20 Rtl., 10 Whl., 2 Mnf.)

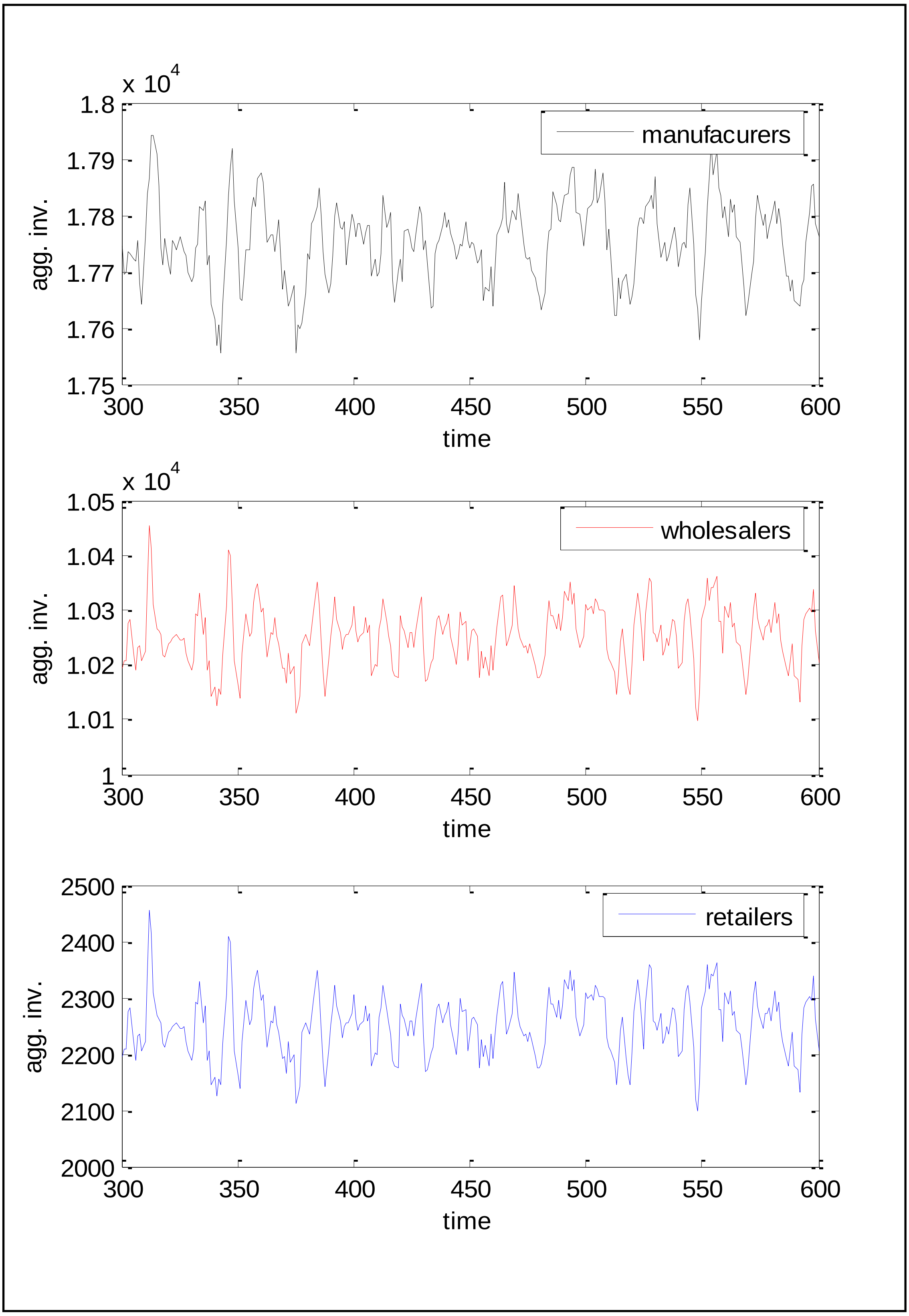


Figure 5.7. Aggregate Inventory Dynamics for Fixed S Policy

(20 Rtl., 10 Whl., 2 Mnf.)

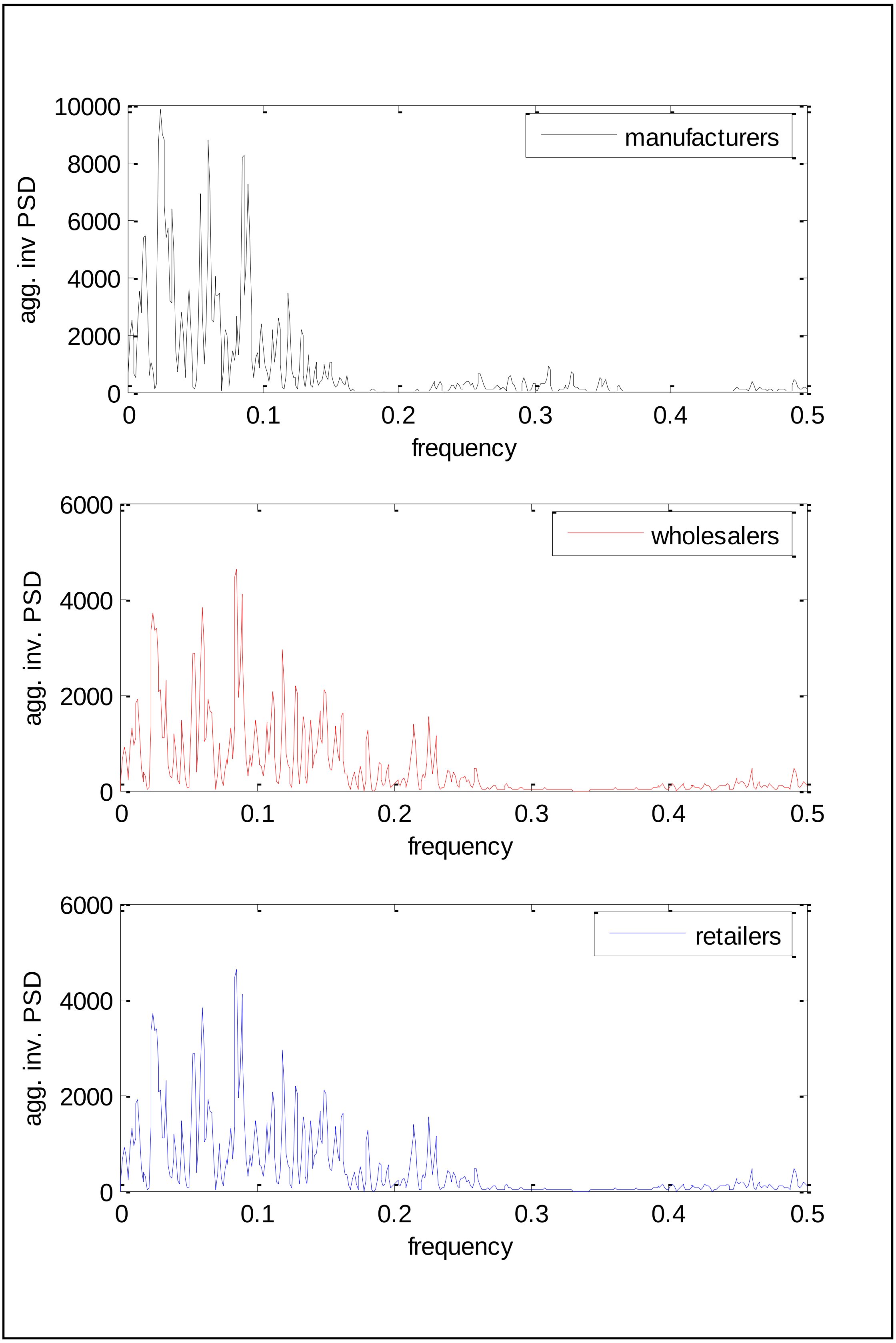


Figure 5.8. Spectral Power Density of Aggregate Inventories for Fixed S Policy (20 Rtl., 10 Whl., 2 Mnf.)

Table 5.3 gives the statistics of inventory data for various settings of the number of agents. Standard deviation and mean statistics are compatible with the analytical formulas in Equations 5.28-5.33. The table illustrates that the bullwhip effect at aggregate inventory variability occurs only as a result of increased lead times of upper level echelons, which can be seen also from Figure 5.7. It can be observed that standard deviation and range are not influenced by the number of agents, which is compatible with the analytical finding that the aggregate echelon inventory dynamics are independent of the number of agents. Aggregate echelon inventory mean increases in the upward direction, as a result of using higher S values in order to prevent individual backlogs.

Table 5.3. Aggregate Inventory Statistics for Fixed S Policy

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 |
|---|---|---|---|---|---|
| RTL. | Mean | 2251.399 | 2249.147 | 2249.506 | 2250.898 |
| | std. dev. | 54.5711 | 56.2308 | 56.015 | 55.7449 |
| | Range | 415.3931 | 467.5417 | 412.0033 | 427.7621 |
| WHL. | Mean | 10251.4 | 10249.15 | 10249.51 | 10250.9 |
| | std. dev. | 54.5711 | 56.2308 | 56.015 | 55.7449 |
| | Range | 415.3931 | 467.5417 | 412.0033 | 427.7621 |
| MNF. | Mean | 17752.32 | 17748.56 | 17749.18 | 17751.48 |
| | std. dev. | 69.8904 | 72.8651 | 72.633 | 72.2719 |
| | Range | 567.5875 | 593.5503 | 562.968 | 518.0251 |

Table 5.4 summarizes the comparison of the system level aggregate dynamics with the individual level firm dynamics by using inventory information. The ratios are very close to the ratios for orders. Conclusions regarding the order dynamics comparison are valid for inventory dynamics comparison. Aggregation of the individual outputs removes most of the variability existing in individual inventory dynamics at higher echelons, under Fixed S policy.

Table 5.4. Fixed S Policy, Aggregate-Individual Inventory Variability Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.432012 | 0.28137 | 0.115268 | 0.07178 | 0.035647 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.035749 | 0.029673 | 0.037645 | 0.039498 | 0.011668 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.028872 | 0.022467 | 0.068223 | 0.056067 | 0.012413 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

To summarize the observations about the Fixed S policy, the final customer demand information is transmitted through the supply chain as purchase or production orders without any modification. Therefore, there is no bullwhip effect in order data. However, this does not indicate that firms should use Fixed S policy in order to avoid the bullwhip effect; because Fixed Order-Up-To-Level is “guessed” in this policy. The results should be interpreted as Order-Up-To-Level should not be over-reactive to the changing demand and longer forecasting horizons should be used (Chen et. al., 2000). As far as inventory dynamics are concerned, aggregate inventory of any echelon can be expressed in terms of the lead time demand -which introduces autocorrelation to inventory data. Bullwhip effect in inventory data is experienced only as a result of the rising lead times in the upward direction. In the model considered, bullwhip effect is observed when manufacturers and lower echelons are compared. As far as individual dynamics are concerned, aggregation of the individual outputs removes most of the variability existing in individual order and inventory dynamics especially at higher echelons. Even though individual order and inventory time series exhibit relatively strong variability, the echelon as a whole adjusts itself such that the variability is removed by the cross-correlations and phase lags among individual agent time series. Therefore, analysis at aggregate level underestimates the variability at individual level.

### 5.2.2. Inventory and Order Dynamics for Variable Order-Up-To-Level Policy

It is not possible to make analytical derivations about order and inventory dynamics in Variable Order-Up-To-Level policy, when exponential smoothing is used for demand forecasting. However, simulation experiments can be used to investigate how supply chain dynamics are influenced by the introduction of demand forecasting. Experiments of Section 5.2.1 are repeated in this section with Variable S policy. Figure 5.9 and Figure 5.18 show aggregate order and inventory dynamics at each echelon for the case of 20 retailers, 10 wholesalers and two manufacturers. Figures for other experiments with Variable Order-Up-To-Level policy can be found in Appendix D.

When Figure 5.9 is compared with Figure 5.1, it can be seen that variability and amplitude are significantly higher at all echelons, especially for wholesalers and manufacturers (One should notice that the y-axis ranges of the graphs are significantly higher for the Variable S case, in comparing the two graphs). The order variability is amplified in the upward direction of the supply chain. Namely, bullwhip effect emerges as a result of demand forecasting. On the other hand, aggregate order averages remain the same. A further remark is that aggregate manufacturer order may become zero reflecting that both two manufacturers do not give production orders at those periods. However, this is an observation depending on the number of manufacturers, as explained below.

Figure 5.10 shows the power spectral densities of aggregate orders for the case of 20 retailers, 10 wholesalers and two manufacturers. It can be observed that the spectral power is spread over the whole spectrum. Thus, there is no dominant periodicity in the aggregate order data at any echelon, for the Variable S policy. Table D.3 in Appendix D shows the spectral analysis results for order data in several scenarios with different number of agents. Total spectral power increases as one moves up in the supply chain. Dominant spectral power is zero at all echelons. These two facts show that there is bullwhip effect in terms of order variation amplification; however aggregate orders do not have any significant periodicity.

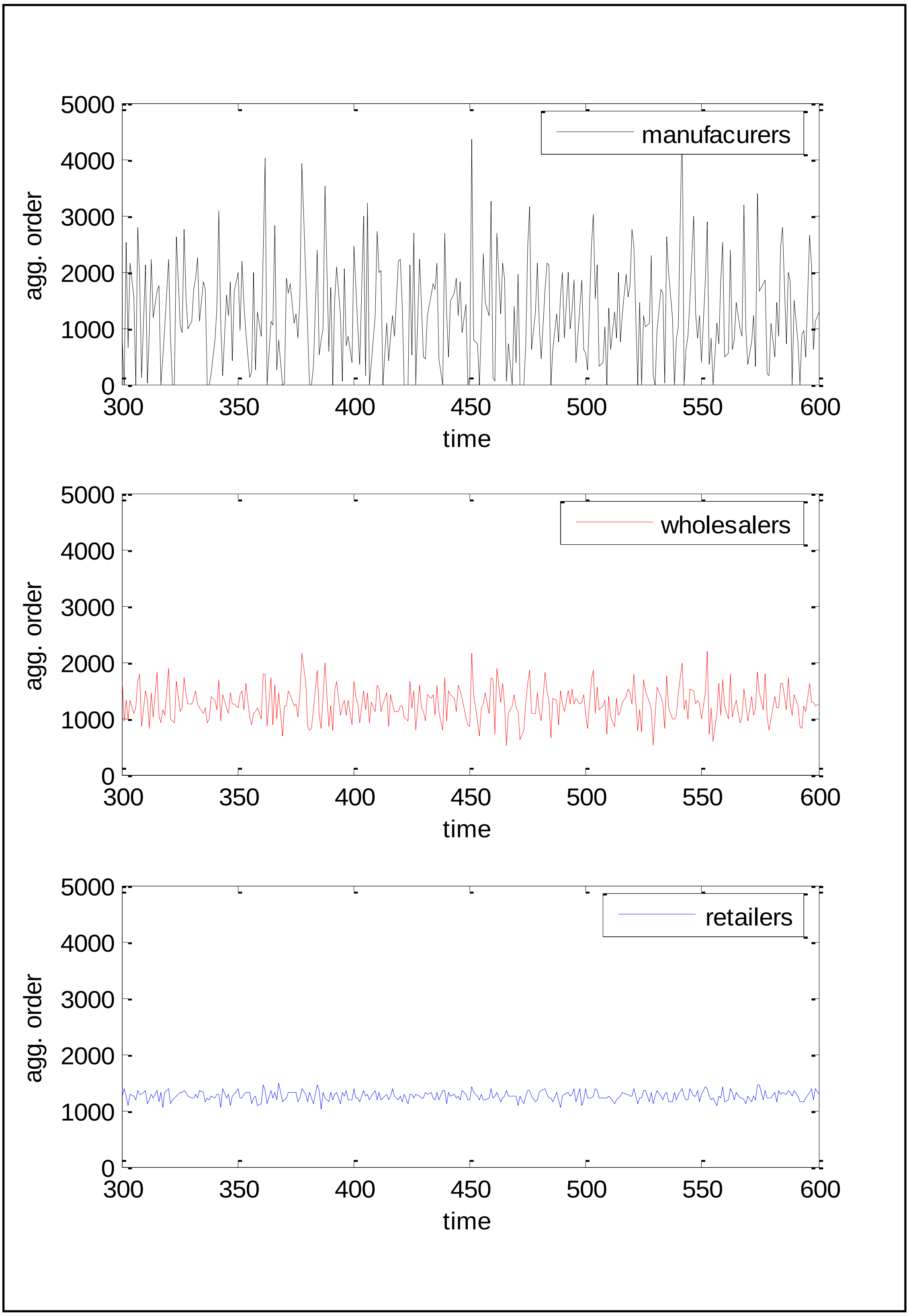


Figure 5.9. Aggregate Order Dynamics for Variable S Policy

(20 Rtl., 10 Whl., 2 Mnf.)

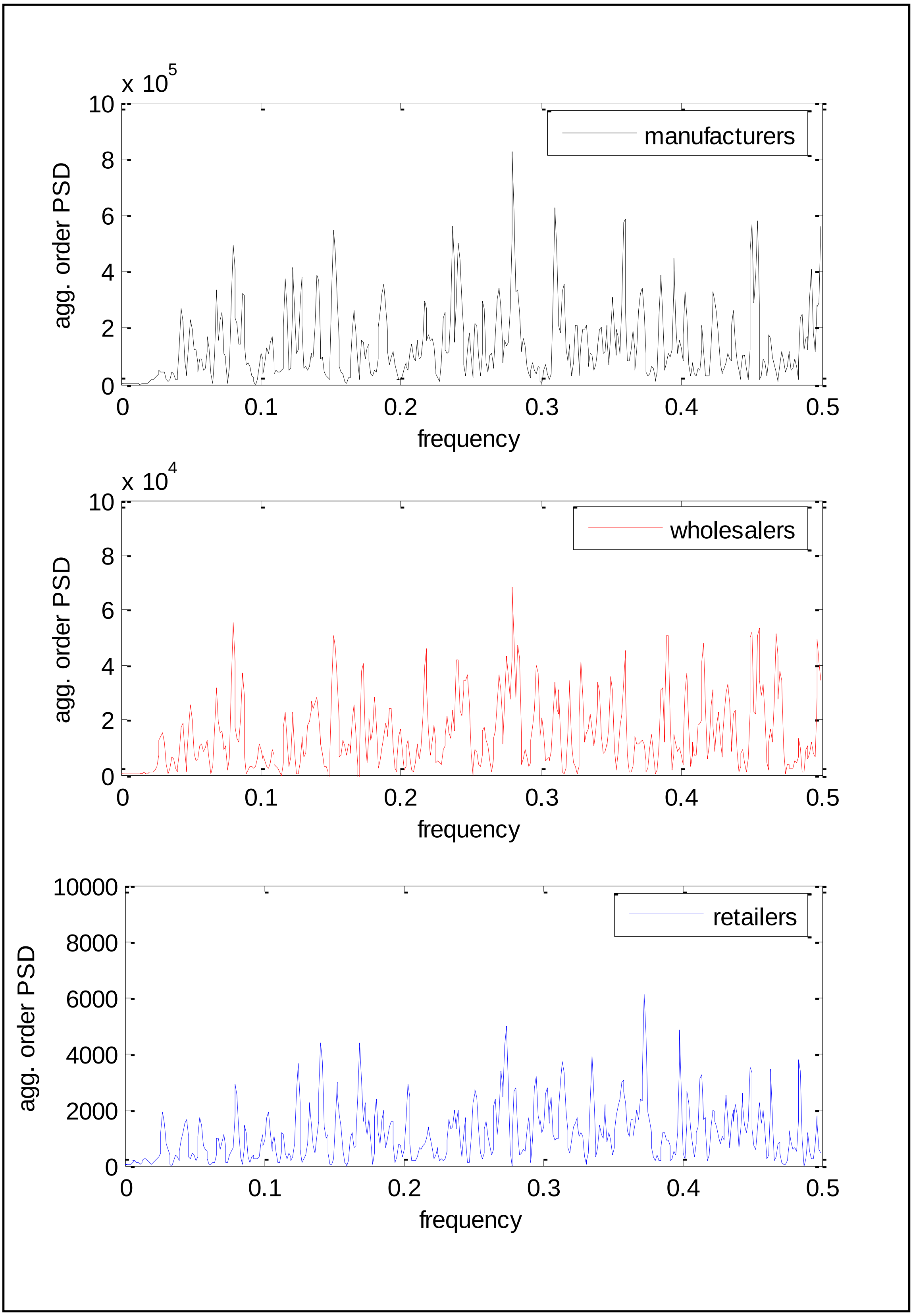


Figure 5.10. Spectral Power Density of Aggregate Orders for Variable S Policy (20 Rtl., 10 Whl., 2 Mnf.)

Table 5.5 gives the statistics of order data for some cases with different number of agents. Note that there are 500 final customers for all cases except the last one, where there are 10000 final customers to demonstrate the scalability of the results. Aggregate order variability and range increase as one moves up in the supply chain. The bullwhip effect in Variable S policy is a consequence of using demand forecasts. The volatility introduced by demand forecasting becomes apparent as one compares Table 5.5 with Table 5.1. Aggregate order average remains the same throughout the supply chain and is not influenced from the introduction of demand forecasting or the number of agents.

Table 5.5. Aggregate Order Statistics for Variable S Policy

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|---|
| RTL. | Mean | 1250.003 | 1249.538 | 1250.17 | 1249.807 | 24997.94 |
| | std. dev. | 82.3996 | 82.4625 | 82.6379 | 83.5178 | 372.3103 |
| | Range | 629.4132 | 682.3579 | 669.77 | 658.8049 | 2917.241 |
| WHL. | Mean | 1249.586 | 1249.799 | 1250.22 | 1249.738 | 24998.26 |
| | std. dev. | 722.7171 | 546.1774 | 305.797 | 246.1062 | 1634.526 |
| | Range | 3034.709 | 3212.442 | 2403.34 | 1946.094 | 12949.79 |
| MNF. | Mean | 1249.733 | 1249.297 | 1250.33 | 1249.518 | 24997.63 |
| | std. dev. | 1656.385 | 1432.203 | 935.482 | 724.2112 | 6431.873 |
| | Range | 7835.358 | 8289.403 | 5787.68 | 4442.687 | 47127.14 |

It is evident from Table 5.5 that the number of agents affects the aggregate order variability significantly. The interaction between the number of agents and the aggregate dynamics has to do with the distribution of demand from the lower echelon to individual agents in the upper echelon, and also with the demand forecasting characteristics of these agents. It has been observed that “the ratio of the number of agents at an echelon to the number of agents at the upper echelon” is critical for the aggregate order variability under Variable S policy. For instance, let’s consider the relationship between the $N_W / N_M$ ratio and the aggregate manufacturer order variability. If the number of wholesalers is very high compared to the number of manufacturers; then the wholesaler orders are distributed almost uniformly to individual manufacturers. Therefore, individual manufacturers face smooth demand, and aggregate manufacturer order variability is not enhanced by the unequal distribution of demand to individuals and the resulting deviation in individual orders. As the number of manufacturers is increased (or the number of wholesalers is decreased), there first emerges a transition regime in which the number of manufacturers becomes comparable to the number of wholesalers. This fact results in such a situation that "the number of wholesalers a randomly selected manufacturer gets order from" varies significantly. Therefore, aggregate manufacturer order variability is high in this transition regime. If the number of manufacturers is further increased (or the number of wholesalers is further decreased), individual manufacturers get low -or possibly no demand- for long time horizons and get comparably high demand intermittently. Individual manufacturers turn out to get high demand intermittently; however demand they receive at these distant time points is more stable than the case in the transition regime. The emergence of this new stable extreme regime reduces the aggregate manufacturer order variability.

These observations will be illustrated below by utilizing individual and aggregate dynamics information in Figures 5.11-5.14. Figures 5.11-5.14 illustrate the demand, order, inventory and Order-Up-To-Level (S) dynamics of a randomly selected manufacturer and the aggregate manufacturer echelon as the number of manufacturers is increased. The number of final customers, retailers, and wholesalers are 500, 20, and 10 for all these cases. Note that the y-axis ranges are tuned to make it simpler to compare the four cases in Figures 5.11-5.14. The scales for aggregate time series plots are the same for all cases; but individual dynamics' scales are reduced proportional to the number of agents –i.e. the scale is halved if the number of manufactures is doubled.

Figure 5.11 shows individual and aggregate manufacturer dynamics for the case of single manufacturer. Since there is only a single manufacturer; aggregate and individual dynamics are the same. The manufacturer gets demand from all of the wholesalers and calculates S by forecasting the demand. Since the manufacturer gets all of the aggregate demand; the individual demand is smooth compared to the cases in the transition regime in Figure 5.12 and Figure 5.13, and therefore is the S. The variability in manufacturer S, order and inventory is low consequently. It is also possible to make some further general observations from the graph. The bullwhip effect can be detected by comparing the demand and order dynamics both at individual and aggregate levels. And, both individual and aggregate S are always higher than the corresponding inventory levels. This gap is eliminated as the number of manufacturers is increased, which can be observed from Figures 5.12-5.14.

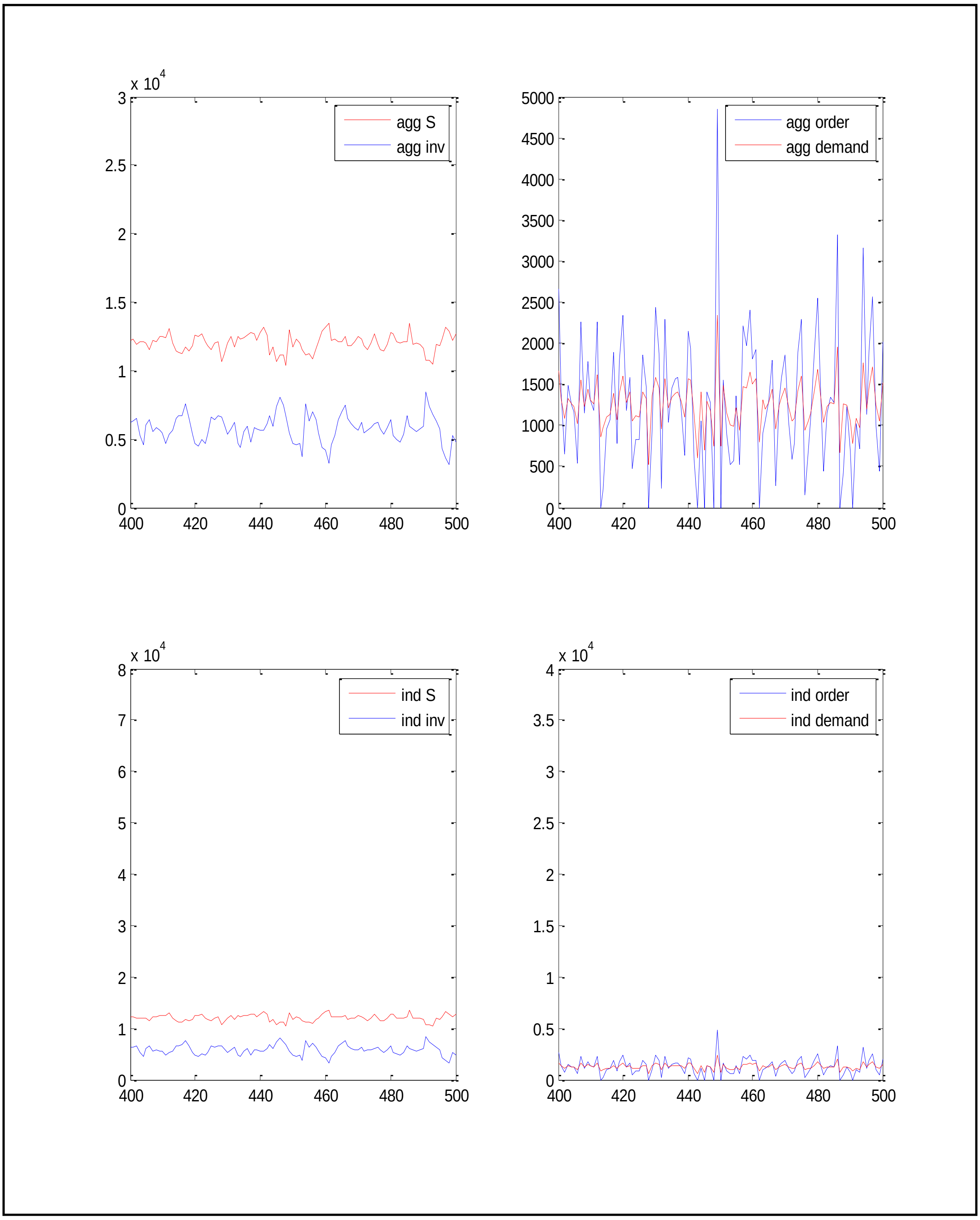


Figure 5.11. Individual and Aggregate Manufacturer Dynamics for Variable S Policy (20 Rtl., 10 Whl., 1 Mnf.)

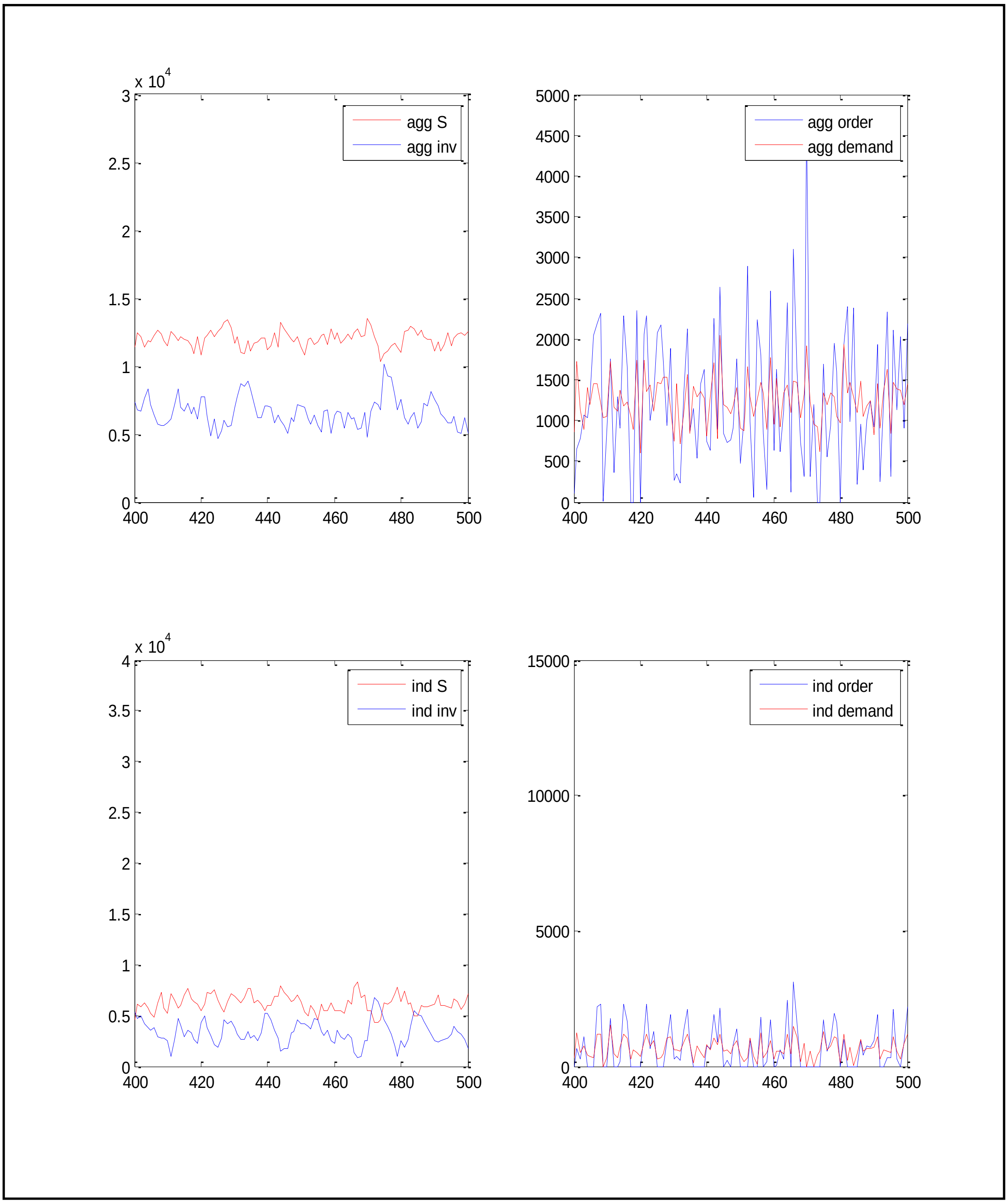


Figure 5.12. Individual and Aggregate Manufacturer Dynamics for Variable S Policy (20 Rtl., 10 Whl., 2 Mnf.)

Figure 5.12 shows aggregate and individual manufacturer dynamics when there are two manufacturers. It has been observed from simulation experiments that the demand is not distributed uniformly between two manufacturers, when there are 10 wholesalers – actually, almost approximately 5 of the 10 wholesalers give order each period. A manufacturer gets demand from only some of the wholesalers ordering at that period, and calculates S by forecasting the demand. The number of wholesalers an individual manufacturer gets order from in a period deviates significantly. Namely, 10 wholesalers-two manufacturers interaction case falls into the transition regime explained above. Individual manufacturer demand has more volatility proportionally, and so does the individual S. It can be observed that aggregate order and inventory are stiffer for the case of two manufacturers, when compared to the one manufacturer case. It has higher variance, as a result of less uniform distribution of demand to individuals and the resulting variability in individual demand forecasts. The bullwhip effect can again be detected by comparing the demand and order dynamics both at individual and aggregate levels. As far as S-inventory interaction is concerned, the gap is reduced both at individual and aggregate levels. There are also some cases where individual inventory exceeds S.

Figure 5.13 reports aggregate and individual manufacturer dynamics for the case of four manufacturers. This case also falls into the transition regime. However it has lower aggregate order variability when compared to the case of two manufacturers; because a transition to intermittent ordering regime is being experienced at the individual level (the dynamics regarding the intermittent ordering regime will be explained in detail below, for the case of 20 manufacturers). This is evident from the fact that average manufacturer ordering period is increased from 1.85 to 2.87 when number of manufacturers is increased to four from two. The intermittency introduced can be detected at the individual order and demand dynamics in Figure 5.13. The bullwhip effect can again be detected by comparing the demand and order dynamics both at individual and aggregate levels. Individual inventory and S levels fluctuate in an interconnected and passing through manner. Individual inventory has higher range and variance when compared to the case of two manufacturers, if the proportion of demand they receive is considered. Aggregate inventory has higher mean and variability when compared to the case of two manufacturers. The underlying mechanism in intermittent ordering regime is explained below.

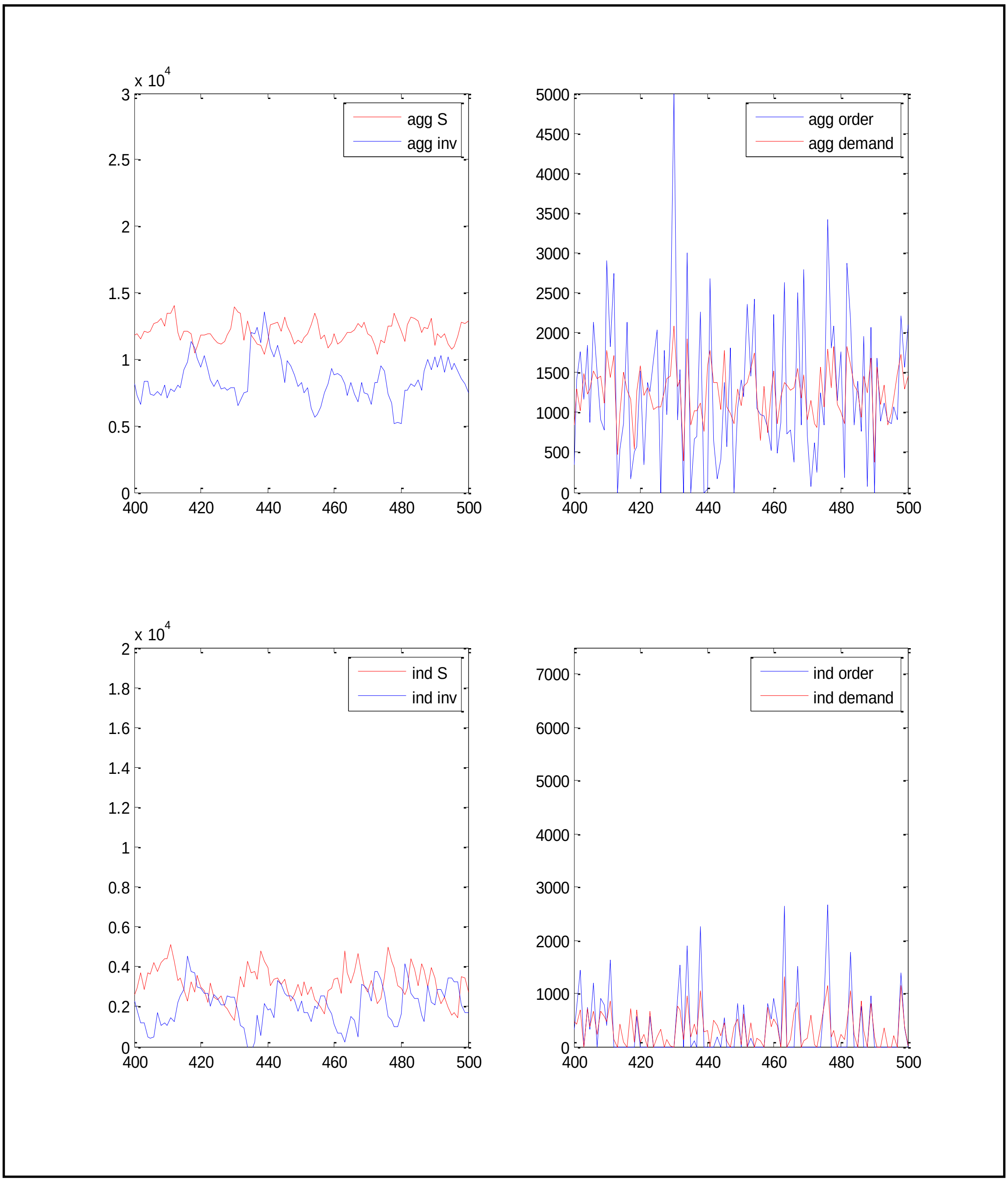


Figure 5.13. Individual and Aggregate Manufacturer Dynamics for Variable S Policy (20 Rtl., 10 Whl., 4 Mnf.)

Figure 5.14 reports individual and aggregate manufacturer dynamics for the case of 20 manufacturers. This case falls into the intermittent ordering regime, which reduces the aggregate order variability. Since the number of manufacturers is high; the individual manufacturers tend to get low or actually zero demand for most of the times. Assuming an agent gets high demand at some time point, its inventory drops and Order-Up-To-Level rises significantly and simultaneously (note that, after lead time units, the inventory is increased by the arrival of the ordered products). Then, S tends to decrease as a result of low or actually zero demand. Inventory tends to decrease - after the arrival of the ordered goods- slowly as long as low orders are received. Assuming a high order is received again at some time point, this process is repeated. On the other hand, assuming no such high order is received, demand forecast drops to its level according to low demand and stays thereabout and provides smooth forecasts. It can be observed from Figure 5.14 that comparably high orders are received at some time points and the intermittent ordering attitude emerges. This is evident from the fact that the average ordering period is increased to 10.52, when there are 20 manufacturers. The intermittency introduced can be detected at the individual demand and order dynamics. Although this attitude enhances the individual order variability, it provides low order variability at the aggregate level. The underlying cause is the fact that high demand points occur infrequently and orders are given mostly at these points. And in the low (or zero) demand intervals, individual inventory and S exhibit smooth dynamics which prevents overreaction to low demand in these intervals. Intermittent individual ordering provides lower order variability at the aggregate level. On the other hand, aggregate inventory variability is increased in the intermittency regime. Since individual inventories are stock variables, intermittent demands and orders have prolonged effects and they amplify the variability at aggregate level. Aggregate inventory mean is higher than the mean in other cases. The bullwhip effect can again be detected by comparing the demand and order dynamics both at individual and aggregate levels.

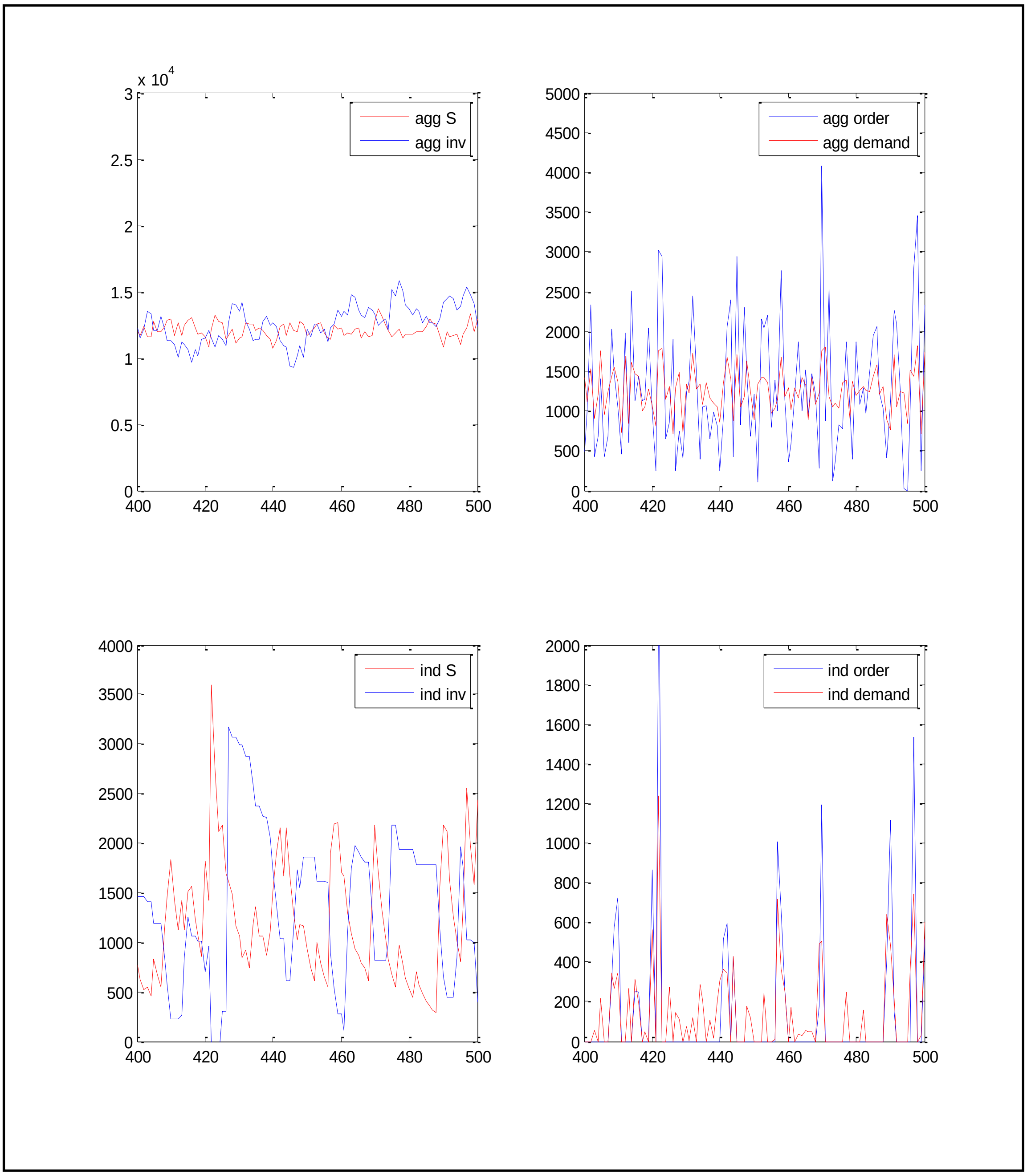


Figure 5.14. Individual and Aggregate Manufacturer Dynamics for Variable S Policy (20 Rtl., 10 Whl., 20 Mnf.)

The transition from smooth demand/order regime to intermittent demand/order regime can be better seen in Figure 5.15 and Figure 5.16. Figure 5.15 and Figure 5.16 illustrate individual demand and order patterns respectively, without scaling the graphs according to the number of agents and within a narrower time scale.

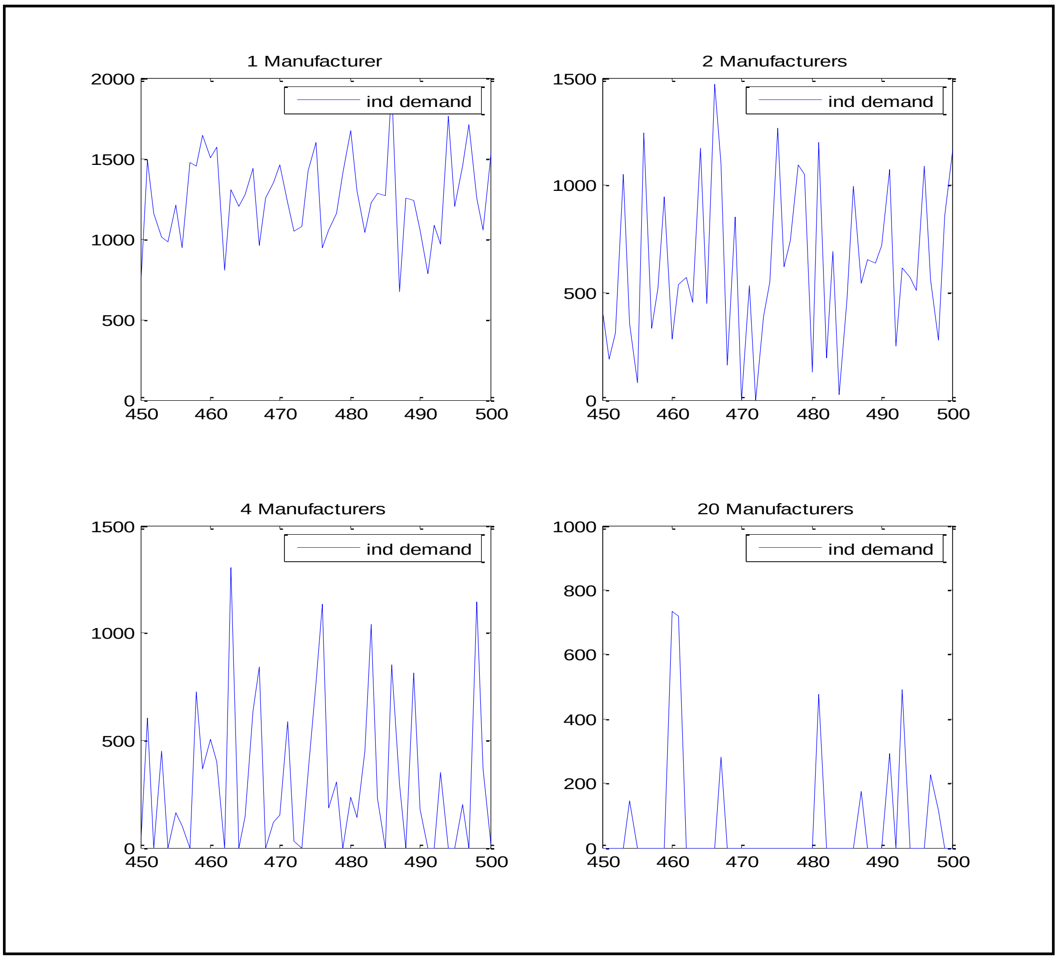


Figure 5.15. Individual Manufacturer Demand Pattern for Variable S Policy (20 Rtl., 10 Whl., $N_M$ Mnf.)

To summarize the discussion about the interaction between the number of agents and the aggregate order dynamics, “the ratio of the number of agents at the lower echelon to the number of agents at the upper echelon” is critical for aggregate order variability at the upper echelon. If the number of agents at the lower echelon is very high compared to the number of agents at the upper echelon; the demand is distributed almost uniformly to individual agents. Therefore, aggregate order variability is low. As this ratio decreases; the aggregate order variability increases first. Since the number of agents demanding from an agent at the upper echelon has high deviation, individual demand forecasts have high variability. Therefore, aggregate variability is high in the transition regime until the ratio decreases down to a critical value. If ratio is further decreased, individual agent orders turn out to exhibit intermittent pattern; which reduces the aggregate order variability -but increases aggregate inventory variability on the other hand. It is important to note that a general rule for the detection of the critical ratio that applies to all cases could not be found at the current state of the research.

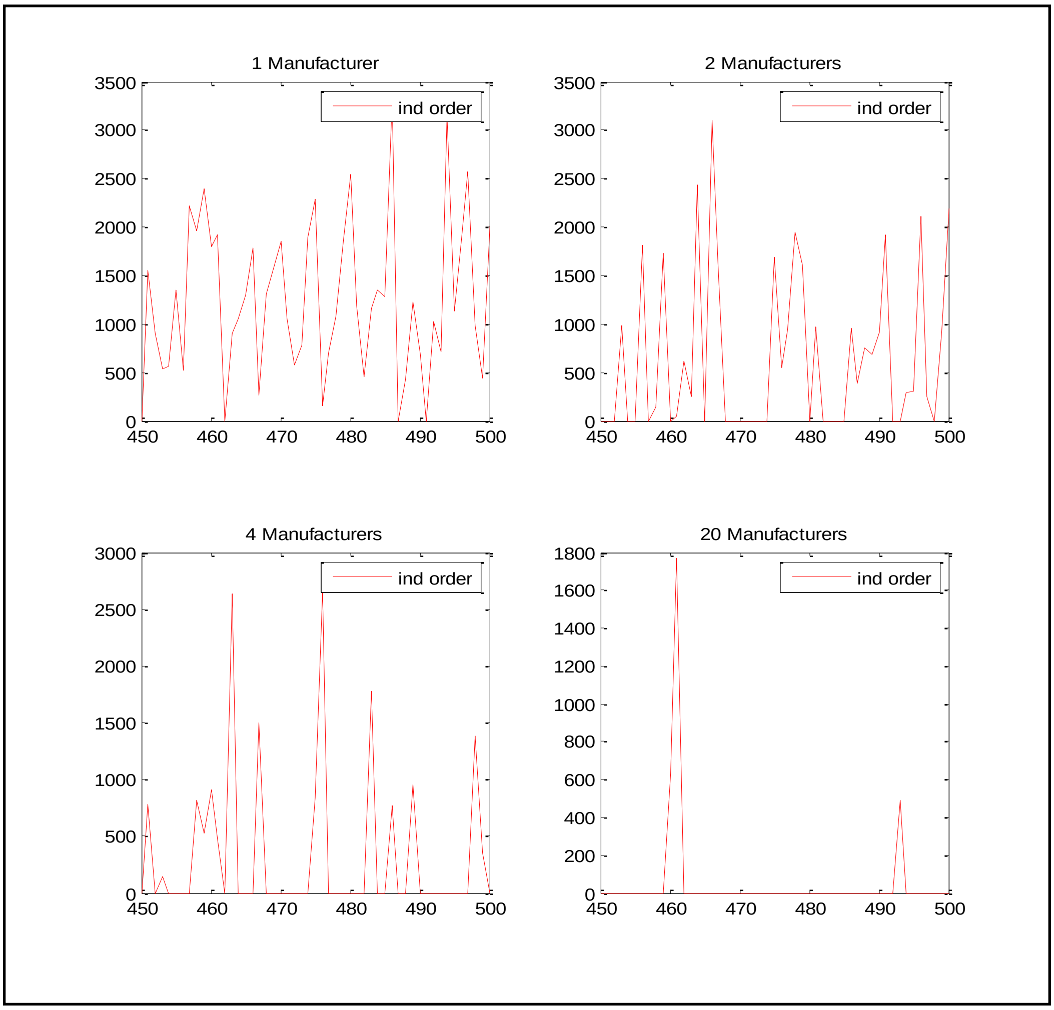


Figure 5.16. Individual Manufacturer Order Pattern for Variable S Policy (20 Rtl., 10 Whl., $N_M$ Mnf.)

The reflection of changes in individual demand patterns on aggregate dynamics can be visualized by “aggregate active S” dynamics, where “aggregate active S” is the sum of the Order-Up-To-Levels of the agents that give order –the ones not giving order at a given period are excluded. Figure 5.17 compares aggregate active S values for the four cases illustrated above. It can be observed that aggregate S dynamics are robust to the number of agents. However, active aggregate S dynamics reflect the changes occurring at the individual level. Because it is only the agents ordering that change the aggregate order patterns. For the single manufacturer case, active aggregate S is equal to aggregate S except at the infrequent points where the manufacturer does not give order. For the case of two manufacturers, there are instances where none, just one or both of the manufacturers give order. Therefore, the variability in active aggregate S is higher compared to the first case. For the case of four manufacturers, aggregate active S fluctuates around a lower value but with less range and variance when compared to the second one. For the case of 20 manufacturers, active aggregate S mean, range and variance are further reduced.

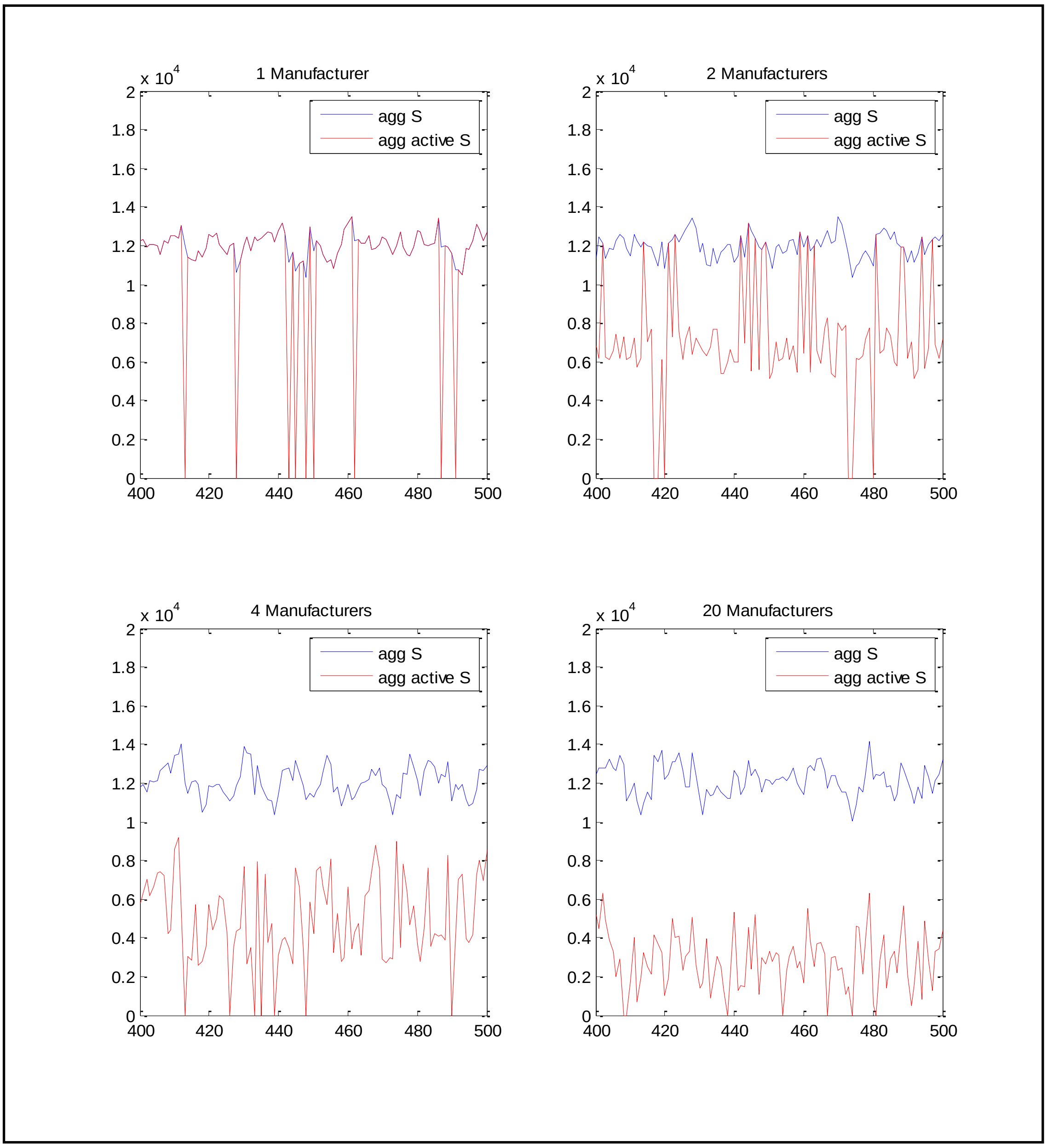


Figure 5.17. Comparison of Aggregate Active S Dynamics for Variable S Policy (20 Rtl., 10 Whl., $N_M$ Mnf.)

Tables 5.6-5.8 report the changes in aggregate order variance, range, and average when number of agents at only one echelon is varied -keeping other number of agents constant. The statistics in Tables 5.6-5.8 are consistent with the discussion above about the relationship between the number of agents and aggregate order dynamics. Notice that the transition from individual continuous demand/order regime to intermittent demand/order regime is more effective on aggregate statistics for higher supply chain echelons.

Table 5.6. Aggregate Order Statistics for the Effect of Number of Retailers under Variable S Policy ($N_R$ Retailers, 1 Wholesalers, 1 Manufacturer)

| | | 2 Rtl. | 20 Rtl. | 100 Rtl. | 150 Rtl. | 200 Rtl. | 500 Rtl. | 1000 Rtl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1249.91 | 1249.87 | 1250.3 | 1250.18 | 1249.89 | 1248.8398 | 1248.7 |
| | std. dev. | 82.7325 | 82.5643 | 82.5778 | 83.8078 | 81.7634 | 76.9964 | 75.5137 |
| | Range | 637.449 | 694.885 | 660.767 | 667.088 | 615.809 | 641.0856 | 583.863 |
| WHL. | Mean | 1249.94 | 1249.82 | 1250.27 | 1250.17 | 1249.77 | 1248.7755 | 1248.76 |
| | std. dev. | 217.396 | 216.944 | 217.484 | 219.386 | 213.58 | 199.7367 | 195.894 |
| | Range | 1687.45 | 1823.65 | 1807.09 | 1711.09 | 1658.3 | 1745.0856 | 1447.86 |
| MNF. | Mean | 1249.99 | 1249.57 | 1250.04 | 1250.14 | 1249.49 | 1248.585 | 1248.98 |
| | std. dev. | 673.32 | 670.762 | 673.204 | 678.156 | 662.744 | 621.4767 | 611.833 |
| | Range | 3977.37 | 4116.4 | 4870.19 | 3806.72 | 3979.89 | 4077.6136 | 3786.28 |

Table 5.6 shows the effect of the number of retailers, when there is one wholesaler and one manufacturer. The aggregate retailer order variance does not change significantly until there are 200 retailers. Aggregate retailer order variance drops for higher number of retailers. The aggregate retailer order dynamics affect aggregate wholesaler and manufacturer order dynamics. Therefore, the trend in aggregate retailer order variability is preserved at higher echelons. It is also important to note that the range statistics is not as informative as the variance statistics.

Table 5.7. Aggregate Order Statistics for the Effect of Number of Wholesalers under Variable S Policy (50 Retailers, $N_W$ Wholesalers, 1 Manufacturer)

| | | 1 Whl. | 2 Whl. | 4 Whl. | 10 Whl. | 20 Whl. | 50 Whl. | 100 Whl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1250.41 | 1249.93 | 1249.8243 | 1249.96 | 1249.72 | 1249.9 | 1250.17 |
| | std. dev. | 83.4078 | 83.9467 | 84.167 | 84.5855 | 83.4247 | 82.8366 | 83.2483 |
| | Range | 691.131 | 666.747 | 686.9901 | 646.096 | 728.143 | 616.425 | 731.575 |
| WHL. | Mean | 1250.34 | 1249.95 | 1249.8255 | 1249.91 | 1249.79 | 1249.92 | 1250.16 |
| | std. dev. | 219.039 | 220.631 | 231.9253 | 247.783 | 245.795 | 237.67 | 226.273 |
| | Range | 1789.13 | 1689.31 | 1933.5756 | 1805.8 | 2034.98 | 1672.98 | 1736.08 |
| MNF. | Mean | 1250.1 | 1249.99 | 1249.8232 | 1249.86 | 1249.77 | 1250.19 | 1249.95 |
| | std. dev. | 676.821 | 681.999 | 712.0278 | 754.715 | 749.872 | 728.155 | 693.317 |
| | Range | 3967.56 | 3919.55 | 4672.0379 | 4156.38 | 4704.65 | 4061.28 | 4307.67 |

Table 5.7 shows the effect of the number of wholesalers, when there are 50 retailers and one manufacturer. The aggregate wholesaler order variance increases until there are 10 wholesalers and then decreases again. The trend in aggregate wholesaler order variability is preserved at the manufacturer echelon.

Table 5.8. Aggregate Order Statistics for the Effect of Number of Manufacturers under Variable S Policy (20 Retailers, 10 Wholesalers, $N_M$ Manufacturers)

| | | 1 Mnf. | 2 Mnf. | 4 Mnf. | 8 Mnf. | 10 Mnf. | 20 Mnf. | 40 Mnf. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1249.7069 | 1250.17 | 1250.3 | 1249.77 | 1249.91 | 1249.96 | 1249.82 |
| | std. dev. | 82.2088 | 82.6379 | 83.0875 | 83.6982 | 83.3614 | 83.0135 | 83.15 |
| | Range | 752.1033 | 669.77 | 660.941 | 693.921 | 669.689 | 675.445 | 645.31 |
| WHL. | Mean | 1249.8954 | 1250.22 | 1250.42 | 1249.66 | 1249.87 | 1249.78 | 1249.84 |
| | std. dev. | 303.8436 | 305.797 | 305.598 | 305.718 | 305.83 | 304.936 | 306.15 |
| | Range | 2209.4515 | 2403.34 | 2251.35 | 2410.79 | 2285.6 | 2421.17 | 2345.84 |
| MNF. | Mean | 1250.1021 | 1250.33 | 1250.85 | 1249.1 | 1247.72 | 1242.93 | 1243.81 |
| | std. dev. | 891.5644 | 935.482 | 908.019 | 886.626 | 848.051 | 826.797 | 828.60 |
| | Range | 4814.822 | 5787.68 | 5202.68 | 6047.3 | 5324.04 | 5648.69 | 5621.61 |

Table 5.8 shows the effect of the number of manufacturers, when there are 20 retailers and 10 wholesalers. The aggregate manufacturer order variance increases when the number of manufacturers is increased to two and declines thereafter. The decrease in variability saturates for high number of agents.

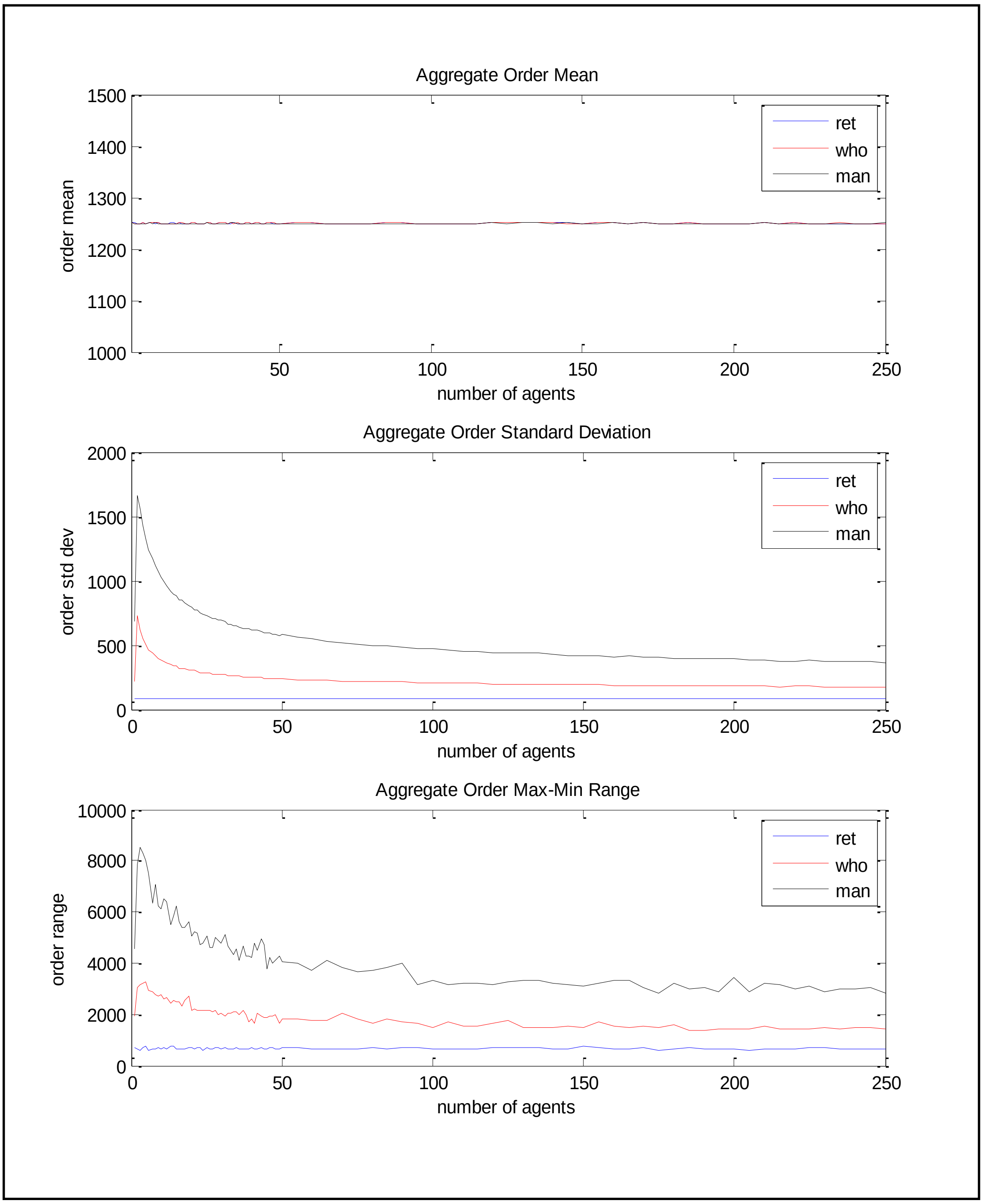


Figure 5.18. Aggregate Order Statistics as Functions of the Number of Agents (All Echelons Equal) for Variable S Policy

Figure 5.18 shows how aggregate order mean, standard deviation and max-min range change as a function of the number of agents in case where the number of retailers, wholesalers, and manufacturers are the same. Aggregate order mean is not affected from the number of agents. The aggregate wholesaler and manufacturer standard deviation increase significantly when the number of agents is increased to two. Then, it decreases smoothly as the number of agents is further increased. Aggregate retailer order variance is not affected as much as the wholesaler and manufacturer order variability. The observations on the effect of the number of agents at all echelons are similar to the effect of the number of agents at just one echelon. Variance is low when there is only single agent at each echelon. Variance makes a maximum when there are two agents because of unequal distribution of demand to individuals. Variance is decreased for higher number of agents as a result of transition to intermittent demand regime. Aggregate order range statistics exhibit similar patterns but with some noise.

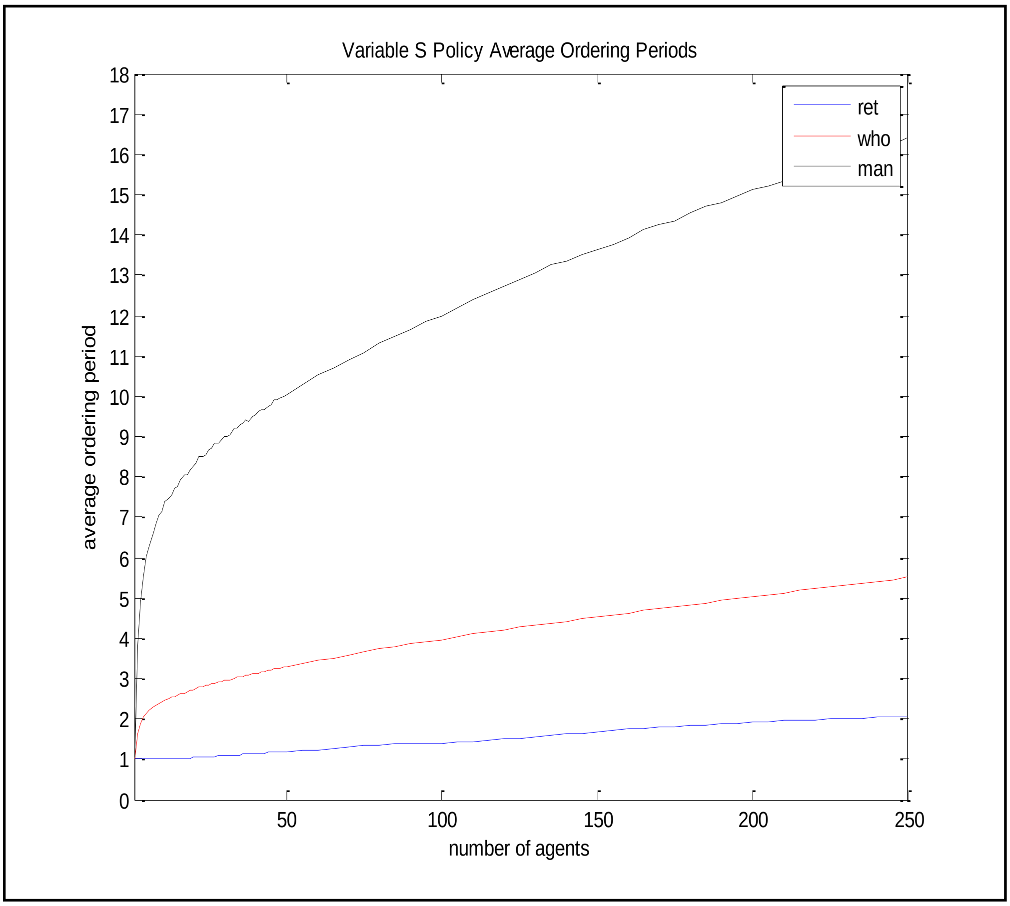


Figure 5.19. Average Echelon Ordering Periods as Functions of the Number of Agents (All Echelons Equal) for Variable S Policy

Individual order intermittency is one of the underlying causes of the trends in Figure 5.18 when coupled with demand forecasting as explained above. The average ordering period is a statistics that is most informative about the intermittency of individual orders. Figure 5.19 shows how the average ordering periods at different supply chain echelons change as a function of the number of agents. Average ordering period increases as the number of agents is increased or as one moves up in the supply chain due to the decrease in the probability of receiving significant demand at a given time point.

Table 5.9. Variable S Policy, Aggregate-Individual Order Variability Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.4517 | 0.2701 | 0.1132 | 0.1168 | 0.0788 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.4754 | 0.2781 | 0.1764 | 0.1259 | 0.0832 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.6277 | 0.4049 | 0.5579 | 0.1702 | 0.1079 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

As far as the relationship between individual and aggregate order dynamics is concerned, Table 5.9 summarizes the comparison of the aggregate order dynamics with the individual firm order dynamics. The variability reduction in retailers is not significantly different from the case for Fixed S policy due to smooth individual demand patterns in retailers. However, it can be observed that aggregation -of the individual outputs- removes a percentage of the variability existing in individual orders of wholesalers and manufacturers, which is very small when compared to the reduction in the Fixed S policy. Under Variable S policy, the aggregate order information is transmitted with some modification due to the updating of the Order-Up-To-Level. Therefore, there is a loss of degrees of freedom at a supply chain echelon as a result of its partial dependence on the aggregate order of the lower echelon. Namely, individual agent orders have some degree of cross-correlation such that the aggregate order information at the lower echelon is partially conserved at the upper echelon. Therefore the variability reduction due to averaging under Variable S policy is very small compared to the reduction under Fixed S policy for high echelons.

Figure 5.20 shows the aggregate echelon inventory dynamics for the case of 20 retailers, 10 wholesalers, and two manufacturers. It can be observed that the bullwhip effect exists for inventory data and that inventory exhibits smoother and oscillatory patterns compared to the rate variable order. This is most obvious from the manufacturer's plot. Figures for other experiments with Variable Order-Up-To-Level policy can be found in Appendix D.

Figure 5.21 shows the power spectral densities of aggregate inventories for the case of 20 retailers, 10 wholesalers and two manufacturers. Spectral power is spread over a large range of low-to-medium scale frequencies at the retailer echelon. There are multiple dominant periodicity candidates; however the powers of these candidates do not differ very much. Spectral powers of aggregate wholesaler and manufacturer inventories are higher and more concentrated around low frequency candidates. Table D.4 in Appendix D shows the spectral analysis results of inventory data for various settings of number of agents. It demonstrates that total and dominant spectral powers increase in the upward echelon direction. The increase in dominant spectral power is an indicator of the bullwhip effect in terms of oscillation amplitude amplification. Both total and dominant spectral powers are higher on the average for the Variable S policy when compared to the Fixed S policy. This increase is caused by the demand forecasting. As a further command, there are multiple periodicities in inventory data; however the dominance of a frequency at an echelon is not preserved at upper echelons as in the Fixed S policy. The dominant periods are not robust, i.e. they are dependent on the random seed; however they are bounded in a region as can be seen from Figure 5.21 and Appendix D.

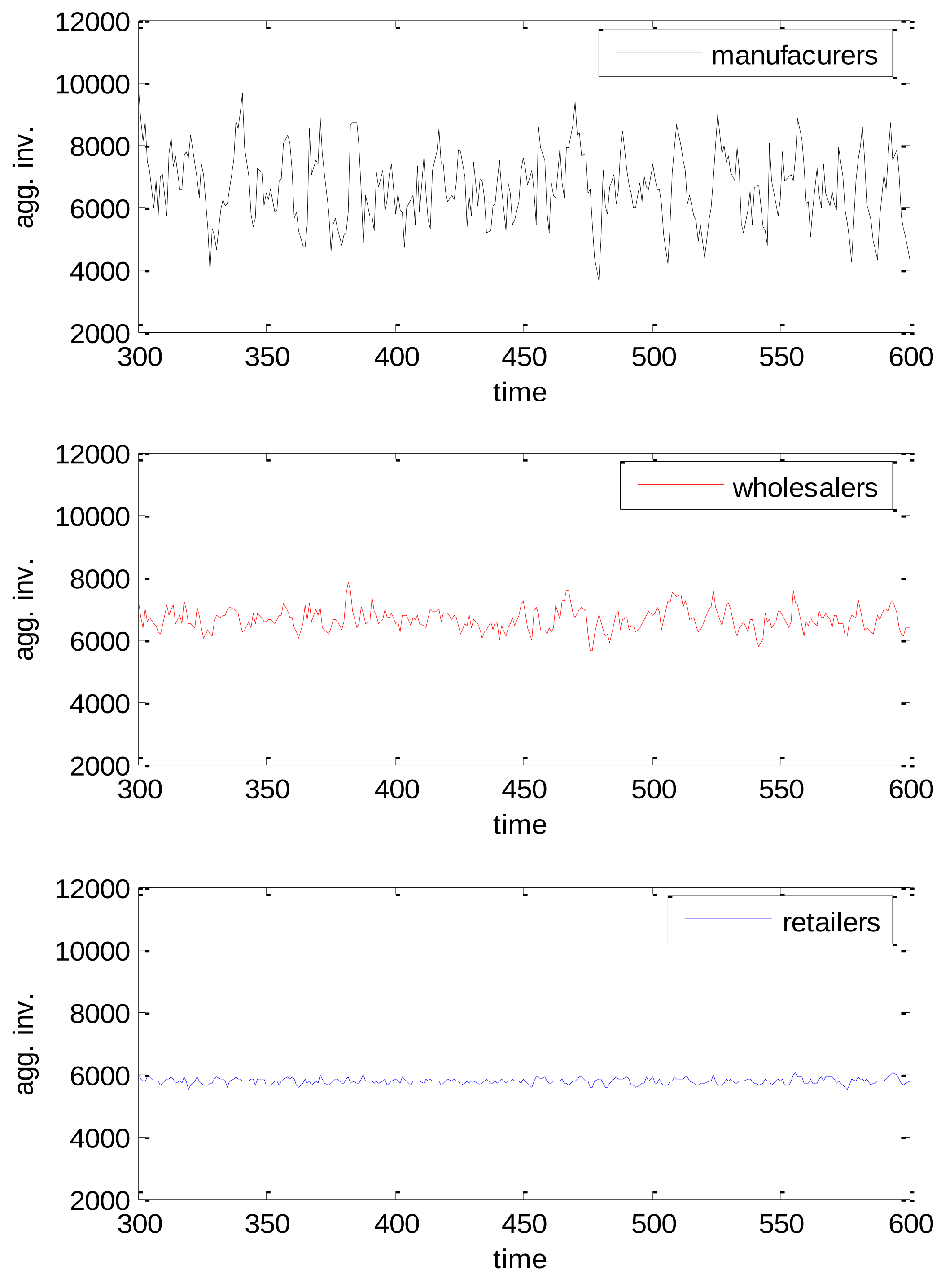


Figure 5.20. Aggregate Inventory Dynamics for Variable S Policy

(20 Rtl., 10 Whl., 2 Mnf.)

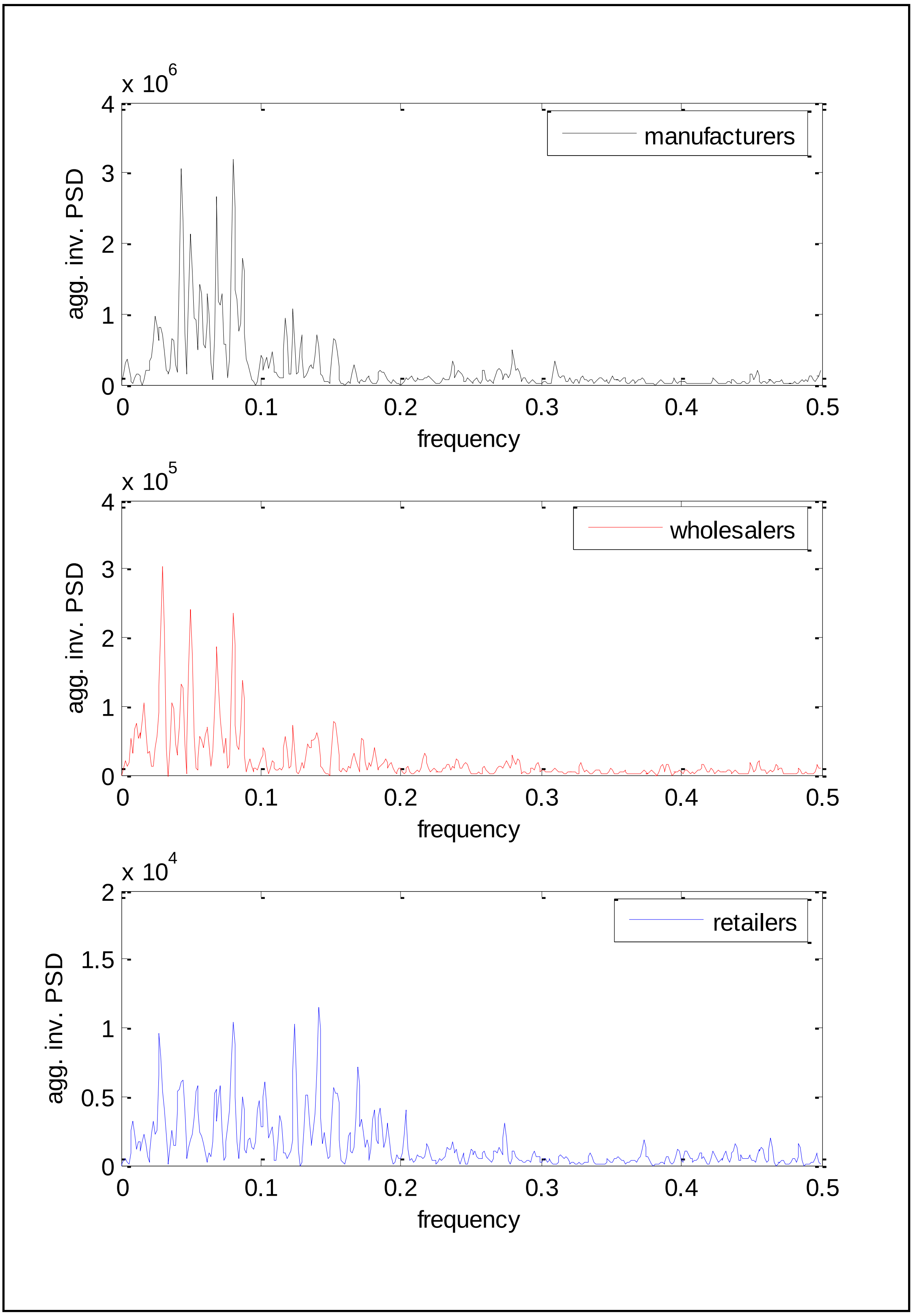


Figure 5.21. Spectral Power Density of Aggregate Inventories for Variable S Policy (20 Rtl., 10 Whl., 2 Mnf.)

Table 5.10 gives the statistics of inventory data for the cases in Table 5.5. Aggregate inventory variance and range increase as one moves up in the supply chain. The bullwhip effect in aggregate inventory is a consequence of using demand forecasts. The volatility introduced by demand forecasting becomes apparent as one compares Table 5.10 with Table 5.3.

Table 5.10. Aggregate Inventory Statistics for Variable S Policy

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|---|
| RTL. | Mean | 5749.918 | 5742.815 | 5753.49 | 5825.794 | 115010.6 |
| | std. dev. | 94.336 | 105.2652 | 96.7337 | 97.8595 | 415.9632 |
| | Range | 2075.514 | 1489.554 | 869.637 | 751.3517 | 3303.682 |
| WHL. | Mean | 6587.174 | 6952.276 | 6605.97 | 6686.965 | 121024.2 |
| | std. dev. | 780.3295 | 691.4554 | 360.802 | 305.8909 | 1785.398 |
| | Range | 6739.474 | 7400.212 | 3007.06 | 2613.279 | 17335.49 |
| MNF. | Mean | 10089.47 | 13032.05 | 6589.52 | 6998.154 | 132443.7 |
| | std. dev. | 2446.189 | 2538.665 | 1166.05 | 966.7248 | 7573.515 |
| | Range | 18046.46 | 20043.2 | 9406.68 | 7610.843 | 65016.89 |

It is evident from Table 5.10 that the number of agents affects not only the variability but also the mean of aggregate inventory significantly. Aggregate inventory mean increases as the number of agents in an echelon increases. The relationship between the number of agents and aggregate inventory variability is different from the relationship in aggregate orders. Although individual intermittent ordering regime provides low variability in aggregate orders; it is not the case for aggregate inventories. Aggregate inventory variability has a monotonically increasing relationship with the number of agents in the same echelon, ceteris paribus. Intermittent individual demand and order patterns have prolonged effects on individual inventory through augmenting the oscillation period and amplitude as in Figures 5.11-5.14. The oscillatory components at the individual inventories are reflected at the aggregate inventories. Thus, aggregate echelon inventory variability increases as the number of agents in that echelon increases. The interaction between the number of agents at the lower echelon and the aggregate inventory variability at the upper echelon is through the variability in aggregate orders of the lower echelon. This is reasonable since the aggregate inventory at the upper echelon is directly affected from the order dynamics at the lower echelon. On the other hand, the interaction between the number of agents at the higher echelon and the aggregate inventory variability at the lower echelon is through the variability in the individual inventories of the upper echelon. This is also reasonable since the aggregate inventory at the lower echelon is affected from the arrival of the orders given to the upper echelon, which interferes with the inventory dynamics of firms at the upper echelon.

