UNIVERSITY OF BERGAMO

Department of Mathematics, Statistics, and Computer Science
"Lorenzo Mascheroni"

Multivariate hedonic models for heterogeneous product prices in dynamic supply chains

PHD THESIS

to obtain the title of

Doctor of Philosophy

Specialty : Applied Mathematics for Business Science

Defended by

Gianfranco LUCCHESE

on April 18, 2012

Thesis Advisors: Jan VAN DALEN
Wolf KETTER

Institutional Affiliations:
Dept. Mathematics, Statistics, and Computer Science, University of Bergamo
Rotterdam School of Management, Erasmus University Rotterdam
Department of Quantitative Methods, University of Brescia

A Valeria, Caterina, e ai miei genitori.

Acknowledgments

I gratefully acknowledge the financial support for this project given by Italian Public Administration and in particular the Universities of Bergamo, Brescia, and Rotterdam.

I am also grateful to my supervisor, Dr. Jan van Dalen, for his encouragement and guidance, and to Dr. Wolf Ketter for providing the application data and for his help over the past three years.

In the same way, I wish to thank Professor Maria Grazia Speranza, Professor Marida Bertocchi, the entire Department of Mathematics in Bergamo for their tuition over the past three years.

Thanks to my parents, who always wished to read this thesis.

Thanks to my mentor and "model", Professor of Probability Enzo Orsingher.

Finally, I thank the support of my wife, Valeria, and my daughter, Caterina, who have followed me and at the same time kept in "line".

A heartfelt thank to Professor Luca Bertazzi of University of Brescia, Professor Jo van Nunen of University of Rotterdam, Dr. John Collins of University of Minnesota, Dr. Amy Greenwald of Brown University of Rhode Island, Professor David Stoffer of University of Pittsburgh, Professor Adelchi Azzalini of University of Padua, to the professors of Ph.D. courses in Bergamo and of LNMB, to the Ph.D. students and colleagues Alda, Antonio, Dario, Vincenzo, Yinyi, Annie, Muhammad, Rob, Pierpaolo, the LARGE group of Erasmus University, and all the students who have accompanied me in seminars and conferences.

Preface

The value of a good is determined by its utility. It is subjective and, in several cases, not determinable. If we consider the components of a product, we can imagine the value of it as a convolution of the individual component values, physical and non physical. For example, in a travel planning, the agency usually charges us costs for social club card and luggage warranty on a combination of flight and hotel for our favorite destination. Our mind goes immediately to an estimation of the single prices of each component. After a rapid extraction, we can add them obtaining our evaluation of the travel as a sum of characteristics. If we assume the additivity property for the value of goods, we can use the hedonic price to evaluate the components. It is defined as the value given by the buyer-consumer at the individual characteristics of the good. Its name comes from the concept of personal assessment of utility (pleasure) that usually is difficult to quantify using the money.

In a supply chain market the interest in parts value is not only for the customer but also for manufacturers. They assemble parts following a technological design and the demand for their products is linked to the components. Procurement prices offer optional costs for each component but they do not include customer evaluations. We are not so interested only in a new methodology for estimating hedonic prices taking into account their succession in time. This thesis includes a chapter for this goal. Rather, we shall show the utility of the hedonic information extracted from the market prices, improving the standard practices in a supply chain. The second and third chapters show new methodologies to be implemented in heterogeneous supply chains after the deconvolution phase. Since the basic idea of our method is very similar to the hedonic one, we designate with hedonic adjective our model. Which are the benefits of using hedonic information in the logistic operations of an heterogeneous supply chain? The core question of our work suffers of the same defects of the hedonic values. It is extremely egoistic because it hides the subjects of our analysis: the customers, the manufacturers, the suppliers of our supply chain. So, we can rewrite the question considering the pros and cons of hedonic prices for all our agents. What will happen if we assume a market where customers publicly state their evaluations of the characteristics of a product before the start of the production process? Maybe manufacturers would produce only the product including higher evaluated components. But in this case the suppliers would be tempted to increase prices for these parts. The effect is an acceleration of the supply chain coordination. We conclude this thesis with some proposals for future extension of our methodology, which can be adapt for many other scopes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, differently from the recent past, the market negotiations are often based on the auction system to price products and components, both in the procurement and consumer markets (Clay *et al.*, 2004). Someone affirms that it is the consequence of the increased use of technology. But we point the finger also on the extremely competitiveness of the modern markets, and on the extension of their geographical limits. In the sequel of our work, we will consider in our applications a multi-commodity supply chain where supply quantity is constrained by factory capacity, and materials availability. Here, the business-to-business (B2B) exchanges are often quick and unpredictable. For instance, a Tablet PCs manufacturer can cope in each production line almost 20000 units per year (90 units per day). If the demand were to grow suddenly he should be back from the suppliers as soon as possible for resources and semi-products. Then, he must face other manufacturers with the same problem and the option costs raise up. The complexity of these supply chains, for parts and products, affects the product pricing decisions, the inventory model, the procurement strategy, and many other aspects of logistics.

In this thesis, the hedonic extra premium (hedonic option prices) placed in the upgraded components are analyzed for a spectrum of products. The importance of these variables is linked to the preferences of the customers (Muellbauer, 1974). Hedonic prices represent the perfect indicators of the component value in the heterogeneous market and should give an idea of the rate between two versions of the same component. First of all, we estimate them with a deconvolution algorithm. Then, we may use those values to solve many logistics problems, from forecast phase to decisional phase. We state that utilizing hedonic variables in operations management can increase the value of the market information. A company must consider implicit prices to represent the evaluation of the characteristics and parts of a product in the market, overall when an oligopolistic system is established.

The need for a quick and accurate method to maximize the multivariate likelihood in state space models arises in many economical, industrial and financial problems (Lei, 1998; Wang *et al.*, 2011). We refer to the proposition in (Engle & Watson, 1981) which has been the main motivation for afford the topic of the thesis: "In contrast to the wide range of applications of the state space model with one measurement equation, there appear to be no time series applications that fully utilize the model when the dimension of input series is higher than one". After 30 years the situation is not changed. Few researchers point to the cross-structural models and estimation of parameters. Grid search methods for the estimation of the parameters was the solution in (Engle & Watson, 1981). Although the Expectation-Maximization (EM) technique (Dempster *et al.*, 1977) is quite effective for low dimensioned cases, new on line (sequential) methodology of inference about parameters are required for higher dimensioned multivariate cases (Ghahramani & Hinton, 2001).

Kalman filter–smoother technique including EM iterations, a procedure for extraction of information from time series, is reviewed. It is a recently developed technique to extract variables and parameters from multivariate time series. First contribution of the thesis is the innovative implementation of on line and off line algorithms for deconvolution in high dimensioned spaces based on the overall procedure of Kalman filter and EM. The introduction of a new and more effective methodology for stopping the convergence of the algorithm is given. Second contribution is the study of forecast performances of hedonic model with respect to the standard autoregressive models. Third contribution: a new methodology for the detection of the breaks in the time series analysis, based on the state variables, is introduced.

1.1 Heterogeneous Supply Chains under Oligopolistic Markets

Mass customization and the product variety push companies to cope the competitiveness of the market through the introduction of new strategies and tactics (Fogliatto & da Silveira, 2011). The problem of the value of the parts affects overall a specific type of heterogeneous supply chains, where procurement prices are unknown. According to the Supply Chain Council (http://www.supply-chain.org), a definition of a supply chain is: "The supply chain, a term now commonly used internationally, encompasses every effort involved in producing and delivering a final product or service, from the supplier's supplier to the customer's customer". Many properties of supply chain networks, like flexibility, dynamic, global, and complexity are analyzed in Simchi-Levi *et al.* (2003). The simplest example of heterogeneous supply chain is given by multi-tier supply chains where suppliers produce heterogeneous parts to be assembled by manufacturers in heterogeneous products. The latter sell the range of products to the retailers (three tiers) or directly to the customers (two tiers). There are many examples of this kind of supply chain markets in the real world, such as automotive, consumer appliances, electronic equipment, apparel and fashions. We classify them in three subgroups.

The heterogeneous supply chains under oligopolistic competition are the supply chains where there is a limited number of manufacturers. They can negotiate with suppliers for large number of parts and long contracts. In this way, they may acquire large power in the procurement market which reflects itself in the customer market. Usually, procurement prices in those kind of markets are unknown to the other manufacturers.

The heterogeneous supply chains with numerous manufacturers. The latter have the same negotiation power in the procurement market and each negotiation is independent by fidelity or long-term decisions, as an auction-based negotiation.

The heterogeneous supply chains under monopoly. Here there is only a manufacturer which buys components from suppliers and sells the end product independently. In the first group, there are the most quantity of real cases. For instance, the automotive industry must be classified in this group, though it includes a third tier, the car dealer. The second group includes the real estate market, some sectors of food and beverage industry, and every scheme where manufacturers are small or individual firms. The third group is recurrent in public industry as military, and space industry where the customers are the citizens of a country. In our thesis we will introduce some operations in the management for supply chain markets of the first group, the oligopolistic one. A scheme of similar heterogeneous supply chain networks is given in figure 1.1. There exists a large set of problems for such supply chains. Manufacturers are faced with decisions about the choice of customer to be satisfied and the right price for the revenue optimization, almost every day. Usually, they go on checking for the best factory where to buy component batches for assembling. They are interested in the optimal quantity of parts and products to stock, and in many other optimization problems. In the sequel, we decide to analyze only a small portion of them:

– the pricing problem. In the recent years, many researchers and many industries have provided dynamic pricing policies. To establish the right price in modern heterogeneous markets, companies collect demand data much more than in the past. In Internet, data warehousing allows to collect information not only about sales, but also about demographic data and customer preferences.

– the regime identification. Economic regimes (Ketter *et al.*, 2006, 2009, 2008) provide an intuitive method for characterizing and modeling market conditions. Initially proposed in the supply chain context by Ketter et al., it may be a useful instrument for real-time

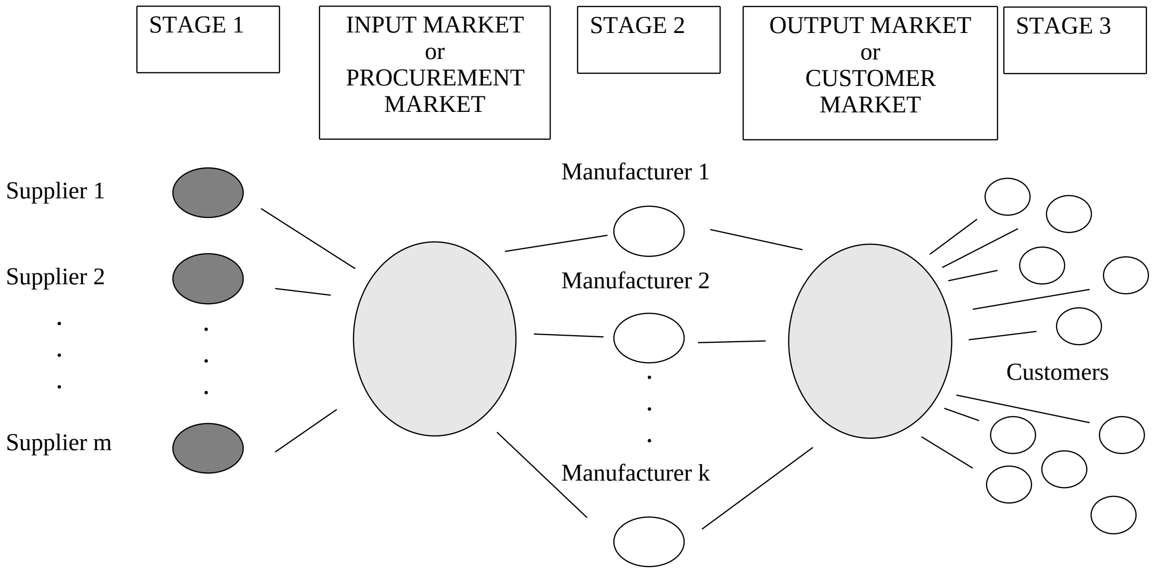

Figure 1.1: Scheme of heterogeneous supply chain markets with two markets, m suppliers, and k manufacturers. Customer market may be also segmented

strategies.

– Make-To-Order vs Make-To-Stock. Manufacturers want to fill customer orders quickly but, at the same time, they desire to keep the costs low for inventories (Gupta & Benjaafar, 2004). Often, end products are produced ahead of demand and kept in stock waiting for the arrivals of orders (Make-To-Stock, MTS). Differently, in the Make-To-Order (MTO), production starts only when customer order is received. The advantages and disadvantages of each mode of assembling refer to the supplier-customer lead times, the scheduling cycles, and other important logistics variables.

1.1.1 Business Tactics and Strategies in Heterogeneous Supply Chains

Recently studies reveal that the problem of the product differentiation has grown becoming one of the main topic in marketing and management sciences (Anderson *et al.*, 1992). In many cases, we have seen the birth of new techniques and the realizations of transformations in the production process. One example is given by the postponement management in a great number of supply chains due to the customization opportunities for the customers (Cheng *et al.*, 2010). "Postponement is about delaying the timing of the crucial processes in which the end products assume their specific functionalities, features, identities or personalities" (from Hau Lee in Gattorna, J.L. (1998)). Consider the way in which product variety is created:

the components can be seen as essential or optional, and the former can be aggregated in the base product, the simplest end-product in the variety (e.g., in a car the customer may install the air conditioned system as well as the extra air-bag system for rear passengers, both optional components). The base product is the origin of the postponement strategy. Manufacturers need a minimum stock of base products to guarantee demand satisfaction in the entire variety, and at the same time, they also have to implement good strategies in component assembling.

Another key driver in the product variety field is the unpredictable demand of customers, which can be assumed dependent on the prices (Dong *et al.*, 2009; Chen *et al.*, 2011). For manufacturers, and hence for retailers, it is more difficult to predict which of their products will be sold, and accordingly to plan productions and orders. In this case, forecast models are remained the same of twenty years ago, except for switching extensions. They often furnish inaccurate forecasts affecting the costs of the products and do not study co-movements between product prices.

The last problem is the sustainability of this kind of supply chain. Companies face inventory problem and quick obsolescence of products according to the variety of the products. Many times inventoried unsold products contribute to the scarcity of profits. Then, like for the "bullwhip" effect, also inventories for parts must be regulated under uncertainty of customer preferences.

There are a lot sectors where the problem of differentiation is stronger: for the electronic products (computers, mobile phones, appliances, tv), transport vehicles (cars, trucks), apparel & fashion, tourism, housing, energy, entertainment, food. See Wazed *et al.* (2008) for a review of journal papers about commonality in manufacturing).

1.1.2 Trading agent computer supply chain management

Perhaps the computer supply chain is the best example of oligopolistic supply chain with heterogeneous products and parts in the real world. US government, like other countries, captures the improvements in the performance of computers through the technological advances in the intermediate parts. Furthermore, the high-tech computer market occupies a large slice of the electronic commerce negotiations since it is characterized by quick and continuous changes in the preferences. Short life of computer model guarantees periodic refills of goods. In the real market there are several manufacturers which operate in this sector from years: Apple, Dell, Hp, Acer, Asus, are only the first groups in the world but they are subjected to rapid overturning. Computers are essentially a mix of components and each one representing in the standard customer a characteristic. For instance, the quantity of random access memory is synonymous of software performance, whereas the quantity of fixed memory, like a hard drive, represents the storage capacity for pictures, videos, and generic files. Since the birth of home computers, the market has been depicted component-dependent and customer preferences have been studied in the marketing science for processors, motherboards, video boards, and peripherals.

We want to underline several factors: firstly, the intense competition in the procurement markets; secondly, the nature of demand driven by the willingness of computer buyers to move to the next generations of products and software, that shorts the life of the products (high obsolescence). In fact, the shelf life for computers is around one year. Thirdly, the differentiated and segmented demand. Strangely, literature offers few effective multivariate models to forecast future product prices in these supply chain markets, while the interest in customer evaluation is relevant. Many websites offer price analysis in similar markets, because customer negotiations are often made via web. Some of them offer price analysis dependent on the performance of parts.

1.2 Multiagent Based Simulation: TAC SCM

In our thesis every application is made utilizing the multi agent based simulations (MAS) of supply chain markets. What is a MAS? An *agent* is a computer-human system placed in some environment, that is capable of autonomous actions in order to meet its designed objectives. We consider a dynamic market as environment and we study the behavior of multiple agents-manufacturers that compete in a computer supply chain.

The Trading Agent Competition for Supply Chain Management (TAC SCM) was conceived by Norman Sadeh in 2002 (Sadeh *et al.*, 2003). Initially, it was an experiment to mix risk management and artificial intelligence (AI), with the goal of testing new techniques for rationalization and optimization of logistic practices. Supply chain agents are modeled to operate according to its own objectives and policies. Every game consists of 220 days (or 44 five-day weeks), a virtual year of life for the computer variety. Six agents trade simultaneously in procurement and customer markets assembling 16 types of computer designed by the compatible combination of the basic parts: motherboard, central processing unit, random memory, and hard disk. After the last day of the game agents are sorted according the total profit, with remaining inventory valued at zero. We shall give a detailed description of TAC SCM in the next chapter, although a compendium about it may be downloaded via Internet on the proper website (Collins *et al.*, 2005). Basically, data coming out multi-agent simulations are one of the few testbed for such applications, where the collection of information is frequently not accessible. Each game provides time series for transparent and not

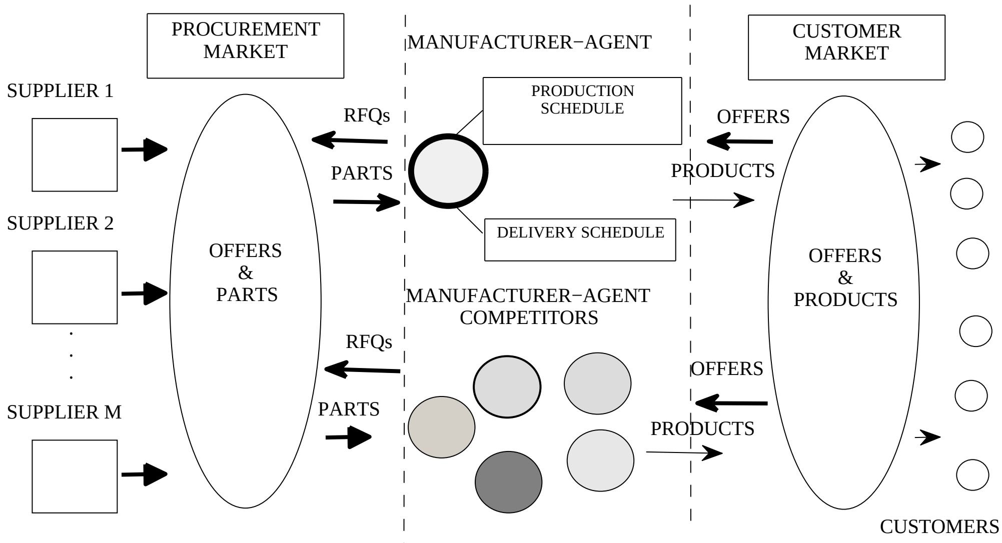

Figure 1.2: Multi-agent based simulations in supply chain markets, including M suppliers, and six human manufacturer-agents that compete with each other for a place in both markets. Only manufacturers are human agent. Customers and suppliers are rationally driven for optimization of revenue and utility

transparent variables such as request for quotes (RFQs), inventories for products and parts, customer daily demand for each model and manufacturer. For our present purpose, TAC SCM allow us to evaluate methodology and algorithm performances with a certain level of confidence. Furthermore, recently many scientific publications reports the need for preference variables and shadow prices in electronic request for quotes, eRFQs (Branco, 1997; Parkes & Kalagnanam, 2005). We point to design in a future a hedonic model for multidimensional auctions for products sharing a set of components.

1.3 Research Scope, Contributions, and Methodologies

1.3.1 Scope

Our research proposal has twofold target: to individuate new fields of application of hedonic information and to collect and organize the methodologies for implementing these applications. Research questions are:

– How can we estimate the hedonic values in heterogeneous dynamic supply chains?

– Can we apply those methodologies to real time framework?

– Which are the hedonic structural relations in a supply chain market?

– How is it possible to predict the dynamics of prices in a range of products through the analysis of their components and characteristics?

– Which is a good way to model several assumptions for parts utilizing hedonic information?

– Often, in data management and information technology applications, several variables and states remain unknown because they are unobservable. In which case can state space hedonic model help us to fill this emptiness? What's the meaning of these estimated variables?

It is innovative since it takes into account the whole information provided by a supply chain system, including the design of a product. In the past decades, there have been several attempts to set a good model for forecasting, inventories, and decision processes (Lee *et al.*, 2006; Song & Zhao, 2009) but state space models based on hedonic regression are rarely applied in this area. Within the forecast models our state space hedonic model can be considered an extension of these models when the structure of the prices is fundamental. They consider many relations between multiple variables and are increasingly evolving towards models with different distributional assumptions and time-varying parameters. Thanks to our mathematical and statistical background, we can achieve to implement innovative models of this kind in unusual areas like logistics and supply chain management. Table (we have to include it) clearly shows the growing interest for these models in business and economics areas, overall during the last months. This is surely due to the exit of many books in the last years that analyze the application in time series of state space methods

1.3.2 Contributions

Contribution 1 A complete specification of the dynamic multivariate hedonic model for heterogeneous supply chain markets with its variants.

What are the benefits of using hedonic information for decision strategies and logistics policies in supply chain management? Using hedonic prices and differentials we extract from the market the hidden latent variables representing customer preferences, dynamic pricing evolution, and regime categorization.

Contribution 2 An algorithm for the identification of the hyper parameter for a state space model with n dimensioned input and m dimensioned states.

Table 1.1: Contributions for subject and chapter

Contribution	Chapter	Fields
1	Chapter 2+3	Supply Chain Management Time Series Analysis
2	Chapter 2+4	Time Series Analysis Statistics
3	Chapter 5	Supply Chain Management Time Series Analysis Computer Science
4	Chapter 5	Time Series Analysis

After the identification of the potential of hedonic prices, two are the key factors for the implementation of the model. First, a good algorithm which extracts dynamically prices in off line and on line situations. The contribution is represented by the study of the behavior of the extraction algorithm in multivariate case and the introduction of new methodologies for stopping rule and structural break detection. Specifically, we consider certain parameters as nuisance and we focus on the transition matrix of the hedonic process.

Contribution 3 A complete specification of a framework for forecast product prices in a dynamic multivariate hedonic model for heterogeneous supply chain markets.

A good framework which uses strategically estimated hedonic values. We think at the implementation of the algorithm for price analysis in the consumer market but also for specific data mining applications. The hedonic information may bring advantages of the agent operations in every decision process and we will show the increasing in forecasting performances for medium/long term predictions.

Contribution 4 An on line forecast combination model in which weights are estimated via linear regression on the previous performances.

A new methodology for decision makers with multiple forecasts when daily observations are available. The contribution is completed with experiments showing the effectiveness of the learning properties of the model. It is the main model in the framework of Contribution 3.

Table 1.1 resumes the contributions in the thesis according to the chapter where each is treated and the fields covered by each of them.

1.3.3 Methodologies

Researchers rarely used hedonic information except in two famous topic of price indexing and consumer model. The main criticism is about the existence of a mapping between parts and products evaluation. We state that customers can only image product utility, whereas often they have a clear vision for component utility. The latter comes from previous experiences with products sharing similar parts. For instance, the evolution of desktop computer of the last decades has seen the “leitmotiv” of the identical design matrix which includes the CPU, RAM, HD, and MB. Users are accustomed to compare different desktop generations using performances analysis for parts and hence, hedonic regressions.

Dynamic hedonic model may be viewed as a variant of the general dynamic factor model (Forni *et al.*, 2000), or a variant of the temporal factor analysis (Cheung & Xu, 2003). The target of the first factor models in literature, was to reduce the number of variables for analysis and data-mining. While the factor analysis finds and gives a meaning to the principal components, here we have the knowledge of several factors, those that are correlated with real parts of the product. In many dynamic supply chain, the manufacturer-agent need for information about derived demand. For example, before you attempt to determine safety stock for components, you need to determine why you need it. It can be very difficult to manage the dependencies in the demand, especially when they change. Imbalances in supply and demand result in unsatisfied demands coupled with wasted supplies and efforts.

State space models are becoming so popular in literature because they consider underlying processes in the system formulation. We consider the case when the researcher does not know anything about parameters of the system. In multivariate case, with n variables for the signal and m variables for the states, the convergence between Kalman filter and likelihood maximization methodologies can be measured through a set of techniques. Likelihood ratio tests represent the common practice in univariate and low dimensioned cases. Our thesis offers different methodologies in high dimensioned cases improving parameter estimation.

Finally, we want to continue for future researches about a formulation of a new test for structural breaks in state space models. It is based on a selection of the multi-parameters based on the forecast performances, see Lutkepohl (2005) for details. Our hedonic algorithm offers a set of estimated parameters containing some outliers. We eliminate them via forecast analysis of each parameter for a large number of simulations. In this way, we can select the proper cluster of correct parameters representing the state space models with regimes.

1.4 Thesis Structure

The thesis is organized as follows:

Chapter 2 describes the concept of hedonic variables and implicit prices in the consumer model and price index formulation. Then, it introduces the dynamic hedonic model and the algorithm for the estimation of parameters. Stopping rules for algorithm convergence are analyzed with respect to time performances. Tests and properties for multivariate state-space models are described and discussed. We show an application with the typical output for the hedonic algorithm in TAC SCM (first application).

The content of the chapter was published in the proceeding of the 12th International Conference on Electronic Commerce, under the title *A Kalman Filter Approach to Analyze Multivariate Hedonics Pricing Model in Dynamic Supply Chain Markets*. Coauthors were Dr. Jan van Dalen, Dr. Wolfgang Ketter, and Dr. John Collins (van Dalen *et al.*, 2010).

Chapter 3 extends hedonic model to accept other formulations and variables, like the premiums. It introduces the dynamic hedonic noise model and the prediction algorithm for the estimation of the implicit prices. We show an application with the typical output for several of extended algorithms in TAC SCM. Some specifications in addition to the noise model are the p-lag hedonic model, and the hedonic-premium model. They are described in the sections 3.2-3.3.

The content of this chapter was not published because it needs of further investigations and applications.

Chapter 4 points to the construction of a real time algorithm based on the second contribution of the thesis. A generative model is applied to study the optimal calibration and measure the average approximation of the identified parameters. Two variants of the base algorithm are created. They will be tested in the chapter 5.

The content of this chapter will be used for the construction of the forecast module in supply chain markets where components are assembled in products, in chapter 5.

Chapter 5 is a collection of forecast models including hedonic variables, to predict product prices in a heterogeneous supply chain with a real time framework. In the first part of the chapter the difficulties in the implementation of a real time application and the specific technique in the algorithm for extracting the hedonic prices are given. We evolve in five algorithms which are used to extract hedonic information in real time analysis. Then, the mainstream of section 5.1 is a comparison with the standard autoregressive models, normally used in exploratory analysis of market trends. In this section, there are developed some indexes to measure performances and their meaning. In section 5.2 we show the innovative combination model with hedonic differentials, core of the entire chapter, and an application in TAC SCM.

Most of the content in this chapter has been submitted to Electronic Commerce Research and Applications (Elsevier, impact factor 1.946) for a journal article, under the title *A Multiple Forecast Model in Heterogeneous Supply Chain Markets Including Hedonic Prices for Components*. Coauthors were Dr. Jan van Dalen, Dr. Wolfgang Ketter, and Dr. John Collins. The contribution with respect to the econometric part of the chapter, was published in the proceeding of the 13th International Conference on Electronic Commerce, under the title *A Multiple Forecast Model in Supply Chain Market Including Hedonic Prices for Components*. Coauthors were Dr. Jan van Dalen, Dr. Wolfgang Ketter, and Dr. John Collins (van Dalen *et al.*, 2011).

At the end of the thesis, appendices provides an useful mathematical and statistical compendium for discrete-time linear systems, Kalman predictor, filter, and smoother estimates, and the expectation-maximization technique.

Chapter 2

Hedonic Information

2.1 Introduction

The use of hedonic models is a common approach in economics for dealing with the valuation of product components. This approach is rooted in household production theory (Lancaster, 1966; Muellbauer, 1974) to estimate consumer demand for heterogeneous products like houses, cars, computers, apparel or washing machines. In this chapter, it is our task to explain the concept of hedonic value and the existing methodologies to extract this value from the analysis of the markets. We also discuss the analogies of existing methodologies with our new formulation.

2.1.1 Literature Review

The hedonic technique is based on the assumption that quality differences among goods can be attributed to measurable characteristics, such as components and other product features. The shadow or implicit prices of these product characteristics (components) are estimated by regressing product selling prices on a relevant set of product characteristics in a sample of product varieties. Obviously, since they are implicit prices, they can be viewed as the coefficients of the objective function in the dual problem linked to price equilibrium in a market. Its origins date back to the beginning of the nineteenth century, when the problem of determining the automobile demand in the US market led the researchers to introduce characteristics of a good and customer preferences in their analysis (Court, 1939). The hedonic technique has been applied to construct quality-corrected consumer price indexes for cars (Van Dalen & Bode, 2004; Reis & Silva, 2006; Hartman, 1987), computers (Berndt & Rappaport, 2001; Schreyer, 2002), spreadsheets and database software (Gandal, 1994; Harhoff & Moch, 1997), durable goods (Gordon, 1990), paintings (Chanel *et al.*, 1996), wine

(Unwin, 1999), residential housing and real estate (Chinloy, 1977; Palmquist, 1980; Meese & Wallace, 2003; Case *et al.*, 2003); see (Triplett, 2006) for a review. All these cases are static regressions where dependent variables are the "quantity" of the components included in the product and the estimated coefficients represent the values of the characteristics varying the quantities. In one example, dynamic hedonic variables appeared: in the Dynamic Multiple Indicator-Multiple Cause model (DYMIMIC) of Engle and Watson (Engle *et al.*, 1985). The latter was apply to extract information for interest rate in the housing market. In the following of this section we will show the static hedonic model and the standard regrssion techniques to extract implicit prices for a single period. Our dynamic model will be introduced in the next section.

2.1.2 The Hedonic Consumer Model

In the consumer theory of Lancaster (1966), the characteristics model of the consumer is not based only on the price of goods. It has a dual representation. In its original form it is static and for this reason we will omit the time index. These are the assumptions about the model:

1. In a vector $\mathbf{x}$ we collect the n goods of the market which are related to some level of K activities $\mathbf{s}$ through the linear expression:

$$x_i = \sum_{k=1}^{K} a_{ik} s_k, \qquad \text{or} \qquad \mathbf{x} = \mathbf{A}\mathbf{s}. \tag{2.1}$$

2. In a vector $\mathbf{z}$ we collect the m characteristics of the goods produced by the same activities but with different technology:

$$z_j = \sum_{k=1}^{K} b_{jk} s_k, \qquad \text{or} \qquad \mathbf{z} = \mathbf{B}\mathbf{s}. \tag{2.2}$$

3. consumption technology matrices $\mathbf{A}$ and $\mathbf{B}$ are known in the market for each consumer.

4. each consumer is provided by rationality and he/she chooses according to the minimum price law.

In the simpler form the consumer decision is given by:

$$\min \mathbf{y}\mathbf{x} \ (= \mathbf{y}\mathbf{A}\mathbf{s}) \text{ subject to } \mathbf{B}\mathbf{s} = \mathbf{z}, \text{ and } \mathbf{z}, \mathbf{x}, \mathbf{y} \geq 0, \tag{2.3}$$

where the solution is the minimum price $\mathbf{y}^*$. The dual form of the same problem has a solution $\mathbf{v}^*$ such that:

$$\max \mathbf{vz}\ (= \mathbf{vBs}) \text{ subject to } \mathbf{vB} \leq \mathbf{yA}, \text{ and } \mathbf{z}, \mathbf{y} \geq 0. \tag{2.4}$$

The dual variables in the vector $\mathbf{v}$ represent the *implicit prices* of the m characteristics of the economy. We can estimate implicit prices or hedonic prices, only if the knowledge of the market is complete.

Now, examine the basic definitions deduced by Lancaster's consumer model. These concepts can be useful when hedonic prices are extracted by the researches.

Definition 1 *Product variety or differentiation.* It exists when some similar but not identical products form a product range sharing a set of characteristics related to consumers. In the consumer model of Lancaster, goods are intermediary transfer items for the characteristics, and production -consumption technologies are known.

Definition 2 *Characteristics spectrum* and *product differentiation curve.* The set of all characteristics combinations, which are available in the market, forms the characteristics spectrum. We can plot a curve for different resources to obtain the product differentiation curve.

Definition 3 *Suboptimal transfer in characteristics spectrum.* Usually, there is an efficiency substitution effect in the market of goods depending mainly by the technology matrices. "*If the consumer's optimal good (which would give him optimal transfer) is not available, there is some quantity of the next best good that will enable him to achieve the utility level he would have attained with some specific quantity of the optimal good.*" (Lancaster, 1990). Actually, the problem of substitution flexibility exists when preferred product stock out. Hence, this problem involves also inventory management of a supply chain.

Definition 4 *Optimal Compensation.* When a good is not available and in the customer preference list it was at the first place, another good could fully compensate the customer.

We have shown some concepts in consumer theory for characteristics included in the end product sold in the consumer market and the theory for the decision-making of the consumer depending on the parts of a range of products. Our review was developed in a static sense: customers evaluate parts according to a hedonic model not dependent on time. We will assume in the dynamic hedonic model a time relation for hedonic prices of the autoregressive type. In this sense, all the properties of the static model can be extended in the dynamic model. We will focus on the day by day evaluation of preferences and then, so that all the previous concepts and terminology will help the lecture.

2.1.3 The Hedonic Regression Methods in Quality Adjustment

In the price index models, hedonic prices influence the quality adjustment procedure in specific product ranges (housing, appliances, computers) and in a restrict number of countries (Us, Great Britain, New Zealand, Australia). Also in this case, the evaluations for the parts are static. Commodities of our supply chains, particularly consumer and producer durables, are sold in many varieties or models. Thus, at any time period, there are multiple prices - y_{it} - where i is the index of the type of product (e.g., the ID number of a SKU, the laptop with a wide screen, the Cinquecento two-door with 1.2 liter engine) and t is the time period (day, week, month, year) of observation. Prices of commodities usually vary in a range, which is dependent on the characteristics of goods (properties, dimensions, levels). One of the utilization of hedonic model is the construction of price indexes, which is spread in many developed countries such as USA, Australia, Canada.

Now, we examine the methodology for the construction of similar indexes. Let $\mathbf{y}_t$ be the vector of multiple prices in the specific period t, as a function of a set of characteristics $\mathbf{z}_t$, and some additional disturbance given by the multidimensional random variable ν:

$$\mathbf{y}_t = f_t(z_{1t}, z_{2t}, \ldots, z_{mt}, \nu_{it}) = f_t(\mathbf{z}_t) + \nu_t, \tag{2.5}$$

for $i = 1, \ldots, n$, where n is the number of products, and m the number of characteristics. Basically, $\mathbf{z}_t$ represents the vector of the amount of the characteristics, and the output vector $\mathbf{y}_t$ the selling prices. Note, that the quantities given by z_{jt} do not necessarily have to be positive values for each component. The existence of hedonic function in (2.5) is not guaranteed. In many cases, we can not find sufficient characteristics that fully explain the prices. But in the following of our work, we shall assume that such relation exists. Here, we define the price function including also other determinants:

$$\mathbf{y}_t = f_t(z_{1t}, \ldots, z_{mt}, v_{1t}, \ldots, v_{kt}, \nu_{it}) = f_t(\mathbf{z}_t, \mathbf{v}_t) + \nu_t, \tag{2.6}$$

where $\mathbf{v}_t$ is the k-dimensioned vector of price determinants unrelated to parts or characteristics. For instance, $\mathbf{v}_t$ may include some macroeconomic index as the gross domestic product.