Tables 5.11-5.13 report the changes in aggregate inventory variance, range, and mean when number of agents at each echelon is varied -keeping other number of agents constant.

Table 5.11. Aggregate Inventory Statistics for the Effect of Number of Retailers under Variable S Policy ($N_R$ Retailers, 1 Wholesalers, 1 Manufacturer)

| | | 2 Rtl. | 20 Rtl. | 100 Rtl. | 150 Rtl. | 200 Rtl. | 500 Rtl. | 1000 Rtl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 5750.9 | 5759.51 | 6030.54 | 6258.51 | 6545.95 | 8705.94 | 14915.0 |
| | std. dev. | 91.528 | 92.8971 | 101.244 | 105.364 | 108.040 | 121.412 | 126.264 |
| | Range | 709.30 | 710.306 | 789.489 | 806.73 | 822.149 | 1082.4 | 954.205 |
| WHL. | Mean | 5750.9 | 5751.21 | 5751.19 | 5749.97 | 5750.01 | 5747.87 | 5747.48 |
| | std. dev. | 223.07 | 224.999 | 231.479 | 232.915 | 229.340 | 224.245 | 219.092 |
| | Range | 1696.4 | 1850.59 | 1927.74 | 1884.67 | 1851.92 | 1861.03 | 1795.14 |
| MNF. | Mean | 5761.5 | 5761.78 | 5762.20 | 5759.36 | 5757.17 | 5753.72 | 5752.66 |
| | std. dev. | 697.28 | 701.358 | 727.735 | 732.171 | 724.235 | 710.693 | 695.562 |
| | Range | 5865.6 | 6051.50 | 6238.10 | 5431.29 | 6338.36 | 5951.54 | 5669.97 |

Table 5.11 shows the effect of the number of retailers, when there is one wholesaler and one manufacturer. The aggregate retailer inventory variance increases as the number of retailers increases. The tendencies in aggregate wholesaler and manufacturer inventory variances are to first increase and then to decrease as in the aggregate wholesaler and manufacturer order cases. However the changes in aggregate inventory variability of wholesaler and manufacturer echelons are relatively quite small. Aggregate retailer inventory mean is augmented by the increase in the number of retailers.

Table 5.12. Aggregate Inventory Statistics for the Effect of Number of Wholesalers under Variable S Policy (50 Retailers, $N_W$ Wholesalers, 1 Manufacturer)

| | | 1 Whl. | 2 Whl. | 4 Whl. | 10 Whl. | 20 Whl. | 50 Whl. | 100 Whl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 5834.5 | 5832.4 | 5834.36 | 5830.73 | 5826.76 | 5786.30 | 5734.71 |
| | std. dev. | 96.612 | 96.708 | 96.6425 | 96.8023 | 98.3246 | 105.767 | 112.58 |
| | Range | 848.35 | 771.24 | 720.644 | 871.193 | 769.230 | 879.898 | 1005.33 |
| WHL. | Mean | 5751.6 | 5753.9 | 5808.24 | 6094.68 | 6690.70 | 8639.08 | 11921.68 |
| | std. dev. | 229.97 | 231.93 | 243.038 | 272.189 | 302.650 | 348.861 | 392.37 |
| | Range | 1883.2 | 1920.9 | 2270.96 | 2144.07 | 2205.28 | 2763.69 | 3397.83 |
| MNF. | Mean | 5762.8 | 5759.4 | 5766.54 | 5768.39 | 5766.22 | 5762.52 | 5759.09 |
| | std. dev. | 716.30 | 723.18 | 746.653 | 804.757 | 834.061 | 836.502 | 811.88 |
| | Range | 5674.3 | 5215.2 | 5910.99 | 6397.60 | 6382.22 | 6502.93 | 6567.67 |

Table 5.12 shows the effect of the number of wholesalers, when there are 50 retailers and one manufacturer. The aggregate wholesaler inventory variance increases as the number of wholesalers increases. The tendency in aggregate retailer inventory variance is to increase with the rising aggregate wholesaler inventory variability. The aggregate manufacturer inventory variance has the tendency first to increase and then to decrease, as in the case of aggregate wholesaler orders. Aggregate wholesaler inventory mean is augmented by the increase in the number of wholesalers.

Table 5.13. Aggregate Inventory Statistics for the Effect of Number of Manufacturers under Variable S Policy (20 Retailers, 10 Wholesalers, $N_M$ Manufacturers)

| | | 1 Mnf. | 2 Mnf. | 4 Mnf. | 8 Mnf. | 10 Mnf. | 20 Mnf. | 40 Mnf. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 5754.06 | 5753.49 | 5752.60 | 5751.84 | 5752.73 | 5753.42 | 5752.61 |
| | std. dev. | 95.3231 | 96.7337 | 97.2631 | 98.6967 | 97.6855 | 98.9754 | 96.4815 |
| | Range | 1099.81 | 869.637 | 1057.51 | 1195.48 | 968.981 | 1076.59 | 1168.57 |
| WHL. | Mean | 6614.46 | 6605.97 | 6575.68 | 6496.67 | 6483.00 | 6433.73 | 6382.93 |
| | std. dev. | 364.191 | 360.802 | 381.428 | 413.336 | 414.607 | 426.016 | 415.34 |
| | Range | 3261.66 | 3007.06 | 3593.24 | 4136.74 | 4186.93 | 4171.15 | 3816.83 |
| MNF. | mean | 5805.55 | 6589.52 | 8307.97 | 11723.6 | 13390.5 | 21792.9 | 37567.3 |
| | std. dev. | 993.824 | 1166.05 | 1319.83 | 1568.80 | 1653.22 | 2161.39 | 2681.49 |
| | range | 8319.18 | 9406.68 | 11263.0 | 11901.9 | 12600.5 | 17563.7 | 20016.9 |

Table 5.13 shows the effect of the number of manufacturers, when there are 20 retailers and 10 wholesalers. The aggregate manufacturer inventory variance increases as the number of manufacturers increases. The tendency in aggregate wholesaler inventory variance is to increase with the rising aggregate manufacturer inventory variability. However, note that the increase is reversed for the 40 manufactures' case. Aggregate manufacturer inventory mean is augmented by the increase in the number of manufacturers.

Figure 5.22 shows how aggregate inventory mean, variability and range change as a function of the number of agents in the case where the number of retailers, wholesalers, and manufacturers are the same. Aggregate inventory average increases as the number of agents increases, which reveals that the system holds more inventory on average for a supply chain with higher number of firms. Aggregate wholesaler and manufacturer inventory standard deviation increase significantly for low number of agents. Then, they decrease smoothly as the number of agents is further increased. Aggregate retailer order variability is not affected as much as the wholesaler and manufacturer variability. Aggregate inventory variability exhibits characteristics similar to the aggregate order variability when the number of agents at all echelons are the same. However, it is important to note that the trends in aggregate inventory variability as a response to the changes in the number of agents at one specific echelon –ceteris paribus- is different from the case in aggregate order variability.

Table 5.14 summarizes the comparison of the system level aggregate inventory dynamics with the individual firm inventory. The observations from the comparison of inventory dynamics are similar to the ones for order dynamics. Aggregation -of the individual outputs- removes a percentage of the variability existing in individual order dynamics, which is very small when compared to the reduction in the Fixed S policy at wholesaler and manufacturer echelons, but still higher than the reference reduction.

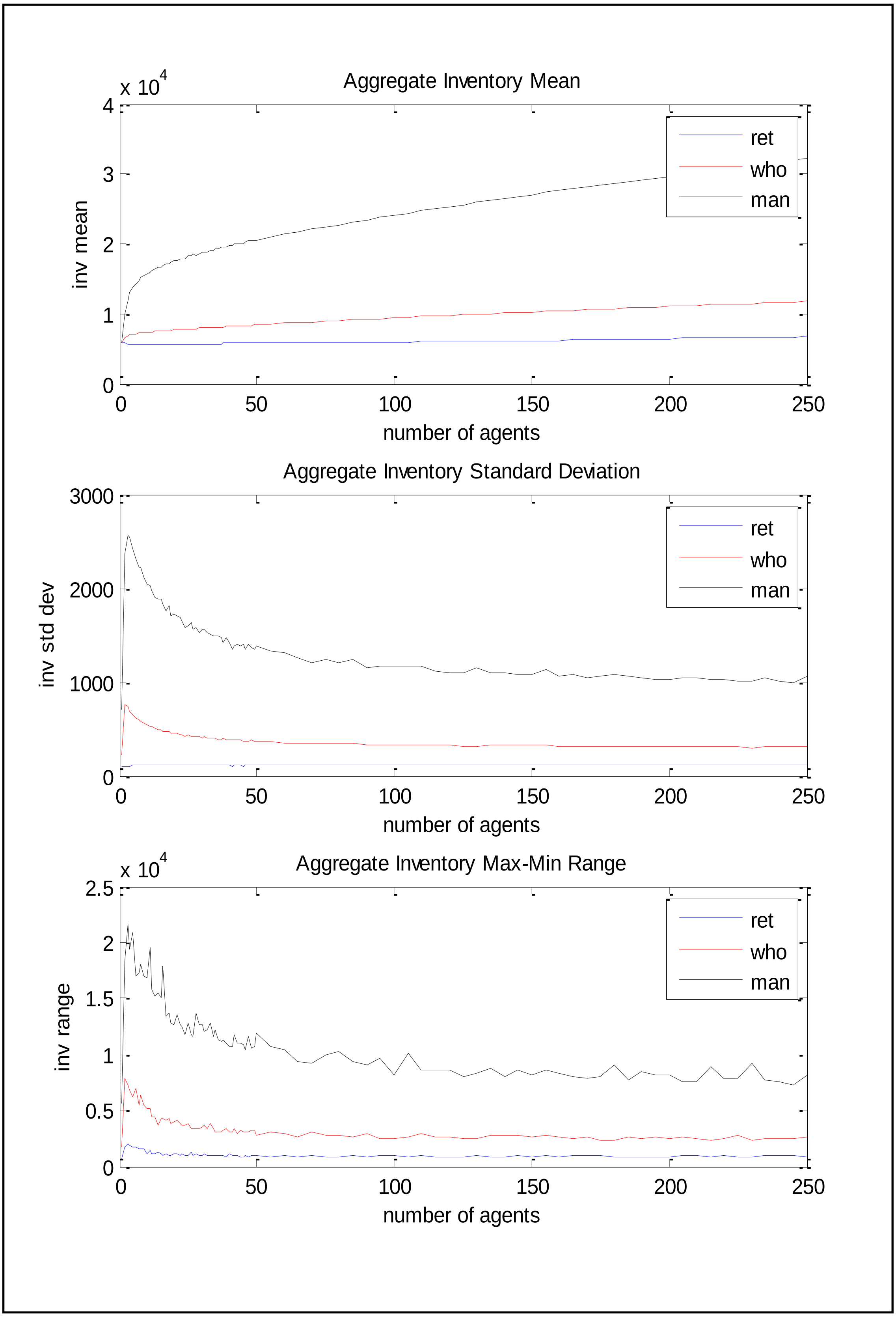


Figure 5.22. Aggregate Inventory Statistics as Functions of the Number of Agents (All Echelons Equal) for Variable S Policy

Table 5.14. Variable S Policy, Aggregate-Individual Inventory Variability Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.448764 | 0.309974 | 0.118779 | 0.077152 | 0.035653 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.341832 | 0.23799 | 0.134365 | 0.112487 | 0.048378 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.492558 | 0.350221 | 0.430622 | 0.299428 | 0.131969 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

To summarize the observations about the Variable Order-Up-To-Level policy, the final customer demand variability is amplified through the supply chain as a result of demand forecasting and rising lead times. Namely bullwhip effect emerges in aggregate orders and inventories. Average ordering period, aggregate order variability and spectral power, aggregate inventory variability and spectral power are amplified as one moves up in the supply chain. All these measures are higher when compared to the Fixed S policy as a result of demand forecasting. Aggregate order average is not influenced from the number of agents or the level in the supply chain.

As far as the relationship between the supply chain topology and the aggregate order dynamics is concerned, “the ratio of the number of agents at the lower echelon to the number of agents at the upper echelon” is critical for the aggregate order variability at the upper echelon. If the number of agents at the lower echelon is very high compared to the number of agents at the upper echelon; aggregate order variability is low. As this ratio is decreased; individual demand forecasts turn out to have high variability and therefore the aggregate order variability is high in this regime, until a critical ratio is reached. If the ratio is further decreased, individual agent orders turn out to exhibit intermittent patterns; which reduces the aggregate order variability. The transition from continuous and smooth demand regime to intermittent demand regime can be best detected by focusing on individual dynamics and the active aggregate Order-Up-To-Level dynamics. If the number of agents is taken the same at all echelons, aggregate order statistics follow well-defined patterns as a function of the number of agents. Aggregate order means are not affected at all by the number of agents. Aggregate order variance and min-max range first increase and then decrease as the number of agents increases.

As far as the relationship between the supply chain topology and the aggregate inventory dynamics is concerned, aggregate inventory variability increases as the number of agents in that echelon increases. If the number of agents is taken to be the same at all echelons, aggregate inventory statistics follow well-defined patterns as a function of the number of agents. Aggregate inventory average increases as the number of agents increases. Therefore, the system tends to hold excess -surplus- inventory for a supply chain with higher number of firms. Aggregate inventory variance and range first increase and then decrease as the number of agents increases –an attitude similar to the aggregate order variability.

As far as the comparison between individual and aggregate order/inventory dynamics are concerned, aggregation -of the individual outputs- removes a percentage of the variability existing in individual order dynamics, which is very small when compared to the reduction in the Fixed S policy at high supply chain echelons.

### 5.2.3. Inventory and Order Dynamics under Reorder Point – Order-Up-To-Level Policy

In Reorder Point - Order-Up-To-Level (s,S) policy, each firm determines its reorder point (s) and orders an amount equal to the discrepancy between the current inventory position and the Order-Up-To-Level (S), whenever inventory drops below this ordering point. This ordering behavior is a form of order batching which is shown to be among the important sources of the bullwhip (Lee *et. al.*, 1997). Although it was not possible to utilize the analytical tools in the current framework; simulation experiments have been used to analyze the effects of order batching on supply chain dynamics. Experiments of Section 5.2.1 and Section 5.2.2 are repeated below with (s,S) policy. Figure 5.23 and Figure 5.32 illustrate aggregate order and inventory dynamics at each echelon for the case of 20 retailers, 10 wholesalers and two manufacturers. Figures for the remaining experiments with (s, S) policy can be found in Appendix D.

Figure 5.23 shows that aggregate order variability increases significantly as one moves up in the supply chain. When Figure 5.23 is compared with Figure 5.1 and Figure 5.9, it can be observed that the variability is amplified at all echelons as a consequence of order batching. Although individual retailer order pattern is intermittent due to order batching (see Figures 5.25-5.28), aggregate ordering attitude is continuous at the retailer echelon. It is the summation of individual intermittent orders –with phase lags among them- which provides continuous aggregate orders. On the other hand, aggregate manufacturer ordering attitude is intermittent due to the low number of manufacturers and the intermittent ordering pattern of individuals. Aggregate wholesaler order exhibits a pattern between continuity and intermittency as a result of being positioned between retailers and manufacturers.

Figure 5.24 shows the power spectral densities of aggregate orders for the case of 20 retailers, 10 wholesalers and two manufacturers. Figures for the remaining experiments with (s,S) policy can be found in Appendix D. By looking at the PSD, it can be observed that power is spread over a large medium-to-high frequency range. More condensation can be observed at the retailer echelon. See Table D.5 in Appendix D for spectral analysis results of several scenarios with different number of agents that provide similar observations.

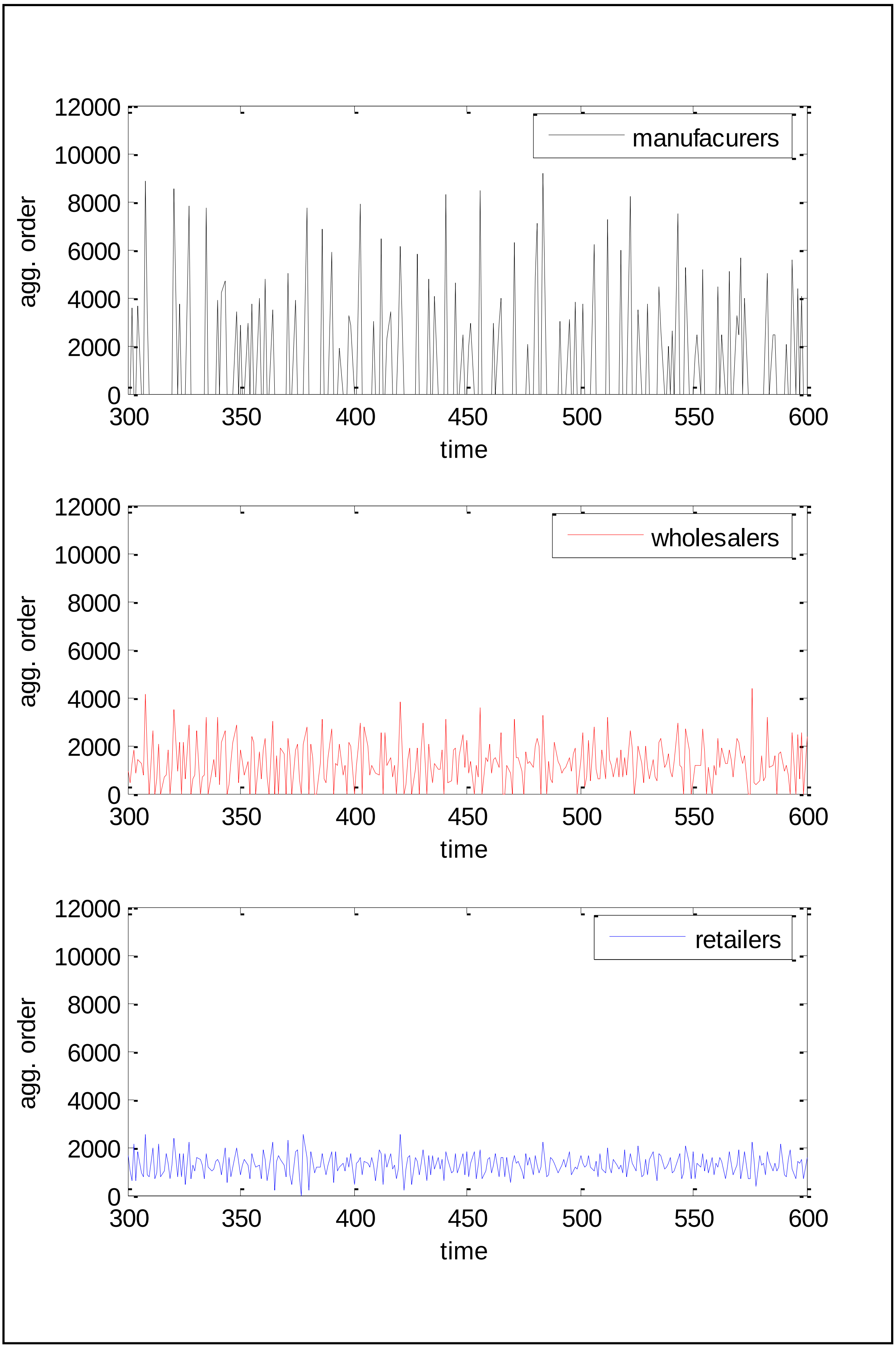


Figure 5.23. Aggregate Order Dynamics for (s,S) Policy

(20 Rtl., 10 Whl., 2 Mnf.)

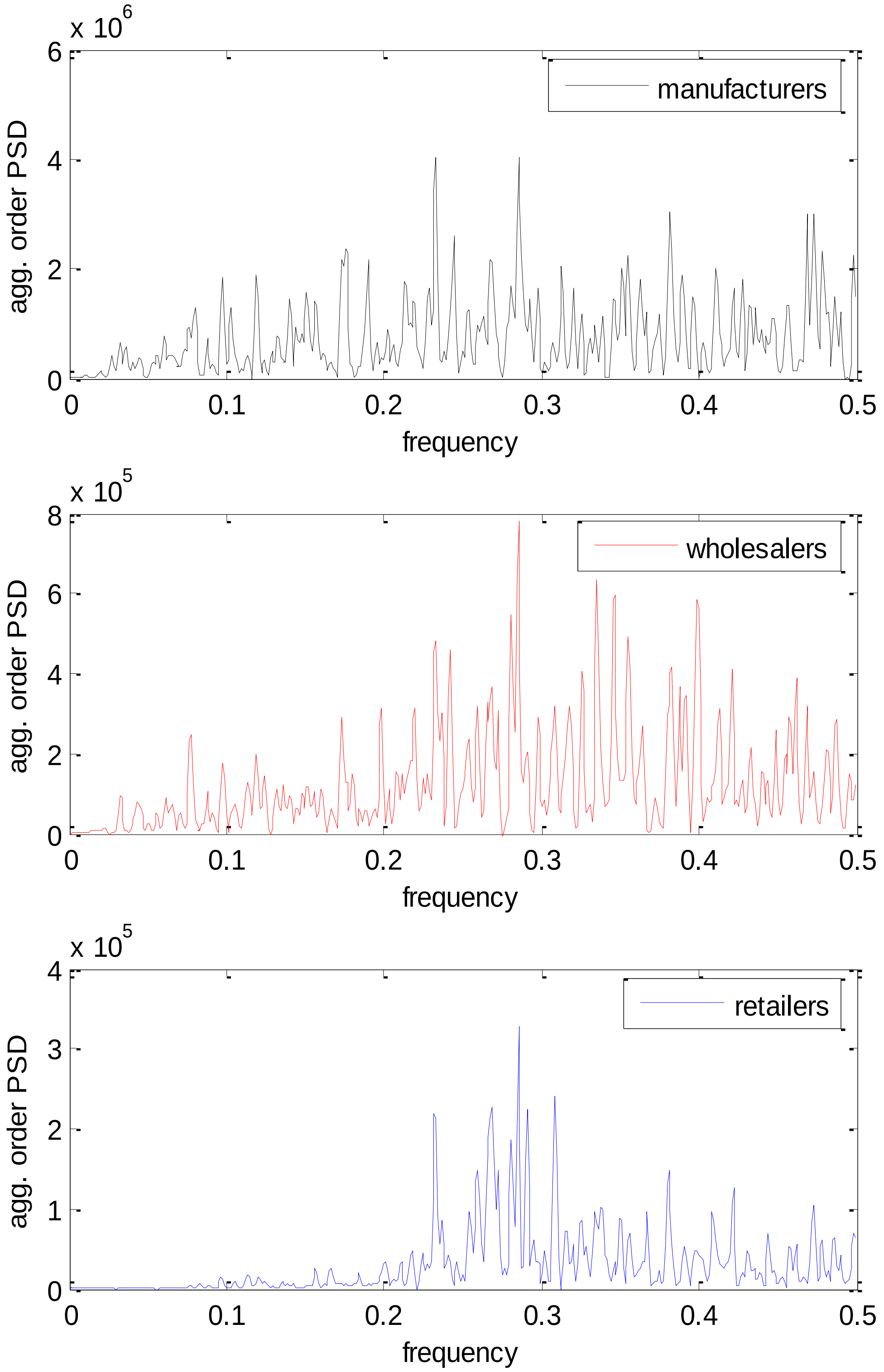


Figure 5.24. Spectral Power Density of Aggregate Orders for (s,S) Policy
(20 Rtl., 10 Whl., 2 Mnf.)

Table 5.15 gives the statistics of order data under the cases in Table 5.5, but with (s,S) policy. Variance and range of aggregate orders increase as one moves up in the supply chain, namely there is bullwhip effect at aggregate orders. The bullwhip effect and high variability at all echelons in (s,S) policy is a consequence of both demand forecasting and order batching. The volatility marginally introduced by order batching becomes apparent as one compares Table 5.15 with Table 5.5. Aggregate order average remains the same throughout the supply chain and is not influenced from demand forecasting, order batching, or the number of agents.

Table 5.15. Aggregate Order Statistics for (s,S) Policy

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|---|
| RTL. | Mean | 1250.017 | 1249.653 | 1250.101 | 1250.198 | 25000.35 |
| | std. dev. | 1395.586 | 982.066 | 438.7033 | 280.9266 | 2777.648 |
| | Range | 5187.238 | 4781.828 | 3061.632 | 2291.613 | 19871.7 |
| WHL. | Mean | 1250.941 | 1248.825 | 1249.608 | 1250.309 | 25005.04 |
| | std. dev. | 2274.518 | 1674.412 | 914.0791 | 624.7832 | 7110.894 |
| | Range | 10902.58 | 9552.575 | 5631.706 | 4354.762 | 55145.11 |
| MNF. | Mean | 1250.513 | 1248.501 | 1250.137 | 1250.785 | 25020.67 |
| | std. dev. | 3582.645 | 2877.139 | 2105.364 | 1507.783 | 17874.01 |
| | Range | 25919.61 | 22754.93 | 13623.41 | 9532.736 | 122980.7 |

Table 5.15 designates the dependency of aggregate order variability on the number of agents. Individual ordering pattern is always intermittent irrespective of the number of agents as a result of order batching. Therefore, a transition to intermittency regime as in Variable Order-Up-To-Level policy is not observed in (s,S) policy, namely the system is always in the individual intermittent ordering regime. Therefore, aggregate echelon order variability decreases steadily due to the summation of individual intermittent orders and the removal of intermittency by phase lags among individual orders as the number of agents in an echelon increases. The relationship between the number of retailers and aggregate retailer order dynamics is illustrated in Figures 5.25-5.28. Similar observations are available for aggregate wholesaler and manufacturer dynamics.

Figures 5.25-5.28 show the order, inventory, reorder point(s), and Order-Up-To-Level (S) dynamics of a randomly selected retailer and the aggregate retailer echelon as the number of retailers increases. The number of final customers, wholesalers, and manufacturers are 500, one, and one for all these cases respectively. Note that the y-axis ranges are tuned to make it simpler to compare the four cases in Figures 5.25-5.28, as before.

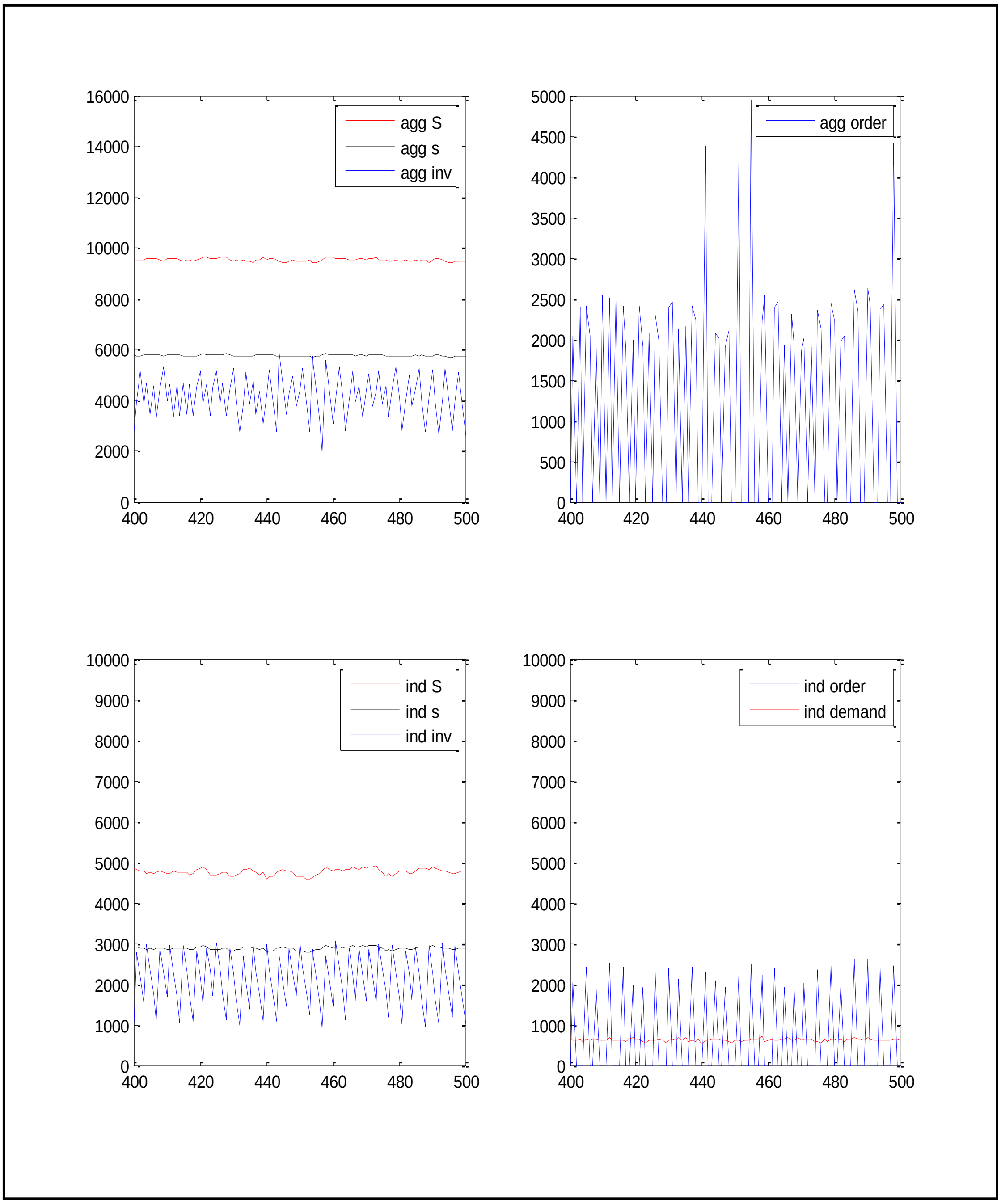


Figure 5.25. Individual and Aggregate Retailer Dynamics for (s,S) Policy

(2 Rtl., 1 Whl., 1 Mnf.)

Figure 5.25 shows individual and aggregate retailer dynamics for the case of two retailers. Since the number of retailers is very small compared to the number of final customers; individual retailers get smooth demand. Consequently, individual s and S levels are stable. Individuals do order batching and give intermittent orders with stable (time-invariant) ordering periods and magnitudes. Following that, individual inventory pattern is also regular and periodic. Since the number of retailers is low, aggregate retailer order and inventory exhibit patterns similar to individual dynamics. Notice that the periodicity and regularity in individual dynamics is removed to a degree at the aggregate scale by the interaction and phase lags between the two retailers. The regularity -at both individual and aggregate scales- pays off as high variability at the aggregate scale.

Figure 5.26 illustrates individual and aggregate retailer dynamics for the case of 20 retailers. Individual ordering intermittency is preserved. However, the deviation in the number of final customers that give order to a randomly selected retailer is higher proportionally than the case in two retailers' scenario. Thus, the variability in individual demand is higher. Consequently, individual s and S levels have more variability compared to the two retailers' scenario. Therefore, individual orders exhibit intermittent dynamics with less stable ordering periods and magnitudes. Following that, individual inventories exhibit less regular oscillatory dynamics. In other words, the regularity and periodicity in individual order and inventory dynamics gets blurred as the number of retailers is increased. Aggregate retailer order and inventory patterns are significantly different than the patterns at the individual scale. The intermittency in individual orders is removed at the aggregate scale by the phase lags among individual retailer orders. The oscillatory component in individual inventories is significantly blurred at the aggregate scale. To summarize, increasing number of retailers removes the regularity at the individual scale; but reduces the variability at the aggregate scale.

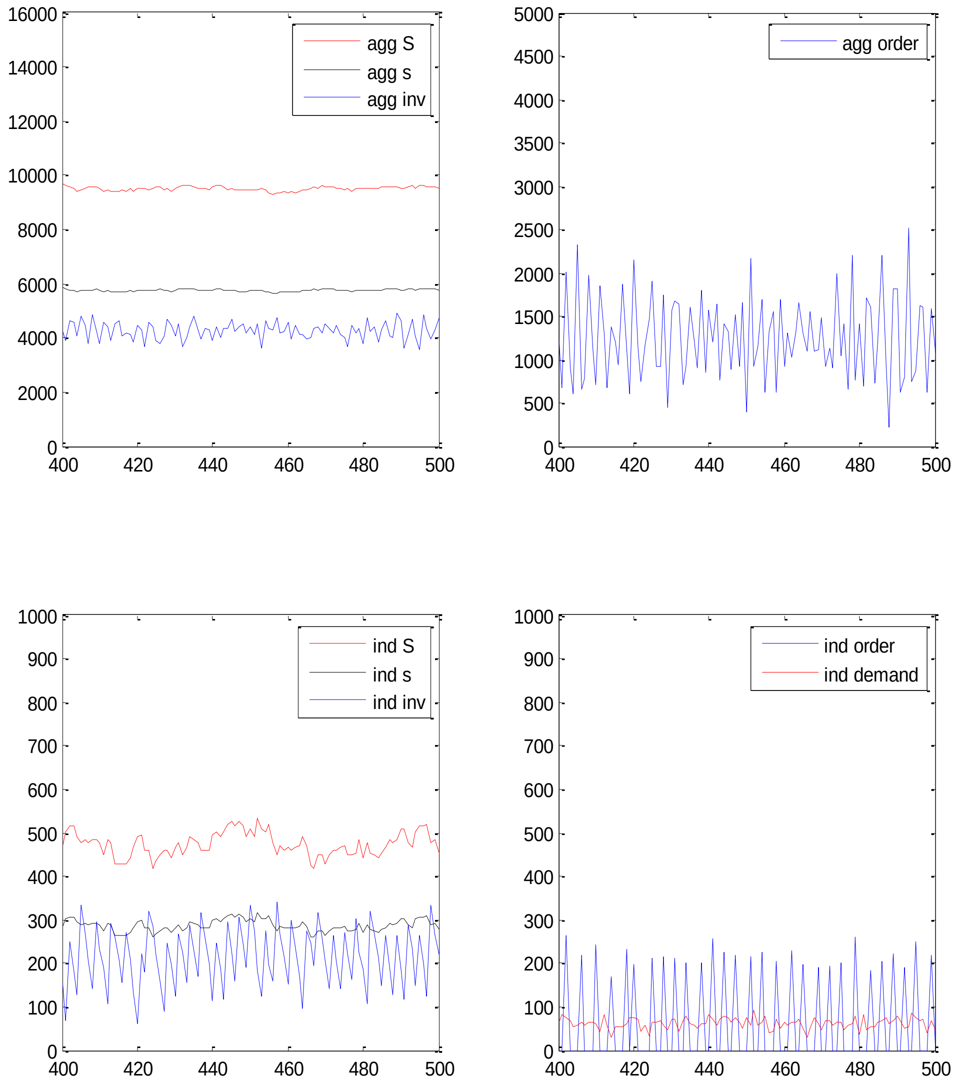


Figure 5.26. Individual and Aggregate Retailer Dynamics for (s,S) Policy (20 Rtl., 1 Whl., 1 Mnf.)

Figure 5.27 illustrates individual and aggregate retailer dynamics for the case of 200 retailers. The variability in individual demand is high. Following that, individual s and S levels fluctuate with high amplitudes. Therefore, individual orders exhibit intermittent dynamics with unstable periods and amplitudes. Consequently, individual inventories exhibit irregular oscillatory dynamics with time-varying periods and amplitudes. The regularity and periodicity in individual order and inventory dynamics are strongly attenuated as the number of retailers is increased to 200. However, aggregate retailer order and inventory patterns have less variability compared to the previous cases at the expense of instability at the individual scale. Similar to the previous case, increasing number of retailers removes the regularity at the individual scale; but reduces the variability at the aggregate scale.

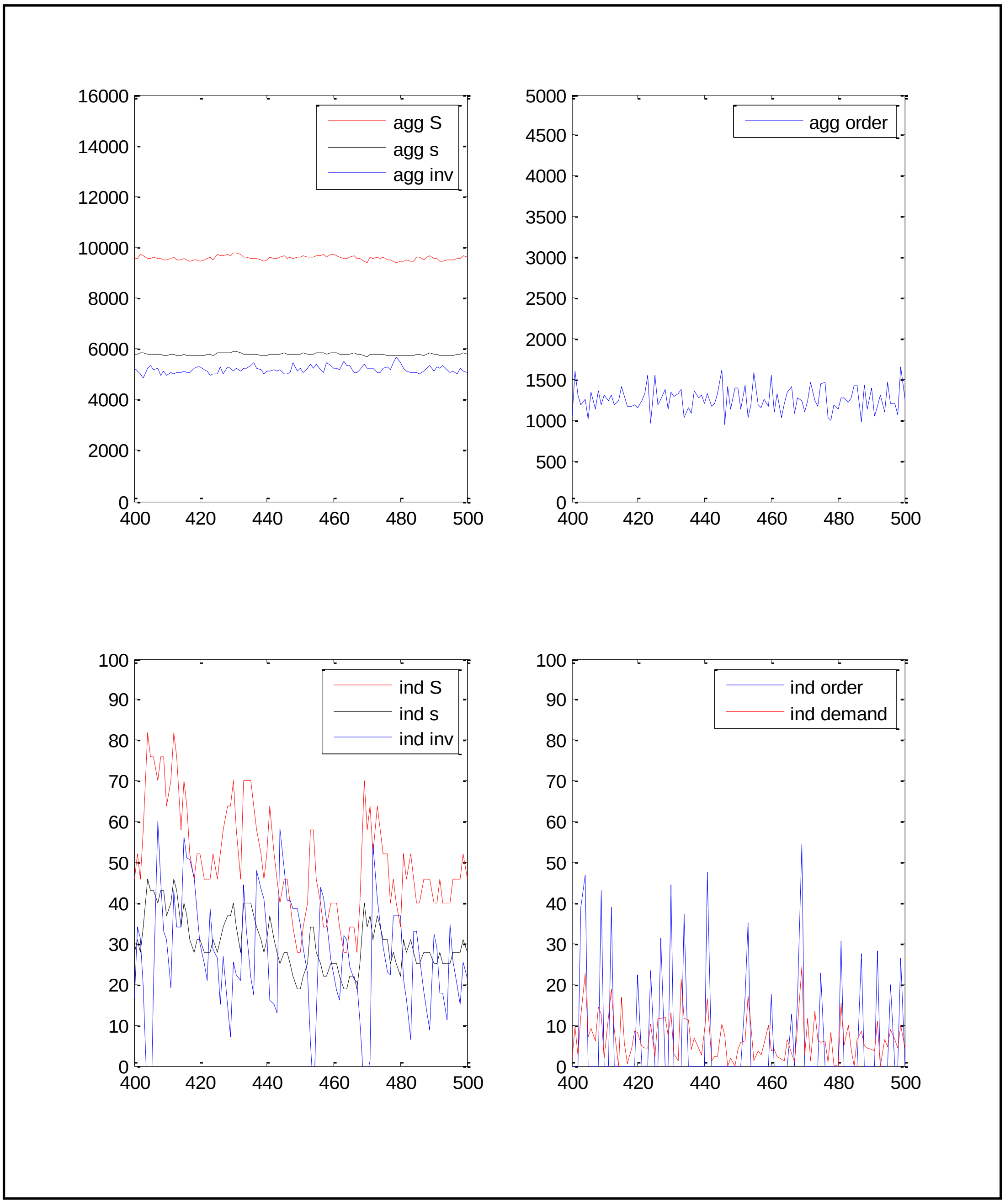


Figure 5.27. Individual and Aggregate Retailer Dynamics for (s,S) Policy (200 Rtl., 1 Whl., 1 Mnf.)

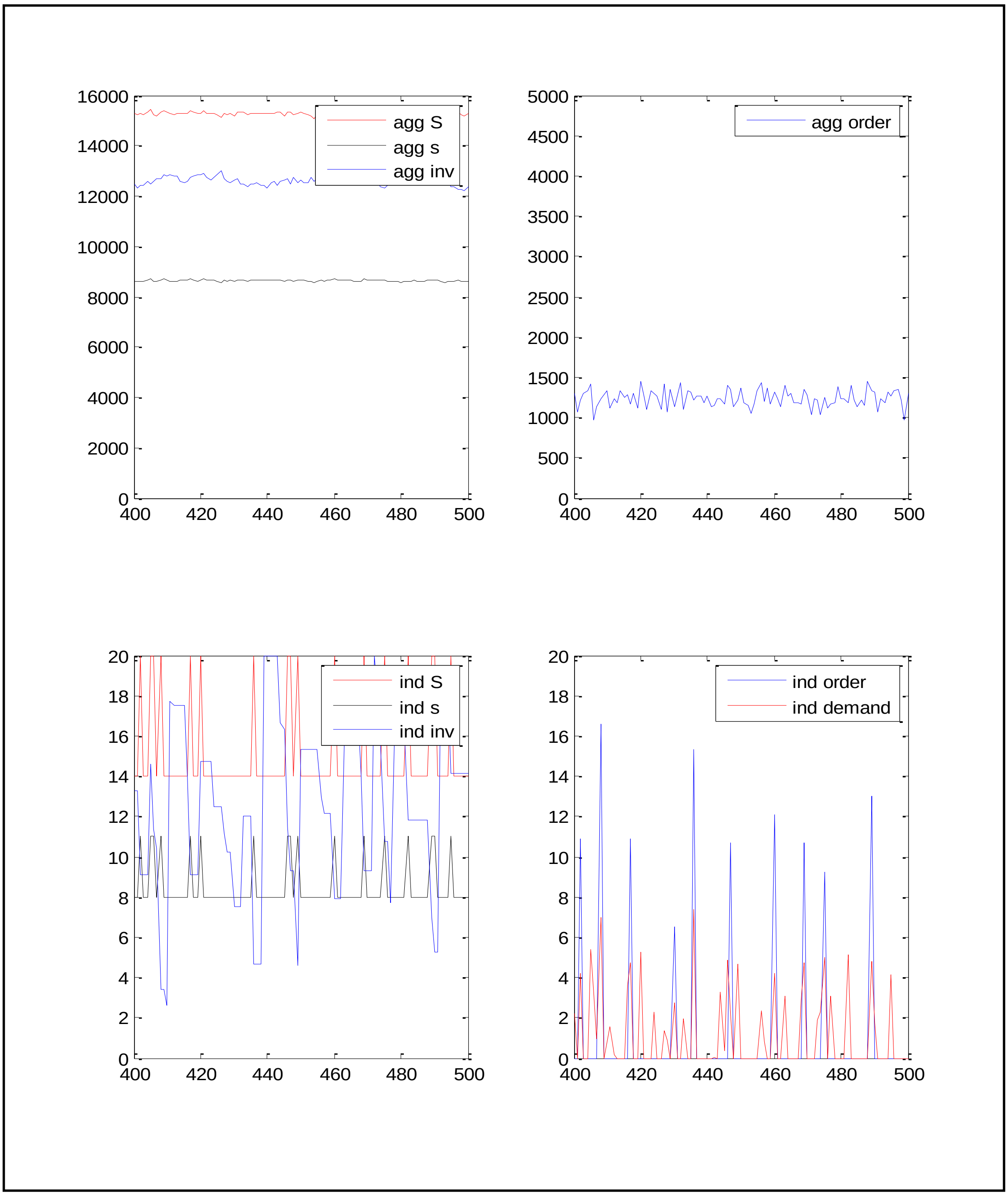


Figure 5.28. Individual and Aggregate Retailer Dynamics for (s,S) Policy
(1000 Rtl., 1 Whl., 1 Mnf.)

Figure 5.28 illustrates individual and aggregate retailer dynamics for the case of 1000 retailers. Individual retailers receive intermittent demand with time-varying period and amplitude due to high number of retailers. Individual s and S levels jump at the points final customer orders are received. Therefore, individual orders exhibit intermittent dynamics with unstable periods and amplitudes. Consequently, individual inventories stay at high levels for prolonged times, decrease sharply when goods are demanded, and jumps back when ordered goods are received, and so on. Individual inventories jump discontinuously between low and high levels with relatively longer periods due to intermittency in individual demands and orders. Aggregation of individual orders removes the intermittency and variability existing at the individual scale. Aggregate order variability is consequently decreased when the number of retailers is increased from 200 to 1000. Individual inventories exhibit smoother dynamics despite the discontinuities at the individual scale. However, individual oscillatory inventory patterns with relatively long periods and high amplitudes cause a slight increase in the aggregate retailer inventory variability. Figure 5.29 illustrates the increase in aggregate inventory variability -when the number of retailers is increased from 200 to 1000- by using time series graphs at a longer time scale. The aggregate echelon inventory spends more time closer to the mean in the case of 200 retailers; therefore the variance is higher in the case of 1000 retailers.

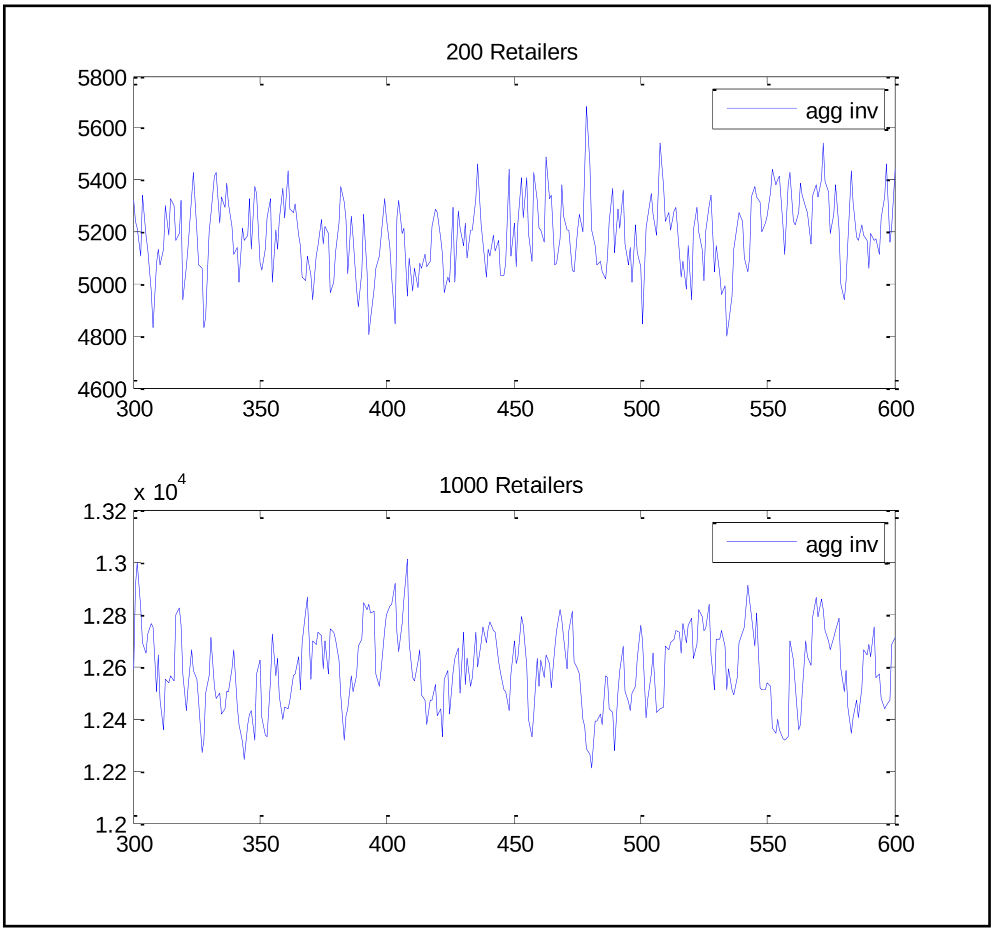


Figure 5.29. Comparison of Aggregate Inventory Dynamics for (s,S) Policy ($N_R$ Rtl., 1 Whl., 1 Mnf.)

To summarize the discussion about the relationship between the number of agents and aggregate order dynamics, the aggregate order variability decreases as the number of agents increases, where aggregate order mean remains the same. If the ratio of the number of agents at the upper echelon to the number of agents at the lower echelon is low, the demand is dispatched almost uniformly to individual agents at the upper echelon. The smoothness of individual demands provides regular periodic individual order and inventory dynamics. Due to low number of agents, aggregate orders exhibit characteristics similar to individual orders. Consequently aggregate order variability is high for low number of agents. Increasing number of agents removes the regularity at the individual scale, but reduces the variability at the aggregate scale as a result of phase lags among individual order time series in summation. Aggregate inventory variability decreases -parallel to the decrease in aggregate order variability- up to relatively high number of agents. If the number of agents is further increased, aggregate inventory variability is raised slightly due to the emergence of oscillations at longer time scales.

Tables 5.16-5.18 report the changes in aggregate order variance, range, and average when number of agents at only one echelon is varied -keeping other number of agents constant. The statistics in Tables 5.16-5.18 are consistent with the discussion above about the relationship between the number of agents and the aggregate order dynamics.

Table 5.16 shows the effect of the number of retailers, when there is one wholesaler and one manufacturer. Aggregate retailer order variance declines with the increasing number of retailers. The decreasing trend in aggregate order variability is preserved at higher echelons.

Table 5.16. Aggregate Order Statistics for the Effect of Number of Retailers under (s,S) Policy ($N_R$ Retailers, 1 Wholesalers, 1 Manufacturer)

| | | 2 Rtl. | 20 Rtl. | 100 Rtl. | 150 Rtl. | 200 Rtl. | 500 Rtl. | 1000 Rtl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1249.637 | 1249.94 | 1249.75 | 1249.738 | 1249.49 | 1245.81 | 1244.281 |
| | std. dev. | 1393.212 | 437.989 | 198.754 | 169.1338 | 152.328 | 112.989 | 105.4345 |
| | Range | 5146.493 | 3069.16 | 1530.09 | 1298.531 | 1266.88 | 888.598 | 819.895 |
| WHL. | Mean | 1249.411 | 1250.50 | 1249.36 | 1250.279 | 1249.57 | 1245.54 | 1244.604 |
| | std. dev. | 2588.978 | 2108.37 | 2023.97 | 2013.979 | 2007.54 | 1998.47 | 1988.941 |
| | Range | 10661.55 | 7782.13 | 6867.26 | 6389.519 | 6073.4 | 5868.70 | 5912.216 |
| MNF. | Mean | 1250.41 | 1250.67 | 1248.91 | 1251.276 | 1250.36 | 1244.72 | 1244.092 |
| | std. dev. | 3759.676 | 3083.77 | 2966.73 | 2969.503 | 2969.80 | 2969.23 | 2962.143 |
| | Range | 26204.21 | 16171.6 | 14298.7 | 14617.87 | 13165.0 | 12376.8 | 12065.98 |

Table 5.17 shows the effect of the number of wholesalers, when there are 50 retailers and one manufacturer. Aggregate wholesaler order variance declines with the increasing number of wholesalers. The decreasing trend in aggregate order variability is preserved at the manufacturer echelon.

Table 5.17. Aggregate Order Statistics for the Effect of Number of Wholesalers under (s,S) Policy (50 Retailers, $N_W$ Wholesalers, 1 Manufacturer)

| | | 1 Whl. | 2 Whl. | 4 Whl. | 10 Whl. | 20 Whl. | 50 Whl. | 100 Whl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1249.57 | 1250.24 | 1249.88 | 1250.09 | 1249.99 | 1249.68 | 1249.754 |
| | std. dev. | 282.151 | 281.697 | 282.123 | 282.426 | 280.132 | 277.112 | 282.2744 |
| | Range | 2084.64 | 2223.63 | 2180.62 | 2290.57 | 2024.41 | 2221.92 | 2147.719 |
| WHL. | Mean | 1248.58 | 1250.35 | 1250.00 | 1250.07 | 1249.87 | 1249.74 | 1249.532 |
| | std. dev. | 2052.09 | 1469.01 | 1092.45 | 772.444 | 619.066 | 499.638 | 449.3104 |
| | Range | 7365.00 | 6524.09 | 6628.62 | 4817.47 | 4327.03 | 3843.29 | 3610.903 |
| MNF. | Mean | 1249.17 | 1250.18 | 1249.63 | 1249.49 | 1249.79 | 1249.73 | 1249.242 |
| | std. dev. | 2984.42 | 2779.44 | 2620.65 | 2426.09 | 2322.97 | 2231.44 | 2199.396 |
| | Range | 15327.2 | 16062.7 | 14584.3 | 12914.3 | 13850.0 | 11003.1 | 11089.33 |

Table 5.18 shows the effect of the number of manufacturers, when there are 20 retailers and 10 wholesalers. Aggregate manufacturer order variance declines with the increasing number of manufacturers.

Table 5.18. Aggregate Order Statistics for the Effect of Number of Manufacturers under (s,S) Policy (20 Retailers, 10 Wholesalers, $N_M$ Manufacturer)

| | | 1 Mnf. | 2 Mnf. | 4 Mnf. | 8 Mnf. | 10 Mnf. | 20 Mnf. | 40 Mnf. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1249.8 | 1250.10 | 1250.29 | 1249.86 | 1249.54 | 1250.16 | 1250.39 |
| | std. dev. | 435.97 | 438.703 | 437.116 | 442.604 | 440.178 | 442.151 | 438.866 |
| | Range | 3451.3 | 3061.63 | 2971.92 | 3127.53 | 3129.09 | 3412.49 | 3298.49 |
| WHL. | Mean | 1249.9 | 1249.61 | 1249.82 | 1250.12 | 1249.92 | 1250.07 | 1250.88 |
| | std. dev. | 910.22 | 914.079 | 911.420 | 915.424 | 907.297 | 909.555 | 913.211 |
| | Range | 5898.1 | 5631.71 | 5772.56 | 5466.91 | 5753.78 | 5226.34 | 5899.81 |
| MNF. | Mean | 1250.3 | 1250.14 | 1251.03 | 1250.37 | 1249.65 | 1247.26 | 1247.15 |
| | std. dev. | 2542.7 | 2105.36 | 1850.78 | 1697.41 | 1650.30 | 1571.63 | 1532.65 |
| | Range | 14161. | 13623.4 | 12363.6 | 11832.3 | 11573.6 | 10207.8 | 11115.7 |

Figure 5.30 shows how the mean, standard deviation and min-max range of the aggregate orders change as a function of the number of agents when the number of retailers, wholesalers, and manufacturers are the same. Aggregate order mean is not affected by the number of agents. Since individual ordering attitude is always intermittent, aggregate echelon standard deviations decrease monotonically with the increasing number of agents. Aggregate order ranges exhibit similar patterns but with some noise. The observations on the effect of the number of agents at all echelons are similar to the effect of the number of agents at just one echelon.

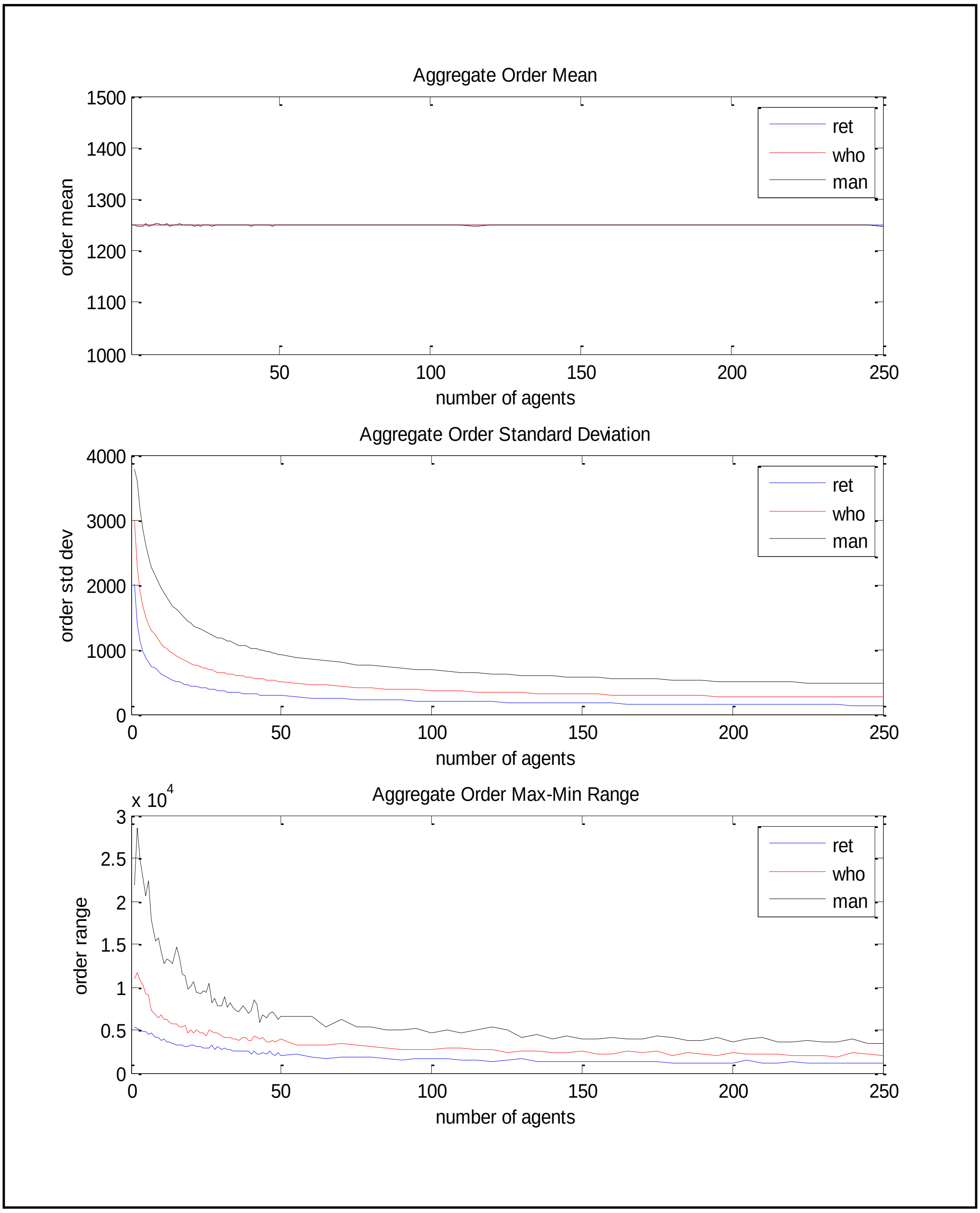


Figure 5.30. Aggregate Order Statistics as Functions of the Number of Agents (All Echelons Equal) for (s,S) Policy

Individual intermittent ordering is observed at all echelons irrespective of the number of agents. Figure 5.31 shows how the average ordering periods at different supply chain echelons change as a function of the number of agents. Average ordering period increases as the number of agents increases or as one moves up in the supply chain. The elongation of average ordering period at all echelons due to order batching becomes apparent as one compares Figure 5.31 with Figure 5.19.

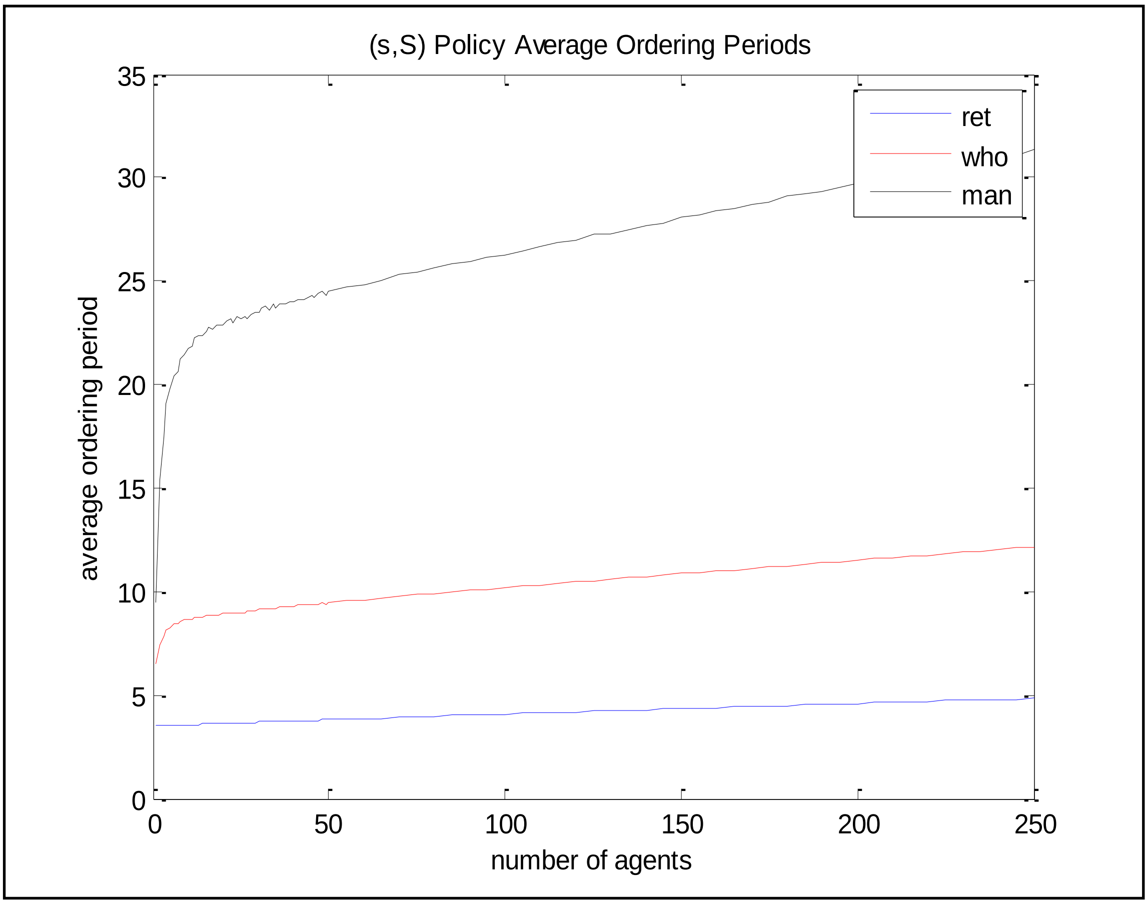


Figure 5.31. Average Echelon Ordering Periods as Functions of the Number of Agents (All Echelons Equal) for (s,S) Policy

Table 5.19 summarizes the comparison of the aggregate order dynamics with the individual firm order dynamics. It can be observed that aggregation -of the individual outputs- removes a percentage of the variability very close to the reference ratio at all echelons irrespective of the number of agents. This observation underlines the fact that individual order dynamics are almost totally independent of each other –due to separation of demand and order in the time scale-, and aggregation at the analysis phase eliminates the variability no more than the number of samples reduction.

Table 5.19. (s,S) Policy, Aggregate-Individual Order Variability Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.696866 | 0.490838 | 0.216811 | 0.129938 | 0.068846 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.670571 | 0.461229 | 0.288424 | 0.198083 | 0.12906 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.696705 | 0.464338 | 0.679253 | 0.469241 | 0.292127 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

Figure 5.32 shows the aggregate echelon inventory dynamics for the case of 20 retailers, 10 wholesalers, and two manufacturers. It can be observed that the bullwhip effect exists for inventory data and that inventory exhibits smoother dynamics compared to the rate variable order. This is most obvious from the manufacturers' plot. The variability introduced by order batching can be detected by comparing Figure 5.32 with Figure 5.20. Figures for other experiments with (s,S) policy can be found in Appendix D. Figure 5.33 shows the spectral power density of aggregate inventories for the case of 20 retailers, 10 wholesalers, and two manufacturers. The spectral power of aggregate retailer inventory is condensed in the high frequency range. The spectral powers of wholesaler and manufacturer inventories are condensed in the low frequency range. There are multiple dominant periodicities at all echelons representing that the oscillations are irregular. Table D.6 summarizes the spectral analysis results for aggregate inventory data under various settings of the number of agents. It infers that the total and dominant spectral powers increase on the average, in the upward direction. This is an indicator of the bullwhip effect in terms of oscillation amplitude amplification. Total and dominant spectral powers are higher for the (s,S) policy when compared to the Fixed S and Variable S policies. This increase in oscillatory dynamics is caused by the order batching.

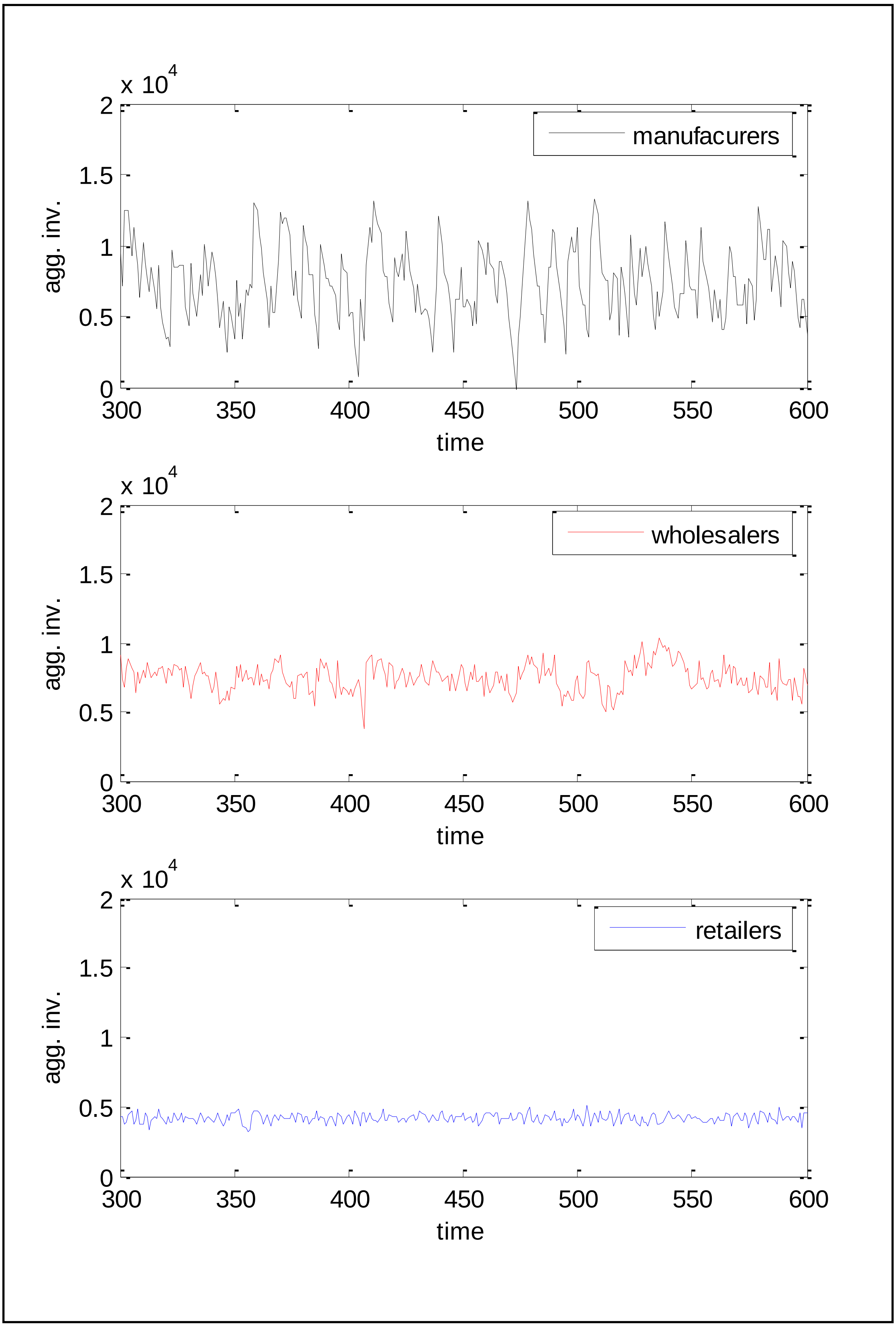


Figure 5.32. Aggregate Inventory Dynamics for (s,S) Policy

(20 Rtl., 10 Whl., 2 Mnf.)

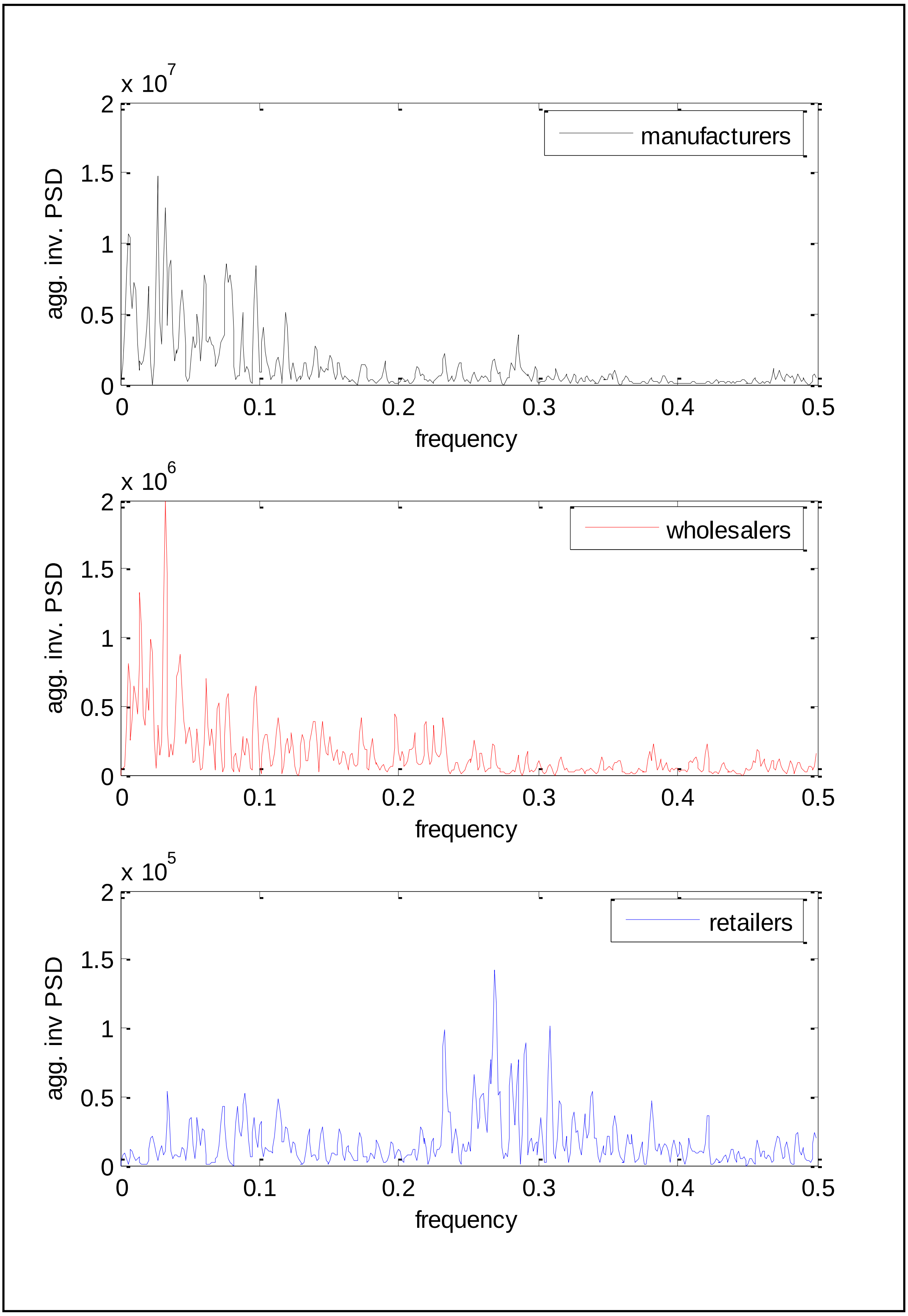


Figure 5.33. Spectral Power Density of Aggregate Inventories for (s,S) Policy

(20 Rtl., 10 Whl., 2 Mnf.)

Table 5.20 gives the statistics of inventory data for the cases in Table 5.15. Aggregate inventory variance and range increase as one moves up in the supply chain. The bullwhip effect in aggregate inventory is a consequence of inflated lead times, demand forecasting, and order batching. The marginal volatility introduced by order batching becomes apparent as one compares Table 5.20 and Table 5.10.

Table 5.20. Aggregate Inventory Statistics for (s,S) Policy

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|---|
| RTL. | Mean | 4136.871 | 4099.486 | 4171.54 | 4307.613 | 83282.15 |
| | std. dev. | 885.6472 | 653.7726 | 317.499 | 221.014 | 1990.184 |
| | Range | 5350.915 | 5149.83 | 2861.97 | 1915.147 | 15896.64 |
| WHL. | Mean | 9011.539 | 9619.029 | 7328.54 | 6920.733 | 113942.5 |
| | std. dev. | 2637.948 | 1972.964 | 1015.41 | 706.5588 | 6934.121 |
| | Range | 19334.01 | 15309.04 | 8665.35 | 5660.17 | 56120.54 |
| MNF. | Mean | 21521.14 | 25820.19 | 7461.00 | 7739.615 | 144250.6 |
| | std. dev. | 6910.15 | 6244.965 | 2704.77 | 1933.966 | 21581.21 |
| | Range | 46305.81 | 42895.31 | 20010.5 | 15670.02 | 179710 |

It is evident from Table 5.20 that the number of agents affects not only the variability but also the mean of aggregate inventory. Aggregate inventory mean at an echelon increases as the number of agents at that echelon increases, as in Variable S policy. A further observation in (s,S) policy is on the relationship between the number of agents at an echelon and aggregate inventory mean at other echelons. As the number of agents at an echelon increases, aggregate inventory means at the remaining echelons decrease. However, the underlying mechanism could not be discovered at the current state of the research.

Increasing number of agents removes the regularity and periodicity at the individual inventories, but reduces the variability at the aggregate scale as a result of phase lags among individual inventory time series until a critical value is reached (see Figures 5.25-5.28). Therefore, aggregate inventory variability decreases -parallel to the decrease in aggregate order variability- up to a critical value of the number of agents. If the number of agents is further increased, aggregate inventory variability is inflated due to the emergence of individual inventory oscillations with long periods and high amplitudes. Therefore, aggregate inventory variability decreases first until a critical value and then increases again. The critical value of the number of agents decreases as one moves up in the supply chain; because average ordering periods are augmented in the upward direction as in Figure 5.31. The effect of the number of agents on the aggregate inventory variability at other echelons is similar to the effect under Variable S policy. The interaction between the number of agents at the lower echelon and the aggregate inventory variability at the upper echelon is through the variability in aggregate orders of the lower echelon.

Tables 5.21-5.23 report the changes in aggregate inventory variance, range, and mean when number of agents at one echelon is varied -keeping number of agents at the remaining echelons constant.

Table 5.21. Aggregate Inventory Statistics for the Effect of Number of Retailers under (s,S) Policy ($N_R$ Retailers, 1 Wholesalers, 1 Manufacturer)

| | | 2 Rtl. | 20 Rtl. | 100 Rtl. | 150 Rtl. | 200 Rtl. | 500 Rtl. | 1000 Rtl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 4166.74 | 4260.09 | 4644.86 | 4896.67 | 5181.55 | 7189.41 | 12610.1 |
| | std. dev. | 882.274 | 292.142 | 164.479 | 153.079 | 146.871 | 141.198 | 158.4758 |
| | Range | 4132.13 | 2241.79 | 1255.15 | 1161.57 | 1127.33 | 1077.94 | 1291.521 |
| WHL. | Mean | 5739.45 | 4463.64 | 4222.52 | 4210.85 | 4201.26 | 4171.65 | 4168.065 |
| | std. dev. | 2296.38 | 1467.63 | 1323.80 | 1305.80 | 1297.53 | 1283.94 | 1276.242 |
| | Range | 12003.5 | 8435.46 | 6147.91 | 6141.58 | 5683.22 | 5759.26 | 5221.55 |
| MNF. | Mean | 12023.1 | 8896.04 | 8173.43 | 8060.04 | 7990.74 | 7853.74 | 7807.764 |
| | std. dev. | 5558.04 | 3419.89 | 3150.62 | 3117.39 | 3104.58 | 3080.92 | 3067.243 |
| | Range | 31763.7 | 22364.9 | 15325.6 | 14820.6 | 14316.2 | 14220.6 | 13718.37 |

Table 5.21 shows the effect of the number of retailers, when there is one wholesaler and one manufacturer. Aggregate retailer inventory variance decreases until the number of retailers reaches 500. Then, aggregate retailer inventory variability increases due to the emergence of oscillations with long periods as in Figure 5.29. Aggregate wholesaler and manufacturer inventory variances decrease as the number of retailers increases due to the decreasing order variability. Aggregate retailer inventory mean increases, where aggregate wholesaler and manufacturer means decrease, as the number of retailers increases.

Table 5.22. Aggregate Inventory Statistics for the Effect of Number of Wholesalers under (s,S) Policy (50 Retailers, $N_W$ Wholesalers, 1 Manufacturer)

| | | 1 Whl. | 2 Whl. | 4 Whl. | 10 Whl. | 20 Whl. | 50 Whl. | 100 Whl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 4398.02 | 4396.715 | 4392.57 | 4364.17 | 4318.15 | 4254.23 | 4230.134 |
| | std. dev. | 205.9252 | 205.8376 | 204.882 | 211.749 | 216.909 | 216.790 | 216.5887 |
| | Range | 1594.18 | 1646.245 | 1594.17 | 1722.66 | 1922.15 | 1728.84 | 1652.705 |
| WHL. | Mean | 4274.315 | 4419.137 | 4693.54 | 5557.07 | 7009.72 | 11274.5 | 18398.87 |
| | std. dev. | 1374.878 | 1044.334 | 851.997 | 721.566 | 686.575 | 707.262 | 785.1978 |
| | Range | 7140.249 | 6923.426 | 6174.75 | 5612.26 | 5291.31 | 6122.49 | 6702.339 |
| MNF. | Mean | 8462.389 | 6973.221 | 5952.40 | 5188.64 | 4860.84 | 4625.83 | 4544.825 |
| | std. dev. | 3220.863 | 2937.126 | 2641.99 | 2335.67 | 2136.84 | 1929.01 | 1864.451 |
| | Range | 20797.59 | 18353.73 | 19159.0 | 17272.4 | 15055.1 | 13754.4 | 12415.96 |

Table 5.22 shows the effect of the number of wholesalers, when there are 50 retailers and one manufacturer. Aggregate wholesaler inventory variance decreases until the number of wholesalers reaches 20. Then, aggregate wholesaler inventory variability increases due to the emergence of oscillations with long periods and high amplitudes at the individual scale. The change in aggregate retailer inventory variance is not intense as a function of the number of wholesalers increases. However, manufacturer inventory variance decreases as the number of wholesalers increases due to the decreasing order variability. Aggregate wholesaler mean increases, where aggregate retailer and manufacturer means decrease, as the number of wholesalers increases.

Table 5.23. Aggregate Inventory Statistics for the Effect of Number of Manufacturers under (s,S) Policy (20 Retailers, 10 Wholesalers, $N_M$ Manufacturer)

| | | 1 Mnf. | 2 Mnf. | 4 Mnf. | 8 Mnf. | 10 Mnf. | 20 Mnf. | 40 Mnf. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 4176.149 | 4171.54 | 4165.31 | 4160.83 | 4162.87 | 4159.49 | 4156.40 |
| | std. dev. | 316.2949 | 317.499 | 325.133 | 322.668 | 324.482 | 327.284 | 323.136 |
| | Range | 2581.992 | 2861.97 | 2799.84 | 2760.37 | 3122.82 | 3165.68 | 3015.48 |
| WHL. | Mean | 7393.668 | 7328.54 | 7249.37 | 7198.78 | 7170.30 | 7158.86 | 7147.45 |
| | std. dev. | 994.4962 | 1015.41 | 1005.95 | 1003.55 | 1011.52 | 1012.45 | 1004.58 |
| | Range | 7486.389 | 8665.35 | 8309.39 | 7677.32 | 8079.01 | 8483.51 | 8853.36 |
| MNF. | Mean | 5554.979 | 7461.00 | 11328.7 | 18887.7 | 22462.6 | 40341.4 | 76339.6 |
| | std. dev. | 2718.732 | 2704.77 | 2877.57 | 3252.23 | 3470.38 | 4159.48 | 5283.92 |
| | Range | 18839.61 | 20010.5 | 22529.8 | 24669.9 | 28754.4 | 28811.7 | 39285.8 |

Table 5.23 shows the effect of the number of manufacturers, when there are 20 retailers and 10 wholesalers. Aggregate manufacturer inventory variance is low up to the case of 2 manufacturers. Then, aggregate manufacturer inventory variability increases due to the emergence of oscillations with long periods and high amplitudes. Aggregate retailer and wholesaler inventory variances are not influenced significantly. Aggregate manufacturer inventory mean increases, where aggregate retailer and wholesaler inventory means decrease slightly, as the number of manufacturers increases.

Figure 5.34 shows how aggregate inventory mean, variability and range change as a function of the number of agents if the number of retailers, wholesalers, and manufacturers are taken to be equal. Aggregate inventory averages increase as the number of agents increases. Therefore, the supply chain as a whole tends to hold more excess inventory as the number of agents increases. Aggregate manufacturer inventory standard deviation increases significantly when the number of agents is increased to two. Then, it decreases smoothly as the number of agents increases. Aggregate retailer and wholesaler inventory variability decreases steadily as the number of agents increases. Aggregate inventory ranges show characteristics similar to the standard deviation. However, it can be observed that an initial increase can be observed at all echelons for range statistics.

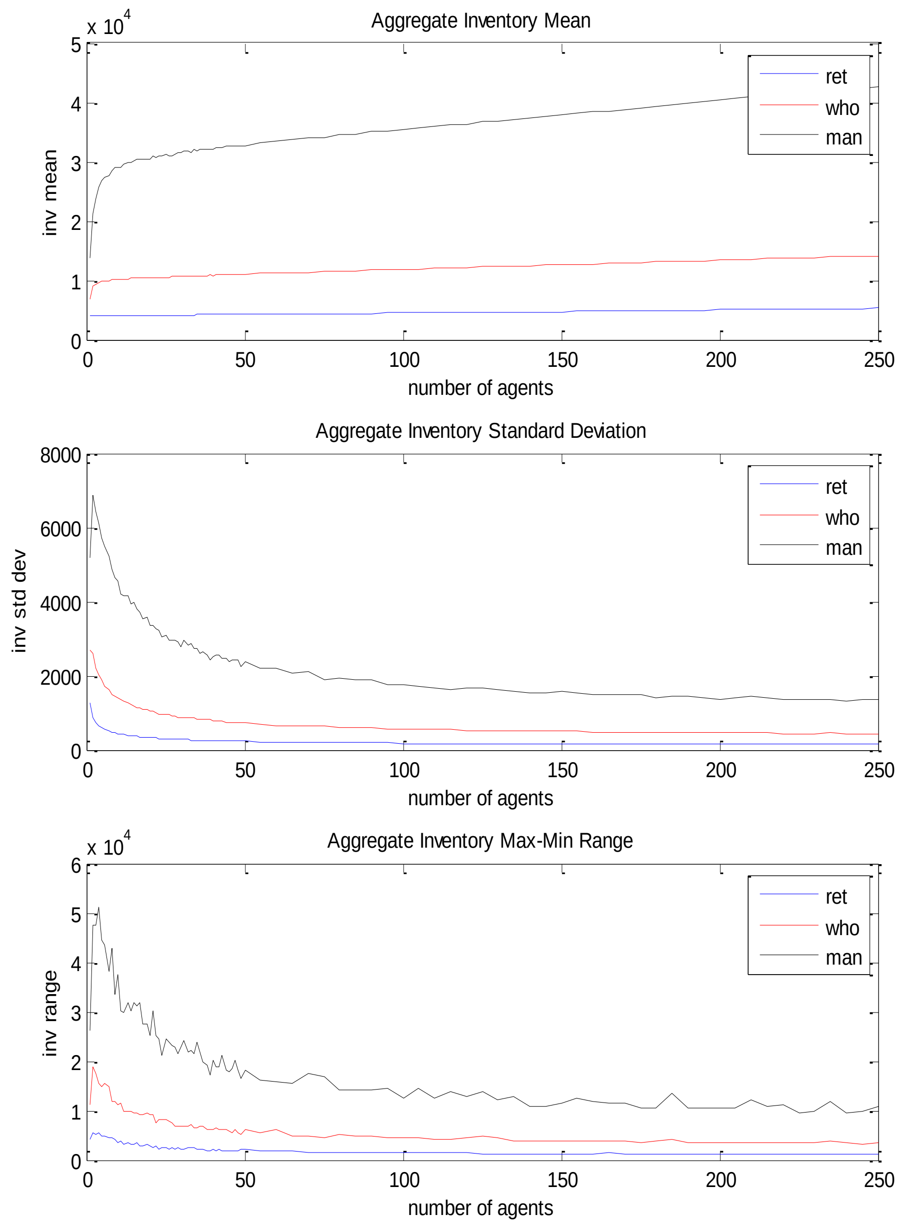


Figure 5.34. Aggregate Order Statistics as Functions of the Number of Agents (All Echelons Equal) for (s,S) Policy

Table 5.24 summarizes the comparison of aggregate inventory dynamics with individual firm inventory dynamics in terms of variability. It can be observed that aggregation removes a percentage of the variability close to the reference ratio – especially at the retailers' echelon (it can be observed that the variability reduction is higher than the reduction for orders). The variability reduction due to aggregation is significantly less than the case in Variable S policy.

Table 5.24. (s,S) Policy, Aggregate-Individual Inventory Variability Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.696368 | 0.502631 | 0.218028 | 0.126143 | 0.073062 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.557384 | 0.357066 | 0.231066 | 0.164122 | 0.103472 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.672678 | 0.44388 | 0.592227 | 0.386994 | 0.234488 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

To summarize the observations about the (s,S) policy, the final customer demand variability is amplified through the supply chain as a result of demand forecasting, and order batching. Bullwhip effect emerges as a result of demand forecasting and order batching both at aggregate orders and inventories. Average ordering period, aggregate order variability and spectral power, aggregate inventory variability and spectral power are amplified as one moves up in the supply chain. Aggregate inventory and order variability are significantly higher than the variability under Fixed S and Variable S policies. Aggregate order mean is not influenced from the number of agents or the level in the chain.

As far as the interaction between the supply chain topology and aggregate order dynamics is concerned, aggregate order variability decreases as the number of agents at that echelon increases. If the ratio of the number of agents at the upper echelon to the number of agents at the lower echelon is low, the demand is dispatched almost uniformly to individual agents at the upper echelon. The smoothness of individual demands provides regular periodic individual order and inventory dynamics. Consequently, aggregate order variability is high for low number of agents. Increasing number of agents removes the regularity at the individual scale, but reduces the variability at the aggregate scale as a result of phase lags among individual order time series and their summation. Aggregate order pattern deviates significantly from individual order pattern as the number of agents increases.

If the number of agents is taken the same at all echelons, aggregate order statistics follow well-defined patterns as a function of the number of agents. Aggregate order mean is not affected from the number of agents. Aggregate order variance and min-max range decrease as the number of agents increases.

As far as the relationship between the supply chain topology and aggregate inventory dynamics is concerned, the number of agents affects not only the variability but also the mean of aggregate inventory. Aggregate inventory mean increases with the increase in the number of agents in the same echelon or with the decrease in the number of agents in one of the remaining echelons. Aggregate inventory variability decreases until a critical value of the number of agents is reached. If the number of agents is further increased, aggregate inventory variability is inflated due to the emergence of individual inventory oscillations with long periods and high amplitudes. Therefore, aggregate inventory variability decreases first until a critical value and then increases. The critical value of the number of agents decreases as one moves up in the supply chain; because average ordering periods increase in this direction.

If the number of agents is taken the same at all echelons, aggregate inventory statistics follow well-defined patterns as a function of the number of agents. Aggregate inventory average increases as the number of agents increases. Therefore, the supply chain as a whole tends to hold more excess inventory as the number of agents increases. Aggregate manufacturer inventory variance increases significantly when the number of is increased to two. Then, it decreases smoothly as the number of agents increases. Aggregate retailer and wholesaler inventory variances decrease steadily as the number of agents increases. Aggregate inventory ranges show characteristics similar to the variance. However, an initial increase can be observed at all echelons for range statistics.

Regarding the comparison of the system level aggregate order/inventory dynamics with the individual firm order/inventory dynamics, aggregation -of the individual outputs- removes a percentage of the variability close to the reference ratio. This reveals the fact that individual order/inventory dynamics are almost totally independent of each other, and aggregation at the analysis phase reduces the variability no more than the extent due to the number of samples.

### 5.2.4. Inventory and Order Dynamics under Anchor-And-Adjust Policy

In Anchor-And-Adjust (A&A) policy, firms aim to keep inventory and supply line at their desired levels, which are functions of demand forecasts and lead times. A&A policy is similar to Variable S policy. The main differences between the two policies are that: A&A policy (i) handles inventory and supply lines separately, and (ii) seeks to eliminate the discrepancies between actual and desired levels smoothly within an interval of adjustment time. On the other hand, Variable S policy handles inventory and supply line together, and seeks to eliminate the discrepancy between actual inventory position and the Order-Up-To-Level instantaneously. Simulation experiments are used to test the effects of A&A policy on order and inventory dynamics. Experiments of Sections 5.2.1-5.2.3 are repeated in this section with A&A policy. Figure 5.35 and Figure 5.43 show aggregate order and inventory dynamics at each echelon for the case of 20 retailers, 10 wholesalers and two manufacturers. Figures for other experiments with Anchor-And-Adjust policy can be found in Appendix D.

Figure 5.35 reflects that aggregate order variability is amplified as one moves up in the supply chain. The behavioral cause of the bullwhip effect under Anchor-And-Adjust policy is the utilization of demand forecasts in determining desired inventory and supply line levels. When Figure 5.35 is compared with Figure 5.9, it can be observed that aggregate orders exhibit smoother patterns compared to the Variable S policy, as expected.

Figure 5.36 shows the power spectral densities of aggregate orders for the case of 20 retailers, 10 wholesalers, and two manufacturers. It can be observed that the spectral power is condensed in the low-to medium frequency range as a result of smooth order dynamics. Table D.7 in Appendix D shows the spectral analysis results for order data in several scenarios with different number of agents. There are multiple dominant periodicities in aggregate orders. Total and dominant spectral powers increase as one moves up in the supply chain. Therefore, there is bullwhip effect in terms of aggregate order oscillation amplitude amplification.

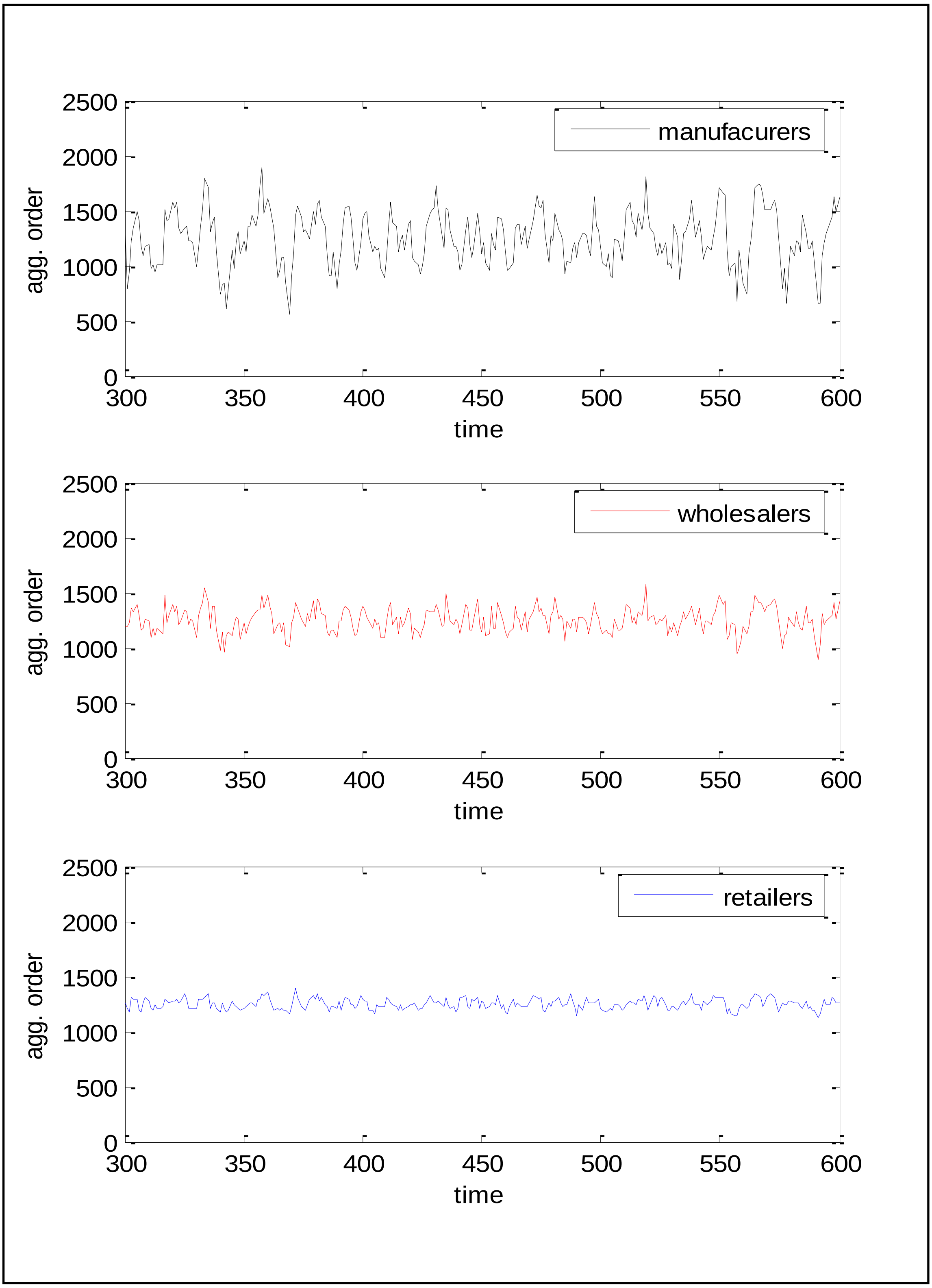


Figure 5.35. Aggregate Order Dynamics for A&A Policy

(20 Rtl., 10 Whl., 2 Mnf.)

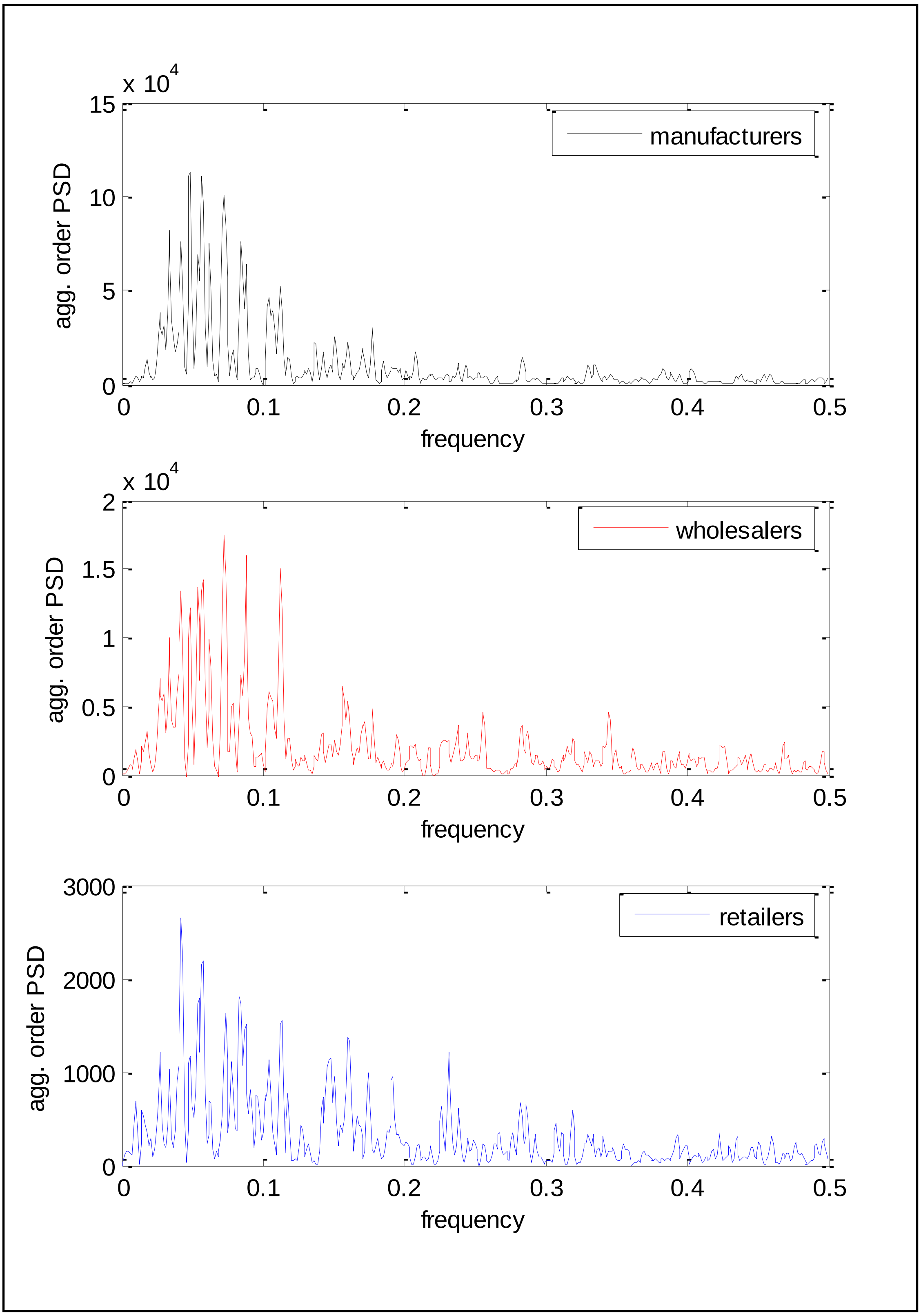


Figure 5.36. Spectral Power Density of Aggregate Orders for A&A Policy (20 Rtl., 10 Whl., 2 Mnf.)

Table 5.25 gives the statistics of order data for the cases in Table 5.15, but with A&A policy. Aggregate order variability and range increase as one moves up in the supply chain. The bullwhip effect is a consequence of using demand forecasts. Aggregate order variability under A&A policy is significantly lower than the variability under Variable S policy as a result of using adjustment times greater than one. Aggregate order average remains the same throughout the supply chain and is not influenced from the number of agents.

Table 5.25. Aggregate Order Statistics for A&A Policy

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|---|
| RTL. | Mean | 1250.073 | 1250.294 | 1249.94 | 1249.862 | 24997.52 |
| | std. dev. | 46.7321 | 46.4438 | 46.6036 | 46.8574 | 208.188 |
| | Range | 373.0897 | 363.5773 | 360.599 | 379.4233 | 1677.151 |
| WHL. | Mean | 1250.258 | 1250.349 | 1249.92 | 1249.831 | 24997.15 |
| | std. dev. | 280.347 | 227.2399 | 114.573 | 96.6947 | 504.6183 |
| | Range | 2404.862 | 1775.977 | 918.004 | 755.3084 | 3905.162 |
| MNF. | Mean | 1250.689 | 1250.741 | 1249.85 | 1249.773 | 24996.18 |
| | std. dev. | 682.8866 | 533.6464 | 256.336 | 224.7749 | 1546.83 |
| | Range | 4051.621 | 3404.175 | 2217.24 | 1915.308 | 12768.47 |

The relationship between the number of agents and aggregate order variability is similar to the one under Variable S policy. “The ratio of the number of agents at the lower echelon to the number of agents at the upper echelon” is critical for aggregate order variability at the upper echelon. If the number of agents at the lower echelon is very high compared to the number of agents at the upper echelon; the demand is distributed almost uniformly to individual agents. Therefore, aggregate order variability is low. As this ratio decreases; aggregate order variability increases first. Since the number of agents demanding from an agent at the upper echelon has high deviation, individual demand forecasts have high variability. If the ratio is further decreased, individual dynamics turn out to exhibit intermittent demand and ordering patterns; which reduces the aggregate order variability. It is important to note that the changes in aggregate order variability are less dramatic than the changes under Variable S policy due to smoothness of orders in any case.

These observations are illustrated by utilizing individual and aggregate dynamics information in Figures 5.37-5.40. Figures 5.37-5.40 illustrate the demand, order, inventory and desired inventory dynamics of a randomly selected manufacturer and the aggregate manufacturer echelon as the number of manufacturers is increased. The number of final customers, retailers, and wholesalers are 500, 20, and 10 respectively for all these cases. Note that the y-axis ranges are tuned to make it simpler to compare the four cases in Figures 5.37-5.40.

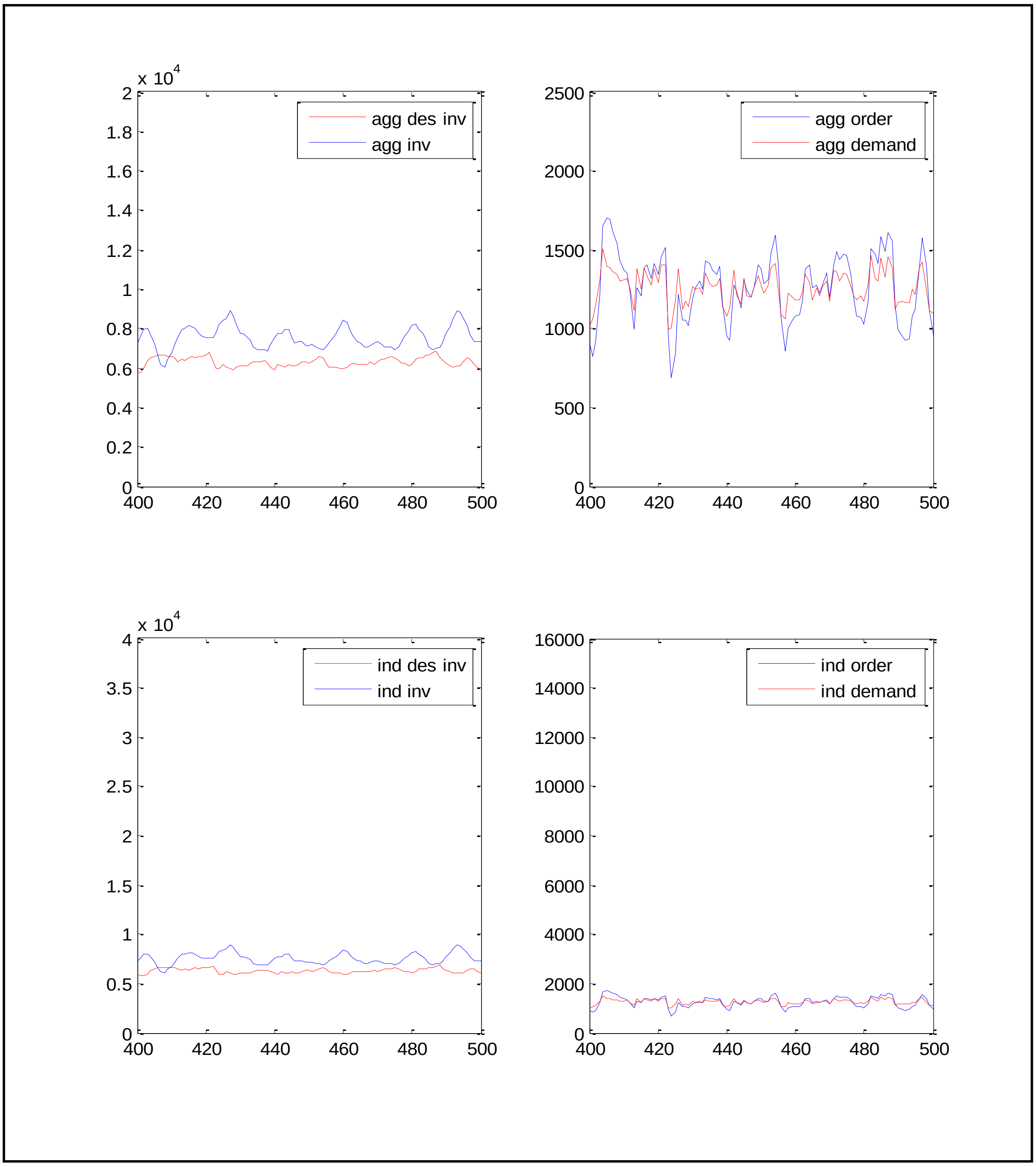


Figure 5.37. Individual and Aggregate Manufacturer Dynamics for A&A Policy (20 Rtl., 10 Whl., 1 Mnf.)

Figure 5.37 shows individual and aggregate manufacturer dynamics for single manufacturer case. Since there is only one manufacturer; aggregate and individual dynamics are the same. Since the manufacturer gets all of the aggregate demand; the individual demand is smooth compared to the cases in the transition regime in Figure 5.38 and Figure 5.39, and therefore is the S. The variability in manufacturer S, order and inventory is low consequently. It can be also observed that the differences between demand and order information are very small compared to the differences under Variable S policy in Figure 5.11.

Figure 5.38 shows individual and aggregate manufacturer dynamics when there are two manufacturers. The number of wholesalers an individual manufacturer gets order from deviates significantly in time. 10 wholesalers-two manufacturers interaction case falls into the transition regime. It can be observed that aggregate order smoothness is reduced to a degree by the increase in the variability of individual orders, when compared to the single manufacturer case. Aggregate inventory variance is also raised by the proportional augmentation of individual inventory oscillation amplitudes.

Figure 5.39 reports aggregate and individual manufacturer dynamics for the case of four manufacturers. This case also falls into the transition regime. The variability in individual demand and order is increased when the number of manufacturers is increased from two to four. Individual ordering behavior is closer to intermittency when there are four manufacturers. Aggregate inventory variance is incremented by the proportional augmentation of individual inventory oscillation amplitudes.

Figure 5.40 reports individual and aggregate manufacturer dynamics for the case of 20 manufacturers. This case falls into the intermittent ordering regime, which reduces the aggregate order variability. Since the number of manufacturers is high; individual manufacturers tend to get low or actually zero demand for most of the times. Comparably high orders are received at some infrequent points and the intermittent individual ordering attitude emerges –as a response to infrequent high demands. The intermittency can be detected at the individual demand and order dynamics. Although this attitude amplifies the individual order variability, it provides low order variability at the aggregate level. On the other hand, aggregate inventory variability is increased in the intermittency regime. Since individual inventories are stock variables, intermittent demands and orders have prolonged effects. Individual inventory oscillations with long periods and high amplitudes are reflected at the aggregate inventories.

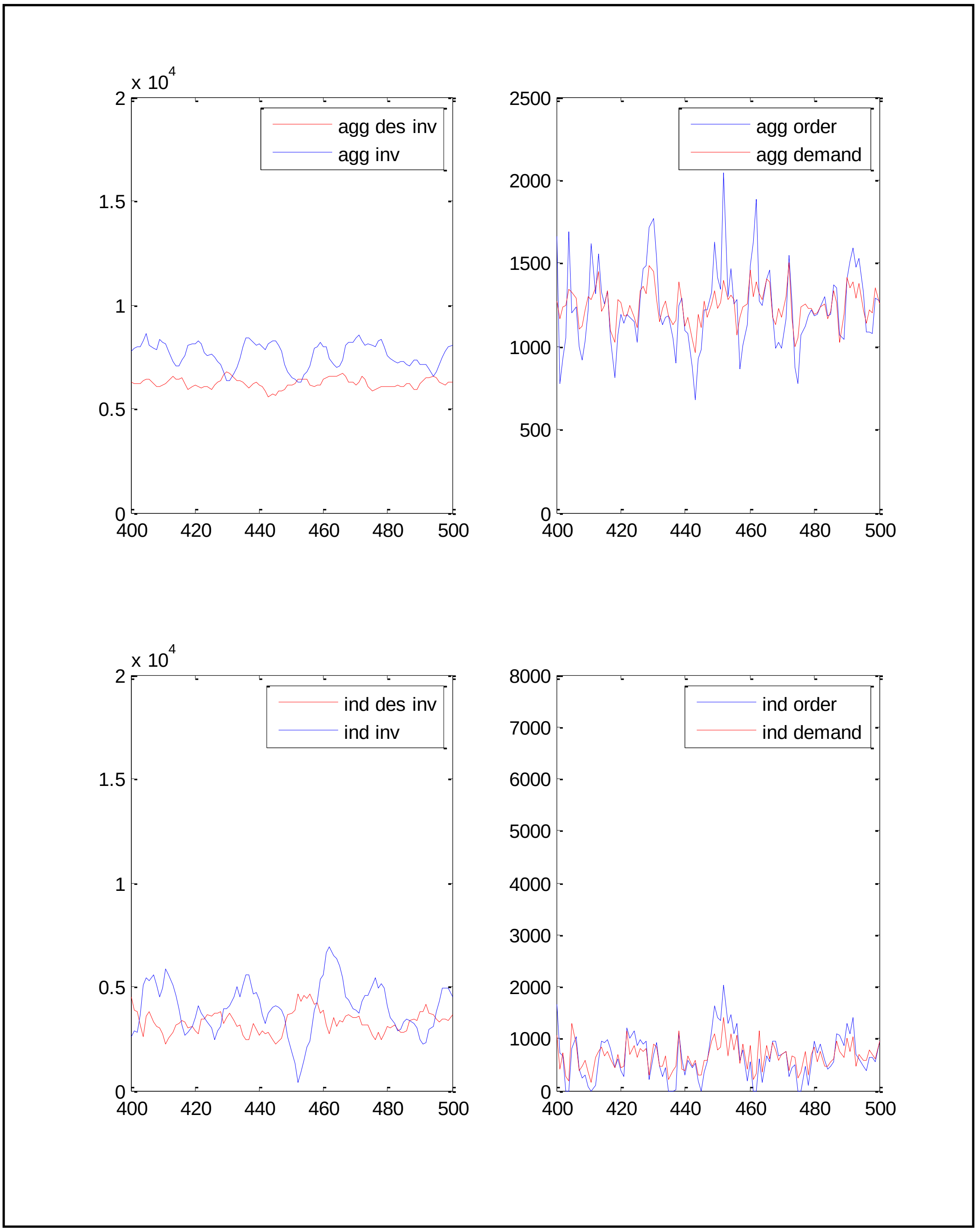


Figure 5.38. Individual and Aggregate Manufacturer Dynamics for A&A Policy (20 Rtl., 10 Whl., 2 Mnf.)

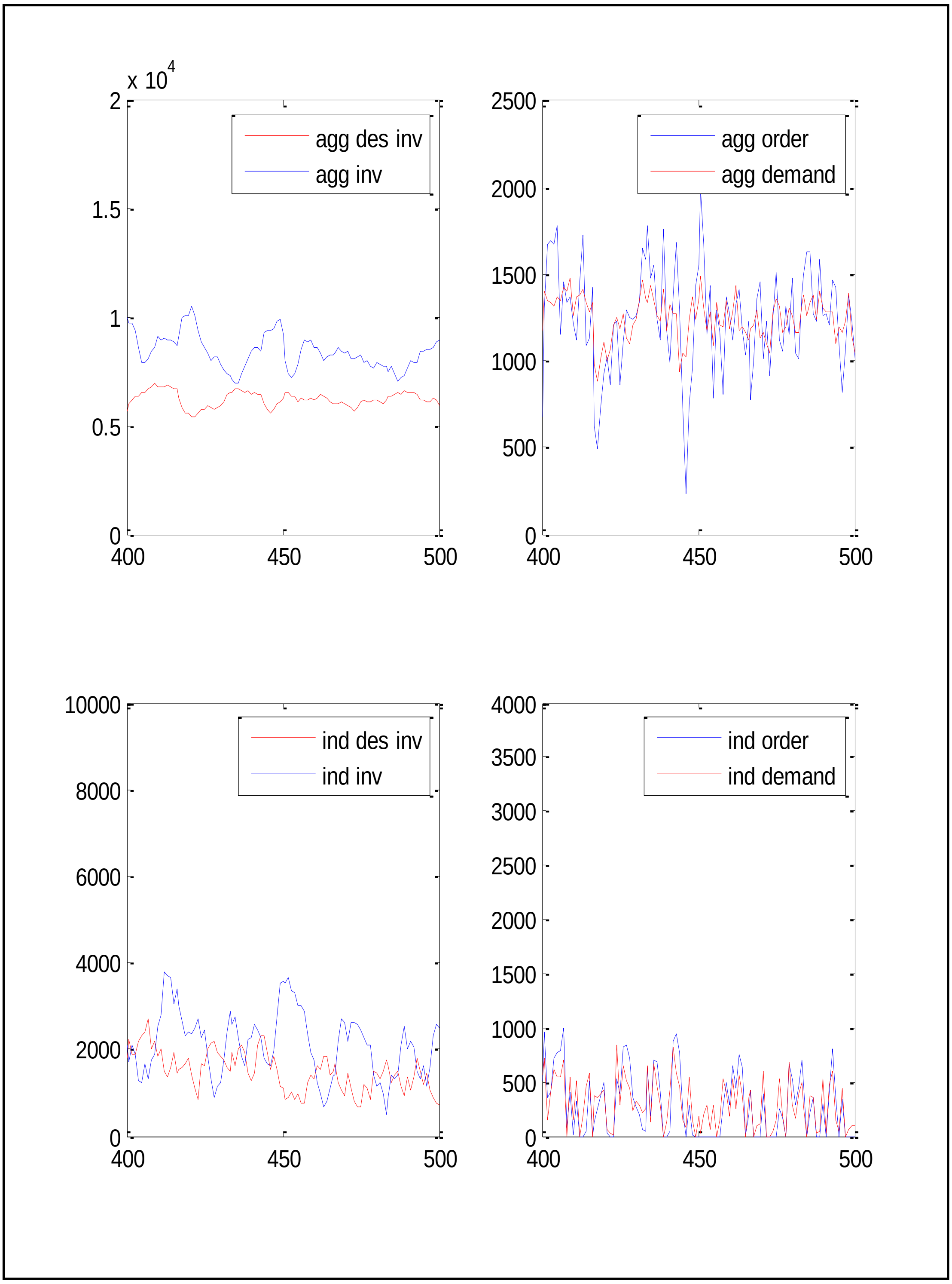


Figure 5.39. Individual and Aggregate Manufacturer Dynamics for A&A Policy (20 Rtl., 10 Whl., 4 Mnf.)

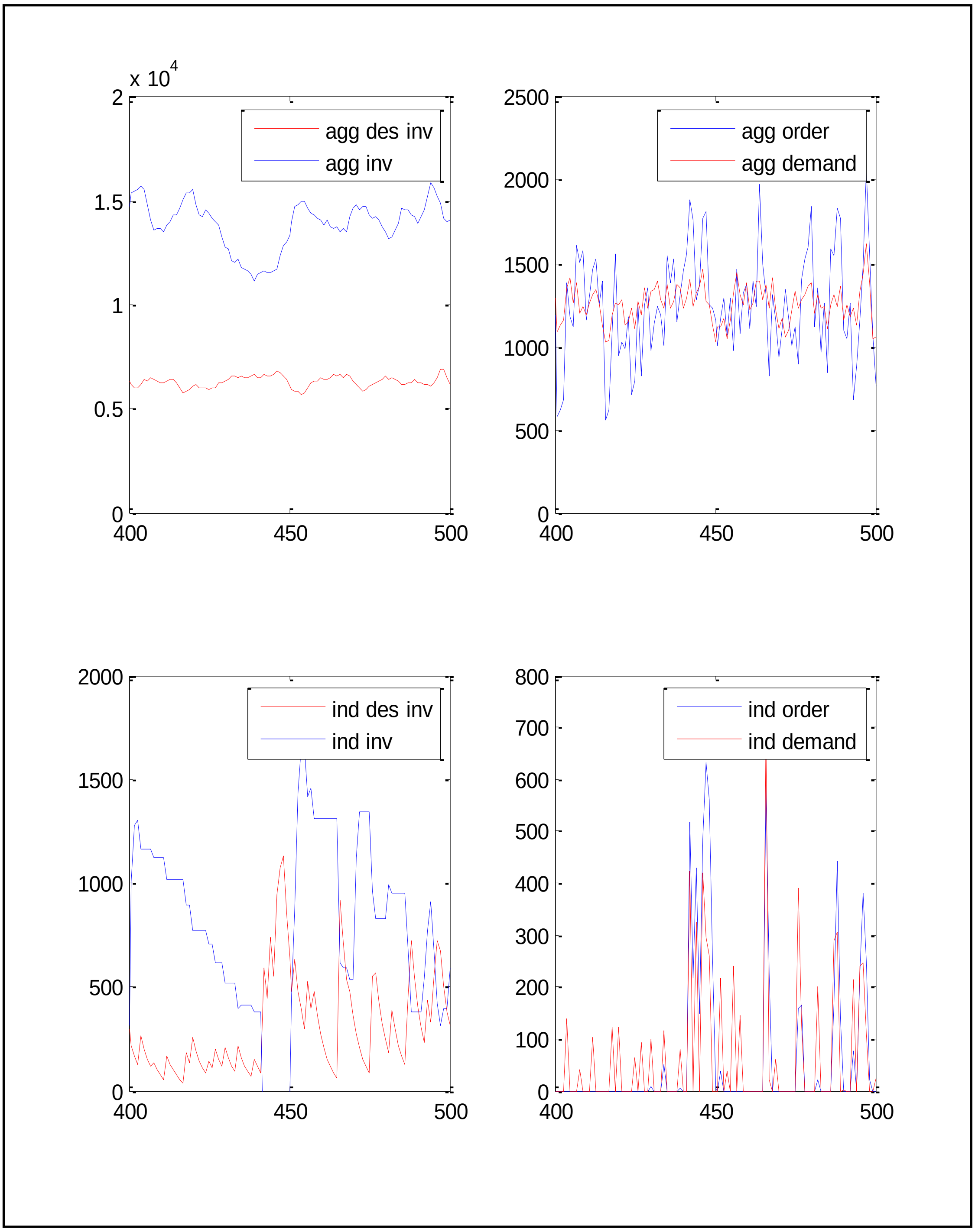


Figure 5.40. Individual and Aggregate Manufacturer Dynamics for A&A Policy
(20 Rtl., 10 Whl., 20 Mnf.)

To summarize the discussion about the relationship between the number of agents and aggregate order dynamics, the relationship is similar to the one under Variable S policy. However, the sensitivity of aggregate order statistics to the number of agents is lower in A&A policy due to the smoothness of orders. “The ratio of the number of agents at the lower echelon to the number of agents at the upper echelon” is critical for aggregate order variability at the upper echelon. If the number of agents at the lower echelon is very high compared to the number of agents at the upper echelon; the demand is distributed almost uniformly to individual agents. Therefore, aggregate order variability is low. As this ratio decreases; the aggregate order variability increases first. Since the number of agents at the lower echelon which demand from an agent at the upper echelon has high deviation, individual demand forecasts have high variability. Therefore, aggregate variability increases as the ratio is decreased until a critical value. If the ratio is further decreased, individual agent orders turn out to exhibit intermittent demand pattern; which reduces the aggregate order variability.

Tables 5.26-5.28 report the changes in aggregate order variance, range, and average when number of agents at only one echelon is varied -keeping other number of agents constant. Aggregate order statistics are consistent with the discussion above about the relationship between the number of agents and aggregate order dynamics. One to one comparison between Tables 5.26-5.28 and Tables 5.6-5.8 indicates that aggregate order variability and its sensitivity to the number of agents is considerably small under A&A policy -relative to the Variable S policy. It can be also observed that “the critical number of agents after which aggregate order variability tends to decrease” is greater than the value under Variable S policy in all cases due to decreased average ordering periods as in Figure 5.42. The effect of the intermittency regime under A&A policy is less apparent due to decreased average ordering periods.

Table 5.26 shows the effect of the number of retailers, when there is one wholesaler and one manufacturer. The sensitivity of aggregate retailer order variance to the number of retailers is quite low. Aggregate retailer order variance drops slightly for very high number (1000) of retailers due to the intermittency in individual orders. The trend in aggregate retailer order variability is preserved at higher echelons.

Table 5.26. Aggregate Order Statistics for the Effect of Number of Retailers under A&A Policy ($N_R$ Retailers, 1 Wholesalers, 1 Manufacturer)

| | | 2 Rtl. | 20 Rtl. | 100 Rtl. | 150 Rtl. | 200 Rtl. | 500 Rtl. | 1000 Rtl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1250.07 | 1249.65 | 1249.99 | 1250.22 | 1249.98 | 1249.87 | 1250.09 |
| | std. dev. | 46.2175 | 46.8907 | 46.1663 | 46.9973 | 47.3184 | 46.2711 | 44.6764 |
| | Range | 369.891 | 365.730 | 389.945 | 358.010 | 353.223 | 358.516 | 383.6948 |
| WHL. | Mean | 1250.12 | 1249.66 | 1250.02 | 1250.22 | 1249.90 | 1249.95 | 1250.152 |
| | std. dev. | 85.8647 | 87.3833 | 84.8683 | 86.4114 | 86.2295 | 82.7209 | 75.0028 |
| | Range | 670.159 | 637.657 | 741.256 | 660.313 | 665.468 | 629.614 | 591.5991 |
| MNF. | Mean | 1250.23 | 1249.73 | 1250.05 | 1250.20 | 1249.77 | 1250.03 | 1250.299 |
| | std. dev. | 207.865 | 211.926 | 204.556 | 208.434 | 207.266 | 196.878 | 172.4827 |
| | Range | 1641.04 | 1560.95 | 1737.13 | 1593.20 | 1613.84 | 1484.28 | 1272.964 |

Table 5.27 shows the effect of the number of wholesalers, when there are 50 retailers and one manufacturer. Aggregate wholesaler order variance increases with the increasing number of wholesalers. The increase in variability saturates for high number (100) of wholesalers. The trend in aggregate wholesaler order variability is preserved at the manufacturer echelon.

Table 5.27. Aggregate Order Statistics for the Effect of Number of Wholesalers under A&A Policy (50 Retailers, $N_W$ Wholesalers, 1 Manufacturer)

| | | 1 Whl. | 2 Whl. | 4 Whl. | 10 Whl. | 20 Whl. | 50 Whl. | 100 Whl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1249.98 | 1250.06 | 1250.24 | 1249.98 | 1249.83 | 1249.85 | 1249.907 |
| | std. dev. | 46.7642 | 46.4414 | 46.918 | 47.08 | 47.104 | 46.8922 | 46.6382 |
| | Range | 363.835 | 364.093 | 389.084 | 393.366 | 353.788 | 379.506 | 358.0407 |
| WHL. | Mean | 1249.95 | 1250.00 | 1250.15 | 1250.01 | 1249.79 | 1249.86 | 1249.902 |
| | std. dev. | 86.5387 | 86.2155 | 87.3494 | 90.6663 | 97.9224 | 102.511 | 101.5788 |
| | Range | 688.275 | 690.203 | 665.500 | 730.089 | 784.609 | 791.772 | 788.1921 |
| MNF. | Mean | 1249.94 | 1249.87 | 1249.86 | 1250.06 | 1249.58 | 1249.79 | 1249.749 |
| | std. dev. | 208.865 | 208.551 | 211.351 | 213.896 | 221.142 | 220.581 | 215.4891 |
| | Range | 1569.63 | 1651.85 | 1670.91 | 1576.28 | 1800.06 | 1635.44 | 1757.551 |

Table 5.28 shows the effect of the number of manufacturers, when there are 20 retailers and 10 wholesalers. The aggregate manufacturer order variance increases until there are eight manufacturers and declines thereafter.

Table 5.28. Aggregate Order Statistics for the Effect of Number of Manufacturers under A&A Policy (20 Retailers, 10 Wholesalers, $N_M$ Manufacturer)

| | | 1 Mnf. | 2 Mnf. | 4 Mnf. | 8 Mnf. | 10 Mnf. | 20 Mnf. | 40 Mnf. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 1249.53 | 1249.94 | 1250.20 | 1249.98 | 1250.10 | 1250.23 | 1249.80 |
| | std. dev. | 47.2539 | 46.6036 | 47.3174 | 46.8053 | 46.5533 | 46.3906 | 46.465 |
| | Range | 352.238 | 360.599 | 362.154 | 372.079 | 406.305 | 393.180 | 380.660 |
| WHL. | Mean | 1249.70 | 1249.92 | 1250.21 | 1249.90 | 1250.01 | 1250.27 | 1249.95 |
| | std. dev. | 115.602 | 114.573 | 115.436 | 114.116 | 113.24 | 114.446 | 115.456 |
| | Range | 866.452 | 918.004 | 959.888 | 924.174 | 1045.13 | 890.151 | 967.550 |
| MNF. | Mean | 1249.99 | 1249.85 | 1250.16 | 1249.72 | 1249.95 | 1249.78 | 1250.30 |
| | std. dev. | 246.397 | 256.336 | 294.435 | 307.979 | 301.430 | 297.921 | 288.375 |
| | Range | 1979.65 | 2217.24 | 2391.60 | 2245.39 | 2147.91 | 2128.67 | 2452.03 |

Figure 5.41 shows how aggregate order mean, standard deviation and max-min range change as a function of the number of agents in case the number of retailers, wholesalers, and manufacturers are the same. The observations are similar to the Variable S policy case in Figure 5.18. Aggregate order mean is not affected from the number of agents. Aggregate wholesaler and manufacturer standard deviation increase significantly when the number of agents is increased to two. Then, it decreases smoothly as the number of agents is increased. Aggregate retailer order variance is not affected as much as the wholesaler and manufacturer variability. Aggregate order range statistics exhibit similar patterns but with some noise. Aggregate order variance and min-max range are significantly smaller than the values under Variable S policy due to smooth ordering patterns.

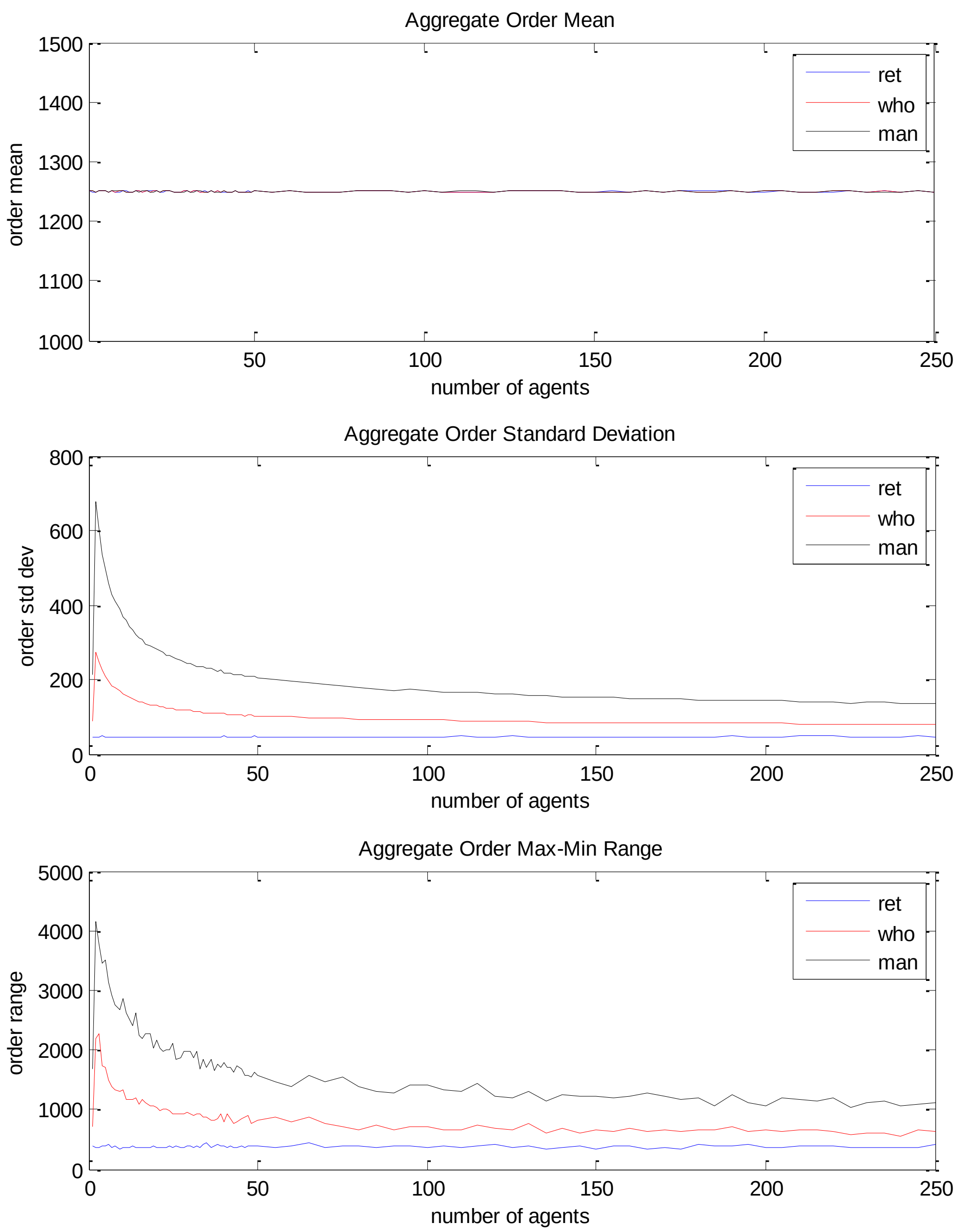


Figure 5.41. Aggregate Order Statistics as Functions of the Number of Agents (All Echelons Equal) for A&A Policy

The decrease in aggregate order variability results from the smoothness of individual orders. The smoothness of individual orders causes average ordering periods at all echelons to decrease. Figure 5.42 shows average ordering periods at all echelons as functions of the number of agents. The comparison between Figure 5.42 and Figure 5.19

indicates that firms tend to give orders more continuously –rather than intermittently- under A&A policy.

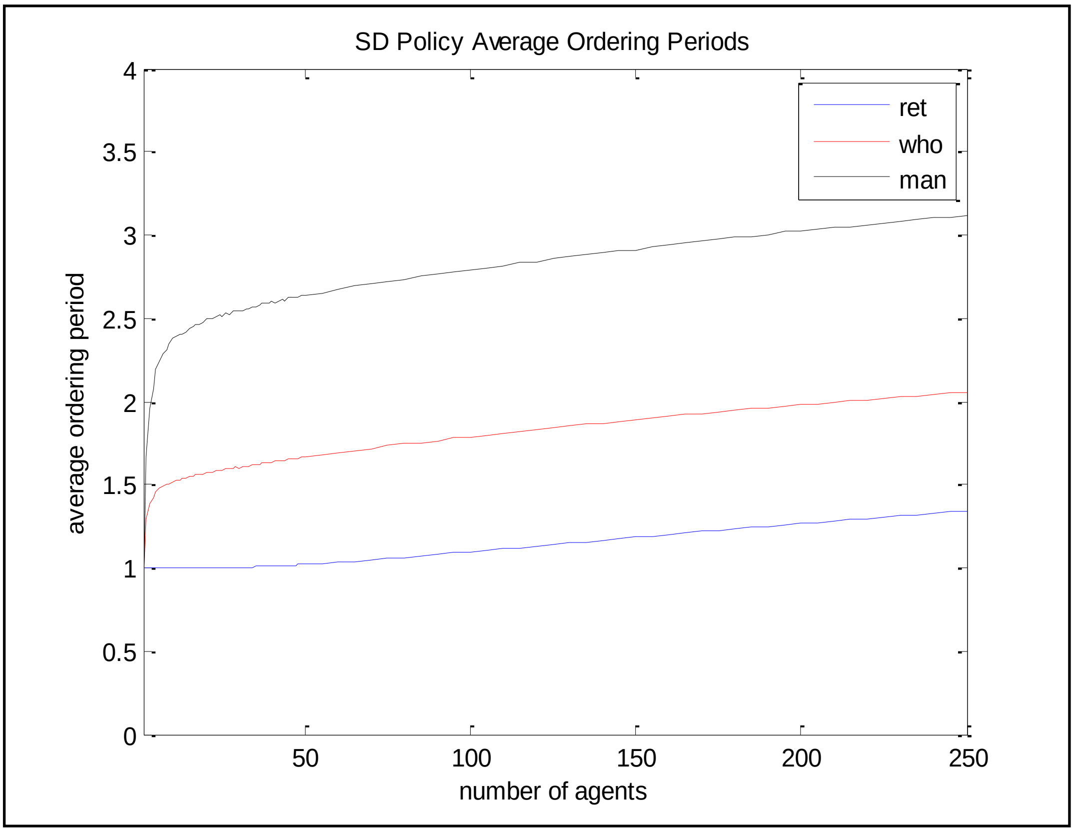


Figure 5.42. Average Echelon Ordering Periods as Functions of the Number of Agents (All Echelons Equal) for A&A Policy

Table 5.29 summarizes the comparison of individual and average order dynamics is concerned. It can be observed that aggregation -of the individual outputs- removes a percentage of the variability existing in individual orders of wholesalers and manufacturers, which is between the reduction in the Fixed S policy and Variable S policy. Anchor-And-Adjust policy facilitates as a transition from Fixed S policy to Variable S policy.

Table 5.29. A&A Policy, Aggregate-Individual Order Variability Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.451963 | 0.272489 | 0.114432 | 0.072739 | 0.035488 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.254082 | 0.170481 | 0.100128 | 0.085235 | 0.028335 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.446001 | 0.271278 | 0.304363 | 0.218583 | 0.077819 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

Figure 5.43 shows aggregate echelon inventory dynamics for the case of 20 retailers, 10 wholesalers, and two manufacturers. It can be observed that the bullwhip effect exists for inventory data. The comparison between Figure 5.43 and Figure 5.20 indicates that aggregate inventory exhibits smoother and oscillatory patterns compared to the aggregate inventory dynamics under Variable S policy. This is most obvious from the manufacturer's plot. Aggregate wholesaler and manufacturer inventory variability is significantly reduced under Anchor-And-Adjust policy. Figures for other experiments with Anchor-And-Adjust policy can be found in Appendix D.

Figure 5.44 shows the power spectral densities of aggregate inventories for the case of 20 retailers, 10 wholesalers and two manufacturers. Spectral power is condensed in the low frequency range around few dominant frequencies. The dominancy of a period in the retailer echelon is conserved at wholesaler and manufacturer echelons. This observation indicates that aggregate inventory dynamics have more predictable patterns under Anchor-And-Adjust policy. Spectral power increases as one moves up in the supply chain, which is an indicator of the bullwhip effect.

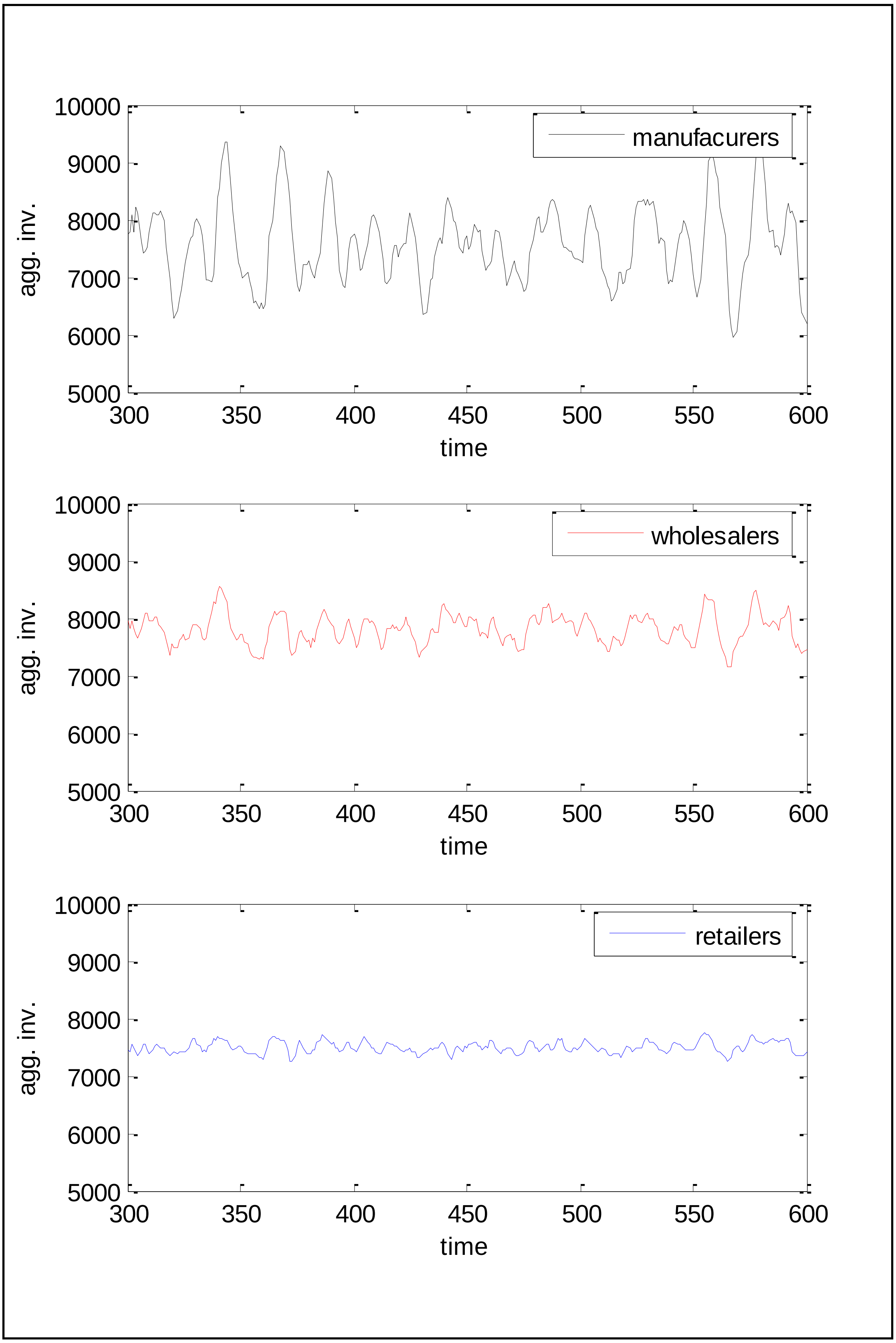


Figure 5.43. Aggregate Inventory Dynamics for A&A Policy

(20 Rtl., 10 Whl., 2 Mnf.)

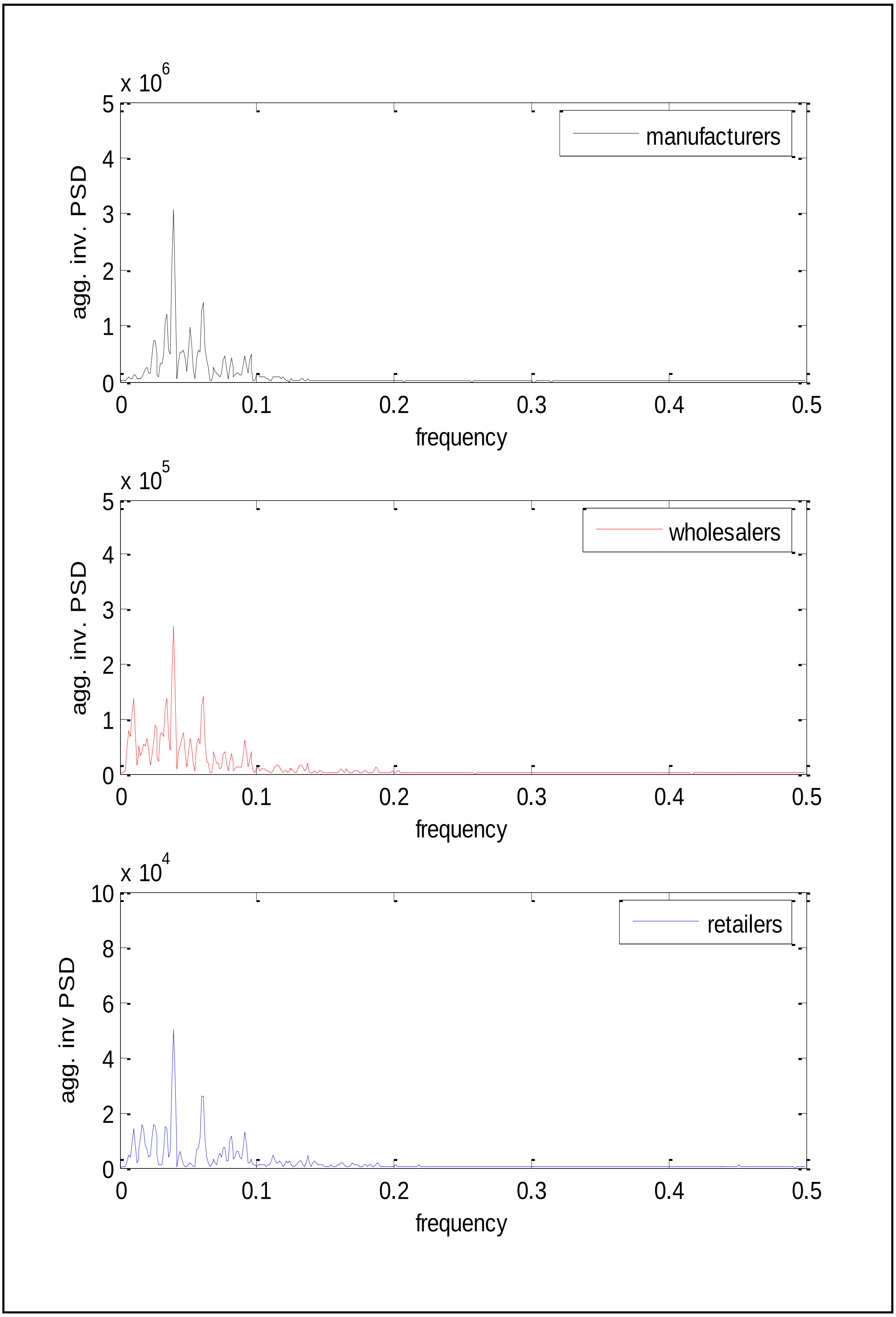


Figure 5.44. Spectral Power Density of Aggregate Inventories for A&A Policy (20 Rtl., 10 Whl., 2 Mnf.)

Table 5.30 gives the statistics of inventory data for the cases in Table 5.25. Aggregate inventory variance and range increase as one moves up in the supply chain. The bullwhip effect in aggregate inventory is a consequence of demand forecasting in Anchor-And-Adjust policy. The comparison between Table 5.30 and Table 5.10 reveals the differences between Anchor-And-Adjust and Variable Order-Up-To-Level policies in terms of aggregate inventory variability. Aggregate inventory variability at retailer echelon is slightly higher in Anchor-And-Adjust policy than in Variable Order-Up-To-Level policy due to the introduction of oscillatory dynamics in the larger time scale. On the other hand, aggregate wholesaler and manufacturer inventory variability are significantly lower under Anchor-And-Adjust policy due to the smoothness of individual inventory dynamics. The sensitivity of aggregate inventory variability to the number of agents is lower under Anchor-And-Adjust policy.

Table 5.30. Aggregate Inventory Statistics for A&A Policy

| | | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|---|
| RTL. | Mean | 7500.773 | 7498.541 | 7498.43 | 7511.107 | 149990.1 |
| | std. dev. | 101.6784 | 105.5676 | 102.931 | 102.1886 | 451.4546 |
| | Range | 928.0303 | 1521.081 | 857.892 | 858.609 | 3354.133 |
| WHL. | Mean | 8019.968 | 8279.877 | 7867.18 | 7830.471 | 151780.5 |
| | std. dev. | 568.9444 | 536.4252 | 261.665 | 226.6299 | 1094.296 |
| | Range | 7457.657 | 6801.982 | 2118.19 | 1790.534 | 10193.5 |
| MNF. | Mean | 9249.305 | 10600.11 | 7619.75 | 7707.112 | 153611.8 |
| | std. dev. | 1694.204 | 1493.704 | 683.150 | 624.4771 | 3648.811 |
| | Range | 13230.74 | 11299.5 | 6046.66 | 5528.834 | 29740.14 |

It is evident from Table 5.30 that the number of agents affects not only the variability but also the mean of aggregate inventory, especially at wholesaler and manufacturer echelons. The relationship between the number of agents and aggregate inventory dynamics is similar to the relationship under Variable S policy. However, the sensitivity to the number of agents is lower under Anchor-And-Adjust policy. Aggregate echelon inventory mean increases as the number of agents at that echelon increases. Although individual intermittent ordering regime provides low variability in aggregate orders; it is not the case for aggregate inventories. Intermittent individual demand and order patterns have prolonged effects on individual inventories through augmenting the individual inventory oscillation period and amplitude as in Figures 5.37-5.40. The oscillatory components at the individual scale are reflected at the aggregate inventories. Thus, aggregate inventory variability increases as the number of agents in that echelon increases. However an exception occurs in Table 5.31 for the case with 1000 retailers.

Tables 5.31-5.33 report the changes in aggregate inventory variance, range, and mean when the number of agents at an echelon is varied -keeping other number of agents constant.

Table 5.31 shows the effect of the number of retailers, when there is one wholesaler and one manufacturer. Aggregate retailer inventory variance increases as the number of retailers increases. However, note that there is a mild decrease in the retailer variance when the number of retailers is increased from 500 to 1000. The tendencies in aggregate wholesaler and manufacturer inventory variances are to first increase and then to decrease as in the aggregate wholesaler and manufacturer order cases. Aggregate retailer inventory mean is augmented by the increase in the number of retailers.

Table 5.31. Aggregate Inventory Statistics for the Effect of Number of Retailers under A&A Policy ($N_R$ Retailers, 1 Wholesalers, 1 Manufacturer)

| | | 2 Rtl. | 20 Rtl. | 100 Rtl. | 150 Rtl. | 200 Rtl. | 500 Rtl. | 1000 Rtl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 7499.81 | 7497.64 | 7595.02 | 7746.41 | 7960.75 | 9714.03 | 16471.4 |
| | std. dev. | 99.8837 | 101.911 | 102.450 | 106.442 | 108.996 | 119.595 | 104.714 |
| | Range | 805.031 | 777.084 | 827.904 | 896.044 | 885.133 | 959.402 | 895.803 |
| WHL. | Mean | 7500.08 | 7497.57 | 7499.74 | 7501.29 | 7500.67 | 7498.47 | 7499.97 |
| | std. dev. | 206.607 | 210.254 | 203.734 | 207.344 | 206.640 | 199.767 | 174.045 |
| | Range | 1661.84 | 1540.94 | 1539.44 | 1664.51 | 1709.35 | 1579.56 | 1296.62 |
| MNF. | Mean | 7499.82 | 7497.49 | 7499.72 | 7501.39 | 7500.52 | 7499.15 | 7499.56 |
| | std. dev. | 614.795 | 628.851 | 613.719 | 617.396 | 614.641 | 587.529 | 498.083 |
| | Range | 5091.17 | 4481.36 | 4436.67 | 4859.95 | 4986.94 | 4534.03 | 3729.13 |

Table 5.32 shows the effect of the number of wholesalers, when there are 50 retailers and one manufacturer. Aggregate wholesaler inventory variance increases as the number of

wholesalers increases. The tendency in aggregate retailer inventory variance is to increase with the rising aggregate wholesaler inventory variability. Aggregate manufacturer inventory variance has the tendency first to increase and then to decrease, as in the case of aggregate wholesaler orders. Aggregate wholesaler inventory mean is augmented by the increase in the number of wholesalers.

Table 5.32. Aggregate Inventory Statistics for the Effect of Number of Wholesalers under A&A Policy (50 Retailers, $N_W$ Wholesalers, 1 Manufacturer)

| | | 1 Whl. | 2 Whl. | 4 Whl. | 10 Whl. | 20 Whl. | 50 Whl. | 100 Whl. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 7512.7 | 7511.83 | 7513.39 | 7512.29 | 7510.20 | 7493.92 | 7460.538 |
| | std. dev. | 101.648 | 100.916 | 102.110 | 103.013 | 104.214 | 106.714 | 109.5724 |
| | Range | 760.133 | 852.017 | 845.308 | 795.232 | 833.784 | 791.663 | 839.82 |
| WHL. | Mean | 7500.15 | 7500.78 | 7503.66 | 7565.47 | 7831.69 | 9005.29 | 11263.65 |
| | std. dev. | 207.106 | 207.251 | 210.193 | 214.717 | 232.218 | 258.976 | 298.67 |
| | Range | 1565.14 | 1632.46 | 1714.42 | 1595.56 | 1855.82 | 1926.71 | 2120.281 |
| MNF. | Mean | 7499.56 | 7500.95 | 7503.44 | 7499.67 | 7500.74 | 7499.54 | 7501.187 |
| | std. dev. | 615.532 | 616.016 | 628.056 | 629.718 | 634.533 | 620.322 | 611.7441 |
| | Range | 4742.42 | 4769.73 | 4777.68 | 4601.07 | 4670.67 | 4761.66 | 5099.419 |

Table 5.33 shows the effect of the number of manufacturers, when there are 20 retailers and 10 wholesalers. Aggregate manufacturer inventory variance increases as the number of manufacturers increases. The tendency in aggregate wholesaler inventory variance is to increase with the rising aggregate manufacturer inventory variability. Aggregate retailer inventory variability is not influenced significantly. Aggregate manufacturer inventory mean is raised by the increase in the number of manufacturers.

Table 5.33. Aggregate Inventory Statistics for the Effect of Number of Manufacturers under A&A Policy (20 Retailers, 10 Wholesalers, $N_M$ Manufacturer)

| | | 1 Mnf. | 2 Mnf. | 4 Mnf. | 8 Mnf. | 10 Mnf. | 20 Mnf. | 40 Mnf. |
|---|---|---|---|---|---|---|---|---|
| RTL. | Mean | 7496.09 | 7498.43 | 7500.44 | 7499.27 | 7500.52 | 7500.29 | 7497.66 |
| | std. dev. | 103.158 | 102.931 | 102.61 | 102.41 | 101.428 | 101.411 | 101.490 |
| | Range | 876.349 | 857.892 | 872.199 | 807.776 | 863.096 | 846.368 | 849.255 |
| WHL. | Mean | 7862.16 | 7867.18 | 7864.83 | 7818.72 | 7797.46 | 7728.59 | 7671.44 |
| | std. dev. | 263.971 | 261.665 | 264.651 | 286.353 | 291.807 | 307.003 | 307.800 |
| | Range | 2098.43 | 2118.19 | 2555.70 | 2886.20 | 2737.37 | 3028.73 | 3002.94 |
| MNF. | Mean | 7495.78 | 7619.75 | 8226.3 | 9761.36 | 10516.2 | 14511.5 | 22546.4 |
| | std. dev. | 678.880 | 683.150 | 779.389 | 869.697 | 890.954 | 1025.87 | 1196 |
| | Range | 4993.93 | 6046.66 | 5850.85 | 6472.02 | 6969.11 | 7365.57 | 9865.85 |

Figure 5.45 shows how aggregate inventory mean, variability and range change as a function of the number of agents in case the number of retailers, wholesalers, and manufacturers are the same. The observations are similar to the ones under Variable S policy. Aggregate inventory averages increase as the number of agents increases. Therefore, the supply chain as a whole tends to hold more excess inventory as the number of agents increases. Aggregate wholesaler and manufacturer inventory standard deviations increase significantly when the number of agents is increased to two. Then, they decrease smoothly as the number of agents is further increased. Aggregate retailer order variability is not affected as much as the wholesaler and manufacturer variability. Min-max range statistics display similar characteristics but with some noise. The comparison between Figure 5.45 and Figure 5.22 reveals the decrease in aggregate inventory variability – especially at manufacturer echelon- due to the smoothness of dynamics under A&A policy.

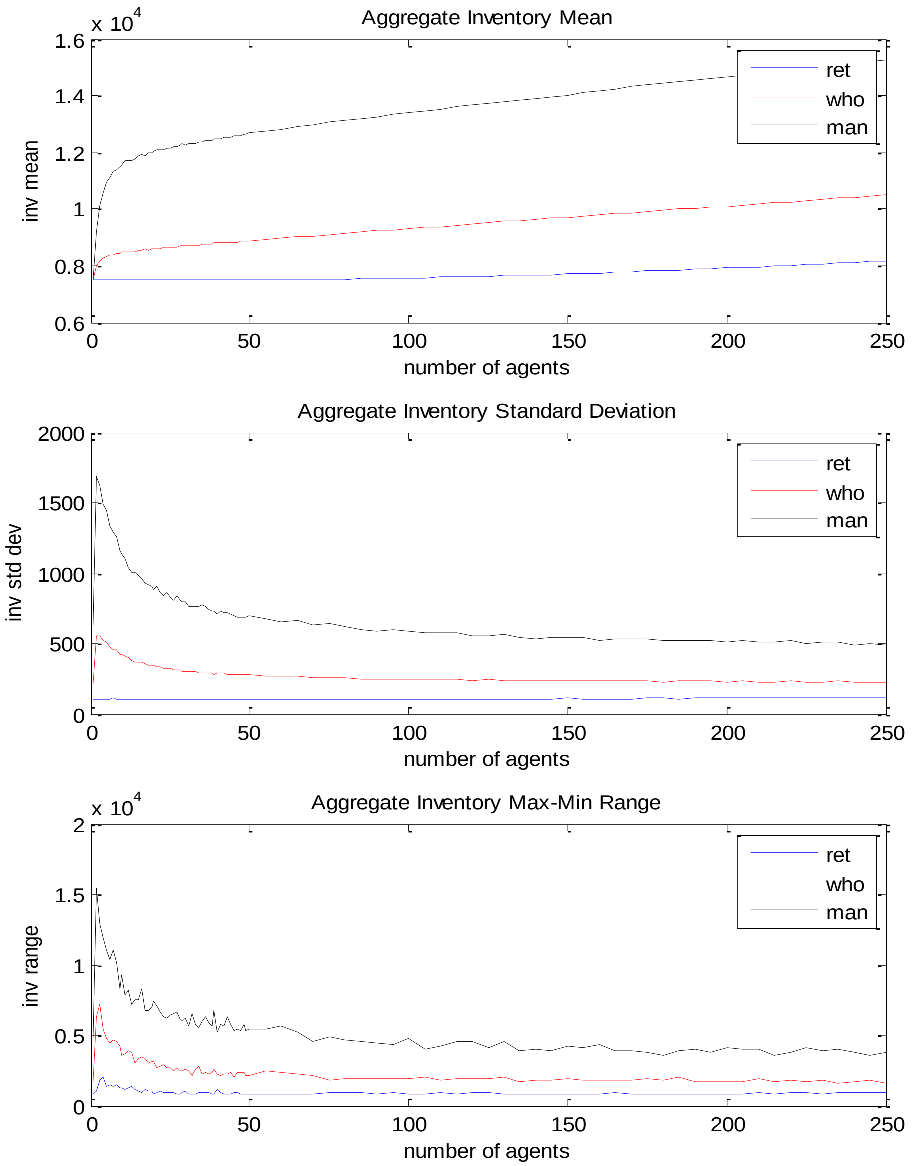


Figure 5.45. Aggregate Inventory Statistics as Functions of the Number of Agents (All Echelons Equal) for A&A Policy

Table 5.34 summarizes the comparison of average inventory dynamics with individual inventory dynamics. The observations regarding the comparison of inventory dynamics comparison are similar to the ones for order dynamics. It can be observed that aggregation -of the individual outputs- removes a percentage of the variability existing in individual inventories of wholesalers and manufacturers, which is between the reduction in

Fixed S and Variable S policies. Anchor-And-Adjust policy facilitates as a transition from Fixed S to Variable S policy.

Table 5.34. A&A Policy, Aggregate-Individual Inventory Variability Comparison

| | R2W2M2 | R4W4M4 | R20W10M2 | R50W20M4 | R200W50M10 |
|---|---|---|---|---|---|
| Actual Rtl. Ratios | 0.452336 | 0.2843 | 0.113919 | 0.072756 | 0.035545 |
| Ref. Rtl. Ratios | 0.707107 | 0.5 | 0.223607 | 0.141421 | 0.070711 |
| Actual Whl. Ratios | 0.22211 | 0.169278 | 0.097386 | 0.087721 | 0.027766 |
| Ref. Whl. Ratios | 0.707107 | 0.5 | 0.316228 | 0.223607 | 0.141421 |
| Actual Mnf. Ratios | 0.399366 | 0.268209 | 0.326927 | 0.242646 | 0.0739 |
| Ref. Mnf. Ratios | 0.707107 | 0.5 | 0.707107 | 0.5 | 0.316228 |

To summarize the observations regarding the Anchor-And-Adjust policy, Anchor-And-Adjust policy is similar to Variable Order-Up-To-Level policy. The main differences between the two policies are that A&A policy: (i) handles inventory and supply lines separately, and (ii) it seeks to eliminate the discrepancies between actual and desired levels smoothly within an interval of adjustment time. Therefore, A&A policy dynamics display characteristics similar to Variable S policy dynamics. However, A&A policy dynamics are smoother.

Bullwhip effect emerges in aggregate orders and inventories due to demand forecasting and inflated lead times. Average ordering period, aggregate order variability and spectral power, aggregate inventory variability and spectral power are amplified as one moves up in the supply chain. All these measures are lower on the average than the values under Variable Order-Up-To-Level policy as a result of the smoothness in individual order dynamics.

As far as the interaction between the supply chain topology and aggregate order dynamics is concerned, conclusions for Anchor-And-Adjust policy are similar to the ones for Variable Order-Up-To-Level policy. However, the sensitivity of aggregate inventory dynamics to the number of agents is lower. “The ratio of the number of agents at the lower echelon to the number of agents at the upper echelon” is critical for the aggregate order variability at the upper echelon. If the number of agents is taken the same at all echelons, aggregate order statistics follow well-defined patterns as a function of the number of agents. Aggregate order mean is not affected from the number of agents. Aggregate order variance and min-max range first increase and then decrease as the number of agents increases.

As far as the relationship between the number of agents and aggregate inventory dynamics is concerned, conclusions for Anchor-And-Adjust policy are similar to the results for Variable Order-Up-To-Level policy. However, the sensitivity of aggregate order dynamics to the number of agents is lower. Aggregate inventory average increases as the number of agents increases. Aggregate inventory variability increases as the number of agents in that echelon increases. If the number of agents is taken the same at all echelons, aggregate inventory statistics follow well-defined patterns as a function of the number of agents. Aggregate inventory average increases as the number of agents increases. Therefore, the system tends to hold excess -surplus- inventory for a supply chain with higher number of firms. Aggregate inventory variance and range first increase and then decrease as the number of agents increases.

As far as the dependency between individual and aggregate order dynamics are concerned, aggregation -of the individual outputs- removes a percentage of the variability existing in individual orders/inventories of wholesalers and manufacturers, which is between the reduction in Fixed S and Variable S policies. Therefore, Anchor-And-Adjust policy facilitates as a transition from Fixed S to Variable S policy.

## 5.3. Aggregate Inventory and Order Dynamics under Adaptive Pricing

In this section, we analyze the relationship between the changes in agent schemata and the aggregate supply chain dynamics under ordering policies considered in Section 5.2. The agents throughout Section 5.2 select their suppliers from the upper echelon randomly and order predetermined amounts independent of the specific suppliers they select. In Section 5.3, agents determine their suppliers according to their inventory positions and prices, and then modify their orders according to the selected suppliers' prices.

### 5.3.1. Adaptive Pricing Formulation

The adaptive pricing formulation here does not claim to model the pricing policies and to reproduce price values of firms in the real market environment. However, it claims to capture the general tendency to adapt the prices according to changing inventory positions and recent prices. This is done by utilizing nonlinear graphical function formulations ubiquitously used in the system dynamics literature. See Sterman (2000) for the formulation of nonlinear graphical functions. At the beginning of each simulation run, all retailer, wholesaler, and manufacturer agents are assigned one of the two alternative plausible pricing functions in Figure 5.46. Agents do not change the assigned pricing functions throughout the simulation. If individual inventory is low with respect to the base stock value -or desired inventory- at any time point, then the agent considers increasing the price. If it is high, then it considers decreasing the price. The price is assumed to be adapted within a range of [0.7-1.3] times the reference price for that echelon. Reference prices $P_R^R$, $P_W^R$, and $P_M^R$ are arbitrarily set to one, two, and three for retailers, wholesalers, and manufacturers respectively. Agents do not use the prices they calculate according to their price functions directly, but take a weighted average of these prices and their prices at the previous step, in order to capture the dependency of the price on its recent values. Equations 5.35-5.37 give the pricing formulations for individual retailers, wholesalers, and manufacturers respectively. Note that Order-Up-To-Level should be replaced by desired inventory, if Anchor-And-Adjust policy is being used.

$$P_{R_i}(t) = (1-\alpha_{R_i}) * P_{R_i}(t-1) + \alpha_{R_i} * (P_R^R * PF_{R_i}(NI_{R_i}(t-1)/S_{R_i}(t-1))), \text{ for } i = 1,2,...,N_R \quad (5.35)$$

$$P_{W_j}(t) = (1-\alpha_{W_j}) * P_{W_j}(t-1) + \alpha_{W_j} * (P_W^R * PF_{W_j}(NI_{W_j}(t-1)/S_{W_j}(t-1))), \text{ for j} = 1,2,...,N_W \quad (5.36)$$

$$P_{M_k}(t) = (1-\alpha_{M_k}) * P_{M_k}(t-1) + \alpha_{M_k} * (P_M^R * PF_{M_k}(NI_{M_k}(t-1)/S_{M_k}(t-1))), \text{ for } k = 1,2,...,N_M \quad (5.37)$$

$P_{R_i}(t)$, $P_{W_j}(t)$, and $P_{M_k}(t)$ are prices of retailer i, wholesaler j, and manufacturer k, respectively. $PF_{R_i}$, $PF_{W_j}$, and $PF_{M_k}$ are pricing functions –exactly one of the two function forms in Figure 5.46- of retailer i, wholesaler j, and manufacturer k, respectively. These two pricing function forms are used in order to introduce heterogeneity and randomness to individual agents. 'Pricing Function 1' in Figure 5.46 reflects a high tendency to adapt the price according to the demand, where 'Pricing Function 2' reflects a low tendency. $\alpha_{R_i}$, $\alpha_{W_j}$, and $\alpha_{M_k}$ are selected as 0.75 for all agents.

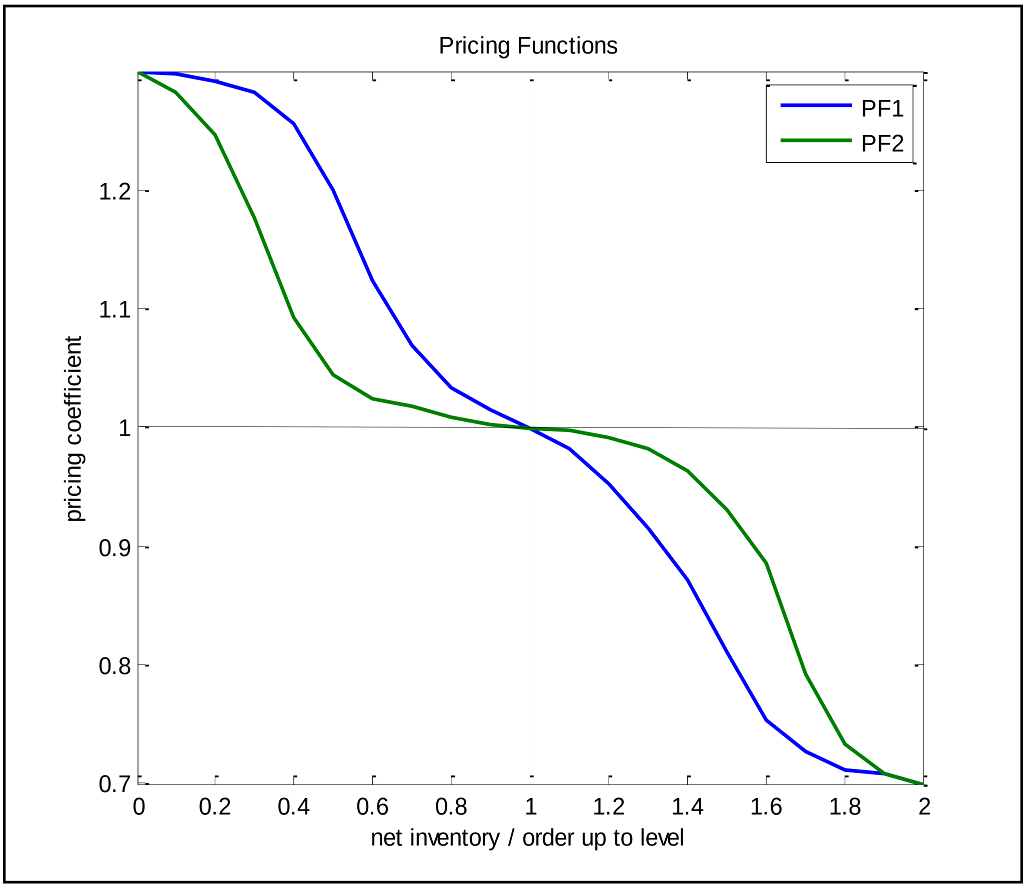


Figure 5.46. Pricing Function for All Retailers, Wholesalers, and Manufacturers

Once individual prices are determined, each agent –other than manufacturers- looks for a suitable supplier that would both be able to supply its demand immediately and will sell for a reasonable price. If a randomly selected supplier satisfies both conditions, then the agent orders from that supplier. Otherwise, the agent continues searching with the same decision rule. The search is carried out until an agent satisfying both constraints is found. If such a supplier does not exist at all, then the last supplier got in contact with in the search is selected.