When is there a necessity to include hedonic prices in the computation of price indexes? There are three cases:

(i) when there is an upgrade or change of the model during the sale period;

(ii) when the manufacturer replaces the product with a newest one;

(iii) when the product is not more available before of the end of the life's cycle.

During the observation period of price changes, each of upper cases may happen. Hence, the difference in qualities such as volume, function and properties between new and old goods, must be removed from price indexes. This is called "quality adjustment of price index", for retailers and producers, also in the harmonized version. The creation of a coherent method is topic, and here are the opportunities of cited National Statistic Institutes:

- Production cost. We can obtain an assessment of the cost of upgrade from the manufacturer;

- 50% Option cost. If there is a cost for purchasing the changed component or characteristic separately, then the fifty per cent is applied to obtain the new product price in the market. The reduction of 50 % is due to the fact that the cost of buying parts separately is usually greater than buying them as a package. Most of the time, previous experiences show the same average percentage of reduction in the package version;

- time dummy method. Here, the hedonic prices v_j are included as coefficients of component 0-1 dummy variables as in the relation:

$$Price = \alpha + \sum_{j=1}^{m} v_j z_j + \delta_1 t_1 + \delta_2 t_2 + \epsilon. \tag{2.7}$$

  The δ coefficients show the price index for time periods 1 and 2. The advantages of time-dummy approach are evident in many applications, overall on single dataset. The disadvantages are the lack of stability when employed over several datasets;

- Indirect hedonic method. It is the method applied by National Statistical Institutions to evaluate quality adjustment for electronic products, especially desktop and laptop computers. The hedonic regressions are calculated on monthly list price data from computer magazines and specialized websites. Data collected includes a set of characteristics as: processor speed (CPU Score), memory size (RAM Quantity), hard drive (HD), monitor size (Screen), type of disk reader (DVD, CD-RW, DVD-RW, Combo), ... In index approach, the function in (2.5) takes the semilogarithmic form, such that the coefficients represent the percentage increase (decrease) of the price caused by a unit change in the level of characteristics, as in:

$$\log y_{it} = a_{0t} + \sum_{j=1}^{m} a_{jt} z_{jt} + \nu_{it}, \tag{2.8}$$

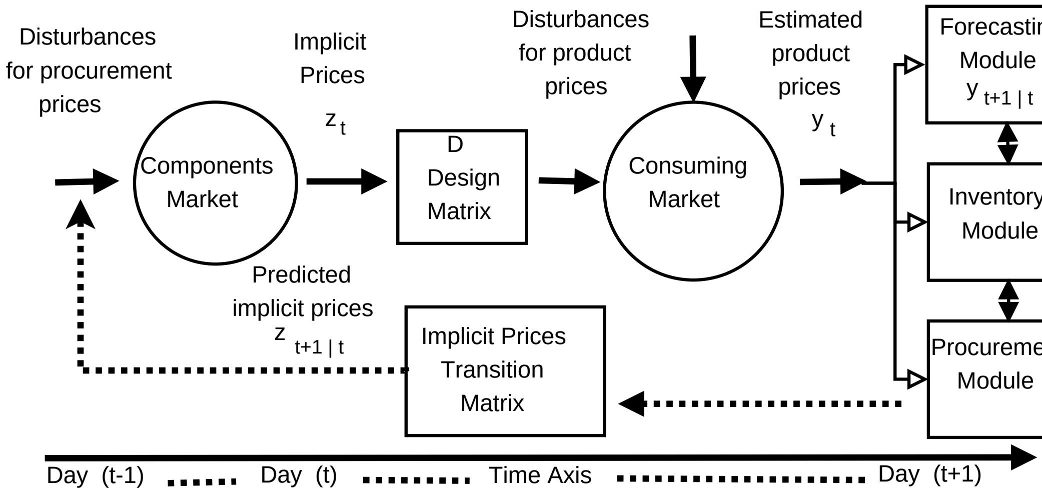

Figure 2.1: Illustration of the phases of the Kalman Filter in a dynamic model

We have seen how hedonic prices are extracted from prices of products through a regression. In our dynamic hedonic model we move on day by day extraction of customer evaluations.

2.2 Hedonic State-Space Model

We have seen the consumer model and the quality adjustment technique, two way to utilize hedonic prices. Now, we want to develop a hedonic model to dynamically describe and forecast the selling prices of a portfolio of product varieties in terms of the development of the implicit prices of shared option prices for components. In this model, we combine a standard static hedonic model of product prices with a vector autoregressive specification of implicit component prices. This results in a linear stochastic system or state-space model, in which the states reflect the implicit input prices. Figure 2.1 illustrates the main idea of our framework, where the input vector $\mathbf{z}_t$ represents the implicit prices and the output vector $\mathbf{y}_t$ the selling prices. Differently from previous section, $\mathbf{z}_t$ represent now the price and not the quantity of characteristic in the component. The state-space model has been applied in finance as well as macro economic analysis (Kellerhals, 2001; Harvey, 1989; Hamilton, 1994); see Watson & Engle (1983) for special cases and a classification. In the appendix we give some details of a state space model with discrete variables.

Next subsection introduces the multivariate hedonic model that relates product prices with product components and corresponding implicit prices. Assuming a time-dependent behavior of implicit prices, the resulting model can be viewed as a state-space model. Next subsections discusses the Kalman-filter approach to estimating this model, as well as tests to evaluate the model.

2.2.1 From a Static Model to a Dynamic Model Formulation.

The idea of the multivariate hedonic model is that observed product prices jointly vary with customer valuations of the constituting product parts, and that these implicit component valuations evolve over time. Specifically, we assume that the observed prices of n related products offered on day t are available in an $n \times 1$-vector $\mathbf{y}_t$. For practical purposes, we assume that each period generates a single product price (which obviously may not always be the case). Further, we introduce an $m \times 1$-vector of latent factor prices $\mathbf{z}_t$, with $m \leq n$. If the factors are synonymous with product components, then $\mathbf{z}_t$ contains the implicit component prices. An $n \times m$ design matrix $\mathbf{D}$ maps the component prices to the product prices. As each product is composed of a fixed set of components, $\mathbf{D}$ is non-stochastic.

Price formation takes place in the consumer market where products are sold. The observed market situation each day is a complex mix of stochastic consumer demand processes, and agent offer policies. Two assumptions are advanced to capture the relevant features of the emerging product prices. First, products are basically bundles of branded components, and the realized product prices can therefore be interpreted as an aggregate of implicit component prices.[1] Second, the implicit component prices evolve in an autocorrelated, possibly non-stationary way over time, which may be formalized as:

$$\mathbf{z}_t = \mathbf{\Phi} \mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t \tag{2.9}$$

Here, $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\varepsilon}})$ is an $m \times 1$-vector of random disturbances in the component price evaluation process, which are uncorrelated over time. It reflects the unobserved consequences of demand idiosyncrasies and manufacturer dependent supply conditions. The Gaussian form of the distribution is not a strict requirement. We will see that it may be substituted by a general distribution without problems. In addition, the relation between observed product prices $\mathbf{y}_t$ and latent component prices $\mathbf{z}_t$ is formalized by means of a hedonic model with a fixed design matrix $\mathbf{D}$:

$$\mathbf{y}_t = \mathbf{D} \mathbf{z}_t + \boldsymbol{\nu}_t \tag{2.10}$$

The $\boldsymbol{\nu}_t$ is an $n \times 1$-vector of random disturbances in the measurement process, which are again assumed to be normally distributed $\boldsymbol{\nu}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_\nu)$. It captures unexpected product price variation not related to product characteristics, random demand variations (RFQs), and variation in price bids by different manufacturers. If the measurement process is perfect,

[1] In our empirical application to TAC SCM, individual product components can not be sold in the consumer market, but used only for production, which may be different from real markets.

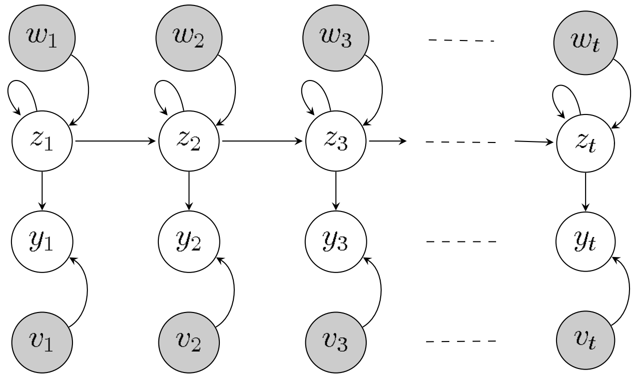

Figure 2.2: A directed acyclic graph with disturbances (gray circles). $\mathbf{z}_t$ is the implicit value for the vector of components. $\mathbf{y}_t$ is the series of prices for the final products. Circular arrows indicate cross relations between the variables of the vector

then the distribution of $\boldsymbol{\nu}_t$ is obviously degenerate, and the measurement model simplifies to $\mathbf{y}_t = \mathbf{D}\mathbf{z}_t$. If the measurement process is not perfect, then the assumed behavior of $\boldsymbol{\nu}_t$ affects the implicit price behavior (2.9). The process disturbances ε_t and measurement disturbances $\boldsymbol{\nu}_t$ are assumed to be independently distributed, $E(\boldsymbol{\varepsilon}_t\boldsymbol{\nu}_t') = \mathbf{0}$. Obviously the model greatly simplifies if the disturbances can be assumed to be independent:

$$\boldsymbol{\Sigma}_\nu = \sigma_\nu^2\mathbf{I} \qquad (2.11)$$

$$\boldsymbol{\Sigma}_\varepsilon = \sigma_\varepsilon^2\mathbf{I}. \qquad (2.12)$$

These conditions are appropriate for uncorrelated product prices and implicit prices; they may be considered similar to the classical multivariate regression assumptions (see Hamilton, 1994).

2.2.2 The EM Algorithm Mixed with Kalman Filter

Equations (2.9) and (2.10) form a state-space model. This model has often been applied to problems of control engineering, but since the nineties also to various fields of economics. The unobserved implicit component prices (or states) in this model are estimated by means of the Kalman-filter approach with smoothed estimators, that is using the entire sample of product prices, $\mathbf{y_1}, \ldots, \mathbf{y_T}$. Differently from signal processing implementations, where the dimension of the vector y is often unitary, in component assembling for manufacturing many times $m < n$. Our model consists of an observation equation (2.10) with a measurement or design matrix $\mathbf{D}$, and a state equation (2.9) with transition matrix $\boldsymbol{\Phi}$. In the dynamic linear model, the process starts in period 0 with implicit prices $\mathbf{z}_0$, which are assumed to be normally distributed with mean μ_0 and $m \times m$ covariance matrix $\mathbf{\Sigma}_0$. If we had observed the real implicit prices (states) $\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$, as well as the product prices $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$, then the parameters of the model could be estimated by maximizing the likelihood function:

$$f(\mathbf{z_i}, \mathbf{y_i}) = f_{\boldsymbol{\mu}_0 \mathbf{\Sigma}_0}(\mathbf{z_0}) \prod_{t=1}^{T} f_{\mathbf{\Phi}, \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}}(\mathbf{z}_t | \mathbf{z}_{t-1}) \prod_{t=1}^{T} f_{\mathbf{\Sigma}_{\boldsymbol{\nu}}}(\mathbf{y}_t | \mathbf{z}_t). \tag{2.13}$$

However, since we do not have the complete data, we adopt the EM-algorithm described by Dempster, Laird, and Rubin (1977) to estimate the unknown parameters. All the computations for the maximization of the likelihood will be treated in the next subsection.

The Kalman filter algorithm (see details in appendix, in section A.2) estimates values for the hedonic prices. We apply the formula from (A.12) to (A.19). In this way, an estimate of $\mathbf{z}_t$ is obtained from observations $\mathbf{y}_t$ for every day t, $1 \leq t \leq T$, where T is the last time of our series. The Kalman filter and smoother are applied to all series in our analysis, even when not stationary. The calculations can be done in real time and even in off-line situations by the maximization of (2.13). The calculations involved are based on the following algorithm.

Our algorithm uses the $n \times 1$-vectors of product prices over the time frame $(0, T)$ as inputs. In Shumway & Stoffer (2006, 1982), a similar algorithm is described to estimate the smoothed values of the (expected) implicit prices, as well as the other model parameters by means of an EM-algorithm. Instead of using the Newton-Raphson method involving the Hessian of the inverse errors matrix, we apply the algorithm described in Dempster, Laird and Rubin Dempster *et al.* (1977), which offers convenient solutions most of the time. Our choice is quite forced and motivated by the inclusion of multivariate equations. In fact, when the number of variables is large, EM is able to reach an acceptable solution differently from Newton-Raphson, interior point method, and the simplex of Nelder-Mead, overall in a stable way (Wu, 1983). A weak point is the slow convergence for the solution.

We summarize the unknown parameters of the model (2.9)-(2.10) in a single vector $\mathbf{\Theta} = \{\boldsymbol{\mu}_0, \mathbf{\Sigma}_0, \mathbf{\Phi}, \mathbf{\Sigma}_{\boldsymbol{\nu}}, \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}\}^T$, which is estimated by means of maximum likelihood using (2.13). For the initial step, we assume starting values $\mathbf{\Theta}^{(\mathbf{0})}$, where the superscript in parentheses refers to the iteration number. At every step we determine an augmented estimate of $\mathbf{\Theta}$ using the following EM-Kalman filter algorithm (EM+KF):

a. Choose the initial values for $\mathbf{\Theta}$, $\mathbf{\Theta}^{(\mathbf{0})}$.

b. On iteration i, with $i = 1, 2, \ldots$:

  b1. apply the Kalman filter and smoother to find values for $\mathbf{z_t}$ using the equations (A.12)-(A.19), under the assumption of $\mathbf{\Theta} = \mathbf{\Theta}^{(\mathbf{i-1})}$;

b2. with these values, apply maximum likelihood to find a new estimate of $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{(\mathbf{i})}$;

b3. if the algorithm converges, repeat one time the steps (b1) to find final values for $\mathbf{z_t}$ and then go to (c), otherwise update $\boldsymbol{\Theta}$ and return to (b1) for a next iteration;

c. test for the results.

Step (a) sets an appropriate starting value $\boldsymbol{\Theta}^{(\mathbf{0})}$, which will be gradually modified satisfying the model equations at every iteration. Step (b) computes smoothed values $\mathbf{z_t}^T$ and their estimated variance-covariance, $\mathbf{P_t^T}$ using $\boldsymbol{\Theta}^{(\mathbf{i-1})}$. With these estimates, the algorithm performs the expectation step of the EM technique, finding the expectation of the likelihood function (2.13). The maximization step updates the estimate of multi-parameter and saves it in $\boldsymbol{\Theta}^{(\mathbf{i})}$. Now, we show how to apply the E-Step of the EM procedure using the hyperparameter $\boldsymbol{\Theta}^{(\mathbf{j})}$, the vector $\mathbf{z_t^T}$, the matrices $\mathbf{P_t^T}$ and $\mathbf{P_{t,t-1}^T}$. We have, under the Gaussian assumption, the complete data log-likelihood respect to (2.13) can be written as:

$$-2\ln L_{\mathbf{z},\mathbf{y}}(\boldsymbol{\Theta}) = \ln|\boldsymbol{\Sigma_0}| + (\mathbf{z_0}-\boldsymbol{\mu_0})'\boldsymbol{\Sigma_0^{-1}}(\mathbf{z_0}-\boldsymbol{\mu_0})+$$

$$+\mathbf{T}\ln|\boldsymbol{\Sigma_\varepsilon}| + \sum_{\mathbf{t=1}}^{\mathbf{T}}(\mathbf{z_t}-\boldsymbol{\Phi}\mathbf{z_{t-1}})'\boldsymbol{\Sigma_\varepsilon^{-1}}(\mathbf{z_t}-\boldsymbol{\Phi}\mathbf{z_{t-1}}) + \mathbf{T}\ln|\boldsymbol{\Sigma_\nu}| + \sum_{\mathbf{t=1}}^{\mathbf{T}}(\mathbf{y_t}-\mathbf{D}\mathbf{z_t})'\boldsymbol{\Sigma_\nu^{-1}}(\mathbf{z_t}-\mathbf{D}\mathbf{z_t}),$$

and, at iteration j, we consider the maximization of the expectation of the log likelihood given by:

$$Q\left(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(j-1)}\right) = E\left\{-2\ln L_{\mathbf{z},\mathbf{y}}\left(\boldsymbol{\Theta} \mid \mathbf{y_T}, \boldsymbol{\Theta}^{(j-1)}\right)\right\} =$$
$$= \ln|\boldsymbol{\Sigma_0}| + \ \mathrm{tr}\{\boldsymbol{\Sigma_0^{-1}}[\mathbf{P_0^T} + (\mathbf{z_0}-\boldsymbol{\mu_0})(\mathbf{z_0}-\boldsymbol{\mu_0})']\} +$$
$$+\mathbf{T}\ln|\boldsymbol{\Sigma_\varepsilon}| + \ \mathrm{tr}\ \{\boldsymbol{\Sigma_\varepsilon^{-1}}[\mathbf{S_{11}} - \mathbf{S_{10}}\boldsymbol{\Phi}' - \boldsymbol{\Phi}\mathbf{S_{10}'} + \boldsymbol{\Phi}\mathbf{S_{00}}\boldsymbol{\Phi}']\} + \mathbf{T}\ln|\boldsymbol{\Sigma_\nu}| +$$
$$+\ \mathrm{tr}\ \{\boldsymbol{\Sigma_\nu^{-1}}\sum_{\mathbf{t=1}}^{\mathbf{T}}[(\mathbf{y_t}-\mathbf{D}\mathbf{z_t})(\mathbf{y_t}-\mathbf{D}\mathbf{z_t})' + \mathbf{D}\mathbf{P_t^T}\mathbf{D}']\}, \tag{2.14}$$

where

$$\mathbf{S}_{11} = \sum_{t=1}^{T}\left(\mathbf{z}_t^T\mathbf{z}_t^{T\prime} + \mathbf{P}_t^T\right), \tag{2.15}$$

$$\mathbf{S}_{10} = \sum_{t=1}^{T}\left(\mathbf{z}_t^T\mathbf{z}_{t-1}^{T}{}' + \mathbf{P}_{t,t-1}^T\right), \tag{2.16}$$

$$\mathbf{S}_{00} = \sum_{t=1}^{T}\left(\mathbf{z}_{t-1}^T\mathbf{z}_{t-1}^{T}{}' + \mathbf{P}_{t-1}^T\right), \tag{2.17}$$

and $\mathbf{P}_{t,t-1}^T$ is the covariance smoother for lag-one values (Shumway & Stoffer, 2006). After the smoother calculations for $\mathbf{K}_t$ in (A.14), $\mathbf{J}_t$ in (A.17), $\mathbf{P}_t^t$ in (A.16), and the last value for the variance-covariance matrix for the smoothed errors in (A.18), $\mathbf{P}_T^T$, the covariance smoother for lag-one values is given by:

$$\mathbf{P}_{t-1,t-2}^T = \mathbf{P}_{t-1}^{t-1}\mathbf{J}_{t-2}' + \mathbf{J}_{t-1}\left(\mathbf{P}_{t,t-1}^T - \mathbf{\Phi}\mathbf{P}_{t-1}^{t-1}\right)\mathbf{J}_{t-2}', \quad \text{for } t = T, T-1, \ldots, 2, \tag{2.18}$$

whereas the first value is given by:

$$\mathbf{P}_{T,T-1}^T = (\mathbf{I} - \mathbf{K}_T\mathbf{D})\,\mathbf{\Phi}\mathbf{P}_{T-1}^{T-1}. \tag{2.19}$$

For a detailed explanation and the derivation of the algorithm and formula we refer to the book of Shumway & Stoffer (2006).

Calculate (2.14) is the expectation step of EM algorithm. The maximization step is the minimization of the same quantity to update the hyperparameter values. The assumption of Gaussian distribution for disturbances helps to facilitate the M-step. It is sufficient to set equal to zero the first derivative of (2.14) respect to each parameter of $\mathbf{\Theta}$ and we obtain individual expression for each of them forming a linear system. In fact, the above argument only tells us that estimated parameters give a stationary point of the likelihood. But in the case of multivariate normal distribution, as for the expectation of likelihood, we have that stationary points obtained are in fact overall maxima (see the theorem 4.2.1, page 104 in Mardia *et al.*, 1979). Closed form expressions for the new estimates are given by:

$$\mathbf{\Phi}^{(j)} = \mathbf{S_{10}}\mathbf{S}_{00}^{-1}, \tag{2.20}$$

$$\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(j)} = T^{-1}\left(\mathbf{S}_{11} - \mathbf{S}_{10}\mathbf{S}_{00}^{-1}\mathbf{S}_{10}'\right), \tag{2.21}$$

$$\mathbf{\Sigma}_{\boldsymbol{\nu}}^{(j)} = 1/T\sum_{t=1}^{T}\left[\left(\mathbf{y_t} - \mathbf{D}\mathbf{z_t^T}\right)\left(\mathbf{y_t} - \mathbf{D}\mathbf{z_t^T}\right)' + \mathbf{D}\mathbf{P}_t^T\mathbf{D}'\right], \tag{2.22}$$

$$\boldsymbol{\mu}_0^{(j)} = \mathbf{z}_0^T, \quad \text{and} \quad , \quad \mathbf{\Sigma}_0^{(j)} = \mathbf{P}_0^T, \tag{2.23}$$

and we use them in every outer iteration of the algorithm.

To measure convergence in step (c), we implement three stopping rules depending on: (i) the distance between the state-transition matrix, (ii) the incomplete-data likelihood function, and (iii) the distance between product prices in two successive iterations. The first rule is based on:

$$\mathbf{n^{(1)}}(\mathbf{\Phi^{(j)}}, \mathbf{\Phi^{(j-1)}}) = \|\mathbf{\Phi^{(j)}} - \mathbf{\Phi^{(j-1)}}\| < \delta_1, \tag{2.24}$$

where δ_1 is a scalar depending on the type of distance selected in the (2.24). It stops

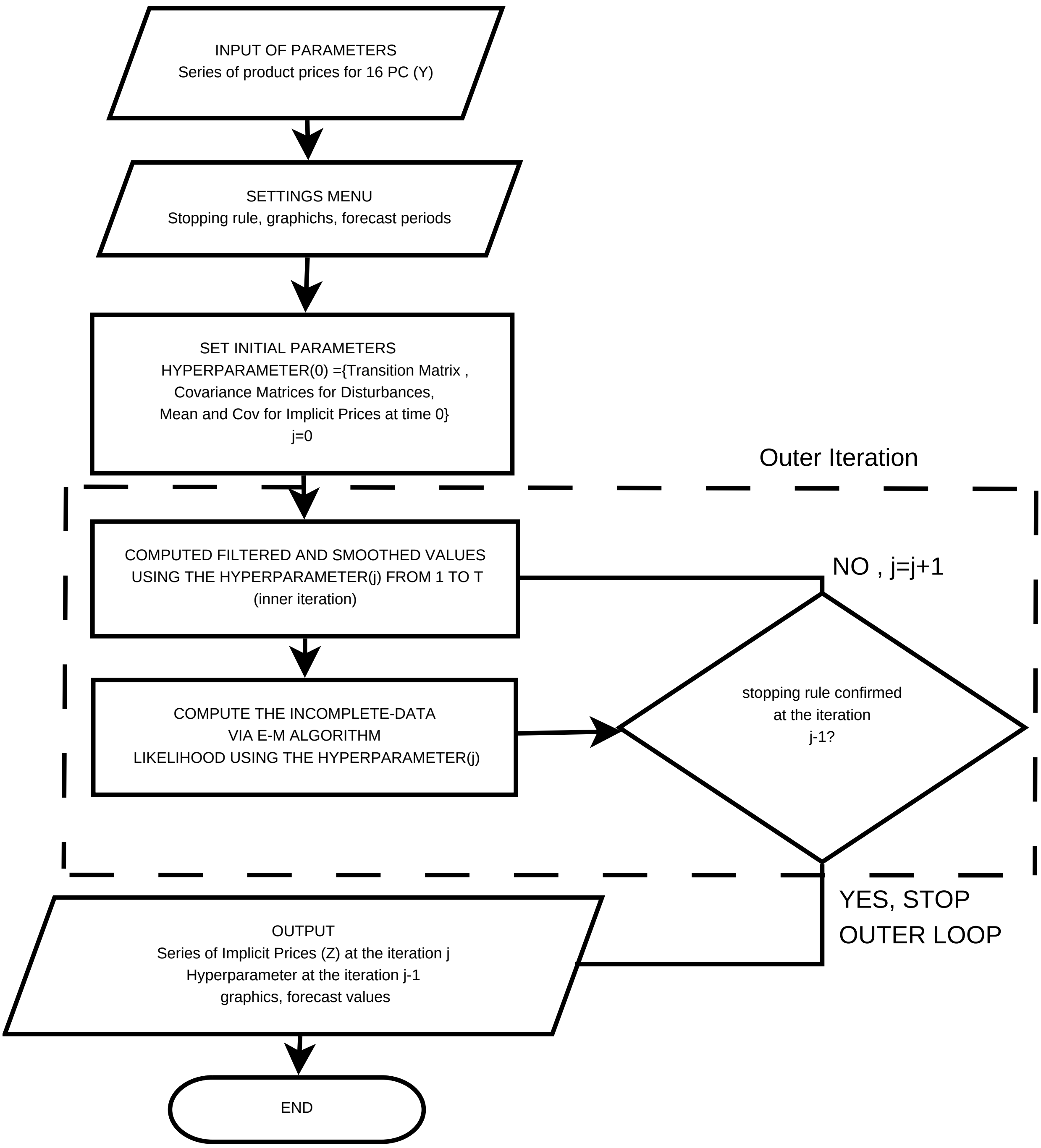

Figure 2.3: Illustration of the steps and iterations of the hedonic algorithm in a dynamic model

the algorithm when the estimated state-transition matrix, $\mathbf{\Phi}$ hardly changes between one iteration and the other. With this rule, interest is in the dynamics of the implicit prices. Common values for δ_1 are in the order of $10^{-2} \cdot m^2$ to $10^{-4} \cdot m^2$, bounds obtained by a multiplication of all elements of the transition matrix multiplied by an average error. Only distance between transition matrices is used in the (2.24) because other estimated parameters are less interesting in sense of pattern of hedonic prices dependencies. The covariance matrix $\mathbf{\Sigma}_\nu$ is related to the premiums of product prices, the difference between product and hedonic price. It provides an idea of agent dynamic strategies for profit. The covariance matrix $\mathbf{\Sigma}_\varepsilon$ is the noise covariance of hedonic process. We will see that customer evaluations rarely show large matrices for those errors. Thus, transition matrix is the main multi-parameter for a dynamic analysis of hedonic prices, and for this reason we based the first stopping rule on it.

The second rule is based on the relative likelihood:

$$\mathbf{n}^{(2)}\left(f_{\mathbf{\Theta}^{(j)}}(\mathbf{y}), f_{\mathbf{\Theta}^{(j-1)}}(\mathbf{y})\right) = \frac{f_{\mathbf{\Theta}^{(j)}}(\mathbf{y})}{f_{\mathbf{\Theta}^{(j-1)}}(\mathbf{y})} < \delta_2. \tag{2.25}$$

This takes into account all model parameters. When δ_2 is close to one, subsequent iterations will not yield substantial changes and the algorithm stops. The incomplete-data likelihood is defined as:

$$L_Y(\mathbf{\Theta}) = f_{\mathbf{\Theta}}(\mathbf{y}) = (2\pi)^{-1}|\mathbf{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{e}_\mathbf{t}'\mathbf{\Sigma}_\mathbf{t}^{-1}\mathbf{e}_\mathbf{t})\right\}, \tag{2.26}$$

where $\mathbf{e}_t$ is the vector of the innovations (independent Gaussian random variables), and the covariance matrices are given by:

$$\mathbf{\Sigma}_\mathbf{t} = \mathbf{D}\mathbf{P}_\mathbf{t}^{\mathbf{t-1}}\mathbf{D}' + \mathbf{\Sigma}_{\boldsymbol{\nu}}. \tag{2.27}$$

In this way, the stopping rule is related to the likelihood ratio test problem (Azzalini, 1996), given by:

$$\begin{cases} H_0: & \mathbf{\Theta} = \mathbf{\Theta}^{(j)} \\ H_1: & \mathbf{\Theta} = \mathbf{\Theta}^{(j-1)} \end{cases} \tag{2.28}$$

For this testing problem, it is usual to base a decision rule for acceptance or rejection, on the ratio of the likelihoods in two close iterations:

$$\lambda(y) = \frac{f_{\mathbf{\Theta}^{(j)}}(\mathbf{y})}{f_{\mathbf{\Theta}^{(j-1)}}(\mathbf{y})}, \tag{2.29}$$

called the *likelihood ratio*. Intuitively, if λ is high, we shall prefer to accept H_0, whereas λ is low, then we shall opt for H_1. Test procedure does not change for monotonic transformation of the test statistic and the corresponding critical value. Therefore, an equivalent test statistic is:

$$W(y) = -2 \ln \lambda(y), \tag{2.30}$$

which is still called the likelihood ratio. But the sense of statistic procedure W is the reversed of the previously explained, since the transformation is decreasing. Therefore, in our code, we may use the incomplete-data log-likelihood ignoring the constant term, which is defined as:

$$-\ln L_Y(\mathbf{\Theta}) = \frac{1}{2}\sum_{t=1}^{T} \log|\mathbf{\Sigma_t}| + \frac{1}{2}\sum_{t=1}^{T} \mathbf{e}_t' \mathbf{\Sigma}_t^{-1} \mathbf{e}_t. \tag{2.31}$$

Then, the second stopping rule may be defined as:

$$\mathbf{n}^{(2)}\left(\mathbf{\Theta}^{(j)}, \mathbf{\Theta}^{(j-1)}\right) = -2\{\ln L_Y(\mathbf{\Theta}^{(j)}) - \ln L_Y(\mathbf{\Theta}^{(j-1)})\} < \delta_2'. \tag{2.32}$$

According Neyman–Pearson methodology, it is the test procedure with maximum power among all test procedures with the same significance level α for the hypotheses (2.28) when the number of observations is large. Whereas (2.24) measures the deviation of one of the parameters, similarly at the Wald test statistic (see (4.13), pp.113 in Azzalini for a description), the convergence criteria in (2.25) measures the distance between the likelihood computed at $\mathbf{\Theta}^{(j)}$ and at $\mathbf{\Theta}^{(j-1)}$. Then, it takes into account of all parameters of the process. When the unknown parameter is multi-dimensional of order k, the asymptotic distribution of (2.30) is at least χ^2_k. In fact, some result of Bartlett (1954) improves the distribution approximation and corrects the number of degrees of freedom (d.f.) when we want to test the mean vector hypotheses. Details and proofs about asymptotic distributions of likelihood ratio test are given in Serfling (1980, Chapter 4). The degrees of freedom of the chi-square distribution should be at least set as the number of variables in the system minus the number of constraint for them, $n - m$. This point is the core of the second research contribution of the thesis: we will show how the likelihood ratio rule is too much rapid to stop the convergence, and it may not provide optimal parameters. The problem of KF+EM algorithm is the not increasing of the statistics in (2.30) due to the jumps of the Kalman gain in the nearing of the solution. Also in the case of Gaussian distributions, and for high degrees of freedom (e.g. for $d.f. = 5$). Exact computation of the distribution of $W(y)$ in (2.30) is not easy, and specific methods usually are used. Our methodology consists to try several values for the number of degree of freedom for the chi-square distribution.

The third rule is based on the differences between predicted product prices:

$$\mathbf{n^{(3)}}\left(\mathbf{y_t^{(j)}},\mathbf{y_t^{(j-1)}}\right)=\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbf{n_{t,i}^{(3)}}(\mathbf{y_t^{(j)}},\mathbf{y_t^{(j-1)}})<\delta_3 \tag{2.33}$$

where $\mathbf{n_{t,i}^{(3)}}(\mathbf{y_t^{(j)}},\mathbf{y_t^{(j-1)}})=|y_{i,t}^{(j)}-y_{i,t}^{(j-1)}|$. It considers the differences between predicted product prices, in close iterations, obtained via (2.10). This rule may be relevant for on line applications of the algorithm, when interest is in the similarity between predicted product prices used for forecasting. Note that the absolute distance is not normalized for the range of product prices.

The contribution of the convergence criteria in our methodology is relevant since only the behavior of the second rule is well reported in literature. The three stopping criteria differing in behavior and distribution. If $\mathbf{\Theta}$ is k-dimensional the only asymptotic result is for second rule (chi square distribution). Because in multivariate field convergence is really complicated for the properties of the likelihood function (multiple critical points), we shall see the opportunities furnished by observing the behavior of each stopping rule.

We now wish to derive the relationship between first and second stopping rule. Expanding the likelihood in $\mathbf{n}^{(2)}$ as a Taylor series about $\hat{\mathbf{\Theta}}$, the generic parameter, we obtain:

$$\mathbf{n}^{(2)}\left(\mathbf{\Theta}^{(j)},\hat{\mathbf{\Theta}}\right)=-2\{\ln L_Y(\mathbf{\Theta}^{(j)})-\ln L_Y(\hat{\mathbf{\Theta}})\}=$$

$$=-2\{(\mathbf{\Theta}^{(j)}-\hat{\mathbf{\Theta}})l'_Y(\hat{\mathbf{\Theta}})+\frac{1}{2}(\mathbf{\Theta}^{(j)}-\hat{\mathbf{\Theta}})^2 l''_Y(\tilde{\mathbf{\Theta}})\}, \tag{2.34}$$

where $\tilde{\mathbf{\Theta}}\in(\hat{\mathbf{\Theta}},\mathbf{\Theta}^{(j)})$, and $l'(l'')$ is the first (second) derivative of the log-likelihood. Since we have found $\hat{\mathbf{\Theta}}$ as a critical point also $l'_Y(\hat{\mathbf{\Theta}})=0$. Substituting the expression $\mathbf{n}^{(1)}$ as a function of $(\mathbf{\Theta}^{(j)}-\hat{\mathbf{\Theta}})$ we have:

$$\mathbf{n}^{(2)}\left(\mathbf{\Theta}^{(j)},\hat{\mathbf{\Theta}}\right)\approx-f_{\mathbf{n}^{(1)}}\left(\mathbf{\Theta}^{(j)},\hat{\mathbf{\Theta}}\right)^2 l''_Y(\tilde{\mathbf{\Theta}})=f_{\mathbf{n}^{(1)}}\left(\mathbf{\Theta}^{(j)},\hat{\mathbf{\Theta}}\right)^2 I(\tilde{\mathbf{\Theta}}), \tag{2.35}$$

where $-l''_Y(\hat{\mathbf{\Theta}})$ is defined as the *observed Fisher information*. It helps the researcher for the choice for a MLE point with respect to other points of the parameter space. Practically, in multidimensional cases, it is a positive definite matrix when Kalman filter provides non divergent output. The relation (2.35) states that the first and the second stopping rule values are linked to the Fisher information. In the next sections we will see that the observed Fisher information is the inverse of the variance of the estimator, an asymptotic result for T large.