The price threshold over which an agent rejects the supplier is a multiplier times the agent's expectation of the purchasing price from the upper echelon. Equations 5.38-40 show the price thresholds for individual final customers, retailers, and wholesalers:

$$PT_{FC_l}(t) = \beta_{FC_l} * \hat{P}^R_{FC_l}(t), \text{ for } l = 1, 2, ..., N_{FC} \tag{5.38}$$

$$PT_{R_i}(t) = \beta_{R_i} * \hat{P}^W_{R_i}(t), \text{ for } i = 1, 2, ..., N_R \tag{5.39}$$

$$PT_{W_j}(t) = \beta_{W_j} * \hat{P}^M_{W_j}(t), \text{ for } j = 1, 2, ..., N_W \tag{5.40}$$

$PT_{FC_l}(t)$ is the maximum retailer price final customer l would accept at time t, which is a multiplier ($\beta_{FC_l}$) times its purchase price expectation ($\hat{P}^R_{FC_l}(t)$). $\beta_{FC_l}$ is fixed at the beginning of the simulation run, and is uniformly distributed between 1.0 and 1.2 for each agent. $\hat{P}^R_{FC_l}(t)$ is the price forecast based on the exponential smoothing of the individual retailers' prices that final customer l faces:

$$\hat{P}^R_{FC_l}(t) = \alpha * \hat{P}^R_{FC_l}(t-1) + (1-\alpha) * P^R_{FC_l}(t-1), \text{ for } l = 1, 2, ..., N_{FC} \tag{5.41}$$

where $P^R_{FC_l}(t-1)$ is the last purchase price of final customer l. Exponential smoothing constant α is selected as 0.75.

$PT_{R_i}(t)$ and $PT_{W_j}(t)$ are defined similarly for retailers and wholesalers.

Once a customer selects its supplier, it adjusts its order amount –unadjusted value is formulated as in Section 4.2- according to the selected supplier's price. This relation is modeled again by a nonlinear graphical function formulation, which is illustrated in Figure 5.47. At the beginning of each simulation run, all final customer, retailer, and wholesaler agents are assigned one of these two price-demand functions randomly, and they do not change their specified functions throughout the simulation. If the price of the selected supplier is low (high) compared to the purchasing price expectation, then the customer considers giving a higher (lower) order. Price Demand Function 1(2) reflects low (high) tendency to adapt the ordering amount according to the supplier price.

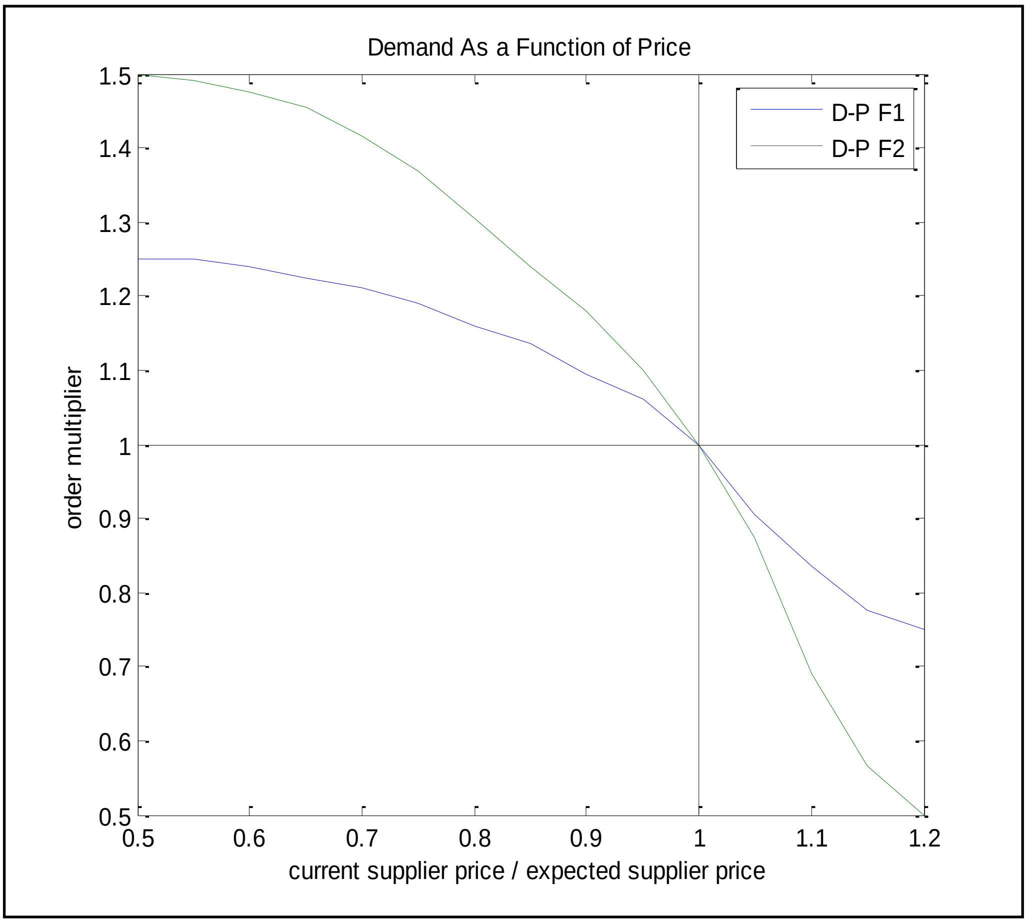


Figure 5.47. Function of Order Modification According to Supplier Price for All Retailers, Wholesalers, and Manufacturers

Table 5.35. Aggregate Order Statistics for the Impact of Adaptive Pricing

| | | Fixed S Policy | Variable S Policy | (s, S) Policy | A&A Policy |
|---|---|---|---|---|---|
| RTL. WITHOUT PRICING | Mean | 1250.164 | 1250.166 | 1249.764 | 1250.182 |
| | Std. Dev. | 32.2224 | 82.6379 | 444.8975 | 47.134 |
| | Range | 245.2582 | 669.7697 | 3277.083 | 443.7717 |
| RTL. WITH PRICING | Mean | 1234,40 | 1236,30 | 1226,20 | 1225,10 |
| | Std. Dev. | 43,27 | 156,82 | 443,69 | 101,68 |
| | Range | 395,54 | 1274,40 | 3091,90 | 908,30 |
| WHL. WITHOUT PRICING | Mean | 1250.164 | 1250.217 | 1250.069 | 1250.183 |
| | Std. Dev. | 32.2224 | 305.7974 | 915.3748 | 116.2191 |
| | Range | 245.2582 | 2403.34 | 6420.852 | 935.9313 |
| WHL. WITH PRICING | Mean | 1234,40 | 1236,30 | 1225,30 | 1225,20 |
| | Std. Dev. | 47,49 | 468,34 | 962,34 | 259,85 |
| | Range | 420,82 | 3573,90 | 5898,30 | 1826,70 |
| MNF. WITHOUT PRICING | Mean | 1250.164 | 1250.33 | 1250.582 | 1250.086 |
| | Std. Dev. | 32.2224 | 935.4818 | 2121.911 | 261.7911 |
| | Range | 245.2582 | 5787.676 | 14540.33 | 2443.842 |
| MNF. WITH PRICING | Mean | 1234,40 | 1236,40 | 1225,80 | 1225,20 |
| | Std. Dev. | 47,49 | 1286,00 | 2191,60 | 401,62 |
| | Range | 420,82 | 7714,50 | 14353,00 | 2861,50 |

Table 5.36. Aggregate Inventory Statistics for the Impact of Adaptive Pricing

| | | Fixed S Policy | Variable S Policy | (s, S) Policy | A&A Policy |
|---|---|---|---|---|---|
| RTL. WITHOUT PRICING | Mean | 2249.506 | 5755.476 | 4172.132 | 7500.23 |
| | Std. Dev. | 56.015 | 95.6173 | 318.5992 | 102.994 |
| | Range | 412.0033 | 986.5683 | 2632.587 | 876.5538 |
| RTL. WITH PRICING | Mean | 2289,40 | 5809,10 | 5090,70 | 7603,60 |
| | Std. Dev. | 54,75 | 148,05 | 418,20 | 163,38 |
| | Range | 415,71 | 1266,80 | 3236,30 | 1420,40 |
| WHL. WITHOUT PRICING | Mean | 10249.51 | 6607.647 | 7326.557 | 7867.503 |
| | Std. Dev. | 56.015 | 356.3823 | 1019.013 | 260.6762 |
| | Range | 412.0033 | 3035.97 | 9542.333 | 2464.122 |
| WHL. WITH PRICING | Mean | 10293,00 | 7339,10 | 8056,50 | 8096,80 |
| | Std. Dev. | 59,26 | 641,85 | 1289,10 | 439,36 |
| | Range | 449,34 | 5637,00 | 9741,40 | 3654,20 |
| MNF. WITHOUT PRICING | Mean | 17749.18 | 6585.143 | 7446.904 | 7617.835 |
| | Std. Dev. | 72.633 | 1131.066 | 2720.012 | 697.0304 |
| | Range | 562.968 | 9227.593 | 20389.5 | 5345.426 |
| MNF. WITH PRICING | Mean | 17828,00 | 7590,50 | 8404,00 | 7633,00 |
| | Std. Dev. | 68,20 | 1762,90 | 3313,60 | 824,73 |
| | Range | 532,61 | 13942,00 | 26210,00 | 7540,50 |

### 5.3.2. Aggregate Inventory and Order Dynamics under Adaptive Pricing for Fixed Order-Up-To-Level Policy

This section investigates the impact of adaptive pricing on aggregate supply chain dynamics, when firms use Fixed S policy. Figure 5.48 and Figure 5.49 show aggregate order and inventory dynamics at each echelon for the case of 20 retailers, 10 wholesalers and two manufacturers.

The behavioral changes in aggregate supply chain orders can be discovered by comparing Figure 5.48 with Figure 5.1. It can be observed that the behavioral pattern is not altered due to adaptive prices; however the aggregate order variability is reinforced -to a degree- at all echelons. Minor decreases in aggregate order means are observed. Closer investigation of aggregate orders shows that wholesaler and manufacturer orders are exactly the same; because individual manufacturers give orders at amounts exactly equal to their demands as a result of using constant Order-Up-To-Levels. It can be also observed that the differences between retailer and wholesaler aggregate orders are indeed not significant. Since the variability in inventories is low compared to the Order-Up-To-Levels in higher echelons; pricing mechanism becomes ineffective for echelons higher than retailers in Fixed S policy, when there are 20 retailers, 10 wholesalers and two manufacturers. To summarize the effect of pricing on aggregate orders under Fixed S policy, aggregate retailer order variability is reinforced significantly as a result of increased variability in individual retailer demands due to adaptive pricing. Aggregate order variability at higher echelons is amplified mainly as a result of the increase in order variability at the retailer echelon. Pricing mechanism becomes ineffective for higher echelons, and thus the aggregate order variability is only slightly further amplified at higher echelons.

Figure 5.49 presents the aggregate inventory dynamics under adaptive pricing for Fixed S policy. The comparison between Figure 5.49 and Figure 5.7 reveals the impact of adaptive pricing on aggregate inventory. The smoothness in aggregate inventory dynamics is attenuated by adaptive pricing. The comparison between Figure 5.50 and Figure 5.8 reveals the changes in the behavioral pattern by utilizing spectral information. Adaptive pricing introduces high frequency oscillatory components while suppressing lower frequency components to a degree. It can be also observed from Figure 5.49 that aggregate inventory averages are augmented as a consequence of adaptive pricing.

Table 5.35 and Table 5.36 summarize the impact of adaptive pricing on aggregate order and inventory dynamics by giving statistics for the supply chain with and without adaptive pricing under all ordering policies. Table 5.35 demonstrates that aggregate retailer order variability is amplified due to adapting prices, when Fixed S policy is being used. This amplification influences the aggregate order variability at higher echelons. Aggregate wholesaler order variability is slightly higher than retailer variability; however the increase is not significant. Aggregate order variability is exactly the same at wholesaler and manufacturer echelons. Aggregate order mean is slightly decremented at the retailer echelon and remains about that level at wholesaler and manufacturer echelons.

Table 5.36 demonstrates that aggregate inventory averages are increased by adaptive pricing. On the other hand, aggregate inventory variability is not significantly altered despite the significant increase in aggregate order variability. Counter-intuitive slight decreases in aggregate retailer and manufacturer inventories are observed. However, Section 6.2.1 shows that adaptive pricing causes significant increases in aggregate inventory variability at all echelons for high number of agents. Therefore, it can be inferred that the effectiveness of the pricing mechanism requires individual inventories to be comparable with Order-Up-To-Levels.

To summarize the observations on the effects of adaptive pricing under Fixed S policy, the pricing mechanism amplifies aggregate retailer order variability significantly and the variability is sustained at upper echelons of the chain. Aggregate inventory variability is slightly affected despite the significant increase in order variability. Even counter-intuitive variability decreases are observed. However, significant increases in aggregate inventory variability are observed at all echelons for high number of agents. The effectiveness of the pricing mechanism requires individual inventories to be comparable with Order-Up-To-Levels. Adaptive pricing affects also the behavioral pattern of aggregate inventories: it introduces high frequency oscillatory components while suppressing lower frequency components. A further significant effect of pricing is the augmentation of aggregate inventory averages at all echelons.

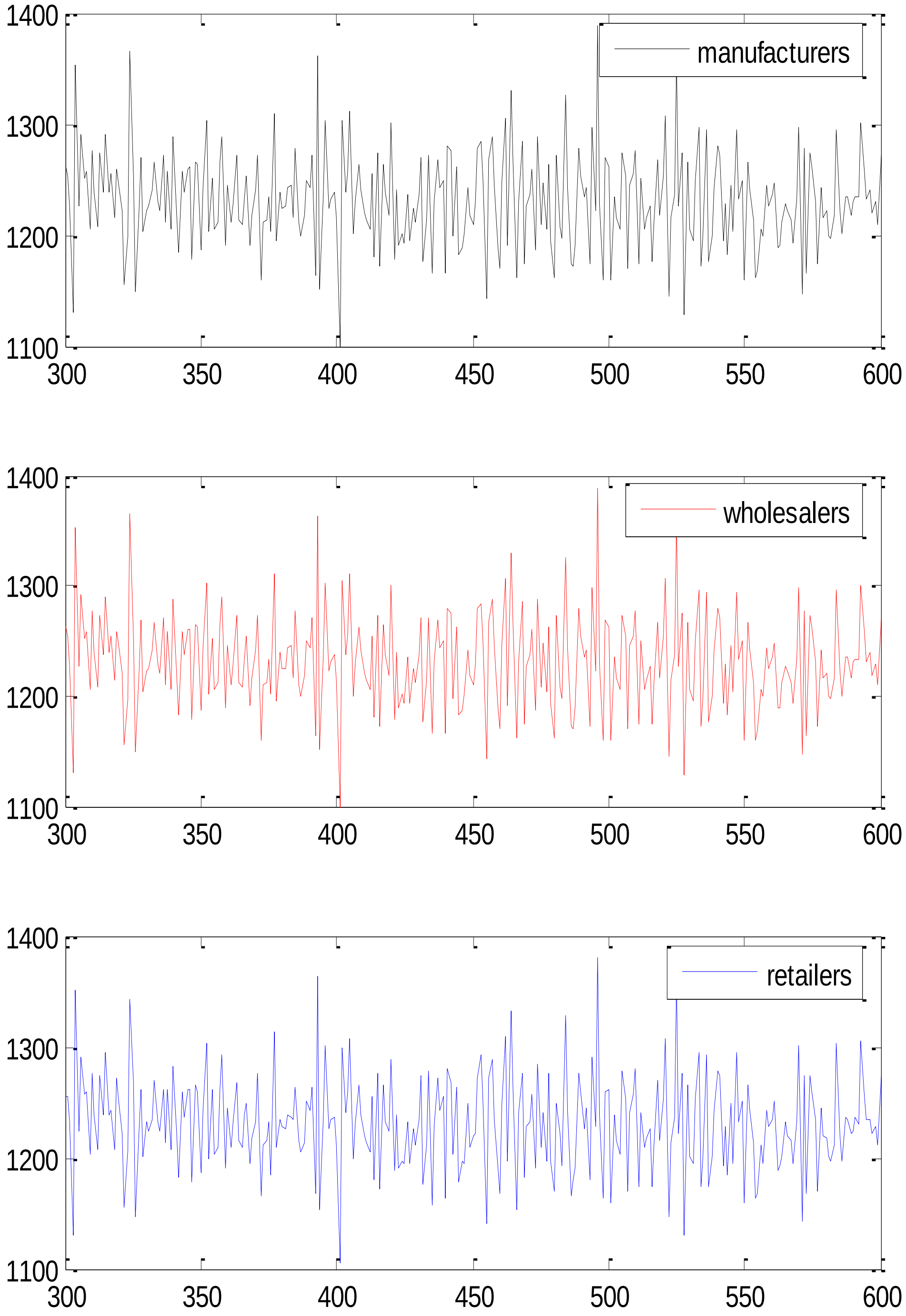


Figure 5.48. Aggregate Order Dynamics under Adaptive Pricing for Fixed S Policy (20 Rtl., 10 Whl., 2 Mnf.)

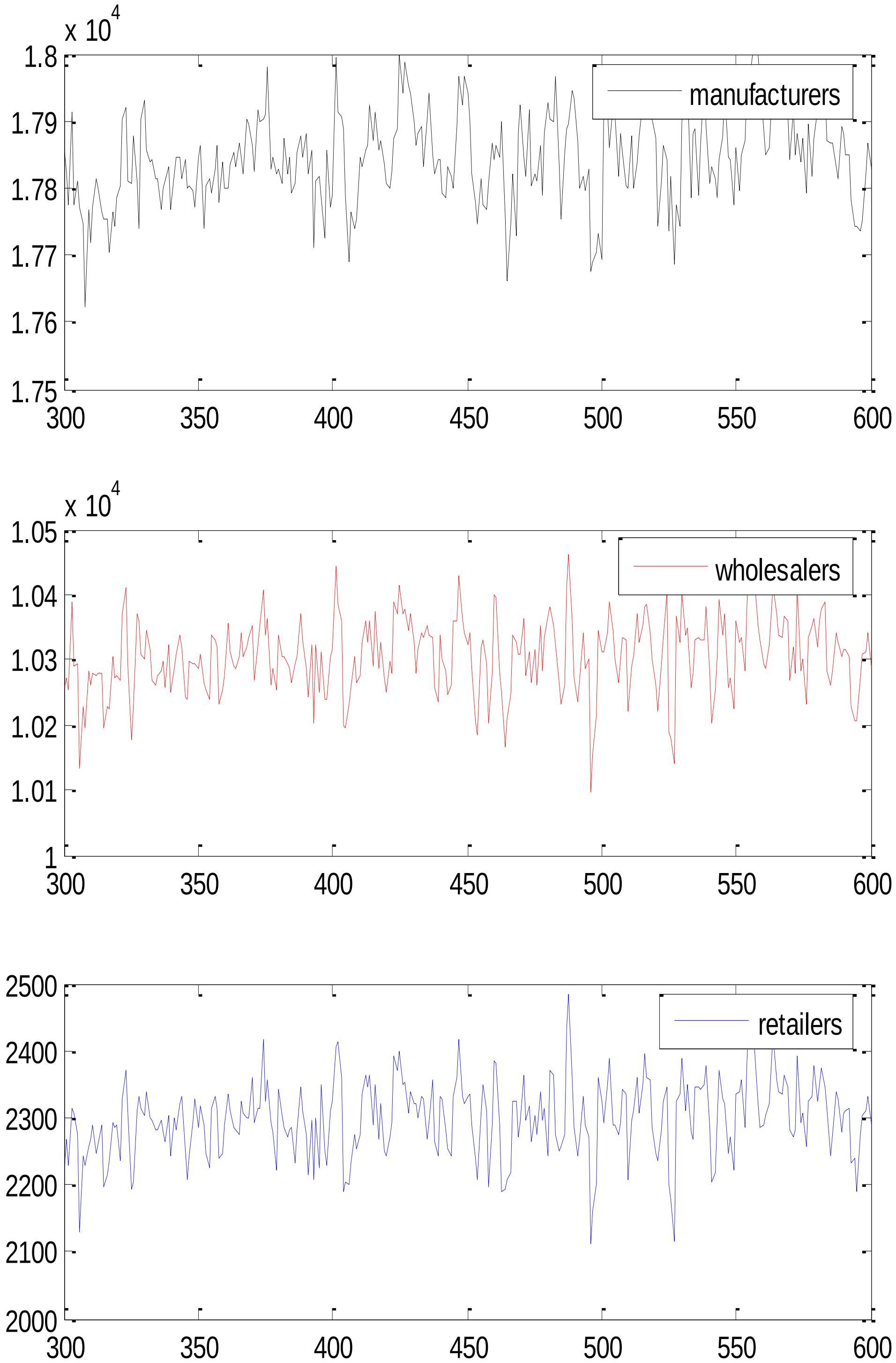


Figure 5.49. Aggregate Inventory Dynamics under Adaptive Pricing for Fixed S Policy (20 Rtl., 10 Whl., 2 Mnf.)

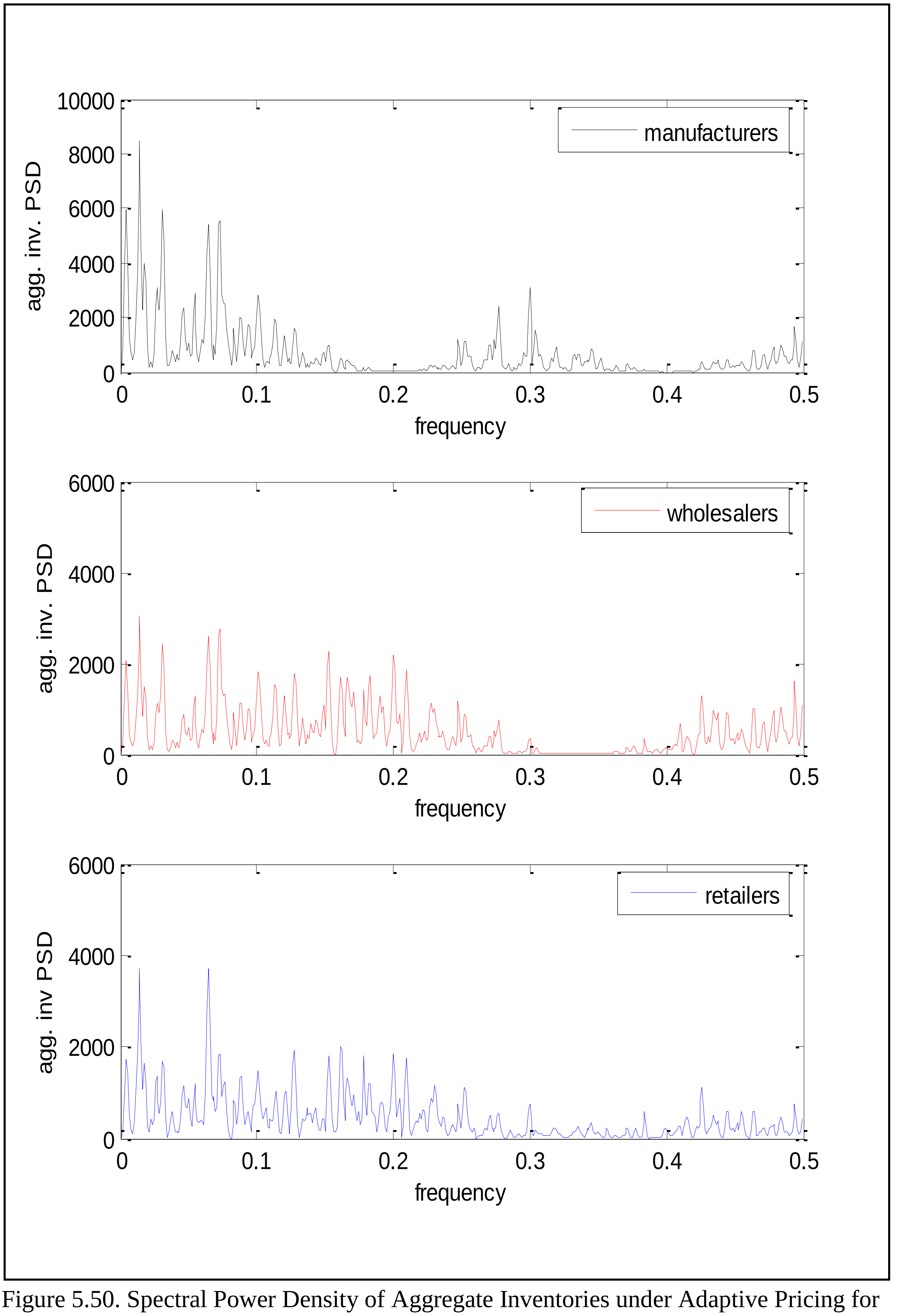


Figure 5.50. Spectral Power Density of Aggregate Inventories under Adaptive Pricing for Fixed S Policy (20 Rtl., 10 Whl., 2 Mnf.)

### 5.3.3. Aggregate Inventory and Order Dynamics under Adaptive Pricing for Variable Order-Up-To-Level Policy

The behavioral changes in aggregate supply chain orders can be discovered by comparing Figure 5.51 with Figure 5.9. The fluctuating pattern is not altered. However, aggregate order variability is reinforced intensely at all echelons. Minor decreases in aggregate order means are again observed. The comparison between Figure 5.52 and Figure 5.20 reveals the impact of adaptive pricing on aggregate inventory. The oscillatory pattern is preserved; however the dominant periods and variability change. Both the oscillation amplitude and the variability of aggregate inventories are significantly amplified by adaptive pricing. It can be also observed that aggregate inventory averages are increased by adaptive pricing. Figure 5.53 shows the spectral power density for aggregate inventory dynamics. The comparison between Figure 5.53 and Figure 5.21 reveals an overall increase in spectral power and at the same time changes in the dominance of periods (notice the difference in y-ranges in the comparison). The range at which spectral power is concentrated is split into two, and this splitting causes significant periodicities at low frequencies -and also at higher ones.

Table 5.35 demonstrates that aggregate order variability is amplified intensely at all echelons, as a consequence of adaptive prices. Aggregate order mean is slightly decremented at the retailer echelon and remains about that level at wholesaler and manufacturer echelons. Table 5.36 demonstrates that aggregate inventory variability is also intensely amplified at all echelons. Aggregate inventory averages are also significantly increased by adaptive pricing.

To summarize the observations on the effect of adaptive prices under Variable S policy, it introduces significant variability to both aggregate orders and inventories, when coupled with demand forecasting. Aggregate inventory mean increases significantly at all echelons as a result of adaptive pricing. Adaptive pricing causes behavioral pattern changes in aggregate inventories. The range at which spectral power is concentrated is split into two, and this splitting causes significant periodicities at low frequencies.

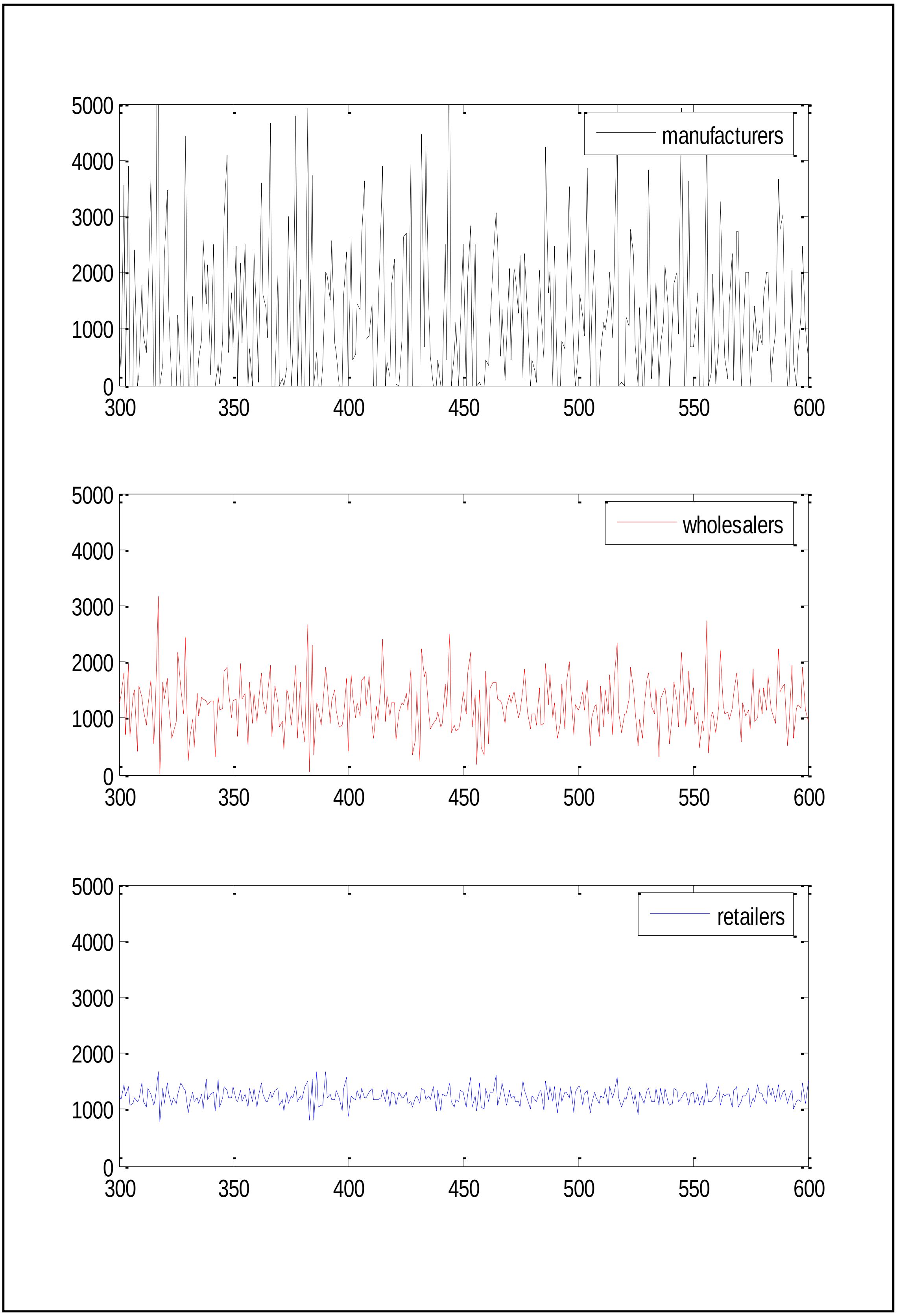


Figure 5.51. Aggregate Order Dynamics under Adaptive Pricing for Variable S Policy (20 Rtl., 10 Whl., 2 Mnf.)

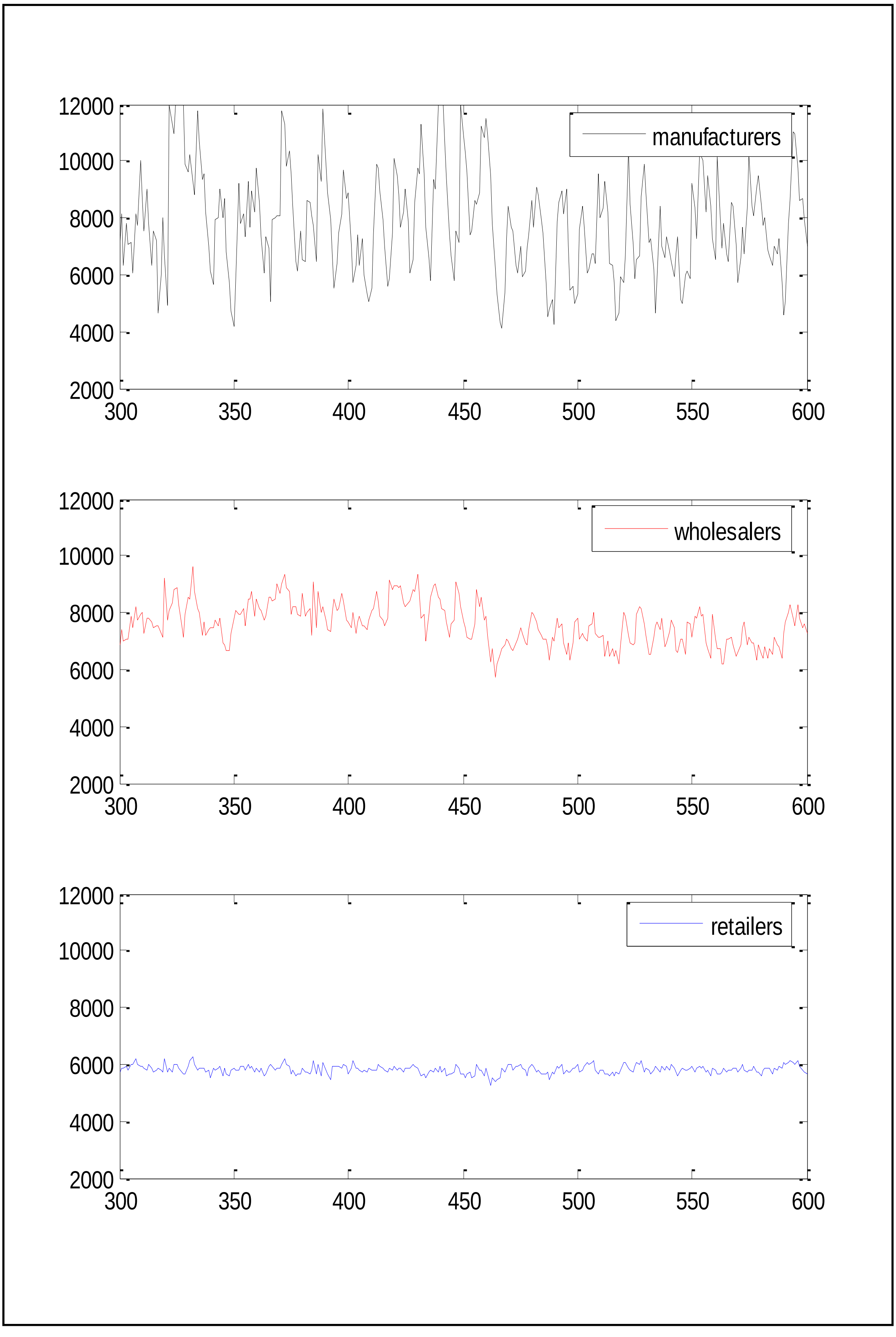


Figure 5.52. Aggregate Inventory Dynamics under Adaptive Pricing for Variable S Policy (20 Rtl., 10 Whl., 2 Mnf.)

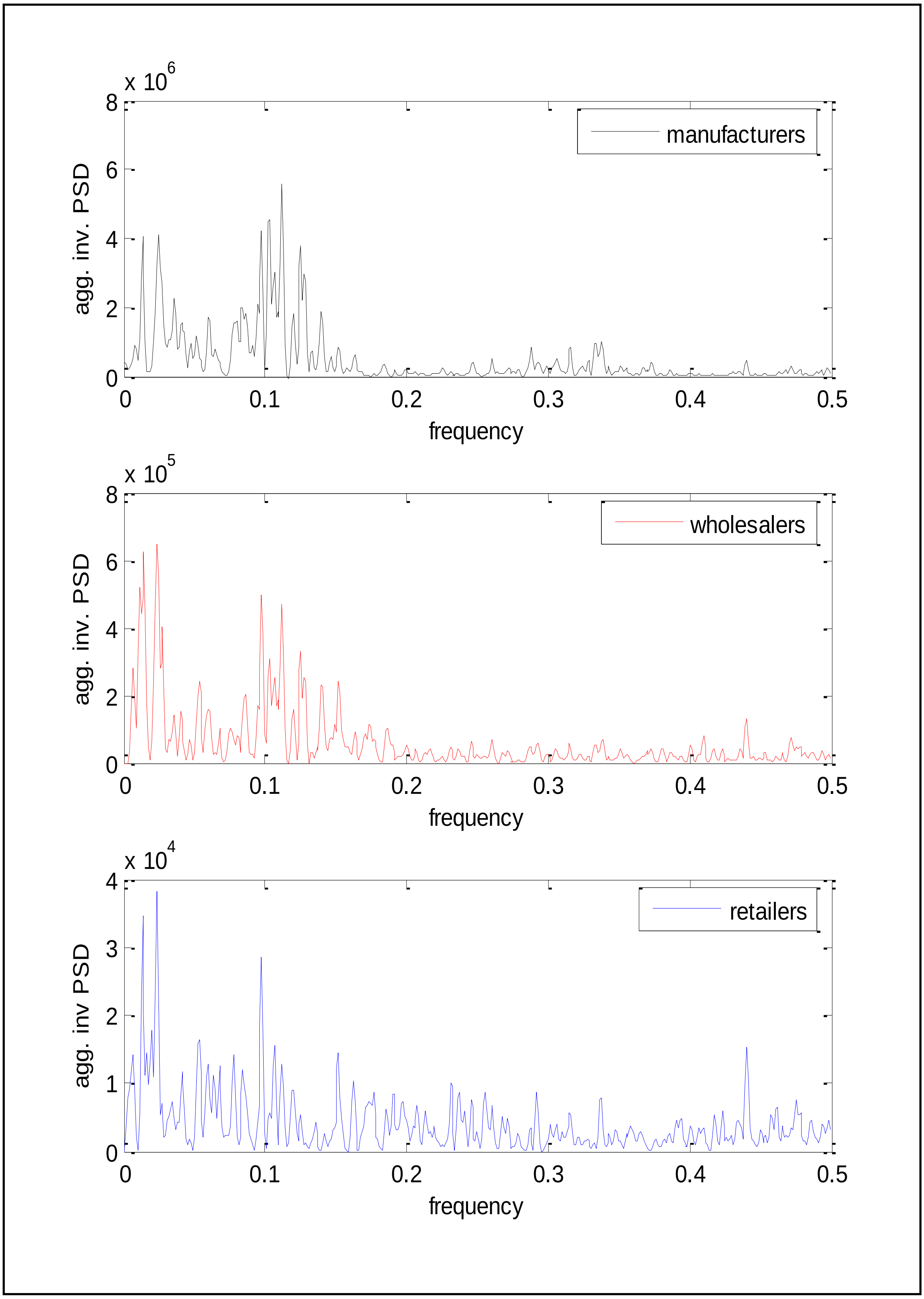


Figure 5.53. Spectral Power Density of Aggregate Inventories under Adaptive Pricing for Variable S Policy (20 Rtl., 10 Whl., 2 Mnf.)

### 5.3.4. Aggregate Inventory and Order Dynamics under Adaptive Pricing for Reorder Point - Order-Up-To-Level Policy

The behavioral changes in aggregate supply chain orders can be discovered by comparing Figure 5.54 with Figure 5.23. The behavioral pattern and variability of aggregate orders are not affected dramatically. The comparison between Figure 5.55 and Figure 5.32 reveals the impact of adaptive pricing on aggregate inventory dynamics. Oscillation amplitude and variability of aggregate inventories are amplified to a degree by adaptive pricing. It can be also observed that aggregate inventory averages are augmented. The comparison between Figure 5.56 and Figure 5.33 reveals the concentration of spectral power around few dominant periods (notice the difference in y-ranges in the comparison). The range in which spectral power is concentrated is split and spectral power is concentrated around small number of frequencies in both higher and lower frequencies. Low frequencies with significantly high spectral power emerge as a result.

Table 5.35 demonstrates that aggregate retailer order variability is not affected significantly. The variability of aggregate wholesaler and manufacturer orders are only slightly amplified. Aggregate order mean is slightly decremented at the retailer echelon and remains about that level at wholesaler and manufacturer echelons. Table 5.36 demonstrates that aggregate inventory variability is amplified mildly at all echelons. Aggregate inventory averages are also significantly increased by adaptive pricing.

To summarize the observations on the effect of adaptive prices under (s,S) policy, the variability in aggregate order dynamics are mostly caused by order batching, i.e. the contribution of adaptive pricing to aggregate order variability is inferior. Aggregate inventory variability is amplified to a degree at all echelons. Adaptive pricing causes behavioral pattern changes in aggregate inventories. The spectral power is concentrated around few dominant periods, including low frequency ones. As a further observation, aggregate inventory mean increases at all echelons due to adaptive pricing.

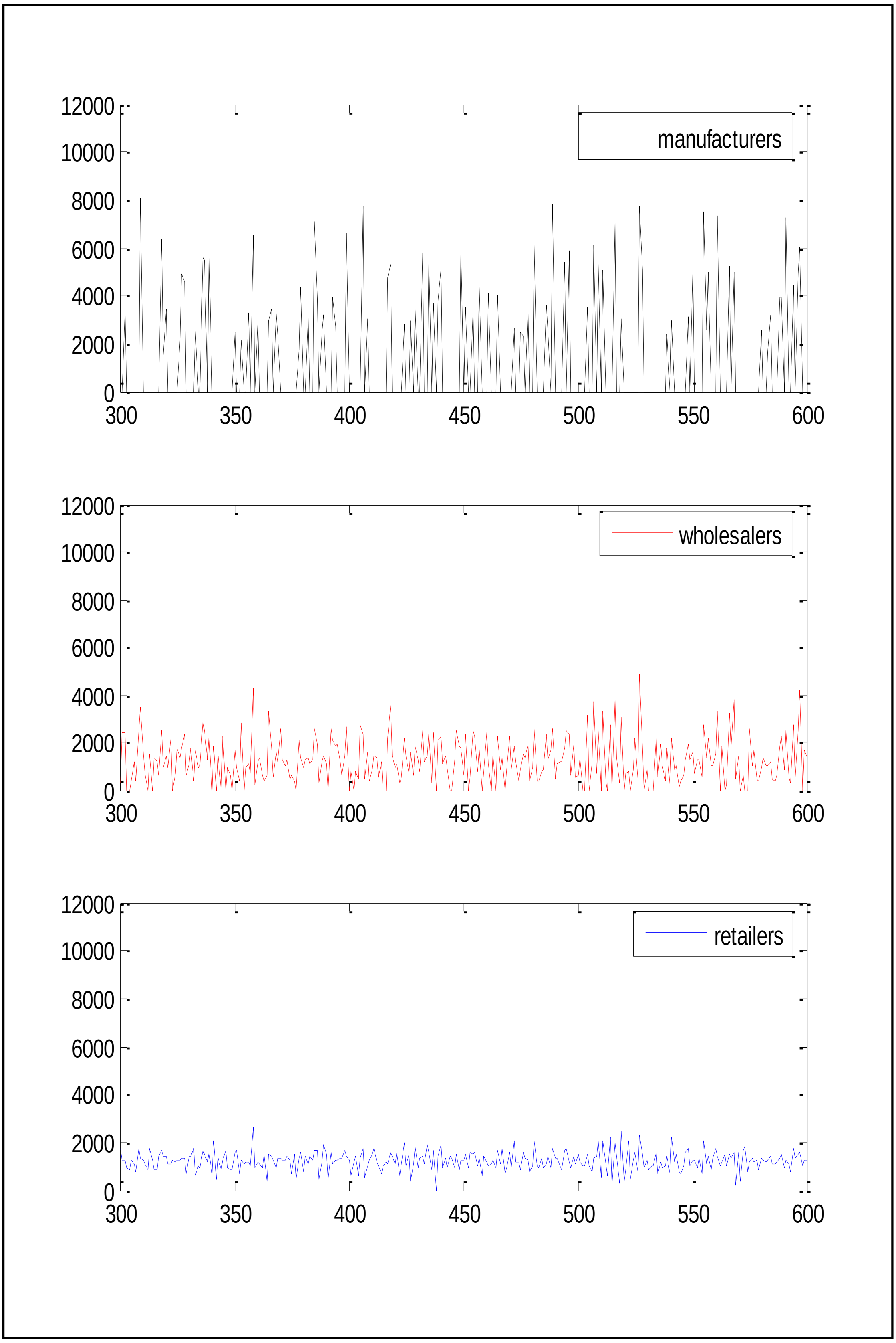


Figure 5.54. Aggregate Order Dynamics under Adaptive Pricing for (s,S) Policy (20 Rtl., 10 Whl., 2 Mnf.)

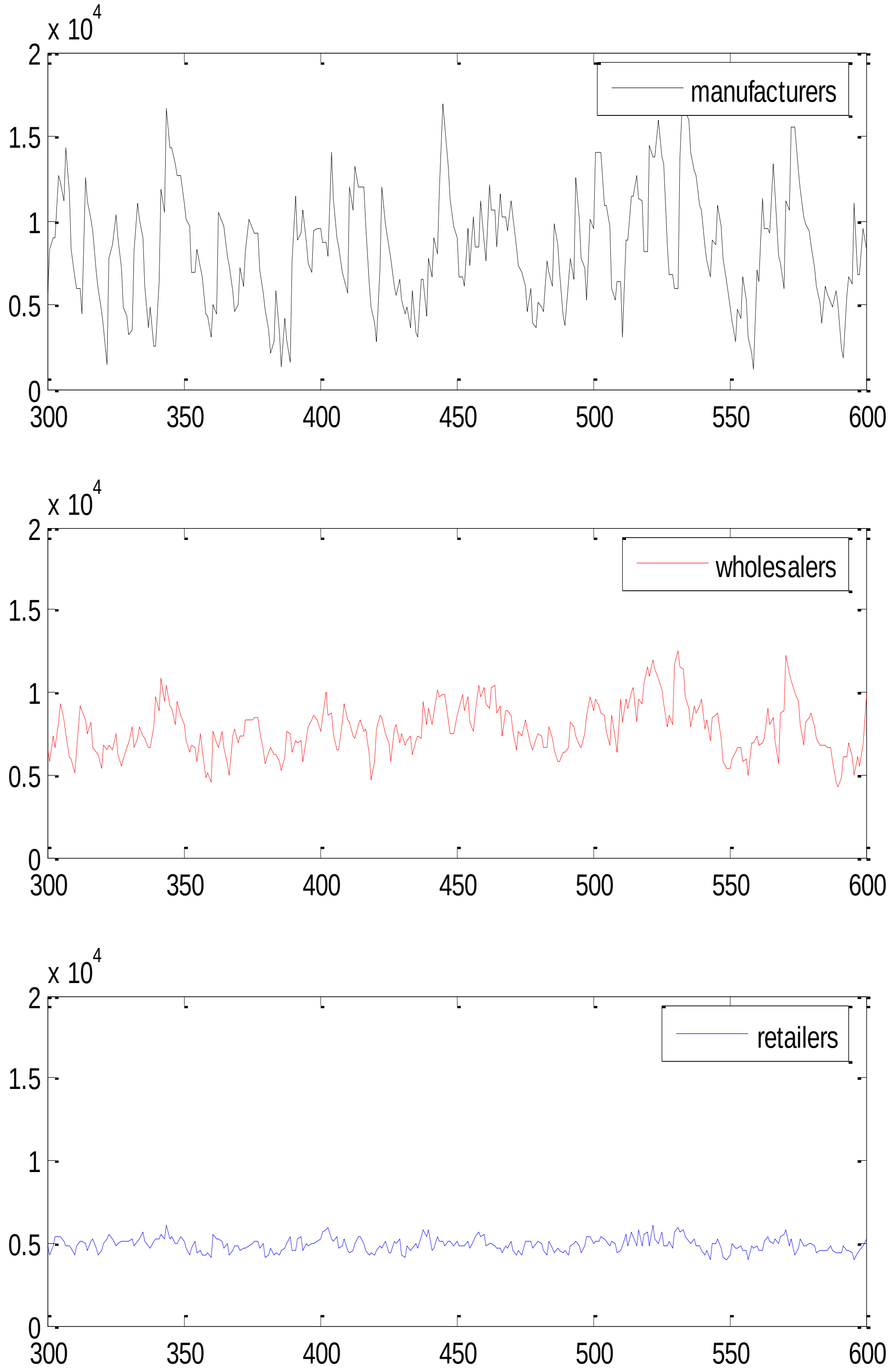


Figure 5.55. Aggregate Inventory Dynamics under Adaptive Pricing for (s,S) Policy (20 Rtl., 10 Whl., 2 Mnf.)

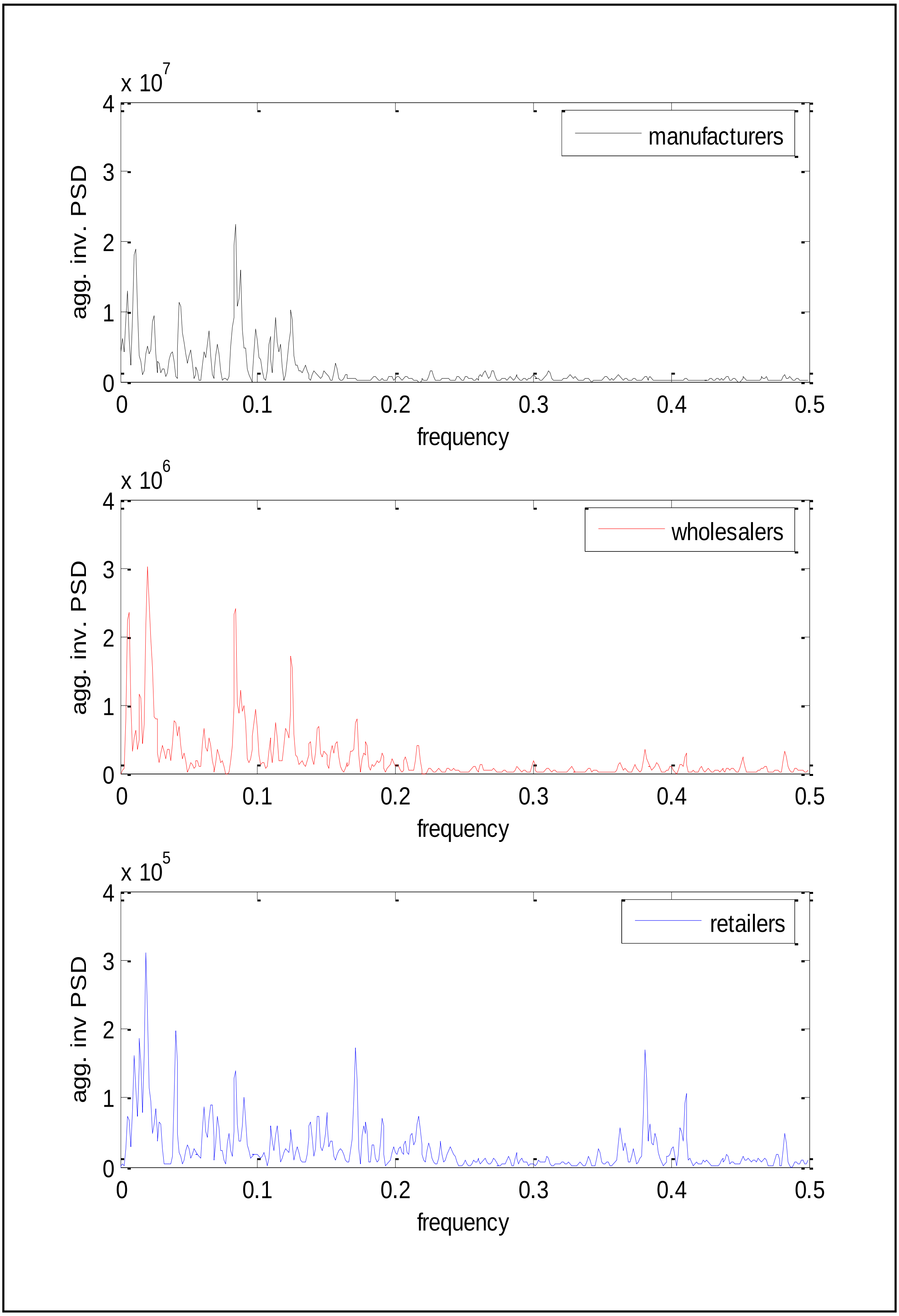


Figure 5.56. Spectral Power Density of Aggregate Inventories under Adaptive Pricing for (s,S) Policy (20 Rtl., 10 Whl., 2 Mnf.)

### 5.3.5. Aggregate Inventory and Order Dynamics under Adaptive Pricing for Anchor-And-Adjust Policy

The behavioral changes in aggregate orders can be discovered by comparing Figure 5.57 with Figure 5.35. The periodicity existing in aggregate orders gets blurred by the introduction of adaptive pricing. Aggregate order variability is reinforced intensely at all echelons. Minor decreases in aggregate order means are again observed. The comparison between Figure 5.58 and Figure 5.43 reveals the impact of adaptive pricing on aggregate inventory dynamics. The oscillatory pattern gets blurred, as in aggregate orders. Oscillation amplitude and variability of aggregate inventories are significantly amplified by adaptive pricing. Aggregate inventory averages are again increased. The comparison between Figure 5.59 and Figure 5.44 reveals that the spectral power is spread to higher number of frequencies, where the most dominant power falls into the low frequency range at retailer and wholesaler echelons. The intensification of the spectral power in the low frequency range is most apparent at the wholesaler echelon.

Table 5.35 demonstrates that aggregate order variability is amplified intensely at all echelons, as a consequence of adapting prices. Aggregate order mean is slightly decremented at the retailer echelon and remains about that level at wholesaler and manufacturer echelons. Table 5.36 demonstrates that aggregate inventory variability is intensely amplified at all echelons. Aggregate inventory averages are mildly increased by adaptive pricing.

To summarize the observations on the effect of adaptive prices under A&A policy, adaptive pricing introduces significant variability to both aggregate orders and inventories. However the oscillatory pattern gets blurred by the spreading of spectral power to higher number of frequencies. The intensification of the spectral power in the low frequency range is most apparent at the wholesaler echelon. Aggregate inventory mean again increases mildly at all echelons as a result of adaptive pricing.

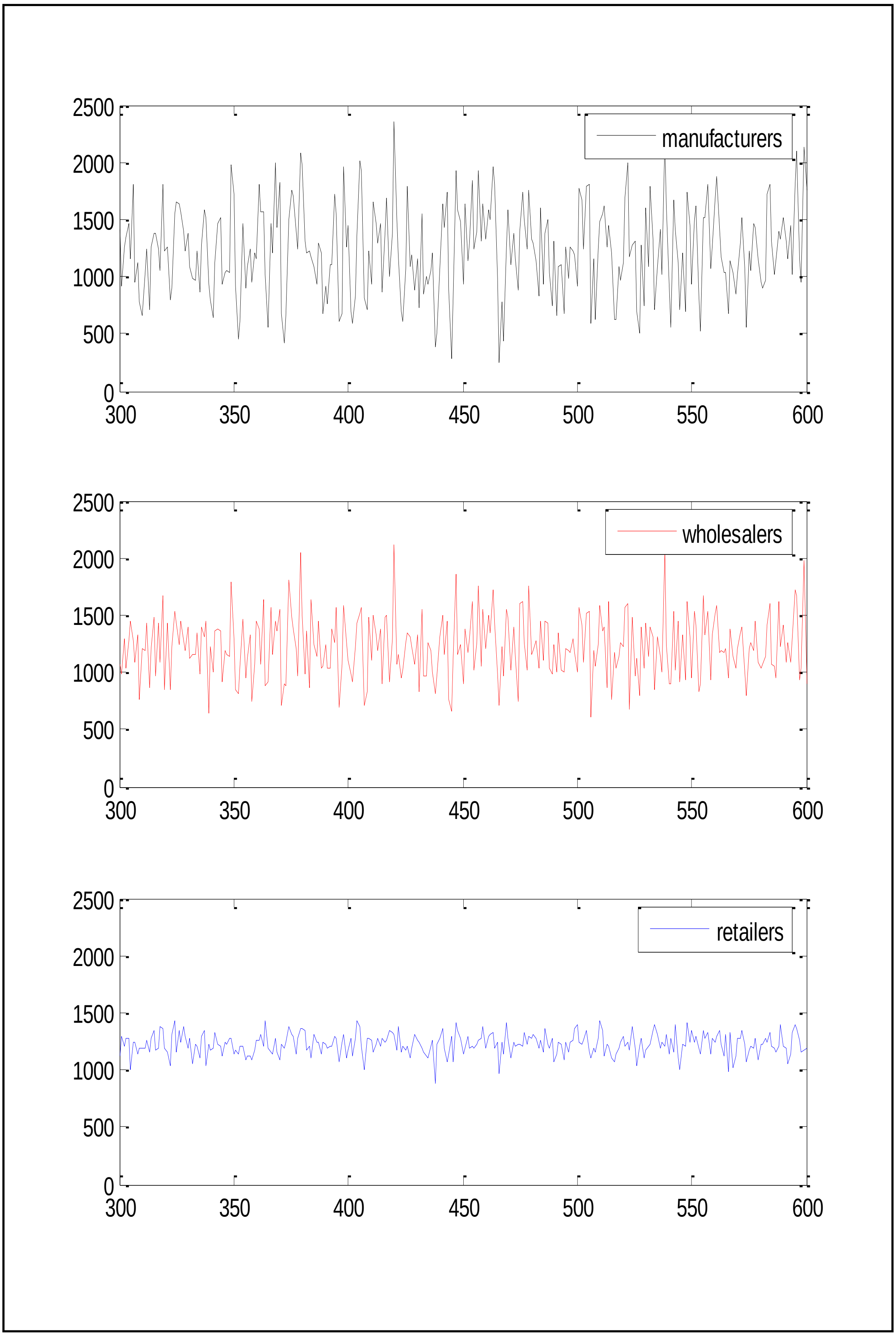


Figure 5.57. Aggregate Order Dynamics under Adaptive Pricing for A&A Policy (20 Rtl., 10 Whl., 2 Mnf.)

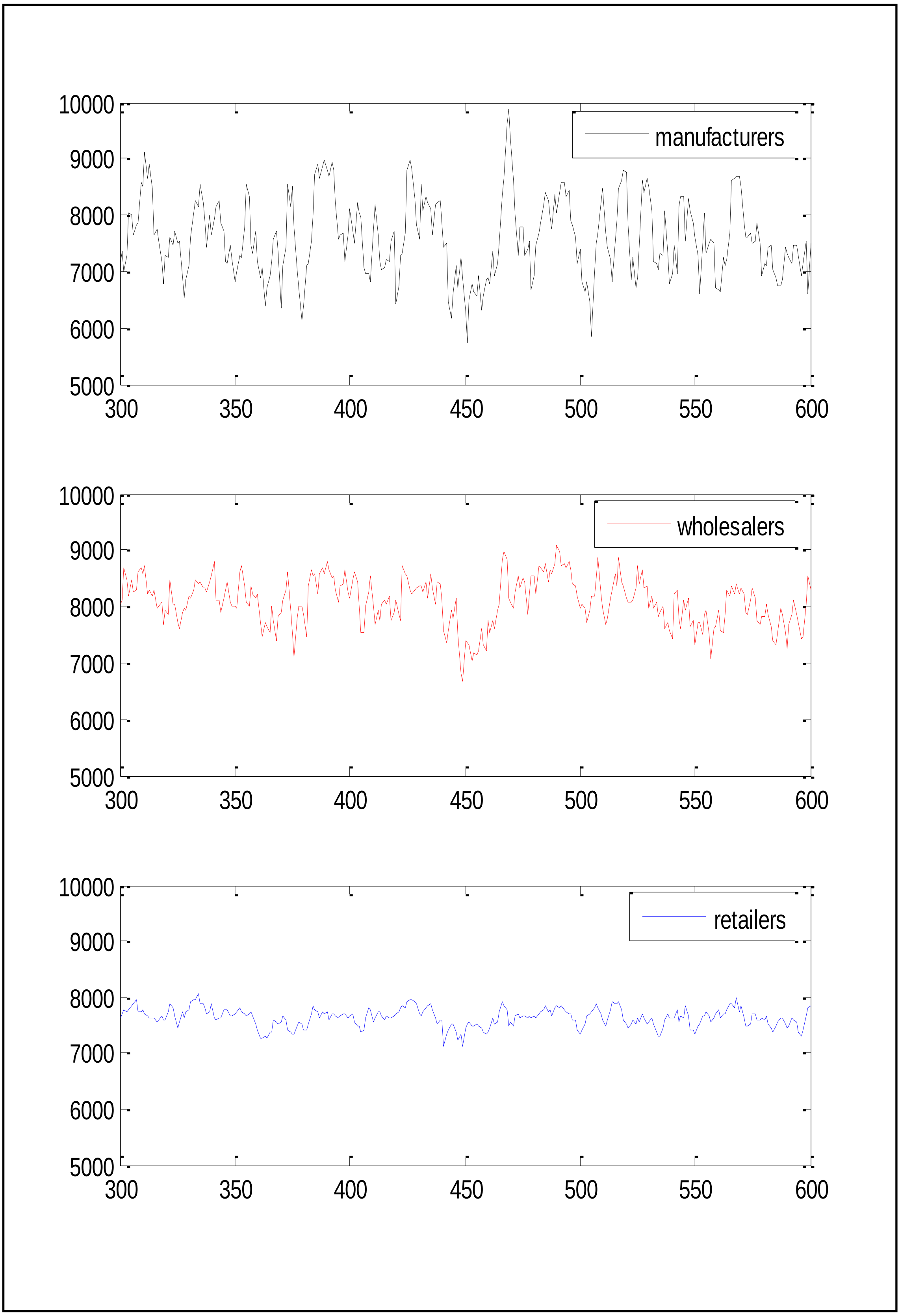


Figure 5.58. Aggregate Inventory Dynamics under Adaptive Pricing for A&A Policy (20 Rtl., 10 Whl., 2 Mnf.)

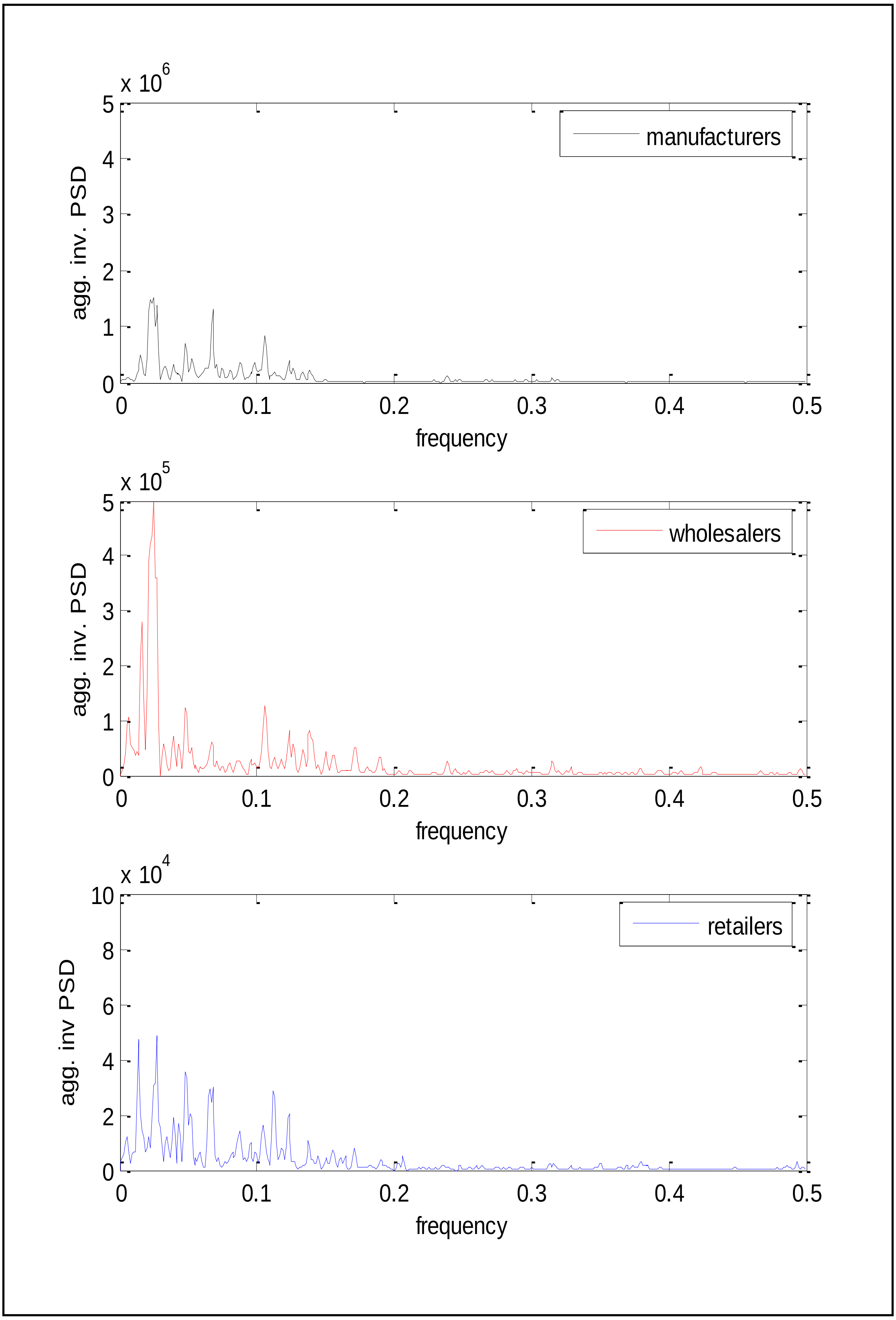


Figure 5.59. Spectral Power Density of Aggregate Inventories under Adaptive Pricing for A&A Policy (20 Rtl., 10 Whl., 2 Mnf.)

# 6. AGGREGATED VS. DISAGGREGATED MODELING

Determining the aggregation level is one of the most important decisions in the model building phase. Aggregation level mostly affects the representative capability of models. This issue brings forward important questions like: "How much does the level of aggregation affect the dynamics of the model? To what extent can an aggregated model represent the dynamics generated by a more detailed disaggregated model?" In aggregated models, it is assumed that the heterogeneity among individuals does not affect the system behavior significantly. On the other hand, the source of emergent behavior is the nonlinear interactions among individuals in disaggregated models. System dynamics is a methodology widely used to develop highly aggregated models. In this section, aggregated system dynamics models of the generic multiagent supply chain are developed and the capability of these models in representing aggregate order and inventory dynamics of the agent-based model (without modeling the individual agents and nonlinear interactions among them) is inquired.

In this section, a system dynamics model of the supply chain is developed. Then, the representative capability of the aggregated system dynamics model for each basic ordering policy is examined by comparing aggregate dynamics of the aggregated and disaggregated models. Following that, the representative capability of the aggregated system dynamics model in capturing aggregate dynamics caused by adaptive pricing is investigated.

## 6.1. Aggregated System Dynamics Model of the Generic Supply Chain

The material and information flow structures of the agent-based model are preserved in the system dynamics model. This is a three-echelon supply chain model where there is not any differentiation among agents. The structure is similar to that of agent-based model; but the observables of agents at each echelon are aggregated under single stocks, rates, and auxiliaries. Each supply chain echelon acts as if there is only one agent in that level. System dynamics model is equivalent to the agent-based model with single agent at each echelon under basic ordering policies. It is a discrete-time model with DT=1 for compatibility with the discrete-time agent-based model.

Figure 6.1 shows the stock-flow diagram for the retailer echelon of the system dynamics model, under (s,S) policy. The general structure is preserved in wholesaler and manufacturer sectors, where order decision at the lower echelon serves as the demand for the upper echelon and goods are dispatched to the lower echelon. The model structure remains the same except the ordering equations for the remaining basic ordering policies. The complete stock-flow diagrams and model equations for all basic ordering policies can be found in Appendix B.

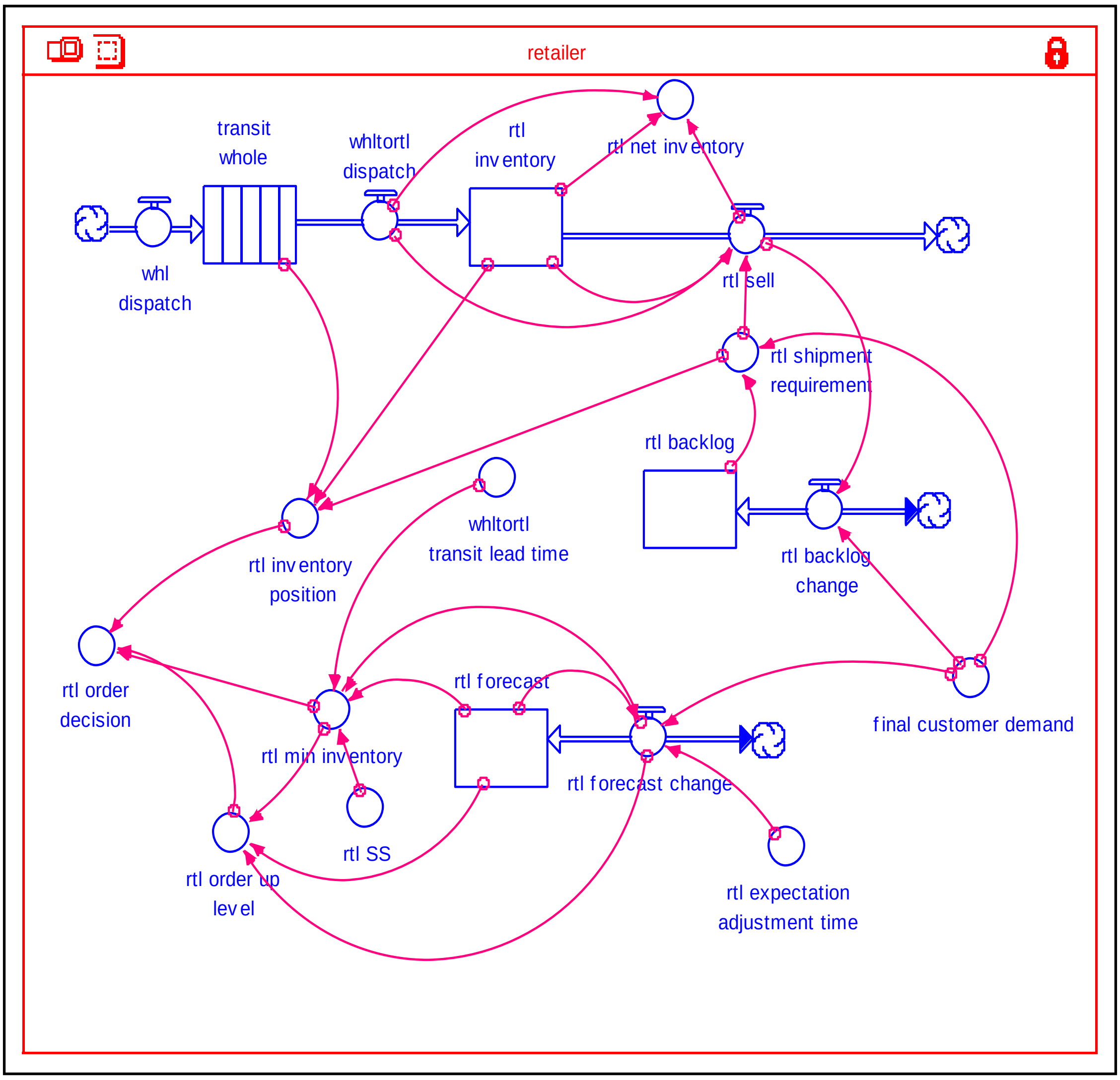


Figure 6.1. Stock-Flow Diagram of the Aggregated System Dynamics Model Retailer Echelon for (s,S) Policy

*Final customer demand* corresponds to the sum of individual final customer demands ($O_{FC_i}$) in the agent-based model. Since $O_{FC_i}$'s are iid Uniform(0,5) and there are 10000 final customers in the disaggregated models for Section 6; *final customer demand* can be modeled as a Normal random variable with mean 25000 and variance "250000/12".

*Rtl forecast* is the demand forecast of retailer echelon. It is formulated as an information delay structure, where the average delay time is *rtl expectation adjustment time*. The information delay structure is the same as exponential smoothing with parameter "1 / *rtl expectation adjustment time*". The equation for "rtl demand forecast" is as follows:

$$rtl\ forecast(t) = rtl\ forecast(t-1) + rtl\ forecast\ change(t) \tag{6.1}$$

where *rtl forecast change* is:

$$rtl\ forecast\ change = (\,final\ customer\ demand - rtl\ forecast)\,/\ rtl\ expectation\ adjustment\ time \tag{6.2}$$

*Rtl inventory* is increased by the arrival of products sent by the wholesaler echelon and decreased by the sales to final customers.

$$rtl\ inventory(\mathrm{t}) = rtl\ inventory(\mathrm{t\text{-}1}) + whltortl\ dispatch(t) - rtl\ sell(t) \tag{6.3}$$

Note that *rtl net inventory* is as in Equation 6.4 for compatibility with the iteration sequence of the agent-based model:

$$rtl\ net\ inventory(t) = rtl\ inventory(t)\ + whltortl\ dispatch(t) - rtl\ sell(t) \tag{6.4}$$

The materials sent by the wholesaler echelon (*whl dispatch*) pass through a discrete material delay structure. Therefore, materials received by the retailer echelon (*whltortl dispatch*) are discrete-delayed versions of *whl dispatch*.

$$whltortl\ dispatch(t) = whl\ dispatch(t - whltortl\ transit\ lead\ time) \tag{6.5}$$

*Rtl sell* is the minimum of possible shipment by the retailer echelon and the shipment requirement due to backlog and current demand:

$$rtl\ sell(t) = \min(rtl\ inventory(t)/1, rtl\ shipment\ requirement(t)) \tag{6.6}$$

where *rtl shipment requirement* is the sum of unmet demand from previous periods and the incoming demand of the current period:

$$rtl\ shipment\ requirement(t) = rtl\ backlog(t)\ + final\ customer\ demand(t) \tag{6.7}$$

The net change in *rtl backlog* is the discrepancy between *rtl shipment requirement* and *rtl sell*:

$$rtl\ backlog(t) = rtl\ backlog(t-1) + rtl\ backlog\ change(t) \tag{6.8}$$

The retailer echelon gives an order in an amount equal to the discrepancy between the *rtl order up to level* and *rtl inventory position*, whenever the inventory drops below the reorder point (*rtl min inventory)*.

$$rtl\ order\ decision(t) = \begin{cases} rtl\ order\ up\ to\ level(t) - rtl\ inventory\ position(t), \\ \qquad \text{if } rtl\ inventory\ position(t) < rtl\ min\ inventory(t) \\ 0, \\ \qquad \text{otherwise} \end{cases} \tag{6.9}$$

*Rtl inventory position* is the sum of net inventory at the end of period and the materials in *transit whole*:

$$\begin{aligned} rtl\ inventory\ position(t) = {} & rtl\ inventory(t) \\ & - rtl\ shipment\ requirement(t) + transit\ whole(t) \end{aligned} \tag{6.10}$$

*Rtl min inventory* is formulated similarly to the agent-based model. However, note that the *rtl forecast* at the end of period is used as the demand forecast for compatibility with the iteration sequence in the agent-based model.

$$\begin{aligned} rtl\ min\ inventory(t) = rtl\ SS\ &+ (whltortl\ transit\ lead\ time)* \\ &(rtl\ forecast(t) + DT * rtl\ forecast\ change(t)) \end{aligned} \tag{6.11}$$

*Rtl order up to level* is formulated in a similar way:

$$\begin{aligned} rtl\ order\ up\ to\ level(t) = rtl\ min\ inventory(t) &+ \\ 3*(rtl\ forecast(t) + DT * rtl\ forecast\ change(t)) \end{aligned} \tag{6.12}$$

The model structures at upper echelons and for other policies are similar and given in Appendix B.

## 6.2. Aggregated vs. Disaggregated Modeling under Basic Ordering Policies

The representative capability of the aggregated system dynamics model for each basic ordering policy is assessed here by comparing aggregate dynamics of the aggregated and disaggregated models. Note that the aggregated system dynamics model is completely equivalent to the agent-based model with single agent at each echelon under basic ordering policies. Therefore, the representative capability regards the effect of the number of agents on aggregate dynamics as explained in Section 5.2. The following set of number of agents is used throughout Section 6, which provides a reasonable supply chain topology and also guarantees that aggregate inventory mean does not deviate significantly from the mean in single agent case whenever pattern representation by aggregated model is possible. Number of final customers, retailers, wholesalers, and manufacturers are 10000, 100, 10, and two respectively.

### 6.2.1. Aggregated vs. Disaggregated Modeling under Fixed Order-Up-To-Level Policy

Figure 6.2 and Figure 6.3 compare aggregated and disaggregated models in terms of aggregate order and inventory dynamics respectively. Aggregated modeling proves to be no worse representative than disaggregated modeling if all agents use Fixed S policy; because aggregate order and inventory dynamics are independent of the number of agents and actual interaction patterns among agents. Aggregate order and inventory patterns are the same for the two models at all echelons. Table 6.1 reports statistics of aggregate echelon orders and inventories under aggregated and disaggregated models. The minor differences between aggregated and disaggregated model outputs and their statistics are caused by using different random numbers in final customer demand generation.

Table 6.1. Statistics of Aggregated vs. Disaggregated Model Comparison for Fixed S Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 44981 | 45020 | 249 | 248 | 25006 | 24993 | 144 | 144 |
| Whl. | 204981 | 205020 | 249 | 248 | 25006 | 24993 | 144 | 144 |
| Mnf. | 354967 | 355034 | 325 | 325 | 25006 | 24993 | 144 | 144 |

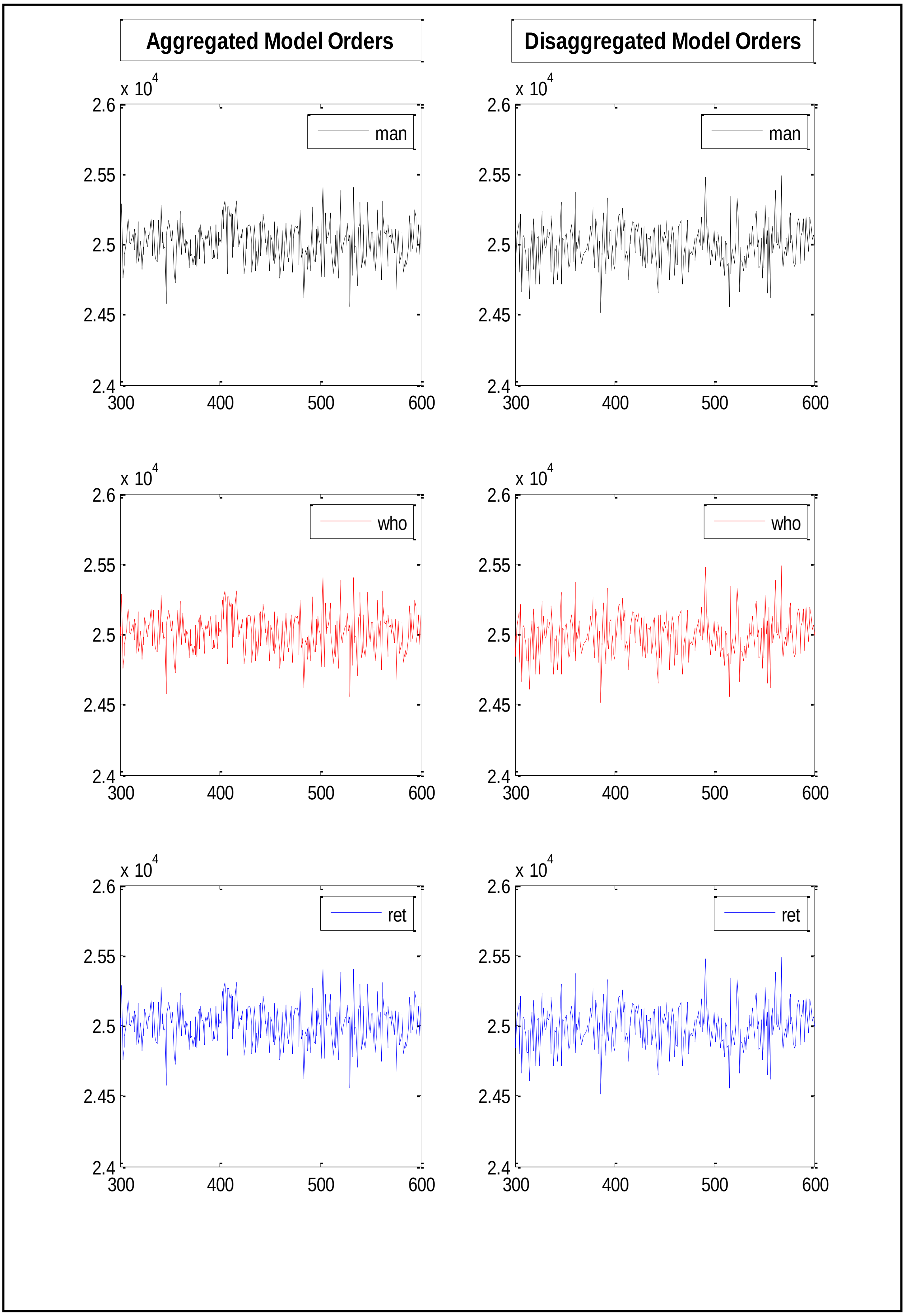


Figure 6.2. Aggregate Order Dynamics of Aggregated & Disaggregated Models for Fixed S Policy

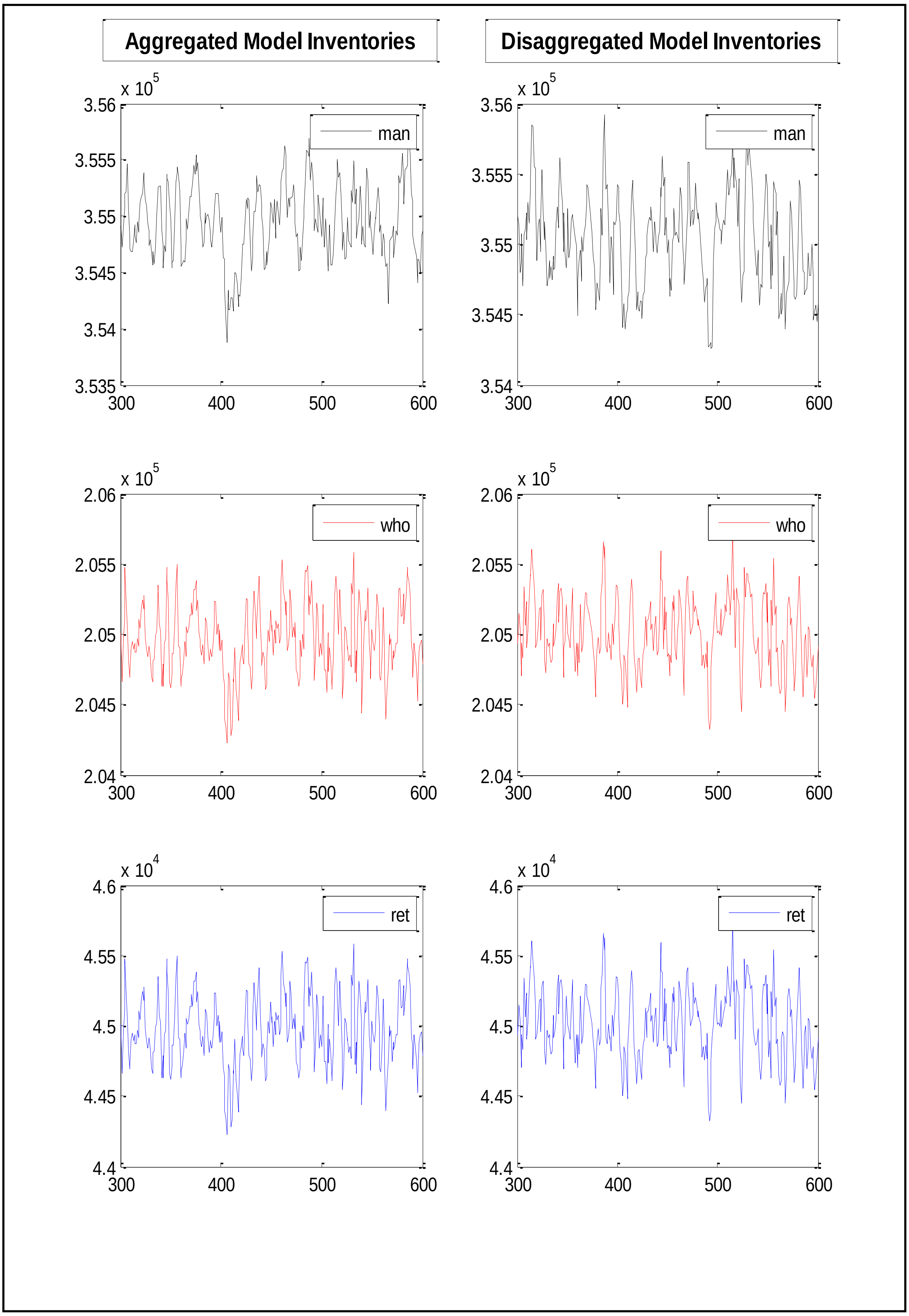


Figure 6.3. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models for Fixed S Policy

### 6.2.2. Aggregated vs. Disaggregated Modeling under Variable Order-Up-To-Level Policy

Figure 6.4 and Figure 6.5 compare aggregated and disaggregated models in terms of order and inventory dynamics respectively, when all agents use Variable Order-Up-To-Level policy. Aggregated modeling captures the general oscillatory patterns of aggregate orders and inventories. There is no significant difference between the behavioral patterns of aggregated and disaggregated models. Aggregate orders fluctuate without any significant periodicity. The autocorrelation and periodicity existing at aggregate inventories are well represented by the aggregated model.