2.2.3 Computational Aspects of the Algorithm

We have seen the standard Kalman filters equations to extract a time series of implicit prices. They are based on a design matrix with dummy variables for inclusion of components in products. We alternated the Kalman filter with an EM procedure to estimate unknown parameters of the process. If we transcribe in a code the Kalman filter and the EM algorithm equations for filter, smoother values, and parameter estimation as we gave in the previous section, we can face a serious problem which could affect the output of algorithm. It is due on the numerical instability of Kalman filter iterations (the filter iterations) that may yield non symmetric or non positive definite matrices. A detailed analysis of computer roundoff errors and its link to the ill conditioned problem is given in Grewal & Andrews (2008). The equation (A.13), computes the new covariance matrix, and after an accumulation of errors due on approximations irregular matrices may cause degeneracy in the filter. A high condition number for the matrix in (A.13) may affect output quality. Furthermore, if we implement the Kalman filter and the EM algorithm together, it means to increase the probability for number of times that this error can happen, since the number of iteration is multiplied. While in many simulations of EM+KF algorithm we observe a high probability to return on the convergence path, sometimes it can be fail.

Which are the factors that contribute to this problem? Grewal & Andrews list in their book some of the main factors, which we report:

- A large covariance matrix for disturbances respect to the actual one as initial assumption in the algorithm;
- A large transition matrix for state equation respect to the actual one as initial assumption in the algorithm;
- The inversion of the matrix in the Kalman gain formula;
- Large matrix dimensions. In our case m and n are large values and they can produce higher roundoff errors;
- Poor machine precision.

Some generic solutions for the ill-conditioned problem due on the roundoff propagation are reported in Verhaegen & Van Dooren (1986). How to solve that problem in our framework? Firstly, we fix not so large matrices for initial assumptions for disturbances. For state noise, we can assume a very tiny diagonal value for almost every variable in the system. It corresponds to assume that customers evaluate components without extreme jumps in the process. The price disturbances in the measurement equation require a previous analysis based on the historical data for a not so large initial assumption. Then, we can study the motivations of the ill-conditioned cases after the algorithm setting. We tested that ill-conditioned problems refer to the rare cases when algorithm is near the solution for some state with respect to other ones (see Simon, section 6.3.1). For the second stopping rule, it has never appeared a bad conditioned alarm because the test procedure is strict tuned-up by the chi-square distribution. In fact, the algorithm stops when the solution is near, providing an output for all the parameters. Differently, first and third stopping rules are not distributed with an exact form. In some cases, algorithm tries to approach to the estimation over the likelihood ratio performances of the first rule, and the covariance matrix assume very small values and causes ill-conditioned problems.

Because the study of the stopping rules is one of the topic of our research, we implemented two different methodologies in off line and on line contexts. For off line applications, we set the parameter δ for the first stopping rule larger than standard cases. For instance, if some data show the risk of degeneracy, we can set a smaller value of δ until we do not face the same problem again. Differently, on line algorithms require a decision-making independent on the stopping rule which we can not change. We will see in the next chapter, when we implement a real time framework a possible solution to this problem.

Many authors have solved the ill-conditioned problem introducing a list of possible corrections to "repair" the covariance matrix in each iteration corrupted from the defect. One of this technique is the famous square root algorithm or square root filtering (Simon, 2006). Square root filtering is a way to augment the precision of the Kalman filter when hardware precision is not available. The problem was widespread in the first years of the Kalman filter, when machine implementations gave numerical issues. While in the case of square root filtering the computational effort is greater then standard methodology, the numerical precision increases and hence mitigate numerical difficulties in implementations. Today, computers offer an high precision computation and the problem is very limited to specific cases: multivariate is one of those cases.

The condition number of a symmetric positive definite matrix $\mathbf{S}$ is defined as:

$$\kappa(\mathbf{S}) = \frac{\sigma_{\max}(\mathbf{S})}{\sigma_{\min}(\mathbf{S})} \geq 1, \text{ where } \sigma^2(\mathbf{S}) = \lambda(\mathbf{S}^T\mathbf{S}), \tag{2.36}$$

and λ is the eigenvector of the matrix. If $\kappa \to \infty$, the matrix $\mathbf{S}$ is said to be poorly conditioned or ill-conditioned, and $\mathbf{S}$ approaches a singular (not invertible) matrix. Round off in those cases may cause large deviations for the state variables. In our code, we implement the square root filter using the *MRDIVIDE* function in *MATLAB*©. It computes the Cholesky factorization of the symmetric positive definite matrix. Every time we divide two matrices with different structure, for instance a rectangular matrix over a square matrix, as in (A.14), we opt for the *MRDIVIDE* instead of the usual inversion. Cholesky factorization twice the precision of the standard Kalman filter and we avoid many alarm in the code for ill-conditioned problems.

Which is the complexity of the EM+KF algorithm? We use the concept of "flop" to measure the complexity in the EM+KF algorithm. To avoid misunderstanding we give the following definition taken by Golub & Van Loan (1996).

Definition 1 A *flop* is a floating point operation. A dot product operation of length n involves $2n$ flops because there are n multiplications and n additions to calculate in it.

The input of the algorithm is a time series of product prices until the time T. Of course, we have to start the computations after a reasonable number of periods, say t_0. We tested that the value of t_0 must be at least 10 periods for providing an output. According to the purpose of the analysis, the output provides hedonic price estimates based on the previous K values, where the value of $K, K \in \{t_0, t_0 + 1, \ldots, T\}$, is chosen by the user. Computational complexity of Kalman filter-smoother at the first step of outer iteration will be given by: $(m^3 \times n \times K)$, for the filter estimation, and $(m^3 \times n \times K)$ for the smoother estimation. Then, computational complexity of each iteration of the EM part is given by: $m \times n$. Since, there will be i_K iterations for the solution, we can say that the total complexity depends on the number of EM (outer) iterations of algorithm, and it is given by:

$$2 \cdot (m^3 \times n \times K) \cdot i_K. \tag{2.37}$$

Finally, the complexity of the algorithm can be reduced using one of the simplified methodology for symmetric positive definite matrix inversion, as Cholesky decomposition or triangularization. For an outline of these methodologies see the book of Grewal & Andrews (2008) (table 6.9, page 251). For instance, using the Cholesky decomposition of the $m \times m$ matrix in (A.13) the new complexity of the algorithm is:

$$2 \cdot \left(\left(\frac{1}{3} m^3 + \frac{1}{2} m^2 - \frac{5}{6} m \right) \times n \times K \right) \cdot i_K. \tag{2.38}$$

2.2.4 On the Convergence of the EM Algorithm

What do exactly work Kalman filter and the EM algorithm together? We have seen the defects of the Kalman filter due on the modeling error and the high variability of input data for multivariate case. In the case of parts and products in supply chain environment, the Kalman filter default and ill-conditioned problems depend respectively on the number of outer iterations and the number of time periods in the input variable. The EM algorithm is a two step procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data (see the Appendix B for details). In our algorithm, we observed only product prices and our hidden values are the hedonic evaluations. We call $\mathbf{Z}$ the multiple random variable for the hedonic prices, and $\mathbf{Y}$ the multiple random variable for the product prices. All the properties of EM procedure are valid in the case of perfect computations in the Kalman filter section of the algorithm. Unfortunately, this can not be the case in the extraction of high dimensioned vectors through Kalman filter. Thus, it is very important to take care of the convergence of EM algorithm, overall when we mix it with another algorithm as Kalman filter. Between two iterations of EM algorithm it is possible to define a mapping as:

$$\mathbf{\Theta^{(j+1)}} = \mathbf{M}(\mathbf{\Theta^{(j)}}), \quad (j = 0, 1, 2, \dots), \tag{2.39}$$

which converges to some point $\mathbf{\Theta}^*$. For details about convergence rate of the EM algorithm see Meng & Rubin (1991) and McLachlan & Krishnan (1997). We will restrict our analysis to the most important parameter in $\mathbf{\Theta}$, the $m \times m$ transition matrix $\mathbf{\Phi}$, considering the other elements as “nuisance” parameters. In fact, the distance in (2.24) is only based on the transition matrix, in the algorithm. We call the convergence point, $\mathbf{\Phi}^*$, which satisfies the stable relation given by:

$$\mathbf{\Phi}^* = \mathbf{M}(\mathbf{\Phi}^*). \tag{2.40}$$

According to the Taylor series expansion around the fixed point $\mathbf{\Phi}^*$ we must be have that:

$$\mathbf{\Phi^{(j+1)}} - \mathbf{\Phi}^* \approx \mathbf{J}(\mathbf{\Phi}^*)(\mathbf{\Phi^{(j)}} - \mathbf{\Phi}^*), \tag{2.41}$$

where $\mathbf{J}(\mathbf{\Phi})$ is the $m \times m$ Jacobian matrix for the mapping $\mathbf{M}(\mathbf{\Phi}) = (\mathbf{M_1}(\mathbf{\Phi}), \dots, \mathbf{M_m}(\mathbf{\Phi}))^T$. Each element of $\mathbf{J}(\mathbf{\Phi})$ is equal to:

$$J_{ik}(\mathbf{\Phi}) = \partial M_i(\mathbf{\Phi}) / \partial \phi_{ik}, \tag{2.42}$$

where $\phi_{ik} = (\mathbf{\Phi})_{ik}$, the jth element of the ith row of $\mathbf{\Phi}$. For the matrix $\mathbf{\Phi}$ a measure of the actual observed convergence rate is the global rate of convergence.

2.2.5 Properties and Tests for State-Space Models

The performance of our dynamic model [2] is qualified by its stability (Deistler & Hannan, 1988; Caines, 1988) meaning that the effects of the initial conditions disappear over time. More precisely, we can distinguish between a marginally stable system where the state $\mathbf{z}_t$ is bounded in each period for all bounded initial states $\boldsymbol{z}_0$, and an asymptotically stable system if the stability is reached after a large number of periods. A necessary and sufficient condition for both definition of stability is that the eigenvalues of the state-transition matrix $\boldsymbol{\Phi}$ are below one in absolute value. If eigenvalues are larger than one in absolute value, the system can be stabilized under certain conditions Harvey (1989). A stable system is also a stationary system but the reverse is not true (Lutkepohl, 2005).

Analysis of the eigenvalues of $\boldsymbol{\Phi}$ provides insight into the dynamics of the system. Particularly relevant is the dominant (largest) eigenvalue (Schoonbeek, 1986), which for many econometric models is close to unity. In our case, the stability of the system is linked with the not varying transition parameter $\boldsymbol{\Phi}$. It means that the behavior of hedonic differential prices does not change during the life of the products. From the Gaussian assumption, the Kalman estimator is very similar to ordinary least squares (OLS) estimator for high value of T. In fact, maximum likelihood estimator is coincident to OLS in the case of normal distributions, and Kalman estimator coincides with MLE under the same assumption of normality. Computing hedonic prices for large values of T corresponds to make a multiple regression via OLS estimator, whereas for small values of T the results may be strictly distinct. Finally, the stability property is dependent on the dominant eigenvalue and for this reason it is reasonable to check this value in each application. In the next chapter, stability of the process for hedonic prices will be an assumption that simplifies the forecast models and it will provide outperforming results.

In the property of the model, we assumed $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{\nu}_t$ are Gaussian. This property brings to a reduction of conditions for the maximum likelihood estimator validity. In our framework, parameters are unknown and a complex shape of error distributions may be source of problems for estimation of them. Although it is not strictly necessary since we have an asymptotic property typical of dynamic linear systems (see Caines, 1988, in chapter 8). Under general conditions, the hyper parameter $\boldsymbol{\Theta}_T$ obtained using the time series $\mathbf{y_1}, \ldots, \mathbf{y_T}$ and maximizing the likelihood as given in our algorithm satisfy:

$$\sqrt{T}\left(\hat{\boldsymbol{\Theta}}_{\mathbf{T}} - \boldsymbol{\Theta}\right) \stackrel{d}{\to} N\left[\mathbf{0}, I(\boldsymbol{\Theta})^{-1}\right], \quad \text{for } T \to \infty, \tag{2.43}$$

[2] In the sense of a dynamic linear model without the control vector, but only with state and output vector, respectively $\mathbf{z}_t$ and $\mathbf{y}_t$ (Simon, 2006). Our hedonic model consists in a linear discrete-time system with not varying parameters.

where $\mathbf{\Theta}$ is the real hyper parameter of the process and $I(\mathbf{\Theta})$ is the asymptotic information matrix given by:

$$I(\mathbf{\Theta}) = \lim_{n\to\infty} \frac{1}{T} E\left[-\partial^2 \ln LL_Y(\mathbf{\Theta})/\partial\mathbf{\Theta}\partial\mathbf{\Theta}'\right]. \tag{2.44}$$

The likelihood LL_Y is computed using the innovations of the process, $\mathbf{e}_t$, which have a zero mean and covariance matrices $\mathbf{\Sigma}_t = c\mathbf{D}\mathbf{P_t^{t-1}}\mathbf{D}' + \mathbf{\Sigma}_{\boldsymbol{\nu}}$. Hence, the model should be adapted for any distribution errors with zero mean.

Gaussian property may be tested for measurement errors (premiums) and for transition equation residuals. Therefore the normality of the product and hedonic prices is checked via the residuals. The non normality distribution affects on forecast interval determination. The reason is that forecast errors used in the construction of forecast intervals are weighted sums of the residuals. If we confirm the normality assumption it is logical to establish interval for predictions.

We apply the Mardia tests Cromwell *et al.* (1994) to evaluate the assumed multivariate normality in (2.9) and (2.10). The test statistics take the skewness ($\mathbf{sk_n}$) and kurtosis ($\mathbf{kr_n}$) of the distributions of the residuals as inputs:

$$\mathbf{sk_n} = T^{-2}\sum_i [(\hat{\mathbf{v}}_\mathbf{i} - \boldsymbol{\mu})'\mathbf{\Sigma^{-1}}(\hat{\mathbf{v}}_\mathbf{i} - \boldsymbol{\mu})]^3, \quad i = 1, \ldots, T, \tag{2.45}$$

$$\mathbf{kr_n} = T^{-1}\sum_i [(\hat{\mathbf{v}}_\mathbf{i} - \boldsymbol{\mu})'\mathbf{\Sigma^{-1}}(\hat{\mathbf{v}}_\mathbf{i} - \boldsymbol{\mu})]^2, \quad i = 1, \ldots, T, \tag{2.46}$$

where $\boldsymbol{\mu}$ is the mean vector of values $\hat{\mathbf{v}}_\mathbf{i}$, and n the dimension of the multivariate distribution. Under multivariate normality, $E[\mathbf{sk_n}] = 0$ and $E[\mathbf{kr_n}] = n(n+2)$. Mardia thus proposes the test statistics:

$$\mathbf{a} = \frac{(T \cdot \mathbf{sk_n})}{6} \sim \chi^2(n(n+1)(n+2)/6) \tag{2.47}$$

$$\mathbf{b} = \frac{\mathbf{kr_n} - n(n+2)}{[(8n(n+2)/T]^{1/2}} \sim N(0,1) \tag{2.48}$$

For the given significance levels, α and z, reject the normality hypothesis if $a > \tau$ and $|b| > |z|$, where:

- → τ is the critical value from a chi-square distribution with $n(n+1)(n+2)/6$ degrees of freedom;
- → z is the critical value from the standard normal distribution.

Then, we introduce a misspecification test for zero center (median or mean), for the series of innovation terms, $\mathbf{e}_t$ under general distribution form assumption. When a Kalman filter is used for state, in our case hedonic prices, estimation, the innovations given in section A.2 in appendices, can be measured and its mean can be approximated using statistical methods. If the mean of the innovation is not as expected, that means something did not work correctly with the filter. Perhaps the entire model is incorrect, or the hypothesis about noises are wrong. Therefore, mean tests for measurement errors and disturbances of the transition equation may be interpreted as misspecification tests for the state-space model.

An exact distribution-free nonparametric test for zero median is the sign test. Under the null hypothesis that the observed univariate series is independent with a zero median, the number of positive observations in a series of size T has the binomial distribution with parameters T and 0.5. Compute, after the implicit prices estimation, the test statistic:

$$S_1 = \sum_{t=1}^{T} I^+(e_t), \qquad \text{where } I^+(e_t) = \begin{cases} 1 \text{ if } e_t > 0, \\ 0 \text{ otherwise }, \end{cases} \tag{2.49}$$

and e_t is the residual error for the period t. For large samples, the function of statistic S_1 given by:

$$\frac{S_1 - T/2}{\sqrt{T/4}}, \tag{2.50}$$

is distributed as a standard normal. For small samples, the function of statistic S_1 has values collected in the binomial table for this nonparametric statistical test. In the case of multivariate distribution for the disturbances, it is reasonable to calculate the number of signs for each variable in the case of Gaussian distribution. Then, the test for the entire collection of errors is given by:

$$S_n = \sum_{j=1}^{n} \sum_{t=1}^{T} I^+(e_{jt}), \tag{2.51}$$

where e_{jt} is an element of the vector of innovations $\mathbf{e}_t$ defined in the previous section. For the multivariate series, if we have that every value of the j statistics:

$$\frac{S_n - Tn/2}{\sqrt{(Tn/4}}, \tag{2.52}$$

is distributed as a standard normal then also the multivariate distribution will have zero median. Although, it can happen that several marginal distributions have not zero median. The procedure can be interpreted as a sign test for the vectorization of the matrix of residuals. The sign test for multivariate distributions performs a two-sided sign test of the null hypothesis that data in the multiple n series of T length come from a continuous distribution with zero median. We suggest to compute the p value with an approximate method for number of periods over the 50. For the residuals of state space models, where Kalman filter use the least square method to obtain the state variable, this test is useful also to test the correctness of the algorithm.

Last tests are for independence of the vector of residuals, checking their whiteness. In this way, we investigate the nature of residuals supposed like a white noise in the state space system. If the hypothesis is rejected we can find another specification of the model like one of the set provided in the next chapter. For instance, a colored process noise can substitute the basic model.

The most famous test is based on the works of Chitturi and Hosking (Lutkepohl, 2005) and takes the name of multivariate Portmanteau test. It is assumed that $\boldsymbol{\nu}_t$ ($\boldsymbol{\varepsilon}_t$) is a n-dimensional (m-dimensional) white noise process with nonsingular covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\nu}}$ ($\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$). After the estimation of the residuals, the corrispondent autocovariance matrices are estimated by:

$$\mathbf{C}_i = \frac{1}{T}\mathbf{U}\mathbf{F}_i\mathbf{U}', \quad i = 0, 1, \ldots, h < T, \tag{2.53}$$

where the $\mathbf{F}_i$ matrices are defined as:

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{O}_i \\ \mathbf{I}_T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_T \\ \mathbf{O}_i \end{bmatrix}'. \tag{2.54}$$

The matrices $\mathbf{O}_i$ are $(i \times T)$ zero matrices, and the matrices $\mathbf{I}_T$ are $(T \times T)$ identity matrices. The matrices $\mathbf{U}$ contains residuals for the T periods. They have n rows in the case of measurement disturbances, whereas m rows for state noises.

We can also calculate the autocorrelation matrices $\mathbf{R}_i$ where the generic element is given by:

$$r_{hl,i} = \frac{c_{hl,i}}{\sqrt{c_{hh,0}}\sqrt{c_{ll,0}}}. \tag{2.55}$$

The estimated autocorrelations are plotted together the bands for acceptation given by $\pm 2/\sqrt{T}$. If any of the estimated coefficients reach out the area between the bounds the white noise hypothesis is rejected.

To facilitate the operations a Portmanteau test is designed for the whiteness hypothesis. The latter can be written as:

$$\begin{aligned} H_0 : \quad & \mathbf{R}_i = 0 \quad \forall \quad && i = 1, \ldots, h < T \\ H_1 : \quad & \mathbf{R}_i \neq 0 \quad \text{for} \quad && i = 1, \ldots, h < T, \end{aligned} \tag{2.56}$$

and the statistic for decision is:

$$p_h = T^2 \cdot \sum_{i=1}^{h} (T-i)^{-1} \mathrm{tr}(\mathbf{R}_i' \mathbf{R}_0^{-1} \mathbf{R}_i \mathbf{R}_0^{-1}). \tag{2.57}$$

Comparing values of p_h with the asymptotic distribution $\chi^2(n^2(h-p))_{.95}$ we can accept (reject) the hypothesis with a level of confidence of 5%. We recall that n is the number of variables in the vector, and p the lag order of the model.

Finally, we prefer to omit test for correlation between process and measurement noise because in that case the identification of parameters is very complicated.

2.3 Experimental Results of Algorithm in TAC SCM

In this section we show our algorithm application in TAC SCM, the trading agent supply chain simulated by human-computer agents. Here we repeat the application of our first paper presented to the conference in electronic commerce (ICEC) in the summer of 2010. Before the results we give a detailed explanation of TAC SCM rules and topics. The application outlines the setting and the performances of hedonic model for $n = 16$ products and $m = 5$ state variables. In this case, the variables represent the base computer hedonic price and the differential of prices between several computer parts: motherboards (MBs), central processing units (CPUs), random access memories (RAMs), and hard drives (HDs), for a total of five states.

2.3.1 TAC SCM: Rules and Details

In the TAC SCM game, a supply chain for PCs is considered in 220 game days of 15 real-time seconds each. This supply chain consists of customers, manufacturers and suppliers. These manufacturers are represented by software agents (such as MinneTAC) developed by competing teams that all try to maximize their profit over a game. Every game day, customers issue RFQs for 16 PC types, on which manufacturers can bid. Customers always place an order with the manufacturer offering the requested product for the lowest price (if this price is at or below their reservation price). The requested products are assembled by the manufacturers using ten different components procured from suppliers. Each of the six agents in a TAC competition decides which computers to assemble based on on line and off-line planned strategy.

A major challenge of the game is the limited visibility of the market environment. Real-time available data consist of information about received RFQs and an agent's own orders,

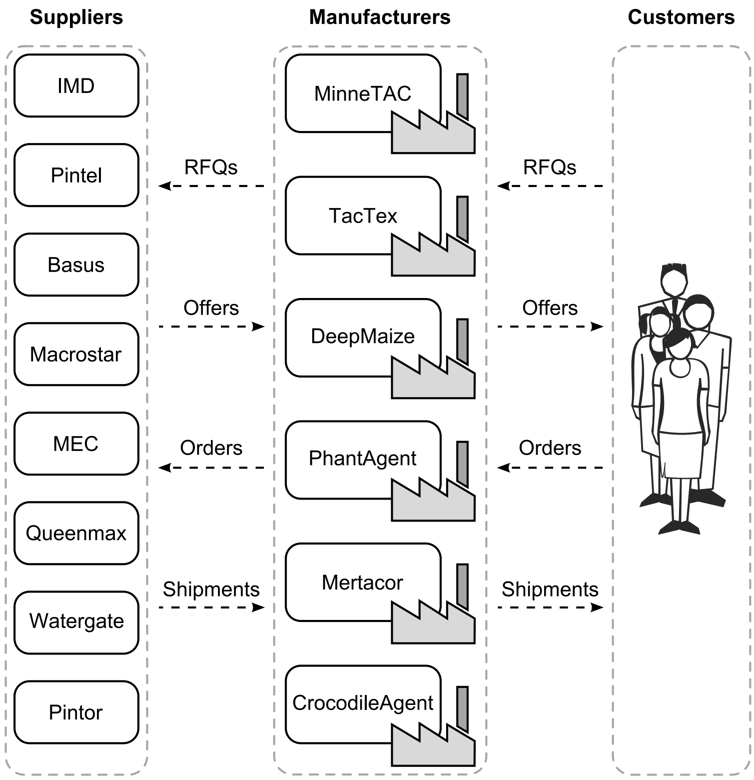

Figure 2.4: Schematic overview of a typical TAC SCM game scenario

the preceding day's minimum and maximum order price of each PC type, and aggregate market statistics issued every 20 days.

Manufacturers (agents) produce 16 different products (PC's), each consisting of four components (CPU, motherboard, memory and hard disk) that come in different varieties. Table 2.1 shows descriptions, base prices and suppliers of the product components. CPUs, for instance, are obtained from Pintel and IMD in two version, 2.0 GHz and 5.0 GHz. Based on the component prices, the price of a base computer with a Pintel motherboard, Pintel 2 GHz processor, 1 Gb Ram and 300 Gb hard disk can be obtained as 1650 (= 250+1000+100+300). The supplier criterion to accept manufacturer bids for component offers is based on revenue maximization. The daily quantities produced by suppliers follow a random walk with a mean of 550 components per day.

The production of products by manufacturers (agents) involves a different capacity usage for each type of computer, as reflected by the production cycles. Table 2.2 reports the nominal prices for each type of PC based on base prices per component.

TAC SCM agents must face uncertainty about the future, and they must afford decision problems before that uncertainty is resolved. They can use stochastic programming techniques (Benisch *et al.*, 2004), or other methodologies (see in Collins *et al.*, 2008, for a

Table 2.1: List of base option prices and suppliers per component in TAC SCM

Component Description	Base Price	Supplier
Pintel CPU 2.0 GHz	1000	Pintel
Pintel CPU 5.0 GHz	1500	Pintel
IMD CPU 2.0 GHz	1000	IMD
IMD CPU 5.0 GHz	1500	IMD
Pintel Motherboard	250	Basus, Macrostar
IMD Motherboard	250	Basus, Macrostar
Memory 1 GB	100	MEC, Queenmax
Memory 2 GB	200	MEC, Queenmax
Hard Disk 300 GB	300	Watergate, Mintor
Hard Disk 500 GB	400	Watergate, Mintor

compendium on them). Forecasting models in this environment were collected in a separate competition (Kiekintveld *et al.*, 2009; Pardoe & Stone, 2009). Several important agents include forecasting module based on those techniques. Recently, Ketter *et al.* (2009) evolve in a regime model, where the market is characterized by microeconomic situations. For instance, if the manufacturers meet a scarcity period for production, prices will increase for the law of supply and demand. Studying the conditional distributions of product prices it is possible to extract regime information for any product, and use it for on line identification of regime.

2.3.2 Product Prices Series

Every day the agent receives a report which includes the minimum and maximum prices of all the computers sold the day before, but not the quantity sold. We define the following series for product prices[3]):

- ${}_m\mathbf{y}_t$, the min-price vector at the day t;

- ${}_M\mathbf{y}_t$, the max-price vector at the day t;

- ${}_R\mathbf{y}_t$, the mid-range price vector at the day t given by the mean of the minimum and maximum prices (the symbol R is for range). The latter can be used to approximate the mean price.

Problems arise because mid-range price does not always provide an accurate estimate of the mean price because of local fluctuations in extreme prices. In fact, both minimum and maximum prices could be affected by temporary fluctuation and represent outliers instead of the true distribution of the prevailing prices.

In figure (2.5) there are plotted lines for minimum, maximum, and mid-range prices for

[3]We omit the product index i to avoid excess in notation. In the sequel we will consider the generic product price. When the specification of the product will be need we introduce again the standard notation

Table 2.2: Nominal prices, segments of the market and assembly cost per product

ID	Description	Segment	Nominal	Cycles
1	Pintel 2/1/300	low	1650	4
2	Pintel 2/1/500	low	1750	5
3	Pintel 2/2/300	mid	1750	5
4	Pintel 2/2/500	mid	1850	6
5	Pintel 5/1/300	mid	2150	5
6	Pintel 5/1/500	high	2250	6
7	Pintel 5/2/300	high	2250	6
8	Pintel 5/2/500	high	2350	7
9	IMD 2/1/300	low	1650	4
10	IMD 2/1/500	low	1750	5
11	IMD 2/2/300	low	1750	5
12	IMD 2/2/500	mid	1850	6
13	IMD 5/1/300	mid	2150	5
14	IMD 5/1/500	mid	2250	6
15	IMD 5/2/300	high	2250	6
16	IMD 5/2/500	high	2350	7

the computer of type four. Also mean price is represented, which is computable after the game, when all game data are available. We can see the discrepancy between mid-range and mean price, overall when minimum or maximum prices are distant from the mean price, around day 5, 10, 30, and 70. Dynamic pricing opportunities increase prices for an agent's order, while most of that day's order for computer of type four were sold at a much lower price.

To avoid the problem of outliers, Ketter *et al.* (2009) computed the smoothed mid-range ${}_R\tilde{\mathbf{y}}_t$ on day t as the average of the smoothed minimum price ${}_m\tilde{\mathbf{y}}_t$ and the smoothed maximum price ${}_M\tilde{\mathbf{y}}_t$ for the same day[4]. The smoothed values for both series can be calculated using a Brown linear exponential smoother (Brown *et al.*, 1961), with a value of the parameter $\alpha = 0.5$. We will show the computation steps of the smoothed minimum price, because for maximum prices the procedure is the same (it is sufficient to change the index m with M). The smoothed minimum price is given by:

$$ {}_m\tilde{\mathbf{y}}_t = 2 \cdot {}_m\tilde{\mathbf{y}}_t{}^{(A)} - {}_m\tilde{\mathbf{y}}_t{}^{(B)}, \tag{2.58}$$

where

$$ {}_m\tilde{\mathbf{y}}_t{}^{(A)} = 0.5 \cdot \left({}_m\mathbf{y}_t + {}_m\tilde{\mathbf{y}}_{t-1}{}^{(A)}\right), \tag{2.59}$$

[4]Actually, they worked with the normalized price, the product price over the nominal product cost. The advantage is given by the comparison of price patterns for different products

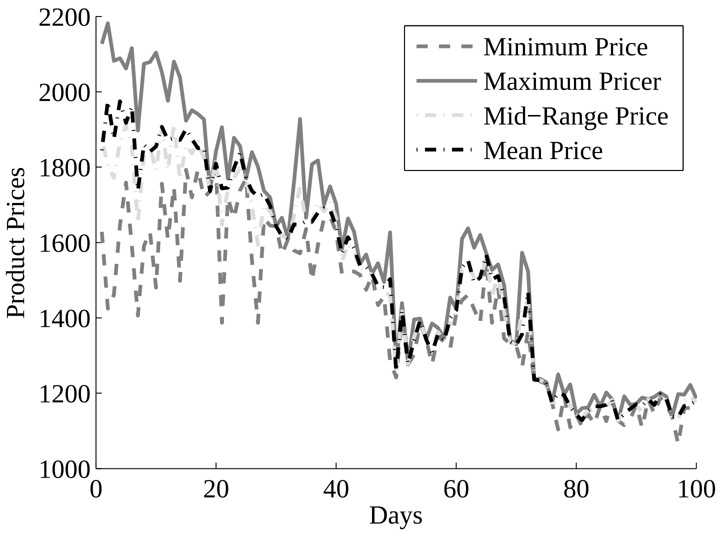

Figure 2.5: Minimum, maximum, mean, and mid-range daily prices of computers sold of the type four, PINTEL 5GHz/1Gb/300Gb. Game 7321@tac3

$$ {}_m\tilde{\mathbf{y}}_t{}^{(B)} = 0.5 \cdot \left({}_m\tilde{\mathbf{y}}_t{}^{(A)} + {}_m\tilde{\mathbf{y}}_{t-1}{}^{(B)} \right). \tag{2.60} $$

After the computation of the same quantity for ${}_M\tilde{\mathbf{y}}_t$, we can obtain the smoothed mid-range price series:

$$ {}_R\tilde{\mathbf{y}}_t = \frac{{}_m\tilde{\mathbf{y}}_t + {}_M\tilde{\mathbf{y}}_t}{2}. \tag{2.61} $$

We consider the same methodology to compute smoothed product price series, because we want to compare patterns and results with the work of Ketter *et al.* (2009).

2.3.3 Output of Hedonic Algorithm in TAC SCM

We obtain product prices from nine games of the 2005 tournament[5] using the previous formula. We call the vector of smoothed prices $\mathbf{y}_t$, and we use it as input of hedonic algorithm.

Application of the dynamic hedonic model to TAC SCM involves the definition of the design matrix $\mathbf{D}$ in (2.10) and initial settings of the mean and variance of the implicit prices in period zero, $\boldsymbol{\mu}_\mathbf{0}$ and $\boldsymbol{\Sigma}_\mathbf{0}$. The specification of the design matrix in the measurement relation (2.10) takes a PC with Pintel motherboard, Pintel 2 GHz CPU, 1 Gb Ram and 300 Gb hard disk as the base product variety (a column of ones), and the implementation of an IMD motherboard, 5 GHz CPU, 2 Gb Ram and 500 Gb hard disk as differentiating characteristics (columns of corresponding indicator variables). The elements of the implicit option price vector $\mathbf{z}_t$ are accordingly interpreted as follows:

[5]TAC SCM 2005 Semi-Finals and Finals (7306-7308tac,7312-7313tac,7367-7368tac,7373-7374tac).

- z_{1t} the implicit price of a base computer composed of a Pintel motherboard, a Pintel 2 GHz processor, 1 Gb Ram, and a 300 Gb hard disk;
- z_{2t} the implicit price differential of a base computer with an IMD motherboard instead of the Pintel;
- z_{3t} the implicit price differential of a base computer with 5 GHz CPU instead of 2 GHz CPU;
- z_{4t} the implicit price differential of a base computer with 2 Gb Ram instead of 1 Gb Ram;
- z_{5t} the implicit price differential of a base computer with a 500 Gb hard disk instead of a 300 Gb hard disk.

We have chosen to study the upper configuration of the hedonic prices for the following reasons: first, the hedonic price for base model is the minimum price that each customer is obliged to spent to acquire any item of the product variety. It represents the trend of the market at the bottom level, and coincides with the simplest computers of the brand numbered with zero in the design matrix. Output for base product evaluation is surely an important factor for many decisions process. Second, the use of differentials decreases the number of states to be considered in a generic system. In this way, the algorithm is simpler and more effective than an algorithm that consider more states. Furthermore, the differential meaning is very important in marketing strategies for branded and optional parts. Customers have different tastes and needs according to brands and technical specifications. In computer market, usually the base model tends to decrease, whereas the optional parts maintain the price during the shelf life.