Table 6.2. Statistics of Aggregated vs. Disaggregated Model Comparison for Variable S Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 114990 | 115021 | 408 | 420 | 24997 | 25006 | 360 | 386 |
| Whl. | 114990 | 116018 | 996 | 1413 | 24996 | 25005 | 946 | 1555 |
| Mnf. | 114988 | 120176 | 3121 | 8419 | 24990 | 25002 | 3032 | 8854 |

Table 6.2 reports statistics for aggregate echelon orders and inventories of aggregated and disaggregated models. It can be observed from Table 6.2 (and also from Figure 6.4 and Figure 6.5) that aggregated modeling captures aggregate order and inventory averages with only small errors. Order and inventory averages are slightly higher in the disaggregated agent-based model. On the other hand, aggregated modeling underestimates the variability of aggregate orders and inventories. The underlying reason is the dependency of aggregate order/inventory variability on the number of agents as in Figure 5.18 and Figure 5.22. Therefore, underestimation of variability by aggregated modeling increases as one moves up in the supply chain or the number of agents at all echelons are decreased. Aggregated modeling also underestimates the intensity of the bullwhip effect existing in the disaggregated model.

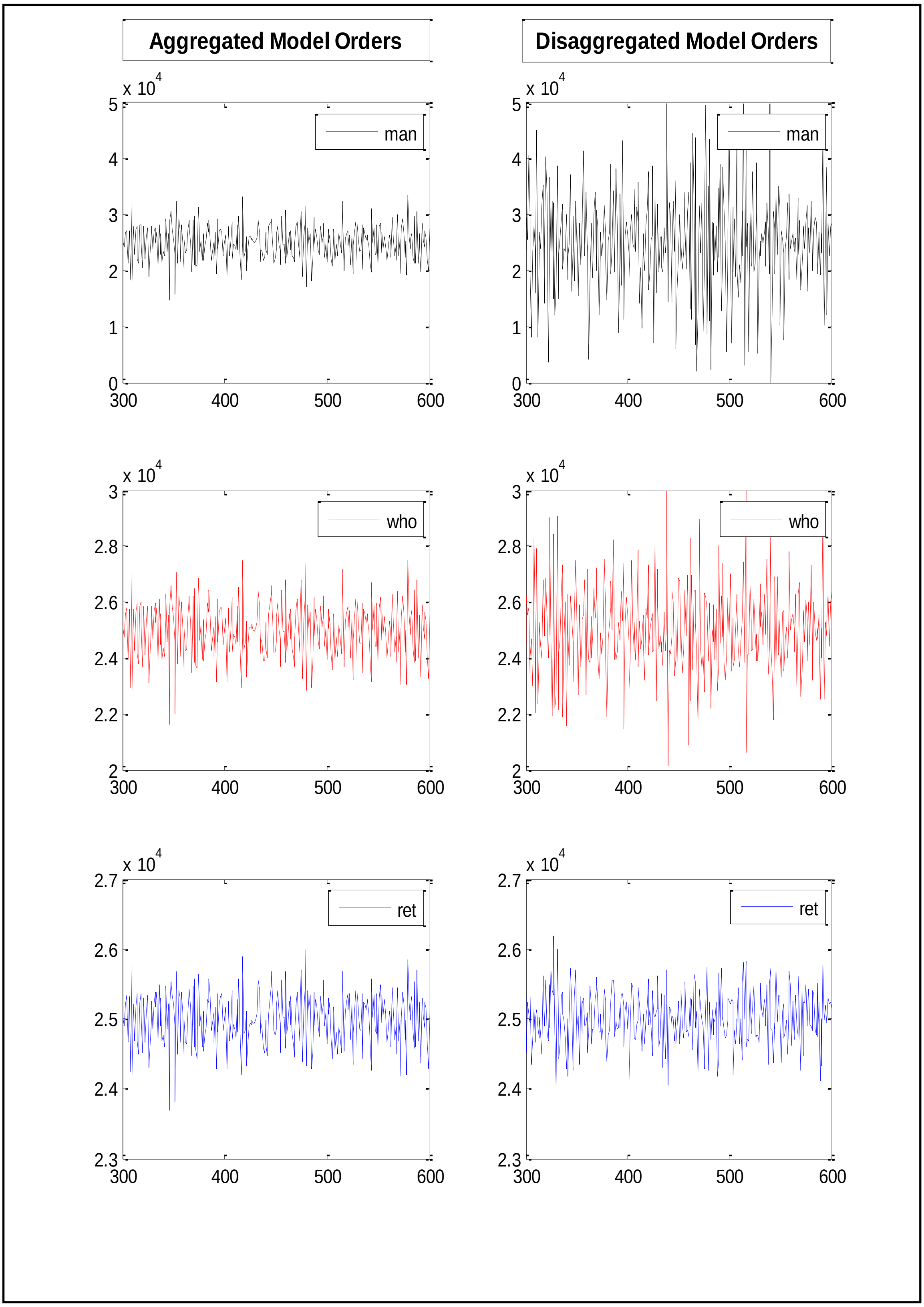


Figure 6.4. Aggregate Order Dynamics of Aggregated & Disaggregated Models for Variable S Policy

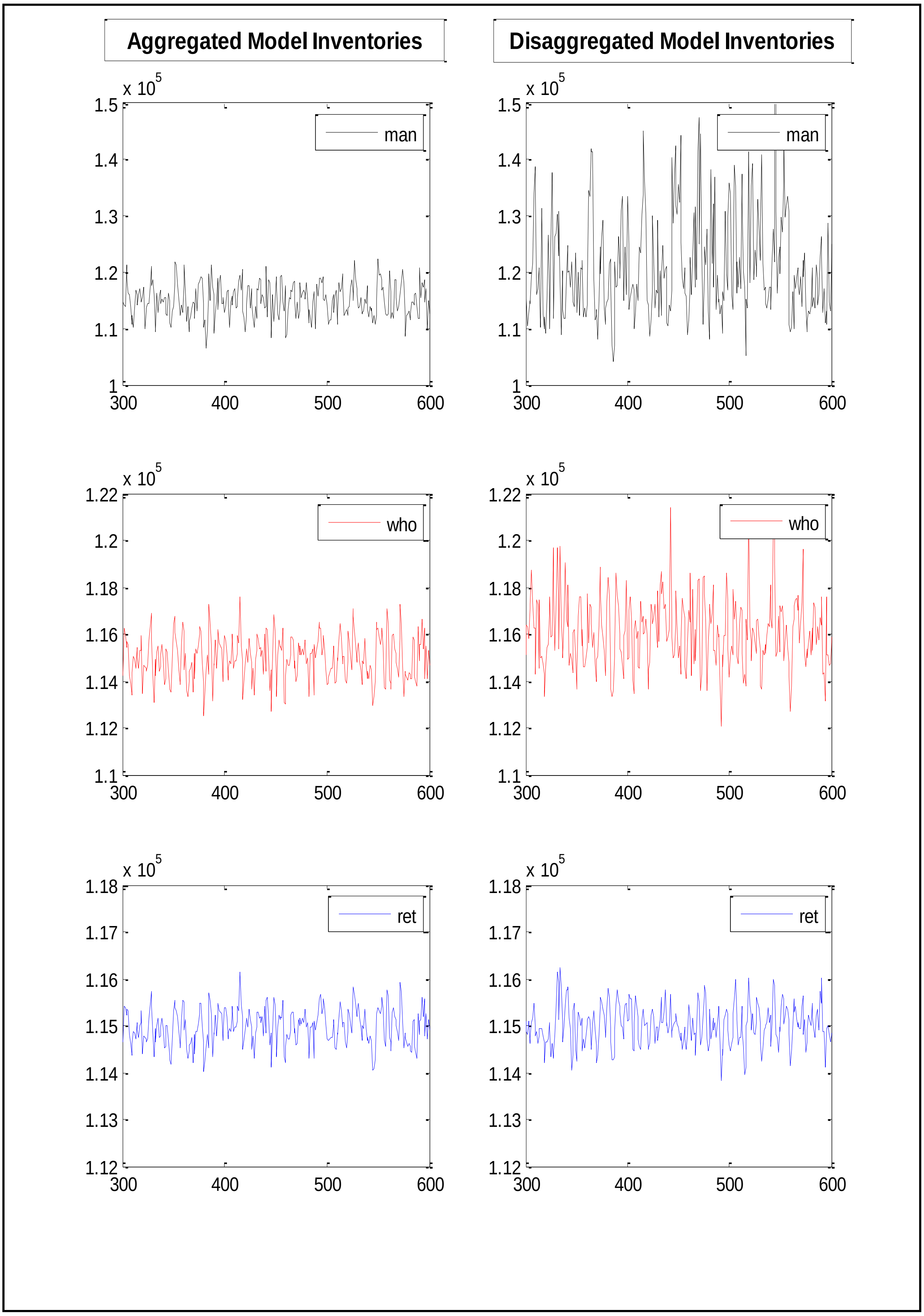


Figure 6.5. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models for Variable S Policy

### 6.2.3. Aggregated vs. Disaggregated Modeling under Reorder Point – Order-Up-To-Level Policy

Figure 6.6 and Figure 6.7 compare aggregated and disaggregated models in terms of order and inventory dynamics respectively, when all agents use Reorder Point - Order-Up-To-Level policy. Aggregated modeling is unable to capture the general patterns of both orders and inventories. There are significant differences between the behavioral patterns of aggregated and disaggregated models for (s,S) policy. The inadequacy of aggregated modeling under (s,S) policy results from the intermittent ordering attitude (as a consequence of order batching). The disaggregated modeling of (s,S) policy provides individual agents to give intermittent orders at different phase angles, which removes the discrete oscillatory nature of individual order/inventory dynamics at the aggregate scale. However, aggregated modeling dynamics has its root in intermittent ordering. On the other hand, aggregated modeling is still capable of capturing the bullwhip effect in both orders and inventories as a result of order batching.

Table 6.3. Statistics of Aggregated vs. Disaggregated Model Comparison
for (s,S) Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 83030 | 83806 | 25666 | 2750 | 25025 | 25003 | 40412 | 4249 |
| Whl. | 138655 | 95426 | 52852 | 11197 | 25013 | 25000 | 58459 | 14431 |
| Mnf. | 257665 | 119418 | 103121 | 39020 | 24948 | 25012 | 75893 | 38547 |

Table 6.3 reports statistics for aggregate echelon orders and inventories of aggregated and disaggregated models. There are significant errors in capturing inventory averages, which dramatically increase as one moves up in the supply chain –also resulting from the decreasing number of agents. On the other hand, aggregated modeling exaggerates the order/inventory variability due to not considering phase lags among individual agent dynamics. Aggregated modeling is still capable of capturing the bullwhip effect in both orders and inventories as a result of order batching.

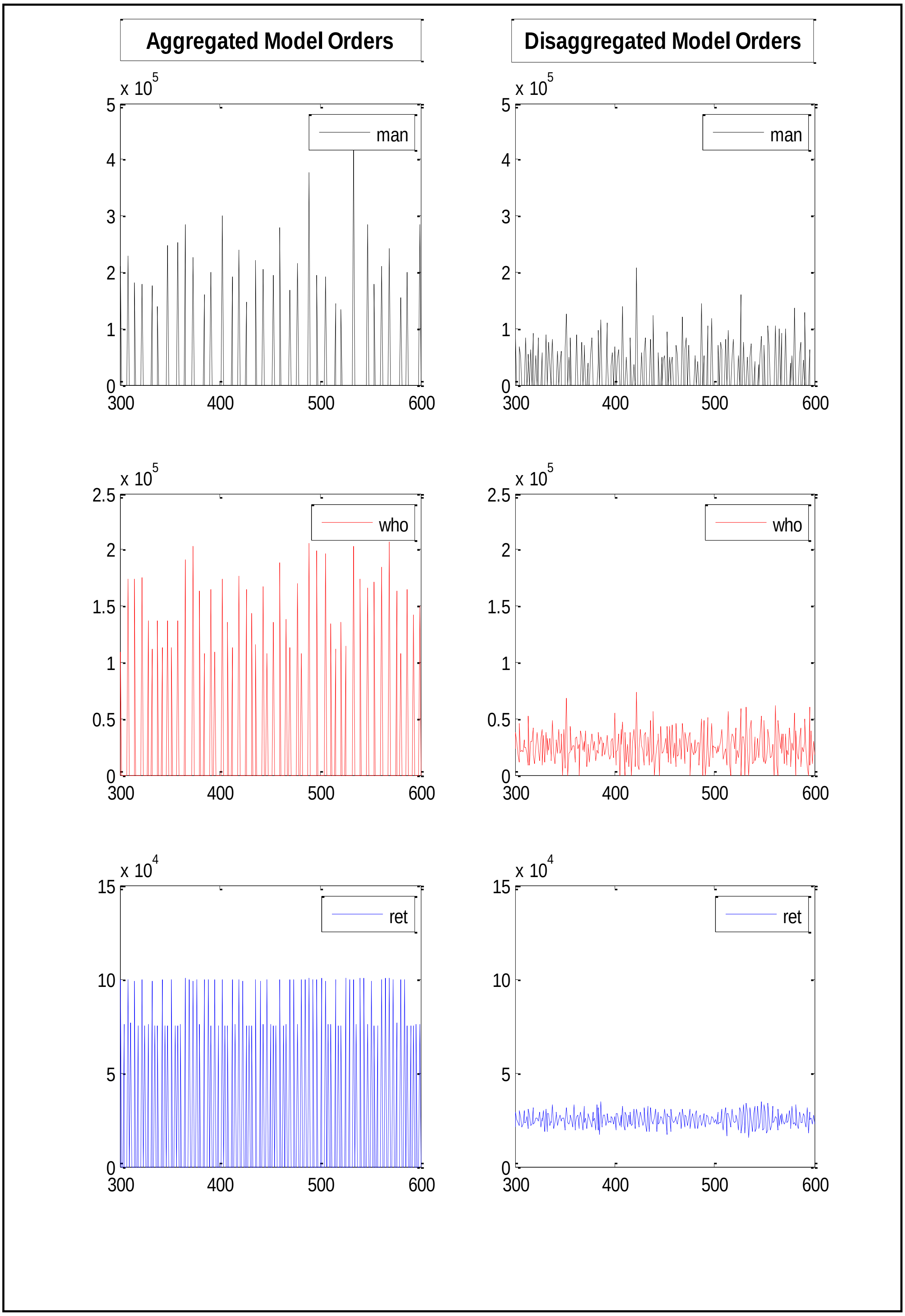


Figure 6.6. Aggregate Order Dynamics of Aggregated & Disaggregated Models for (s,S) Policy

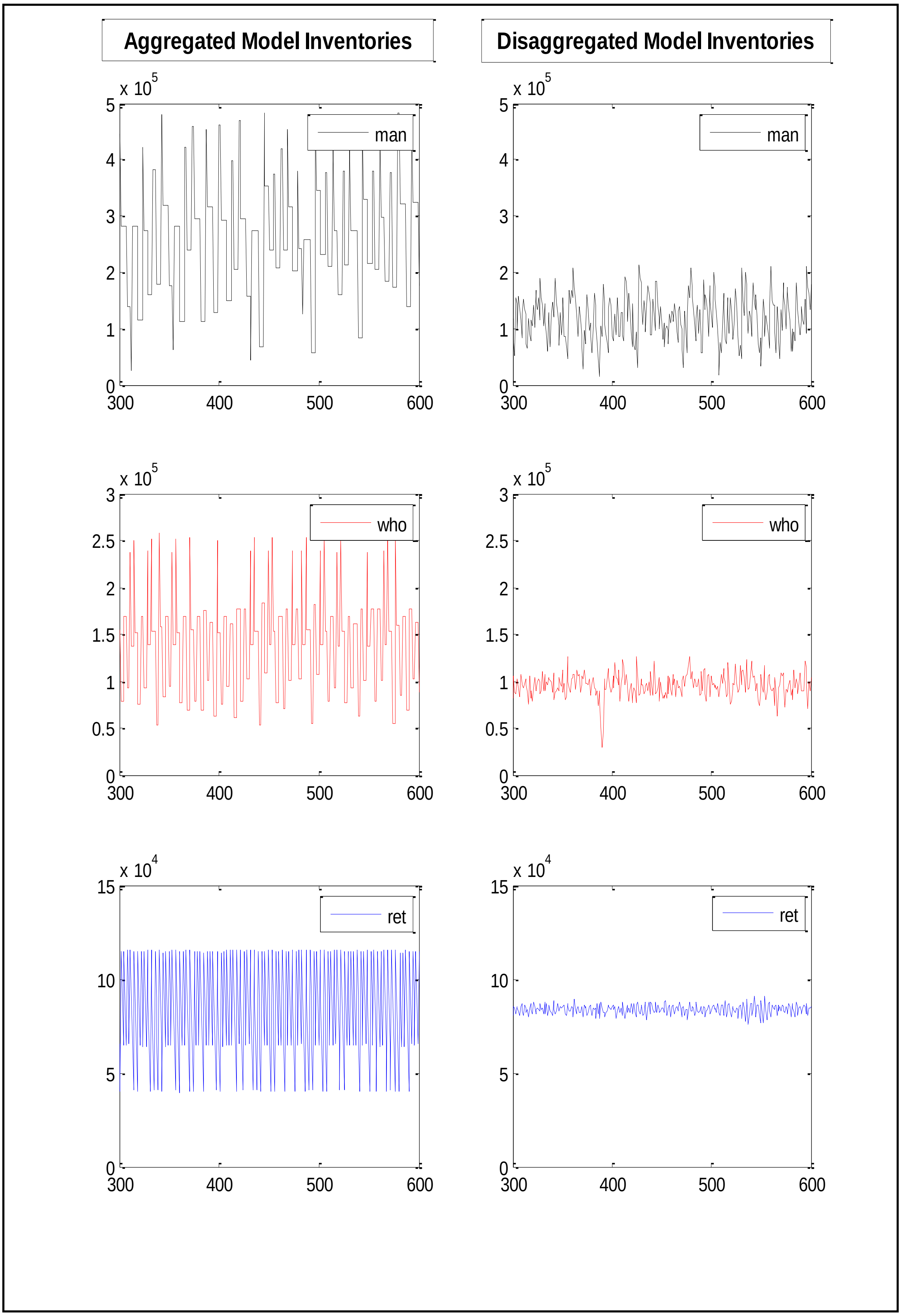


Figure 6.7. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models for (s,S) Policy

### 6.2.4. Aggregated vs. Disaggregated Modeling under Anchor-And-Adjust Policy

Figure 6.8 and Figure 6.9 compare aggregated and disaggregated models in terms of order and inventory dynamics respectively, when all agents use Anchor-And-Adjust ordering policy. Aggregated modeling captures the general oscillatory patterns of aggregate orders and inventories. The smooth oscillations -with long periods relative to other ordering policies- of both aggregate orders and inventories are well represented by the aggregated model. Aggregated modeling captures the autocorrelation and periodicity information sufficiently enough. Therefore, there is no significant difference between the behavioral patterns of aggregated and disaggregated models.

Table 6.4. Statistics of Aggregated vs. Disaggregated Model Comparison for A&A Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 150031 | 149947 | 438 | 490 | 25005 | 24992 | 188 | 215 |
| Whl. | 150038 | 150018 | 940 | 1001 | 25005 | 24992 | 369 | 420 |
| Mnf. | 150044 | 150702 | 2904 | 3339 | 25004 | 24993 | 935 | 1497 |

Table 6.4 reports statistics for aggregate echelon orders and inventories of aggregated and disaggregated models. It can be observed that aggregated modeling captures order and inventory averages with only small errors, as in Variable S policy. On the other hand, aggregated modeling underestimates the variability of aggregate orders and inventories. However, the proportional intensity of variability underestimation is significantly lower in A&A ordering policy with respect to the Variable S policy. These facts are caused by the smooth ordering attitude of Anchor-And-Adjust policy analyzed and explained in Section 5.2.4.

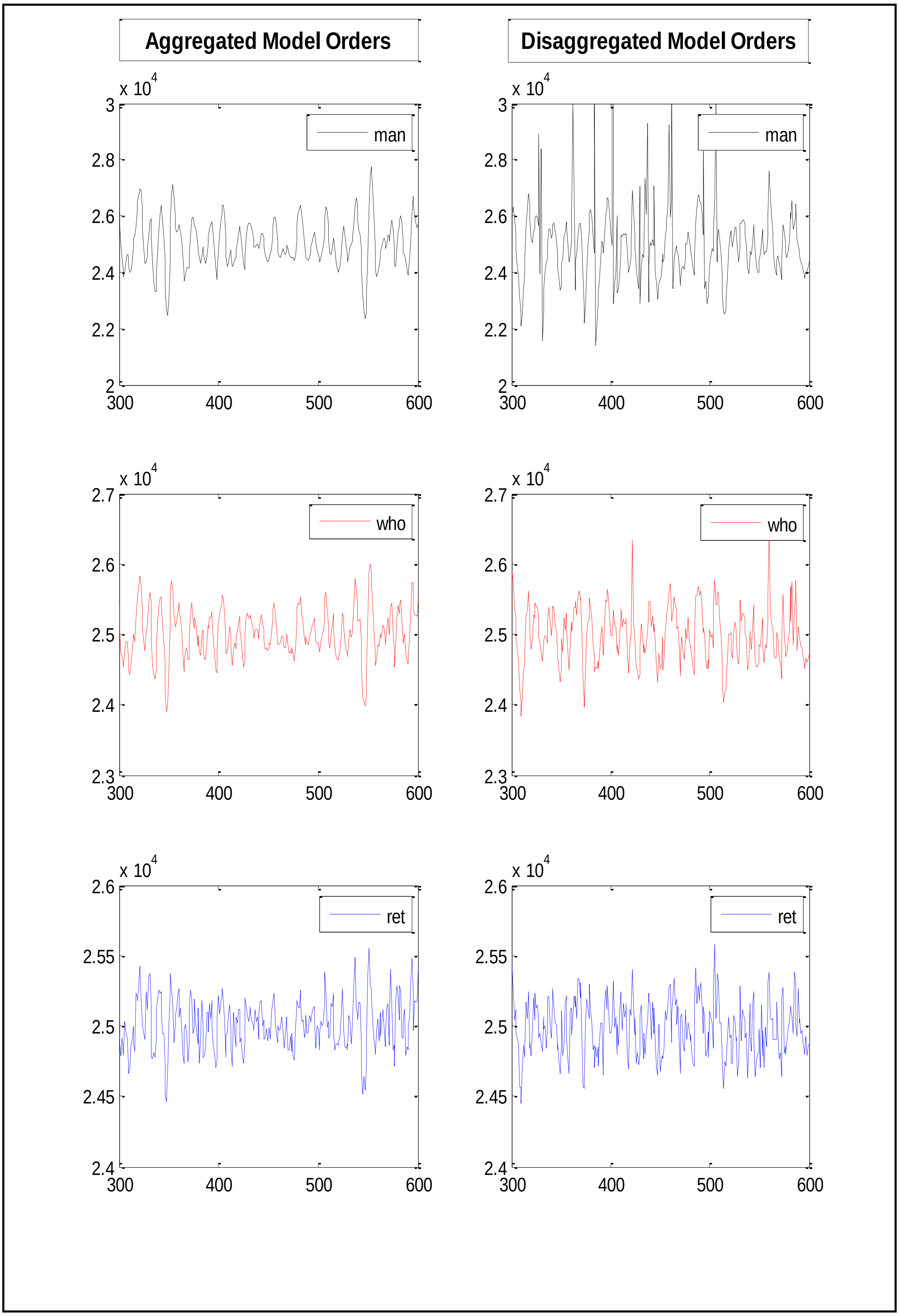


Figure 6.8. Aggregate Order Dynamics of Aggregated & Disaggregated Models for A&A Policy

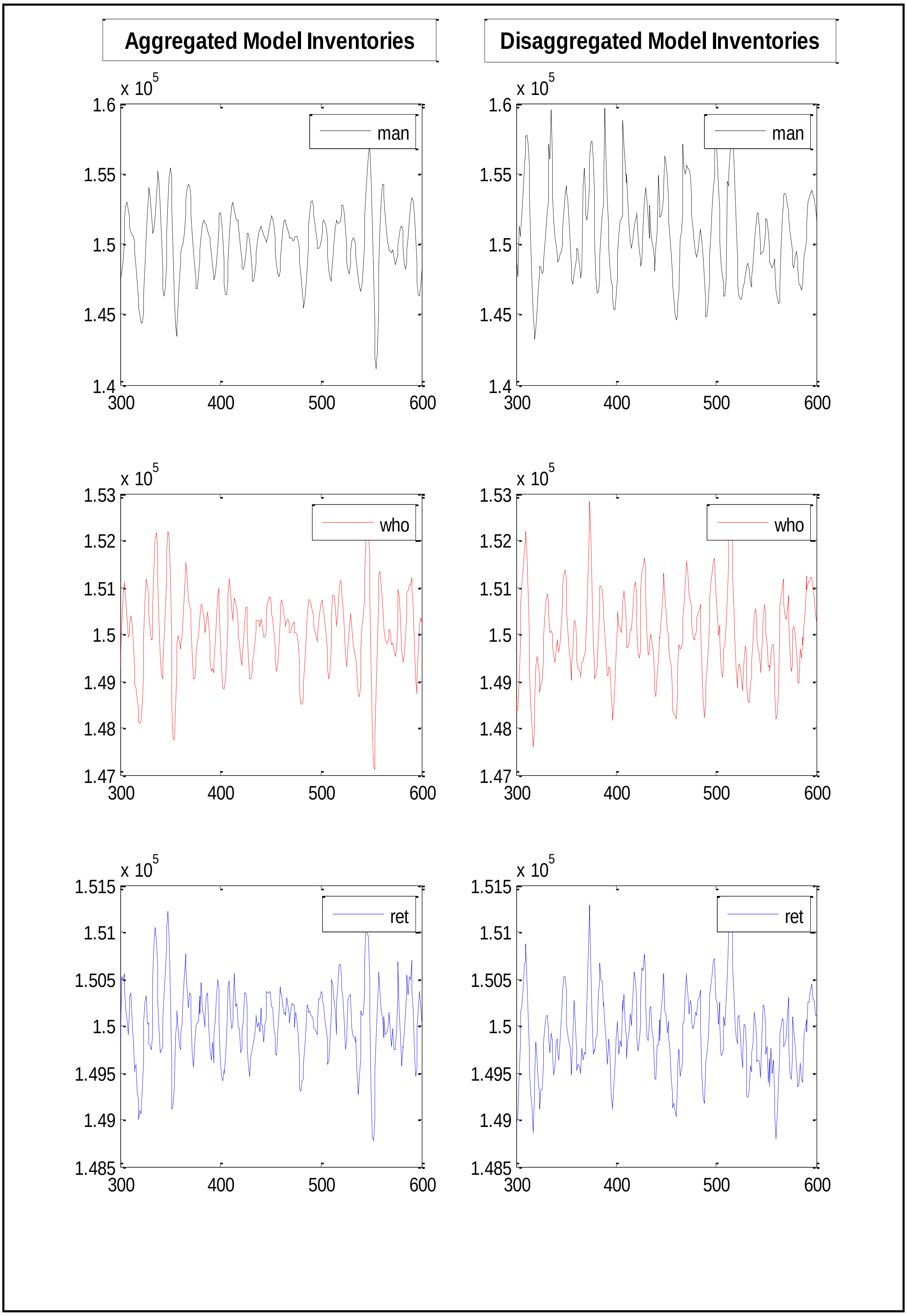


Figure 6.9. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models for A&A Policy

### 6.3. Aggregated vs. Disaggregated Modeling under Adaptive Pricing

The representative capability of the aggregated system dynamics model under adaptive pricing is assessed by comparing dynamics of the aggregated and disaggregated models for each basic ordering policy. Adaptive pricing formulation in Section 5.3 is formulated at an aggregate manner; however one of the two key elements in the adaptive pricing formulation ("supplier selection according to price" and "order modification according to supplier price") cannot be captured by this aggregated approach. Since aggregated SD model does not distinguish individuals, it is not possible to represent "supplier selection according to price" in this model and order modification is made according to echelon price –not individual supplier price.

Aggregate echelon prices are defined as functions of aggregate net stocks and Order-Up-To-Levels. Aggregate pricing function is the average of the two function forms in Figure 5.46, i.e. pricing coefficient at a specific "*net stock / order up to level*" ratio is the average of the coefficients given by the two distinct function forms in Figure 5.46. Each echelon adjusts its order amount according to the aggregate price of the upper echelon. The order adjustment is made by using the average of the two function forms in Figure 5.47.

Figure 6.10 shows the aggregated model structure at retailer echelon for (s,S) policy. The variables in red denote the elements included or modified for adaptive pricing mechanism. One should note that the links from *rtl forecast change* to *rtl order up to level* and *rtl min inventory* in Figure 6.1 are removed; since they create a circular connection problem due to the effect of inventory position on the demand –a dependency defined by the adaptive pricing mechanism. However, the removed links have only very small effects on aggregate order and inventory statistics. Hence, the effect of removal is negligible. The remaining variables are in one to one correspondence with the ones defined for the disaggregated mechanism, and the relationships are defined as in Section 5.3.1. *Rtl inv ratio* is the "*rtl net inventory / rtl order up to level*" ratio. *Rtl price coefficient* is the average of the two function forms in Figure 5.46. *Rtl inst price* holds the price value without considering the previous price value. *Rtl price delay* conveyor structure holds the price at the last step. Then, *rtl price* is the weighted average of *rtl last price* and *rtl inst price*. *Rtl price forecast* calculates the price expectation for the final customers. Note that *rtl price forecast* is updated only if the *final customer demand* is greater than 0, as in the disaggregated model. Then, the ratio of the *rtl price* to *rtl price forecast* is used to adjust the *normal final customer demand* through *final customer order coefficient*, which is the average of the two function forms in Figure 5.47. The structure at upper echelons is similar. Complete stock-flow diagrams and equations of the aggregated model under adaptive pricing are given in Appendix B.

### 6.3.1. Aggregated vs. Disaggregated Modeling under Adaptive Pricing for Fixed Order-Up-To-Level Policy

Figure 6.11 and Figure 6.12 compare aggregated and disaggregated models in terms of aggregate order and inventory dynamics respectively. Order and inventory means of the two models are close at all echelons. However, the variability at the aggregated model dynamics is significantly low. One can also observe that the pattern is smoother for the aggregated model –which can be easily seen from Figure 6.12.

Table 6.5. Statistics of Aggregated vs. Disaggregated Model Comparison under Adaptive Pricing for Fixed S Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 45038 | 45745 | 228 | 403 | 24993 | 24742 | 141 | 366 |
| Whl. | 205038 | 205711 | 225 | 467 | 24993 | 24741 | 141 | 462 |
| Mnf. | 355064 | 356293 | 273 | 519 | 24993 | 24741 | 141 | 462 |

Table 6.5 reports statistics of aggregate echelon orders and inventories for aggregated and disaggregated models. Table 6.5 shows that the variance is different for the two models. The comparison between Table 6.5 and Table 6.1 reveals that aggregated modeling causes a slight decrease in aggregate order and inventory variability, which is contrary to the disaggregated modeling findings.

The differences in the effectiveness of the pricing mechanism for the two models are illustrated by giving price dynamics in Figure 6.13 which shows the individual prices for the ABM model and the echelon prices for the aggregated SD model. Pricing mechanism acts through individual prices in the ABM model. Individual retailer prices vary according to individual inventory positions, and final customers select the retailers with relatively low prices at each time step and modify their orders according to the selected retailers' prices (the same mechanism applies for the higher echelons). This mechanism enables customers to adapt to the dynamic pricing environment through both selecting their suppliers and modifying their order amounts according to individual supplier prices. The adaptiveness of agents causes the amplification of the variability in the ABM model. However in the aggregated SD model, supplier selection mechanism cannot be represented. Aggregate orders are modified according to the echelon prices. The feedback structure between aggregate inventory positions and aggregate demand through the echelon prices result in a slight decrease in the variability. However, it has not been possible to detect how this feedback suppresses the order and inventory variability. On the other hand, it indicates that it is the combination of supplier selection and order modification according to the individual prices that amplifies the order and inventory variability in the ABM model. The impossibility of individual supplier selection and aggregation of individual prices turns out to form a significantly different price structure in the aggregated SD model.

Therefore, it can be concluded that aggregated modeling is incapable of capturing the variability amplification due to adaptive prices under Fixed S policy. However, it is important to note that this incapability is valid under the pricing formulation in this framework that controls prices according to inventory levels. Other pricing formulations not considered in the scope of this thesis may still be captured by aggregated models.

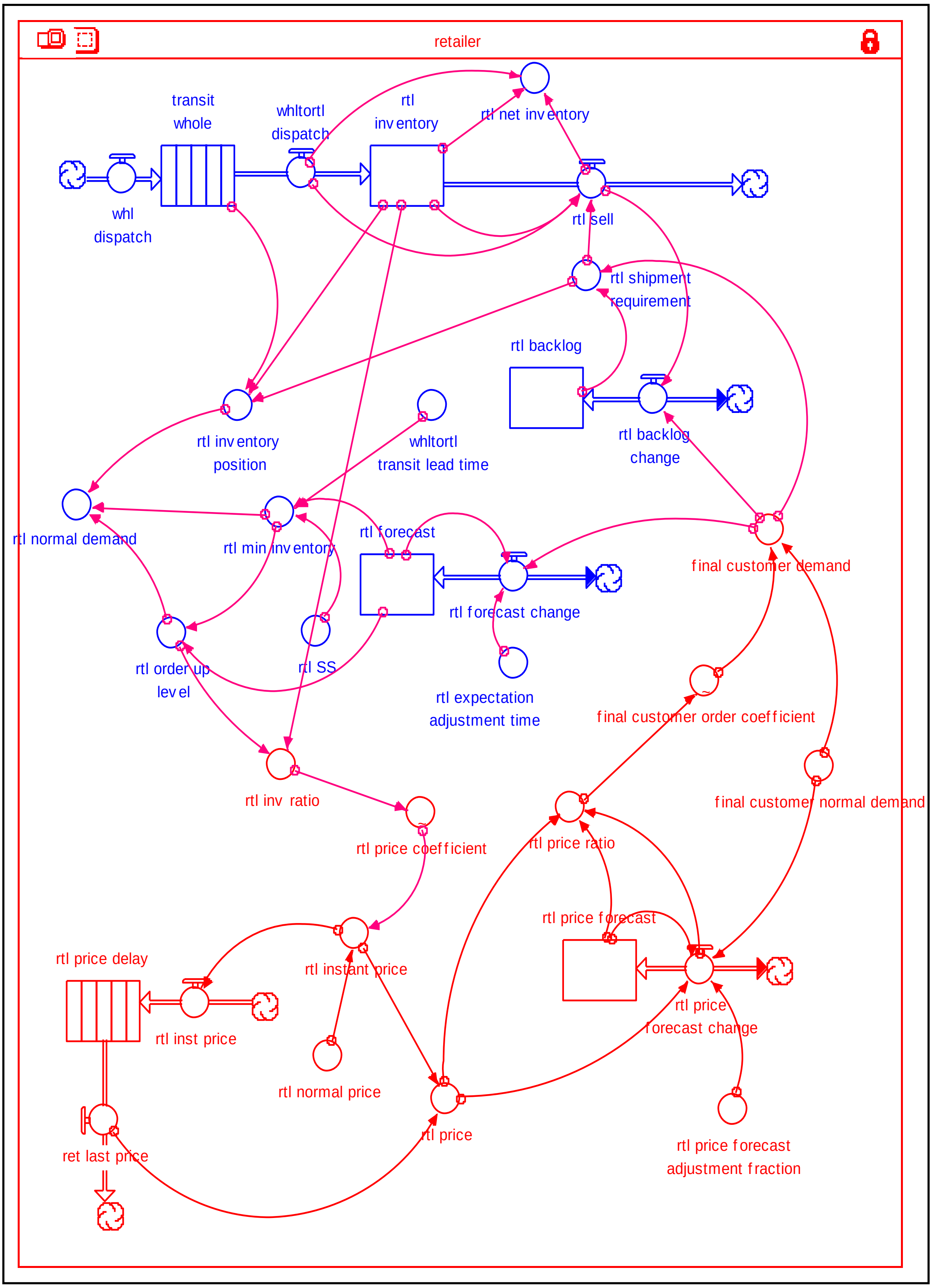


Figure 6.10. Stock-Flow Diagram of the Aggregated System Dynamics Model Retailer Echelon Under Adaptive Pricing for (s,S) Policy

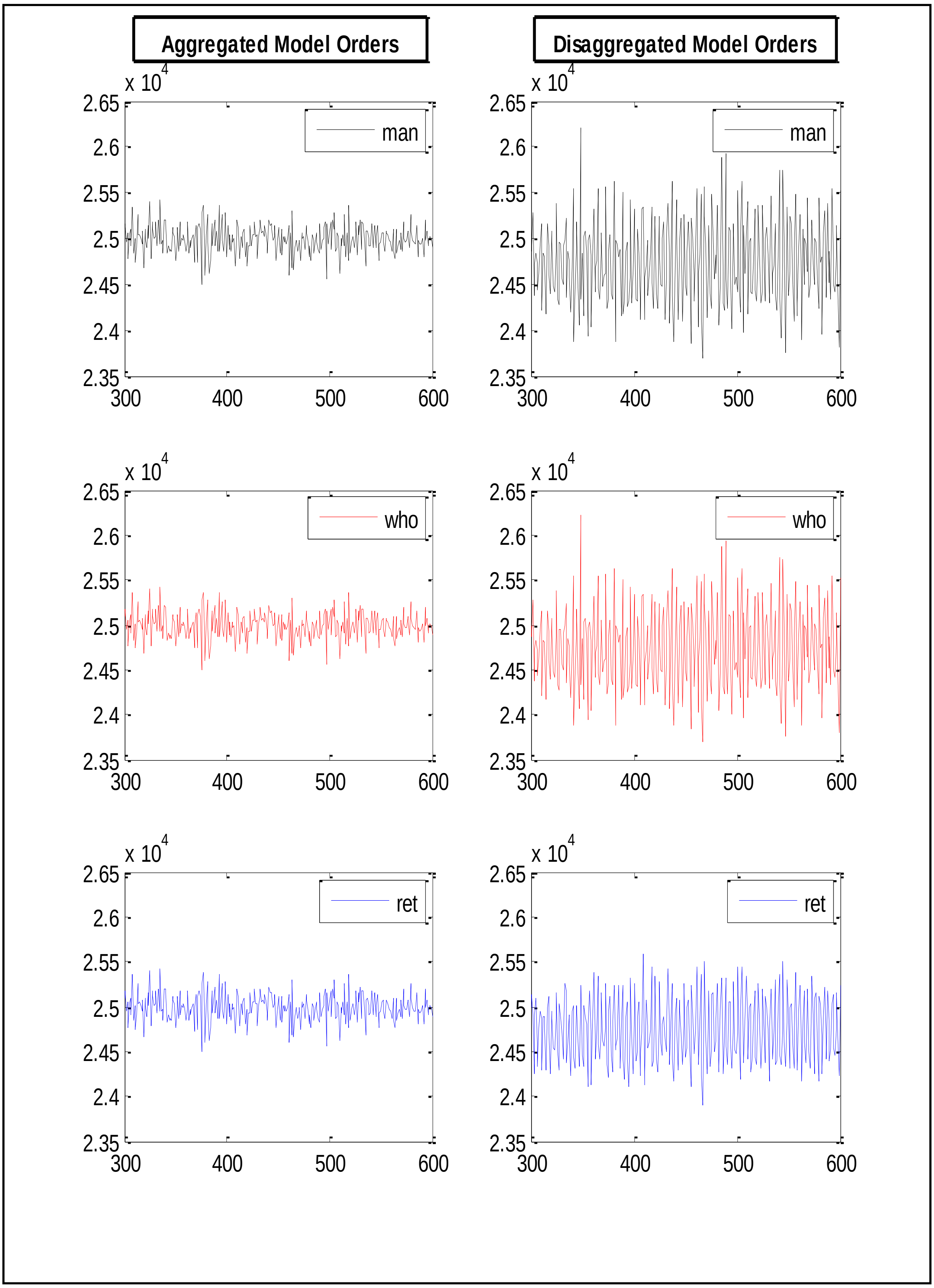


Figure 6.11. Aggregate Order Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for Fixed S Policy

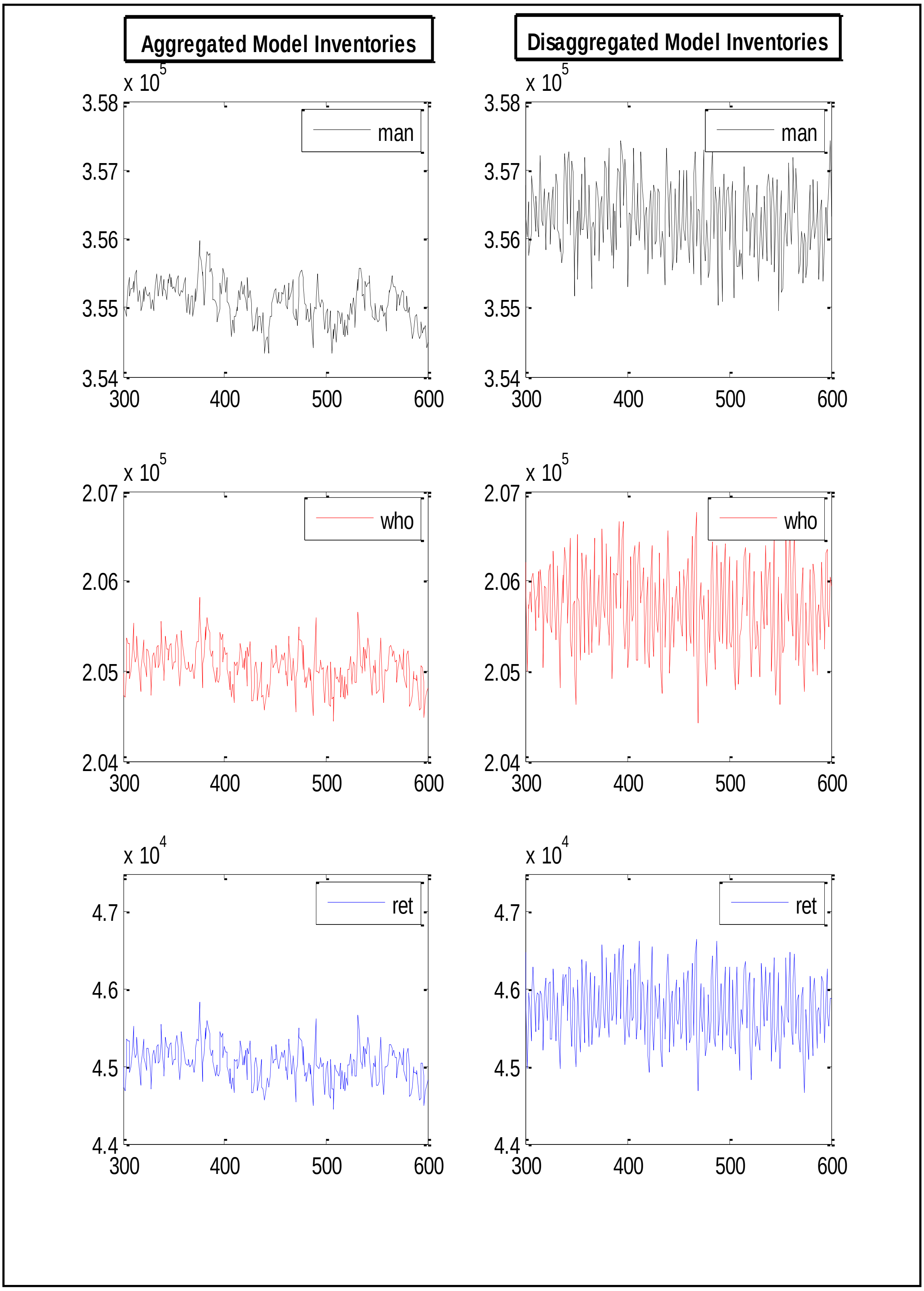


Figure 6.12. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for Fixed S Policy

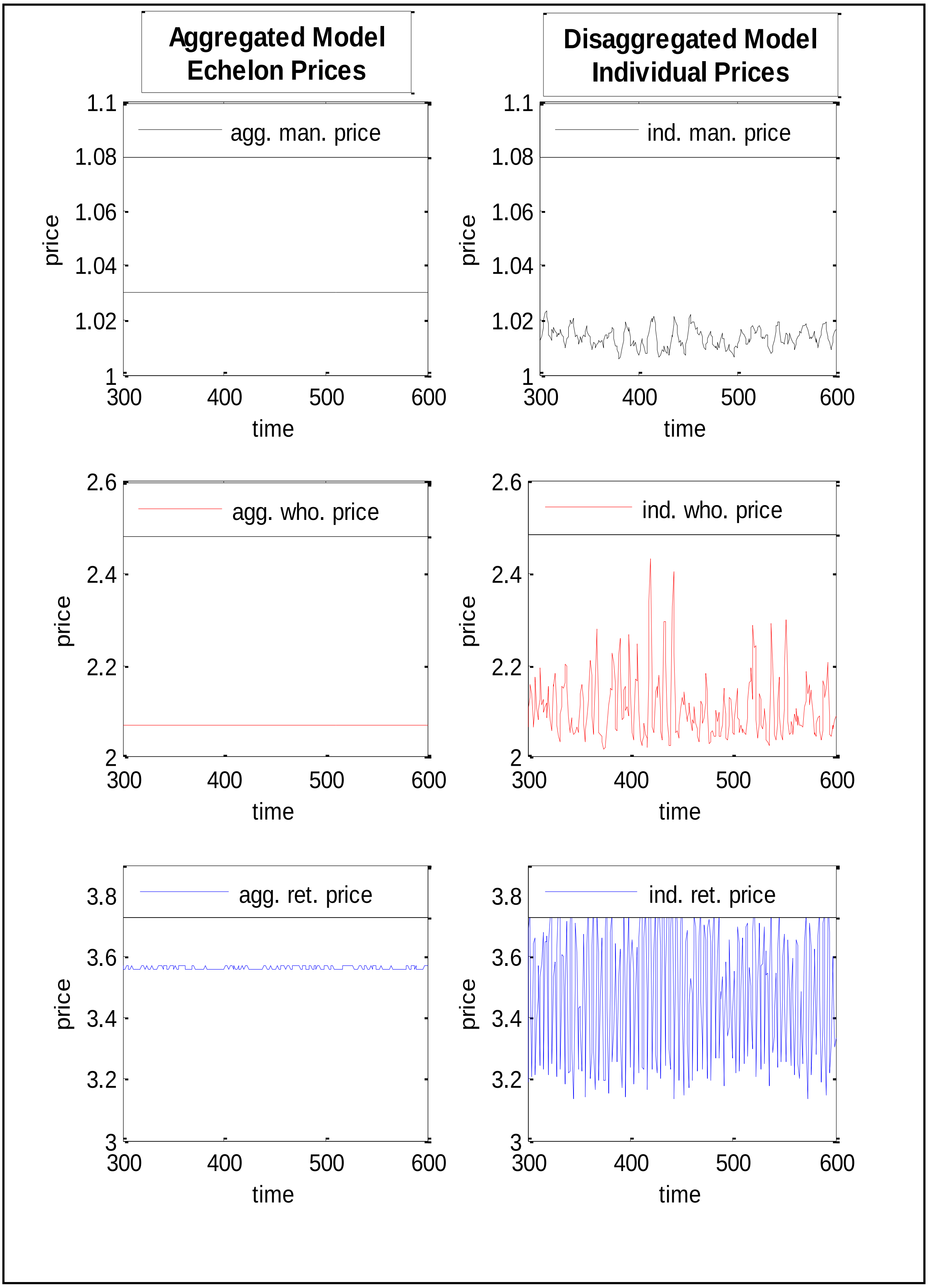


Figure 6.13. Echelon Prices of Aggregated Model & Individual Prices of Disaggregated Models for Fixed S Policy

### 6.3.2. Aggregated vs. Disaggregated Modeling under Adaptive Pricing for Variable Order-Up-To-Level Policy

Figure 6.14 and Figure 6.15 compare aggregated and disaggregated models in terms of order and inventory dynamics respectively, when all agents use Variable S policy. Table 6.6 reports statistics for aggregate orders and inventories of aggregated and disaggregated models. It can be observed from Table 6.6, Figure 6.14 and Figure 6.15 that aggregated modeling captures aggregate order and inventory averages with small errors, especially at retailer and wholesaler echelons. Orders and inventories fluctuate similarly in the two models; however the variability is significantly low in the aggregated model. Aggregated modeling also underestimates the bullwhip effect existing in the disaggregated model.

The comparison between Table 6.6 and Table 6.2 reveals that aggregated modeling causes a mild decrease in aggregate order and inventory variability, where disaggregated modeling causes a significant increase. The observations regarding the effects of pricing in aggregated and disaggregated models under Fixed S policy are valid under Variable S policy. It is the combination of supplier selection and order modification according to the price that amplifies the order and inventory variability in the ABM model. The impossibility of individual supplier selection and aggregation of individual prices results in a significantly different price structure in the aggregated SD model. Therefore, aggregated modeling is incapable of capturing the variability amplification due to adaptive prices, also under Variable S policy.

Table 6.6. Statistics of Aggregated vs. Disaggregated Model Comparison under Adaptive Pricing for Variable S Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 114990 | 115360 | 357 | 662 | 24997 | 24755 | 264 | 777 |
| Whl. | 114970 | 119320 | 714 | 3528 | 24996 | 24756 | 508 | 3908 |
| Mnf. | 115000 | 134460 | 1873 | 17872 | 24992 | 24731 | 1375 | 17065 |

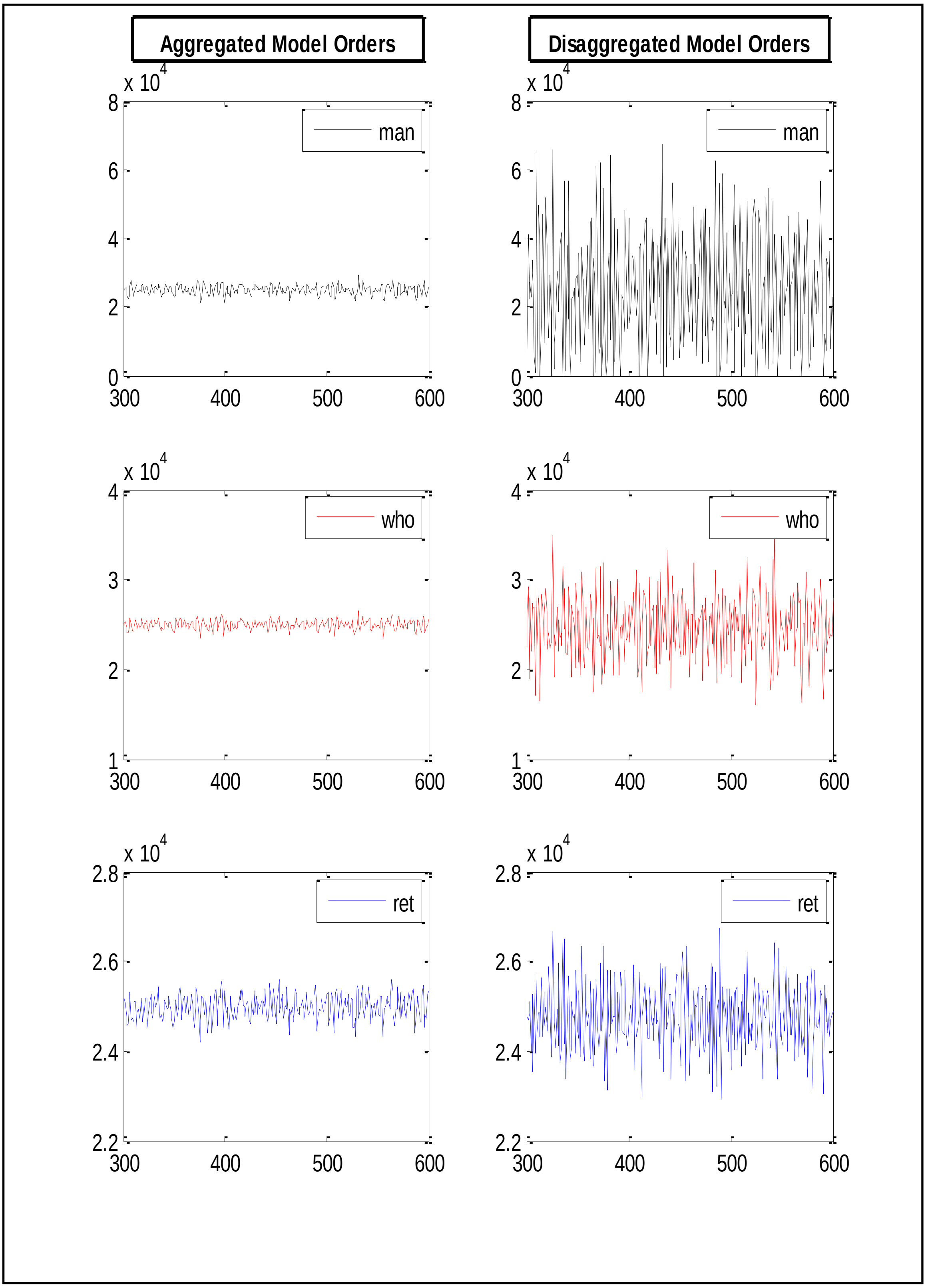


Figure 6.14. Aggregate Order Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for Variable S Policy

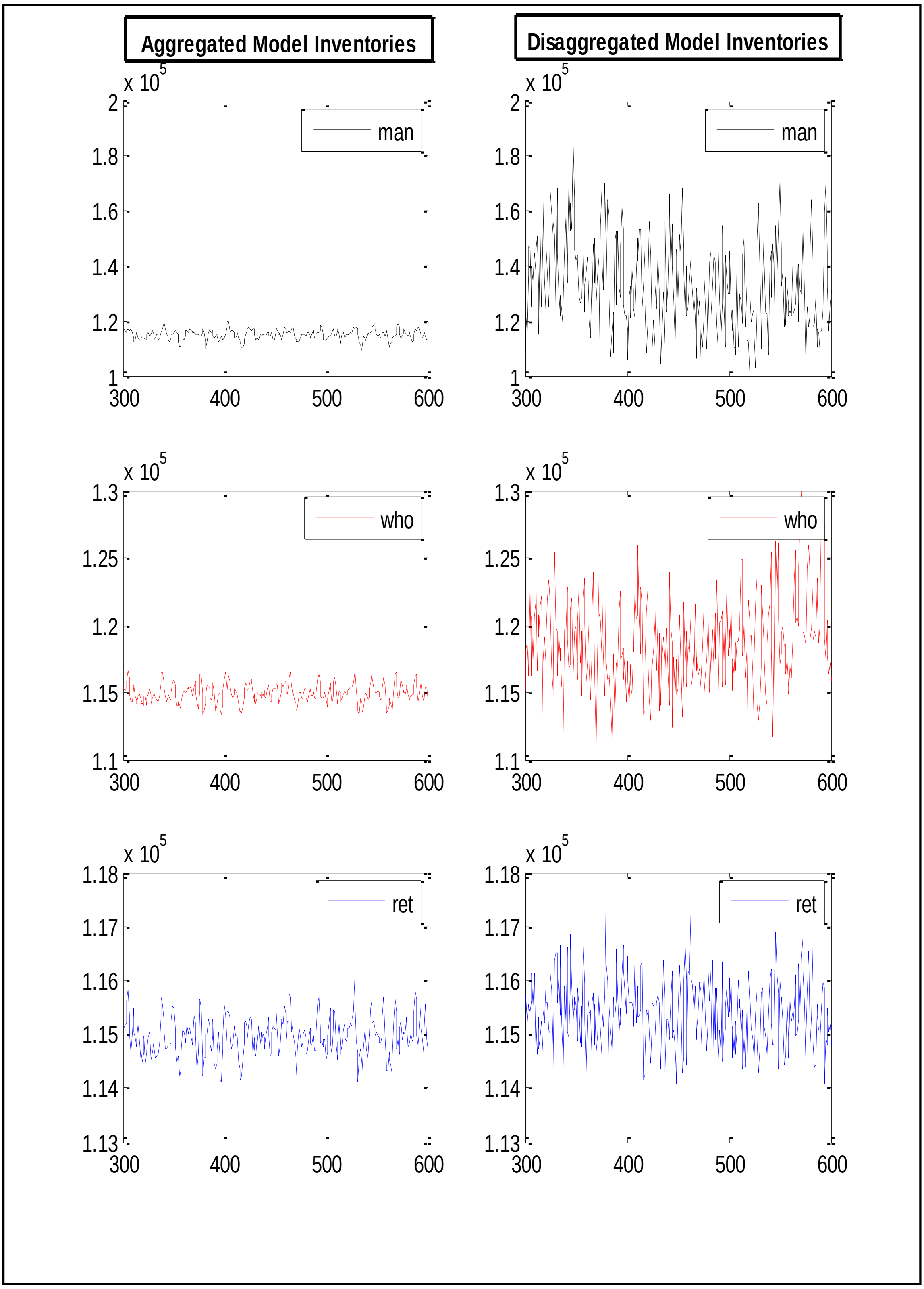


Figure 6.15. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for Variable S Policy

### 6.3.3. Aggregated vs. Disaggregated Modeling under Adaptive Pricing for Reorder Point – Order-Up-To-Level Policy

Figure 6.16 and Figure 6.17 compare aggregated and disaggregated models in terms of order and inventory dynamics respectively for (s,S) policy. It can be observed from Figure 6.16 and Figure 6.17 that aggregated modeling cannot capture the general behavioral patterns of the disaggregated model. It results from the incapability of aggregated modeling under order batching –due to fundamental pattern differences between individual and aggregate dynamics. The pricing mechanism does not alter intermittent ordering pattern; therefore aggregated SD model still preserves the intermittent pattern and cannot capture the aggregate pattern of the ABM model.

Furthermore, the comparison between Table 6.7 and Table 6.3 reveals that aggregated modeling causes a decline in aggregate order and inventory variability. Order batching causes highly-varying inventory dynamics in the SD model which yields aggregate echelon prices as in Figure 6.18. The variability of aggregate prices in the SD model is comparable with the variability of individual prices in the ABM model (it is even higher in the manufacturer echelon). However, pricing structure still yields reduction in aggregate variability for the SD model. On the other hand, adaptive pricing causes an increase in aggregate inventory variability under the disaggregated ABM model. The impossibility of individual supplier selection and aggregation of individual prices turns out to form a significantly different price structure in the SD model. It can be concluded that aggregated modeling is incapable of capturing the variability amplification effect of adaptive prices on aggregate inventory dynamics under (s,S) policy.

Table 6.7. Statistics of Aggregated vs. Disaggregated Model Comparison under Adaptive Pricing for (s,S) Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 81420 | 97096 | 23823 | 3839 | 24678 | 24570 | 38318 | 4178 |
| Whl. | 116168 | 116760 | 43636 | 14751 | 24774 | 24558 | 48106 | 14465 |
| Mnf. | 173364 | 141350 | 95129 | 50666 | 24806 | 24564 | 68011 | 39384 |

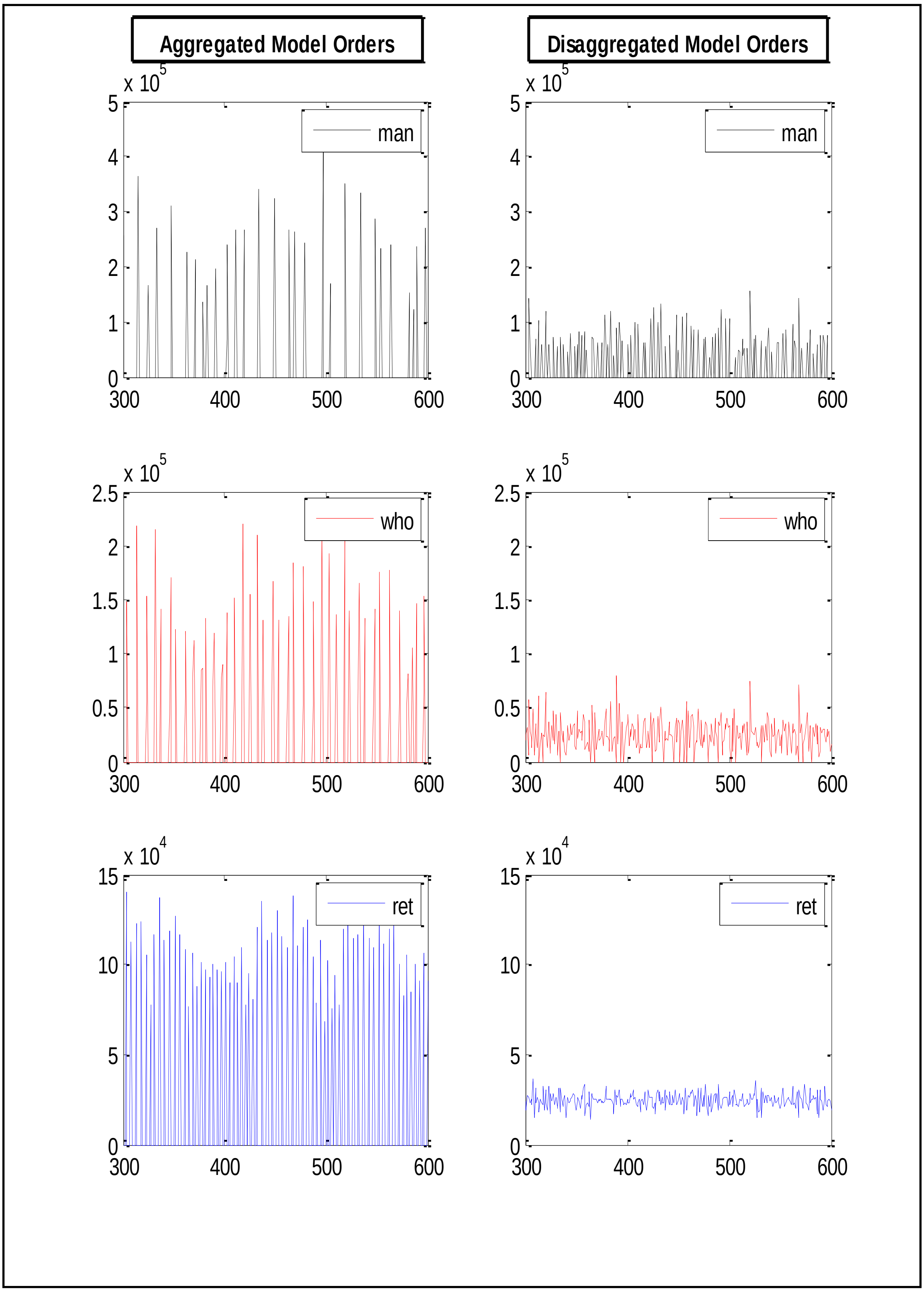


Figure 6.16. Aggregate Order Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for (s,S) Policy

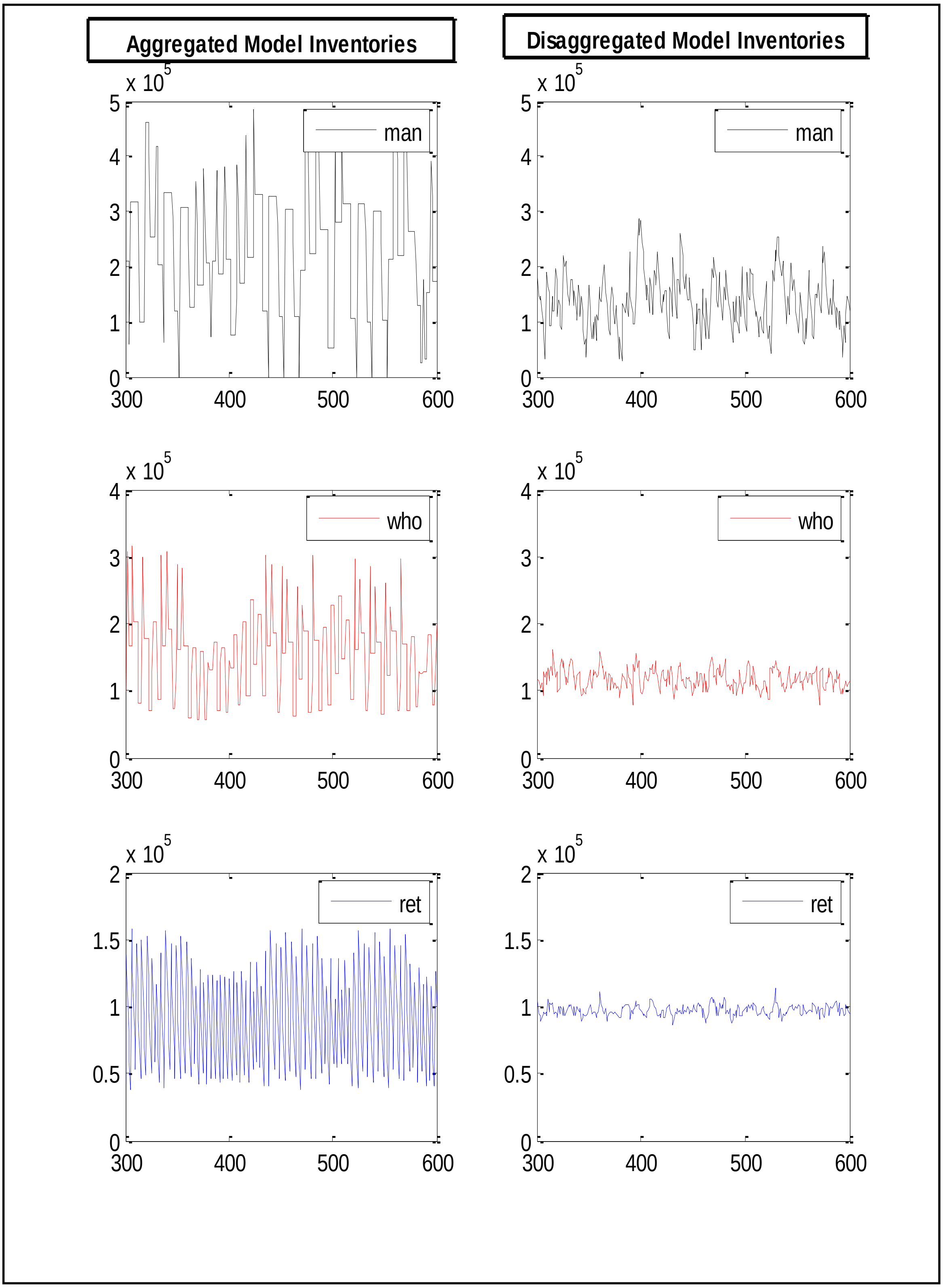


Figure 6.17. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for (s,S) Policy

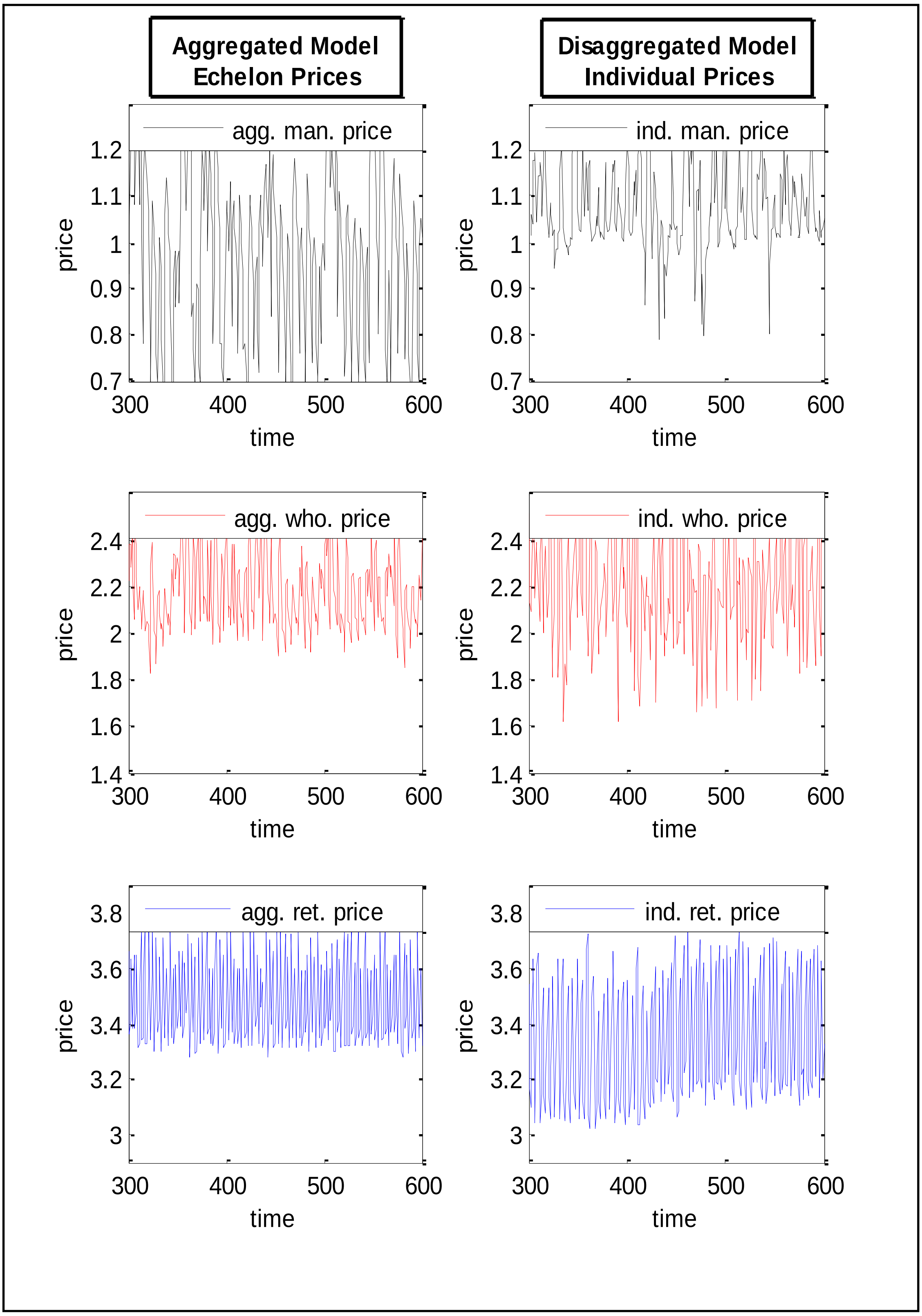


Figure 6.18. Echelon Prices of Aggregated Model & Individual Prices of Disaggregated Models for (s,S) Policy

### 6.3.4. Aggregated vs. Disaggregated Modeling under Adaptive Pricing for Anchor-And-Adjust Policy

Figure 6.19 and Figure 6.20 compare aggregated and disaggregated models in terms of order and inventory dynamics respectively, when all agents use A&A policy. Table 6.8 reports statistics for aggregate echelon orders and inventories of aggregated and disaggregated models. It can be observed from Table 6.8, Figure 6.19 and Figure 6.20 that aggregated modeling captures aggregate order and inventory averages with small errors. However, the variability is significantly low in the aggregated model.

The comparison between Table 6.8 and Table 6.4 reveals that aggregated modeling causes a decline in aggregate order and inventory variability, where disaggregated modeling causes a significant increase. The observations regarding the effects of pricing in aggregated and disaggregated models are valid under Anchor-And-Adjust policy. It is the combination of supplier selection and order modification according to the price that amplifies the order and inventory variability in the disaggregated model. Aggregated modeling is incapable of capturing the variability amplification due to adaptive prices also under A&A policy.

Table 6.8. Statistics of Aggregated vs. Disaggregated Model Comparison under Adaptive Pricing for A&A Policy

| | Agg. Inv. Mean | Disagg. Inv. Mean | Agg. Inv. Std.Dev. | Disagg. Inv. Std.Dev. | Agg. Order Mean | Disagg. Order Mean | Agg. Order Std.Dev. | Disagg. Order Std.Dev. |
|---|---|---|---|---|---|---|---|---|
| Rtl. | 149950 | 149870 | 413 | 1880 | 25004 | 24477 | 158 | 1578 |
| Whl. | 149930 | 154360 | 924 | 5723 | 25010 | 24478 | 270 | 4264 |
| Mnf. | 150060 | 150910 | 2315 | 11221 | 25015 | 24491 | 618 | 6410 |

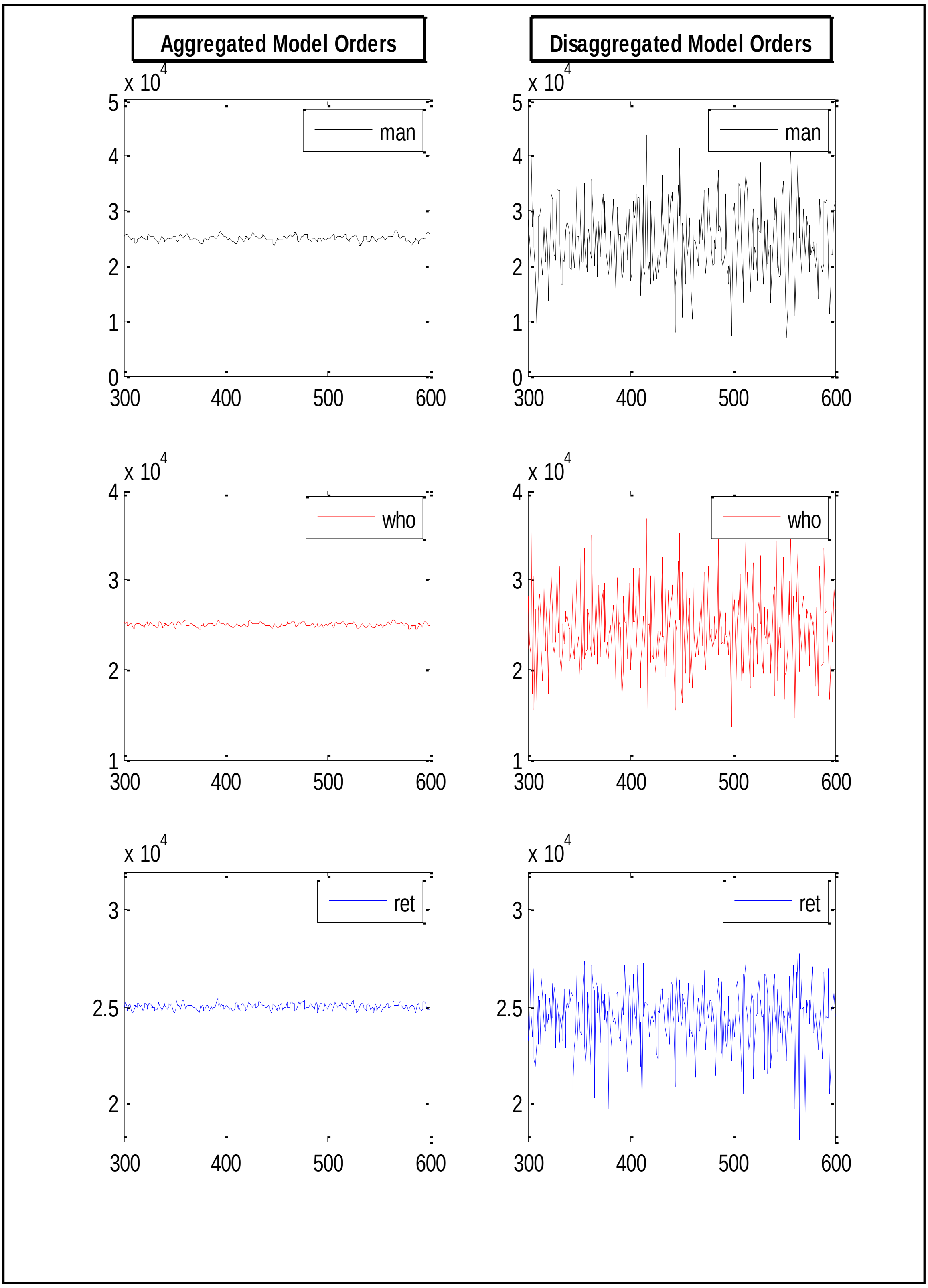


Figure 6.19. Aggregate Order Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for A&A Policy

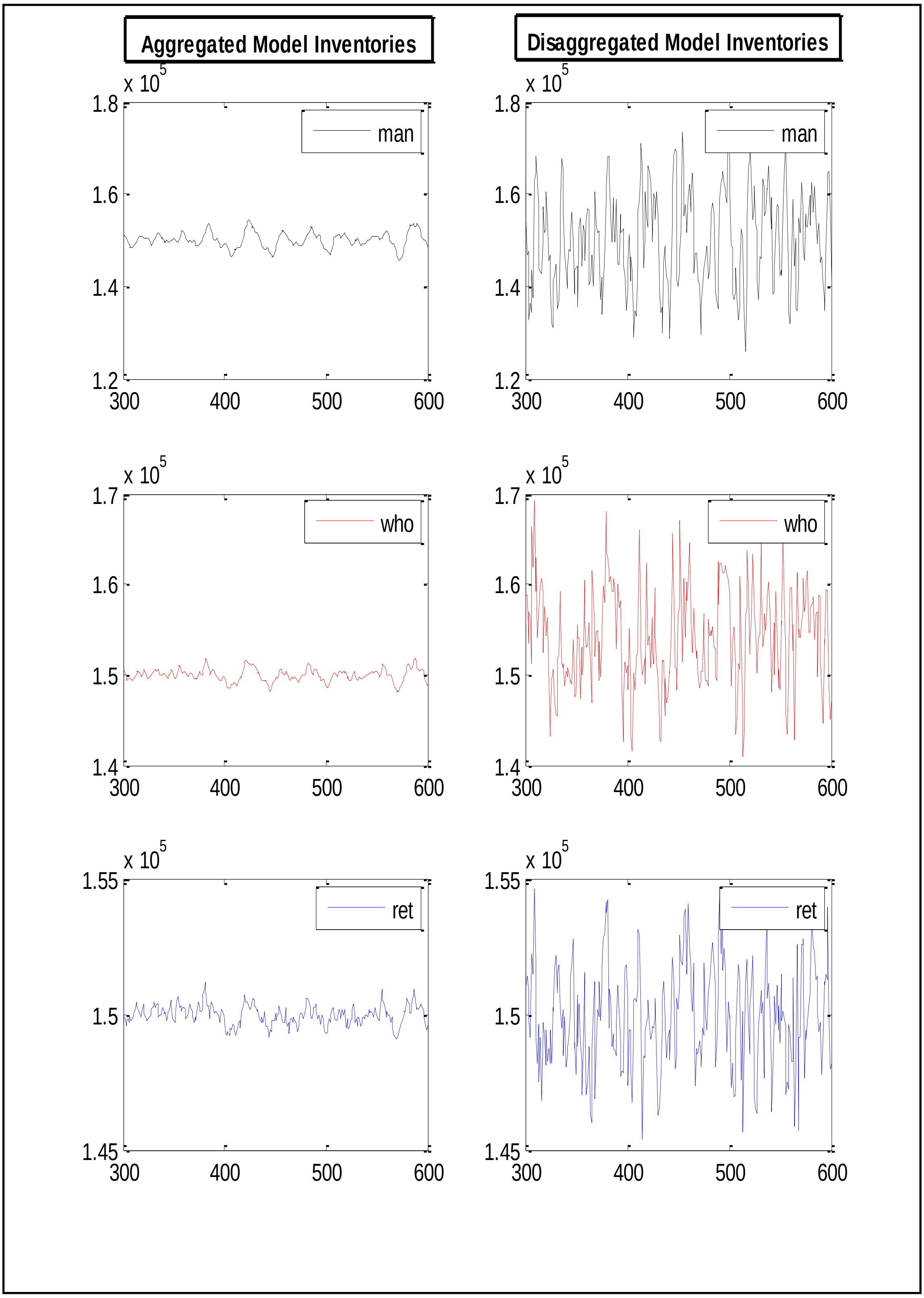


Figure 6.20. Aggregate Inventory Dynamics of Aggregated & Disaggregated Models under Adaptive Pricing for A&A Policy

# 7. CONCLUSIONS AND FURTHER RESEARCH

The level of aggregation is a major issue in the modeling and analysis of complex systems. Agent-based modeling and system dynamics are two commonly used methodologies which are conventionally placed at the two ends of the aggregation spectrum. This research asks the general questions about the level of aggregation in the scope of a generic, three-echelon, multiagent supply chain. First, an agent-based model of the supply chain is developed. Supply chain dynamics are analyzed with a special focus on the inventory and order variability, and also on the bullwhip effect. Demand forecasting, order batching, and dynamic pricing are among the most important behavioral causes of the variability and the bullwhip effect in supply chains. The effects of these structural factors on the supply chain dynamics are examined in the context of the agent-based models developed here. The interplays between agent features, agent dynamics, supply chain topology, and aggregate dynamics are investigated. Then, an aggregated system dynamics model of the supply chain is developed and its representative power is examined under different settings.