The design matrix that represents production process for our latent factor in equation (2.10) is:

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T, \tag{2.62}$$

where each column represents the implicit price variable $\mathbf{z_i}$. Note as first column is full of one because it consists in the base computer, common component for every PC. Negative values for the estimated implicit prices (price differentials) may occur except for z_1. For example, a negative estimated z_{2t} simply indicates that an IMD motherboard is valued less than a Pintel motherboard. In addition, we select the following settings of the initial implicit price distribution. In line with the nominal procurement prices in table 2.1, we set the mean value of the initial implicit prices equal to:

$$\boldsymbol{\mu_0} = \{1650, 0, 500, 100, 100\}. \tag{2.63}$$

When, no data are available for the value of $\boldsymbol{\mu_0}$, an alternative choice is based on the initial value for product prices, $\mathbf{y}_1$, such that:

$$\boldsymbol{\mu_0} = \mathbf{D^{-1}\Phi^{-1}y_1}. \tag{2.64}$$

Here and in the following of the thesis, every time we invert a not square matrix like the design matrix $\mathbf{D}$, we want to calculate the Moore-Penrose inverse of it, or the generalized inverse. Eventually, an estimation of $\boldsymbol{\mu_0}$ through the historical values for a set of games may be more realistic. Then, algorithm requires initial value for the covariance matrices. The variance-covariance matrix of the initial implicit prices is set to:

$$\boldsymbol{\Sigma_0} = \begin{pmatrix} 5000 & 1000 & 1000 & 1000 & 1000 \\ 1000 & 5000 & 1000 & 1000 & 1000 \\ 1000 & 1000 & 5000 & 1000 & 1000 \\ 1000 & 1000 & 1000 & 5000 & 1000 \\ 1000 & 1000 & 1000 & 1000 & 5000 \end{pmatrix}. \tag{2.65}$$

This choice takes into account the substantial variability of the product prices, as illustrated in figure 2.8. Different $\boldsymbol{\Sigma_0}$'s have been tried without finding relevant differences. The estimation results seem therefore not particularly sensitive to the choice of $\boldsymbol{\Sigma_0}$. Finally, the initial value of the state-transition matrix $\boldsymbol{\Phi}$ has been set equal to the identity matrix, and that of the two variance-covariance matrices of the disturbances, $\boldsymbol{\Sigma}_\nu$ and $\boldsymbol{\Sigma}_\epsilon$ equal to diagonal matrices, with the same value of 10000 as entry.

Our methodology generates a massive amount of insight into the dynamic price development during different settings of the competitive environment. In this section, we give a descriptive account of the actual price developments of base products and estimated implicit prices, explore the dynamics of these price developments by means of the properties of the estimate state-transition matrices, and illustrate how our model can be used for forecasting.

2.3.4 Product and Implicit Component Price Behavior

Selling prices are characterized by considerable variability throughout the course of the game. Figure 2.6 illustrates the pattern of price volatility, calculated as a standard deviation

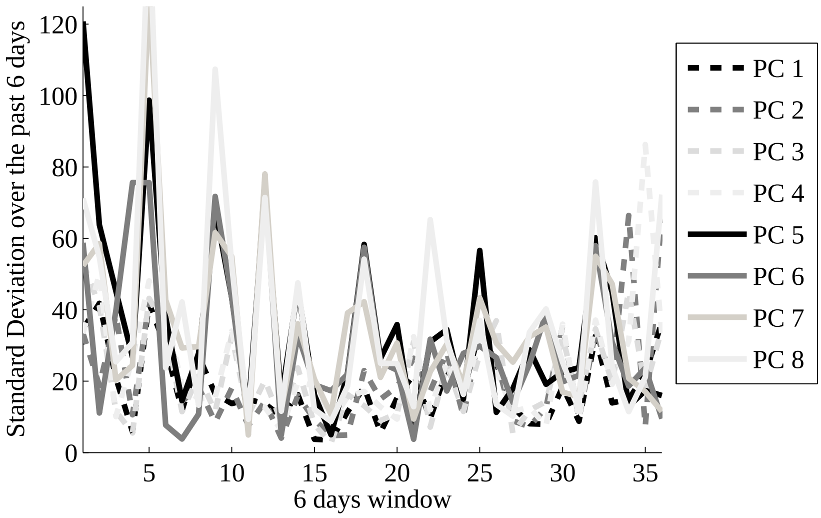

Figure 2.6: Price volatility of the eight products of the brand Pintel in the game 7306tac

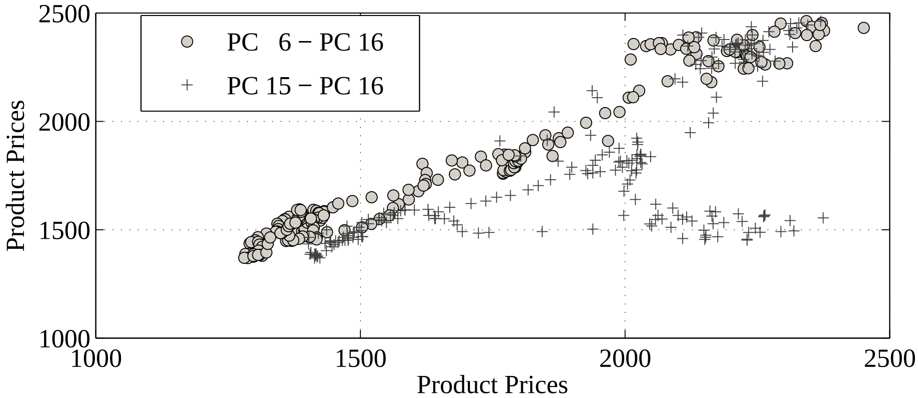

Figure 2.7: Scatterplots for two pairs of computers for all the days in Game 6.

of prices in moving windows of 6-days, for eight different products in a game. The general impression is that the price volatility is high during the beginning of the game, then rapidly drops, has a moderate, transient revival during the mid-game, and steeply increases toward the end of the game. Naturally, different patterns can discerned between different products. This is true not only for the price variability but also for the specific price patterns within each game.

To show how is important but at the same time difficult to consider co-dependencies between the product prices we have included two examples of scatter plots in Figure 2.7. Products sharing four similar components, like the computers with ID 15 and 16, should be more correlated than PC6 and PC16 that share only three components over five. Instead

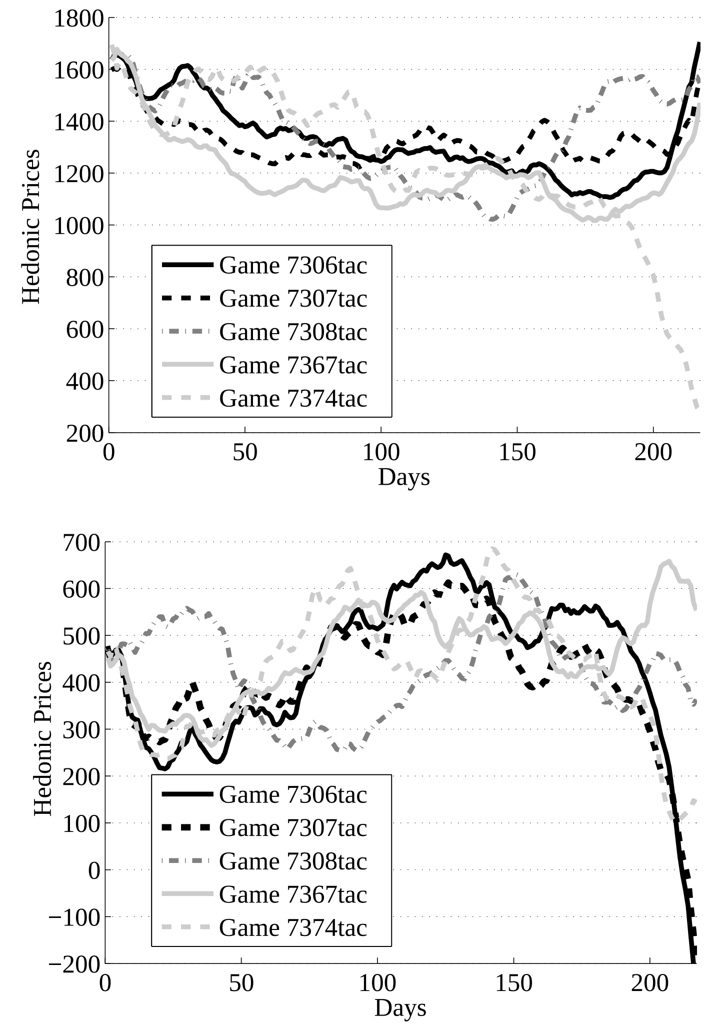

Figure 2.8: Price patterns of the base product (top) and CPU differential (bottom) hedonic prices for five games

they show a minor linear correlation. The mechanism of multivariate dependencies is usually the reason that pushes the researcher to the choice of a VAR model but we offer now an alternative instrument based on hedonic variables.

Figure 2.8 gives an impression of the pattern of base product prices over time for five selected games. The selection of games has been made to illustrate the variety of distinct patterns. The base product prices for game 7374, for instance, reveal a persistent downward trend, while the price patterns for games 7307 and 7367 have a bathtub shape. Price volatility is markedly present in all cases.

Application of the dynamic hedonic model to the selling prices of all products leads to the estimated implicit prices of the base product and the differentiating characteristics. Figure 2.9 presents these estimates for four selected games (7306, 7312, 7367 and 7373).

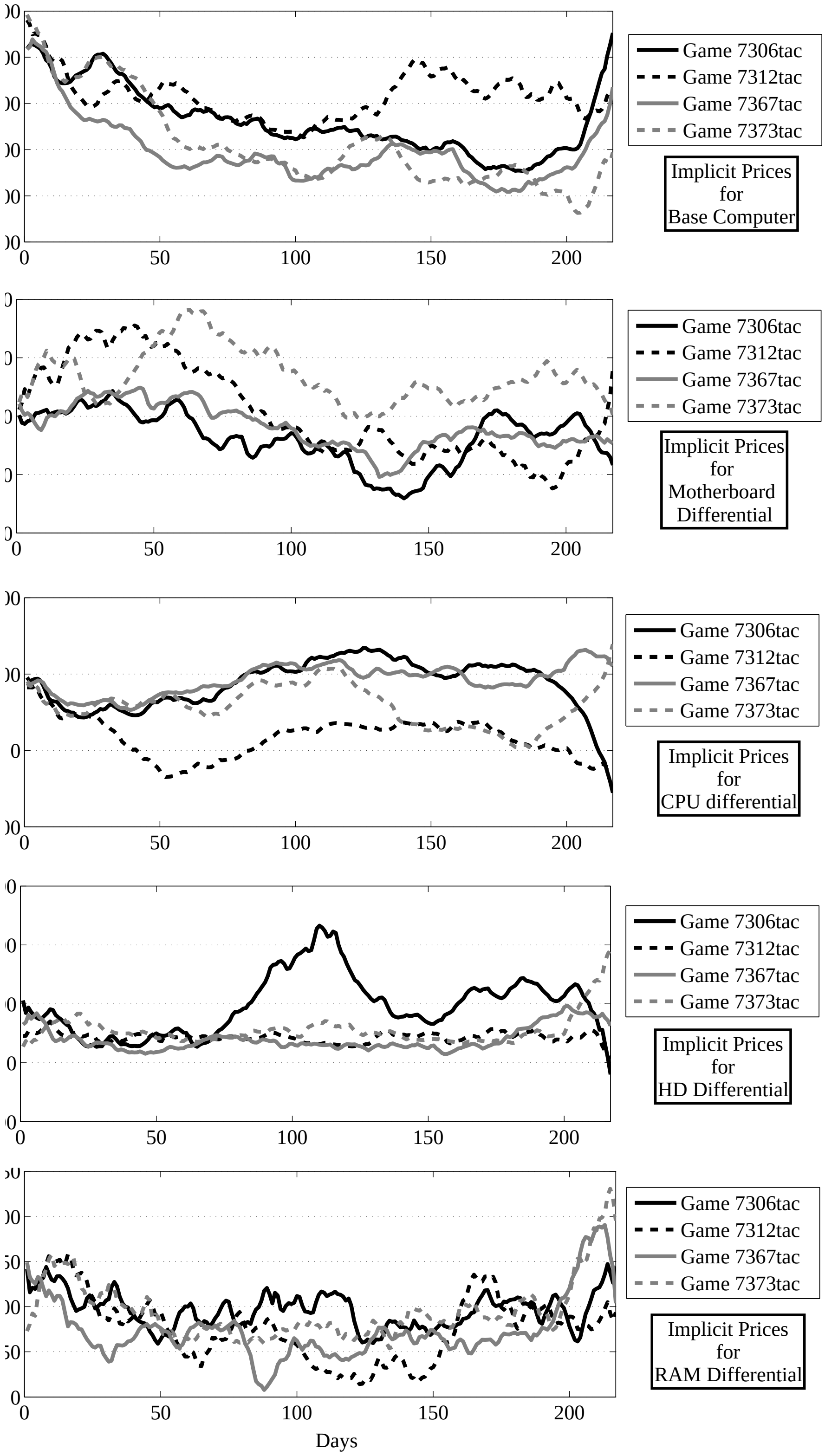

Figure 2.9: Estimated Implicit Prices for four TAC games

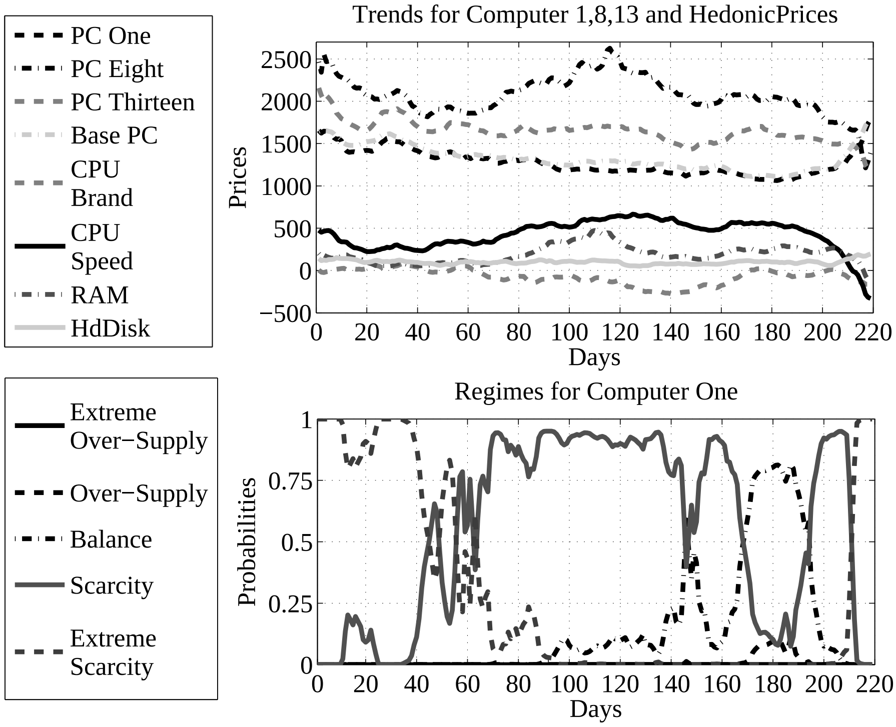

Figure 2.10: Product and component price developments together with regime classifications for game 7306 (semifinal 2005)

It illustrates that the developments of implicit prices vary within games and can be quite different between games. For instance, the additional (implicit) price of an IMD motherboard in games 7312 and 7373 is persistently above that in the other two games. In game 7373 this price differential with respect to the base product is positive throughout the entire game, whereas it is negative on almost all days for game 7306. The implicit price development of 2 GHz Ram seems relatively stable over time, but again the exception is game 7373 which reveals a sharp price increase during days 80-110. Sharp price increases or decreases are typical for the end periods (last 20 days) of all the games, but the specific direction, up or down, does not seem very systematic.

Taken together, these outcomes underline the importance of dynamic models of price developments. Static models are simply not consistent with the observed daily price fluctuations, changes in price developments, and marked differences between product markets (games) with different competitive settings.

In part, the observed variation in the (implicit) price developments within and between games may be explained by structural changes in the economic conditions that drive the market outcomes. Although, we do not allow for varying market regimes in the current hedonic model, we explore the issue by means of a comparison between estimated (implicit)

Table 2.3: Eigenvalues of $\hat{\mathbf{\Phi}}$

TAC Game	Eigenvalues of Transition Matrix				
7306	0.945	1.031	1.016	0.986	0.996
7307	0.913	1.056	0.999	0.990	0.990
7308	0.914	0.990	0.990	0.992	1.001
7312	0.967	0.967	1.000	0.987	0.987
7313	0.995	0.995	0.999	0.976	0.976
7367	0.985	0.985	0.999	0.972	0.981
7368	1.003	1.003	0.999	0.982	0.982
7373	1.003	1.003	0.986	0.986	0.999
7374	0.989	0.989	0.988	0.988	0.994

prices for a specific set of products and the probabilities of market regimes (excess supply, equilibrium and excess demand) defined in Ketter *et al.* (2009); see figure 2.10. Casual observation of the results suggests that, for instance, the price increase of PC 8 after day 100 is mirrored by an implicit price increase of Ram memory, during a period of excess demand. Likewise, the price drop of PC 13 before day 150 seems related with implicit price decreases of CPU brand and Ram during a time when the market moves toward equilibrium. Incidentally note that the observed and estimated implicit prices for the base model move closely together. More robust analysis of these interdependencies is left for further research.

2.3.5 Algorithm Results in TAC SCM

The dynamics of the implicit price developments are further explored by means of the properties of the state-space matrix $\mathbf{\Phi}$. We analyze the output provided by algorithm after the maximum time of estimation, 217 days. It used the entire information of product prices, like in analysis "at posteriori". Table 2.3 gives the eigenvalues of the estimated state-transition matrix $\hat{\mathbf{\Phi}}$ for nine different games. The fact that the eigenvalues can be quite different, again points at the existence of market/game-specific price dynamics. All eigenvalues are around one indicating stability. Out of the nine games, three have eigenvalues strictly below one reflecting both stability and stationarity.

Agent strategies will be similar in games with the same players, though random events

Table 2.4: Dominant eigenvalue of $\hat{\mathbf{\Phi}}^l$ at different lags (l) for first three games

TAC Game	Lag 5	Lag 10	Lag 20	Lag 50	Lag 100
7306	1.16	1.35	1.83	4.54	20.57
7307	1.31	1.72	2.96	15.10	228.12
7308	1.00	1.01	1.02	1.04	1.08

Table 2.5: Dynamic multipliers (×100) for base model implicit prices due to unit changes of the five component prices

Game	ϕ_{11}	ϕ_{12}	ϕ_{13}	ϕ_{14}	ϕ_{15}
7306	100.64	-5.74	-7.21	7.50	4.26
7307	100.81	-3.89	-5.76	4.58	1.13
7308	99.23	1.77	-0.99	3.78	4.86
7312	100.02	-0.38	-0.58	6.57	-8.97
7313	100.49	10.44	2.21	51.38	-73.42
7367	97.92	-0.01	3.50	-0.66	12.18
7368	100.20	-3.25	-1.69	0.63	0.64
7373	99.57	-1.23	-1.43	21.35	-16.25
7374	98.49	3.20	3.94	-6.48	0.21

in the TAC game can create drastically unexpected outcomes.

We explore the dominant eigenvalue at different lags to gain a better understanding of how every game matches different patterns of implicit prices. In the long run, the dominant eigenvalue determines whether the implicit prices will move upward, downward or oscillate. This is illustrated by the results in table 2.4. The first game, 7306, shows a normal trend for implicit prices, while the game 7307 is characterized by a strong volatility. Game 7308 reveals the most conservative behavior with dominant eigenvalues remaining close to one. Also, this game shows price stability toward the end of the game. The development of implicit prices can be further characterized by means of the dynamic multipliers (Hamilton, 1994). If at time t the implicit price $\mathbf{z_t}$ is known, then the implicit price after j periods can be determined by recursively evaluating the state equation (2.9):

$$\mathbf{z_{t+j}} = \mathbf{\Phi^j z_t} + \mathbf{\Phi^{j-1}}\boldsymbol{\varepsilon}_{\mathbf{t+1}} + \mathbf{\Phi^{j-2}}\boldsymbol{\varepsilon}_{\mathbf{t+2}} + \cdots + \mathbf{\Phi}\boldsymbol{\varepsilon}_{\mathbf{t+j-1}} + \boldsymbol{\varepsilon}_{\mathbf{t+j}} \tag{2.66}$$

The dynamic multiplier, which reflects the effect of current implicit prices on the prices of j-periods ahead, follows as:

$$\frac{\partial E(\mathbf{z}_{t+j})}{\partial \mathbf{z}_t'} = \mathbf{\Phi}^j. \tag{2.67}$$

Table 2.5 presents dynamic multipliers of unit changes in the five implicit prices (differentials) on the implicit price of a base model 20 days ahead. Each row of the table corresponds to a row of the transition matrix $\hat{\mathbf{\Phi}}$, estimated after 217 days in every game. For game 7308, the negative result -0.0721 (-7.21%) for $\phi_{1,3}$ implies that an increase of the implicit price differential for CPU leads to a decrease of the implicit price of the base computer. All values of $\hat{\mathbf{\Phi}}$ are close to unity for the base product effect and usually close to zero for the effects of the implicit prices of other components. Exceptions are observed for games 7313

Table 2.6: Sign test p-values for measurement disturbances (first row) and for state noise (second row) in the nine games. Acceptance (0) and rejection (1) of zero mean hypothesis for several time windows. $T = x$ means that statistics refer to the first x days of the game

TAC	Sign Test p-values											
Game	T=20		T=50		T=100		T=150		T=200		T=215	
7306	0.162	(0)	0.915	(0)	0.030	(1)	0.380	(0)	0.086	(0)	0.039	(1)
	1	(0)	0.798	(0)	0.928	(0)	0.164	(0)	0.228	(0)	0.691	(0)
7307	0.240	(0)	0.003	(1)	0	(1)	0	(1)	0	(1)	0	(1)
	0.838	(0)	0.798	(0)	1	(0)	0.124	(0)	0.163	(0)	0.081	(0)
7308	0.105	(0)	0	(1)	0	(1)	0.171	(0)	0.985	(0)	0.932	(0)
	0.838	(0)	0.798	(0)	0.418	(0)	0.942	(0)	0.447	(0)	0.561	(0)
7312	0.696	(0)	0.972	(0)	0.043	(1)	0.185	(0)	0.469	(0)	0.423	(0)
	0.682	(0)	0.798	(0)	0.369	(0)	0.826	(0)	0.612	(0)	0.409	(0)
7313	0.342	(0)	0.548	(0)	0.532	(0)	0.083	(0)	0.584	(0)	0.443	(0)
	1	(0)	0.609	(0)	0.369	(0)	0.942	(0)	0.485	(0)	0.737	(0)
7367	0.780	(0)	0.804	(0)	0.600	(0)	0.639	(0)	0.684	(0)	0.670	(0)
	1	(0)	0.307	(0)	0.590	(0)	0.379	(0)	0.526	(0)	0.062	(0)
7368	0.615	(0)	0.860	(0)	0	(1)	0	(1)	0	(1)	0	(1)
	1	(0)	0.201	(0)	0.787	(0)	0.187	(0)	0.526	(0)	0.603	(0)
7373	0.012	(1)	0.377	(0)	0.094	(0)	0.919	(0)	0.930	(0)	0.798	(0)
	0.682	(0)	0.250	(0)	0.928	(0)	0.510	(0)	0.526	(0)	0.691	(0)
7374	0.162	(0)	1	(0)	0.861	(0)	0.241	(0)	0.737	(0)	0.443	(0)
	0.838	(0)	0.609	(0)	1	(0)	0	(1)	0.009	(1)	0.002	(1)

and 7373, which reveal substantial effects of memory ($\phi_{1,4}$) and hard disk ($\phi_{1,5}$). Clearly, it is not a case because the same pattern with minor effects appears in the game 7312. The dependencies between base model and hard drives have positive signs and compensate negative codependencies with random memory in many agents' strategies.

Test results are given in table 2.6 and 2.7. Mean test is based on the statistic in (2.52). Statistics are computed for several estimation windows: after 20, 50, 100, 150, 200, 215 periods, the algorithm provides output and statistics for check the validity of the model specifications. In this way, we check zero medians for residual distribution for short, medium, or long input series. Table reports the p-value of the sign test and the rejection (1) or the fail of rejection to the null hypothesis of zero median at the 5% significance level. For measurement equation (first rows in the table), the disturbances sometimes deviate from the zero median. In three games, 7307tac, 7368tac, and 7374tac, only the residuals for the case $T = 20$ are centered in zero. We expected such outputs for premium mean, overall in longest time windows. Like in finance, where assets present not centered disturbances, also in consumer markets dynamic pricing offer asymmetric distributions for premium random variables.

For transition equation (second rows in the table), the assumption of a zero mean distribution is acceptable if the distribution is Gaussian. Only in the game, 7374tac, the hypothesis is rejected. In fact, the customer evaluations in the last periods of the game 7374 rapidly decreases due to the emptying out of stored products and highest production levels. Mardia test results about Gaussian distribution (MVN) hypothesis are quite all positive for skewness and kurtosis for short initial time series. In table 2.7 we give the results for all the games and for several lengths of input series. The disturbances in the measurement equation have no normal shape except for short time series, whereas the disturbances in the transition equation provides no rejection tests until time series of length 50. It clearly means that normality assumptions are valid only for short time series but not for the entire duration of the game, and only for state noise distribution. Finally, the assumptions of Gaussian distribution and zero mean can be confirmed only for noise in hedonic process. Because measurement disturbances usually have mean values depending on the strategies of the players, an estimation of the mean value required a large set of games, and we want to explore the methodology in a future research. Obviously, it should be also conducted a parallel research about the type of distribution of ν's random variables. The latter is an n-dimensional joint distribution and the methodology is not simple.

Portmanteau test for measurement residuals shows a strong autocorrelation for any lengths of input series greater than 20 periods. Then, we reject the hypothesis of whiteness for measurement disturbances in all the nine games. Differently, for state noises the same test provide acceptance for short series (20 values) and rejection for longer residual series.

We conclude that model may be modified in measurement equation to include another variable or a colored process for disturbances. The transition equation may be also modified in the same sense when we want to model medium/long pattern of the process.

2.3.6 Algorithm Performances and Convergence

Table 2.8 shows the time and the number of steps required for convergence of the algorithm for different values of δ_1 defined in (2.24). We opt for the matrix distance of order one such as:

$$\mathbf{n^{(1)}}(\mathbf{\Phi^{(j)}}, \mathbf{\Phi^{(j-1)}}) = \|\mathbf{\Phi^{(j)}} - \mathbf{\Phi^{(j-1)}}\| = \sum_{i,k=1}^{5} \left|\phi_{ik}^{(j)} - \phi_{ik}^{(j-1)}\right|, \tag{2.68}$$

which may be substitutes by Euclidean distance to emphasize the differences between matrices in close iterations. We will try several values of δ_1 for convergence of the EM algorithm as the values in the header of table 2.8. What do represent those values? They want to

Table 2.7: Mardia test results: p-values and rejection (1)/no rejection (0) for skewness (sk) and kurtosis (ku) of measurement and transition noise

TAC Game	Test	Mardia Test p-values for skewness and kurtosis T=20	T=50	T=100	T=150	T=200	T=215
7306	sk_ν	0.981 (0)	0 (1)	0 (1)	0(1)	0(1)	0(1)
	ku_ν	0.047 (1)	0 (1)	0 (1)	0(1)	0(1)	0(1)
	sk_ε	0.344 (0)	0.686 (0)	0.553 (0)	0.151(0)	0.001(1)	0(1)
	ku_ε	0.985 (0)	0.089 (0)	0.823 (0)	0.244(0)	0(1)	0(1)
7307	sk_ν	0.996 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.015 (1)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.978 (0)	0.143 (0)	0.002 (1)	0 (1)	0(1)	0(1)
	ku_ε	0.160 (0)	0.054 (0)	0.317 (0)	0 (1)	0(1)	0(1)
7308	sk_ν	0.999 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.010 (1)	0.030 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.407 (0)	0.432 (0)	0 (1)	0 (1)	0(1)	0(1)
	ku_ε	0.900 (0)	0.760 (0)	0 (1)	0 (1)	0(1)	0(1)
7312	sk_ν	0.999 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.013 (1)	0.003 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.987 (0)	0.034 (0)	0.001 (1)	0 (1)	0(1)	0(1)
	ku_ε	0.276 (0)	0.073 (0)	0 (1)	0 (1)	0(1)	0(1)
7313	sk_ν	0.999 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.008 (1)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.407 (0)	0.123 (0)	0.015 (1)	0.004(1)	0(1)	0(1)
	ku_ε	0.932 (0)	0.985 (0)	0 (1)	0 (1)	0(1)	0(1)
7367	sk_ν	0.998 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.013 (1)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.168 (0)	0.798 (0)	0.199 (0)	0.013(1)	0(1)	0(1)
	ku_ε	0.779 (0)	0.904 (0)	0.727 (0)	0.002(1)	0(1)	0(1)
7368	sk_ν	0.998 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.014 (1)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.933 (0)	0.343 (0)	0 (1)	0 (1)	0(1)	0(1)
	ku_ε	0.540 (0)	0.320 (0)	0.025 (1)	0 (1)	0(1)	0(1)
7373	sk_ν	0.998 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.012 (1)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.303 (0)	0.382 (0)	0.023 (1)	0 (1)	0(1)	0(1)
	ku_ε	0.955 (0)	0.938 (0)	0.866 (1)	0.001(1)	0(1)	0(1)
7374	sk_ν	0.997 (0)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	ku_ν	0.013 (1)	0 (1)	0 (1)	0 (1)	0(1)	0(1)
	sk_ε	0.449 (0)	0.315 (0)	0.265 (0)	0 (1)	0(1)	0(1)
	ku_ε	0.914 (0)	0.723 (0)	0.848 (0)	0 (1)	0(1)	0(1)

Table 2.8: Convergence of the algorithm for different games and settings applying the first stopping rule. Results in seconds (number of iterations)

	Desired Precision for First Stopping Rule (δ_1)				
Game	$0.5 \cdot 10^{-4}$	10^{-4}	$0.5 \cdot 10^{-3}$	10^{-3}	10^{-2}
7306	52.54	22.12	4.70	2.98	1.18
	(653)	(324)	(72)	(43)	(17)
7307	36.78	19.54	5.78	3.82	1.46
	(473)	(273)	(87)	(57)	(21)
7308	91.42	47.79	9.50	4.70	1.01
	(964)	(580)	(142)	(72)	(13)
7312	38.78	19.24	4.98	3.02	1.13
	(492)	(273)	(80)	(48)	(16)
7313	-	-	198.06	64.42	5.00
	(+5000)	(+5000)	(1682)	(758)	(79)
7367	16.85	9.57	2.94	2.00	1.00
	(244)	(147)	(46)	(31)	(14)
7368	14.99	8.77	3.04	2.13	0.96
	(224)	(135)	(48)	(32)	(13)
7373	35.53	20.53	6.67	4.38	1.44
	(452)	(278)	(100)	(67)	(21)
7374	7.18	4.55	1.82	1.39	0.77
	(109)	(69)	(27)	(20)	(10)

measure the distance between two matrix whose entries are dynamic multipliers. As we have see in section 2.2.4 the convergence of EM algorithm depending on the Kalman filter calculations and a good calibration consists to set not so small values for δ_1. In this way we avoid ill-conditioned problem for covariance matrix of Kalman filter and we save a lot of time cutting the iterations.

Differences between games are relevant and appear not to be related to the development of product prices. For example, games satisfying stationarity (7313, 7367, and 7374) show quite different stopping times, the minimum and the maximum of all the games. An acceptable value is given by $\delta_1 = 0.0025$. We shall see in the chapter 5 the distribution of number of iterations for the first stopping rule.

Table 2.9 gives the convergence results for the second stopping rule defined as in (2.32). The degrees of freedom of the chi-square distribution should be at least set as the number of variables in the system minus the number of constraint for them, $n - m$. This point is the core of the first research contribution of the thesis. Exact computation of the distribution of $W(y)$ in (2.30) is not easy, and approximate methods must be used. A method consists to convert the state space model in a multivariate linear regression model, like in the work of Durbin & Koopman (2001). We opt for testing three critical values of chi-square distribution,

Table 2.9: Convergence of the algorithm for different games and chi square degrees of freedom applying stopping rule 2, in seconds (number of iterations)

	Desired Precision (δ_2)					
Game	$\geq \chi_{10,0.975}$	$\geq \chi_{10,0.990}$	$\geq \chi_{5,0.975}$	$\geq \chi_{5,0.990}$	$\geq \chi_{3,0.975}$	$\geq \chi_{3,0.990}$
7306	2.02	2.82	3.63	7.55	7.50	34.71
	(31)	(44)	(57)	(116)	(115)	(446)
7307	2.23	3.13	4.14	9.00	8.73	49.98
	(33)	(47)	(63)	(135)	(134)	(597)
7308	1.89	2.62	3.45	7.67	7.52	45.12
	(28)	(39)	(52)	(115)	(113)	(546)
7312	1.85	2.54	3.38	7.64	7.10	44.28
	(28)	(39)	(52)	(115)	(109)	(546)
7313	1.95	2.68	3.50	7.15	21.32	30.59
	(30)	(42)	(55)	(111)	(296)	(402)
7367	1.94	2.61	3.41	7.09	7.03	32.01
	(30)	(41)	(54)	(111)	(109)	(426)
7368	2.93	3.67	4.40	7.78	7.03	28.02
	(30)	(42)	(54)	(107)	(106)	(368)
7373	2.03	2.79	3.43	6.75	6.66	24.11
	(29)	(40)	(51)	(100)	(99)	(327)
7374	1.83	2.43	3.10	6.05	6.01	22.50
	(28)	(38)	(49)	(96)	(95)	(319)

the first one with ten degrees of freedom, the second one with a value of five, and the last with three degrees of freedom. Although the convergence rates are similar, the resulting transition matrices $\hat{\mathbf{\Phi}}$ are different. This indicates that more than one set of values for the parameters $\mathbf{\Theta}$ gives rise to similar values of the likelihood function. Note the homogeneity of computational times and the absence of divergent cases.

The observation of the product price is of little help for choosing between the optima obtained (Hamilton, 1994). The results for the third stopping rule (2.33) showed low values for product prices differences between two iterations, with a mean of 0.1%. Since, it was difficult to establish a good value for $\delta_{3,t,i}$ and its sum, δ_3, we will prefer the first stopping rule to use in the on line algorithm of the next chapter.

Finally, we can calculate the algorithm complexity of equations (2.37) and (2.38), for $m = 5$, $n = 16$, and $K = 220$ and using the number of iterations given by tables 2.8 and 2.9 required by the nine games under the different values for the distances $\mathbf{n}_1$ and $\mathbf{n}_2$. In the first case, we obtain $440000 \times i_{220}$, and through Cholesky decomposition $176000 \times i_{220}$, where i_{220} is the number of iterations showed in parenthesis in the tables.

2.3.7 Forecasting Results in TAC SCM

Although the hedonic model is not created only for prediction goals, we want to test the performance of the model to analyze the stability of hedonic preferences. According to previous results, we expect to find best results after a minimum interval of days. What can be the utility of a hedonic forecast model in TAC SCM? During the game, agents receive component supply reports generated by the system with information about: (*i*) the aggregate quantities shipped by all suppliers in the given period; (*ii*) aggregate quantities ordered from all suppliers in the given period; and (*iii*) mean prices per type of computer for all components ordered during the period (price is available for CPU, motherboard, memory, and hard disk). The publication of these reports allows agents to implement the information in their own algorithm to predict future prices for products and components. Taking into account the hedonic prices estimated day by day using the daily information about product prices, the agent can substitute the lack of information in procurement market with the hedonic one.