It is shown that using Fixed Order-Up-To-Level (Fixed S) policy -where firms do not use demand forecasting, order batching, and adaptive pricing features- provides unmodified transmission of the final customer demand throughout the supply chain. Hence the bullwhip effect does not exist in orders, and it emerges in inventories only if the lead times are relatively high for the upper echelons of the supply chain. These observations hold irrespective of the number of agents at any echelon and the numerical values of the base stock levels. As far as the relationship between individual and aggregate dynamics is concerned, averaging of the individual outputs removes most of the variability existing in individual dynamics, especially at higher echelons. The variability in individual dynamics is reduced as a result of the correlations and phase lags among them.

The impact of demand forecasting on the supply chain dynamics is illustrated through the analysis of Variable Order-Up-To-Level (Variable S) and Anchor-And-Adjust (A&A) policies. Under Variable S policy, the final customer demand variability is amplified through the supply chain as a result of demand forecasting. Bullwhip effect emerges in orders and inventories irrespective of the number of agents. The variability is significantly higher than the variability under Fixed S policy. As far as the relationship between individual and aggregate dynamics is concerned, aggregation of individual outputs removes only a small percentage of the variability existing in individual dynamics, compared to the large reduction under Fixed S policy, especially at high echelons. Aggregate order and inventory dynamics are affected from the topology through the interaction patterns among individual agents and the resulting individual dynamics. The "ratio of the number of agents at the upper echelon to the number of agents at the lower echelon" is critical for the aggregate order variability at the upper echelon. A transition from smooth individual demand/order patterns to intermittent individual demand/order patterns is observed, as this ratio is increased. The aggregate order variability is low under these two pattern extremes; however relatively high in between. On the other hand, aggregate inventory variability increases monotonically as the number of agents in an echelon increases. An important observation about the effect of supply chain topology is that aggregate inventory average increases with the increasing number of agents; whereas aggregate order averages remain the same. In other words, the system holds more -excess- inventory for a supply chain with higher number of firms; despite the aggregate average demand staying the same.

A&A policy seeks to eliminate the discrepancy between actual and desired levels smoothly within an interval of adjustment time. Therefore, A&A policy yields results similar to the Variable S policy, but the dynamics are smoother and display more regular oscillatory patterns, as one would expect.

The impact of order batching on the supply chain dynamics is illustrated through the analysis of Reorder Point - Order-Up-To-Level (s,S) policy. Bullwhip effect emerges in both aggregate order and inventory dynamics as a result of demand forecasting and order batching. Aggregate inventory and order variability are significantly higher than the variability under other basic policies, which is a direct consequence of order batching. As far as the relationship between individual and aggregate dynamics is concerned, aggregation reduces the variability in an amount close to the one expected by central limit theorem due to mere summing effect. In terms of the supply chain topology, the number of agents at an echelon affects both aggregate order and inventory dynamics significantly. As the number of agents in an echelon increases, aggregate order pattern evolves from intermittency to continuity due to the emergence of phase lags among (increased number of) individual intermittent orders. Therefore, aggregate order variability decreases as the number of agents at that echelon increases. Aggregate inventory variability behaves similarly up to a critical value of the number of agents at that echelon, but tends to increase for higher number of agents. The observation about excess inventories as a result of increased number of agents also holds for the (s,S) policy.

The impact of dynamic pricing on the supply chain dynamics is illustrated through an adaptive pricing scheme, where each supplier determines its price according to its inventory. Aggregate inventory averages are raised by the adaptive pricing mechanism, while a slight decrease in aggregate order averages is observed. Therefore, it can be concluded that adaptive pricing also leads to excess inventories in the supply chain. Aggregate order variability is also significantly increased by the adaptive pricing mechanism for all policies other than (s,S). Aggregate inventory variability is enhanced significantly by the adaptive pricing mechanism for all policies other than Fixed S. The counter-intuitive slight decrease in aggregate inventory variability under Fixed S policy becomes a significant increase for high number of agents. Therefore, it can be concluded that adaptive pricing leads to the amplification of aggregate inventory variability in almost all cases considered. In addition to these, adaptive pricing mechanism causes behavioral pattern changes in aggregate inventories, dominant periodicities also shifting toward low frequencies.

An aggregated system dynamics model of the multiagent supply chain is next developed and its power in representing aggregate order and inventory dynamics is analyzed. The representative power of the aggregated model is tested for all basic ordering policies, and also under adaptive pricing. Aggregated modeling proves to be no worse representative than disaggregated modeling for Fixed S policy, because aggregate order and inventory dynamics are independent of the number of agents and the interaction patterns among individual agents for this policy. Under Variable S and A&A policies, aggregated modeling captures the general patterns, periodicities, and averages of both orders and inventories well. It only underestimates the variability of orders and inventories to a degree, under these policies. It can be concluded that aggregated modeling is capable of representing the dynamics due to demand forecasting with the caution that it underestimates the variability. On the other hand, there are significant differences between the behavioral patterns of aggregated and disaggregated models for (s,S) policy. Aggregated modeling captures the variability amplification due to order batching, but fails to capture the behavioral patterns. It is additionally shown that aggregated modeling cannot capture the variability amplification due to adaptive pricing for any of these policies. This incapability basically results from the fact that the impossibility of having individual supplier selection mechanism in the aggregated model and aggregation of individual prices turns out to form a significantly different pricing structure. It is important to note that this incapability is true under the dynamic pricing formulation in this framework. Other pricing formulations not considered here may still be captured by aggregated modeling.

This research has theoretical implications for complexity research, as well as some practical implications for supply chain dynamics. A reasonable future research agenda would be to select one of two major directions, and to modify and improve both the model and the analysis accordingly. The first direction will require a simplified model with few parameters and less complicated agent schemata. Such a model thus may enable analytical work on the emergence of aggregate dynamics from local interactions and decision rules. Statistical physics, stochastics, and social network analysis may be the reference fields for such a research agenda. On the other hand, the second direction would be to apply the model to a real supply chain case. Demand patterns of final customers; priority rules, ordering policy parameters, and pricing policies of firms; topology of supply chain; and financial interactions among firms should be modeled based on the real case. It would require building a more realistic, heterogeneous, and possibly more complicated model based on the inventory management and supply chain literatures.

# APPENDIX A: CODE OF THE AGENT-BASED MODEL CODE

```
globals [
        time
        price-normal-retailer
        average-price-retailer
        price-normal-wholesaler
        average-price-wholesaler
        price-normal-manufacturer
        average-price-manufacturer
        retavgorderperiod
        whoavgorderperiod
        manavgorderperiod
        totretinv
        totwhoinv
        totmaninv
        totretorder
        totwhoorder
        totmanorder
        aggretdemand
        aggwhodemand
        aggmandemand
        ]

breed [ manufacturer ]
breed [ transit_man ]
breed [ wholesaler ]
breed [ transit_whole ]
breed [ retailer ]
breed [ customer ]

manufacturer-own [inventory
                  inventory_position
                  inventoryb
                  production_quantity
                  production_placed
                  production_list
                  time_list
                  lead_time
                  forecast
                  orderup_level
                  min_inventory
                  SS
                  backlog
                  incoming_demand
                  id_list
                  order_list
                  bid_list
                  backlog_list
                  order-style
                  ordercount
                  inventory_AT
                  supplyline_AT
                  desired_inventory
                  desired_supplyline
                  sales
                  expected_sales
                  price
                  lastprice
                  price-modification-time
                  pricefunction
                  incoming_demand_plot
                  reference_inventory
                  demandpercent
               ]

transit_man-own [id_list
                 order_list
                 time_list
                 lead_time]
```

```
wholesaler-own [inventory
                inventory_position
                inventoryb
                order
                order_given
                forecast
                orderup_level
                min_inventory
                SS
                backlog
                lead_time
                incoming_demand
                id_list
                order_list
                bid_list
                backlog_list
                order-style
                ordercount
                inventory_AT
                supplyline_AT
                desired_inventory
                desired_supplyline
                sales
                expected_sales
                price
                lastprice
                price-modification-time
                pricefunction
                demandpricefunction
                maxacceptprice
                incoming_demand_plot
                reference_inventory
                demandpercent
                indpriceforecast
                lastpurchaseprice
                loyal-man
                loyal-time
               ]

transit_whole-own [id_list
                   order_list
                   time_list
                   lead_time]

retailer-own[inventory
             inventory_position
             inventoryb
             order
             order_given
             forecast
             lead_time
             orderup_level
             min_inventory
             SS
             backlog
             incoming_demand
             forecast_demand
             order-style
             ordercount
             inventory_AT
             supplyline_AT
             desired_inventory
             desired_supplyline
             sales
             expected_sales
             price
             lastprice
             price-modification-time
             pricefunction
             demandpricefunction
             maxacceptprice
             incoming_demand_plot
             reference_inventory
             demandpercent
```

```
             indpriceforecast
             lastpurchaseprice
             loyal-who
             loyal-time
           ]

customer-own [demand maxacceptprice demandpricefunction indpriceforecast lastpurchaseprice
loyal-ret loyal-time]


;-------------------------SETUP-----------------------------
;----------------------------------------------------------
to setup ; creation of agents and initialization of variables

clear-turtles
clear-patches
clear-all-plots
clear-output
set time 0
set price-normal-retailer 1
set price-normal-wholesaler 1
set price-normal-manufacturer 1
let i ( 0 + min-pxcor + 5)
ask patches [ set pcolor 9.9999 ]

;-----------------manufacturer------------------------------

set-default-shape manufacturer "factory"
 create-manufacturer number-of-manufacturers
ask manufacturer
[without-interruption
 [set SIZE 10
  setxy i 15
  set i ( i + round (( 2 * max-pxcor ) / ( number-of-manufacturers + 1) ) )
  set production_quantity 0
  set production_placed 0
  set forecast (number-of-customers * 2.5 / number-of-manufacturers)
  set SS 2000 / number-of-manufacturers
  set orderup_level SS * 12
  set ordercount 0
  ifelse lead
  [set lead_time 5]
  [set lead_time 0]
  set inventory 5 * forecast
  set backlog 0
  set inventoryb ( inventory - backlog)
  set inventory_AT 3
  set supplyline_AT 3
  set desired_inventory inventory
  set desired_supplyline inventory
  set expected_sales forecast
  set incoming_demand 0
  set production_list []
  set time_list []
  set order_list []
  set id_list []
  set backlog_list []
  set bid_list []
  if orderpolicy = 1
  [set order-style 1]
  if orderpolicy = 2
  [set order-style 2]
  if orderpolicy = 3
  [set order-style 3]
  if orderpolicy = 4
  [set order-style 4]
  set price price-normal-manufacturer
  set lastprice price
  set price-modification-time -5
  set pricefunction ((random 2) + 1)
  set ordercount 0
  ]]

;----------------transit manufacturer------------------------
set-default-shape transit_man "truck-right"
```

```
create-transit_man 1
ask transit_man
[set color 3
set SIZE 6
setxy random-float world-width
        15
set order_list []
set time_list []
set id_list []
 ifelse lead
  [set lead_time 3]
  [set lead_time 0]]


;----------------wholesaler---------------------------------
set-default-shape wholesaler "whole"
create-wholesaler number-of-wholesalers
set i ( 0 + min-pxcor + 10)
ask wholesaler
[without-interruption
 [set SIZE 8
  setxy i 10
  set i ( i + round (( 2 * max-pxcor ) / ( number-of-wholesalers + 1) ) )

  set order 0
  set forecast (number-of-customers * 2.5 / number-of-wholesalers)
  set SS 2000 / number-of-wholesalers
  set orderup_level SS * 7
  set order_given 0
  set incoming_demand 0
  set ordercount 0
  set order 0
  set inventory 5 * forecast
  set backlog 0
  set inventoryb ( inventory - backlog)
  set inventory_AT 3
  set supplyline_AT 3
  set desired_inventory inventory
  set desired_supplyline inventory
  set expected_sales forecast
  set order_list []
  set id_list []
  set backlog_list []
  set bid_list []
   ifelse lead
  [set lead_time 3]
  [set lead_time 0]
  if orderpolicy = 1
  [set order-style 1]
  if orderpolicy = 2
  [set order-style 2]
  if orderpolicy = 3
  [set order-style 3]
  if orderpolicy = 4
  [set order-style 4]
  set price price-normal-wholesaler
  set lastprice price
  set price-modification-time -5
  set pricefunction ((random 2) + 1)
  set demandpricefunction ((random 2) + 1)
  set maxacceptprice (1.0 + random-float 0.2)
  set ordercount 0
  set indpriceforecast price-normal-manufacturer
  set lastpurchaseprice price-normal-manufacturer
  set loyal-man one-of manufacturer
  set loyal-time 0
  ]]
;---------------transit_wholesaler--------------------------

set-default-shape transit_whole "truck-left"
create-transit_whole 1
ask transit_whole
[set color 23
set SIZE 4
setxy random-float world-width 10
```

```
set order_list []
set time_list []
set id_list []
 ifelse lead
  [set lead_time 3]
  [set lead_time 0]]

;----------------retailer----------------------------------

set-default-shape retailer "house1"
create-retailer number-of-retailers
ask retailer
 [set SIZE 3
  setxy random-float world-width
        (-15 + random-float 20 )

  set order 0
  set forecast (number-of-customers * 2.5 / number-of-retailers)
  set SS 2000 / number-of-retailers
  set orderup_level SS * 3
  set order_given 0
  set incoming_demand 0
  set ordercount 0
  set inventory 5 * forecast
  set backlog 0
  set inventoryb ( inventory - backlog)
  set inventory_AT 3
  set supplyline_AT 3
  set desired_inventory inventory
  set desired_supplyline inventory
  set expected_sales forecast
  ifelse lead
  [set lead_time 3]
  [set lead_time 0]
  if orderpolicy = 1
  [set order-style 1]
  if orderpolicy = 2
  [set order-style 2]
  if orderpolicy = 3
  [set order-style 3]
  if orderpolicy = 4
  [set order-style 4]
  set price price-normal-retailer
  set lastprice price
  set price-modification-time -5
  set pricefunction ((random 2) + 1)
  set demandpricefunction ((random 2) + 1)
  set maxacceptprice (1.0 + random-float 0.2)
  set ordercount 0
  set indpriceforecast price-normal-wholesaler
  set lastpurchaseprice price-normal-wholesaler
  set loyal-who one-of wholesaler
  set loyal-time 0
  ]

;---------------customer------------------------------------
set-default-shape customer "person"
create-customer number-of-customers
ask customer
 [set SIZE 2
  setxy random-float world-width
        (-15 + random-float 20 )
  set demandpricefunction ((random 2) + 1)
  set maxacceptprice (1.0 + random-float 0.2)
  set indpriceforecast price-normal-retailer
  set lastpurchaseprice price-normal-retailer
  set loyal-ret one-of retailer
  set loyal-time 0
 ]

end

;-----------------------------------------------------------------
;--------------------GO-------------------------------------------
;-----------------------------------------------------------------
```

```
to go ; the recursive procedures that take place at each cycle

set time (time + 1)

ask retailer [set-price]
ask wholesaler [set-price]
ask manufacturer [set-price]

set average-price-retailer (sum [price] of retailer)/(number-of-retailers)
set average-price-wholesaler (sum [price] of wholesaler)/(number-of-wholesalers)
set average-price-manufacturer (sum [price] of manufacturer)/(number-of-manufacturers)

;------------Customer Procedures-----------------------------------------
ask customer
    [
    set indpriceforecast ( 0.75 * indpriceforecast + 0.25 * lastpurchaseprice)
    move
    set demand random-float 5
    buy
    ]


;-----------Retailer Procedures-------------------------------------------
set aggretdemand sum [incoming_demand] of retailer

ask transit_whole
    [
    right 90
    fd 3
    deliver
    ]

ask retailer
    [
    set indpriceforecast ( 0.75 * indpriceforecast + 0.25 * lastpurchaseprice)
    ifelse aggretdemand > 0
        [set demandpercent incoming_demand / aggretdemand] [set demandpercent -0.1]
    sell
    forecasting
    set incoming_demand_plot incoming_demand
    set incoming_demand 0
    set order 0
    ordering_to_wholesaler
    ]

;---------Wholesaler Procedures-------------------------------------------
set aggwhodemand sum [incoming_demand] of wholesaler

ask transit_man
    [
    left 90
    fd 1
    deliver
    ]

ask wholesaler
    [
    set indpriceforecast ( 0.75 * indpriceforecast + 0.25 * lastpurchaseprice)
    ifelse aggwhodemand > 0
        [set demandpercent incoming_demand / aggwhodemand] [set demandpercent -0.1]
    ship
    forecasting
    set incoming_demand_plot incoming_demand
    set incoming_demand 0
    set order 0
    ordering_to_manufacturer
    ]

;---------Manufacturer Procedures-----------------------------------------
set aggmandemand sum [incoming_demand] of manufacturer

ask manufacturer
    [
    ifelse aggmandemand > 0
        [set demandpercent incoming_demand / aggmandemand] [set demandpercent -0.1]
```

```
    finish-production
    ship
    forecasting
    set incoming_demand_plot incoming_demand
    set incoming_demand 0
    set production_quantity 0
    production
    ]

if ( time >= 300 )
    [do-plot]
if ( time = 1200 )
    [
    set retavgorderperiod (901 * number-of-retailers) / (sum [ordercount] of retailer)
    set whoavgorderperiod (901 * number-of-wholesalers) / (sum [ordercount] of wholesaler)
    set  manavgorderperiod  (901  *  number-of-manufacturers)  /  (sum  [ordercount]  of
manufacturer)
    stop
    ]

end ; end of go

;-------------------------------------------------------------------------
;------------PROCEDURE DECLARATIONS---------------------------------------
;-------------------------------------------------------------------------
;--------------------MOVE-------------------------------------------------
to move  ;; turtle procedure

  SET HEADING 360
  fd 1

end

;---------------BUY------------------------------------------------------------------
to buy

 let retailer-list 0
 let choice_retailer 0
 let i 0
 let change_ret 0

 set loyal-time (loyal-time + 1)
 set change_ret 0

 if (loyal-time >= 20 and [price] of loyal-ret  >= 1.3 * average-price-retailer) or (loyal-
time >= 20 and loyal-time < 10 and [price] of loyal-ret  >= 1.2 * average-price-retailer)
or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-ret  >= 1.1 * average-price-
retailer) or (loyal-time < 5 and [price] of loyal-ret  >= 1.05 * average-price-retailer)
   [ set change_ret 1]

 ifelse change_ret = 1
 [
 set loyal-time 0
 set i ( number-of-manufacturers + number-of-wholesalers + 2)
 set retailer-list []
 while [ i < ( number-of-manufacturers + number-of-wholesalers + number-of-retailers + 2)]
     [
     set retailer-list lput (turtle i) retailer-list
     set i (i + 1)
     ]

 while [ not empty? retailer-list ]
     [
     set choice_retailer one-of retailer-list
     ifelse  ( [inventory]  of  choice_retailer  -  [incoming_demand]  of  choice_retailer  )  >=
demand
         [
         ifelse  [price] of choice_retailer <= maxacceptprice * indpriceforecast
             [
             if demandpricefunction = 1
                 [
                 let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                 let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
```

```
                let xvalue ([price] of choice_retailer / indpriceforecast)
                ifelse xvalue >= 1.2
                    [
                    set demand (demand * 0.75)
                    ]
                    [
                    ifelse xvalue < 0.5
                        [
                        set demand (demand * 1.25)
                        ]
                        [
                        let xlowind floor (20 * (xvalue - 0.5))
                        set  demand  demand  *  (item  xlowind  yaxis  +  (item  (xlowind  +  1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                        ]
                    ]
                ]

            if demandpricefunction = 2
                [
                let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                let  yaxis  [1.500  1.490  1.475  1.455  1.415  1.370  1.305  1.240  1.180  1.100
1.000 0.875 0.690 0.565 0.500]
                let xvalue ([price] of choice_retailer / indpriceforecast)
                ifelse xvalue >= 1.2
                    [
                    set demand (demand * 0.5)
                    ]
                    [
                    ifelse xvalue < 0.5
                        [
                        set demand (demand * 1.5)
                        ]
                        [
                        let xlowind floor (20 * (xvalue - 0.5))
                        set  demand  demand  *  (item  xlowind  yaxis  +  (item  (xlowind  +  1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                        ]
                    ]
                ]

            set [incoming_demand] of choice_retailer ([incoming_demand] of choice_retailer
+ demand)
            set loyal-ret choice_retailer
            set lastpurchaseprice [price] of choice_retailer
            set retailer-list []
            ]
            [
            set retailer-list remove (choice_retailer) retailer-list
            set choice_retailer nobody
            ]
        ]
        [
        set retailer-list remove (choice_retailer) retailer-list
        set choice_retailer nobody
        ]
    ]

    if choice_retailer = nobody
        [without-interruption
            [
            set choice_retailer one-of retailer
            if demandpricefunction = 1
                [
                let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                let  yaxis  [1.25  1.25  1.24  1.225  1.210  1.19  1.16  1.135  1.095  1.060  1.000
0.905 0.835 0.775 0.75]
                let xvalue ([price] of choice_retailer / indpriceforecast)
                ifelse xvalue >= 1.2
                    [
                    set demand (demand * 0.75)
```

```
                    ]
                    [
                    ifelse xvalue < 0.5
                        [
                        set demand (demand * 1.25)
                        ]
                        [
                        let xlowind floor (20 * (xvalue - 0.5))
                        set demand demand * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                        ]
                    ]
                ]

            if demandpricefunction = 2
                [
                let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                let xvalue ([price] of choice_retailer / indpriceforecast)
                ifelse xvalue >= 1.2
                    [
                    set demand (demand * 0.5)
                    ]
                    [
                    ifelse xvalue < 0.5
                        [
                        set demand (demand * 1.5)
                        ]
                        [
                        let xlowind floor (20 * (xvalue - 0.5))
                        set demand demand * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                        ]
                    ]
                ]
            set [incoming_demand] of choice_retailer ([incoming_demand] of choice_retailer
+ demand)
            set loyal-ret choice_retailer
            set lastpurchaseprice [price] of choice_retailer
            ]
        ]
  ]
  [
    set [incoming_demand] of loyal-ret ([incoming_demand] of loyal-ret + demand)
  ]

end

;------------FORECAST-----------------------------------------------------------------
to forecasting
set forecast round (0.75 * forecast + 0.25 * incoming_demand)
set expected_sales (0.75 * expected_sales + 0.25 * sales)
end

;----------SELL---------------------------------------------------------------------
to sell
set sales 0
ifelse (inventory >= backlog) ; backlog kontrol
[without-interruption
 [set inventory (inventory - backlog)
 set sales (sales + backlog)
 set backlog 0]]
[without-interruption
 [set backlog (backlog - inventory)
 set sales inventory
 set inventory (0)]
]

ifelse (incoming_demand <= inventory) ; o dönemin kontrolü
 [set inventory (inventory - incoming_demand)
 set sales (sales + incoming_demand)
```

```
 set backlog 0]
 [without-interruption
 [set backlog (backlog + incoming_demand - inventory)
 set sales (sales + inventory)
 set inventory (0)]
  ]
set inventoryb (inventory - backlog)
end

;------------------------------------------------------------
;------------------------------------------------------------

to ordering_to_wholesaler

let choice_wholesaler 0
let change_who 0

set inventory_position (inventory + order_given - backlog)
set inventoryb (inventory - backlog)

if order-style = 1
  [if (orderup_level > inventory_position)
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let wholesaler-list []
      let i 0
      set order (orderup_level - inventory_position)
      set i ( number-of-manufacturers + 1)
      set loyal-time (loyal-time + 1)
      set change_who 0
      if (loyal-time >= 20 and [price] of loyal-who  >= 1.3 * average-price-wholesaler) or
(loyal-time >= 20 and loyal-time < 10 and [price] of loyal-who  >= 1.2 * average-price-
wholesaler) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-who  >= 1.1 *
average-price-wholesaler) or (loyal-time < 5 and [price] of loyal-who  >= 1.05 * average-
price-wholesaler)
          [ set change_who 1]

   ifelse change_who = 1
      [
      set loyal-time 0
      while [ i < ( number-of-manufacturers + number-of-wholesalers + 1)]
         [
         set wholesaler-list lput (turtle i) wholesaler-list
         set i (i + 1)
         ]

      while [ not empty? wholesaler-list ]
          [
          set choice_wholesaler one-of wholesaler-list
          ifelse  (  [inventory]  of  choice_wholesaler  -  [incoming_demand]  of
choice_wholesaler ) >= order
              [
              ifelse  [price] of choice_wholesaler <= maxacceptprice * indpriceforecast
                  [
                  if demandpricefunction = 1
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060
1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_wholesaler / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.25)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
```

```
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  if demandpricefunction = 2
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180
1.100 1.000 0.875 0.690 0.565 0.500]
                      let xvalue ([price] of choice_wholesaler / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.5)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.5)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  set order_given (order_given + order)
                  set [incoming_demand] of choice_wholesaler ([incoming_demand] of
choice_wholesaler + order)
                  set loyal-who choice_wholesaler
                  set lastpurchaseprice [price] of choice_wholesaler
                  set ([order_list] of choice_wholesaler) lput order ([order_list] of
choice_wholesaler)
                  set ([id_list] of choice_wholesaler) lput (turtle who) ([id_list] of
choice_wholesaler)
                  setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
                  set wholesaler-list []
                  ]
                  [
                  set wholesaler-list remove (choice_wholesaler) wholesaler-list
                  set choice_wholesaler nobody
                  ]
              ]
              [
              set wholesaler-list remove (choice_wholesaler) wholesaler-list
              set choice_wholesaler nobody
              ]
          ]

      if choice_wholesaler = nobody
          [without-interruption
              [
              set choice_wholesaler one-of wholesaler
              if demandpricefunction = 1
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                  let xvalue ([price] of choice_wholesaler / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.75)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
```

```
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_wholesaler / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.5)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]

              set order_given (order_given + order)
              set [incoming_demand] of choice_wholesaler ([incoming_demand] of
choice_wholesaler + order)
              set lastpurchaseprice [price] of choice_wholesaler
              set ([order_list] of choice_wholesaler) lput order ([order_list] of
choice_wholesaler)
              set ([id_list] of choice_wholesaler) lput (turtle who) ([id_list] of
choice_wholesaler)
              set loyal-who choice_wholesaler
              setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-who ([incoming_demand] of loyal-who + order)
      set ([order_list] of loyal-who) lput order ([order_list] of loyal-who)
      set ([id_list] of loyal-who) lput (turtle who) ([id_list] of loyal-who)
      setxy [xcor] of loyal-who [ycor] of loyal-who
      ]
    ]
    ]
  ]



if order-style = 2
  [
  set orderup_level ((forecast * lead_time  + SS) + 3 * forecast)
  if (orderup_level > inventory_position)
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let wholesaler-list []
      let i 0
      set order (orderup_level - inventory_position)
       set i ( number-of-manufacturers + 1)
      set loyal-time (loyal-time + 1)
      set change_who 0
      if (loyal-time >= 20 and [price] of loyal-who  >= 1.3 * average-price-wholesaler) or
(loyal-time >= 20 and loyal-time < 10 and [price] of loyal-who  >= 1.2 * average-price-
```

```
wholesaler) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-who  >= 1.1 * average-price-wholesaler) or (loyal-time < 5 and [price] of loyal-who  >= 1.05 * average-price-wholesaler)
          [ set change_who 1]

   ifelse change_who = 1
      [
      set loyal-time 0
      while [ i < ( number-of-manufacturers + number-of-wholesalers + 1)]
         [
         set wholesaler-list lput (turtle i) wholesaler-list
         set i (i + 1)
         ]

      while [ not empty? wholesaler-list ]
          [
          set choice_wholesaler one-of wholesaler-list
          ifelse ( [inventory] of choice_wholesaler - [incoming_demand] of choice_wholesaler ) >= order
              [
              ifelse  [price] of choice_wholesaler <= maxacceptprice * indpriceforecast
                  [
                  if demandpricefunction = 1
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1 1.15 1.20]
                      let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_wholesaler / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.25)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item xlowind xaxis )))
                              ]
                          ]
                      ]

                  if demandpricefunction = 2
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1 1.15 1.20]
                      let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100 1.000 0.875 0.690 0.565 0.500]
                      let xvalue ([price] of choice_wholesaler / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.5)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.5)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item xlowind xaxis )))
                              ]
                          ]
                      ]

                  set order_given (order_given + order)
                  set [incoming_demand] of choice_wholesaler ([incoming_demand] of choice_wholesaler + order)
```

```
                  set loyal-who choice_wholesaler
                  set lastpurchaseprice [price] of choice_wholesaler
                  set ([order_list] of choice_wholesaler) lput order ([order_list] of
choice_wholesaler)
                  set ([id_list] of choice_wholesaler) lput (turtle who) ([id_list] of
choice_wholesaler)
                  setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
                  set wholesaler-list []
                  ]
                  [
                  set wholesaler-list remove (choice_wholesaler) wholesaler-list
                  set choice_wholesaler nobody
                  ]
              ]
              [
              set wholesaler-list remove (choice_wholesaler) wholesaler-list
              set choice_wholesaler nobody
              ]
          ]

      if choice_wholesaler = nobody
          [without-interruption
              [
              set choice_wholesaler one-of wholesaler
              if demandpricefunction = 1
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                  let xvalue ([price] of choice_wholesaler / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.75)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_wholesaler / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.5)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]

              set order_given (order_given + order)
```

```
              set    [incoming_demand]    of    choice_wholesaler    ([incoming_demand]    of
choice_wholesaler + order)
              set lastpurchaseprice [price] of choice_wholesaler
              set   ([order_list]   of   choice_wholesaler)   lput   order   ([order_list]   of
choice_wholesaler)
              set   ([id_list]   of   choice_wholesaler)   lput   (turtle   who)   ([id_list]   of
choice_wholesaler)
              set loyal-who choice_wholesaler
              setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-who ([incoming_demand] of loyal-who + order)
      set ([order_list] of loyal-who) lput order ([order_list] of loyal-who)
      set ([id_list] of loyal-who) lput (turtle who) ([id_list] of loyal-who)
      setxy [xcor] of loyal-who [ycor] of loyal-who
      ]
    ]
    ]
  ]


if order-style = 3
  [
  set min_inventory (forecast * lead_time  + SS)
  set orderup_level ((forecast * lead_time  + SS) + 3 * forecast)
  if (min_inventory > inventory_position)
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let wholesaler-list []
      let i 0
      set order (orderup_level - inventory_position)
      set i ( number-of-manufacturers + 1)
      set loyal-time (loyal-time + 1)
      set change_who 0
      if (loyal-time >= 20 and [price] of loyal-who  >= 1.3 * average-price-wholesaler) or
(loyal-time >= 20 and loyal-time < 10 and [price] of loyal-who  >= 1.2 * average-price-
wholesaler) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-who  >= 1.1 *
average-price-wholesaler) or (loyal-time < 5 and [price] of loyal-who  >= 1.05 * average-
price-wholesaler)
          [ set change_who 1]

   ifelse change_who = 1
      [
      set loyal-time 0
      while [ i < ( number-of-manufacturers + number-of-wholesalers + 1)]
         [
         set wholesaler-list lput (turtle i) wholesaler-list
         set i (i + 1)
         ]

      while [ not empty? wholesaler-list ]
          [
          set choice_wholesaler one-of wholesaler-list
          ifelse    (    [inventory]    of    choice_wholesaler    -    [incoming_demand]    of
choice_wholesaler ) >= order
              [
              ifelse  [price] of choice_wholesaler <= maxacceptprice * indpriceforecast
                  [
                  if demandpricefunction = 1
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let  yaxis  [1.25  1.25  1.24  1.225  1.210  1.19  1.16  1.135  1.095  1.060
1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_wholesaler / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
```

```
                        ifelse xvalue < 0.5
                            [
                            set order (order * 1.25)
                            ]
                            [
                            let xlowind floor (20 * (xvalue - 0.5))
                            set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                            ]
                        ]
                    ]

                if demandpricefunction = 2
                    [
                    let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                    let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180
1.100 1.000 0.875 0.690 0.565 0.500]
                    let xvalue ([price] of choice_wholesaler / indpriceforecast)
                    ifelse xvalue >= 1.2
                        [
                        set order (order * 0.5)
                        ]
                        [
                        ifelse xvalue < 0.5
                            [
                            set order (order * 1.5)
                            ]
                            [
                            let xlowind floor (20 * (xvalue - 0.5))
                            set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                            ]
                        ]
                    ]

                set order_given (order_given + order)
                set [incoming_demand] of choice_wholesaler ([incoming_demand] of
choice_wholesaler + order)
                set loyal-who choice_wholesaler
                set lastpurchaseprice [price] of choice_wholesaler
                set ([order_list] of choice_wholesaler) lput order ([order_list] of
choice_wholesaler)
                set ([id_list] of choice_wholesaler) lput (turtle who) ([id_list] of
choice_wholesaler)
                setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
                set wholesaler-list []
                ]
                [
                set wholesaler-list remove (choice_wholesaler) wholesaler-list
                set choice_wholesaler nobody
                ]
            ]
            [
            set wholesaler-list remove (choice_wholesaler) wholesaler-list
            set choice_wholesaler nobody
            ]
        ]

    if choice_wholesaler = nobody
        [without-interruption
            [
            set choice_wholesaler one-of wholesaler
            if demandpricefunction = 1
                [
                let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                let xvalue ([price] of choice_wholesaler / indpriceforecast)
                ifelse xvalue >= 1.2
                    [
                    set order (order * 0.75)
```

```
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_wholesaler / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.5)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]

              set order_given (order_given + order)
              set [incoming_demand] of choice_wholesaler ([incoming_demand] of
choice_wholesaler + order)
              set lastpurchaseprice [price] of choice_wholesaler
              set ([order_list] of choice_wholesaler) lput order ([order_list] of
choice_wholesaler)
              set ([id_list] of choice_wholesaler) lput (turtle who) ([id_list] of
choice_wholesaler)
              set loyal-who choice_wholesaler
              setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-who ([incoming_demand] of loyal-who + order)
      set ([order_list] of loyal-who) lput order ([order_list] of loyal-who)
      set ([id_list] of loyal-who) lput (turtle who) ([id_list] of loyal-who)
      setxy [xcor] of loyal-who [ycor] of loyal-who
      ]
    ]
    ]
  ]

if order-style = 4
  [
  set desired_inventory (5 * forecast)
  set desired_supplyline (lead_time * forecast)
  if (desired_inventory - inventoryb) / inventory_AT + (desired_supplyline - order_given) /
supplyline_AT + expected_sales > 0
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let wholesaler-list []
      let i 0
```

```
      set order ((desired_inventory - inventoryb) / inventory_AT + (desired_supplyline - order_given) / supplyline_AT + expected_sales)
       set i ( number-of-manufacturers + 1)
      set loyal-time (loyal-time + 1)
      set change_who 0
      if (loyal-time >= 20 and [price] of loyal-who  >= 1.3 * average-price-wholesaler) or (loyal-time >= 20 and loyal-time < 10 and [price] of loyal-who  >= 1.2 * average-price-wholesaler) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-who  >= 1.1 * average-price-wholesaler) or (loyal-time < 5 and [price] of loyal-who  >= 1.05 * average-price-wholesaler)
          [ set change_who 1]

   ifelse change_who = 1
      [
      set loyal-time 0
      while [ i < ( number-of-manufacturers + number-of-wholesalers + 1)]
         [
         set wholesaler-list lput (turtle i) wholesaler-list
         set i (i + 1)
         ]

      while [ not empty? wholesaler-list ]
          [
          set choice_wholesaler one-of wholesaler-list
          ifelse ( [inventory] of choice_wholesaler - [incoming_demand] of choice_wholesaler ) >= order
              [
              ifelse  [price] of choice_wholesaler <= maxacceptprice * indpriceforecast
                  [
                  if demandpricefunction = 1
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1 1.15 1.20]
                      let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_wholesaler / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.25)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item xlowind xaxis )))
                              ]
                          ]
                      ]

                  if demandpricefunction = 2
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1 1.15 1.20]
                      let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100 1.000 0.875 0.690 0.565 0.500]
                      let xvalue ([price] of choice_wholesaler / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.5)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.5)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item xlowind xaxis )))
```

```
                                ]
                            ]
                        ]

                    set order_given (order_given + order)
                    set [incoming_demand] of choice_wholesaler ([incoming_demand] of
choice_wholesaler + order)
                    set loyal-who choice_wholesaler
                    set lastpurchaseprice [price] of choice_wholesaler
                    set ([order_list] of choice_wholesaler) lput order ([order_list] of
choice_wholesaler)
                    set ([id_list] of choice_wholesaler) lput (turtle who) ([id_list] of
choice_wholesaler)
                    setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
                    set wholesaler-list []
                    ]
                    [
                    set wholesaler-list remove (choice_wholesaler) wholesaler-list
                    set choice_wholesaler nobody
                    ]
                ]
                [
                set wholesaler-list remove (choice_wholesaler) wholesaler-list
                set choice_wholesaler nobody
                ]
            ]

      if choice_wholesaler = nobody
          [without-interruption
              [
              set choice_wholesaler one-of wholesaler
              if demandpricefunction = 1
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                  let xvalue ([price] of choice_wholesaler / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.75)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_wholesaler / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.5)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
```

```
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]

              set order_given (order_given + order)
              set   [incoming_demand]   of   choice_wholesaler   ([incoming_demand]   of
choice_wholesaler + order)
              set lastpurchaseprice [price] of choice_wholesaler
              set  ([order_list]  of  choice_wholesaler)  lput  order  ([order_list]  of
choice_wholesaler)
              set  ([id_list]  of  choice_wholesaler)  lput  (turtle  who)  ([id_list]  of
choice_wholesaler)
              set loyal-who choice_wholesaler
              setxy [xcor] of choice_wholesaler [ycor] of choice_wholesaler
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-who ([incoming_demand] of loyal-who + order)
      set ([order_list] of loyal-who) lput order ([order_list] of loyal-who)
      set ([id_list] of loyal-who) lput (turtle who) ([id_list] of loyal-who)
      setxy [xcor] of loyal-who [ycor] of loyal-who
      ]
    ]
    ]
  ]

end
;------------------------------------------------------------------------------------------
---

to ship
let a 0

ifelse (breed = manufacturer)
[set a true]
[set a false]
set sales 0
while  [(inventory > 0) and (not empty? backlog_list)]
[ifelse (inventory >= first backlog_list)
[without-interruption
[set inventory (inventory - first backlog_list)
 set sales (sales + first backlog_list)
 ifelse (a)
 [set ([order_list] of one-of transit_man) lput first backlog_list ([order_list] of one-of
transit_man)
  set  ([id_list]  of  one-of  transit_man)  lput  first  bid_list  ([id_list]  of  one-of
transit_man)
  set ([time_list] of one-of transit_man) lput time ([time_list] of one-of transit_man)]
 [set ([order_list] of one-of transit_whole) lput first backlog_list ([order_list] of one-
of transit_whole)
  set  ([id_list]  of  one-of  transit_whole)  lput  first  bid_list  ([id_list]  of  one-of
transit_whole)
  set   ([time_list]   of   one-of   transit_whole)   lput   time   ([time_list]   of   one-of
transit_whole)]
  set bid_list but-first bid_list
 set backlog_list but-first backlog_list]]
[ without-interruption
 [ifelse (a)
 [set  ([order_list]  of  one-of  transit_man)  lput  inventory  ([order_list]  of  one-of
transit_man)
  set  ([id_list]  of  one-of  transit_man)  lput  first  bid_list  ([id_list]  of  one-of
transit_man)
  set ([time_list] of one-of transit_man) lput time ([time_list] of one-of transit_man)]
  [set  ([order_list]  of  one-of  transit_whole)  lput  inventory  ([order_list]  of  one-of
transit_whole)
  set  ([id_list]  of  one-of  transit_whole)  lput  first  bid_list  ([id_list]  of  one-of
transit_whole)
  set   ([time_list]   of   one-of   transit_whole)   lput   time   ([time_list]   of   one-of
transit_whole)]
```

```
 set backlog_list replace-item 0 backlog_list (first backlog_list - inventory)
 set sales (sales + inventory)
 set inventory 0
 ]
 ];else commands
 ]

while [(inventory > 0) and (not empty? order_list)]
[ifelse (inventory >= first order_list)
[without-interruption
[set inventory (inventory - first order_list)
 set sales (sales + first order_list)
ifelse (a)
 [set ([order_list] of one-of transit_man) lput first order_list ([order_list] of one-of transit_man)
  set ([id_list] of one-of transit_man) lput first id_list ([id_list] of one-of transit_man)
  set ([time_list] of one-of transit_man) lput time ([time_list] of one-of transit_man)]
  [set ([order_list] of one-of transit_whole) lput first order_list ([order_list] of one-of transit_whole)
  set ([id_list] of one-of transit_whole) lput first id_list ([id_list] of one-of transit_whole)
  set ([time_list] of one-of transit_whole) lput time ([time_list] of one-of transit_whole)]

 set id_list but-first id_list
 set order_list but-first order_list]]

[without-interruption
  [ifelse (a)
 [set ([order_list] of one-of transit_man) lput inventory ([order_list] of one-of transit_man)
  set ([id_list] of one-of transit_man) lput first id_list ([id_list] of one-of transit_man)
  set ([time_list] of one-of transit_man) lput time ([time_list] of one-of transit_man)]
  [set ([order_list] of one-of transit_whole) lput inventory ([order_list] of one-of transit_whole)
  set ([id_list] of one-of transit_whole) lput first id_list ([id_list] of one-of transit_whole)
  set ([time_list] of one-of transit_whole) lput time ([time_list] of one-of transit_whole)]

 set order_list replace-item 0 order_list (first order_list - inventory)
 set sales (sales + inventory)
 set inventory 0]
 ];else commands
 ]

 without-interruption
 [set backlog_list sentence backlog_list order_list
 set bid_list sentence bid_list id_list
 set backlog sum backlog_list]

 while [not empty? order_list] ;liste bosalt
 [set order_list but-first order_list]
 while [not empty? id_list]
 [set id_list but-first id_list]

end

;-----------------------------------------------------------------------------
to deliver
while [not empty? order_list]
[ifelse (time - first time_list) >= lead_time
[set [inventory] of first id_list ([inventory] of first id_list + first order_list)
set [order_given] of first id_list ([order_given] of first id_list - first order_list)
set order_list but-first order_list
set id_list but-first id_list
set time_list but-first time_list]
[stop]]
end

;-----------------------------------------------------------------------------

to ordering_to_manufacturer
```

```
let choice_manufacturer 0
let change_man 0
set inventory_position (inventory + order_given - backlog)
set inventoryb (inventory - backlog)

if order-style = 1
  [if (orderup_level > inventory_position)
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let manufacturer-list []
      let i 0
      set order (orderup_level - inventory_position)
      set loyal-time (loyal-time + 1)
      set change_man 0
      if (loyal-time >= 20 and [price] of loyal-man  >= 1.3 * average-price-manufacturer)
or (loyal-time >= 10 and loyal-time < 20 and [price] of loyal-man  >= 1.2 * average-price-
manufacturer) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-man  >= 1.1 *
average-price-manufacturer) or (loyal-time < 5 and [price] of loyal-man  >= 1.05 * average-
price-manufacturer)
   [ set change_man 1]

ifelse change_man = 1
   [set loyal-time 0
      while [ i < ( number-of-manufacturers )]
         [ set manufacturer-list lput (turtle i) manufacturer-list
         set i (i + 1)
          ]

      while [ not empty? manufacturer-list ]
          [
          set choice_manufacturer one-of manufacturer-list
          ifelse   (   [inventory]   of   choice_manufacturer   -   [incoming_demand]   of
choice_manufacturer ) >= order
              [
              ifelse  [price] of choice_manufacturer <= maxacceptprice * indpriceforecast
                  [
                  if demandpricefunction = 1
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let  yaxis  [1.25  1.25  1.24  1.225  1.210  1.19  1.16  1.135  1.095  1.060
1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.25)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  if demandpricefunction = 2
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let  yaxis  [1.500  1.490  1.475  1.455  1.415  1.370  1.305  1.240  1.180
1.100 1.000 0.875 0.690 0.565 0.500]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.5)
                          ]
```

```
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.5)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  set order_given (order_given + order)
                  set  [incoming_demand]  of  choice_manufacturer  ([incoming_demand]  of
choice_manufacturer + order)
                  set lastpurchaseprice [price] of choice_manufacturer
                  set  ([order_list]  of  choice_manufacturer)  lput  order  ([order_list]  of
choice_manufacturer)
                  set  ([id_list]  of  choice_manufacturer)  lput  (turtle  who)  ([id_list]  of
choice_manufacturer)
                  set loyal-man choice_manufacturer
                  setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
                  set manufacturer-list []
                  ]
                  [
                  set manufacturer-list remove (choice_manufacturer) manufacturer-list
                  set choice_manufacturer nobody
                  ]
              ]
              [
              set manufacturer-list remove (choice_manufacturer) manufacturer-list
              set choice_manufacturer nobody
              ]
          ]

      if choice_manufacturer = nobody
          [without-interruption
              [
              set choice_manufacturer one-of manufacturer
              if demandpricefunction = 1
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.75)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
```

```
                    ]
                    [
                    ifelse xvalue < 0.5
                        [
                        set order (order * 1.5)
                        ]
                        [
                        let xlowind floor (20 * (xvalue - 0.5))
                        set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                        ]
                    ]
                ]

              set order_given (order_given + order)
              set [incoming_demand] of choice_manufacturer ([incoming_demand] of
choice_manufacturer + order)
              set lastpurchaseprice [price] of choice_manufacturer
              set ([order_list] of choice_manufacturer) lput order ([order_list] of
choice_manufacturer)
              set ([id_list] of choice_manufacturer) lput (turtle who) ([id_list] of
choice_manufacturer)
              set loyal-man choice_manufacturer
              setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-man ([incoming_demand] of loyal-man + order)
      set ([order_list] of loyal-man) lput order ([order_list] of loyal-man)
      set ([id_list] of loyal-man) lput (turtle who) ([id_list] of loyal-man)
      setxy [xcor] of loyal-man [ycor] of loyal-man
      ]
    ]
    ]
  ]

if order-style = 2
  [
  set orderup_level ((forecast * lead_time  + SS) + 3 * forecast)
  if (orderup_level > inventory_position)
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let manufacturer-list []
      let i 0
      set order (orderup_level - inventory_position)
      set loyal-time (loyal-time + 1)
      set change_man 0
      if (loyal-time >= 20 and [price] of loyal-man  >= 1.3 * average-price-manufacturer)
or (loyal-time >= 10 and loyal-time < 20 and [price] of loyal-man  >= 1.2 * average-price-
manufacturer) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-man  >= 1.1 *
average-price-manufacturer) or (loyal-time < 5 and [price] of loyal-man  >= 1.05 * average-
price-manufacturer)
   [ set change_man 1]

ifelse change_man = 1
   [set loyal-time 0
      while [ i < ( number-of-manufacturers )]
         [ set manufacturer-list lput (turtle i) manufacturer-list
         set i (i + 1)
          ]

      while [ not empty? manufacturer-list ]
          [
          set choice_manufacturer one-of manufacturer-list
          ifelse ( [inventory] of choice_manufacturer - [incoming_demand] of
choice_manufacturer ) >= order
              [
              ifelse  [price] of choice_manufacturer <= maxacceptprice * indpriceforecast
                  [
                  if demandpricefunction = 1
```

```
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060
1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.25)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  if demandpricefunction = 2
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180
1.100 1.000 0.875 0.690 0.565 0.500]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.5)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.5)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  set order_given (order_given + order)
                  set [incoming_demand] of choice_manufacturer ([incoming_demand] of
choice_manufacturer + order)
                  set lastpurchaseprice [price] of choice_manufacturer
                  set ([order_list] of choice_manufacturer) lput order ([order_list] of
choice_manufacturer)
                  set ([id_list] of choice_manufacturer) lput (turtle who) ([id_list] of
choice_manufacturer)
                  set loyal-man choice_manufacturer
                  setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
                  set manufacturer-list []
                  ]
                  [
                  set manufacturer-list remove (choice_manufacturer) manufacturer-list
                  set choice_manufacturer nobody
                  ]
              ]
              [
              set manufacturer-list remove (choice_manufacturer) manufacturer-list
              set choice_manufacturer nobody
              ]
          ]

      if choice_manufacturer = nobody
          [without-interruption
              [
```

```
              set choice_manufacturer one-of manufacturer
              if demandpricefunction = 1
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.75)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.5)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]

              set order_given (order_given + order)
              set [incoming_demand] of choice_manufacturer ([incoming_demand] of
choice_manufacturer + order)
              set lastpurchaseprice [price] of choice_manufacturer
              set ([order_list] of choice_manufacturer) lput order ([order_list] of
choice_manufacturer)
              set ([id_list] of choice_manufacturer) lput (turtle who) ([id_list] of
choice_manufacturer)
              set loyal-man choice_manufacturer
              setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-man ([incoming_demand] of loyal-man + order)
      set ([order_list] of loyal-man) lput order ([order_list] of loyal-man)
      set ([id_list] of loyal-man) lput (turtle who) ([id_list] of loyal-man)
      setxy [xcor] of loyal-man [ycor] of loyal-man
      ]
    ]
    ]
  ]
```

```
if order-style = 3
  [
  set min_inventory (forecast * lead_time  + SS)
  set orderup_level ((forecast * lead_time  + SS) + 3 * forecast)
  if (min_inventory > inventory_position)
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let manufacturer-list []
      let i 0
      set order (orderup_level - inventory_position)
      set loyal-time (loyal-time + 1)
      set change_man 0
      if (loyal-time >= 20 and [price] of loyal-man  >= 1.3 * average-price-manufacturer)
or (loyal-time >= 10 and loyal-time < 20 and [price] of loyal-man  >= 1.2 * average-price-
manufacturer) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-man  >= 1.1 *
average-price-manufacturer) or (loyal-time < 5 and [price] of loyal-man  >= 1.05 * average-
price-manufacturer)
   [ set change_man 1]

ifelse change_man = 1
   [set loyal-time 0
      while [ i < ( number-of-manufacturers )]
         [ set manufacturer-list lput (turtle i) manufacturer-list
         set i (i + 1)
          ]

      while [ not empty? manufacturer-list ]
          [
          set choice_manufacturer one-of manufacturer-list
          ifelse ( [inventory] of choice_manufacturer - [incoming_demand] of
choice_manufacturer ) >= order
              [
              ifelse  [price] of choice_manufacturer <= maxacceptprice * indpriceforecast
                  [
                  if demandpricefunction = 1
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060
1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.25)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  if demandpricefunction = 2
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180
1.100 1.000 0.875 0.690 0.565 0.500]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.5)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
```

```
                              set order (order * 1.5)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  set order_given (order_given + order)
                  set  [incoming_demand]  of  choice_manufacturer  ([incoming_demand]  of
choice_manufacturer + order)
                  set lastpurchaseprice [price] of choice_manufacturer
                  set  ([order_list]  of  choice_manufacturer)  lput  order  ([order_list]  of
choice_manufacturer)
                  set  ([id_list]  of  choice_manufacturer)  lput  (turtle  who)  ([id_list]  of
choice_manufacturer)
                  set loyal-man choice_manufacturer
                  setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
                  set manufacturer-list []
                  ]
                  [
                  set manufacturer-list remove (choice_manufacturer) manufacturer-list
                  set choice_manufacturer nobody
                  ]
              ]
              [
              set manufacturer-list remove (choice_manufacturer) manufacturer-list
              set choice_manufacturer nobody
              ]
          ]

      if choice_manufacturer = nobody
          [without-interruption
              [
              set choice_manufacturer one-of manufacturer
              if demandpricefunction = 1
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.75)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let  yaxis  [1.500  1.490  1.475  1.455  1.415  1.370  1.305  1.240  1.180  1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
                      ]
                      [
                      ifelse xvalue < 0.5
```

```
                          [
                          set order (order * 1.5)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]

              set order_given (order_given + order)
              set   [incoming_demand]   of   choice_manufacturer   ([incoming_demand]   of
choice_manufacturer + order)
              set lastpurchaseprice [price] of choice_manufacturer
              set  ([order_list]  of  choice_manufacturer)  lput  order  ([order_list]  of
choice_manufacturer)
              set  ([id_list]  of  choice_manufacturer)  lput  (turtle  who)  ([id_list]  of
choice_manufacturer)
              set loyal-man choice_manufacturer
              setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-man ([incoming_demand] of loyal-man + order)
      set ([order_list] of loyal-man) lput order ([order_list] of loyal-man)
      set ([id_list] of loyal-man) lput (turtle who) ([id_list] of loyal-man)
      setxy [xcor] of loyal-man [ycor] of loyal-man
      ]
    ]
    ]
  ]


if order-style = 4
  [
  set desired_inventory (5 * forecast)
  set desired_supplyline (lead_time * forecast)
  if (desired_inventory - inventoryb) / inventory_AT + (desired_supplyline - order_given) /
supplyline_AT + expected_sales > 0
    [without-interruption
    [
      if time >= 300
        [set ordercount ordercount + 1]
      let manufacturer-list []
      let i 0
      set  order  (desired_inventory  -  inventoryb)  /  inventory_AT  +  (desired_supplyline  -
order_given) / supplyline_AT + expected_sales
      set loyal-time (loyal-time + 1)
      set change_man 0
      if (loyal-time >= 20 and [price] of loyal-man  >= 1.3 * average-price-manufacturer)
or (loyal-time >= 10 and loyal-time < 20 and [price] of loyal-man  >= 1.2 * average-price-
manufacturer) or (loyal-time >= 5 and loyal-time < 10 and [price] of loyal-man  >= 1.1 *
average-price-manufacturer) or (loyal-time < 5 and [price] of loyal-man  >= 1.05 * average-
price-manufacturer)
   [ set change_man 1]

ifelse change_man = 1
   [set loyal-time 0
      while [ i < ( number-of-manufacturers )]
         [ set manufacturer-list lput (turtle i) manufacturer-list
         set i (i + 1)
          ]

      while [ not empty? manufacturer-list ]
          [
          set choice_manufacturer one-of manufacturer-list
          ifelse   (   [inventory]   of   choice_manufacturer   -   [incoming_demand]   of
choice_manufacturer ) >= order
              [
              ifelse  [price] of choice_manufacturer <= maxacceptprice * indpriceforecast
                  [
```

```
                  if demandpricefunction = 1
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060
1.000 0.905 0.835 0.775 0.75]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.75)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.25)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  if demandpricefunction = 2
                      [
                      let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
1.1 1.15 1.20]
                      let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180
1.100 1.000 0.875 0.690 0.565 0.500]
                      let xvalue ([price] of choice_manufacturer / indpriceforecast)
                      ifelse xvalue >= 1.2
                          [
                          set order (order * 0.5)
                          ]
                          [
                          ifelse xvalue < 0.5
                              [
                              set order (order * 1.5)
                              ]
                              [
                              let xlowind floor (20 * (xvalue - 0.5))
                              set order order * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis -
item xlowind xaxis )))
                              ]
                          ]
                      ]

                  set order_given (order_given + order)
                  set [incoming_demand] of choice_manufacturer ([incoming_demand] of
choice_manufacturer + order)
                  set lastpurchaseprice [price] of choice_manufacturer
                  set ([order_list] of choice_manufacturer) lput order ([order_list] of
choice_manufacturer)
                  set ([id_list] of choice_manufacturer) lput (turtle who) ([id_list] of
choice_manufacturer)
                  set loyal-man choice_manufacturer
                  setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
                  set manufacturer-list []
                  ]
                  [
                  set manufacturer-list remove (choice_manufacturer) manufacturer-list
                  set choice_manufacturer nobody
                  ]
              ]
              [
              set manufacturer-list remove (choice_manufacturer) manufacturer-list
              set choice_manufacturer nobody
              ]
          ]

      if choice_manufacturer = nobody
          [without-interruption
```

```
              [
              set choice_manufacturer one-of manufacturer
              if demandpricefunction = 1
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.25 1.25 1.24 1.225 1.210 1.19 1.16 1.135 1.095 1.060 1.000
0.905 0.835 0.775 0.75]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.75)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.25)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]
              if demandpricefunction = 2
                  [
                  let xaxis [0.5 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05 1.1
1.15 1.20]
                  let yaxis [1.500 1.490 1.475 1.455 1.415 1.370 1.305 1.240 1.180 1.100
1.000 0.875 0.690 0.565 0.500]
                  let xvalue ([price] of choice_manufacturer / indpriceforecast)
                  ifelse xvalue >= 1.2
                      [
                      set order (order * 0.5)
                      ]
                      [
                      ifelse xvalue < 0.5
                          [
                          set order (order * 1.5)
                          ]
                          [
                          let xlowind floor (20 * (xvalue - 0.5))
                          set order order * (item xlowind yaxis + (item (xlowind + 1) yaxis
- item xlowind yaxis) * ( (xvalue - item xlowind xaxis) / (item (xlowind + 1) xaxis - item
xlowind xaxis )))
                          ]
                      ]
                  ]

              set order_given (order_given + order)
              set [incoming_demand] of choice_manufacturer ([incoming_demand] of
choice_manufacturer + order)
              set lastpurchaseprice [price] of choice_manufacturer
              set ([order_list] of choice_manufacturer) lput order ([order_list] of
choice_manufacturer)
              set ([id_list] of choice_manufacturer) lput (turtle who) ([id_list] of
choice_manufacturer)
              set loyal-man choice_manufacturer
              setxy [xcor] of choice_manufacturer [ycor] of choice_manufacturer
              ]
          ]
      ]
      [
      set order_given (order_given + order)
      set [incoming_demand] of loyal-man ([incoming_demand] of loyal-man + order)
      set ([order_list] of loyal-man) lput order ([order_list] of loyal-man)
      set ([id_list] of loyal-man) lput (turtle who) ([id_list] of loyal-man)
      setxy [xcor] of loyal-man [ycor] of loyal-man
      ]
    ]
    ]
  ]
```

```
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to production

set inventory_position (inventory + production_placed - backlog)
set inventoryb ( inventory - backlog)

if order-style = 1
    [
    if orderup_level > inventory_position
        [
        without-interruption
            [
            if time >= 300
                [set ordercount ordercount + 1]
            set production_quantity (orderup_level - inventory_position)
            set production_placed (production_placed + production_quantity)
            set production_list lput production_quantity production_list
            set time_list lput time time_list
            ]
        ]
    ]

if order-style = 2
    [
    set orderup_level ((forecast * lead_time + SS) + 3 * forecast)
    if orderup_level > inventory_position
        [
        without-interruption
            [
            if time >= 300
                [set ordercount ordercount + 1]
            set production_quantity (orderup_level - inventory_position)
            set production_placed (production_placed + production_quantity)
            set production_list lput production_quantity production_list
            set time_list lput time time_list
            ]
        ]
    ]

if order-style = 3
    [
    set min_inventory (forecast * lead_time + SS)
    set orderup_level ((forecast * lead_time + SS) + 3 * forecast)
    if min_inventory > inventory_position
        [
        without-interruption
            [
            if time >= 300
                [set ordercount ordercount + 1]
            set production_quantity (orderup_level - inventory_position)
            set production_placed (production_placed + production_quantity)
            set production_list lput production_quantity production_list
            set time_list lput time time_list
            ]
        ]
    ]


if order-style = 4
    [
    set desired_inventory (5 * forecast)
    set desired_supplyline (lead_time * forecast)
    if   (desired_inventory   -   inventoryb)   /   inventory_AT   +   (desired_supplyline   -
production_placed) / supplyline_AT + expected_sales > 0
        [
        without-interruption
            [
            if time >= 300
                [set ordercount ordercount + 1]
            set  production_quantity  ((desired_inventory  -  inventoryb)  /  inventory_AT  +
(desired_supplyline - production_placed) / supplyline_AT + expected_sales)
            set production_placed (production_placed + production_quantity)
```

```
            set production_list lput production_quantity production_list
            set time_list lput time time_list
            ]
        ]
    ]

end

;-----------------------------------------------------------------------

to finish-production
while [not empty? production_list]
[ifelse (time - first time_list >= lead_time)
[without-interruption
 [set inventory (inventory + first production_list)
 set production_placed (production_placed - first production_list)
 set production_list but-first production_list
 set time_list but-first time_list]]
[stop]
]

end
to set-price

ifelse order-style = 4
  [
  set reference_inventory desired_inventory
  ]
  [
  set reference_inventory orderup_level
  ]

set lastprice price

if breed = retailer
    [
    if pricefunction = 1
        [
        let xaxis [0  0.1  0.2  0.3  0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6
1.7 1.8 1.9 2.0]
        let yaxis [1.3  1.297 1.291 1.282 1.255 1.200 1.123 1.069 1.033 1.015 1.000 0.982
0.952 0.916 0.871 0.811 0.754 0.727 0.712 0.709 0.700]
        let xvalue (inventory / reference_inventory)
        ifelse xvalue >= 2
            [
            set price ( price-normal-retailer * 0.7 )
            set price-modification-time time
            ]
            [
            let xlowind floor (10 * xvalue)
            set price price-normal-retailer * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * (xvalue * 10 - xlowind))
            set price-modification-time time
            ]
        ]

    if pricefunction = 2
        [
        let xaxis [0  0.1  0.2  0.3  0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6
1.7 1.8 1.9 2.0]
        let yaxis [1.3 1.282 1.246 1.177 1.093 1.045 1.024 1.018 1.009 1.003 1.000 0.997
0.991 0.982 0.964 0.931 0.886 0.793 0.733 0.709 0.700]
        let xvalue (inventory / reference_inventory)
        ifelse xvalue >= 2
            [
            set price ( price-normal-retailer * 0.7 )
            set price-modification-time time
            ]
            [
            let xlowind floor (10 * xvalue)
            set price price-normal-retailer * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * (xvalue * 10 - xlowind))
            set price-modification-time time
            ]
        ]
```

```
    ]

if breed = wholesaler
    [
    if pricefunction = 1
        [
        let xaxis [0  0.1  0.2  0.3  0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6
1.7 1.8 1.9 2.0]
        let yaxis [1.3  1.297 1.291 1.282 1.255 1.200 1.123 1.069 1.033 1.015 1.000 0.982
0.952 0.916 0.871 0.811 0.754 0.727 0.712 0.709 0.700]
        let xvalue (inventory / reference_inventory)
        ifelse xvalue >= 2
            [
            set price ( price-normal-wholesaler * 0.7 )
            set price-modification-time time
            ]
            [
            let xlowind floor (10 * xvalue)
            set price price-normal-wholesaler * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * (xvalue * 10 - xlowind))
            set price-modification-time time
            ]
        ]

    if pricefunction = 2
        [
        let xaxis [0  0.1  0.2  0.3  0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6
1.7 1.8 1.9 2.0]
        let yaxis [1.3 1.282 1.246 1.177 1.093 1.045 1.024 1.018 1.009 1.003 1.000 0.997
0.991 0.982 0.964 0.931 0.886 0.793 0.733 0.709 0.700]
        let xvalue (inventory / reference_inventory)
        ifelse xvalue >= 2
            [
            set price ( price-normal-wholesaler * 0.7 )
            set price-modification-time time
            ]
            [
            let xlowind floor (10 * xvalue)
            set price price-normal-wholesaler * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * (xvalue * 10 - xlowind))
            set price-modification-time time
            ]
        ]
    ]


if breed = manufacturer
    [
    if pricefunction = 1
        [
        let xaxis [0  0.1  0.2  0.3  0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6
1.7 1.8 1.9 2.0]
        let yaxis [1.3  1.297 1.291 1.282 1.255 1.200 1.123 1.069 1.033 1.015 1.000 0.982
0.952 0.916 0.871 0.811 0.754 0.727 0.712 0.709 0.700]
        let xvalue (inventory / reference_inventory)
        ifelse xvalue >= 2
            [
            set price ( price-normal-manufacturer * 0.7 )
            set price-modification-time time
            ]
            [
            let xlowind floor (10 * xvalue)
            set price price-normal-manufacturer * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * (xvalue * 10 - xlowind))
            set price-modification-time time
            ]
        ]

    if pricefunction = 2
        [
        let xaxis [0  0.1  0.2  0.3  0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6
1.7 1.8 1.9 2.0]
        let yaxis [1.3 1.282 1.246 1.177 1.093 1.045 1.024 1.018 1.009 1.003 1.000 0.997
0.991 0.982 0.964 0.931 0.886 0.793 0.733 0.709 0.700]
        let xvalue (inventory / reference_inventory)
```

```
        ifelse xvalue >= 2
            [
            set price ( price-normal-manufacturer * 0.7 )
            set price-modification-time time
            ]
            [
            let xlowind floor (10 * xvalue)
            set price price-normal-manufacturer * (item xlowind yaxis + (item (xlowind + 1)
yaxis - item xlowind yaxis) * (xvalue * 10 - xlowind))
            set price-modification-time time
            ]
        ]
   ]
set price ( 0.25 * lastprice + 0.75 * price )
end
      ;-----------------------------------------------------------------------------
```

# APPENDIX B: STOCK-FLOW DIAGRAMS AND EQUATIONS OF AGGREGATED SYSTEM DYNAMICS MODELS

The aggregated system dynamics model structure remains the same except the ordering equations, for different ordering policies. Therefore, the stock-flow diagram and equations of the system dynamics model in one of the ordering policies –actually (s,S) policy- are given below and only the ordering related equations are given for the remaining ones. The modifications under the adaptive pricing are shown in stock-flow diagram for the (s,S) policy, and the model equations are given for all policies.

## B.1. Aggregated System Dynamics Model Structure for Basic Policies

### B.1.1 Stock-Flow Diagram for (s,S) Policy

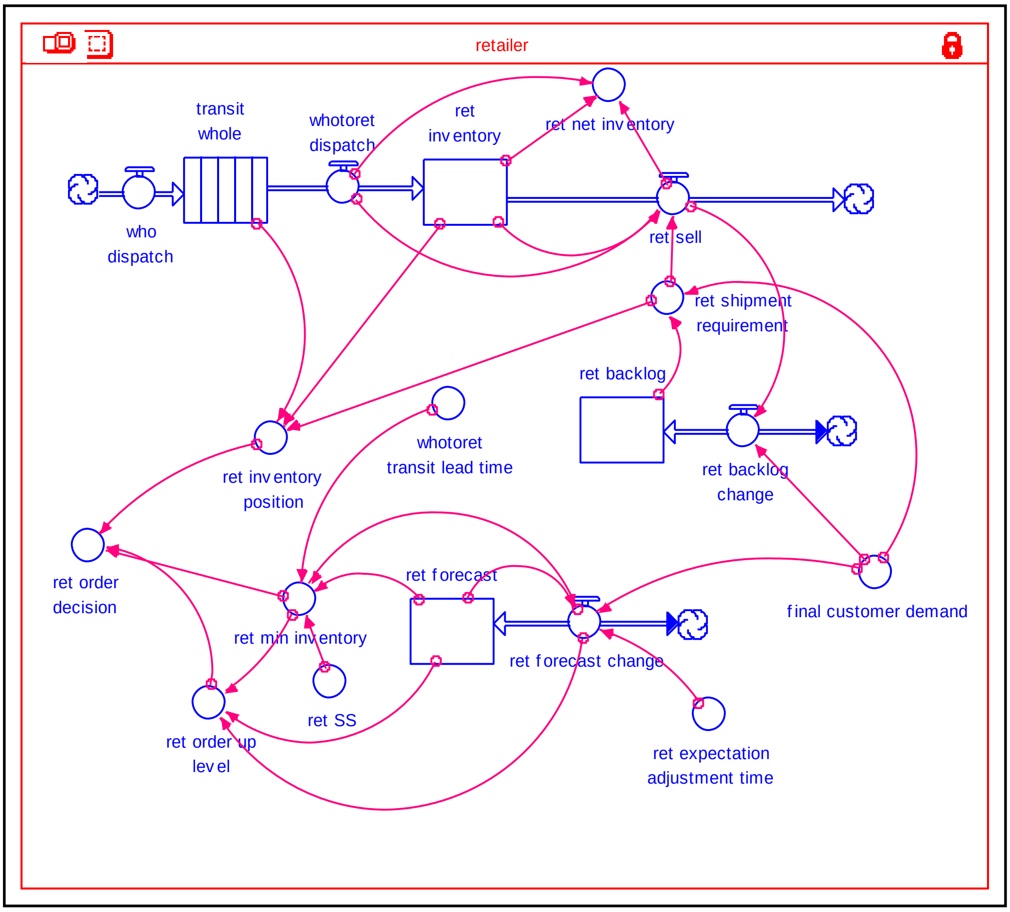


Figure B.1. Stock-Flow Diagram of the Aggregated System Dynamics Model Retailer Echelon for (s,S) Policy

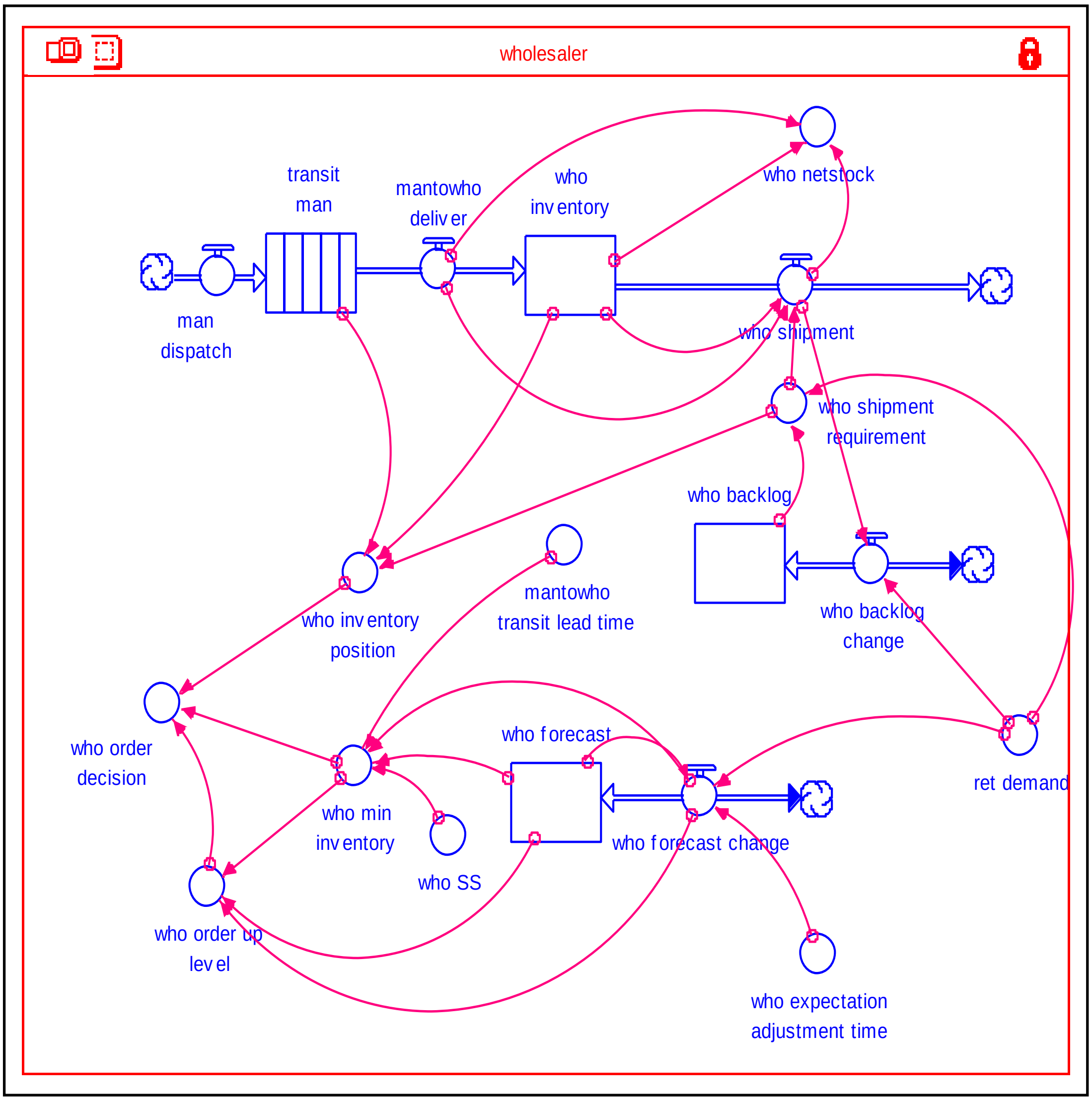


Figure B.2. Stock-Flow Diagram of the Aggregated System Dynamics Model Wholesaler Echelon for (s,S) Policy

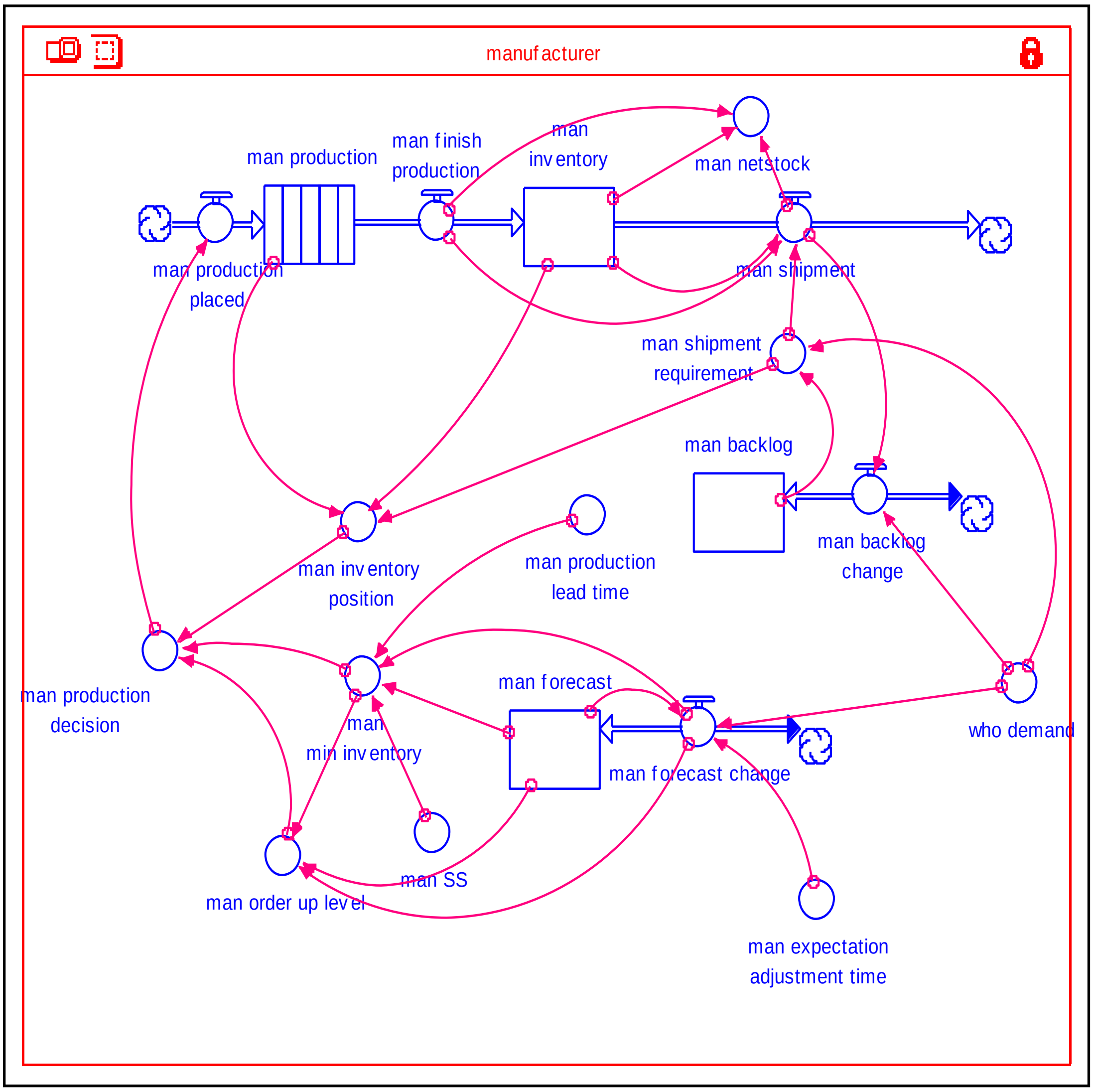


Figure B.3. Stock-Flow Diagram of the Aggregated System Dynamics Model Manufacturer Echelon for (s,S) Policy

### B.1.2 Model Equations for (s,S) Policy

*RETAILER*

ret_backlog(t) = ret_backlog(t - dt) + (ret_backlog_change) * dt
INIT ret_backlog = 0
INFLOWS:
ret_backlog_change = final_customer_demand-ret_sell
ret_forecast(t) = ret_forecast(t - dt) + (ret_forecast_change) * dt
INIT ret_forecast = 1250
INFLOWS:
ret_forecast_change = (final_customer_demand-ret_forecast)/ret_expectation_adjustment_time
ret_inventory(t) = ret_inventory(t - dt) + (whotoret_dispatch - ret_sell) * dt
INIT ret_inventory = 5000
INFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
OUTFLOWS:
ret_sell = min(ret_shipment_requirement,ret_inventory+whotoret_dispatch)
transit_whole(t) = transit_whole(t - dt) + (who_dispatch - whotoret_dispatch) * dt
INIT transit_whole = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_dispatch = who_shipment
OUTFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
final_customer_demand = NORMAL(2.5*500,sqrt(25*500/12))
ret_expectation_adjustment_time = 4
ret_inventory_position = ret_inventory+who_backlog+transit_whole-ret_shipment_requirement
ret_min_inventory = whotoret__transit_lead_time*(ret_forecast+ret_forecast_change*DT)+ret_SS
ret_net_inventory = ret_inventory +whotoret_dispatch-ret_sell
ret_order_decision = IF int(ret_inventory_position-ret_min_inventory) < 0 then (ret_order_up_level-ret_inventory_position)
else 0
ret_order_up_level = ret_min_inventory+(ret_forecast+ret_forecast_change*DT)*3
ret_shipment_requirement = ret_backlog+final_customer_demand
ret_SS = 2000
whotoret__transit_lead_time = 3

*WHOLESALER*

transit_man(t) = transit_man(t - dt) + (man_dispatch - mantowho_deliver) * dt
INIT transit_man = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_dispatch = man_shipment
OUTFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
who_backlog(t) = who_backlog(t - dt) + (who_backlog_change) * dt
INIT who_backlog = 0
INFLOWS:
who_backlog_change = ret_demand-who_shipment
who_forecast(t) = who_forecast(t - dt) + (who_forecast_change) * dt
INIT who_forecast = 1250

INFLOWS:
who_forecast_change = (ret_demand-who_forecast)/who_expectation_adjustment_time
who_inventory(t) = who_inventory(t - dt) + (mantowho_deliver - who_shipment) * dt
INIT who_inventory = 5000
INFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
OUTFLOWS:
who_shipment = min(who_inventory+mantowho_deliver,who_shipment_requirement)
mantowho_transit_lead_time = 3
ret_demand = ret_order_decision
who_expectation_adjustment_time = 4
who_inventory_position = who_inventory+man_backlog+transit_man-who_shipment_requirement
who_min_inventory = (who_forecast+who_forecast_change*DT)*mantowho_transit_lead_time+who_SS
who_netstock = who_inventory+mantowho_deliver-who_shipment
who_order_decision = IF who_inventory_position-who_min_inventory < 0 then (who_order_up_level-who_inventory_position)
else 0
who_order_up_level = who_min_inventory+(who_forecast+who_forecast_change*DT)*3
who_shipment_requirement = ret_demand+who_backlog
who_SS = 2000

*MANUFACTURER*

man_backlog(t) = man_backlog(t - dt) + (man_backlog_change) * dt
INIT man_backlog = 0
INFLOWS:
man_backlog_change = who_demand-man_shipment
man_forecast(t) = man_forecast(t - dt) + (man_forecast_change) * dt
INIT man_forecast = 1250
INFLOWS:
man_forecast_change = (who_demand-man_forecast)/man_expectation_adjustment_time
man_inventory(t) = man_inventory(t - dt) + (man_finish_production - man_shipment) * dt
INIT man_inventory = 5000
INFLOWS:
man_finish_production = CONVEYOR OUTFLOW
OUTFLOWS:
man_shipment = min(man_inventory+man_finish_production,man_shipment__requirement)
man_production(t) = man_production(t - dt) + (man_production_placed - man_finish_production) * dt
INIT man_production = 0
TRANSIT TIME = 5
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_production_placed = man_production_decision
OUTFLOWS:
man_finish_production = CONVEYOR OUTFLOW
man_expectation_adjustment_time = 4
man_inventory_position = man_inventory+man_production-man_shipment__requirement
man_netstock = man_inventory+man_finish_production-man_shipment
man_order_up_level = man__min_inventory+(man_forecast+man_forecast_change*DT)*3
man_production_decision = IF man_inventory_position-man__min_inventory < 0 then (man_order_up_level-man_inventory_position)
else 0
man_production_lead_time = 5
man_shipment__requirement = man_backlog+who_demand
man_SS = 2000
man__min_inventory = (man_forecast+man_forecast_change*DT)*man_production_lead_time+man_SS
who_demand = who_order_decision

### B.1.3 Ordering Equations for Fixed S Policy

*RETAILER*
ret_order_decision = IF (ret_inventory_position-ret_order_up_level) < 0 then (ret_order_up_level-ret_inventory_position) ELSE 0
ret_order_up_level = 120000

*WHOLESALER*
who_order_decision = IF who_inventory_position-who_order_up_level < 0 then (who_order_up_level-who_inventory_position) ELSE 0
who_order_up_level = 280000

*MANUFACTURER*
man_production_decision = IF man_inventory_position-man_order_up_level< 0 then (man_order_up_level-man_inventory_position) ELSE 0
man_order_up_level = 480000

### B.1.4 Ordering Equations for Variable S Policy

*RETAILER*
ret_order_decision = IF (ret_inventory_position-ret_order_up_level) < 0 then (ret_order_up_level-ret_inventory_position) ELSE 0
ret_order_up_level = (ret_forecast+ret_forecast_change*DT)*(3+who_to_ret_transit_lead_time)+ret_SS
ret_SS = 40000

*WHOLESALER*
who_order_decision = IF who_inventory_position-who_order_up_level < 0 then (who_order_up_level-who_inventory_position) ELSE 0
who_order_up_level =
(who_foreacst+who_forecast_change*DT)*(mantowho_transit_lead_time+3)+who_SS
who_SS = 40000

*MANUFACTURER*
man_production_decision = IF man_inventory_position-man_order_up_level< 0 then (man_order_up_level-man_inventory_position) ELSE 0
man_order_up_level = (man_forecast+man_forecast_change*DT)*(3+man_production_lead_time)+man_SS
man_SS = 40000

### B.1.5 Ordering Equations for A&A Policy

*RETAILER*
desired_ret_inventory = 5*ret_expected_demand
desired_ret_supply_line = who_to_ret_transit_lead_time*ret_expected_demand
ret_inventory_AT = 3
ret_order_decision = MAX((desired_ret_inventory- ret_inventoryb)/ret_inventory_AT+
(desired_ret_supply_line-ret_supply_line)/ret_supply_line_AT+DELAY1(ret_sell,4),0)
ret_supply_line = transit_whole+who_backlog
ret_supply_line_AT = 3

*WHOLESALER*
desired_who_inventory = 5*who_expected_demand
desired_who_supply_line = mantowho_transit_lead_time*who_expected_demand

who_inventory_AT = 3
who_order_decision = MAX((desired_who_supply_line-who_supply_line)/who_supply_line_AT+(desired_who_inventory-who_inventoryb)/who_inventory_AT+DELAY1(who_shipment,4),0)
who_supply_line = man_backlog+transit_man
who_supply_line_AT = 3

*MANUFACTURER*
desired_man_inventory = 5*man_expected_demand
desired_man_production_line = man_expected_demand*man_production_lead_time
man_inventory_AT = 3
man_production_decision = MAX((desired_man_production_line-man_production_line)/man_production_line_AT+(desired_man_inventory-man_inventoryb)/man_inventory_AT +DELAY1(man_shipment,4),0)
man_production_line = man_production
man_production_line_AT = 3

## B.2. Aggregated System Dynamics Model Structure under Adaptive Pricing

### B.2.1 Stock-Flow Diagram for (s,S) Policy

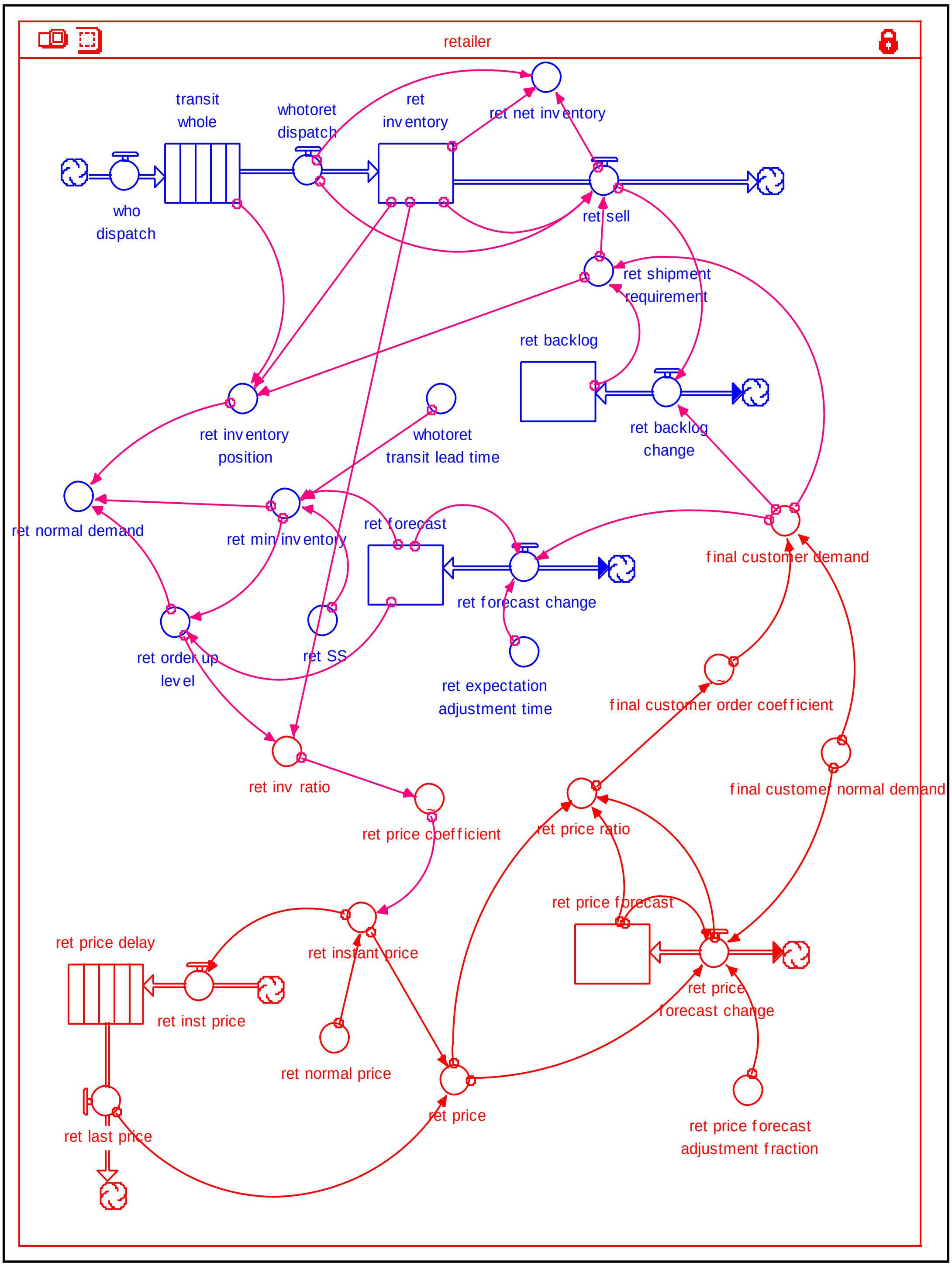


Figure B.4. Stock-Flow Diagram of the Aggregated System Dynamics Model Retailer Echelon under Adaptive Pricing for (s,S) Policy

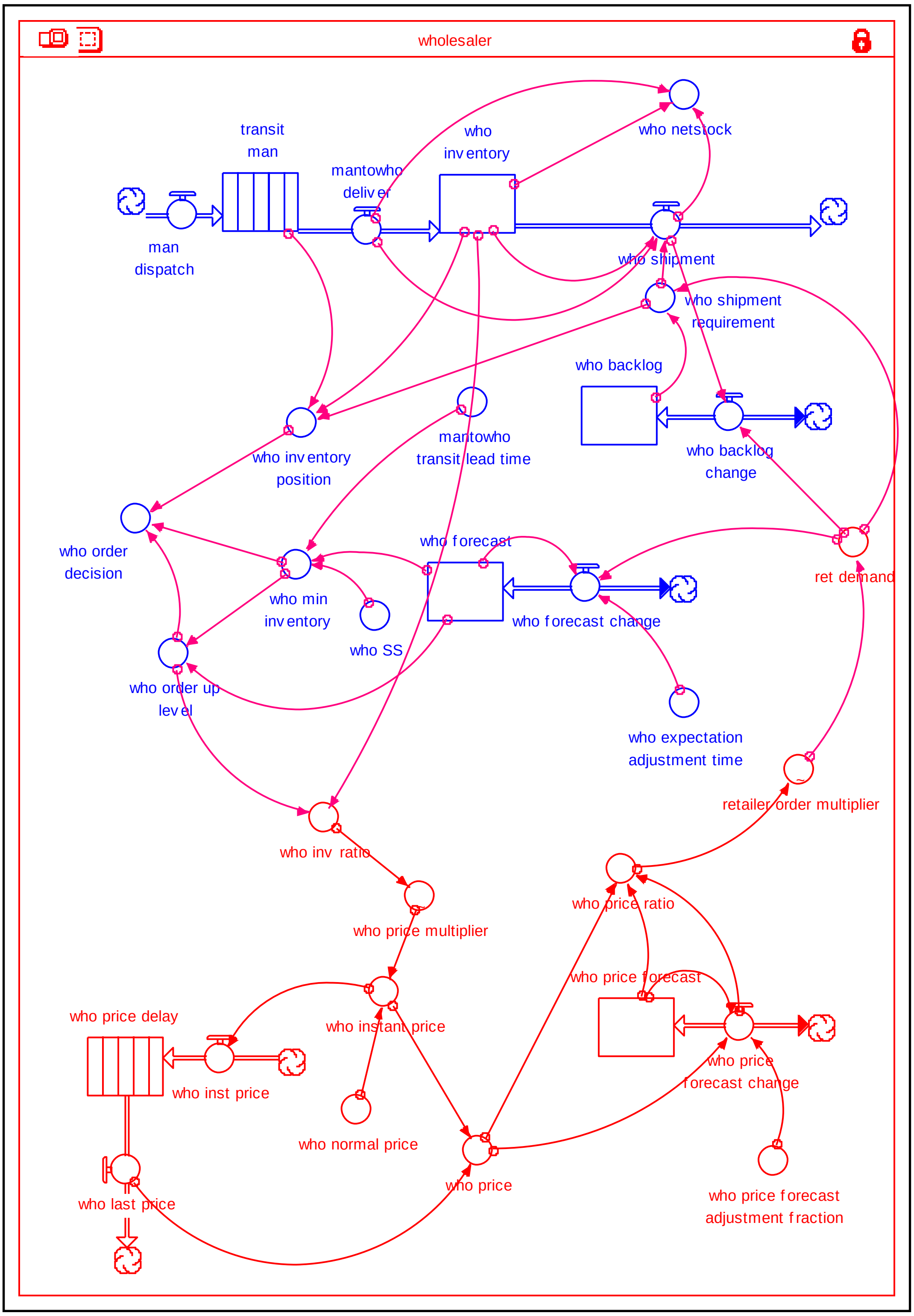


Figure B.5. Stock-Flow Diagram of the Aggregated System Dynamics Model Wholesaler Echelon under Adaptive Pricing for (s,S) Policy

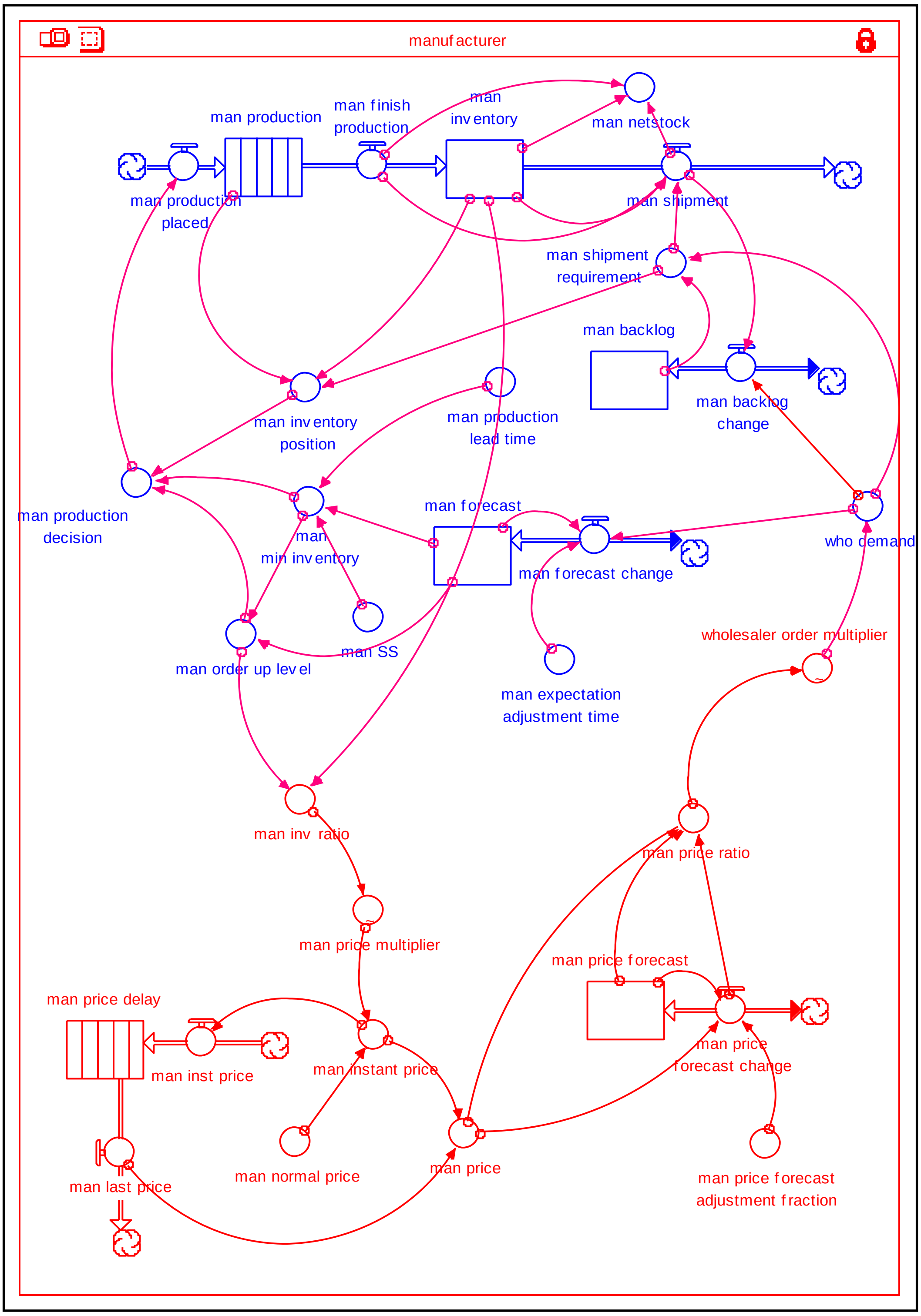


Figure B.6. Stock-Flow Diagram of the Aggregated System Dynamics Model Manufacturer Echelon under Adaptive Pricing for (s,S) Policy

## B.2.2 Model Equations for (s,S) Policy

*RETAILER*

ret_backlog(t) = ret_backlog(t - dt) + (ret_backlog_change) * dt
INIT ret_backlog = 0
INFLOWS:
ret_backlog_change = final_customer_demand-ret_sell
ret_forecast(t) = ret_forecast(t - dt) + (ret_forecast_change) * dt
INIT ret_forecast = 1250
INFLOWS:
ret_forecast_change = (final_customer_demand-ret_forecast)/ret_expectation_adjustment_time
ret_inventory(t) = ret_inventory(t - dt) + (whotoret_dispatch - ret_sell) * dt
INIT ret_inventory = 25000
INFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
OUTFLOWS:
ret_sell = min(ret_shipment_requirement,ret_inventory+whotoret_dispatch)
ret_price_delay(t) = ret_price_delay(t - dt) + (ret_inst_price - ret_last_price) * dt
INIT ret_price_delay = 2
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
ret_inst_price = ret_instant_price
OUTFLOWS:
ret_last_price = CONVEYOR OUTFLOW
ret_price_forecast(t) = ret_price_forecast(t - dt) + (ret_price__forecast_change) * dt
INIT ret_price_forecast = ret_normal_price
INFLOWS:
ret_price__forecast_change = if (final_customer_normal_demand>0.000000001) then ((ret_price-ret_price_forecast)*ret_price_forecast_adjustment_fraction)else(0)
transit_whole(t) = transit_whole(t - dt) + (who_dispatch - whotoret_dispatch) * dt
INIT transit_whole = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_dispatch = who_shipment
OUTFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
final_customer_demand = final_customer_normal_demand*final_customer_order_coefficient
final_customer_normal_demand = NORMAL(2.5*10000,sqrt(25*10000/12))
ret_expectation_adjustment_time = 4
ret_instant_price = ret_normal_price*ret_price_coefficient
ret_inventory_position = ret_inventory+who_backlog+transit_whole-ret_shipment_requirement
ret_inv_ratio = ret_inventory/ret_order_up_level
ret_min_inventory = whotoret__transit_lead_time*(ret_forecast)+ret_SS
ret_net_inventory = ret_inventory +whotoret_dispatch-ret_sell
ret_normal_demand = IF int(ret_inventory_position-ret_min_inventory) < 0 then (ret_order_up_level-ret_inventory_position)
else 0
ret_normal_price = 3
ret_order_up_level = ret_min_inventory+(ret_forecast)*3

ret_price = 0.75*ret_instant_price+0.25*ret_last_price
ret_price_forecast_adjustment_fraction = 0.25
ret_price_ratio = ret_price/(ret_price_forecast+ret_price__forecast_change*DT)
ret_shipment_requirement = ret_backlog+final_customer_demand
ret_SS = 40000
whotoret__transit_lead_time = 3
final_customer_order_coefficient = GRAPH(ret_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
ret_price_coefficient = GRAPH(ret_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*WHOLESALER*

transit_man(t) = transit_man(t - dt) + (man_dispatch - mantowho_deliver) * dt
INIT transit_man = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_dispatch = man_shipment
OUTFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
who_backlog(t) = who_backlog(t - dt) + (who_backlog_change) * dt
INIT who_backlog = 0
INFLOWS:
who_backlog_change = ret_demand-who_shipment
who_forecast(t) = who_forecast(t - dt) + (who_forecast_change) * dt
INIT who_forecast = 1250
INFLOWS:
who_forecast_change = (ret_demand-who_forecast)/who_expectation_adjustment_time
who_inventory(t) = who_inventory(t - dt) + (mantowho_deliver - who_shipment) * dt
INIT who_inventory = 25000
INFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
OUTFLOWS:
who_shipment = min(who_inventory+mantowho_deliver,who_shipment_requirement)
who_price_delay(t) = who_price_delay(t - dt) + (who_inst_price - who_last_price) * dt
INIT who_price_delay = 2
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_inst_price = who_instant_price
OUTFLOWS:
who_last_price = CONVEYOR OUTFLOW
who_price_forecast(t) = who_price_forecast(t - dt) + (who_price__forecast_change) * dt
INIT who_price_forecast = who_normal_price
INFLOWS:
who_price__forecast_change = if (ret_normal_demand>0.000000001) then ((who_price-who_price_forecast)*who_price_forecast_adjustment_fraction)else(0)
mantowho_transit_lead_time = 3
ret_demand = ret_normal_demand*retailer_order_multiplier
who_expectation_adjustment_time = 4
who_instant_price = who_normal_price*who_price_multiplier

who_inventory_position = who_inventory+man_backlog+transit_man-who_shipment_requirement
who_inv_ratio = who_inventory/who_order_up_level
who_min_inventory = (who_forecast)*mantowho_transit_lead_time+who_SS
who_netstock = who_inventory+mantowho_deliver-who_shipment
who_normal_price = 2
who_order_decision = IF who_inventory_position-who_min_inventory < 0 then (who_order_up_level-who_inventory_position)
else 0
who_order_up_level = who_min_inventory+(who_forecast)*3
who_price = 0.75*who_instant_price+0.25*who_last_price
who_price_forecast_adjustment_fraction = 0.25
who_price_ratio = who_price/(who_price_forecast+who_price__forecast_change*DT)
who_shipment_requirement = ret_demand+who_backlog
who_SS = 40000
retailer_order_multiplier = GRAPH(who_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
who_price_multiplier = GRAPH(who_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*MANUFACTURER*

man_backlog(t) = man_backlog(t - dt) + (man_backlog_change) * dt
INIT man_backlog = 0
INFLOWS:
man_backlog_change = who_demand-man_shipment
man_forecast(t) = man_forecast(t - dt) + (man_forecast_change) * dt
INIT man_forecast = 1250
INFLOWS:
man_forecast_change = (who_demand-man_forecast)/man_expectation_adjustment_time
man_inventory(t) = man_inventory(t - dt) + (man_finish_production - man_shipment) * dt
INIT man_inventory = 25000
INFLOWS:
man_finish_production = CONVEYOR OUTFLOW
OUTFLOWS:
man_shipment = min(man_inventory+man_finish_production,man_shipment__requirement)
man_price_delay(t) = man_price_delay(t - dt) + (man_inst_price - man_last_price) * dt
INIT man_price_delay = 1
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_inst_price = man_instant_price
OUTFLOWS:
man_last_price = CONVEYOR OUTFLOW
man_price_forecast(t) = man_price_forecast(t - dt) + (man_price__forecast_change) * dt
INIT man_price_forecast = man_normal_price
INFLOWS:
man_price__forecast_change = if (who_order_decision>0.000000001) then ((man_price-man_price_forecast)*man_price_forecast_adjustment_fraction)else(0)
man_production(t) = man_production(t - dt) + (man_production_placed - man_finish_production) * dt
INIT man_production = 0
TRANSIT TIME = 5
INFLOW LIMIT = INF
CAPACITY = INF

INFLOWS:
man_production_placed = man_production_decision
OUTFLOWS:
man_finish_production = CONVEYOR OUTFLOW
man_expectation_adjustment_time = 4
man_instant_price = man_normal_price*man_price_multiplier
man_inventory_position = man_inventory+man_production-man_shipment__requirement
man_inv_ratio = man_inventory/man_order_up_level
man_netstock = man_inventory+man_finish_production-man_shipment
man_normal_price = 1
man_order_up_level = man__min_inventory+(man_forecast)*3
man_price = 0.75*man_instant_price+0.25*man_last_price
man_price_forecast_adjustment_fraction = 0.25
man_price_ratio = man_price/(man_price_forecast+man_price__forecast_change*DT)
man_production_decision = IF man_inventory_position-man__min_inventory < 0 then
(man_order_up_level-man_inventory_position)
else 0
man_production_lead_time = 5
man_shipment__requirement = man_backlog+who_demand
man_SS = 2000
man__min_inventory = (man_forecast)*man_production_lead_time+man_SS
who_demand = who_order_decision*wholesaler_order_multiplier
man_price_multiplier = GRAPH(man_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)
wholesaler_order_multiplier = GRAPH(man_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)

## B.2.3 Model Equations for Fixed S Policy

*RETAILER*

ret_backlog(t) = ret_backlog(t - dt) + (ret_backlog_change) * dt
INIT ret_backlog = 0
INFLOWS:
ret_backlog_change = final_customer_demand-ret_sell
ret_inventory(t) = ret_inventory(t - dt) + (whotoret_dispatch - ret_sell) * dt
INIT ret_inventory = 25000
INFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
OUTFLOWS:
ret_sell = min(ret_shipment_requirement,ret_inventory+whotoret_dispatch)
ret_price_delay(t) = ret_price_delay(t - dt) + (ret_inst_price - ret_last_price) * dt
INIT ret_price_delay = 2
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
ret_inst_price = ret_instant_price
OUTFLOWS:
ret_last_price = CONVEYOR OUTFLOW
ret_price_forecast(t) = ret_price_forecast(t - dt) + (ret_price__forecast_change) * dt
INIT ret_price_forecast = ret_normal_price

INFLOWS:
ret_price__forecast_change = if (final_customer_normal_demand>0.000000001) then ((ret_price-ret_price_forecast)*ret_price_forecast_adjustment_fraction)else(0)
transit_whole(t) = transit_whole(t - dt) + (who_dispatch - whotoret_dispatch) * dt
INIT transit_whole = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_dispatch = who_shipment
OUTFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
final_customer_demand = final_customer_normal_demand*final_customer_order_multiplier
final_customer_normal_demand = NORMAL(2.5*10000,sqrt(25*10000/12))
ret_instant_price = ret_normal_price*ret_price_multiplier
ret_inventory_position = ret_inventory+who_backlog+transit_whole-ret_shipment_requirement
ret_inv_ratio = ret_inventory/ret_order_up_level
ret_netstock = ret_inventory +whotoret_dispatch-ret_sell
ret_normal_price = 3
ret_order_decision = IF int(ret_inventory_position-ret_order_up_level) < 0 then (ret_order_up_level-ret_inventory_position)
else 0
ret_order_up_level = 120000
ret_price = 0.75*ret_instant_price+0.25*ret_last_price
ret_price_forecast_adjustment_fraction = 0.25
ret_price_ratio = ret_price/(ret_price_forecast+ret_price__forecast_change*DT)
ret_shipment_requirement = ret_backlog+final_customer_demand
final_customer_order_multiplier = GRAPH(ret_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
ret_price_multiplier = GRAPH(ret_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*WHOLESALER*

transit_man(t) = transit_man(t - dt) + (man_dispatch - mantowho_deliver) * dt
INIT transit_man = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_dispatch = man_shipment
OUTFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
who_backlog(t) = who_backlog(t - dt) + (who_backlog_change) * dt
INIT who_backlog = 0
INFLOWS:
who_backlog_change = ret_demand-who_shipment
who_inventory(t) = who_inventory(t - dt) + (mantowho_deliver - who_shipment) * dt
INIT who_inventory = 25000
INFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
OUTFLOWS:
who_shipment = min(who_inventory+mantowho_deliver,who_shipment_requirement)
who_price_delay(t) = who_price_delay(t - dt) + (who_inst_price - who_last_price) * dt

INIT who_price_delay = 2
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_inst_price = who_instant_price
OUTFLOWS:
who_last_price = CONVEYOR OUTFLOW
who_price_forecast(t) = who_price_forecast(t - dt) + (who_price__forecast_change) * dt
INIT who_price_forecast = who_normal_price
INFLOWS:
who_price__forecast_change = if (ret_order_decision>0.000000001) then ((who_price-who_price_forecast)*who_price_forecast_adjustment_fraction)else(0)
ret_demand = ret_order_decision*retailer_order_multiplier
who_instant_price = who_normal_price*who_price_multiplier
who_inventory_position = who_inventory+man_backlog+transit_man-who_shipment_requirement
who_inv_ratio = who_inventory/who_order_up_level
who_netstock = who_inventory+mantowho_deliver-who_shipment
who_normal_price = 2
who_order_decision = IF who_inventory_position-who_order_up_level < 0 then (who_order_up_level-who_inventory_position)
else 0
who_order_up_level = 280000
who_price = 0.75*who_instant_price+0.25*who_last_price
who_price_forecast_adjustment_fraction = 0.25
who_price_ratio = who_price/(who_price_forecast+who_price__forecast_change*DT)
who_shipment_requirement = ret_demand+who_backlog
retailer_order_multiplier = GRAPH(who_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
who_price_multiplier = GRAPH(who_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*MANUFACTURER*

man_backlog(t) = man_backlog(t - dt) + (man_backlog_change) * dt
INIT man_backlog = 0
INFLOWS:
man_backlog_change = who_demand-man_shipment
man_inventory(t) = man_inventory(t - dt) + (man_finish_production - man_shipment) * dt
INIT man_inventory = 25000
INFLOWS:
man_finish_production = CONVEYOR OUTFLOW
OUTFLOWS:
man_shipment = min(man_inventory+man_finish_production,man_shipment__requirement)
man_price_delay(t) = man_price_delay(t - dt) + (man_inst_price - man_last_price) * dt
INIT man_price_delay = 1
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_inst_price = man_instant_price
OUTFLOWS:
man_last_price = CONVEYOR OUTFLOW

man_price_forecast(t) = man_price_forecast(t - dt) + (man_price__forecast_change) * dt
INIT man_price_forecast = man_normal_price
INFLOWS:
man_price__forecast_change = if (who_order_decision>0.000000001) then ((man_price-man_price_forecast)*man_price_forecast_adjustment_fraction)else(0)
man_production(t) = man_production(t - dt) + (man_production_placed - man_finish_production) * dt
INIT man_production = 0
TRANSIT TIME = 5
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_production_placed = man_production_decision
OUTFLOWS:
man_finish_production = CONVEYOR OUTFLOW
man_instant_price = man_normal_price*man_price_multiplier
man_inventory_position = man_inventory+man_production-man_shipment__requirement
man_inv_ratio = man_inventory/man_order_up_level
man_netstock = man_inventory+man_finish_production-man_shipment
man_normal_price = 1
man_order_up_level = 480000
man_price = 0.75*man_instant_price+0.25*man_last_price
man_price_forecast_adjustment_fraction = 0.25
man_price_ratio = man_price/(man_price_forecast+man_price__forecast_change*DT)
man_production_decision = IF man_inventory_position-man_order_up_level< 0 then (man_order_up_level-man_inventory_position)
else 0
man_shipment__requirement = man_backlog+who_demand
who_demand = who_order_decision*wholesaler_order_multiplier
man_price_multiplier = GRAPH(man_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)
wholesaler_order_multiplier = GRAPH(man_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)

### B.2.4 Model Equations for Variable S Policy

*RETAILER*

ret_backlog(t) = ret_backlog(t - dt) + (ret_backlog_change) * dt
INIT ret_backlog = 0
INFLOWS:
ret_backlog_change = final_customer_demand-ret_sell
ret_forecast(t) = ret_forecast(t - dt) + (ret_forecast_change) * dt
INIT ret_forecast = 1250
INFLOWS:
ret_forecast_change = (final_customer_demand-ret_forecast)/ret_expectation_adjustment_time
ret_inventory(t) = ret_inventory(t - dt) + (whotoret_dispatch - ret_sell) * dt
INIT ret_inventory = 25000
INFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
OUTFLOWS:
ret_sell = min(ret_shipment_requirement,ret_inventory+whotoret_dispatch)
ret_price_delay(t) = ret_price_delay(t - dt) + (ret_inst_price - ret_last_price) * dt

INIT ret_price_delay = 2
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
ret_inst_price = ret_instant_price
OUTFLOWS:
ret_last_price = CONVEYOR OUTFLOW
ret_price_forecast(t) = ret_price_forecast(t - dt) + (ret_price__forecast_change) * dt
INIT ret_price_forecast = ret_normal_price
INFLOWS:
ret_price__forecast_change = if (final_customer_normal_demand>0.000000001) then ((ret_price-ret_price_forecast)*ret_price_forecast_adjustment_fraction)else(0)
transit_whole(t) = transit_whole(t - dt) + (who_dispatch - whotoret_dispatch) * dt
INIT transit_whole = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_dispatch = who_shipment
OUTFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
final_customer_demand = final_customer_normal_demand*final_customer_order_multiplier
final_customer_normal_demand = NORMAL(2.5*10000,sqrt(25*10000/12))
ret_expectation_adjustment_time = 4
ret_instant_price = ret_normal_price*ret_price_multiplier
ret_inventory_position = ret_inventory+who_backlog+transit_whole-ret_shipment_requirement
ret_inv_ratio = ret_inventory/ret_order_up_level
ret_netstock = ret_inventory +whotoret_dispatch-ret_sell
ret_normal_price = 3
ret_order_decision = IF int(ret_inventory_position-ret_order_up_level) < 0 then (ret_order_up_level-ret_inventory_position)
else 0
ret_order_up_level = (ret_forecast)*(3+who_to_ret_transit_lead_time)+ret_SS
ret_price = 0.75*ret_instant_price+0.25*ret_last_price
ret_price_forecast_adjustment_fraction = 0.25
ret_price_ratio = ret_price/(ret_price_forecast+ret_price__forecast_change*DT)
ret_shipment_requirement = ret_backlog+final_customer_demand
ret_SS = 40000
who_to_ret_transit_lead_time = 3
final_customer_order_multiplier = GRAPH(ret_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
ret_price_multiplier = GRAPH(ret_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*WHOLESALER*

transit_man(t) = transit_man(t - dt) + (man_dispatch - mantowho_deliver) * dt
INIT transit_man = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_dispatch = man_shipment

OUTFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
who_backlog(t) = who_backlog(t - dt) + (who_backlog_change) * dt
INIT who_backlog = 0
INFLOWS:
who_backlog_change = ret_demand-who_shipment
who_foreacst(t) = who_foreacst(t - dt) + (who_forecast_change) * dt
INIT who_foreacst = 1250
INFLOWS:
who_forecast_change = (ret_demand-who_foreacst)/who_expectation_adjustment_time
who_inventory(t) = who_inventory(t - dt) + (mantowho_deliver - who_shipment) * dt
INIT who_inventory = 25000
INFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
OUTFLOWS:
who_shipment = min(who_inventory+mantowho_deliver,who_shipment_requirement)
who_price_delay(t) = who_price_delay(t - dt) + (who_inst_price - who_last_price) * dt
INIT who_price_delay = 2
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_inst_price = who_instant_price
OUTFLOWS:
who_last_price = CONVEYOR OUTFLOW
who_price_forecast(t) = who_price_forecast(t - dt) + (who_price__forecast_change) * dt
INIT who_price_forecast = who_normal_price
INFLOWS:
who_price__forecast_change = if (ret_order_decision>0.000000001) then ((who_price-who_price_forecast)*who_price_forecast_adjustment_fraction)else(0)
mantowho_transit_lead_time = 3
ret_demand = ret_order_decision*retailer_order_multiplier
who_expectation_adjustment_time = 4
who_instant_price = who_normal_price*who_price_multiplier
who_inventory_position = who_inventory+man_backlog+transit_man-who_shipment_requirement
who_inv_ratio = who_inventory/who_order_up_level
who_netstock = who_inventory+mantowho_deliver-who_shipment
who_normal_order = IF who_inventory_position-who_order_up_level < 0 then (who_order_up_level-who_inventory_position)
else 0
who_normal_price = 2
who_order_up_level = (who_foreacst)*(mantowho_transit_lead_time+3)+who_SS
who_price = 0.75*who_instant_price+0.25*who_last_price
who_price_forecast_adjustment_fraction = 0.25
who_price_ratio = who_price/(who_price_forecast+who_price__forecast_change*DT)
who_shipment_requirement = ret_demand+who_backlog
who_SS = 40000
retailer_order_multiplier = GRAPH(who_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
who_price_multiplier = GRAPH(who_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*MANUFACTURER*

man_backlog(t) = man_backlog(t - dt) + (man_backlog_change) * dt
man_backlog(t) = man_backlog(t - dt) + (man_backlog_change) * dt
INIT man_backlog = 0
INFLOWS:
man_backlog_change = who_demand-man_shipment
man_forecast(t) = man_forecast(t - dt) + (man_forecast_change) * dt
INIT man_forecast = 1250
INFLOWS:
man_forecast_change = (who_demand-man_forecast)/man_expectation_adjustment_time
man_inventory(t) = man_inventory(t - dt) + (man_finish_production - man_shipment) * dt
INIT man_inventory = 25000
INFLOWS:
man_finish_production = CONVEYOR OUTFLOW
OUTFLOWS:
man_shipment = min(man_inventory+man_finish_production,man_shipment__requirement)
man_price_delay(t) = man_price_delay(t - dt) + (man_inst_price - man_last_price) * dt
INIT man_price_delay = 1
TRANSIT TIME = 1
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_inst_price = man_instant_price
OUTFLOWS:
man_last_price = CONVEYOR OUTFLOW
man_price_forecast(t) = man_price_forecast(t - dt) + (man_price__forecast_change) * dt
INIT man_price_forecast = man_normal_price
INFLOWS:
man_price__forecast_change = if (who_normal_order>0.000000001) then ((man_price-man_price_forecast)*man_price_forecast_adjustment_fraction)else(0)
man_production(t) = man_production(t - dt) + (man_production_placed - man_finish_production) * dt
INIT man_production = 0
TRANSIT TIME = 5
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_production_placed = man_production_decision
OUTFLOWS:
man_finish_production = CONVEYOR OUTFLOW
man_expectation_adjustment_time = 4
man_instant_price = man_normal_price*man_price_multiplier
man_inventory_position = man_inventory+man_production-man_shipment__requirement
man_inv_ratio = man_inventory/man_order_up_level
man_netstock = man_inventory+man_finish_production-man_shipment
man_normal_price = 1
man_order_up_level = (man_forecast)*(3+man_production_lead_time)+man_SS
man_price = 0.75*man_instant_price+0.25*man_last_price
man_price_forecast_adjustment_fraction = 0.25
man_price_ratio = man_price/(man_price_forecast+man_price__forecast_change*DT)
man_production_decision = IF man_inventory_position-man_order_up_level< 0 then (man_order_up_level-man_inventory_position)
else 0
man_production_lead_time = 5
man_shipment__requirement = man_backlog+who_demand
man_SS = 40000
who_demand = who_normal_order*wholesaler_order_multiplier
man_price_multiplier = GRAPH(man_inv_ratio)

(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)
wholesaler_order_multiplier = GRAPH(man_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)

### B.2.5 Model Equations for A&A Policy

*RETAILER*

ret_backlog(t) = ret_backlog(t - dt) + (ret_backlog_change) * dt
INIT ret_backlog = 0
INFLOWS:
ret_backlog_change = customer_demand-ret_sell
ret_forecast(t) = ret_forecast(t - dt) + (ret_forecast_change) * dt
INIT ret_forecast = 1250
INFLOWS:
ret_forecast_change = (customer_demand-ret_forecast)/ret_expectation_adjustment_time
ret_inventory(t) = ret_inventory(t - dt) + (whotoret_dispatch - ret_sell) * dt
INIT ret_inventory = 25000
INFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
OUTFLOWS:
ret_sell = min(ret_shipment_requirement,ret_inventory+whotoret_dispatch)
ret_price_forecast(t) = ret_price_forecast(t - dt) + (ret_price__forecast_change) * dt
INIT ret_price_forecast = ret_normal_price
INFLOWS:
ret_price__forecast_change = (ret_price-ret_price_forecast)/ret_price_forecast_adjustment_time
transit_whole(t) = transit_whole(t - dt) + (who_dispatch - whotoret_dispatch) * dt
INIT transit_whole = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
who_dispatch = who_shipment
OUTFLOWS:
whotoret_dispatch = CONVEYOR OUTFLOW
customer_demand = customer_normal_demand*customer_order_multiplier
customer_normal_demand = NORMAL(2.5*10000,sqrt(25*10000/12))
desired_ret_inventory = 5*ret_expected_demand
desired_ret_supply_line = who_to_ret_transit_lead_time*ret_expected_demand
ret_expectation_adjustment_time = 4
ret_expected_demand = ret_forecast
ret_inventoryb = ret_inventory-ret_shipment_requirement
ret_inventory_AT = 3
ret_inv_ratio = ret_inventory/desired_ret_inventory
ret_netstock = ret_inventory +whotoret_dispatch-ret_sell
ret_normal_price = 3
ret_order_decision = MAX((desired_ret_inventory-ret_inventoryb)/ret_inventory_AT+(desired_ret_supply_line-ret_supply_line)/ret_supply_line_AT+DELAY1(ret_sell,4),0)
ret_price = ret_normal_price*ret_price_multiplier
ret_price_forecast_adjustment_time = 4
ret_price_ratio = ret_price/ret_price_forecast

ret_shipment_requirement = ret_backlog+customer_demand
ret_supply_line = transit_whole+who_backlog
ret_supply_line_AT = 3
who_to_ret_transit_lead_time = 3
customer_order_multiplier = GRAPH(ret_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
ret_price_multiplier = GRAPH(ret_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*WHOLESALER*

transit_man(t) = transit_man(t - dt) + (man_dispatch - mantowho_deliver) * dt
INIT transit_man = 0
TRANSIT TIME = 3
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_dispatch = man_shipment
OUTFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
who_backlog(t) = who_backlog(t - dt) + (who_backlog_change) * dt
INIT who_backlog = 0
INFLOWS:
who_backlog_change = ret_demand-who_shipment
who_forecast(t) = who_forecast(t - dt) + (who_forecast_change) * dt
INIT who_forecast = 1250
INFLOWS:
who_forecast_change = (ret_demand-who_forecast)/who_expectation_adjustment_time
who_inventory(t) = who_inventory(t - dt) + (mantowho_deliver - who_shipment) * dt
INIT who_inventory = 25000
INFLOWS:
mantowho_deliver = CONVEYOR OUTFLOW
OUTFLOWS:
who_shipment = min(who_inventory+mantowho_deliver,who_shipment_requirement)
who_price_forecast(t) = who_price_forecast(t - dt) + (who_price__forecast_change) * dt
INIT who_price_forecast = who_normal_price
INFLOWS:
who_price__forecast_change = (who_price-who_price_forecast)/who_price_forecast_adjustment_time
desired_who_inventory = 5*who_expected_demand
desired_who_supply_line = mantowho_transit_lead_time*who_expected_demand
mantowho_transit_lead_time = 3
ret_demand = ret_order_decision*retailer_order_multiplier
who_expectation_adjustment_time = 4
who_expected_demand = who_forecast
who_inventoryb = who_inventory-who_shipment_requirement
who_inventory_AT = 3
who_inv_ratio = who_inventory/desired_who_inventory
who_netstock = who_inventory+mantowho_deliver-who_shipment
who_normal_price = 2
who_order_decision = MAX((desired_who_supply_line-who_supply_line)/who_supply_line_AT+(desired_who_inventory-who_inventoryb)/who_inventory_AT+DELAY1(who_shipment,4),0)
who_price = who_normal_price*who_price_multiplier
who_price_forecast_adjustment_time = 4

who_price_ratio = who_price/who_price_forecast
who_shipment_requirement = ret_demand+who_backlog
who_supply_line = man_backlog+transit_man
who_supply_line_AT = 3
retailer_order_multiplier = GRAPH(who_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)
who_price_multiplier = GRAPH(who_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)

*MANUFACTURER*

man_backlog(t) = man_backlog(t - dt) + (man_backlog_change) * dt
INIT man_backlog = 0
INFLOWS:
man_backlog_change = who_demand-man_shipment
man_forecast(t) = man_forecast(t - dt) + (man_forecast_change) * dt
INIT man_forecast = 1250
INFLOWS:
man_forecast_change = (who_demand-man_forecast)/man_expectation_adjustment_time
man_inventory(t) = man_inventory(t - dt) + (man_finish_production - man_shipment) * dt
INIT man_inventory = 25000
INFLOWS:
man_finish_production = CONVEYOR OUTFLOW
OUTFLOWS:
man_shipment = min(man_inventory+man_finish_production,man_shipment__requirement)
man_price_forecast(t) = man_price_forecast(t - dt) + (man_price__forecast_change) * dt
INIT man_price_forecast = who_normal_price
INFLOWS:
man_price__forecast_change = (man_price-man_price_forecast)/man_price_forecast_adjustment_time
man_production(t) = man_production(t - dt) + (man_production_placed - man_finish_production) * dt
INIT man_production = 0
TRANSIT TIME = 5
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
man_production_placed = man_production_decision
OUTFLOWS:
man_finish_production = CONVEYOR OUTFLOW
desired_man_inventory = 5*man_expected_demand
desired_man_production_line = man_expected_demand*man_production_lead_time
man_expectation_adjustment_time = 4
man_expected_demand = man_forecast
man_inventoryb = man_inventory-man_shipment__requirement
man_inventory_AT = 3
man_inv_ratio = man_inventory/desired_man_inventory
man_netstock = man_inventory+man_finish_production-man_shipment
man_normal_price = 1
man_price = man_normal_price*man_price_multiplier
man_price_forecast_adjustment_time = 4
man_price_ratio = man_price/man_price_forecast
man_production_decision = MAX((desired_man_production_line-man_production_line)/man_production_line_AT+(desired_man_inventory-man_inventoryb)/man_inventory_AT+DELAY1(man_shipment,4),0)

man_production_lead_time = 5
man_production_line = man_production
man_production_line_AT = 3
man_shipment__requirement = man_backlog+who_demand
who_demand = who_order_decision*wholesaler_order_multiplier
man_price_multiplier = GRAPH(man_inv_ratio)
(0.00, 1.30), (0.1, 1.29), (0.2, 1.27), (0.3, 1.23), (0.4, 1.17), (0.5, 1.12), (0.6, 1.07), (0.7, 1.04), (0.8, 1.02), (0.9, 1.01), (1, 1.00), (1.10, 0.99), (1.20, 0.972), (1.30, 0.949), (1.40, 0.917), (1.50, 0.871), (1.60, 0.82), (1.70, 0.76), (1.80, 0.723), (1.90, 0.709), (2.00, 0.7)
wholesaler_order_multiplier = GRAPH(man_price_ratio)
(0.5, 1.38), (0.55, 1.37), (0.6, 1.36), (0.65, 1.34), (0.7, 1.31), (0.75, 1.28), (0.8, 1.23), (0.85, 1.19), (0.9, 1.14), (0.95, 1.08), (1.00, 1.00), (1.05, 0.89), (1.10, 0.762), (1.15, 0.67), (1.20, 0.625)

# APPENDIX C: SPECTRAL ANALYSIS OF ARTIFICIAL DATA SETS

It is common practice to illustrate the use of a tool by applying it on a data set characteristics of which is already known. Fourier spectral analysis is applied to artificial data sets in this section and Table C.1 summarizes the spectral analysis results.

## C.1. Spectral Analysis of a Sine Wave

Periodicity analysis is applied to a sine wave:

$$X(t) = A * \sin(\frac{2\pi t}{T} + \phi) \tag{C.1}$$

A, T, and φ are amplitude, period, and phase angle respectively. A, T, and φ are set as 100, 10, and 0 respectively. The time series is shown in C.1. Fourier power spectral density is given in Figure C.2. As seen in Figure C.2, the spectral power is concentrated around f=0.1 as expected. Although we expected the whole spectral power to be concentrated at the f=0.1, Table C.1 shows that there exists some power in non-dominant frequencies. This is due to Tukey-Hanning estimate, which provides smoothness of PSD at the expense of spectral leakage at some degree. Table C.1 also shows that there is a single dominant periodicity for sine wave and it is at period 10, as expected.

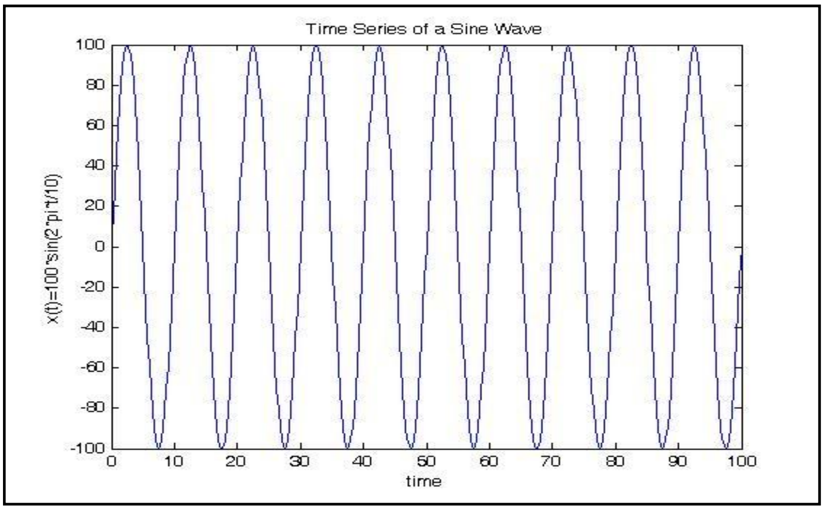


Figure C.1. Time Series of a Sine Wave

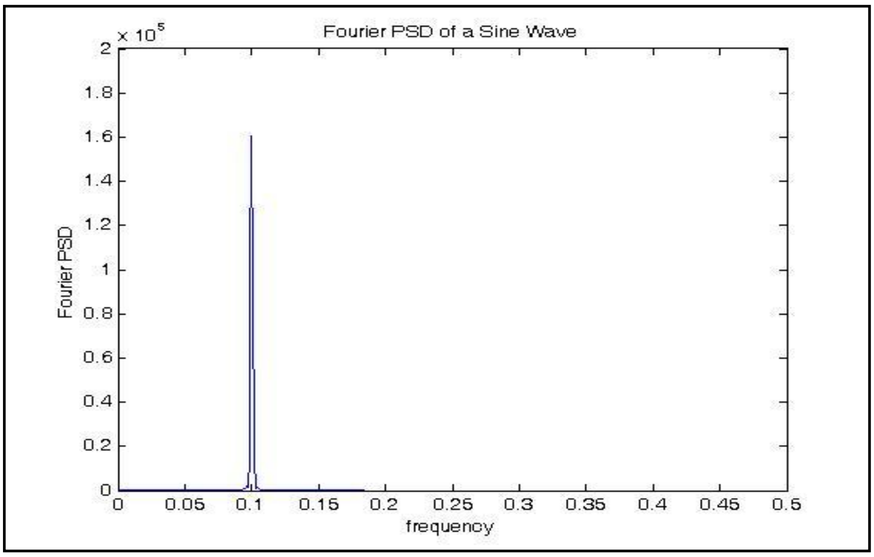


Figure C.2. Fourier PSD of a Sine Wave

### C.2. Spectral Analysis of a Sine Wave with Gaussian White Noise

Periodicity analysis is applied to a sine wave with Gaussian white noise:

$$X(t) = A * \sin(\frac{2\pi t}{T} + \phi) + \varepsilon(t) \tag{C.2}$$

$\varepsilon(t)$ is Gaussian white noise. This experiment is made twice with standard deviation 10 and 100. Mean of $\varepsilon(t)$ is 0. (A, T, φ) is again set as (100, 10, 0). Time series are shown in Figures C.3 and C.5. Fourier power spectral densities are given in Figures C.4 and C.6.

As seen in Figures C.3 and C.4, introduction of low Gaussian white noise into the time series does not affect the pattern and periodicity significantly. Table C.1 shows that total power of the signal is increased by the introduction of white noise; but the power of dominant periodicity is not enhanced. This is competent with the fact that white noise doesn't affect the pattern of the data. The only dominant periodicity is at frequency 0.1.

Figure C.5 reveals that introduction of high Gaussian white noise disrupts the periodic pattern; however the pattern can still be identified. PSD in Figure C.6 captures the dominant periodicity at frequency 0.1; however introduction of high white noise results in not negligible power at all frequencies. Table C.1 demonstrates that total power of the signal is raised due to the increase in white noise, but the power of dominant periodicity is diminished; which is consistent with the change in the pattern.

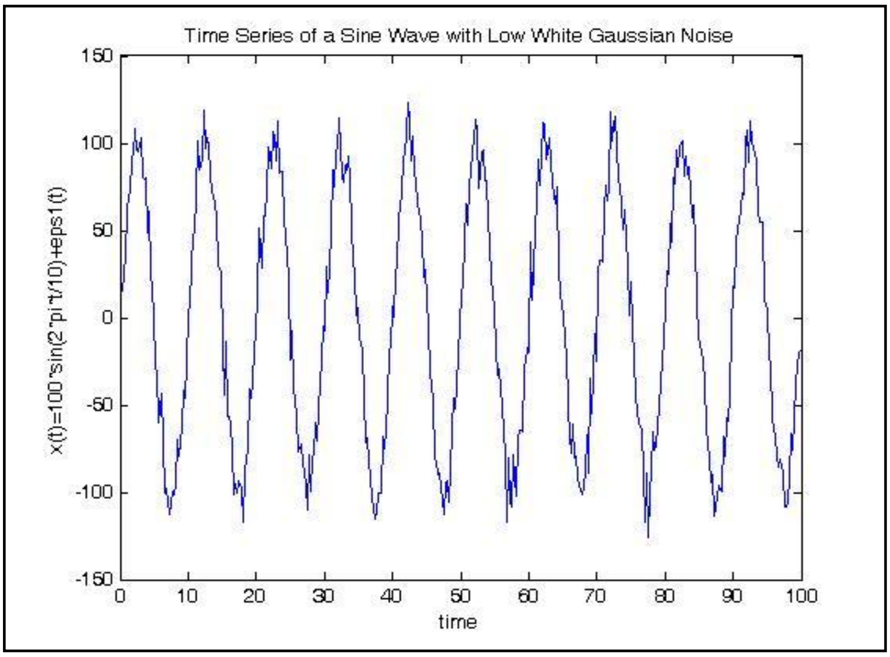

Figure C.3. Time Series of a Sine Wave with Low Gaussian White Noise

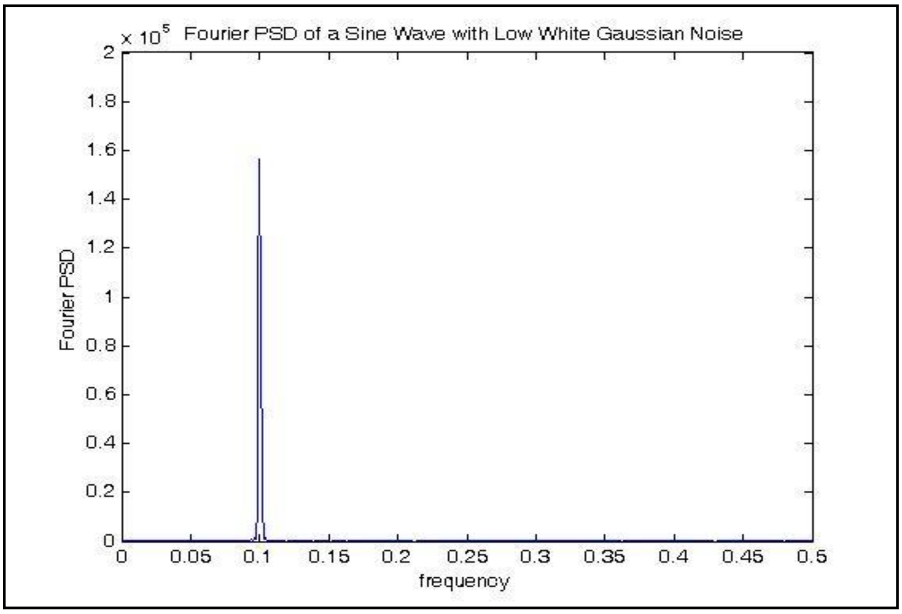

Figure C.4. Fourier PSD of a Sine Wave with Low Gaussian White Noise

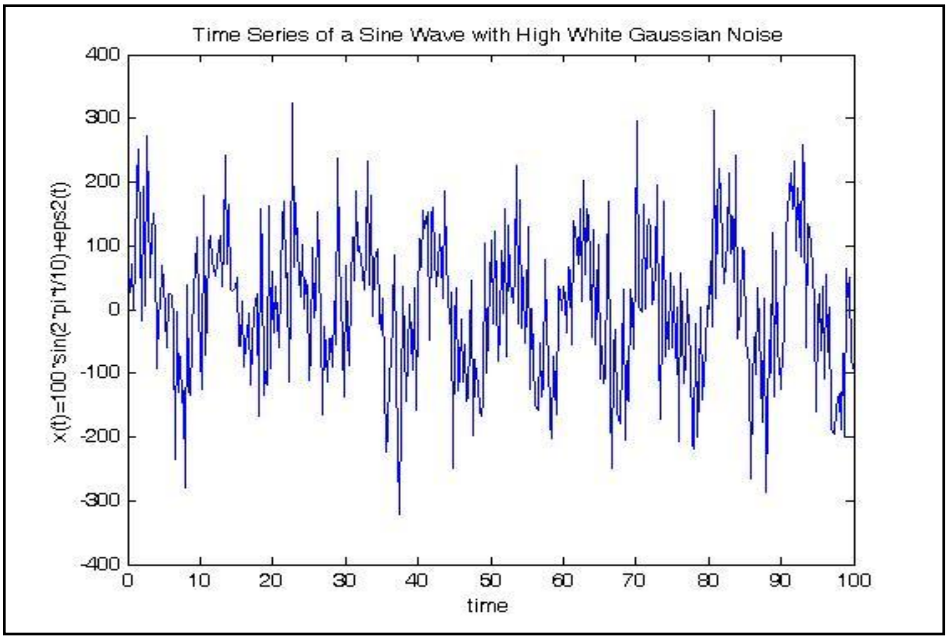


Figure C.5. Time Series of a Sine Wave with High Gaussian White Noise

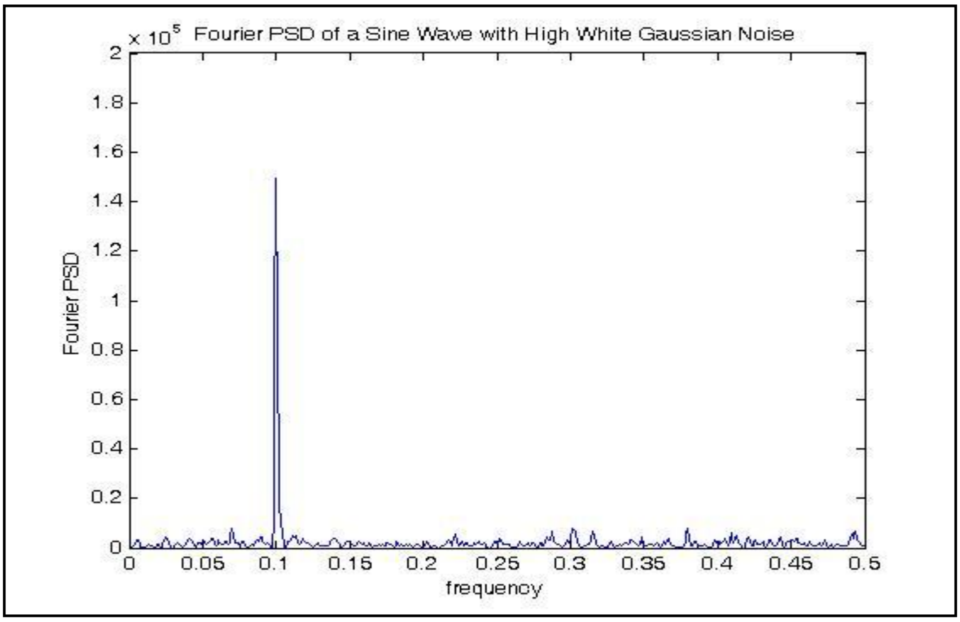


Figure C.6. Fourier PSD of a Sine Wave with High Gaussian Wave

### C.3. Spectral Analysis of a Sine Wave with Gaussian Pink Noise

Periodicity analysis is applied to a sine wave with Gaussian white noise:

$$X(t) = A * \sin(\frac{2\pi t}{T} + \phi) + smooth(\varepsilon(t)) \qquad \text{(C.3)}$$

$\varepsilon(t)$ is Gaussian white noise. Pink noise is produced by introducing autocorrelation through exponential smoothing. Exponential smoothing parameter is selected as 1/3. Fourier spectral analysis is applied to smoothed time series of Section C.2. Time series are shown in Figures C.7 and C.9. Fourier power spectral densities are given in Figures C.8 and C.10.

As seen in Figures C.7 and C.8, introduction of low Gaussian pink noise does not disturb the pattern significantly. Table C.1 demonstrates that both total power and total dominant power are decreased by the smoothing of white noise. The only dominant periodicity is at frequency 0.1 and almost all of the total power is concentrated at this frequency, which can be seen from both Table C.1 and Figure C.8.

Figure C.9 reveals that smoothing white noise has made the pattern more identifiable. This argument is supported by Table C.1, which shows that the total periodicity is decreased as a result of smoothing but the dominant periodicity is enhanced by the presence of a clearer pattern. Low-pass filter removes the high frequency noise in the data, which can be observed from Figure C.10. Two dominant periodicities are observed. The most dominant one is at frequency 0.1 with power 4.57E+04; the other one is at frequency 0.015 with relatively small power 4.28E+03. The second dominant periodicity is a result of applying low-pass filter to white noise.

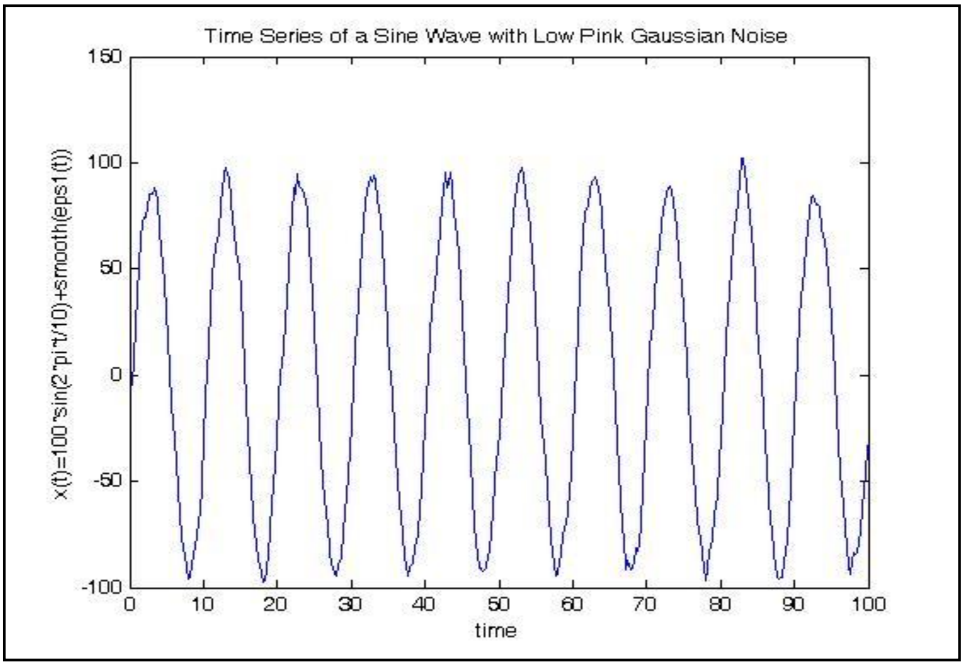


Figure C.7. Time Series of a Sine Wave with Low Gaussian Pink Noise

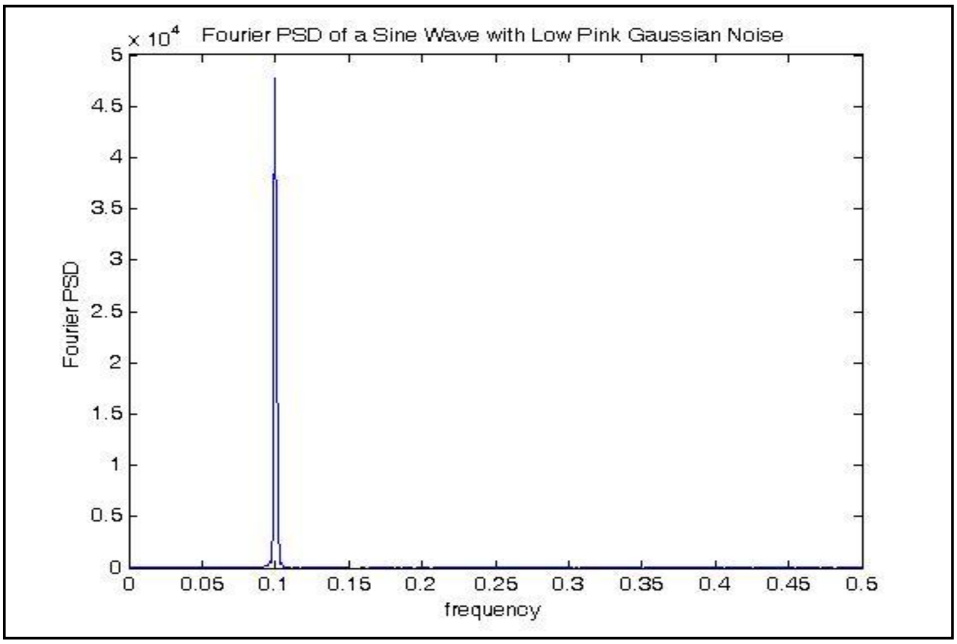


Figure C.8. Fourier PSD of a Sine Wave with Low Gaussian Pink Noise

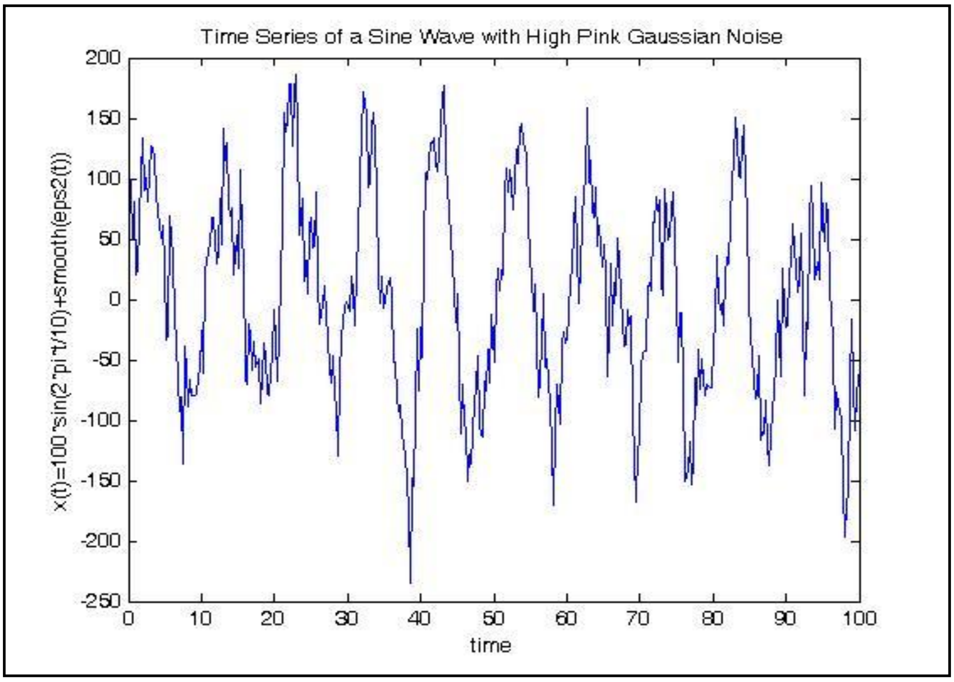


Figure C.9. Time Series of a Sine Wave with High Gaussian Pink Noise

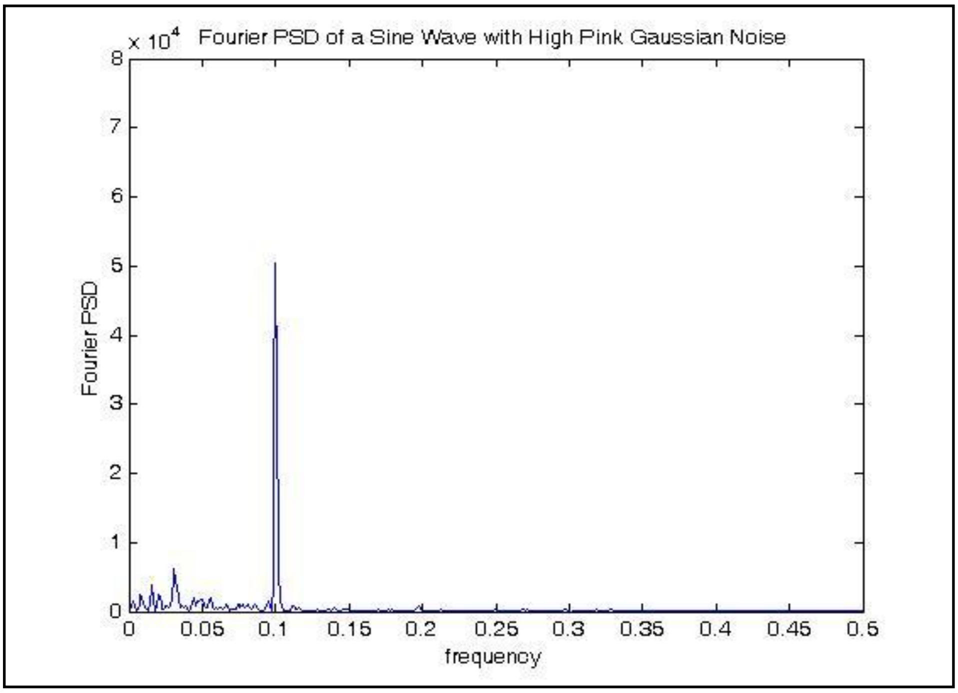


Figure C.10. Fourier PSD of a Sine Wave with High Gaussian Pink Noise

### C.4. Spectral Analysis of a Multi-periodic Wave

The analysis is applied to a wave with k-periodicities:

$$X(t) = \sum_{i=1}^{k} A_i * \sin(\frac{2\pi t}{T_i} + \phi_i) \quad \text{(C.4)}$$

The specific form of the wave used in this section is given below:

$$X(t) = 250 * \sin(\frac{2\pi t}{40}) + 400 * \sin(\frac{2\pi t}{25}) - 100 * \sin(\frac{2\pi t}{10}) \quad \text{(C.5)}$$

As seen in Figure C.12, the time series can be represented as a combination of these three periodicities. Table C.1 shows the dominant periodicities with powers proportional to their amplitudes.

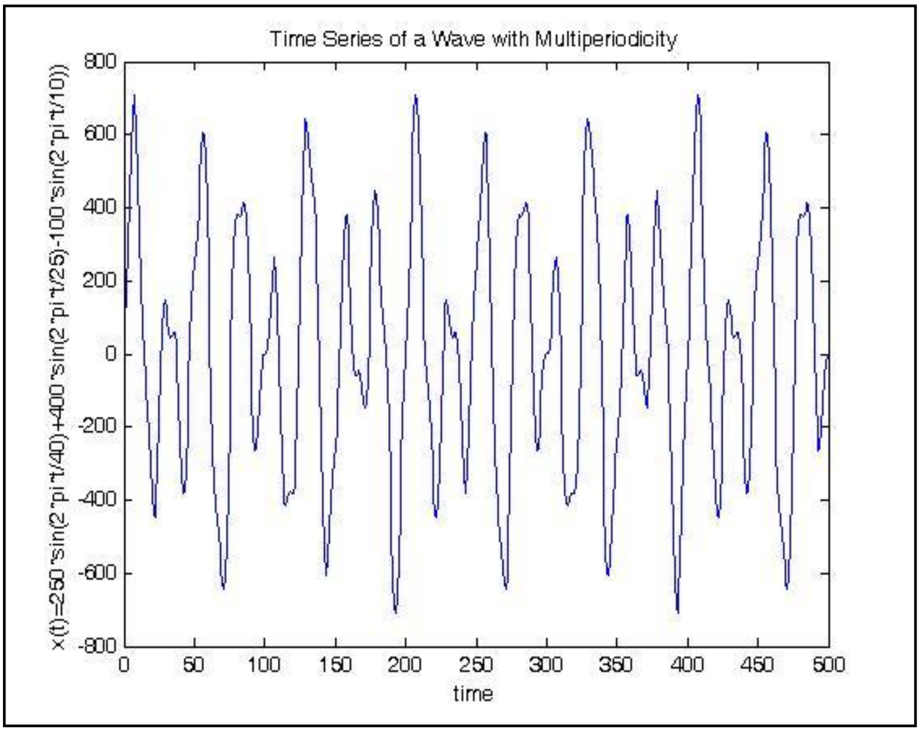


Figure C.11. Time Series of a Wave with Multiple Periodicities

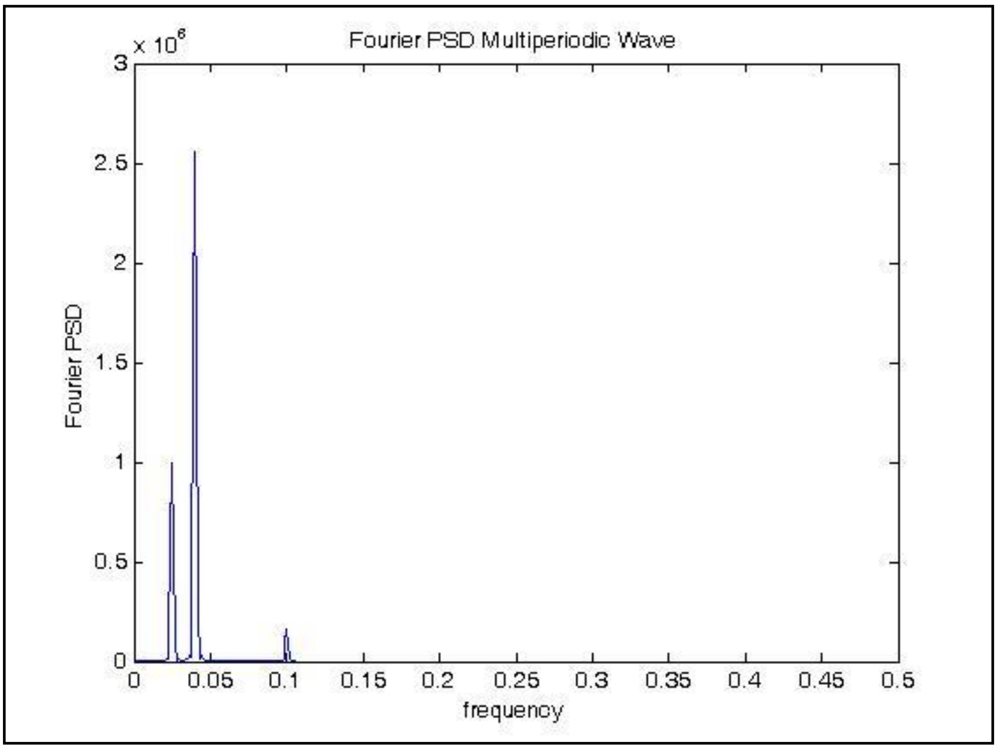


Figure C.12. Fourier PSD of a Wave with Multiple Periodicities

Table C.1 Summary of Periodicity Analysis for Synthetic Data

| Time Series | Total Dominant Spectral Power | Total Spectral Power | Periodicity Ratio | Power Threshold | 1st Dominant Period | 1st Dominant Period Power | 2nd Dominant Period | 2nd Dominant Period Power | 3rd Dominant Period | 3rd Dominant Period Power | 4th Dominant Period | 4th Dominant Period Power | 5th Dominant Period | 5th Dominant Period Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sine Wave | 3.83E+05 | 3.98E+05 | 0.9624 | 5.66E+03 | 10.00 | 1.60E+05 | - | - | - | - | - | - | - | - |
| Sine Wave with Low White Noise | 3.86E+05 | 4.10E+05 | 0.9436 | 5.99E+03 | 10.00 | 1.62E+05 | - | - | - | - | - | - | - | - |
| Sine Wave with High White Noise | 3.38E+05 | 1.16E+06 | 0.2922 | 1.65E+04 | 10.00 | 1.45E+05 | - | - | - | - | - | - | - | - |
| Sine Wave with Low Pink Noise | 1.16E+05 | 1.23E+05 | 0.9401 | 1.86E+03 | 10.00 | 4.86E+04 | - | - | - | - | - | - | - | - |
| Sine Wave with High Pink Noise | 1.11E+05 | 2.54E+05 | 0.4384 | 4.27E+03 | 10.00 | 4.57E+04 | 66.67 | 4.28E+03 | - | - | - | - | - | - |
| White Noise | 0.00E+00 | 7.16E+05 | 0.0000 | 1.20E+04 | - | - | - | - | - | - | - | - | - | - |
| Pink Noise | 3.43E+04 | 1.57E+05 | 0.2182 | 2.24E+03 | 100.00 | 6.52E+03 | 58.82 | 5.58E+03 | 76.92 | 4.78E+03 | 21.28 | 3.34E+03 | 37.04 | 3.20E+03 |
| Multiperiodic Wave | 8.47E+06 | 9.23E+06 | 0.9174 | 1.35E+05 | 25.00 | 2.56E+06 | 40.00 | 9.96E+05 | 10.00 | 1.61E+05 | - | - | - | - |

# APPENDIX D: AGGREGATE ORDER AND INVENTORY DYNAMICS UNDER BASIC ORDERING POLICIES

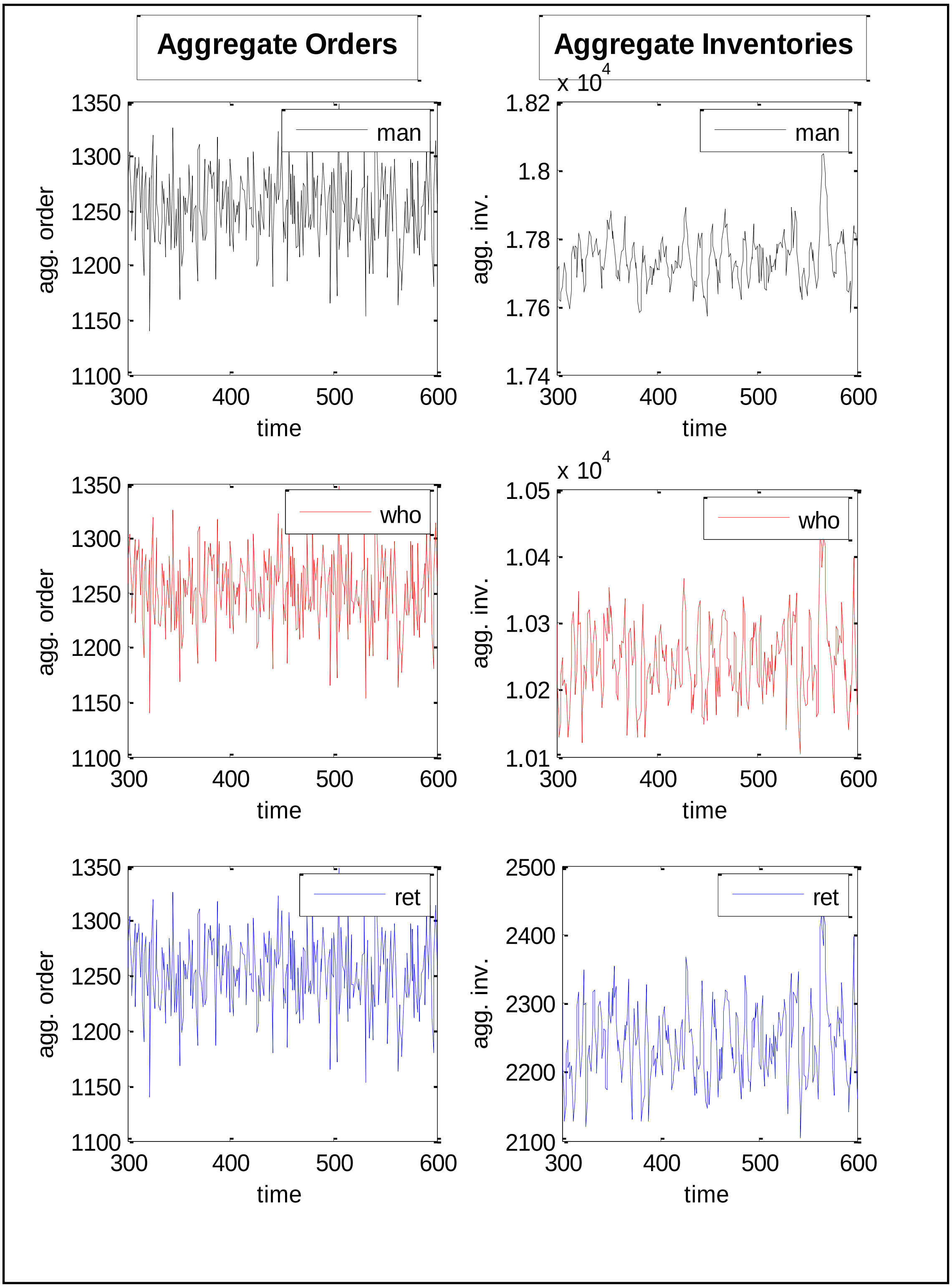


Figure D.1. Aggregate Order & Inventory Dynamics for Fixed S Policy
(2 Rtl., 2 Whl., 2 Mnf.)

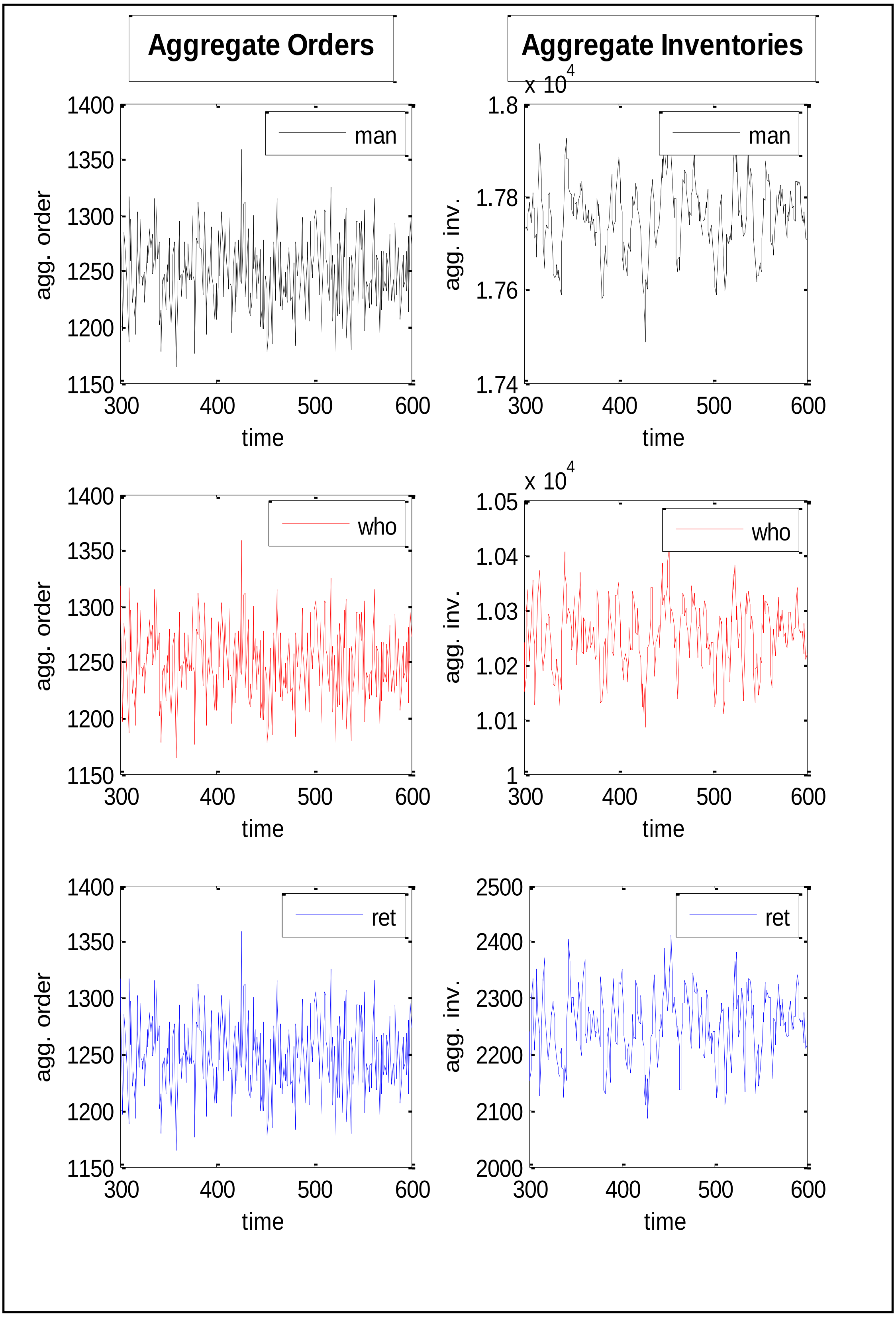


Figure D.2. Aggregate Order & Inventory Dynamics for Fixed S Policy

(4 Rtl., 4 Whl., 4 Mnf.)