Predicted product prices are calculated by:

$$\hat{\mathbf{y}}_{T+h} = \mathbf{D}\hat{\mathbf{z}}_{T+h}, \tag{2.69}$$

whereas the predicted hedonic prices are calculated by:

$$\hat{\mathbf{z}}_{T+h} = \hat{\mathbf{\Phi}}^{\mathbf{h}}\hat{\mathbf{z}}_{T}. \tag{2.70}$$

The minimum estimation (extraction) window for hedonic algorithm is five day. It means that algorithm starts to work after five days of the game.

In each period T we may apply the algorithm to estimate not only $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{z}}_{1:T}$, but also predicted prices for products and component evaluations. Figure 2.11 illustrates price forecasts for two computers (PC1 and PC8) up to 20 days ahead, based on product price information for the first 20 days of the game. The predicted product prices within the estimation period tend to develop in line with the actually observed prices. The forecast product prices correctly indicate the stabilizing price trend for the first five days (days 21-25 of the game), but then rapidly diverge. After ten days (game day 30) the forecasts rapidly deteriorate, possibly caused by an unexpected shift of the market toward an excess demand regime (with high product prices). If the estimation period is extended to 60 days, this diverging effect seems to be less prominent. In this case, figure 2.11 shows that the predicted product prices correctly follow the observed market prices within the estimation period, even in the presence alleged regime shifts, and are relatively consistent with the observed price

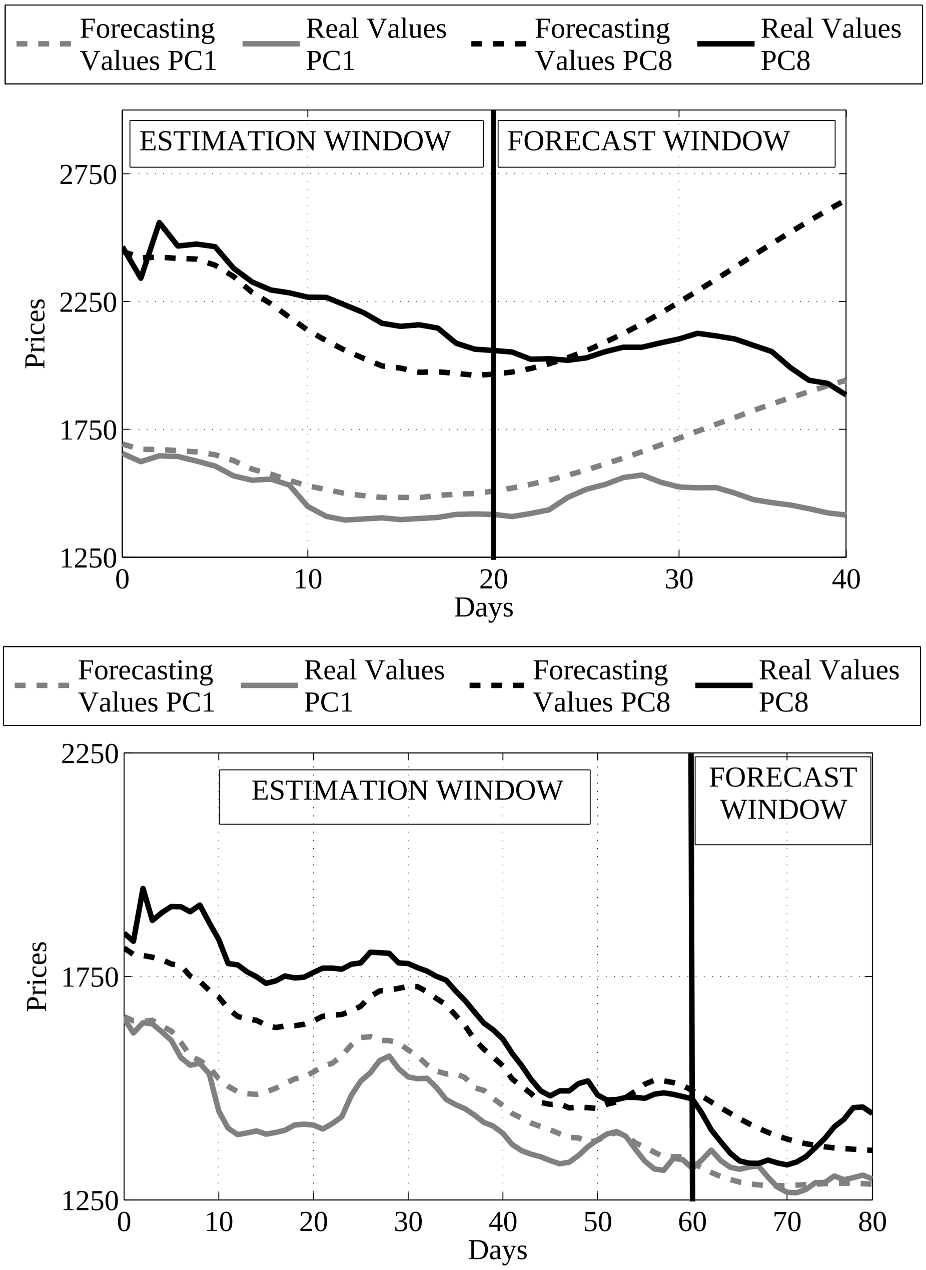

Figure 2.11: 20-day-ahead product price forecasts (PC1 and PC8) based on the first 20 days of the game (upper plot) and on the first 60 days of the game (bottom plot). TAC Game 7306

behavior in the 20-day forecast period.

The analysis of a forecast model based on hedonic prices is postponed in the next chapter of the thesis. It issues in any case, the property of the multiple forecast model which is affected by the codependencies between the prices, whereas the univariate models take into account only of the individual histories of prices. In fact, the anomalous behavior for product number eight in figure 2.11 (bottom graph) is not possible in univariate models with a small number of lags. Differently, in multiple models we can attend a change in trend which does not follow the mean reversion property.

2.3.8 Conclusions and Summary of the Application

We presented an application of the dynamic multivariate hedonic model to explain and forecast prices of heterogeneous products sharing common components. The model was tested in a set of nine games. Output of the algorithm under changeable stopping rules shows stability while the implementation of the Kalman filter is not so simple. The extracted hedonic values can be used for selection of components during the assembly operations, to select the quantity and the type of parts of supply refilling, for quality analysis, customer oriented strategies, and overall for relationship with procurement prices (option costs).

Based on the results, the model may be extended for several way to explore other relevant hypotheses. First, the estimated implicit component prices may be related with the actually observed procurement prices to gain further understanding of the conceptual relation between these prices, and to explore if any discrepancies point at upcoming changes in market conditions, either at the procurement or the sales side.

Secondly, it is worthwhile to integrate our model with the Markov regime-switching methodology to cope with structural changes in the model parameters and to improve the forecasting performance in markets with changing regimes. In Ketter *et al.* (2009), the authors showed that market conditions, such as over-supply, balance or scarcity, alternate during the history of the market for each product. These varying market conditions are inconsistent with the constant parameters of our model, and warrant attention in future extensions.

Chapter 3

Alternate Hedonic Models Formulations

Previous algorithm is primarily intended to estimate parameters of the discrete linear system of the supply chain. In the sequel of the thesis we call it *Algorithm 1*, the technique for the base hedonic model. In spite of its intuitive logic, the proposed algorithm has serious drawbacks when applied in dynamic real-time contexts, such as uncertainty for stopping rule and divergence risks. Furthermore, the high dimensioned vectors of state and product prices affect estimation performances which barely coincides with the actual parameters of the system. We will try to solve the estimation problem in the next chapter, when we shall implement a real time algorithm for that scope. Now, we consider the same state space model for supply chain systems under several assumptions of the knowledge of the parameters. This section wants to give an example of the utilization of the hedonic model, and, in the same time, to test the goodness of parameters and their distributions under specific hypothesis. For instance, in the first section we test the easiest way of modeling the hedonic price process, setting a unitary transition matrix and diagonal covariance matrices. A second option is the assumption of a diagonal transition matrix. The latter must be estimated according to previous data. In this way, the risks of failure for ill-conditioned problems decreases as well as the time of computation, like in the noise model. We will give an example in TAC SCM of the hedonic model under those assumptions. Obviously, our goal is to increase the forecasting performances for medium/long term predictions under the assumption that stable parameters outperform estimated ones.

3.1 The Noise Model

3.1.1 Formulation

To illustrate an example of the hedonic model when the agent knows the parameters of the process we opt for the simple noise model for state variables defined by:

$$\begin{aligned} \mathbf{z}_t &= \mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t, \\ \mathbf{y}_t &= \mathbf{D}\mathbf{z}_t + \boldsymbol{\nu}_t, \end{aligned} \tag{3.1}$$

where the covariance matrices for disturbances are diagonal as in (2.11) and (2.12). Both disturbances are assumed multivariate normal distributed with zero means, like the initial state vector. The latter is distributed according to multivariate normal distribution with mean vector $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$. We can set the initial state parameters as we have seen in the previous chapter. This time the algorithm does not estimate new parameters which remain set to the input value. The differences between the model in (2.9)-(2.10) are the restrictions about the transition matrix $\boldsymbol{\Phi}$ and the covariance matrices. In the noise model, $\boldsymbol{\Phi}$ is the identity $m \times m$ matrix $\mathbf{I}$, whereas the covariance matrices are all assumed diagonal and known. The choice of that alternative parameter setting transform the basic hedonic model in a multidimensional random walk named noise model. Which values should we assign to diagonal entries of covariance matrices? From the application of the hedonic algorithm analyzed in section 2.2 we have learned about possible candidates for those values. The variability of product prices can be estimated using historical data, and assuming zero covariances between product prices. The variability of hedonic prices can be estimated on the base of the product price variances.

After those assumptions, apply a Kalman prediction algorithm to estimate hedonic prices (see section A.1). The methodology is simpler than EM+KF procedure, because it is based on single prediction procedure. In fact, in the Kalman prediction algorithm we do not estimate parameters of the system. A representation of the algorithm for Kalman prediction of hedonic values is reported in figure 3.1 under the name *Algorithm 2*. The algorithm can work also for other restrictions of the transition matrix $\boldsymbol{\Phi}$. In those cases, we call it *Algorithm 3*. Here, the transition matrix is indicated as $\mathbf{F}$, and it is supposed known and in some cases different from identity matrix. In this way, a variant of the simplest noise model is obtained. Assuming that the dynamic multipliers of hedonic process are all zero except in the diagonal of the transition matrix, the problem of the estimation of $\boldsymbol{\Phi}$ is reduced to the estimation of the m diagonal entries of the transition matrix $\mathbf{F}$. Hence, the latter is assumed with non zero elements only on the diagonal. We estimate those m diagonal values via *Algorithm 1* output

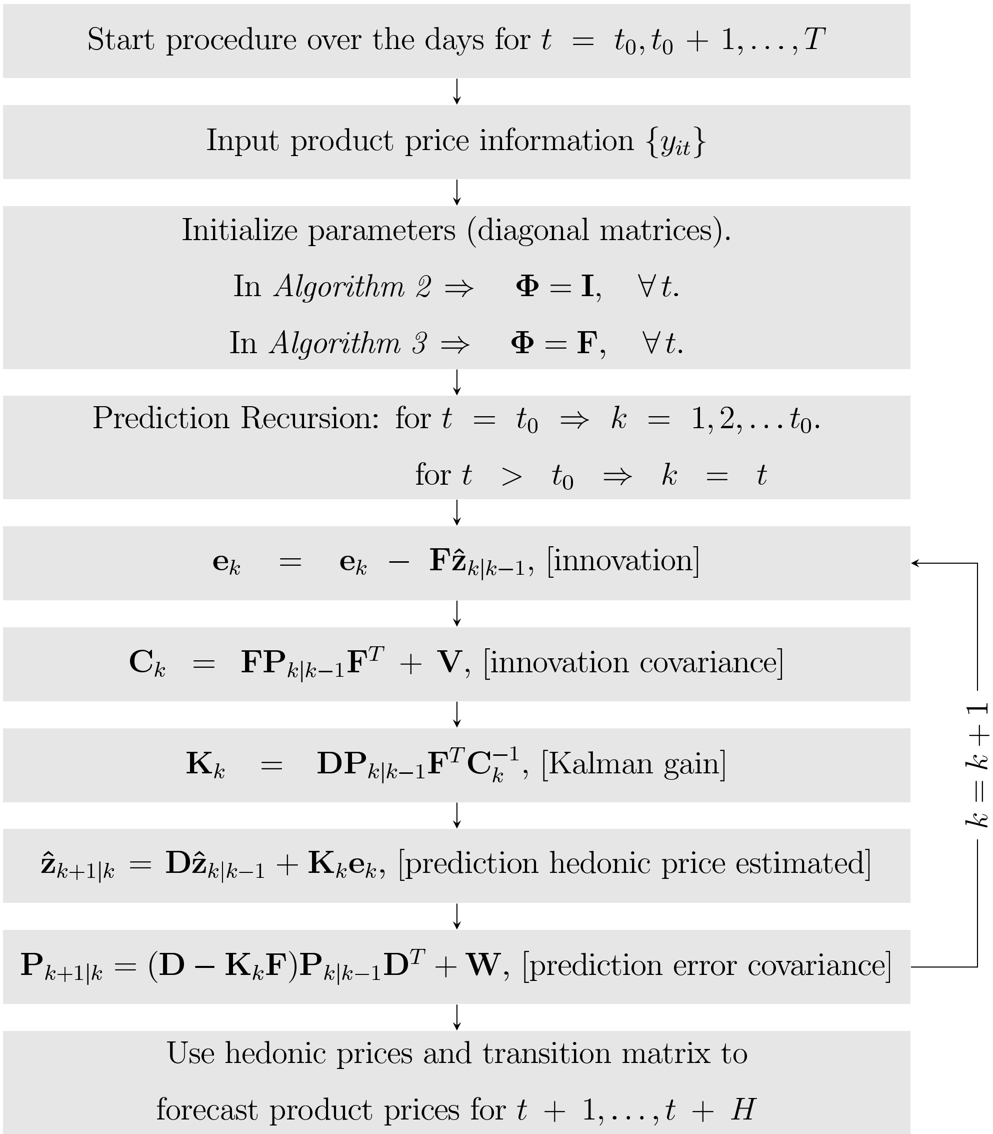

Figure 3.1: Prediction algorithm used to impute implicit component prices for each time period t based on diagonal matrices

for a set of historical data. In fact, from the off line output of *Algorithm 1*, we can derive empirical distributions for the dynamic multipliers in the transition matrix. From those distributions we can estimate the mean values of the diagonal entries during the process. In this case, the hedonic model is defined by:

$$\mathbf{y}_t = \mathbf{D}\mathbf{z}_t + \sigma_v \mathbf{I},$$
$$\mathbf{z}_t = \mathbf{F}\mathbf{z}_{t-1} + \sigma_u \mathbf{I}, \tag{3.2}$$

where $t = 1, \dots, T$, $\mathbf{F}, \sigma_v, \sigma_u$ estimated by historical data. Here, the estimate of the matrix $\mathbf{F}$ is of the type:

$$\hat{\mathbf{F}} = \begin{bmatrix} \hat{\phi}_1 & 0 & \dots & 0 \\ 0 & \hat{\phi}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\phi}_m \end{bmatrix}. \tag{3.3}$$

We call this approach, *Diagonal Model* and the relative Kalman prediction technique for extracting hedonic prices the *Algorithm 3*.

An example of an application of the noise model in TAC SCM will be given in the next chapter, when we implement it in a real time algorithm.

3.1.2 An application in TAC SCM of the Noise Model

In the application of the first chapter we estimate the parameters of hedonic model given a time series of product prices of length T. After the initial settings, the algorithm alternated between Kalman filter estimation and Expectation-Maximization until the convergence for the maximum likelihood estimators. Now, the algorithm is reduced to a simple prediction procedure where parameters remain fixed to initial values chosen by user. In the case of the noise model the advantages are that the transition matrix is a matrix with all unitary entries, or that hedonic prices are stable for the entire game. Our application wants to extract information about hedonic price vector $\mathbf{z}_T$ with the same meaning of formulation in sub-section 2.3.3. The design matrix coincides with the matrix in (2.62), and the initial mean is fixed as in (2.63). Covariance for initial state is given by the diagonal matrix:

$$\mathbf{\Sigma}_0 = \text{diag}(50000, 10000, 20000, 15000, 15000), \tag{3.4}$$

where the *diag* operator gives the ordered entries of the main diagonal, the only non zero values of the matrix. Values are larger than previous algorithm because the vector for the mean of initial state does not change during the framework. In this way, also for very distant initial assumptions we do not risk to start from strange values.

The measurement uncertainty is represented by the following diagonal matrix:

$$\mathbf{\Sigma}_\nu = 20000 \cdot \mathbf{I}_{16\times 16}, \tag{3.5}$$

and the noise in transition equation has a covariance matrix set to:

$$\mathbf{\Sigma}_\varepsilon = \mathrm{diag}(10000, 2000, 4000, 3000, 3000), \tag{3.6}$$

proportional to the hedonic prices and smaller than $\mathbf{\Sigma}_\nu$ as suggested in the first chapter application.

We are interested in differences between the output of two algorithm. Times of computations for the noise model are shorter than times of *Algorithm 1*. In the chapter 5 we will give a detailed analysis over time of computations. Off line extraction with a time window until the end of the game is compared with algorithm outlined in figure 2.3. Figure 3.2 shows the smoothed trend estimated together the parameters (that is, smoothing is done using the parameter estimated), and the extracted hedonic prices via prediction algorithm after 215 days in the games 7306tac and 7307tac. Differences between methodologies are relevant. Base model evaluations are overestimated in the noise model, whereas the differentials for optional upgraded parts are underestimated. In general, hedonic price for the base model decreases, and the stability assumptions should be modified for less than one values in the transition matrix. Because of the greatest uncertainty in customer demand for optional parts, the patterns for $z_2, \ldots, z_5$ are quite similar in both methodologies. By specifying unitary dynamic multipliers, we predicted the actual direction of trends of those variables. Implicit prices for recurrent parts tend to decrease, as the implicit prices for optional parts tend to maintain their values for all the history of product variety.

We postpone forecast analysis of the noise and diagonal models in the next chapter.

3.2 Hedonic State-Space Models with Lags

We introduce some modifications of the original hedonic model with the relative new algorithms for lagged linear models. We have modified previous algorithm in the case of more lags in hedonic transition equation. This modification requires a different implementation of the algorithm. In the following subsection, we outline the methodology for hedonic price and parameters estimation of lagged model, and we examine the effect of such hypothesis in a TAC SCM application.

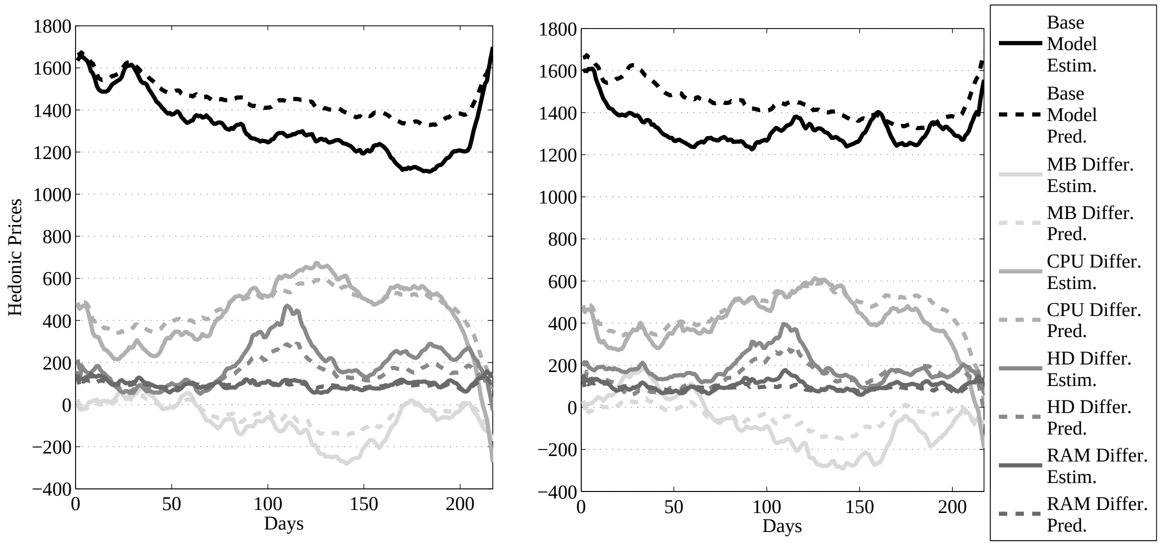

Figure 3.2: Comparison of price patterns of the base model, MB, CPU, HD, and RAM differentials for the noise model and the estimation algorithm in two games (7306tac to the left, 7307tac to the right)

3.2.1 Lags in the Hedonic Transition Equation

Sometimes hedonic prices may show a strong dependency with previous lagged values as in the standard autoregressive models. Instead of relation (2.9) we can consider an alternative law of the type:

$$\mathbf{z}_t = \Phi_1 \mathbf{z}_{t-1} + \cdots + \Phi_p \mathbf{z}_{t-p} + \boldsymbol{\varepsilon}_t, \tag{3.7}$$

and in this case we have to select the correct value of p to perform our time series estimation. In some cases length p of lag dependency could be known to the researcher. Otherwise, it could be practical to test some lag dependency by observing the auto-correlation function of product series. In fact, if the values of the vector $\mathbf{y}$ show a strong dependency on the previous p values, we can assume the same dependency on the hedonic prices since the latter satisfies relation (2.10).

Assuming valid the relation (3.7) the Kalman filter part of the algorithm remain identical but the variables $\mathbf{z}_t$ and $\mathbf{\Phi}$ must be substitute by a new vector and a new matrix such as:

$$\mathbf{z}_t = [\mathbf{z}_t \; \mathbf{z}_{t-1} \ldots \mathbf{z}_{t-p}]', \quad \text{and} \quad \mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_1 & \mathbf{\Phi}_2 & \ldots & \mathbf{\Phi}_p \\ \mathbf{I} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix}. \tag{3.8}$$

The EM part of the algorithm must compute now the derivative of the new matrix in (3.8). Since the dimension of the state vector may highly increase using several lags, we prefer to take into account the dimensionality in the choice of the model, overall for the estimation of the parameters. In fact, with high dimensioned state variables there is need of more time to estimate correctly the hyperparameter. Furthermore, the estimation performances of algorithm are negatively correlated to the value of m in terms of precision.

When we must maximize the score function given in (2.14), the derivative is now considered respect to $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_p$. The other derivatives respect to the three covariance matrices and initial mean vector remain unchanged. To find a relation for the new transition matrix, as in the simplest case of (2.20), we start from the new expression of the central term in (2.14):

$$\mathrm{tr}\{\mathbf{\Sigma}_{\varepsilon}^{-1}[\mathbf{S_{11}} - \mathbf{S_{10}}\mathbf{\Phi}' - \mathbf{\Phi}\mathbf{S}_{\mathbf{10}}' + \mathbf{\Phi}\mathbf{S_{00}}\mathbf{\Phi}']\}. \tag{3.9}$$

Both matrices in (2.16) and (2.17), $\mathbf{S}_{00}$ and $\mathbf{S}_{10}$, are now $(m \times p) \times (m \times p)$, which can be partitioned in p sub-matrices each $m \times m$ dimensioned, as:

$$\mathbf{S}_{00} = \begin{bmatrix} \mathbf{S}_{00}^{(11)} & \mathbf{S}_{00}^{(12)} & \ldots & \mathbf{S}_{00}^{(1p)} \\ \mathbf{S}_{00}^{(12)} & \mathbf{S}_{00}^{(22)} & \ldots & \mathbf{S}_{00}^{(2p)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{00}^{(1p)} & \mathbf{S}_{00}^{(2p)} & \ldots & \mathbf{S}_{00}^{(pp)} \end{bmatrix}, \quad \mathbf{S}_{10} = \begin{bmatrix} \mathbf{S}_{10}^{(11)} & \mathbf{S}_{10}^{(12)} & \ldots & \mathbf{S}_{10}^{(1p)} \\ \mathbf{S}_{10}^{(21)} & \mathbf{S}_{10}^{(22)} & \ldots & \mathbf{S}_{10}^{(2p)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{10}^{(p1)} & \mathbf{S}_{10}^{(p2)} & \ldots & \mathbf{S}_{10}^{(pp)} \end{bmatrix}. \tag{3.10}$$

Note the first matrix is symmetric and the second one is not symmetric. In fact, from section 2.2.2 we have seen how the matrix in (2.17) is a sum of semi symmetric matrices, because the product $\mathbf{z}_t^T \mathbf{z}_t^{T\prime}$ always generate a symmetric matrix. Differently, in (2.16) the product $\mathbf{z}_t^T \mathbf{z}_{t-1}^{T}{}'$ can not generate a symmetric matrix. We must rewrite also the covariance matrix for product prices disturbances using the same partition, as:

$$\mathbf{\Sigma}_{\varepsilon} = \begin{bmatrix} \mathbf{\Sigma}_{\varepsilon}^{(11)} & \mathbf{\Sigma}_{\varepsilon}^{(12)} & \ldots & \mathbf{\Sigma}_{\varepsilon}^{(1p)} \\ \mathbf{\Sigma}_{\varepsilon}'^{(12)} & \mathbf{\Sigma}_{\varepsilon}^{(22)} & \ldots & \mathbf{\Sigma}_{\varepsilon}^{(2p)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Sigma}_{\varepsilon}'^{(1p)} & \mathbf{\Sigma}_{\varepsilon}'^{(2p)} & \ldots & \mathbf{\Sigma}_{\varepsilon}^{(pp)} \end{bmatrix}. \tag{3.11}$$

Next step is to compute the p derivatives of (3.9) respect to each transition sub-matrix $\mathbf{\Phi}_i, i = 1, \ldots, p$, and to set them equal to zero, as:

$$-\frac{\partial \mathrm{Tr}(\mathbf{\Sigma}_{\varepsilon}^{-1}\mathbf{S}_{10}\mathbf{\Phi}')}{\partial \mathbf{\Phi}_i} - \frac{\partial \mathrm{Tr}(\mathbf{\Sigma}_{\varepsilon}^{-1}\mathbf{\Phi}\mathbf{S}_{10}')}{\partial \mathbf{\Phi}_i} + \frac{\partial \mathrm{Tr}(\mathbf{\Sigma}_{\varepsilon}^{-1}\mathbf{\Phi}\mathbf{S}_{00}\mathbf{\Phi}')}{\partial \mathbf{\Phi}_i} = 0, \quad \forall \quad i = 1, \ldots, p. \tag{3.12}$$

After we solved the system in p matrix-variables, we find the estimates of each submatrix

$\mathbf{\Phi}_i^{(j)}$ at the jth iteration step of the EM algorithm. We plug each of those solutions to obtain the new entire matrix $\mathbf{\Phi}^{(j)}$ given by:

$$\mathbf{\Phi}^{(j)} = \begin{bmatrix} \mathbf{\Phi}_1^{(j)} & \mathbf{\Phi}_2^{(j)} & \dots & \mathbf{\Phi}_p^{(j)} \\ \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}. \tag{3.13}$$

To solve the system given in (3.12) means to compute the derivative of the trace of a product for each of the three terms. We can write the product of three generic matrices in the first term, such as:

$$\mathbf{ABC}' = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{2n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nn} \end{bmatrix} \begin{bmatrix} c_{11} & \dots & c_{n1} \\ c_{12} & \dots & c_{n2} \\ \vdots & \ddots & \vdots \\ c_{1n} & \dots & c_{nn} \end{bmatrix} =$$

$$= \begin{bmatrix} \sum_{j,k} a_{1k}b_{kj}c_{1j} & \dots & \sum_{j,k} a_{1k}b_{kj}c_{nj} \\ \sum_{j,k} a_{2k}b_{kj}c_{1j} & \dots & \sum_{j,k} a_{1k}b_{kj}c_{nj} \\ \vdots & \ddots & \vdots \\ \sum_{j,k} a_{nk}b_{kj}c_{1j} & \dots & \sum_{j,k} a_{1k}b_{kj}c_{nj} \end{bmatrix} \tag{3.14}$$

Then the trace of the generic product is given as:

$$\mathrm{Tr}(\mathbf{ABC}') = \sum_{i,j,k}^{n} a_{ik}b_{kj}c_{ij}. \tag{3.15}$$

At the same way, we can derive the second term of the numerator in (3.12), which will be given by:

$$\mathrm{Tr}(\mathbf{ACB}') = \sum_{i,j,k}^{n} a_{ik}c_{kj}b_{ij}. \tag{3.16}$$

For the product of four items, as in the last term of (3.12), we have:

$$\mathrm{Tr}(\mathbf{ACBC}') = \sum_{i,j,k,l}^{n} a_{lk}c_{kj}b_{ji}c_{li}. \tag{3.17}$$

The generic partial derivative respect to the transition matrix $\mathbf{C}$ (in the system $\mathbf{C} = \mathbf{\Phi}_i$)

results for the first term equal to:

$$\frac{\partial \mathrm{Tr}(\mathbf{ABC}')}{\partial \mathbf{C}} = \begin{bmatrix} \partial \mathrm{Tr}(\mathbf{ABC}')/\partial c_{11} & \dots & \partial \mathrm{Tr}(\mathbf{ABC}')/\partial c_{1n} \\ \vdots & \ddots & \vdots \\ \partial \mathrm{Tr}(\mathbf{ABC}')/\partial c_{n1} & \dots & \partial \mathrm{Tr}(\mathbf{ABC}')/\partial c_{nn} \end{bmatrix}. \tag{3.18}$$

Differently, for the second term we have:

$$\frac{\partial \mathrm{Tr}(\mathbf{ACB}')}{\partial \mathbf{C}} = \begin{bmatrix} \partial \mathrm{Tr}(\mathbf{ACB}')/\partial c_{11} & \dots & \partial \mathrm{Tr}(\mathbf{ACB}')/\partial c_{1n} \\ \vdots & \ddots & \vdots \\ \partial \mathrm{Tr}(\mathbf{ACB}')/\partial c_{n1} & \dots & \partial \mathrm{Tr}(\mathbf{ACB}')/\partial c_{nn} \end{bmatrix}, \tag{3.19}$$

and for the third term:

$$\frac{\partial \mathrm{Tr}(\mathbf{ACBC}')}{\partial \mathbf{C}} = \begin{bmatrix} \partial \mathrm{Tr}(\mathbf{ACBC}')/\partial c_{11} & \dots & \partial \mathrm{Tr}(\mathbf{ACBC}')/\partial c_{1n} \\ \vdots & \ddots & \vdots \\ \partial \mathrm{Tr}(\mathbf{ACBC}')/\partial c_{n1} & \dots & \partial \mathrm{Tr}(\mathbf{ACBC}')/\partial c_{nn} \end{bmatrix}. \tag{3.20}$$

We give the expression for partial derivatives for the case $p = 3$. In this case, given the matrices:

$$\begin{aligned} \mathbf{K}_1 &= \left[\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(12)}\left(\mathbf{S}_{00}^{(11)} - 2\mathbf{S}_{10}^{(21)}\right) + \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(13)}\left(\mathbf{S}_{00}^{(12)} - 2\mathbf{S}_{10}^{(31)}\right)\right]\left(2\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(11)} + \mathbf{S}_{10}^{(11)}\right)^{-1} \\ \mathbf{K}_2 &= \left[\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(12)}\left(\mathbf{S}_{00}^{(12)} - 2\mathbf{S}_{10}^{(22)}\right) + \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(13)}\left(\mathbf{S}_{00}^{(22)} - 2\mathbf{S}_{10}^{(32)}\right)\right]\left(2\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(11)} + \mathbf{S}_{10}^{(12)}\right)^{-1} \\ \mathbf{K}_3 &= \left[\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(12)}\left(\mathbf{S}_{00}^{(13)} - 2\mathbf{S}_{10}^{(23)}\right) + \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(13)}\left(\mathbf{S}_{00}^{(32)} - 2\mathbf{S}_{10}^{(33)}\right)\right]\left(2\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{(11)} + \mathbf{S}_{10}^{(13)}\right)^{-1}. \end{aligned} \tag{3.21}$$

Each of the matrix in (3.21) is a submatrix in the square matrix $\mathbf{K}$ given by:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_2 & \mathbf{K}_3 \\ \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}, \tag{3.22}$$

and after dividing $\mathbf{K}$ by $\mathbf{S}_{00}$ we obtain the matrix with the solutions for the system in (3.12):

$$\mathbf{K}\mathbf{S}_{00}^{-1} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \boldsymbol{\Phi}_3 \\ \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}. \tag{3.23}$$

Substituting the matrices in a $\boldsymbol{\Phi}$ matrix as in (3.13), we obtain a new matrix for the next $(j+1)$th iteration.

3.2.2 An Application of the Lagged Hedonic Model

We show an example of an application of the lagged model in TAC SCM. We calculate the value of p based on the partial auto correlation function of the product prices. The assumption in this case is that components and products have the same auto correlation. The transition equation of the model is replaced by:

$$\mathbf{z}_t = \boldsymbol{\Phi}_1\mathbf{z}_{t-1} + \boldsymbol{\Phi}_2\mathbf{z}_{t-2} + \boldsymbol{\Phi}_3\mathbf{z}_{t-3} + \boldsymbol{\varepsilon}_t, \tag{3.24}$$

and the algorithm is updated under the expression resulting in the previous sub section. The input of the algorithm is a time series of product prices until the time T. The output of the algorithm consists of the series of hedonic prices of the same length, the three estimates of the transition matrices, the estimates of the other parameters. Obviously, we are interested in the differences between the estimates in the two models. Also an analysis of the transition matrices $\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \boldsymbol{\Phi}_3$ is very useful.

In figure 3.3 we compare the time series of hedonic prices estimated by the model given in (2.9)–(2.10), which we name it as “Model 1”, and the time series of hedonic prices estimated by the model given in (3.24) and (2.10), which we have named “Model 3” according to the number of lags. Differences between the results may be due to the higher number of variables in the model with lag respect to the simplest model with a single transition matrix. Although, we see how the differences between the series tend to shrink in some periods with respect to other periods. Both outputs refer to smoothed values and hence, the motivation of the unlike behavior is inherent to the lag dependency. In some periods the model has a strong dependency on the lags and in other periods it depends on the unique lag.

Figure 3.3 compares extracted filter series for both models. Each graph refers to a single time series for hedonic price for components. In the first graph of the example in figure

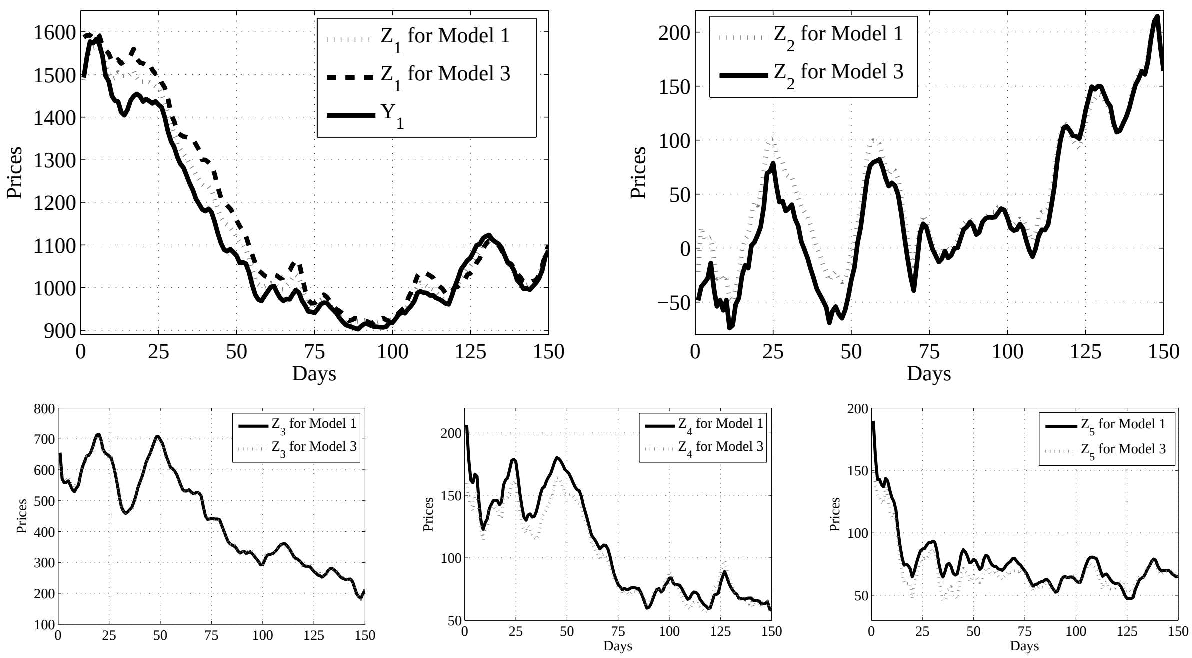

Figure 3.3: Estimated Implicit Prices for two different models, with one or three lags, in the game 0001tac

3.3, it coincides with an initial periods where hedonic prices are better valued from *Model 1* with respect to *Model 3*. But in other games we have tested the opposite situation where hedonic price for base product computed by *Model 3* are closer than hedonic prices of base product computed by *Model 1* to the product price of computer number one. We conclude that the lagged model provides similar results for the hedonic price series. But what can we say about the weight of dynamic multipliers in the three lags.

We have seen in the lag one case that dynamic behavior of the model may be measured in a robust way by the dominant eigenvalue. It is well known that in the lag one case the eigenvalue of matrix $\hat{\mathbf{\Phi}}$ with largest absolute value (spectral radius) is often very close to unity. We see the results for $\rho(\mathbf{\Phi})$ in table 2.4. In the three lag case we found that there exists a relation between the dominant eigenvalue of the single matrix of *Model 1*, $\rho(\mathbf{\Phi})$, and the three dominant eigenvalues of the matrices in the *Model 3*, which we call $\rho(\mathbf{\Phi}_i), i = 1, 2, 3$. We found that in the relation linking the models, given by:

$$\rho(\mathbf{\Phi}) \approx \rho(\mathbf{\Phi}_1) + \rho(\mathbf{\Phi}_2) + \rho(\mathbf{\Phi}_3), \tag{3.25}$$

the sign of the central matrix is always negative to compensate the first matrix, whereas the third sign is always positive. In terms of values the third matrix has an average weight about 20% in the system. But the hedonic prices are prevalently obtained from the third

Table 3.1: Eigenvalues of $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{\Phi}}_i, i = 1, 2, 3$ in ten TAC SCM games. In parentheses the proportion of the second, and third eigenvalue over the first eigenvalue is given. Last column show the validity of the relation (3.25)

TAC Game	Dominant Eigenvalue of Transition Matrices				Differences between Max. Eig.
	$\rho(\hat{\mathbf{\Phi}})$	$\rho(\hat{\mathbf{\Phi}}_1)$	$\rho(\hat{\mathbf{\Phi}}_2)$	$\rho(\hat{\mathbf{\Phi}}_3)$	$\rho(\hat{\mathbf{\Phi}}) - [\rho(\hat{\mathbf{\Phi}}_1) + \rho(\hat{\mathbf{\Phi}}_2) + \rho(\hat{\mathbf{\Phi}}_3)]$
1	1.0003	2.3094	-1.8294 (0.79)	0.5199 (0.23)	1.0003 - 0.9999 = 0.0004
2	1.0403	2.0106	-1.3738 (0.68)	0.4828 (0.24)	1.0403 - 1.1196 = -0.0793
3	0.9940	2.0173	-1.5546 (0.77)	0.4951 (0.25)	0.9940 - 0.9579 = 0.0361
4	0.9976	1.8865	-1.4078 (0.75)	0.5303 (0.28)	0.9976 - 1.0090 = -0.0114
5	0.9976	2.1866	-1.6725 (0.76)	0.4753 (0.22)	0.9976 - 0.9894 = 0.0082
6	1.0046	2.5468	-2.4028 (0.94)	0.8747 (0.34)	1.0046 - 1.0186 = -0.0140
7	1.0042	1.8920	-1.3236 (0.70)	0.4560 (0.24)	1.0042 - 1.0244 = -0.0202
8	0.9934	2.2886	-1.6470 (0.72)	0.3560 (0.16)	0.9934 - 0.9976 = -0.0042
9	0.9960	1.8209	-1.1676 (0.64)	0.4291 (0.24)	0.9960 - 1.0824 = -0.0863
10	0.9975	1.8328	-1.2103 (0.66)	0.3953 (0.22)	0.9975 - 1.0178 = -0.0203

lag values: around 50% in all the games, except a minimum value of 36% and a maximum value of 87%. The rest of the product price is explained from the first and second lagged values. We deduce that there are few games where hedonic prices are strongly correlated to the previous values at lag three.

It seems logical that the behavior of the model is differently affected by the three lagged values. In the table 3.1 there are listed the eigenvalues estimated for the set of 10 games. The games used to test the lagged model are different from the games used in the previous chapter so we listed also results for simplest model. In the same table we included the relative values in the case of dominant eigenvalues of *Model 3*. Following those values we can compare games to find which are the correlations between lag values similarly to the univariate partial autocorrelation plot.

For the model with three lags, we measured the forecast performances as well as the previous model. In this case, after the estimation of the hedonic prices, we calculate future prices for each product by:

$$\hat{\mathbf{y}}_{T+h} = \mathbf{D}\hat{\mathbf{z}}_{T+h}, \tag{3.26}$$

whereas the predicted hedonic prices are calculated by:

$$\hat{\mathbf{z}}_{T+1} = \hat{\mathbf{\Phi}}_{\mathbf{1}}\hat{\mathbf{z}}_T + \hat{\mathbf{\Phi}}_{\mathbf{2}}\hat{\mathbf{z}}_{T-1} + \hat{\mathbf{\Phi}}_{\mathbf{3}}\hat{\mathbf{z}}_{T-2}. \tag{3.27}$$

The mean relative error for the h period ahead prediction is calculated by:

$$ME_{T,h} = \left(\sum_{g=1}^{10} \sum_{j=1}^{16} \frac{|y_{j,T+h}^{(g)} - \hat{y}_{j,T+h}^{(g)}|}{np_j} \right) / (10 \cdot 16), \quad h = 1, \ldots, 40, \tag{3.28}$$

Table 3.2: Relative mean error of forecast values for several values of T, the input series length, and h, the ahead period for predictions, in ten TAC SCM games.

$ME_{T,h}$	Ahead Period (h)									
T	1		5		10		20		40	
	Mdl 1	Mdl 3	Mdl 1	Mdl 3	Mdl 1	Mdl 3	Mdl 1	Mdl 3	Mdl 1	Mdl 3
10	2.8	4.5	5.0	6.8	10.3	10.0	40.7	18.5	23.6	34.3
30	121.6*	2.5	129.6*	4.5	115.6*	6.4	108.4*	13.3	110.4*	196.6
50	2.8	2.9	4.1	4.7	6.2	7.1	10.0	12.0	17.6	25.0
90	3.6	3.6	5.7	5.6	9.4	8.8	20.6	16.1	109.8	55.7
130	4.0	4.1	4.9	4.7	6.8	6.4	9.9	9.5	13.9	13.6
170	4.0	4.0	5.6	5.8	6.8	7.4	10.1	11.2	16.1	17.8
210	6.5	6.5	12.7	12.2	-	-	-	-	-	-

* Algorithm does not provide good results for the game number five.
For this reason, performances for input series of length 30 are corrupted in the case of model 1.

where the upper script (g) is referred to the game, and the np_j are the nominal prices of sixteen types of product. The latter are given in table 2.2 by the sum of base prices for components:

$$np_j = AssCost_j + \sum_{i=j}^{numParts} NomPartCost_{i,j}, \tag{3.29}$$

where $NomPartCost_{i,j}$ is the nominal cost of the i-th part for good j, $numParts$ is the number of parts needed to make the good j, and $AssCost_j$ is the cost of manufacturing the good j. A nominal component cost is defined as the reference price for an individual component known from each agent at the beginning of the game. It may be used to normalize the mean error for comparison between performances. In the next chapter, the forecast indexes are developed and extended. In table 3.2 there are listed mean error performances for both models tested in ten games. Unfortunately, the algorithm with a unique lag provides unreal results in the period $T = 30$. In the next chapter we will cope with this problem and we provide a solution. In all other cases, algorithm with three lags starts to give best result after at least 90 periods, whereas for short estimation windows it is preferable to use the simplest *Model 1*. Also in the last days of the game, when high variability affects product prices, *Model 1* provides a better output than *Model 3*. Thus, we conclude the lagged model application in TAC SCM observing the opportunity of a change in the model for those periods where mean reversion is stronger than other periods.

3.3 The Problem of Identification of Characteristics: the Premium Variables

Here, we consider a possible extension of the basic state-space formulation in the measurement equation: the model with the partition in hedonic prices for components and for the entire product. Instead of considering a disturbance vector to model the part of product prices do not affected by component evaluations, we include a new vector of state variables of hedonic prices for the entire product seen as a sum of individual components. Eventually, if the vector is quite similar for the product variety, we may substitute it with a unique parameter, the average premium for product. We call those variables *premiums*, in the sense of global evaluation of the product, and of "surplus" over the customer evaluation. Next subsection illustrates the methodology for the extraction of hedonic prices and premium values, and the estimation procedure for the parameters. Last sub-section gives an application of the "premium" model in TAC SCM.

3.3.1 Extraction of the Premium from Product Price and Estimation of Unknown Parameters

For a customer, a product consists of components but its value is given by the sum of each component value plus a holistic value for the entire product, the premium. In figure 3.4 we see the new classification of the variables which can be extracted by a product price in a spectrum or variety of products.

The premium is typical of the dynamic pricing markets since in such markets agents have the opportunity to not satisfy uncertain demand for the unpredictable production times. In the previous sub-section we tried to estimate parameters both for hedonic and premiums process, and we saw that this is possible only when the researcher knows the parameter of one of the process. Premium process is not symmetric and in many applications negative premiums are rare events. They corresponds to the agent default for the previous customer demand. In this situation, agent prefers to sell out its production against the risk of high levels in final inventory.

Under the inclusion of premiums, the new model relations are:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{D}\mathbf{z}_t + \mathbf{v}_t + \boldsymbol{\nu}_t \\ \mathbf{z}_t &= \boldsymbol{\Phi}\mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t \\ \mathbf{v}_t &= \boldsymbol{\Psi}\mathbf{v}_{t-1} + \boldsymbol{\eta}_t, \end{aligned} \tag{3.30}$$

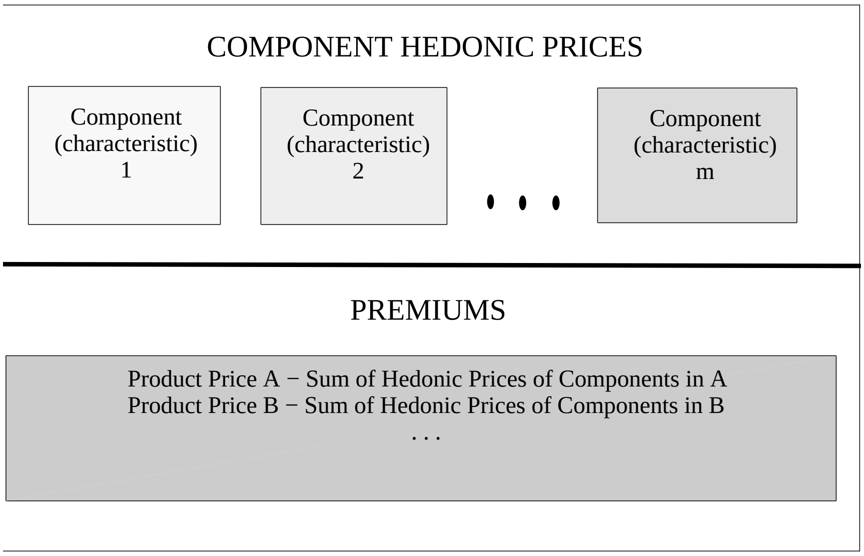

Figure 3.4: Classification of product variety space of variables in two complementary subsets. We individuate m components of a range of products and n complementary variables (premiums) for each single product

where $\mathbf{v}_t$ is the n dimensioned vector of premiums, $\boldsymbol{\Psi}$ is the transition matrix for them, and $\boldsymbol{\eta}_t$ is the n dimensioned vector of random disturbances, which is Gaussian distributed with zero mean and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$. We may rewrite (3.30) as:

$$\mathbf{y}_t = [\,\mathbf{D} \quad \mathbf{I}\,] \begin{pmatrix} \mathbf{z}_t \\ \mathbf{v}_t \end{pmatrix} + \boldsymbol{\nu}_t$$
$$\begin{pmatrix} \mathbf{z}_t \\ \mathbf{v}_t \end{pmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi} \end{bmatrix} \begin{pmatrix} \mathbf{z}_{t-1} \\ \mathbf{v}_{t-1} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\varepsilon}_t \\ \boldsymbol{\eta}_t \end{pmatrix}, \tag{3.31}$$

or, with a change of notation:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{H}\mathbf{x}_t + \boldsymbol{\nu}_t \\ \mathbf{x}_t &= \boldsymbol{\Upsilon}\mathbf{x}_{t-1} + \boldsymbol{\omega}_t \\ \boldsymbol{\nu}_t &\sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}}) \\ \boldsymbol{\omega}_t &\sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}}) \end{aligned} \tag{3.32}$$

where $\mathbf{H}$ is a $n \times (m+n)$ matrix such that $\mathbf{H} = [\mathbf{D} \quad \mathbf{I}]$, $\mathbf{x}$ is the $(m+n) \times 1$ vector of state variables which the first m elements are the hedonic prices of components and the last n

elements are the premium for products. The transition matrix for state variables is now:

$$\mathbf{\Upsilon} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Psi} \end{bmatrix}. \tag{3.33}$$

Through (3.33) we assume that evolution for hedonic prices is not affected by evolution of premiums, or equivalently we assume that dynamic multipliers between two variables are all zeros. Last assumption is for the disturbances of the measurement equation. The covariance matrix may be assumed partitioned (uncorrelated disturbances between hedonic and holistic prices), or full (disturbances between hedonic and premiums are correlated). In the sequel of this section, we do not restrict the covariance matrix but we will accept full matrices for it.

The hedonic algorithm must be now changed in several steps to allow the extraction of the new state vector $\mathbf{x}$ in (3.32). The modifications are:

1. the innovations are now given by $\mathbf{e}_t = \mathbf{y}_t - \mathbf{z}_t - \mathbf{v}_t$

2. in the M-Step of the EM algorithm, we differentiate the Q function respect to the new hyperparameter elements, $\mathbf{\Theta}^{(\mathbf{j})} = \{\mathbf{\Upsilon}^{(\mathbf{j})}, \mathbf{\Sigma}_{\nu}^{(\mathbf{j})}, \mathbf{\Sigma}_{\omega}^{(\mathbf{j})}, \boldsymbol{\mu}_{\mathbf{0}}^{(\mathbf{j})}, \mathbf{\Sigma}_{\mathbf{0}}^{(\mathbf{j})}\}$. Since we have to differentiate respect to individual matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$, the matrices in $\mathbf{\Upsilon}$, we rewrite the quantities in (2.15)–(2.17) partitioned as:

$$\mathbf{S_{11}} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{B}_{11} \\ \mathbf{B}_{11}' & \mathbf{C}_{11} \end{bmatrix}, \quad \mathbf{S_{10}} = \begin{bmatrix} \mathbf{A}_{10} & \mathbf{B}_{10} \\ \mathbf{C}_{10} & \mathbf{D}_{10} \end{bmatrix}, \quad \text{and } \mathbf{S_{00}} = \begin{bmatrix} \mathbf{A}_{00} & \mathbf{B}_{00} \\ \mathbf{B}_{00}' & \mathbf{C}_{00} \end{bmatrix},$$

similarly to the transformation for the lagged case in (3.10).

3. the central term in (2.14), which we rewrite for convenience:

$$\text{tr}\ \{\mathbf{\Sigma}_{\omega}^{-1}[\mathbf{S_{11}} - \mathbf{S_{10}}\mathbf{\Upsilon}' - \mathbf{\Upsilon}\mathbf{S}_{\mathbf{10}}' + \mathbf{\Upsilon}\mathbf{S_{00}}\mathbf{\Upsilon}']\}$$

is now:

$$\text{tr}\left\{\mathbf{\Sigma}_{\omega}^{-1}\left\{\mathbf{S_{11}} + \begin{bmatrix} -\mathbf{A_{10}}\mathbf{\Phi}' - \mathbf{\Phi}\mathbf{A_{10}}' + \mathbf{\Phi}\mathbf{A_{00}}\mathbf{\Phi}' & -\mathbf{B_{10}}\mathbf{\Psi}' - \mathbf{\Phi}\mathbf{B_{10}}' + \mathbf{\Phi}\mathbf{B_{00}}\mathbf{\Psi}' \\ -\mathbf{D_{10}}\mathbf{\Phi}' - \mathbf{\Psi}\mathbf{D_{10}}' + \mathbf{\Psi}\mathbf{B_{00}}'\mathbf{\Phi}' & -\mathbf{C_{10}}\mathbf{\Psi}' - \mathbf{\Psi}\mathbf{C_{10}}' + \mathbf{\Psi}\mathbf{C_{00}}\mathbf{\Psi}' \end{bmatrix}\right\}\right\}.$$

According to the notations in (3.11) we bipartite $\mathbf{\Sigma}_{\omega}$. The derivatives of the trace with respect to the two transition matrices set to zero give the following system:

$$\begin{cases} 2\boldsymbol{\Sigma_\omega}^{(11)}\mathbf{A}_{10} + \boldsymbol{\Sigma_\omega}^{(12)}\mathbf{C}_{10} + \boldsymbol{\Sigma_\omega}'^{(12)}\mathbf{B}_{10} - 2\boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{(11)}\mathbf{A}_{00}\boldsymbol{\Phi} + 2\boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{(12)}\mathbf{B}_{00}\boldsymbol{\Psi} &= 0, \\ 2\boldsymbol{\Sigma_\omega}^{(22)}\mathbf{D}_{10} + \boldsymbol{\Sigma_\omega}^{(12)}\mathbf{C}_{10} + \boldsymbol{\Sigma_\omega}'^{(12)}\mathbf{B}_{10} - 2\boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{(22)}\mathbf{C}_{00}\boldsymbol{\Psi} + 2\boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{(12)}\mathbf{B}_{00}\boldsymbol{\Phi} &= 0. \end{cases} \tag{3.34}$$

Unfortunately, the resulting specification is not sufficiently restrictive to be able to identify the unknown parameters. Which are the possible solutions to estimate together hedonic prices and premiums? First solution can be estimate a constant instead of a variable which collects the premium information for all the products. Second solution is given by the estimation of hedonic prices via minimum product prices. We analyze the latter solution in the next sub-section.

3.3.2 Hedonic Prices and Minimum Prices

In this paragraph, we introduce the minimum product price series. It will be used for extracting hedonic information as an alternative definition to the basic model of section 2.2. Furthermore, we avoid the problem of estimation of parameters arose in the previous sub-section.

In many situations, market reports minimum and maximum daily prices for a product variety. According to the work of Rosen, customers evaluate parts observing essentially minimum prices in the market. Thus, the premium consists in the difference between average price and the minimum price in the market. Hedonic prices can be evaluated by the minimum price series, whereas the premium are extracted by residual series. Replacing them as input series is possible to estimate premiums from the following model:

$$\begin{aligned} \mathbf{y}_t &= {}_m\mathbf{y}_t + \mathbf{v}_t \\ {}_m\mathbf{y}_t &= \mathbf{D}\mathbf{z}_t + \boldsymbol{\nu}_t \\ \mathbf{z}_t &= \boldsymbol{\Phi}\mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t \\ \mathbf{v}_t &= \mathbf{y}_t - {}_m\mathbf{y}_t, \end{aligned} \tag{3.35}$$

where ${}_m\mathbf{y}_t$ is the $n \times 1$-vector of minimum prices, and $\mathbf{y}_t$ is the series of average prices at the time t in the market, as defined in section 2.3.2. The decision to leave the disturbances in the measurement equation arise from the extraction of hedonic prices with the consequent generation of noise.

Now, we test the model in a different way, following the relations in (3.35), and estimating product prices with the smoothed mid-range price, and the smoothed minimum prices. Then, we have:

$$
\begin{aligned}
{}_R\tilde{\mathbf{y}}_t &= {}_m\tilde{\mathbf{y}}_t + \hat{\mathbf{v}}_t \\
{}_m\mathbf{y}_t &= \mathbf{D}\mathbf{z}_t + \boldsymbol{\nu}_t \\
\mathbf{z}_t &= \mathbf{\Phi}\mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t \\
\hat{\mathbf{v}}_t &= {}_R\tilde{\mathbf{y}}_t - {}_m\tilde{\mathbf{y}}_t.
\end{aligned}
\tag{3.36}
$$

We extract hedonic prices from the sole smoothed minimum price series according the algorithm in the previous chapter. After the computation of the smoothed mid-range prices (the input of the algorithm) we calculate the estimate series of premiums for single products. The latter is quite smoothed because is obtained as difference of smoothed series.

3.3.3 The extraction of Premiums in TAC SCM

In TAC SCM every day market reports both minimum and maximum price for each product $i, i = 1, \ldots, n$. In the previous applications we used the smoothed mid-range product price series to extract information about hedonic prices for components. Now, we repeat the algorithm but the input series is given by ${}_m\tilde{\mathbf{y}}_t$.

In figure 3.5, in the upper graph, we compare hedonic prices computed via the classic hedonic model with the output provided by model utilizing minimum prices. Differences are relevant, overall for intervals where the market is characterized by high premiums. For instance, at the 40th day, $z_{1,40}$ is higher than the same hedonic price extracted by the model in (3.36). Also the hedonic price for the CPU computed through the model in (2.9)-(2.10) is higher than ${}_mz_{3,40}$, the hedonic price extracted via minimum price series. On the contrary, in the last days, when the hedonic price for CPU dramatically drops, the values for $z_{3,(\cdot)}$ are lower than ${}_mz_{3,(\cdot)}$. It means that implicit price distances are significantly correlated with premiums largeness. In the graphs of premiums, we see the advantages of on line extraction of those variables since they are correlated in long interval of time. The agent-manufacturers which had produced IMD computers had started to get profits after one hundred days. But higher premiums in the interval (150,220) are correlated to IMD computers with SKU from nine to twelve, without the upgraded central processing unit. An important scope of hedonic model is for the forecast of trends for parts together the premium analysis. For instance, if a manufacturer have predicted the drop of CPU in the final of the game, he would focus on the other computers with a better scheduling. Observing the high premiums for computers IMD branded around 150th day, he could select the production for those computers of IMD not including the CPU.

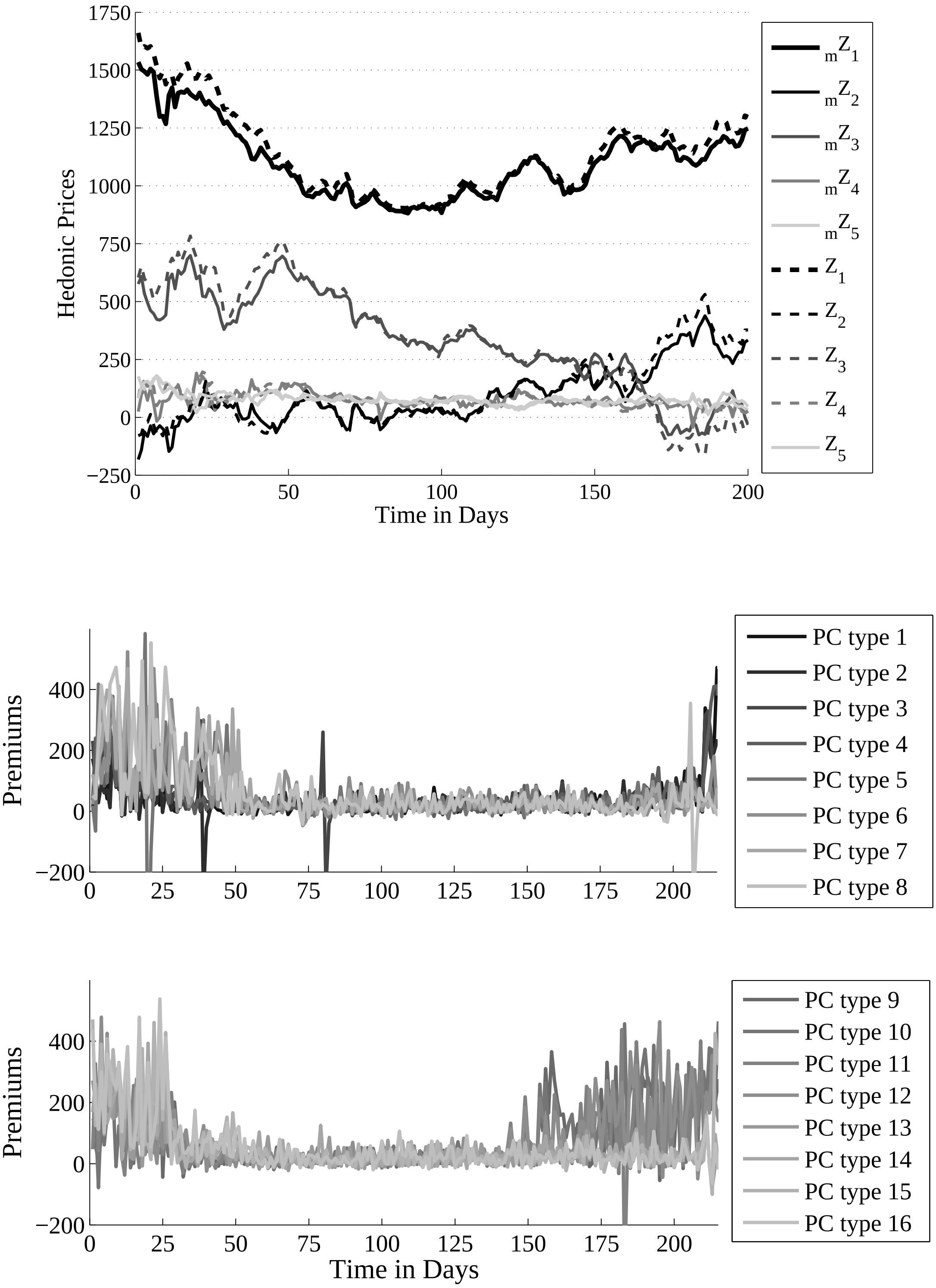

Figure 3.5: Upper: estimated implicit prices for two different models (basic and premium). Middle and Bottom: estimated premiums in the same game 0001tac, for PINTEL computers (middle) and IMD computers (bottom)

Table 3.3: Mean values, sample standard deviations in games, and percentage with respect to nominal prices of ${}_{m}z_{i,t}$, in 30 TAC SCM games

PINTEL premium statistics				IMD premium statistics			
SKU	Mean	Std	Percentage	SKU	Mean	Std	Percentage
1	47.2	20.6	2.86	9	48.8	21.2	2.96
2	49.6	18.0	2.83	10	50.8	19.5	2.90
3	55.2	20.9	3.15	11	55.2	20.7	3.15
4	56.0	20.6	3.03	12	55.1	19.7	2.98
5	65.5	27.4	3.05	13	63.1	26.5	2.93
6	66.4	25.1	2.95	14	62.6	21.2	2.78
7	72.8	28.7	3.24	15	70.2	26.4	3.12
8	73.5	29.2	3.13	16	70.2	23.1	2.99

The computation of premiums can lead to an easy detection of periods of high/low dynamic pricing as figure 3.5 outlines for the game 0001tac. High premiums are correlated to high volatility periods for product and hedonic prices. In the initial days, prices slowly decrease due to the diminishing of cost for parts. Also hedonic prices show the same behavior linked to the customer needs.

In table 3.3 are reported descriptive statistics for premiums in 30 games. While mean values show some differences between PINTEL and IMD products, t-tests for equal means not reject the null hypothesis both for equal and not equal variances assumption (t-student and Behrens-Fisher problems). The average premium is homogeneous for the product variety. It is around the 3% of the nominal prices for any product (e.g. the average premium of product with SKU five is around 66, the 3.05% of the nominal cost of 2150).

Tests for whiteness and normality of residuals for the model in (3.36) show better results than those in section 2.3.5. Assumptions are valid if the length of input series is shorter than 25 days.

3.4 Conclusions

We have examined a set of hedonic models under several assumptions. To facilitate the comprehension and summarize the different techniques for extracting hedonic information we report in table 3.4 all the models and correspondent algorithms detailed in previous chapters. In the table we have listed them according to the order we have faced them in the thesis. We omitted the algorithm based on the third stopping rule in (2.33). In fact, as we have seen in the first application of chapter 2, it is very difficult to establish a threshold value for δ_3.

Obviously, there are many other specifications for the hedonic model. For instance, a

Table 3.4: Hedonic models and their algorithms

Name of the Model	Name of the Algorithm	Comments
Base Hedonic Model	Algorithm 1	It uses the first stopping rule based on the distance between the transition matrices in two close iterations
Base Hedonic Model	Algorithm LR	It uses the second stopping rule based on the likelihood ratio in two close iterations
Noise Model	Algorithm 2	The simplest model. Random walk for the hedonic process
Diagonal Model	Algorithm 3	The hedonic process is simplified via the assumption of zero entries for the transition matrix except for its diagonal
Lagged Hedonic Model	Algorithm LG	The hedonic process is extended to include 3 lags in the transition equation
Premium Model	Algorithm PR	Hedonic prices are extracted by minimum prices and premiums are determined for a parallel process

colored model must be tested to verify the hypothesis for the presence of a dynamic system also for disturbances, both in measurement and transition equation. Another interesting modeling is given by the introduction of market trend and seasonality effects. By the way, in the next chapter we will start from the simplest model because an example of a high multivariate methodology is needed in the case of EM + KF procedures.

Chapter 4

Real Time Hedonic Model

In this chapter we want to extend the basic hedonic algorithm for real time applications. We will explore new algorithms to extract hedonic prices from time series of product prices generated periodically (day by day) from the negotiations between manufacturers and customers. We shall expose the complications in implementing such algorithms due to the difficulties that we have already faced in the second chapter. They are focus on the stopping rule used for convergence of Kalman filter together Expectation-Maximization algorithm. Two methodologies are given to overcome those difficulties. A first algorithm with a drastic reduction in computations, and a second algorithm which can be considered adaptable in many contexts. They consist in the second contribution of the thesis. When the dimension of the vectors for input and state variables is high in a state space model with unknown parameters, the usual stopping rule based on the likelihood should be substitute by the stopping rule in (2.24). Usually authors uses the convergence criteria between two log-likelihood values because in standard cases the behavior of the log-likelihood is monotonically decreasing. Differently, when Kalman filter and EM are mixed, the log-likelihood can show a strange form overall in the nearing of the solution due to the Kalman filter gain. The latter, which approaches to the zero values in the nearing of the solution, provides anomalous values for the log-likelihood. Consequently, the behavior of the log-likelihood is sometimes not monotonically decreasing. If we use the likelihood ratio test as convergence criteria the problem does not appear because the convergence is reached before the Kalman filter gain approaches to zero. But if the criteria is given by (2.24) the problem is not rare and it affects hedonic price extraction as well as forecast performances.

In the following section we outline the characteristic of the real time hedonic model with respect to the convergence steps. The final framework must be used by manufacturers day by day to take notes of the market situation in hedonic sense. The algorithm must provide a good approximation of the actual parameters of the system and, for that reason, a refinement of the stopping rule and a test for acceptance of the parameters will be introduced. The *Algorithm 1* for the base model will be modified in two different algorithms for including those modifications. A generative data set is created to estimate of the level of approximation provided by the algorithm.

4.1 A Convergence Criteria for Real Time Hedonic Model

To adapt the previous algorithm for real time application we need to iterate the extraction of hedonic prices from time t_0 until the last time in day, T. In section 2.2.3 we have anticipated some details about the usual problem that a researcher meet in the Kalman filter algorithm implementation. For instance, in real-time imaging of satellite systems the large amount of data and the short time of computations between two time windows requires a fast algorithm, the fast Kalman filter. Here, we point to an optimal algorithm, a procedure with low probability of false solutions to be implemented in our supply chain context with many suppliers, manufacturers, and customers, where parameters are unknown. In multivariate case, the dimensions of the input and output vector, (n, T) and (m, T), determines the largeness of the entire system underlying the system of n measurement equations and m transition equations. Despite of facts to use some symmetrization rules for the covariance matrix that do not even assure an acceptable output for the filter, we prefer to analyze the behavior of the filter together the Expectation-Maximization algorithm in the particular case of design matrices, and try to limit the number of iterations of algorithm in the outer loop. Obviously, if the model was misspecified we meet an high number of divergence cases.

In several of the applications of *Algorithm 1* and *Algorithm LR* we have measured the convenience to use the stopping rule based on the distance between the transition matrices instead of the likelihood ratio rule. The explanation of that result is linked to the misspecification of the normality assumption together the high dimensionality of the likelihood function. Obviously, we want to start from that point to elaborate the best algorithm for the estimation of parameters and hedonic prices. In *Algorithm 1* the extraction filter diverges when the solution of the maximization problem for the unknown parameters is near. Across the first stopping rule the algorithm reaches out the likelihood ratio estimate and outperform the second stopping rule. But there is the risk that Kalman filter diverges. When divergence occurs the estimates can be wrong. The advantage is the precision of a large amount of parameters, whereas the disadvantage is the risk of several wrong estimates worse than likelihood ratio ones. Our technique is based on the research of the best stopping time for the first rule. Then, we will test a decision rule to omit the wrong parameters.

In the next chapter we analyze forecast performances. The advantage is twofold : they measure the parameter performances and secondly, provide an increasing in the performances of the model in terms of cost and revenues. Forecast methodologies must be viewed like a test for goodness of parameters (Lutkepohl, 2005) obtained applying the following algorithms.

4.1.1 Outer and Inner Iterations: Computational Complexity

In the hedonic algorithm of section 2.2.2, we introduced two kind of iterations. For simplicity, we called them *inner iterations* and *outer iterations* (see figure 2.3). Hence, we can call the "large loop", the loop referring to the outer iterations and the "small loop", the loop for inner iterations. Large and small refer to the number of operations to be completed in each of them. In the large loop the algorithm estimates both hedonic prices and parameters, whereas in the small loop, only the hedonic prices are estimated via Kalman filter, from the initial time until the time T. In a real time framework (see figure 4.1), we have to update hedonic estimates and hyper parameter in each period t taking into account the information accumulated in the past. The main differences between two algorithm are:

(i) a maximum number of outer iterations instead of a simple stopping rule for breaking the loop;

(ii) a test procedure on the output of outer iteration after the extraction of the hedonic prices and the hyper parameter.