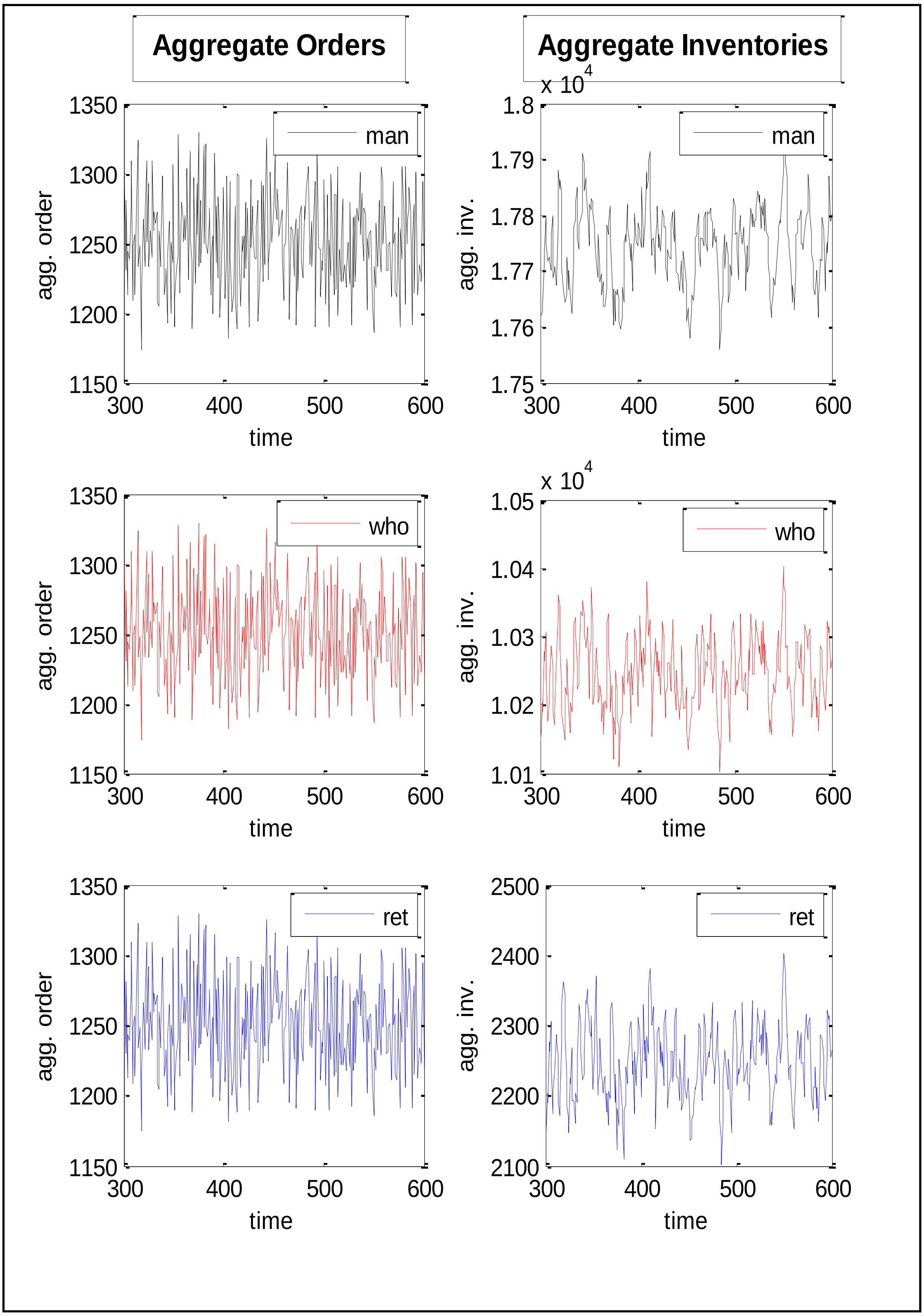


Figure D.3. Aggregate Order & Inventory Dynamics for Fixed S Policy

(50 Rtl., 20 Whl., 4 Mnf.)

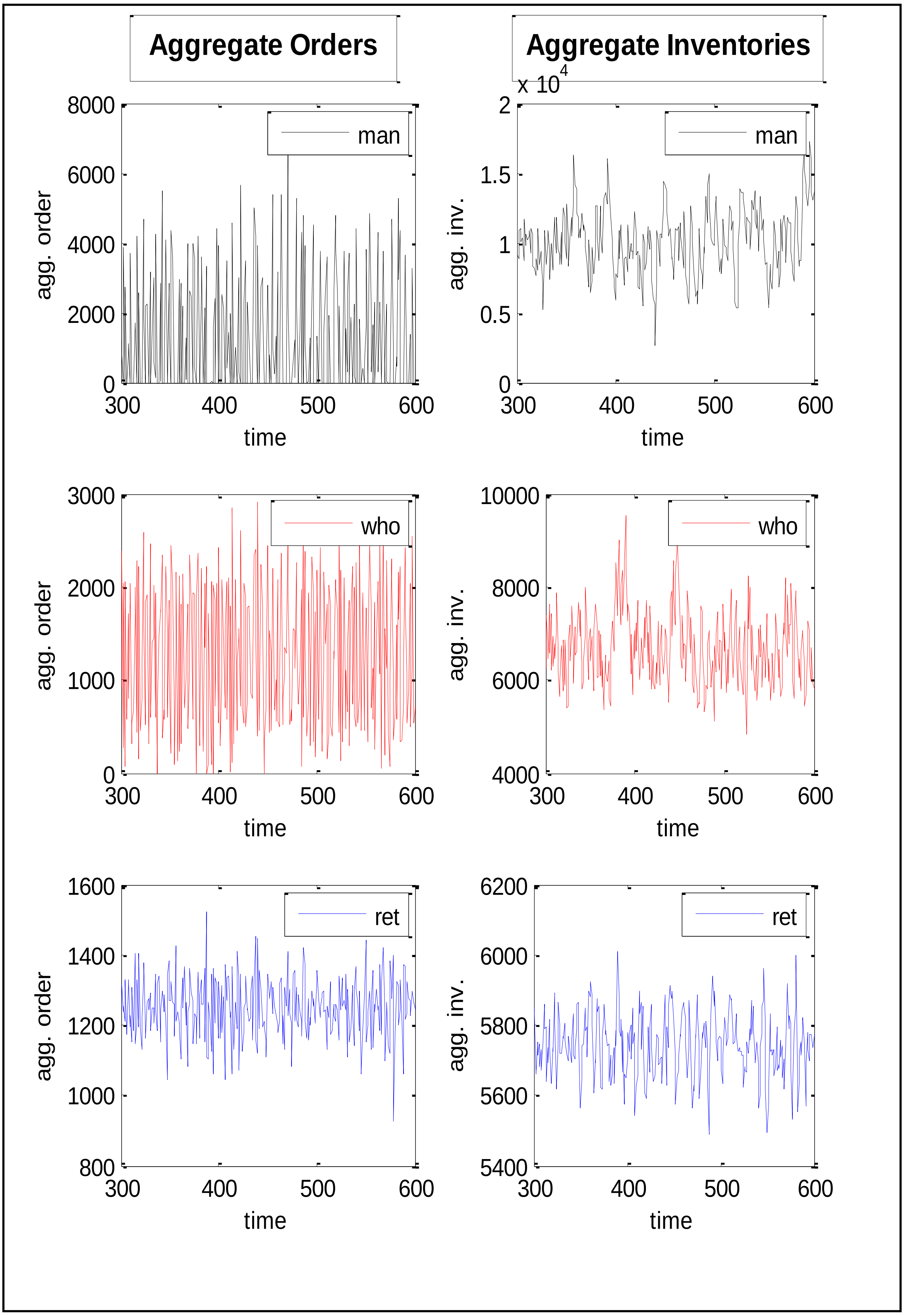


Figure D.4. Aggregate Order & Inventory Dynamics for Variable S Policy
(2 Rtl., 2 Whl., 2 Mnf.)

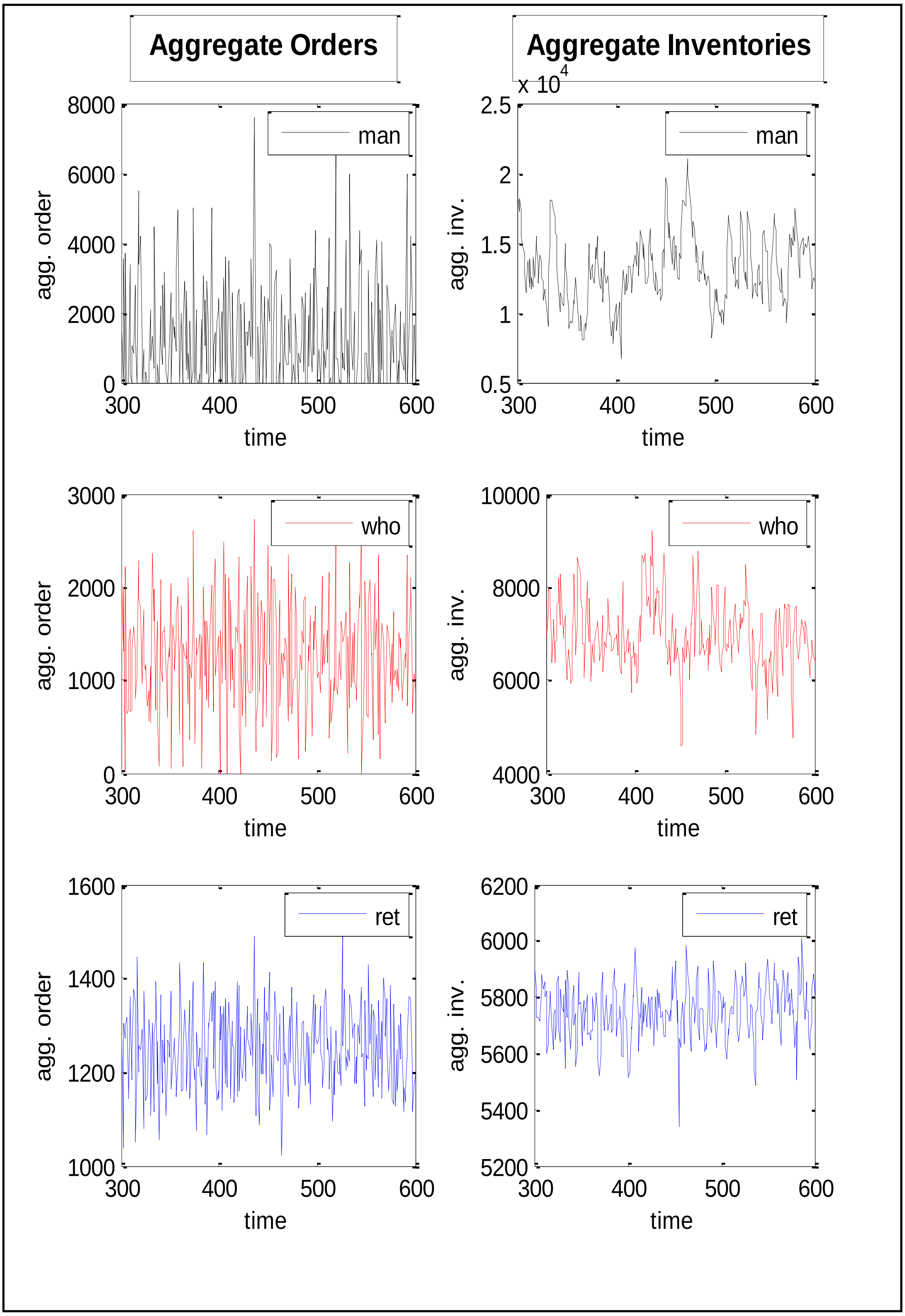


Figure D.5. Aggregate Order & Inventory Dynamics for Variable S Policy

(4 Rtl., 4 Whl., 4 Mnf.)

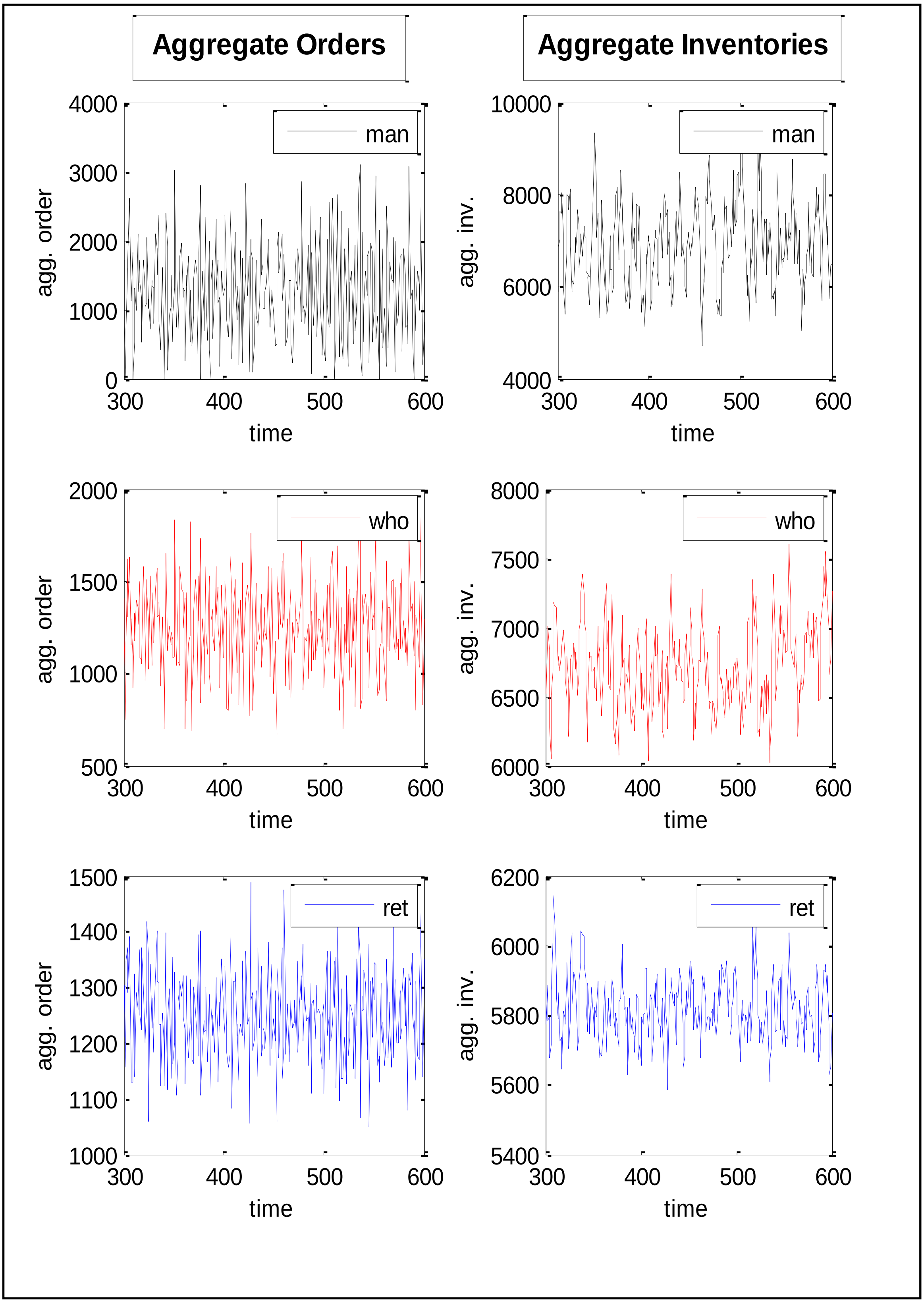


Figure D.6. Aggregate Order & Inventory Dynamics for Variable S Policy (50 Rtl., 20 Whl., 4 Mnf.)

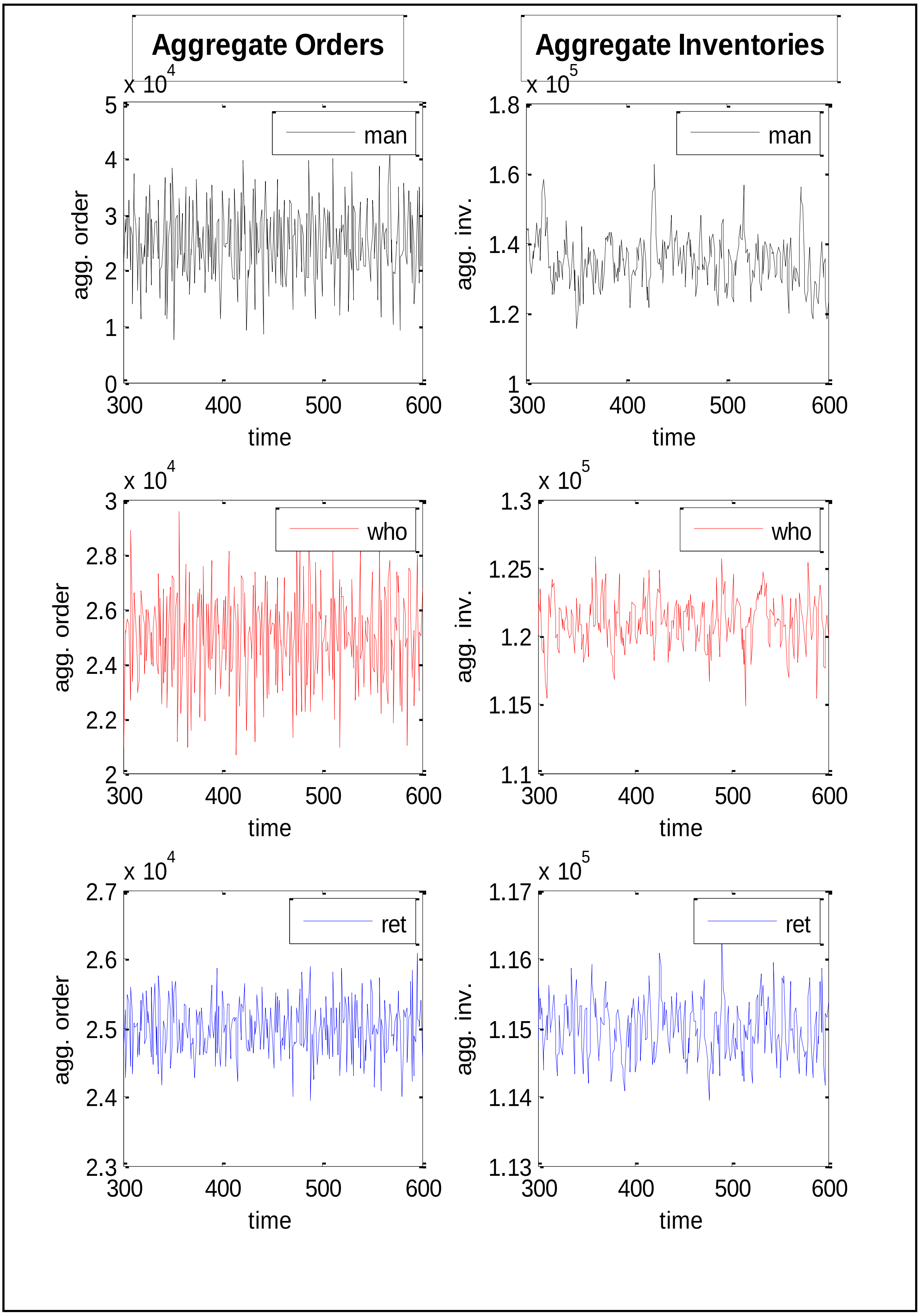


Figure D.7. Aggregate Order & Inventory Dynamics for Variable S Policy (200 Rtl., 50 Whl., 10 Mnf.)

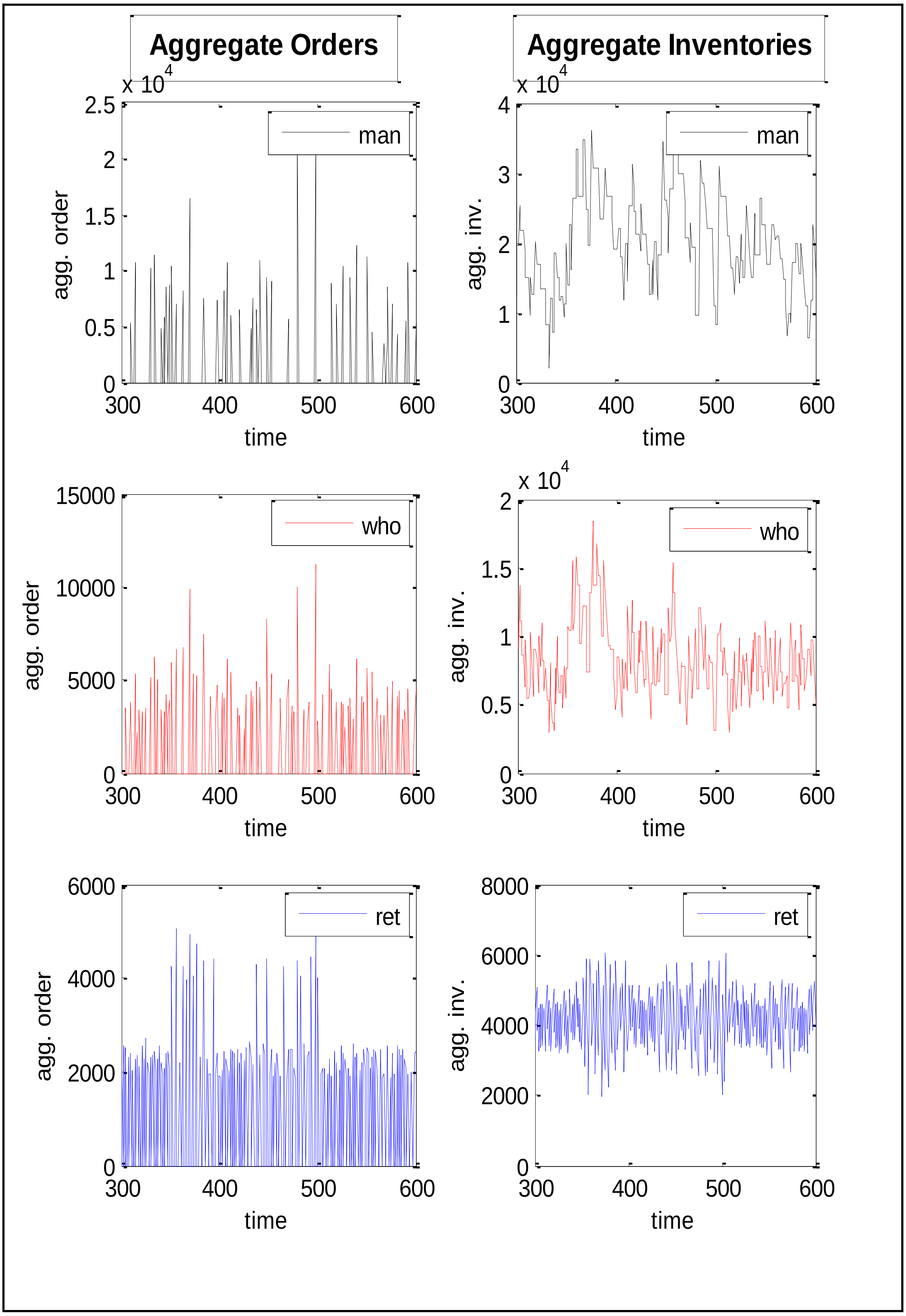


Figure D.8. Aggregate Order & Inventory Dynamics for (s,S) Policy
(2 Rtl., 2 Whl., 2 Mnf.)

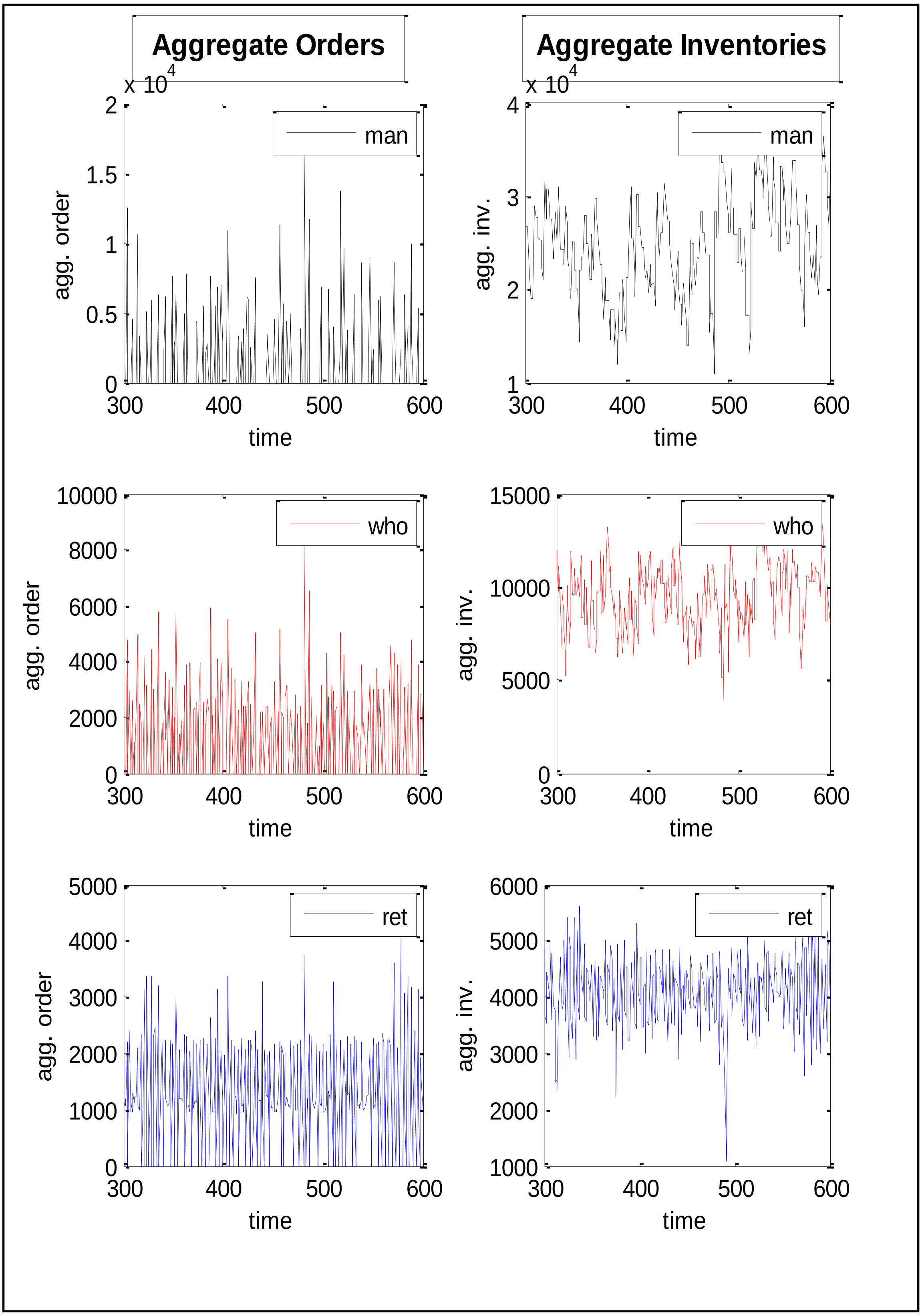


Figure D.9. Aggregate Order & Inventory Dynamics for (s,S) Policy

(4 Rtl., 4 Whl., 4 Mnf.)

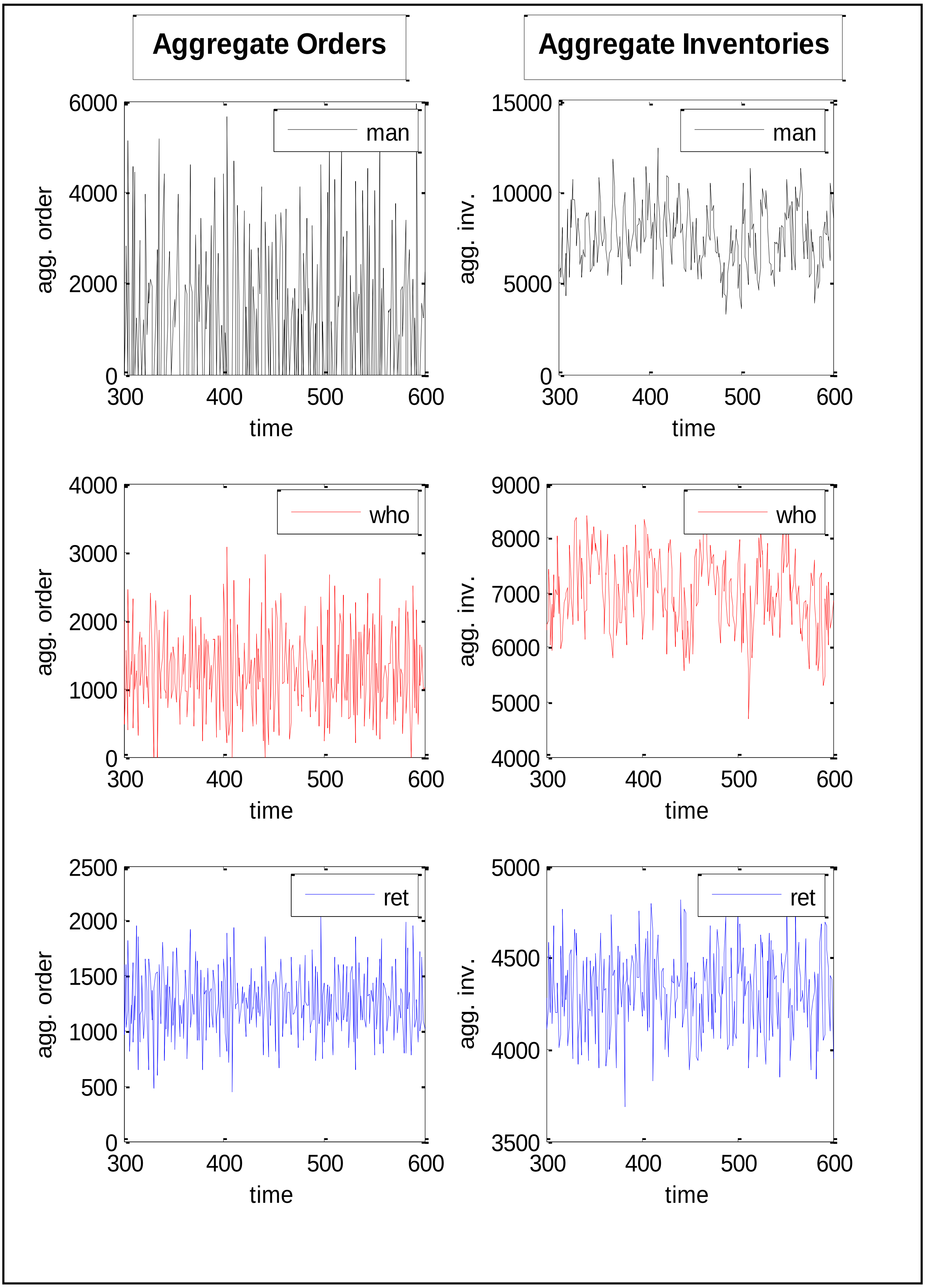


Figure D.10. Aggregate Order & Inventory Dynamics for (s,S) Policy
(50 Rtl., 20 Whl., 4 Mnf.)

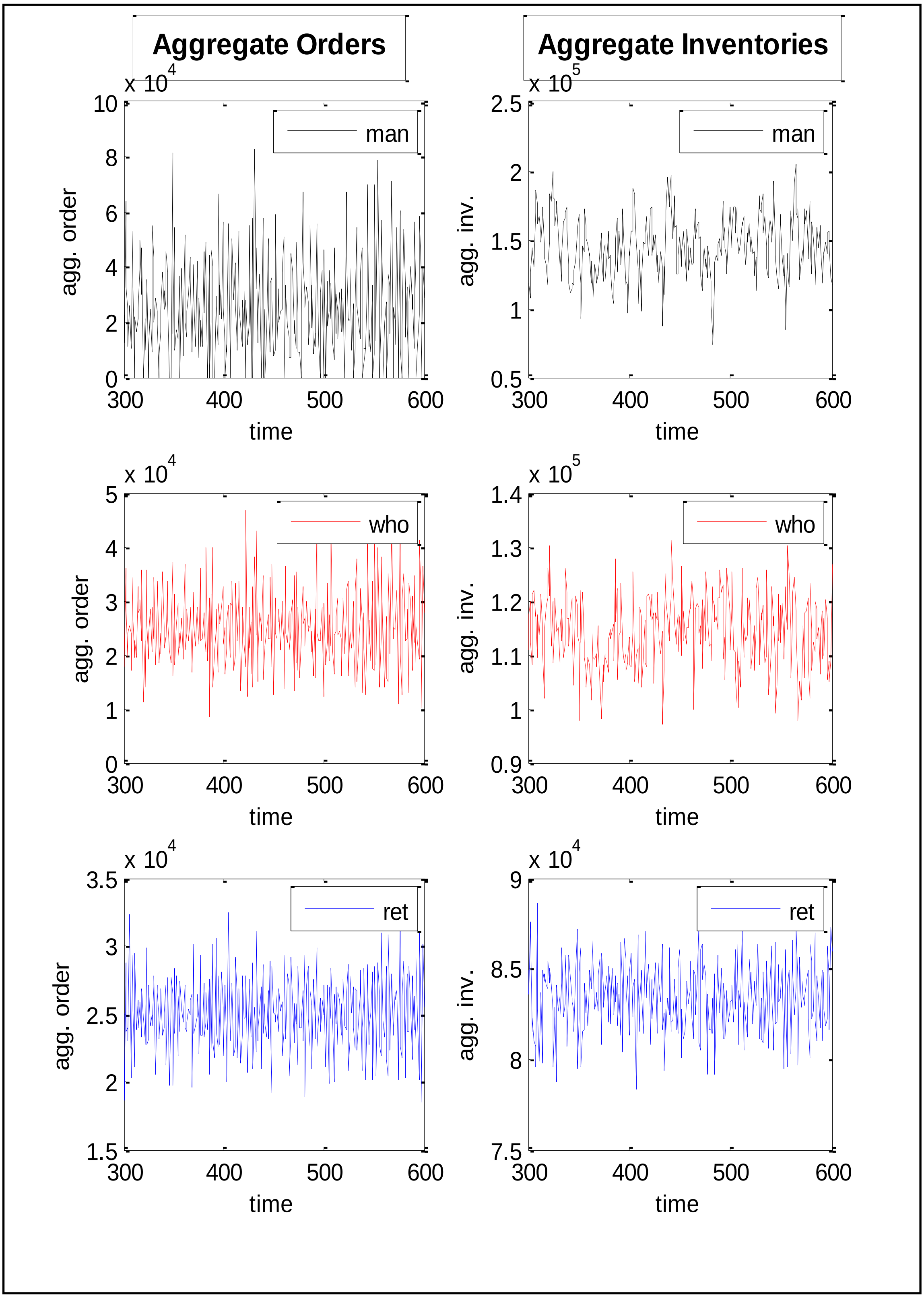


Figure D.11. Aggregate Order & Inventory Dynamics for (s,S) Policy

(200 Rtl., 50 Whl., 10 Mnf.)

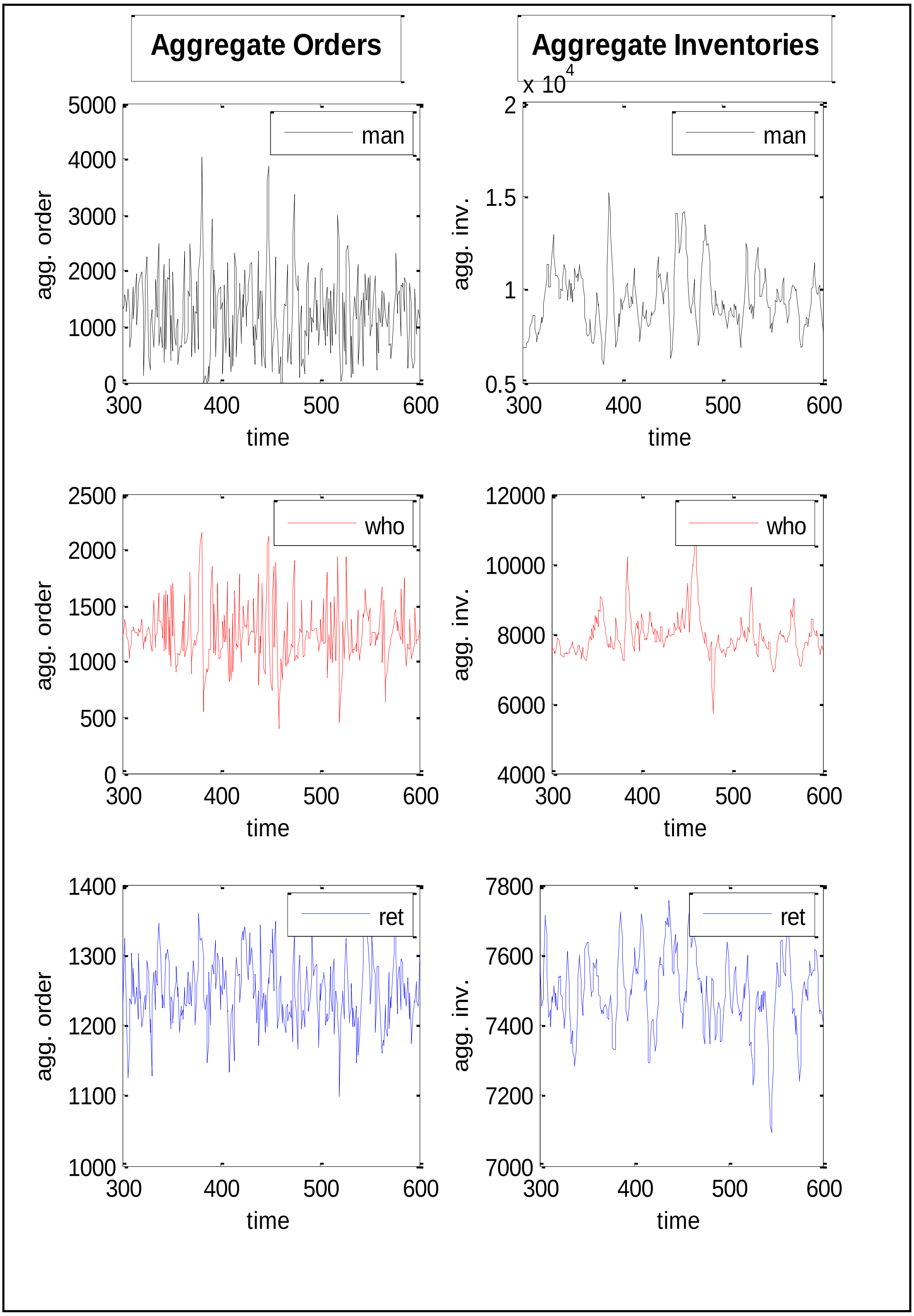


Figure D.12. Aggregate Order & Inventory Dynamics for A&A Policy
(2 Rtl., 2 Whl., 2 Mnf.)

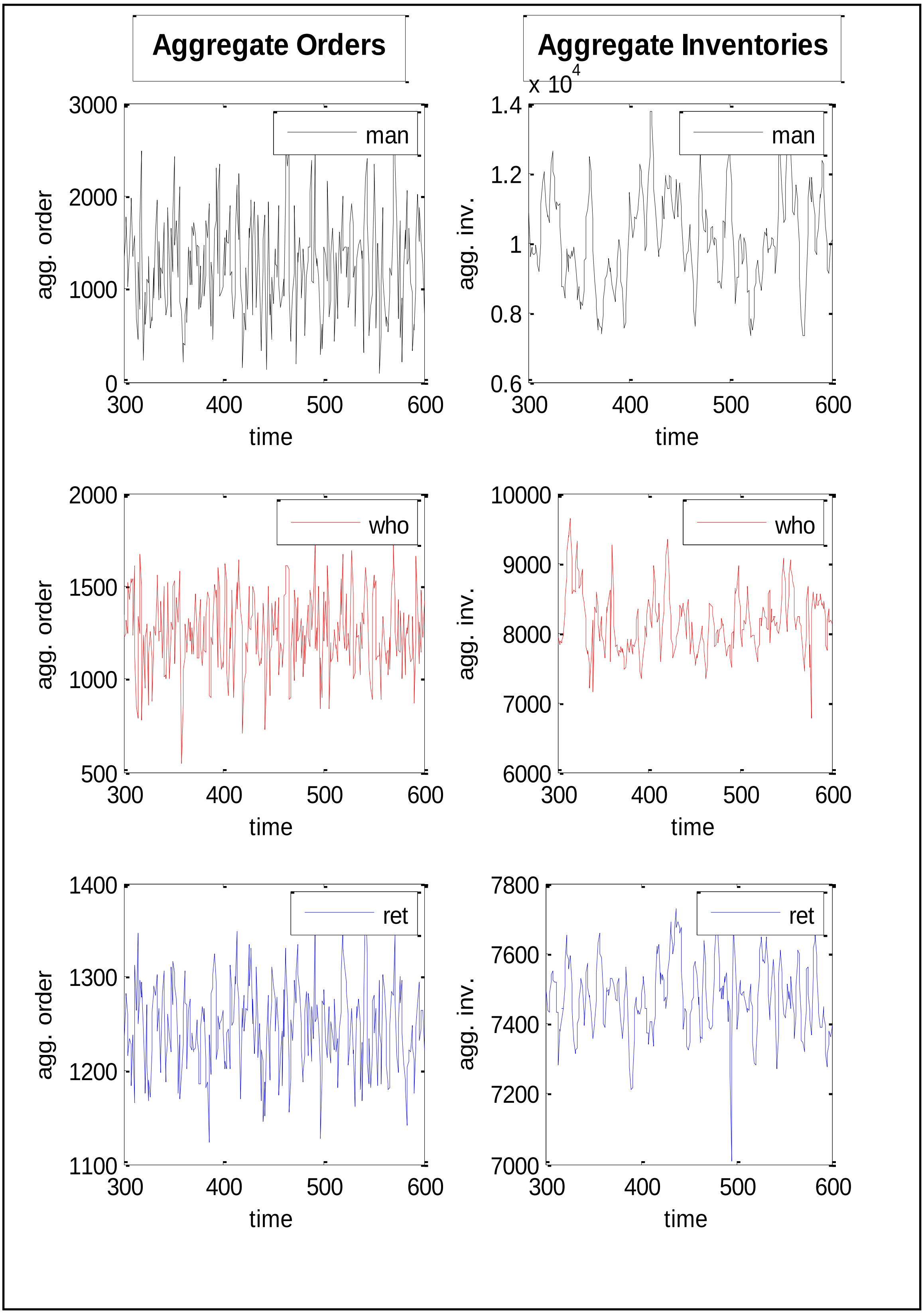


Figure D.13. Aggregate Order & Inventory Dynamics for A&A Policy

(4 Rtl., 4 Whl., 4 Mnf.)

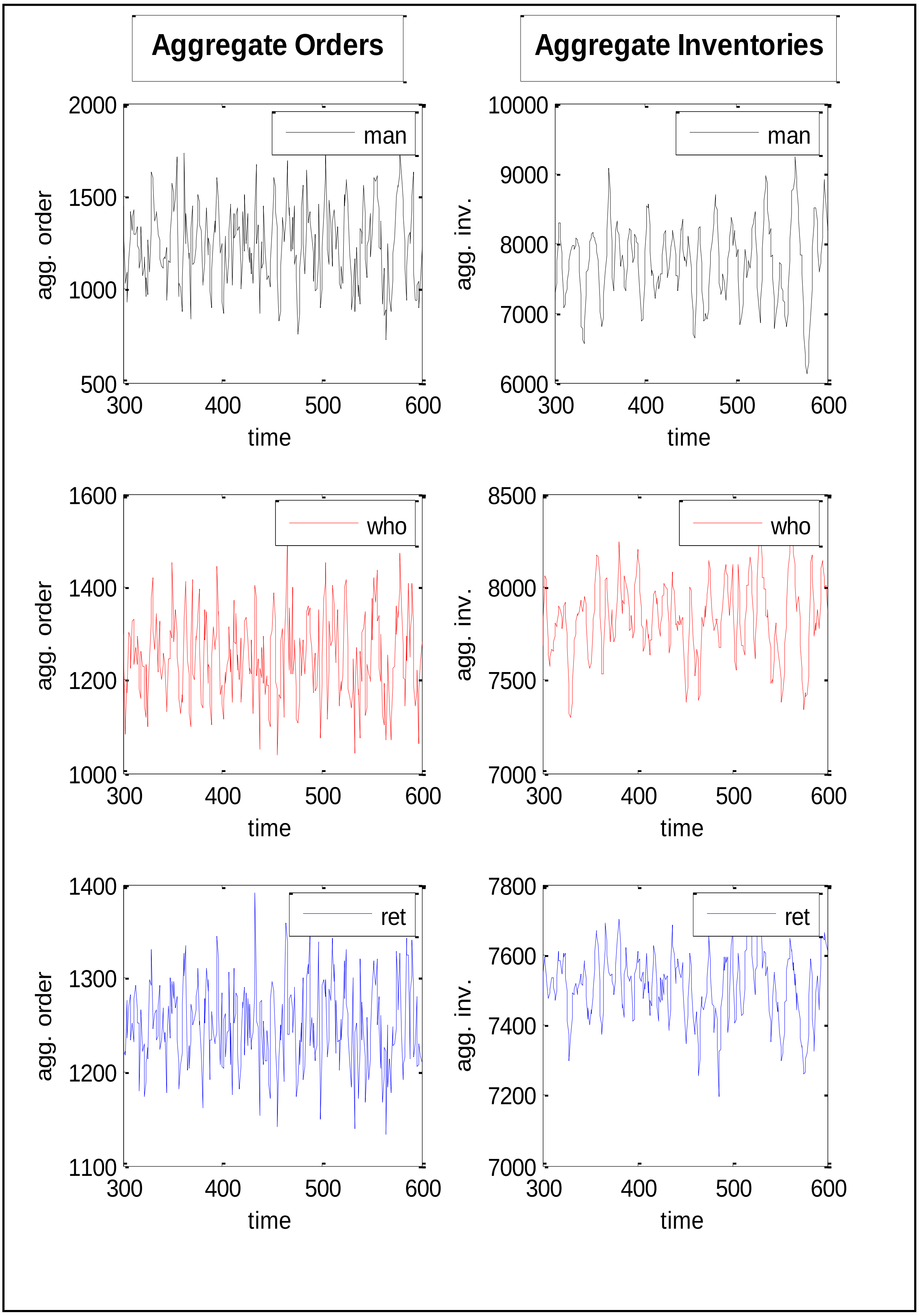


Figure D.14. Aggregate Order & Inventory Dynamics for A&A Policy
(50 Rtl., 20 Whl., 4 Mnf.)

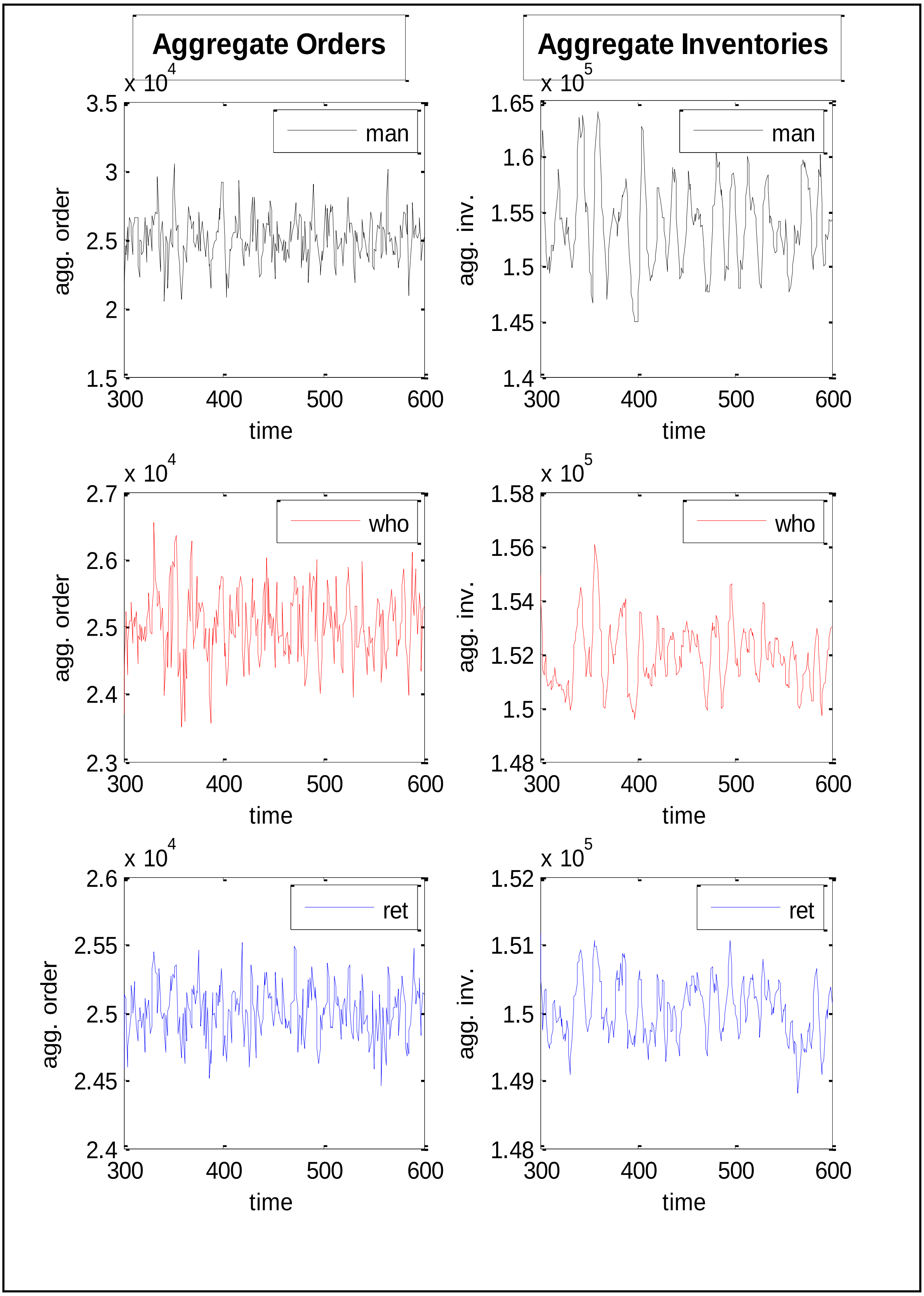


Figure D.15. Aggregate Order & Inventory Dynamics for A&A Policy

(200 Rtl., 50 Whl., 10 Mnf.)

Table D.1. Spectral Analysis of Fixed S Policy Order Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Order- 2R, 2W, 2M* | 0 | 8.30 E+02 | 0.000 | 2.97 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 2R, 2W, 2M* | 0 | 8.30 E+02 | 0.000 | 2.69 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 2R, 2W, 2M* | 0 | 8.30 E+02 | 0.000 | 2.82 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order - 4R, 4W, 4M* | 0 | 9.01 E+02 | 0.000 | 2.74 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 4R, 4W, 4M* | 0 | 9.01 E+02 | 0.000 | 2.87 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 4R, 4W, 4M* | 0 | 9.01 E+02 | 0.000 | 2.97 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 10W, 2M* | 0 | 9.71 E+02 | 0.000 | 2.81 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 10W, 2M* | 0 | 9.71 E+02 | 0.000 | 2.79 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 20R, 10W, 2M* | 0 | 9.71 E+02 | 0.000 | 2.93 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 20W,20M* | 0 | 9.75 E+02 | 0.000 | 3.15 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 20W,20M* | 0 | 9.75 E+02 | 0.000 | 3.07 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 20R, 20W,20M* | 0 | 9.75 E+02 | 0.000 | 2.89 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 50R, 50W,50M* | 0 | 9.75 E+02 | 0.000 | 2.88 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 50R, 50W,50M* | 0 | 9.75 E+02 | 0.000 | 2.81 E+01 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 50R, 50W,50M* | 0 | 9.75 E+02 | 0.000 | 3.01 E+01 | - | - | - | - | - | - | - | - | - | - |

Table D.2. Spectral Analysis of Fixed S Policy Inventory Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Inv- 2R, 2W, 2M* | 1.97 E+04 | 2.02 E+05 | 0.098 | 3.45 E+03 | 81.82 | 4.58 E+03 | 20.00 | 4.06 E+03 | 7.50 | 3.69 E+03 | 15.25 | 3.49 E+03 | - | - |
| *Whl. Inv- 2R, 2W, 2M* | 1.62 E+04 | 2.02 E+05 | 0.081 | 3.57 E+03 | 81.82 | 4.58 E+03 | 20.00 | 4.06 E+03 | 7.50 | 3.69 E+03 | - | - | - | - |
| *Mnf. Inv- 2R, 2W, 2M* | 8.96 E+04 | 3.48 E+05 | 0.257 | 5.82 E+03 | 81.82 | 1.25 E+04 | 20.00 | 9.80 E+03 | 15.25 | 7.69 E+03 | 128.57 | 7.21 E+03 | 16.67 | 6.09 E+03 |
| *Rtl. Inv - 4R, 4W, 4M* | 2.13 E+04 | 2.27 E+05 | 0.094 | 4.10 E+03 | 42.86 | 4.76 E+03 | 56.25 | 4.16 E+03 | 225.00 | 4.10 E+03 | - | - | - | - |
| *Whl. Inv- 4R, 4W, 4M* | 1.72 E+04 | 2.27 E+05 | 0.094 | 4.09 E+03 | 42.86 | 4.76 E+03 | 56.25 | 4.16 E+03 | 225.00 | 4.10 E+03 | - | - | - | - |
| *Mnf. Inv- 4R, 4W, 4M* | 1.43 E+05 | 3.82 E+05 | 0.376 | 5.93 E+03 | 42.86 | 1.28 E+04 | 56.25 | 1.14 E+04 | 225.00 | 1.14 E+04 | 19.57 | 7.79 E+03 | 300.00 | 7.30 E+03 |
| *Rtl. OInv- 20R, 10W, 2M* | 2.11 E+04 | 2.20 E+05 | 0.096 | 3.23 E+03 | 40.91 | 5.08 E+03 | 9.68 | 4.46 E+03 | - | - | - | - | - | - |
| *Whl. Inv- 20R, 10W, 2M* | 1.78 E+04 | 2.20 E+05 | 0.081 | 3.57 E+03 | 40.91 | 5.08 E+03 | 9.68 | 4.46 E+03 | - | - | - | - | - | - |
| *Mnf. Inv- 20R, 10W, 2M* | 1.04 E+05 | 3.51 E+05 | 0.295 | 5.48 E+03 | 40.91 | 1.40 E+04 | 23.68 | 7.09 E+03 | 9.68 | 7.06 E+03 | 27.27 | 7.05 E+03 | 16.67 | 5.89 E+03 |
| *Rtl. Inv- 20R, 20W, 20M* | 2.41 E+04 | 2.22 E+05 | 0.109 | 3.51 E+03 | 40.91 | 8.24 E+03 | 7.20 | 3.89 E+03 | - | - | - | - | - | - |
| *Whl. Inv- 20R, 20W, 20M* | 2.41 E+04 | 2.22 E+05 | 0.109 | 3.64 E+03 | 40.91 | 8.24 E+03 | 7.20 | 3.89 E+03 | - | - | - | - | - | - |
| *Mnf. Inv- 20R, 20W, 20M* | 7.02 E+04 | 3.56 E+05 | 0.197 | 6.20 E+03 | 40.91 | 2.20 E+04 | 34.62 | 6.93 E+03 | - | - | - | - | - | - |
| *Rtl. Inv- 50R, 50W, 50M* | 1.91 E+04 | 2.32 E+05 | 0.082 | 3.95 E+03 | 15.25 | 5.11 E+03 | 10.11 | 4.94 E+03 | - | - | - | - | - | - |
| *Whl. Inv- 50R, 50W, 50M* | 1.90 E+04 | 2.35 E+05 | 0.081 | 4.05 E+03 | 15.25 | 5.05 E+03 | 10.11 | 4.94 E+03 | - | - | - | - | - | - |
| *Mnf. Inv- 50R, 50W, 50M* | 6.63 E+04 | 3.95 E+05 | 0.168 | 6.97 E+03 | 15.25 | 1.12 E+04 | 16.07 | 8.28 E+03 | 23.68 | 8.03 E+03 | 10.11 | 7.69 E+03 | 16.98 | 7.37 E+03 |

Table D.3. Spectral Analysis of Variable S Policy Order Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Order- 2R, 2W, 2M* | 0 | 5.28 E+03 | 0 | 1.92 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 2R, 2W, 2M* | 0 | 5.06 E+05 | 0 | 1.61 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 2R, 2W, 2M* | 0 | 2.65 E+06 | 0 | 9.05 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order - 4R, 4W, 4M* | 0 | 5.79 E+03 | 0 | 2.07 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 4R, 4W, 4M* | 0 | 2.58 E+05 | 0 | 8.45 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 4R, 4W, 4M* | 0 | 1.73 E+06 | 0 | 6.08 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 10W, 2M* | 0 | 5.76 E+03 | 0 | 1.82 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 10W, 2M* | 0 | 1.02 E+05 | 0 | 3.12 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 20R, 10W, 2M* | 0 | 9.15 E+05 | 0 | 2.61 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 20W,20M* | 0 | 6.50 E+03 | 0 | 1.98 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 20W,20M* | 0 | 8.71 E+04 | 0 | 2.67 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 20R, 20W,20M* | 0 | 5.55 E+05 | 0 | 1.77 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 50R, 50W,50M* | 0 | 6.34 E+03 | 0 | 2.05 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 50R, 50W,50M* | 0 | 4.56 E+04 | 0 | 1.56 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 50R, 50W,50M* | 0 | 2.65 E+05 | 0 | 9.57 E+03 | - | - | - | - | - | - | - | - | - | - |

Table D.4. Spectral Analysis of Variable S Policy Inventory Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Inv- 2R, 2W, 2M* | 3.75 E+04 | 6.19 E+05 | 0.061 | 9.19 E+03 | 28.125 | 1.38 E+04 | 15.517 | 1.02 E+04 | - | - | - | - | - | - |
| *Whl. Inv- 2R, 2W, 2M* | 8.65 E+06 | 4.88 E+07 | 0.177 | 7.89 E+05 | 25.714 | 2.06 E+06 | 34.615 | 1.17 E+06 | 39.130 | 1.06 E+06 | - | - | - | - |
| *Mnf. Inv- 2R, 2W, 2M* | 1.23 E+08 | 4.49 E+08 | 0.274 | 7.21 E+06 | 25.000 | 1.95 E+07 | 23.077 | 1.28 E+07 | 90.000 | 8.51 E+06 | 42.857 | 8.29 E+06 | 10.588 | 8.03 E+06 |
| *Rtl. Inv - 4R, 4W, 4M* | 2.47 E+04 | 7.65 E+05 | 0.032 | 1.13 E+04 | 11.250 | 1.28 E+04 | 10.588 | 1.19 E+04 | - | - | - | - | - | - |
| *Whl. Inv- 4R, 4W, 4M* | 2.06 E+06 | 2.96 E+07 | 0.070 | 5.00 E+05 | 36.000 | 7.36 E+05 | 64.286 | 5.93 E+05 | - | - | - | - | - | - |
| *Mnf. Inv- 4R, 4W, 4M* | 2.68 E+08 | 4.98 E+08 | 0.538 | 7.66 E+06 | 75.000 | 2.68 E+07 | 225.000 | 2.37 E+07 | 100.000 | 2.22 E+07 | 36.000 | 1.69 E+07 | 300.000 | 1.63 E+07 |
| *Rtl. OInv- 20R, 10W, 2M* | 0 | 6.33 E+05 | 0.000 | 1.01 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Inv- 20R, 10W, 2M* | 9.95 E+05 | 1.14 E+07 | 0.087 | 1.76 E+05 | 21.429 | 2.48 E+05 | 100.000 | 1.97 E+05 | 7.438 | 1.92 E+05 | - | - | - | - |
| *Mnf. Inv- 20R, 10W, 2M* | 1.86 E+07 | 1.16 E+08 | 0.161 | 1.57 E+06 | 21.429 | 2.92 E+06 | 7.500 | 2.07 E+06 | 14.754 | 2.02 E+06 | 16.071 | 1.84 E+06 | - | - |
| *Rtl. Inv- 20R, 20W, 20M* | 6.75 E+04 | 9.51 E+05 | 0.071 | 1.46 E+04 | 10.112 | 1.92 E+04 | 9.184 | 1.59 E+04 | 7.317 | 1.50 E+04 | - | - | - | - |
| *Whl. Inv- 20R, 20W, 20M* | 3.09 E+06 | 1.44 E+07 | 0.214 | 2.19 E+05 | 39.130 | 3.35 E+05 | 30.000 | 3.29 E+05 | 16.981 | 3.04 E+05 | 75.000 | 2.71 E+05 | 26.471 | 2.67 E+05 |
| *Mnf. Inv- 20R, 20W, 20M* | 1.91 E+08 | 2.97 E+08 | 0.643 | 4.61 E+06 | 112.500 | 6.49 E+07 | 64.286 | 7.69 E+06 | 450.000 | 6.06 E+06 | - | - | - | - |
| *Rtl. Inv- 50R, 50W, 50M* | 5.85 E+04 | 7.88 E+05 | 0.074 | 1.15 E+04 | 18.367 | 1.63 E+04 | 13.235 | 1.58 E+04 | - | - | - | - | - | - |
| *Whl. Inv- 50R, 50W, 50M* | 2.66 E+06 | 9.74 E+06 | 0.273 | 1.56 E+05 | 34.615 | 3.35 E+05 | 112.500 | 2.65 E+05 | 75.000 | 1.88 E+05 | 450.000 | 1.87 E+05 | 60.000 | 1.86 E+05 |
| *Mnf. Inv- 50R, 50W, 50M* | 6.90 E+07 | 1.04 E+08 | 0.666 | 1.49 E+06 | 81.818 | 9.31 E+06 | 33.333 | 3.18 E+06 | 52.941 | 3.12 E+06 | 150.000 | 2.70 E+06 | 40.909 | 2.64 E+06 |

Table D.5. Spectral Analysis of (s,S) Policy Order Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Order- 2R, 2W, 2M* | 9.12 E+04 | 1.66 E+06 | 0.055 | 5.62 E+04 | 3.531 | 9.12 E+04 | - | - | - | - | - | - | - | - |
| *Whl. Order- 2R, 2W, 2M* | 0 | 4.17 E+06 | 0.000 | 1.57 E+05 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 2R, 2W, 2M* | 0 | 1.04 E+07 | 0.000 | 3.45 E+05 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order - 4R, 4W, 4M* | 3.36 E+04 | 8.14 E+05 | 0.041 | 2.81 E+04 | 3.593 | 3.36 E+04 | - | - | - | - | - | - | - | - |
| *Whl. Order- 4R, 4W, 4M* | 1.03 E+05 | 2.47 E+06 | 0.042 | 8.09 E+04 | 3.568 | 1.03 E+05 | - | - | - | - | - | - | - | - |
| *Mnf. Order- 4R, 4W, 4M* | 0 | 7.53 E+06 | 0.000 | 2.54 E+05 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 10W, 2M* | 0 | 1.79 E+05 | 0.000 | 6.89 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 10W, 2M* | 5.07 E+04 | 7.15 E+05 | 0.071 | 2.34 E+04 | 2.510 | 2.60 E+04 | 3.021 | 2.46 E+04 | - | - | - | - | - | - |
| *Mnf. Order- 20R, 10W, 2M* | 0 | 3.93 E+06 | 0.000 | 1.26 E+05 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 20W,20M* | 6.33 E+03 | 1.77 E+05 | 0.036 | 5.99 E+03 | 4.303 | 6.33 E+03 | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 20W,20M* | 0 | 5.59 E+05 | 0.000 | 1.97 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 20R, 20W,20M* | 0 | 1.87 E+06 | 0.000 | 6.15 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 50R, 50W,50M* | 3.28 E+03 | 6.67 E+04 | 0.049 | 2.17 E+03 | 2.709 | 3.28 E+03 | - | - | - | - | - | - | - | - |
| *Whl. Order- 50R, 50W,50M* | 8.07 E+03 | 1.99 E+05 | 0.041 | 6.52 E+03 | 4.267 | 8.07 E+03 | - | - | - | - | - | - | - | - |
| *Mnf. Order- 50R, 50W,50M* | 0 | 7.53 E+05 | 0.000 | 2.39 E+04 | - | - | - | - | - | - | - | - | - | - |

Table D.6 .Spectral Analysis of (s,S) Policy Inventory Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Inv- 2R, 2W, 2M* | 1.54 E+07 | 5.83 E+07 | 0.264 | 9.10 E+05 | 3.422 | 1.93 E+06 | 3.614 | 1.49 E+06 | 3.719 | 1.17 E+06 | 3.846 | 1.08 E+06 | 3.475 | 1.04 E+06 |
| *Whl. Inv- 2R, 2W, 2M* | 0 | 4.18 E+08 | 0 | 8.28 E+06 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Inv- 2R, 2W, 2M* | 4.49 E+08 | 2.93 E+09 | 0.153 | 5.68 E+07 | 128.57 | 1.22 E+08 | 90 | 7.42 E+07 | - | - | - | - | - | - |
| *Rtl. Inv - 4R, 4W, 4M* | 4.86 E+06 | 3.21 E+07 | 0.151 | 5.92 E+05 | 3.422 | 1.03 E+06 | 3.370 | 8.15 E+05 | 3.515 | 6.19 E+05 | 3.6 | 6.02 E+05 | - | - |
| *Whl. Inv- 4R, 4W, 4M* | 1.50 E+08 | 3.80 E+08 | 0.396 | 6.41 E+06 | 37.5 | 2.37 E+07 | 56.25 | 1.29 E+07 | 100 | 1.11 E+07 | - | - | - | - |
| *Mnf. Inv- 4R, 4W, 4M* | 1.18 E+09 | 2.69 E+09 | 0.438 | 4.58 E+07 | 150 | 2.06 E+08 | 90 | 1.31 E+08 | 300 | 1.03 E+08 | 225 | 9.78 E+07 | 450 | 7.53 E+07 |
| *Rtl. OInv- 20R, 10W, 2M* | 4.83 E+05 | 6.77 E+06 | 0.071 | 1.26 E+05 | 3.515 | 2.07 E+05 | - | - | - | - | - | - | - | - |
| *Whl. Inv- 20R, 10W, 2M* | 2.14 E+07 | 8.03 E+07 | 0.267 | 1.24 E+06 | 100 | 3.08 E+06 | 56.25 | 2.51 E+06 | 180 | 2.03 E+06 | 225 | 1.78 E+06 | 31.03 | 1.72 E+06 |
| *Mnf. Inv- 20R, 10W, 2M* | 5.79 E+07 | 4.79 E+08 | 0.121 | 8.05 E+06 | 56.25 | 1.26 E+07 | 15.789 | 8.86 E+06 | 37.5 | 8.72 E+06 | 12.86 | 8.53 E+06 | - | - |
| *Rtl. Inv- 20R,20W, 20M* | 1.05 E+06 | 8.09 E+06 | 0.130 | 1.29 E+05 | 3.6 | 4.05 E+05 | 3.734 | 1.42 E+05 | - | - | - | - | - | - |
| *Whl. Inv- 20R,20W, 20M* | 1.29 E+07 | 6.49 E+07 | 0.199 | 1.30 E+06 | 81.818 | 2.96 E+06 | 28.125 | 1.71 E+06 | 128.57 | 1.63 E+06 | 47.37 | 1.48 E+06 | - | - |
| *Mnf. Inv- 20R,20W, 20M* | 3.21 E+08 | 6.90 E+08 | 0.465 | 1.38 E+07 | 300 | 1.15 E+08 | 450 | 9.51 E+07 | 225 | 6.13 E+07 | 900 | 3.35 E+07 | - | - |
| *Rtl. Inv- 50R,50W, 50M* | 6.72 E+04 | 3.32 E+06 | 0.020 | 6.23 E+04 | 4.147 | 6.72 E+04 | - | - | - | - | - | - | - | - |
| *Whl. Inv- 50R, 50W,50M* | 1.12 E+07 | 3.68 E+07 | 0.305 | 5.46 E+05 | 150 | 2.18 E+06 | 64.286 | 9.67 E+05 | 225 | 8.42 E+05 | - | - | - | - |
| *Mnf. Inv- 50R,50W, 50M* | 1.17 E+08 | 2.71 E+08 | 0.430 | 5.91 E+06 | 300 | 3.65 E+07 | 450 | 2.96 E+07 | 225 | 2.01 E+07 | 900 | 9.54 E+06 | 50 | 7.49 E+06 |

Table D.7. Spectral Analysis of A&A Policy Order Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Order- 2R, 2W, 2M* | 0 | 5.28 E+03 | 0 | 1.92 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 2R, 2W, 2M* | 0 | 5.06 E+05 | 0 | 1.61 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 2R, 2W, 2M* | 0 | 2.65 E+06 | 0 | 9.05 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order - 4R, 4W, 4M* | 0 | 5.79 E+03 | 0 | 2.07 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 4R, 4W, 4M* | 0 | 2.58 E+05 | 0 | 8.45 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 4R, 4W, 4M* | 0 | 1.73 E+06 | 0 | 6.08 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 10W, 2M* | 0 | 5.76 E+03 | 0 | 1.82 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 10W, 2M* | 0 | 1.02 E+05 | 0 | 3.12 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 20R, 10W, 2M* | 0 | 9.15 E+05 | 0 | 2.61 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 20R, 20W,20M* | 0 | 6.50 E+03 | 0 | 1.98 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 20R, 20W,20M* | 0 | 8.71 E+04 | 0 | 2.67 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 20R, 20W,20M* | 0 | 5.55 E+05 | 0 | 1.77 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Rtl. Order- 50R, 50W,50M* | 0 | 6.34 E+03 | 0 | 2.05 E+02 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Order- 50R, 50W,50M* | 0 | 4.56 E+04 | 0 | 1.56 E+03 | - | - | - | - | - | - | - | - | - | - |
| *Mnf. Order- 50R, 50W,50M* | 0 | 2.65 E+05 | 0 | 9.57 E+03 | - | - | - | - | - | - | - | - | - | - |

Table D.8. Spectral Analysis of A&A Policy Inventory Data

| Time Series | Dominant Power | Total Power | Per. Ratio | Power Threshold | 1st Dom. Per. | 1st Dom. Per. Power | 2nd Dom. Per. | 2nd Dom. Per. Power | 3rd Dom. Per. | 3rd Dom. Per. Power | 4th Dom. Per. | 4th Dom. Per. Power | 5th Dom. Per. | 5th Dom. Per. Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Rtl. Inv- 2R, 2W, 2M* | 3.75 E+04 | 6.19 E+05 | 0.061 | 9.19 E+03 | 28.125 | 1.38 E+04 | 15.517 | 1.02 E+04 | - | - | - | - | - | - |
| *Whl. Inv- 2R, 2W, 2M* | 8.65 E+06 | 4.88 E+07 | 0.177 | 7.89 E+05 | 25.714 | 2.06 E+06 | 34.615 | 1.17 E+06 | 39.130 | 1.06 E+06 | - | - | - | - |
| *Mnf. Inv- 2R, 2W, 2M* | 1.23 E+08 | 4.49 E+08 | 0.274 | 7.21 E+06 | 25.000 | 1.95 E+07 | 23.077 | 1.28 E+07 | 90.000 | 8.51 E+06 | 42.857 | 8.29 E+06 | 10.588 | 8.03 E+06 |
| *Rtl. Inv - 4R, 4W, 4M* | 2.47 E+04 | 7.65 E+05 | 0.032 | 1.13 E+04 | 11.250 | 1.28 E+04 | 10.588 | 1.19 E+04 | - | - | - | - | - | - |
| *Whl. Inv- 4R, 4W, 4M* | 2.06 E+06 | 2.96 E+07 | 0.070 | 5.00 E+05 | 36.000 | 7.36 E+05 | 64.286 | 5.93 E+05 | - | - | - | - | - | - |
| *Mnf. Inv- 4R, 4W, 4M* | 2.68 E+08 | 4.98 E+08 | 0.538 | 7.66 E+06 | 75.000 | 2.68 E+07 | 225.000 | 2.37 E+07 | 100.000 | 2.22 E+07 | 36.000 | 1.69 E+07 | 300.000 | 1.63 E+07 |
| *Rtl. OInv- 20R, 10W, 2M* | 0 | 6.33 E+05 | 0.000 | 1.01 E+04 | - | - | - | - | - | - | - | - | - | - |
| *Whl. Inv- 20R, 10W, 2M* | 9.95 E+05 | 1.14 E+07 | 0.087 | 1.76 E+05 | 21.429 | 2.48 E+05 | 100.000 | 1.97 E+05 | 7.438 | 1.92 E+05 | - | - | - | - |
| *Mnf. Inv- 20R, 10W, 2M* | 1.86 E+07 | 1.16 E+08 | 0.161 | 1.57 E+06 | 21.429 | 2.92 E+06 | 7.500 | 2.07 E+06 | 14.754 | 2.02 E+06 | 16.071 | 1.84 E+06 | - | - |
| *Rtl. Inv- 20R,20W, 20M* | 6.75 E+04 | 9.51 E+05 | 0.071 | 1.46 E+04 | 10.112 | 1.92 E+04 | 9.184 | 1.59 E+04 | 7.317 | 1.50 E+04 | - | - | - | - |
| *Whl. Inv- 20R,20W, 20M* | 3.09 E+06 | 1.44 E+07 | 0.214 | 2.19 E+05 | 39.130 | 3.35 E+05 | 30.000 | 3.29 E+05 | 16.981 | 3.04 E+05 | 75.000 | 2.71 E+05 | 26.471 | 2.67 E+05 |
| *Mnf. Inv- 20R,20W, 20M* | 1.91 E+08 | 2.97 E+08 | 0.643 | 4.61 E+06 | 112.500 | 6.49 E+07 | 64.286 | 7.69 E+06 | 450.000 | 6.06 E+06 | - | - | - | - |
| *Rtl. Inv- 50R,50W, 50M* | 5.85 E+04 | 7.88 E+05 | 0.074 | 1.15 E+04 | 18.367 | 1.63 E+04 | 13.235 | 1.58 E+04 | - | - | - | - | - | - |
| *Whl. Inv- 50R, 50W,50M* | 2.66 E+06 | 9.74 E+06 | 0.273 | 1.56 E+05 | 34.615 | 3.35 E+05 | 112.500 | 2.65 E+05 | 75.000 | 1.88 E+05 | 450.000 | 1.87 E+05 | 60.000 | 1.86 E+05 |
| *Mnf. Inv- 50R,50W, 50M* | 6.90 E+07 | 1.04 E+08 | 0.666 | 1.49 E+06 | 81.818 | 9.31 E+06 | 33.333 | 3.18 E+06 | 52.941 | 3.12 E+06 | 150.000 | 2.70 E+06 | 40.909 | 2.64 E+06 |

# APPENDIX E: REFERENCE RATIO DETERMINATION

The extent of variability reduction due to aggregation of individual time series has to do with the cross-correlations among the individual time series. Reference ratio corresponds to the variability reduction, if there were no cross-correlation among them. Reference ratio is calculated by removing the cross-correlations among individual time series while preserving other characteristics. For this purpose, phase lags are introduced to each time series as follows.

Let $X_1(t), X_2(t), ..., X_n(t)$ be individual time series. Then, $\hat{X}(t) = \sum_{i=1}^{N} X_i(t)/n$ is the average time series. The actual ratio is calculated according to $\hat{X}(t)$. Let $Y_i(t)$ be the lagged version of $X_i(t)$. $Y_i(t) = X_i(t-\tau_i),\ for\ t \geq \tau_i$, where $\tau_i$ is selected randomly from a discrete uniform distribution with parameters 0 and 50. Then, $\hat{Y}(t) = \sum_{i=1}^{N} Y_i(t)/n$ is the average lagged time series. Then, the reference ratio is calculated using $\hat{Y}(t)$ and taking of variability ratios in 100 different random lag settings. Table E.1 gives ratios for the Fixed S policy order data.

Table E.1. Reference Ratios Calculated By the Algorithm for Fixed S Policy Order Data

| | | R2W2M2 | R4W4M4 | R20W10M2 | R20W20M20 | R50W50M50 |
|---|---|---|---|---|---|---|
| REF. RTL. RATIOS | std. dev. | 0.707 | 0.4978 | 0.2215 | 0.2219 | 0.1402 |
| | range | 0.6953 | 0.5022 | 0.2211 | 0.2295 | 0.1415 |
| REF. WHL. RATIOS | std. dev. | 0.6976 | 0.4955 | 0.3133 | 0.2211 | 0.1396 |
| | range | 0.9694 | 0.629 | 0.3558 | 0.2516 | 0.154 |
| REF. MNF. RATIOS | std. dev. | 0.7073 | 0.4961 | 0.7031 | 0.2204 | 0.1408 |
| | range | 0.9779 | 0.7116 | 0.788 | 0.2289 | 0.1418 |

# REFERENCES


Anderson, P.W., 1972, *More is Different*, Science, Vol. 177, No. 4047, 393-396.

Akkermans, H., 2001, *Emergent Supply Networks: System Dynamics Simulation of Adaptive Supply Agents***,** Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001.

Angerhofer, B.J. and M.C. Angelides, 2000, *System Dynamics Modeling in Supply Chain Management: Research Review*, Proceedings of the 2000 Winter Simulation Conference.

Barlas, Y., 1996, *Formal Aspects of Model Validity and Validation in System Dynamics*, System Dynamics Review, Vol. 12, No. 3, 183-210

Chen, F., Z. Drezner, J.K. Ryan, and D. Simchi-Levi., 2000, *Quantifying the Bullwhip Effect in a Simple Supply Chain: The Impact of Forecasting, Lead Times, and Information,* Management Science, Vol. 46, No. 3, 436-444

Demirel, G., 2006, *Aggregated and Disaggregated Modeling Approaches to Multiple Agent Dynamics*, Proceedings of the 24th International Conference of the System Dynamics Society.

Doran, Jim., 2006, *Agent Design for Agent-Based Modeling*, In Billari, F.C. (eds.), *Agent-Based Computational Modeling: Applications in Demography, Social, Economic and Environmental Sciences*. Physica-Verlag, Heidelberg.

Edmons, B., 2000, *Complexity and Scientific Modeling*, Foundations of Science, Vol. 5, No. 3, 379-390.

Forrester, J.W., 1961, *Industrial Dynamics*, Productivity Press, Portland.

Gavirneni, S., R. Kapuscinski, and S. Tayur, 1999, *Value of Information of Capacitated Supply Chains*, Management Science, Vol. 45, No. 1, 16–24.

Gershenson, C. and F. Heylingen, 2004, *How can we think the complex?*, In Richardson, K.A. (eds.), *Managing the Complex: Philosophy, Theory, and Practice*. Information Age Publishing.

Gilbert, N., 1995, *Emergence in Social Simulation*, In Gilbert, N., R. Conte (eds.), *Artificial Societies: The Computer Simulation of Social Life*, UCL Press, London.

Gilbert, N. and K.G. Troitzsch, 2003, *Simulation for the Social Scientist*, Open University Press, Buckingham.

Gündüz, A.B., 2003, *Strategies to Reduce Fluctuations in Supply Chains: A Dynamic Modeling Approach*, Boğaziçi University, Industrial Engineering Department, MS Thesis.

Lee, H.L., V. Padmanabhan, and S. Whang, 1997, *Bullwhip Effect in Supply Chains*, Sloan Management Review/Spring 1997.

Lee, H.L., and S. Whang, 1998, *Information Sharing in a Supply Chain*, International Journal of Manufacturing Technology and Management, Vol. 1, No. 1, 79-93.

Miller, J.H., and S.E. Page, 2007, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, Princeton University Press.

Moyaux, T., B. Chaib-draa, and S. D'AmourMoyaux, 2006, *Supply Chain Management and Multiagent Systems: An Overview*, In Chaib-draa, B., and J. Müller (eds.) *Multiagent Based Supply Chain Management*, Springer Berlin Heidelberg.

Panti, M., F. Pagliarecci, and L. Spalazzi, 2005, *An Agent-Based Supply-Chain Management*, Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems.

Parunak, H.V., R. Savit, R., R.L. Riolo, 1998, *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide*, In Sichman, J.S., R. Conte, N. Gilbert, (eds.) Proceedings of Multi-agent Systems and Agent-Based Simulation -MABS' 98.

Phelan, S.E., 1999, *A Note on the Correspondence between Complexity and Systems Theory,* Systems Practice and Action Research, Vol. 12, No. 3, 237-246.

Rahmandad, H., 2004, *Heterogeneity and Network Structure in the Dynamics of Contagion: Comparing Agent-Based and Differential Equation Models*, Proceedings of the 22nd International Conference of the System Dynamics Society.

Surana, A., S. Kumara, M. Greaves, U.N. Raghavan, 2005, *Supply-Chain Networks: A Complex Adaptive Systems Perspective*, International Journal of Production Research, Vol. 43, No. 20, 4235-4265.

Schieritz, N., and A. Grossler, 2003, *Emergent Structures in Supply Chains: A Study Integrating Agent-Based and System Dynamics Modeling*, Proceedings of the 36th Hawaii International Conference on System Sciences.

Scholl, H.J., 2001, *Agent-Based Modeling and System Dynamics Modeling: A Call for Cross Study and Joint Research*, Proceedings of the 34th Hawaiian International Conference on Systems Science.

Scholl, H.J., 2001, *Looking Across the Fence: Comparing Findings From SD Modeling Efforts with Those of Other Modeling Techniques*, Proceedings of the 19th International Conference of the System Dynamics Society.

Sterman, J.D., 2000, *Business Dynamics: Systems Thinking and Modeling for a Complex World*, Irwin McGraw-Hill.

M Vlachos, P. Yu, V. Castelli, 2005, *On Periodicity Detection and Structural Periodic Similarity*, Proceedings of the 5th International Conference on Data Mining.

Wilson, W.G., 1998, *Resolving Discrepancies Between Deterministic Population Models and Individual-Based Simulations*, American Naturalist, Vol. 151, No. 2, 116-134.

Wooldridge, M., N.R. Jennings, 1995, *Intelligent Agents: Theory and Practice*, The Knowledge Engineering Review, Vol. 10.

Xu, K., Y. Dong, and P.T. Evers, 2001, *Towards Better Coordination of the Supply Chain*, Transportation Research Part E: Logistics and Transportation Review, Vol. 37, No. 1, 35-54.

# REFERENCES NOT CITED

Anderson, P., 1999, *Complexity Theory and Organization Science*, Organization Science, Vol. 10., No. 3, 216-233

Axtell, R., J.M. Epstein, 1996, *Growing Artificial Societies: Social Science From the Bottom Up*, The MIT Press.

Billari, F.C., T. Fent, A. Prskawetz, and J. Scheffran, 2006, *Agent-Based Computational Modeling: Applications in Demography, Social, Economic and Environmental Sciences*, Physica-Verlag, Heidelberg.

Conte, R., N. Gilbert, and J.S. Sichman, 1998, *MAS and Social Simulation: A Suitable Commitment*, In Sichman, J.S., R. Conte, and N. Gilbert, (eds.), Proceedings of Multi-agent Systems and Agent-Based Simulation -MABS' 98.

Kohler, T. A., and G.J. Gumerman (eds.), 1999, *Dynamics in Human and Primate Societies: Agent-based Modeling of Social and Spatial Processes*, Oxford University Press.

Kimbrough, S.O., D.J. Wu, and F. Zhong, 2002, *Computers Play the Beer Game: Can Artificial Agents Manage Supply Chains?*, Decision Support Systems, Vol. 33, No. 3, 323-333.

Mosekilde, E., and E.R. Larsen, 1988, *Deterministic Chaos in the Beer Production-Distribution Model*, System Dynamics Review, Vol. 4, No. 1-2, 131-147.

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

Russell, S., and P. Norvig, 2003, *Artificial Intelligence: A Modern Approach*, Prentice Hall.