In figure 4.1 is represented an example of the real time algorithm which extract hedonic information from time series of 16 product prices sharing five components like in the TAC SCM testbed. Here, the game lasts T days and the algorithm starts to extract prices and parameters from the day t_0. In each period t, it computes the new filter value $\mathbf{z}_t|y_t$ of the vector of hedonic prices, and re-calculate the entire smoothed series $\mathbf{z}_0|\{y_1,\ldots,y_t\},\ldots,\mathbf{z}_t|\{y_1,\ldots,y_t\}$. Using the smoothed series, it computes the expectation of the incomplete data log-likelihood for the estimation of the hyper parameter. The reason for a limited number of outer iterations is twofold: to limit as much as possible the risk of divergence of the algorithm, and to speed the computation times for each period. The reason for the test procedure is inherent to the real time implementation of the algorithm: in each period we need a valid estimate to submit to the user. If the test provides not rejection for the estimated parameters and values the algorithm output for the day t shows hedonic prices, parameters, and forecast values of product prices. Otherwise, if the test result is negative the algorithm does not offer

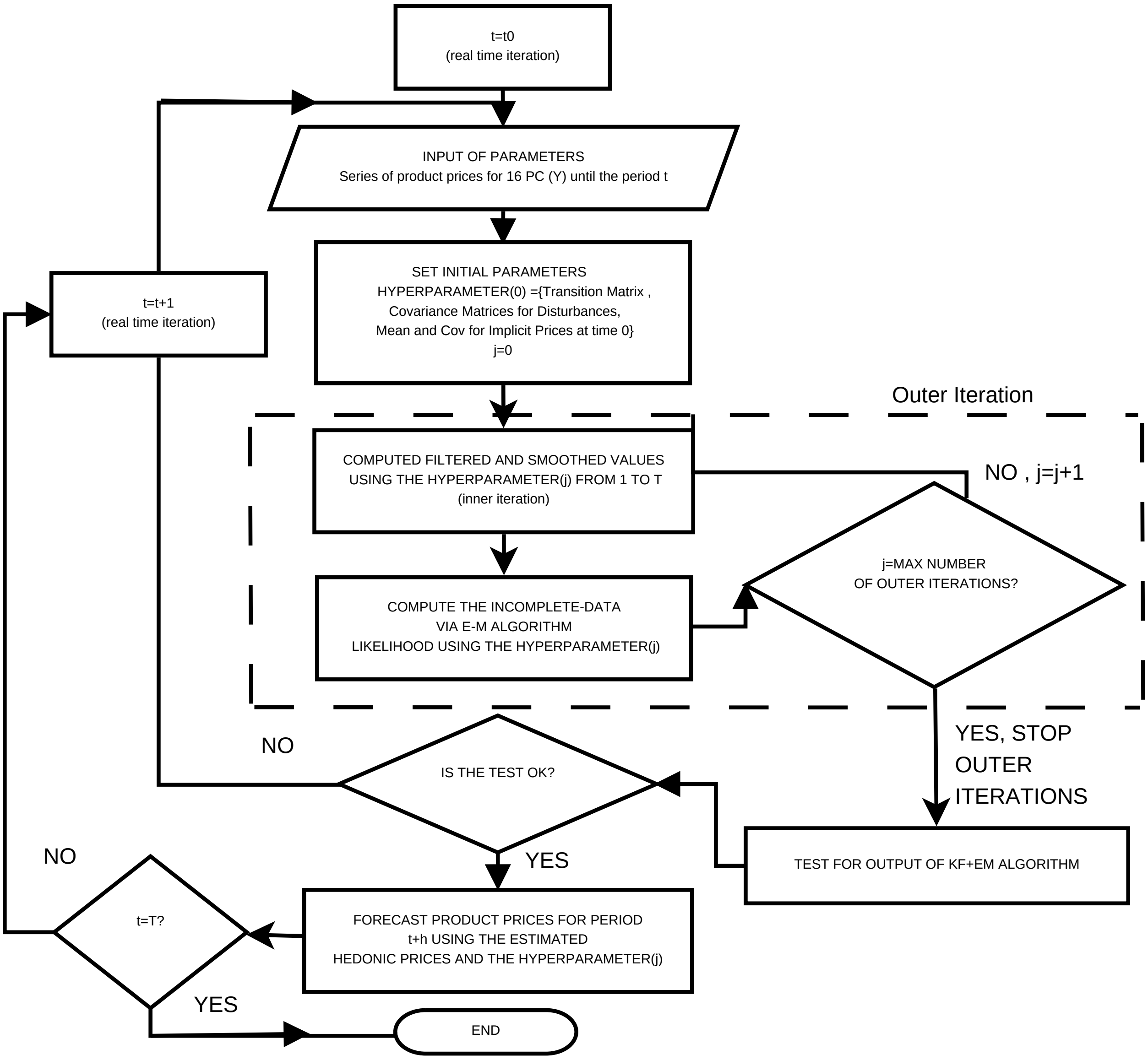

Figure 4.1: Illustration of the steps and iterations of the hedonic algorithm in the real time model

an output for that period or a less restrictive convergence criteria is chosen (e.g. a greater value of $delta_1$).

For assigning a maximum number to outer iterations we explored times for convergence and the probability of failure in off line simulations and historical training data. As seen in previous chapter, we opt for the convergence criteria given in (2.24) for measuring convergence, the best stopping rule tested in the TAC SCM application of chapter two. Observing the convergence for the absolute distance instead of the incomplete-data log-likelihood is a practical choice, overall in high dimensioned data, and in the following of this chapter we want to proof it. Anyway, the same technique may be extended including the likelihood-based distance as in (2.25). Our target is to calibrate the algorithm for the best output given by the minimum number of iterations or, equivalently, saving the maximum time for computations under an expected value for the transition matrix error of δ_1. For instance, in table 2.8 we saw how many iterations were needed in each of the nine games for the convergence of the algorithm depending on the setting value for δ_1.

In the next paragraphs, we will give two alternative methodologies which cap the number of iterations in Expectation-Maximization including Kalman filter algorithms (EM+KF), under the first stopping rule. Both algorithms are based on the results obtained in generative simulations of a process simpler than actual one but with similar characteristics. In a first step we generate a lot of product prices for different values of simplified transition parameter $\mathbf{\Phi}$ and we derive a distribution for the value of the absolute distance $\mathbf{n}_1$, depending on time and the kind of transition matrix. After this step, we can fix the minimum number of iterations required to obtain an output with a expected precision, and the maximum number of iterations required to avoid as much as possible ill conditioned matrices in Kalman filter procedure. In this way, we decide the correct interval for the number of iterations to be implemented in a specific application. Obviously, the result must provide much higher performances than likelihood ratio test criteria for convergence.

4.1.2 The Empirical Distribution for the Number of Iterations

We analyze the behavior of the iteration gains in the conjoint algorithm, EM+KF, when we work with high dimensioned data, under the stopping rule based on the distance between two transition matrices. Since, in each of the outer iterations, the conditional likelihood (or score function) of the process is greater than likelihood in the previous iteration, we can state that after j iterations our estimates must be better than the estimates at the iteration $j-1$, if the numerical error does not appear. Vice versa, if the error has corrupted the Kalman filter computations, the decreasing behavior of likelihood in EM algorithm stops.

How can we measure the probability of divergence of Kalman filter? We generate a lot of simulated time series for product prices for several transition matrices. According to the assumption of normality, the disturbances are generated by Gaussian processes with fixed diagonal covariance matrices. After the random generation of the noise vector $\boldsymbol{\varepsilon}_0$ for initial period, the vector $\mathbf{z}_0$ is derived from the transition equation. Multivariate normal random numbers are selected through the MATLAB@ function *mvnrnd*. Then, other two vectors for noises and disturbances, respectively $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\nu}_1$ are generated for obtaining $\mathbf{z}_1$ and $\mathbf{y}_1$. Iterating the procedure we obtain a time series of product prices of length T. Finally, we have a testbed of one thousand of time series of product prices generated by different transition matrices.

After the application of *Algorithm 1* with a value of $\delta_1 = 0.0025$, we can count the number of negative jumps in the log-likelihood depending on the number of outer iterations. We repeat the analysis for different length of the input series for the Kalman filter and several transition matrices in simple diagonal form. Each transition matrix tested has diagonal entries from 0.6 to 1.1. In this way, the maximum eigenvalue coincides with the value of one of the entry. We will see how results are depending on the maximum eigenvalue of the transition matrix of the system. Then, we apply *Algorithm 1* for several length of Kalman filter series, the input time series of product prices.

We generate observations (product prices) from the following s models, to represent a discretization of the entire space of simplified parameters:

$$\begin{aligned} &\mathbf{y}_t = \mathbf{D}\mathbf{z}_t + \boldsymbol{\nu}_t \\ &\mathbf{z}_t = \tilde{\boldsymbol{\Phi}}_s \mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t \\ &\boldsymbol{\nu}_t \sim \mathrm{MVN}(\mathbf{0}, \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\nu}}) \\ &\boldsymbol{\varepsilon}_t \sim \mathrm{MVN}(\mathbf{0}, \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\varepsilon}}) \\ &\mathbf{z}_0 \sim \mathrm{MVN}(\tilde{\boldsymbol{\mu}}_0, \tilde{\boldsymbol{\Sigma}}_0) \end{aligned} \tag{4.1}$$

where all the tilde values are fixed in the following way:

$$\tilde{\boldsymbol{\Phi}}_s = \alpha_s \mathbf{I}_m,\ \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\nu}} = 5000 \cdot \mathbf{I}_n,\ \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\varepsilon}} = \begin{bmatrix} \tilde{\sigma}_1 & 0 & \dots & 0 & 0 \\ 0 & \tilde{\sigma}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \tilde{\sigma}_{m-1} & 0 \\ 0 & 0 & \dots & 0 & \tilde{\sigma}_m \end{bmatrix},\ \tilde{\boldsymbol{\Sigma}}_0 = \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\varepsilon}},\ \tilde{\boldsymbol{\mu}}_0 = \begin{bmatrix} nc_1 \\ \dots \\ nc_m \end{bmatrix}. \tag{4.2}$$

Values for $\tilde{\boldsymbol{\mu}}_0$ may be set as the nominal component costs, whereas the variances for components must be set according to historical data. The identity matrix as selected in the transition matrix is due on the simplification for the simulation benchmark.

After the generation of data, we estimate the hedonic prices and the hyperparameter using our algorithm (for off line analysis) with initial assumptions given by:

$$\mathbf{\Theta}^{(0)} = \left\{ \tilde{\boldsymbol{\mu}}_0, \tilde{\mathbf{\Sigma}}_0, \tilde{\mathbf{\Sigma}}_{\boldsymbol{\nu}}, \tilde{\mathbf{\Phi}}_0, \tilde{\mathbf{\Sigma}}_{\boldsymbol{\varepsilon}} \right\}, \qquad \text{and } \alpha_0 = 0.95.$$

Summarizing, we use the same mean vector for the initial state mean, the same disturbance covariance matrices, but a different transition matrix very similar to the identity matrix.

Repeating the extraction for 1000 of simulations based on those parameters, we pass to analyze performances of EM+KF algorithm for each of them. For the analysis of convergence we define three indexes:

1. the first index based on the distance used in the algorithm of the previous chapter in (2.24), which measure the distance between matrices in two close outer iterations. We opt for the Manhattan distance between entries of the matrices, given by:

$$It^{(k,l)} = \sum_{i,j=1}^{m} \left| \phi_{ij}^{(k)} - \phi_{ij}^{(k-1)} \right|, \tag{4.3}$$

   where the transition matrix $\mathbf{\Phi}_{m \times m} = \{\phi_{ij}\}$ is estimated using product prices for l periods.

2. the Manhattan distance between the entries of the generating matrix $\tilde{\mathbf{\Phi}}_s$ and the entries of the matrix estimated at the kth outer iteration:

$$At^{(k,l)} = \sum_{i,j=1}^{m} \left| \tilde{\phi}_{ij} - \phi_{ij}^{(k)} \right|, \tag{4.4}$$

   where $\tilde{\phi}_{ij}$, with $i, j = 1, \dots, m$ are the elements of the actual matrix utilized to generate simulation data;

3. the Manhattan distance between the actual hedonic prices and the estimated hedonic values at the kth outer iteration:

$$Zt_j^{(k,l)} = \frac{\sum_{t=1}^{l} \left| \tilde{z}_{jt} - \hat{z}_{jt}^{(k)} \right|}{l}, \tag{4.5}$$

   where $\tilde{z}_{jt}$, with $j = 1, \dots, m$ and $t = 1, \dots, l$ are the actual hedonic prices generated in each simulation for the component j at the time t. The values of $\hat{z}_{jt}^{(k)}$ are the estimated hedonic prices via the EM+KF algorithm at the iteration k.

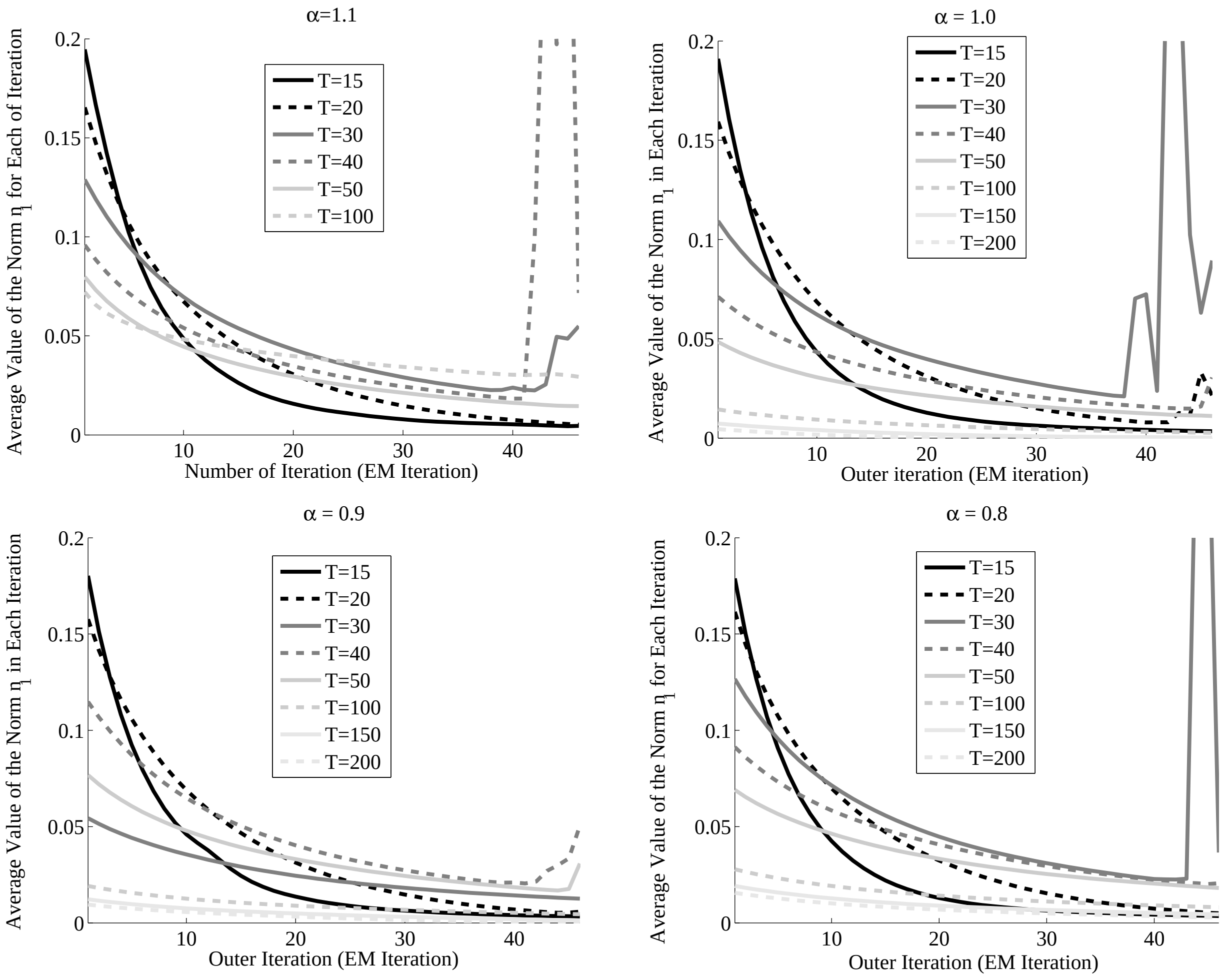

Figure 4.2: Average values for the convergence distance $\mathbf{n}_1$ for four transition matrices with $\alpha = 1.1, 1.0, 0.9, 0.8$, in the model $n = 16$ and $m = 5$, obtained in 1000 simulations

Last two indexes are very interesting because they measures the closeness and the effectiveness of the algorithm depending on the number of iterations, for the parameter (At), and for extracted values (Zt). The first index measures the speed of the EM convergence in diverse situations. It coincides with the absolute distance utilized in the off line algorithm.

4.1.3 Generative Model Results

Through the generative model data we can measure the performances of Kalman filter in terms of the indexes $It^{(k,l)}$, $At^{(k,l)}$, $Zt^{(k,l)}$. In the first graph in figure 4.2 we see the average values of the indexes $It^{(k,l)}$ for different values of l (the length of input series), and k, the iteration step of EM algorithm. They were computed for data obtained by a transition matrix with diagonal entries equal 1.1. They are mean values calculated over 1000 simulations. Note the jumps after 40 iterations due on the default of Kalman filter for ill conditioned covariance matrix when $T = 40$. Obviously, the Kalman filter is more stable for long time series as input but when the series is not stable the results for long time series are not more optimal. For this reason, in the first graph we avoid to show results for $T = 150$ or $T = 200$ periods. In both cases, the algorithm did not reach a solution. This do not represent a problem because high non stability or stationarity is typical of short periods under the one hundred days. Usually the maximum value of dominant eigenvalue reaches values a little higher than one, when T is over one hundred days.

In the other graphs of figure 4.2, we see how the combination of two algorithms works very well for time series longer than 100 elements after few iterations in the case of $\alpha \leq 1$. For instance, it is clear that an application in TAC SCM gives an optimal estimation of parameters after 100 days. of course the estimate is better than that for the algorithm with likelihood ratio stopping rule, since the convergence process is the longest. When the input time series has a length of $15, 20, 30$ periods the Kalman filter could fail before to reach the convergence of EM procedure. We see in the plots the jump of the lines which after few iterations usually come back to the standard values. This is due to the Kalman gain which becomes too much small providing ill-conditioned problems. Our calibration would be optimal if it is based on stopping the convergence of EM algorithm in a period between the 15 and the 35 iterations. But in this cases the distance $\mathbf{n}_1$ has an average value of 0.05 which corresponds to an error for each dynamic multiplier of the transition matrix around 0.002, the 0.2%. Since dynamic multipliers project the prices for a large number of days ahead, the error grows up if we consider long projection using short estimation windows as in the usual forecast methodologies. But, we will find that setting the calibration on a value of $\delta_1 = 0.0025$ the estimates can provide better forecast performances.

The shortest window for Kalman filter estimation ($T = 15$) generates the same shape for the plots, for all the test values of α. We can resume the behavior of the KF+EM algorithm observing that the flattening of the lines is correlated with the length of the estimation window but the convergence is better for extreme values of it. Thus, the problem of default is particular sensible for time windows around $30, 40, 50$ periods.

This kind of analysis will be very useful for time-varying extensions of original hedonic model. In time invariant state space model we point to transition matrix that characterize the behavior of hedonic prices in the entire history. In real time application we update in every period the estimate for the transition matrix. Observing the distribution of the number of iterations which are needed for the convergence, we accomplish to know the mean time of algorithm computations. Furthermore, we can calibrate it avoiding numerical errors for ill-conditioned cases.

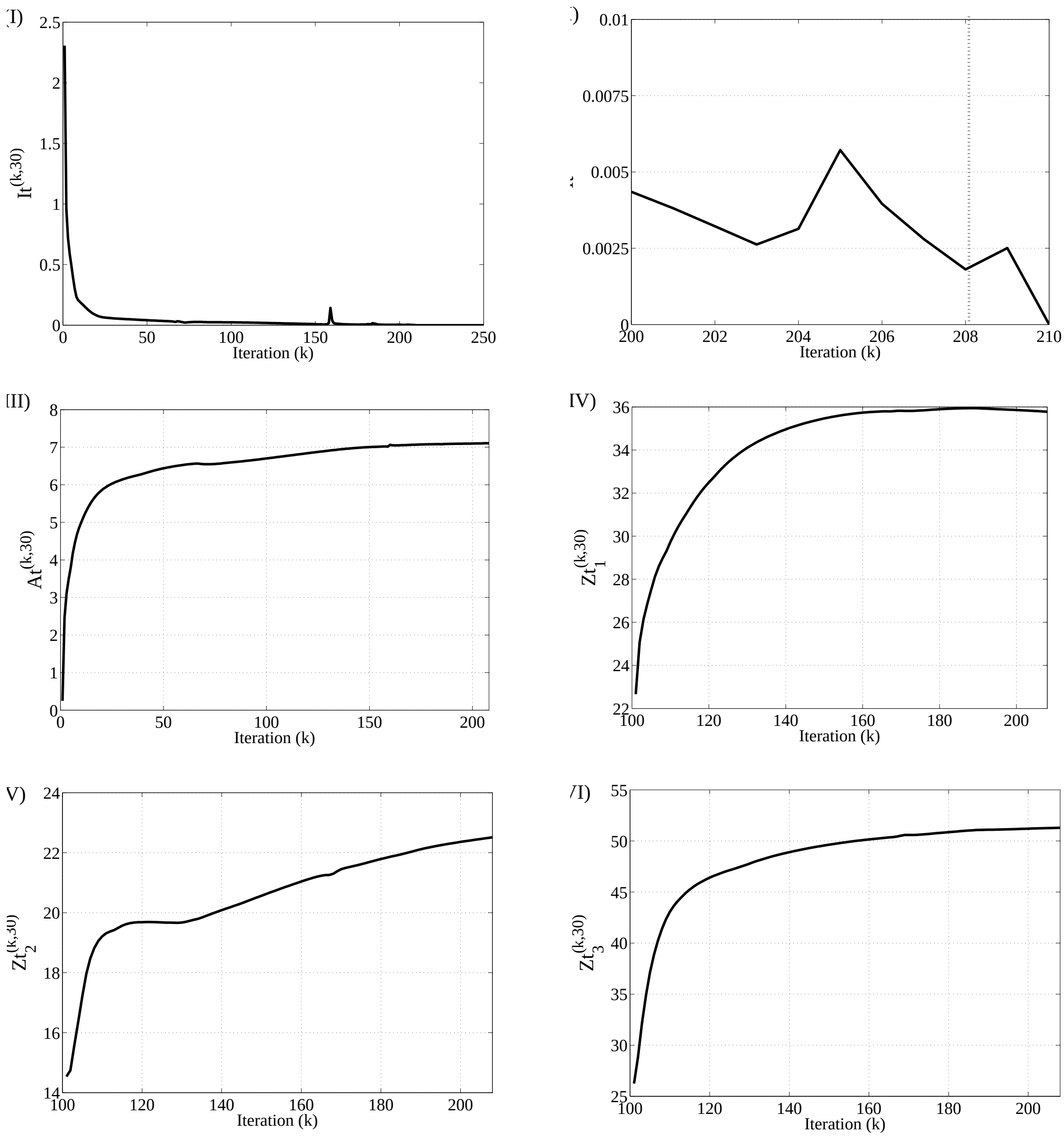

Figure 4.3: Convergence indexes in the first simulation for $\alpha = 1.0$ and with $T = 30$, m=5, n=16. (I) $It^{k,30}$, the stopping rule alternative to the likelihood ratio. (II) Detail of graph in (I) for $k \geq 200$. (III) $At^{k,30}$, the distance between the actual transition matrix of the generative system and the estimated one. (IV) $Zt_1^{k,30}$, the distance between the actual hedonic price for the base model and the estimated one. (V) $Zt_2^{k,30}$, the distance between the actual hedonic price for the motherboard differential and the estimated one. (VI) $Zt_3^{k,30}$, the distance between the actual hedonic price for the CPU differential and the estimated one.

In figure 4.3 there is an example of the results for the three indexes in the case of one convergent simulation for $\alpha = 1.0$, and $T = 30$: in this case convergence is determined for a value of $\mathbf{n}_1$ under the value of 0.0025, a good value returning estimates better than ones of other stopping rules. In the first and second graphs are reported the values of $It^{k,30}$. We saw in previous figure that in this case the average number of iterations can be affected by many jumps and possible divergence. Differently, in the top plots of figure 4.3, there is represented a positive case of convergence, that is reached at the iteration step 208. After 208 iterations, for the first time the value of $\boldsymbol{\delta}_1$ is below 0.0025. Around 160th iteration a small jump affects the decreasing trend, due to the problem of Kalman gain. While the value of $It^{155,30}$ is equal to 0.015, after few steps it arrives to 0.145, and at the iteration number 165 it comes back to the value of 0.01. Second plot examines the same distance around the last iteration. Also at the iteration 205th there is a small adjustment. Therefore we can state that the distance decreasing line is affected by many adjustment due to the Kalman gain increments.

In the third graph of figure 4.3, we can observe the same convergence iterations of the algorithm to a value of $\mathbf{\Phi}^*$ different from the actual value of $\mathbf{\Phi}$ given by the five dimensioned identity matrix. The first value is given by the difference between the unitary matrix and the initial transition, $\mathbf{F} = 0.95 * \mathbf{I}$. Because initial value of the transition matrix is nearest than the value of $\mathbf{\Phi}^*$ at the last iteration (208th in the example), the extracted hedonic prices are increasingly distant from the actual ones. This is a proof that many times EM algorithm provides sub optimal solutions when the number of variables of the system is higher than two, and for tiny input series (in this case 30 days). The sub optimal solutions are very similar to the actual ones. For instance, from figure 4.3 we see the values of $Zt_1^{(k,30)}$, $Zt_2^{(k,30)}$, and $Zt_3^{(k,30)}$. The rate given by:

$$30 \cdot Zt_j^{(k,30)} / \left(\sum_{t=1}^{30} \tilde{z}_{jt} \right),$$

is a mean distance between actual and extracted values. In the case of the simulation in the plots this rate starts from the 9% and arrives to the 15%. Since the likelihood distribution is multivariate normal we can motivate those errors on the ill conditioned problems in Kalman computations and on the misspecification of the model (transition matrix).

We pass to examine a case of convergence with large number of iterations (over 1000 steps). The prolongation of the algorithm is caused by the ill conditioned matrices. In figure 4.4 a simulation without convergence before 1000 iterations is showed. Plots say us that many times the filter approaches to the solution around a value of 0.0025 for the distance $\mathbf{n}_1$, but it never reaches out it. Anyway, the algorithm is stable and it may offer solutions very similar after a certain number of iteration around values of the absolute distance under 0.05. We can

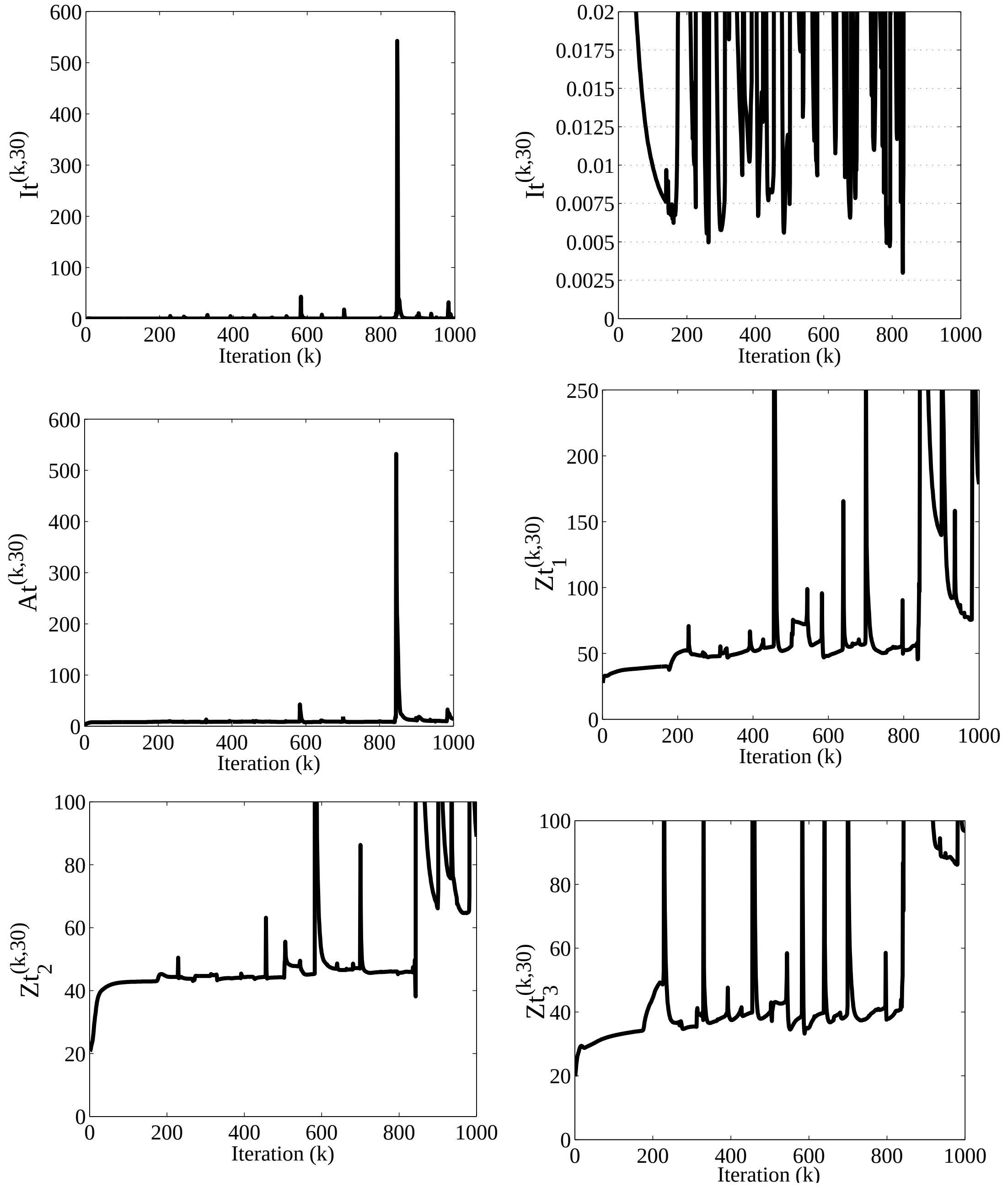

Figure 4.4: Indexes values in the case of non convergence of the algorithm. Seventh simulation for $\alpha = 1.0$ and with $T = 30$, m=5, n=16

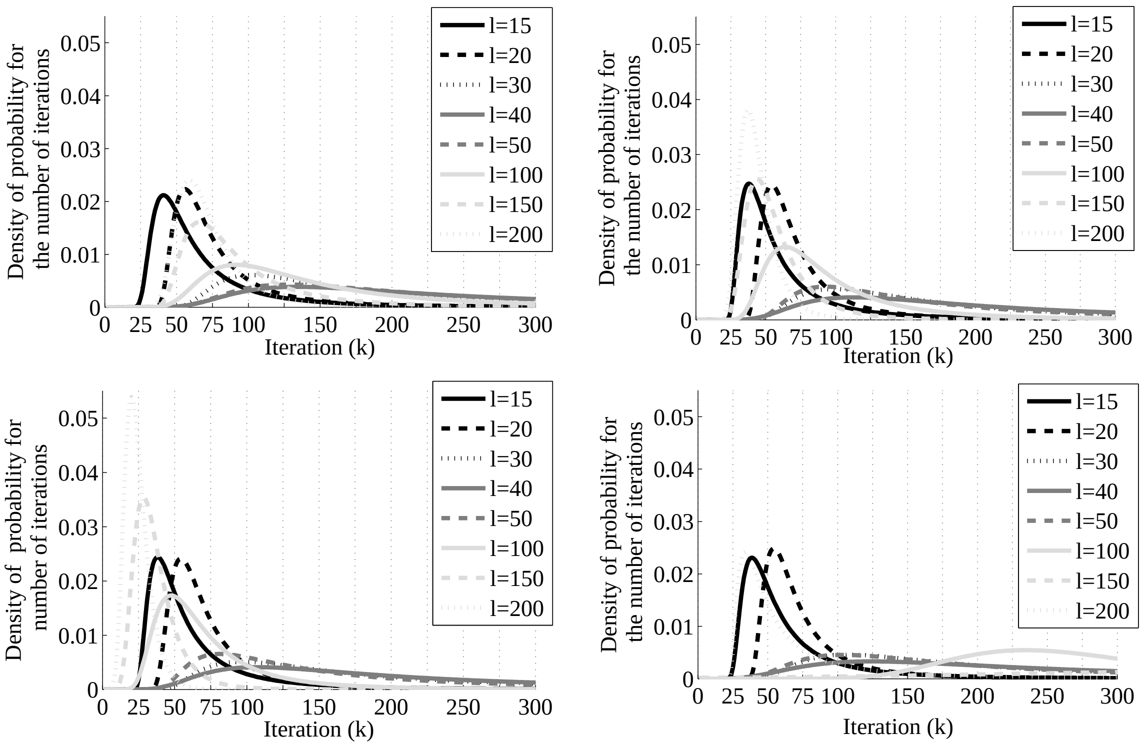

Figure 4.5: Empirical distributions for the number of iterations in 1000 simulations for different time lengths and for $\alpha = 0.8$ (left-up), $\alpha = 0.9$ (right-up), $\alpha = 1.0$ (left-bottom), $\alpha = 1.1$ (right-bottom)

deduce that EM+KF algorithm in high dimensions ($n = 16, m = 5$) usually gives a stable output with sub optimal properties, but sometimes it never reaches out the outperforming estimates of other stopping rules. The problem of no convergence may be solved with another stopping criteria for anomalous cases such as the likelihood ratio. Furthermore, the number of iterations required by the latter is definitely low being around 20-40 iterations or equivalently 20-30 seconds of calculations.

However, the goal of the study of the distribution of $\mathbf{n}_1$ is not to obtain a real time algorithm with time invariants parameters. We postpone the problem of selection of the best estimates when we will find the time varying parameters. For the moment, we point to establish if the negative results of criteria based on the distance between transition matrices affects forecast performances with respect to other criteria.

4.1.4 Discussion about Settings of the *Algorithm 1*

We now consider the empirical distribution of the number of iterations required by the first stopping rule depending on the stability parameter α of the transition matrix. For the set of simulations we are interested in the number of minimum and maximum iterations and the value of the distance depending on the length of time series of input and eigenvalue of transition matrix. For a perfect calibration we fit every distribution of number of iterations depending on the stability of the system (the parameter α), with a generalized extreme value distribution of the Frechet type. This family of distribution is used to model the largest (or smallest) value from a group of measurements. It is included in the more general family of distribution called *generalized extreme value (GEV) distribution* with a shape, a scale, and a location parameter (Kotz & Nadarajah, 2000). In our case, the character of study is discrete and positive. Distributions can be represented by a histogram which can be fitted by a continuous line. As alternative we can fit the empirical data using a Poisson distribution of parameter λ.

The probability density function for the generalized extreme value distribution with location parameter μ, scale parameter σ, and shape parameter $k \neq 0$ is:

$$f(x|k,\mu,\sigma) = \left(\frac{1}{\sigma}\right)\exp\left(-\left(1+k\frac{(x-\mu)}{\sigma}\right)^{-1/k}\right)\left(1+k\frac{(x-\mu)}{\sigma}\right)^{-1-1/k}, \tag{4.6}$$

for values of x such that:

$$1+k\frac{x-\mu}{\sigma} > 0,$$

and when the parameter $k > 0$, it corresponds to the Frechet type. Because the tail of those empirical distributions decrease as a polynomial it perfectly represents the situation of convergence iterations. In figure 4.5 the graphs for these fitted distributions refer to the four values of α analyzed in the testbed. For time series of length under the value of 20 periods the algorithm performs well and it arrives to a convergent suboptimal solution in 30–50 iterations. We know that it is motivated by the Kalman filter performance properties for short time series: in this case the algorithm rarely meets problem in the Kalman gain and high condition numbers for covariance matrix $\mathbf{P}_t$. When further observations are included to the time series, Kalman filter starts to give problems and it increases the probability of default. When the number of items in the input time series reaches the value of 100, the algorithm turn back to a good behavior and this time it often finds the actual solution of the problem in few iterations.

In table 4.1 are reported the number of times for failures, non convergence, and performances of $Zt_1^{(k,l)}$ of the algorithm until 1000 iterations in the four set $\alpha = 0.8, 0.9, 1.0, 1.1$. The conclusions of our analysis about the empirical distribution of the number of iterations in EM+KF algorithm in high dimensions are:

– a low number of iterations is sufficient to obtain sub optimal solutions because the algorithm falls into a "trap" after few iterations and change very slowly;

– for transition matrices with small eigenvalues the probability of degeneracy decreases and performances are better than the estimation in the case of transition matrices with large eigenvalues;

– algorithm provides only sub optimal solutions. Although they are very similar to the actual parameters and variables;

– algorithm performs well for an initial transition matrix below the actual one. In fact, for $\alpha = 1.0$ the probability of a nearest solution are higher than for $\alpha = 0.9$;

– when the transition matrix has high eigenvalues (out of the unitary circle), results are acceptable only for short time series, under 50 periods.

All those properties will be used to implement the real time algorithms in the next subsection.

4.2 Real Time Algorithms

In this section, the generative model results are used for the construction of an identification parameter procedure. According to the analysis of simulations, we will create two algorithms for real time applications based on the distance between transition matrices. The first algorithm requires more iterations but the results are more precise. The second algorithm is based on a fixed number of iterations dependent on the input data length. It is faster than the first algorithm of course. While the computation times decreasing a lot in the second algorithm, we prefer to discuss mainly the first one in the sequel of our work.

4.2.1 Two variants of the *Algorithm 1*

The optimal settings for a first variant of the *Algorithm 1*, which we named *Algorithm 1.A* are:

1. set a stopping rule based on the distance like in (2.24). It is possible to choose the type of the distance according to further test, but in our application we always used Manhattan distance;

2. set a not so small value of δ_1. For instance, an error of 0.0001 for each entries of the transition matrix must be sufficient. For the application in TAC SCM, it corresponds to set $\delta_1 = 0.0025$, because the transition matrix has 25 entries with 5 states;

Table 4.1: Divergence, convergence, and performance for Zt_1 in each α test for a value of $\delta_1 = 0.0025$

$\alpha = 0.8$								
	$l = 15$	$l = 20$	$l = 30$	$l = 40$	$l = 50$	$l = 100$	$l = 150$	$l = 200$
High Ncond	18	13	116	76	55	0	0	0
+1000 Iterations	2	5	199	315	273	32	1	0
Under 1%	43	5	0	0	0	0	0	0
(1% , 2%)	701	373	108	106	148	675	945	991
(2% , 3%)	216	523	398	383	443	292	53	9
(3% , 4%)	11	72	136	87	43	0	0	0
(4% , 125%)	9	9	43	33	38	1	1	0
$\alpha = 0.9$								
	$l = 15$	$l = 20$	$l = 30$	$l = 40$	$l = 50$	$l = 100$	$l = 150$	$l = 200$
High Ncond	9	16	140	140	80	0	0	0
+1000 Iterations	5	3	216	308	255	1	1	0
Under 1%	0	1	0	0	0	0	0	0
(1% , 2%)	329	128	40	72	112	713	970	995
(2% , 3%)	555	594	356	349	489	285	29	5
(3% , 4%)	95	250	205	110	47	0	0	0
(4% , 125%)	7	8	43	21	17	1	0	0
$\alpha = 1.0$								
	$l = 15$	$l = 20$	$l = 30$	$l = 40$	$l = 50$	$l = 100$	$l = 150$	$l = 200$
High Ncond	18	15	172	112	64	0	0	0
+1000 Iterations	2	3	222	299	194	1	0	0
Under 1%	45	4	0	0	0	0	0	0
(1% , 2%)	739	441	136	141	288	868	998	1000
(2% , 3%)	172	474	334	333	393	131	2	0
(3% , 4%)	8	57	67	48	13	0	0	0
(4% , 125%)	16	6	69	67	48	0	0	0
$\alpha = 1.1$								
	$l = 15$	$l = 20$	$l = 30$	$l = 40$	$l = 50$	$l = 100$	$l = 150$	$l = 200$
High Ncond	32	20	206	152	107	558	247	525
+1000 Iterations	5	11	243	305	279	24	660	475
Under 1%	1	0	0	0	0	0	0	0
(1% , 2%)	258	136	35	70	144	40	87	0
(2% , 3%)	579	581	281	356	421	20	3	0
(3% , 4%)	117	245	205	92	39	2	3	0
(4% , 125%)	8	7	30	25	10	356	0	0

3. set a minimum and a maximum number of iterations depending on the length of time series of input (see plots in figure 4.2 for this step). For instance, when $T = 15$, we set $i_{min} = 25$ and $i_{max} = 200$, and for $T = 40$, we set $i_{min} = 30$ and $i_{max} = 1000$;

4. it is necessary to follow the convergence of EM iterations checking the decreasing behavior of the distance. When the value of $\mathbf{n}_1$ is enough low but it increases respect to previous outer iteration store the previous parameter. In this way, we obtain a set of possible solutions. There may be three cases:

   (a) if the algorithm reaches the value of δ_1 after i_{min} and before i_{max} without large jumps (e.g. higher than 0.001) in the distance behavior, and the parameter passes the acceptance tests, then store the value for that iteration with minimum distance in the set of possible solutions, $\mathbf{\Psi}$ and go to the step (5);

   (b) if there are large jumps in the distance behavior store the values for that iteration with minimum distance which passes the acceptance tests in the set of possible solutions, $\mathbf{\Psi}$ and go to the next step reaches the i_{max} iterations;

   (c) if the algorithm does not reaches the value of δ_1 before the last iteration i_{max}, repeat the procedure with a higher value of δ_1 or take the solution with the minimum value of δ_1 which passes the acceptance tests;

5. for each possible solution we have already applied the acceptance tests for the consistency of the transition matrix (eigenvalues, sum of the entries, distribution of each entry). Then, we can choose the solution according to the minimum distance.

Several of the possible cases of *Algorithm 1.A* in the step (4) are represented in figure 4.8. The standard case is showed in the first plot: the absolute distance decreases and we accept the solution if it passes the acceptance test, otherwise we continue for the research. Second plot shows the case of a set of possible solutions: we select the third solution if it passes the acceptance test, otherwise we select one of the previous solution according to the minimum absolute distance and the acceptance tests. The third case refers to the divergence of the filter. If a solution does not passes the acceptance tests due on the ill-conditioned problem and the numerical errors, we select the solution with the minimum absolute distance reaches during the previous iterations.

The optimal settings for a second real time algorithm, *Algorithm 1.B* are:

1. to fix the maximum number of iterations according to the mean of the distributions obtained by (4.6) and taking into account of the results for the distance $\mathbf{n}_1$ (see the graphs in figure 4.2). For example, a number of iterations included in the interval $(25, 40)$ guarantees a result with an average value for $\mathbf{n}_1$ of 0.02-0.04;

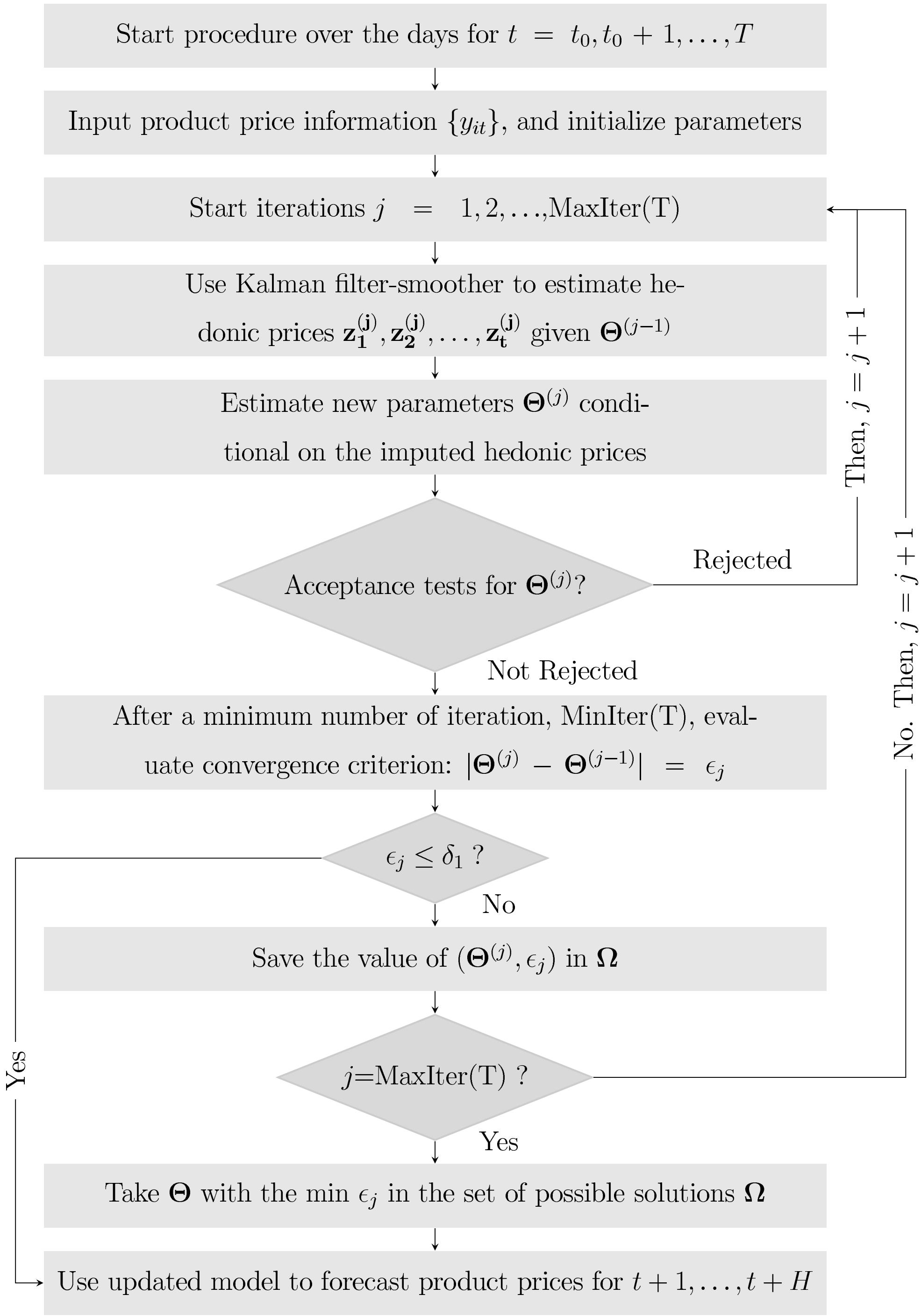

Figure 4.6: Expectation-maximization (EM) *Algorithm 1.A* used to estimate model parameters and impute implicit component prices for each time period t

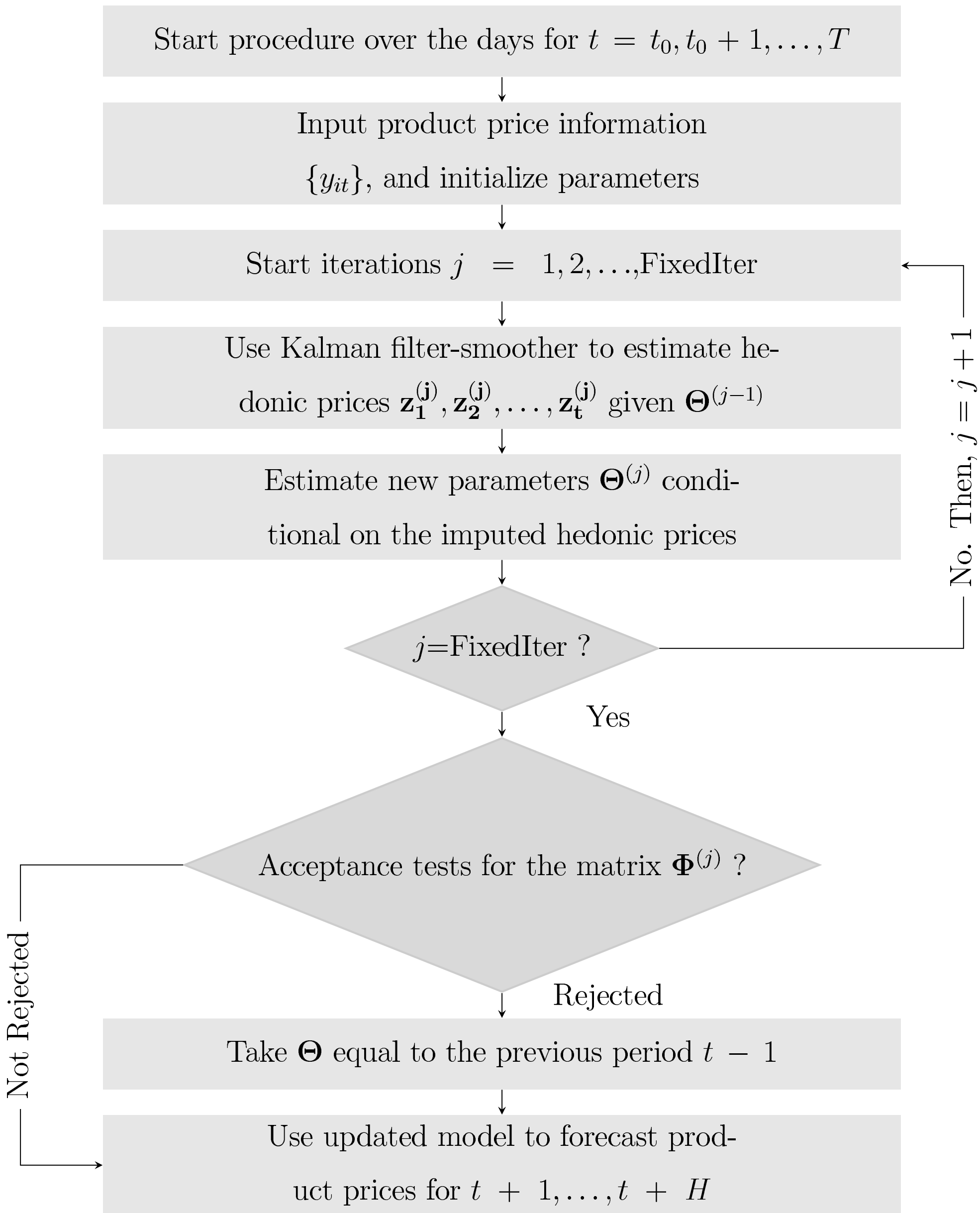

Figure 4.7: Expectation-maximization (EM) *Algorithm 1.B* used to estimate model parameters and impute implicit component prices for each time period t

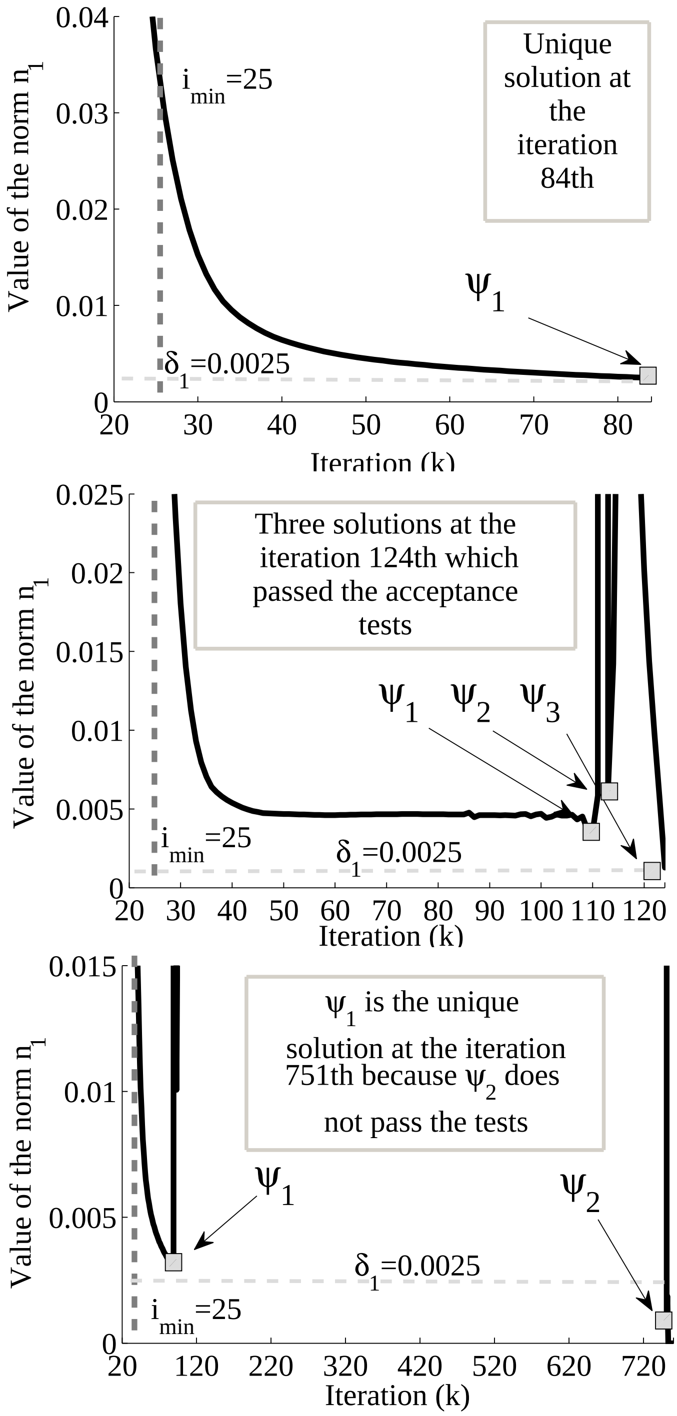

Figure 4.8: Three cases of selection of the optimal solution in the *Algorithm 1. A*

2. choose in the set of solutions the solution with the minimum value of δ_1 for the distance $\mathbf{n}_1$ which passes the acceptance tests.

3. if the transition matrix does not pass the acceptance test we opt for the first previous matrix which passes the test independent on the value of the absolute distance.

4.2.2 Tests for Verifying the Consistency of the Transition Matrix

Finally, we define some criteria to establish when a multi parameter, and in particular the transition matrix, is acceptable or not. We choose five criteria for the acceptance test:

- the maximum (dominant) eigenvalue of the matrix. It must be included in the interval $(\lambda_{min}, \lambda_{max})$;

- the maximum element of the transition matrix must be smaller than e_{max} and the minimum element of the transition matrix must be greater than e_{min};

- the sum of all the entries of the transition matrix must be smaller than S_{max} and greater than S_{min};

- the sum of the row elements must be smaller than s_{max} and greater than s_{min};

- each of the m diagonal entries must be included in the interval $(\phi_{m,min}, \phi_{m,max})$;

After the algorithm estimates the transition parameter it applies the acceptance test according to upper criteria. If the matrix fails the test the algorithm finds for another solution as described in the previous paragraph. The alternative solution must satisfy larger values for δ_1. At least the last solution to be apply is to use likelihood ratio test if the first stopping rule is not effective. But this option will never be used in none of the applications.

4.3 Conclusions

For standard models that satisfy weak regularity conditions, the maximization of the likelihood provides "good estimators" in terms of risk. When we apply Kalman filter mixed with EM procedure the convergence criteria based on the likelihood is not the preferable method for identifying of the parameters of the process, overall in multivariate case. We have considered a generative model for the extraction of a data set, in which parameters are known. We have applied the *Algorithm 1* to study its approach to the solution. How do we choose among the estimators of several convergence rules? In decision theory, the situation corresponds to have multiple actions. Analyzing the loss functions we have found the optimal calibration for the procedure and its average loss.

In the next chapter, the standard autoregressive models used to forecast prices will be explained. Then, an application of *Algorithm 1.A* and *Algorithm LR* is given to verify the optimality of the first methodology respect to the second one.

Chapter 5

Real Time Forecasting in Heterogeneous Supply Chain Markets

In this chapter we want to apply previous hedonic algorithms for real time forecasting in the specific environment of the dynamic heterogeneous supply chain markets. Precisely, we wish to improve standard methodologies in forecasting product prices with the hedonic information. The scope is twofold: first, to test the convergence criteria for the maximum likelihood parameters via the forecast results in a set of large simulations. In theory, the more the parameters are estimated with a good criteria the more forecast performances outperform other standard model. Second, to test the assumptions on the hedonic prices for improving forecast models. Standard models for forecasting product prices are usually based on the mean reversion effect. We can decompose the product price in m hedonic prices, and make the same assumption on each of them in a new prediction framework for components and products.

We will test the previous algorithms to extract hedonic prices from time series of product prices generated periodically (day by day) from the negotiations between manufacturers and customers. After an overview of the literature in forecasting models utilizing extra information in a supply chain management, the standard methodologies and performances indexes together with an application in TAC SCM will be compared to hedonic methodologies.

Autoregressive integrated moving average (ARIMA), exponential smoothing and spectral domain, are only a small part of the numerous models that econometric discipline offers (Hamilton, 1994; Box & Jenkins, 1976). They are based on the assumption that previous values are informative about the future ones. If we consider multiple time series of product prices, vector autoregressive models (VAR) include correlation between products. Normally, we can use those models to forecast prices based on their performances. In this case, we have multiple choices to select and the best way is to study their previous performance and assume that it will remain the same in the future periods. We select and test one combination of models observing on line performances. Output forecasts depend on the model used in that period and not only on the estimation of the parameters. There are many methods to use multiple forecasts (Bates & Granger, 1969; Diebold & Pauly, 1987). Today, with the growth of innovative models as regime switching, and threshold autoregressive, prediction techniques take advantages of multiple estimates. In our case, we propose a combination model which spans from univariate to multivariate estimates including component hedonic models. In this case, the determinants of the price of a good are assumed to be its components, which behaves independently in several regimes but correlated in other ones. We have seen in the previous chapter as state space model representation may help researcher to extract hedonic evaluations from time series. In Stadtler & Kilger (2008) there are several examples of applications in supply chain management of each of those techniques applied in such contexts. When we want to extract information about components, factors, or latent variables, we may apply those methodologies, but they are still poorly extended to dynamic analysis of real components (Harvey, 1989). On line applications of Kalman filter are numerous and in many fields. Instead in supply chain context we have only recent researches in Econometrics (Mazzocchi *et al.*, 2010), which test the estimation of parameters and state variables together.

Section 5.1 introduces the standard autoregressive models largely used in forecast analysis. Univariate and multivariate forecast models will be compared with a set of hedonic models to determine the properties of estimated values and parameters. If a hedonic algorithm exhibits good prediction results respect to other models we can state that the estimated parameters are better estimated than another algorithm. Obviously, we would repeat the experiment for a large number of applications for a high confidence in that statement.

In section 5.2 the hedonic models included in the forecast analysis will be described. Besides listing the previous model formulations (see table tbl:AllModels), we also consider a mix of standard forecast and pure hedonic techniques. In sub section 5.2.1 multiple autoregressive hedonic models (MAHR) are explained. In fact, after the extraction of hedonic information via one of the algorithm in table 3.4, we can use it as additional series in a standard univariate autoregressive model. The latter becomes a bivariate model with extra information. The analysis of forecast performances of MAHR models consents to verify the behavior of single variable in the vector of hedonic prices. In that sense, if algorithm estimates a hedonic price series with a minor precision than another the correspondent root mean squared error in a testbed of many applications shows higher values. Finally, in sub section 5.2.2 an on line combination model is defined. The latter is based on a set of on line forecast models. The combination model forecasts product prices through a regression of daily predictions coming from standard and hedonic models. It may be considered the final model to be included in a future forecast module of an agent manufacturer in a heterogeneous supply chain.

An application in TAC SCM of our forecast framework considering parts and products is given in section 5.3. Across a collection of product price time series for 50 game, we study results about performance indexes. The scope of the application is to give a comparison amongst various methodologies which can be use together in an expert system for analysis of time series in supply chain computer markets. Obviously, there are many other possible applications of the dynamic hedonic methodology, especially in complex systems where product variety includes many products and parts.

5.1 Standard Autoregressive Models

We want to introduce the conventional autoregressive models largely used in forecast analysis. The scope is twofold: to compare the performance of hedonic models with robust models, and to create a framework for forecasting in combination with hedonic modeling. Many specific models appeared in the last decades for our environment, the dynamic supply chain markets where components are assembled in end-products. For instance, the regime analysis in (Ketter *et al.*, 2009) achieves to extract the important information about product availability and productivity. But univariate and multivariate forecast models are optimal references to compare the prediction performances of estimated values and parameters in other models. Furthermore, if our hedonic algorithm exhibits good prediction results respect to conventional models we can state that the estimated parameters are better estimated than another algorithm. Obviously, we would repeat the experiment for a large number of applications for a high confidence in that statement.

5.1.1 Forecast Models Based on Single Series of Product Prices

Preliminary analysis for the application in the second chapter has shown the lack of covariance-stationarity in the product price series. For instance, an analysis of figure 2.5 leads to specify a trend for the series. In non stationarity cases, the approach to forecasting advocated by Box and Jenkins (Box & Jenkins, 1976) can not be applied without a transformation of data. Usual transformations are the detrending and differencing.

The first choice for a test model is a univariate autoregressive model, AR(p), with p lag-parameter dependent on the number of periods shows high partial autocorrelation[1]. Since

[1] This choice is based on the explorative analysis of partial autocorrelation functions (PACF). For example, if every product has a PACF graph shows high values (0.3-0.5) for the first three lags we can opt for $p = 3$. Also the shape and the length of the bars in the graph can show the correct modeling

our data usually does not display long memory property we omitted a long moving average component, typical of moving averages models (MA). Autoregressive model is based on the following relation:

$$y_{i,t} = \beta_{0,i} + \beta_{1,i} y_{i,t-1} + \beta_{2,i} y_{i,t-2} + \cdots + \beta_{p,i} y_{i,t-p} + \epsilon_{i,t} \tag{5.1}$$

for $t = 1, \ldots, T$, where T is the last value known in the series and $i = 1, \ldots, n$ is the index for each type of product. We assume $\epsilon_{i,t} \sim NID(0, {\sigma_{\epsilon,i}}^2)$, or $E[\epsilon_{i,t}] = 0$ and $E[{\epsilon_{i,t}}^2] = {\sigma_{\epsilon,i}}^2$ for each $i = 1, \ldots, n$. The estimation method is the ordinary least squares (OLS) technique without restriction. Output to validate the model includes some statistics like t-value and significance level, the residual squared sum (RSS) and the log-likelihood[2]. To simulate the agent on line application of the model, we may compute forecast performances for many windows and for each of them we will measure performances. Forecasts of AR(p) are computed in dynamic way, using the last values known at the time t to estimate the future h values by the relation:

$$\hat{y}_{i,t+h} = \hat{\beta}_{0,i} \sum_{l=1}^{h} \hat{\beta}_{1j,i}^{(l-1)} + \hat{\beta}_{11,i}^{(h)} y_{i,t} + \cdots + \hat{\beta}_{1p,i}^{(h)} y_{i,t-p}, \quad h = 1, \ldots, H, \tag{5.2}$$

where $\hat{\beta}_{1j,i}^{(h)}$ denote the $(1, j)$ element of $\mathbf{B}_i^h$, the h-th power of the following $(p \times p)$ matrix:

$$\mathbf{B}_i \equiv \begin{bmatrix} \hat{\beta}_{11,i} & \hat{\beta}_{12,i} & \ldots & \hat{\beta}_{1p-1,i} & \hat{\beta}_{1p,i} \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix}. \tag{5.3}$$

For $h = 0$, values of $\hat{\beta}_{1j,i}^{(0)} = 1$ for all $j = 1, \ldots, p$, and $i = 1, \ldots, n$. Since an agent must predict short-medium future behavior of prices, a value of $H = 40$ is convenient. In fact, our agent's goal is to update the model day after day receiving information about product prices for customers, and use dynamic information for future investment in the procurement market for production planning. A value of forty days must be ideal to test also medium-long strategy for an agent.

[2]The assumption about disturbance distribution is not relevant in our work. Anyway, we tested the zero mean hypothesis in every model via the standard t-test for normality parameters and non parametric median tests for general parameters

5.1.2 Forecast models based on multiple series

Vector autoregressive models (VAR) take into account the co-movements among a set of variables. The covariation of time series that behave in the same way is a source of information that can improve forecast precision in many cases (du Preez & Witt, 2003). Each variable is regressed against its own value and all the other variables in the model for p periods back, VAR(p). The technique for the estimation of the coefficients of the system is the stepwise least squares, computationally efficient in particular when the time series data $\mathbf{y}_t$ are high–dimensional (Schneider & Neumaier, 2001).

To determine the number of lags into the past to be used usually it is need a canonical correlation analysis. The choice is between the Aikake's information criterion (AIC), final prediction error (FPE) criterion, and Schwarz's Bayesian Criterion (SBC) (Lutkepohl, 2005). Lutkepohl (2005) compared those criteria and found that SBC is the best methodology for the smallest mean-squared prediction error. In our methodology, for each p-model, where $p_{\min} \leq p \leq p_{\max}$, the residual covariance matrix and SBC are calculated. Then, the optimal order is determined. Differently from the previous model, VAR requires a lot of parameters[3] to estimate the coefficients and this is the greater disadvantage. Because our time series are increasing, from t_0 to T, we find that the best order is given by $p = 1$ when time series data are shorter than one hundred values. Hence, many times we will use the simplest VAR(1) to measure performance of multivariate models for product price series.

Our unrestricted reduced form of the system is:

$$\mathbf{y}_t = \mathbf{\Pi_1}\mathbf{y}_{t-1} + \cdots + \mathbf{\Pi_p}\mathbf{y}_{t-p} + \mathbf{u}_t, \quad \text{for } t = 1, \ldots, T, \tag{5.4}$$

where every $\mathbf{\Pi}_i$ is the $n \times n$ matrix of coefficients constant over time and $\mathbf{u}_t \sim MVN(\mathbf{0}, \mathbf{\Sigma_u})$. Note that $\mathbf{\Sigma_u}$ is constant over time and it means that OLS estimation coincides with maximum likelihood estimation (MLE). To facilitate the maximization of the likelihood of (5.4) we used the OLS despite of MLE, using the MATLAB package described in Schneider & Neumaier (2001). In fact, even without the stationarity restriction, fitting a VAR(1) model by maximization of likelihood (MLE) is computationally demanding (Lutkepohl, 2005). A typical output of VAR regression provides the estimates of the coefficients of $\mathbf{\Pi}_i$ and their standard errors. A t-value and a p-value tells us whether individual coefficients are significantly different from zero (null hypothesis). The square root of the residual variance, the sum of diagonal entries of $\mathbf{\Sigma_u}$, can be used to measure how the model fit the multivariate

[3]For instance, a VAR(1) of 16 equations requires 256 parameters as a VAR(3) for the same number of variables requires 768 of them. This is the reason why Akaike information is very important in multivariate models.

series. Although, to validate the model we used the coefficient of determination R^2. It represents the proportion of variation in the dependent variable that has been explained or accounted for by the regression model and can be used to measure how the model fits the multivariate series. Values of $R^2 < 0.25$, which corresponds to an $R < 0.5$, would never be acceptable.

Unfortunately, VAR models are not always adequate description of real series (Juselius & Hendry, 2000). In our case we are assuming that every product price depend on the other ones but it may be not the case. Why an agent should be consider an increasing of price for a product if the price of another one increases or decreases? Furthermore, there is the risk that multivariate assumption is not satisfied for co-integration. This problem often invalidates the model assumption and it forces the researcher to find a more adequate formulation of the interrelationship among the variables. Obviously, all these defects affect the forecast performances and in several cases multivariate models behave worse than univariate models. Finally, we arrived to understand the importance of DHMM model as alternative to VAR model: if we use DHMM we avoid a co-integration analysis of product series, that is may be very troublesome in some cases.

Following the methodology in univariate case, we analyze VAR performances in non-overlapping increasing estimation windows. Here, we have to wait for a certain number of periods before method could be applied. For all the estimation windows we compute the ahead predictions for the next $H = 40$ days via the iterated relations:

$$\begin{aligned}
\hat{\mathbf{y}}_{t+1|t} &= \hat{\mathbf{\Pi}}_{\mathbf{1}}\mathbf{y}_t + \cdots + \hat{\mathbf{\Pi}}_{\mathbf{p}}\mathbf{y}_{t-p} \\
\hat{\mathbf{y}}_{t+2|t} &= \hat{\mathbf{\Pi}}_{\mathbf{1}}\hat{\mathbf{y}}_{t+1|t} + \cdots + \hat{\mathbf{\Pi}}_{\mathbf{p}}\mathbf{y}_{t+1-p} \\
&\dots \\
\hat{\mathbf{y}}_{t+H|t} &= \hat{\mathbf{\Pi}}_{\mathbf{1}}\hat{\mathbf{y}}_{t+H|t} + \cdots + \hat{\mathbf{\Pi}}_{\mathbf{p}}\hat{\mathbf{y}}_{t+H-p}, \\
\text{for } t = 1, \dots, T \quad &, \quad h = 1, \dots, H, \text{ and } H > p,
\end{aligned} \tag{5.5}$$

where $\hat{\mathbf{\Pi}}_{\mathbf{i}}$ is the estimated matrix for the i-th lag.

5.1.3 Forecast performance indexes

We have argued in the first chapter that forecasting is one of the main objectives of our research. The importance of forecast analysis in extraction of information is twofold. Point forecasts and interval forecasts will be considered in a large amount of applications to validate the estimation of our algorithm. Thus, forecasting methods become a real testbed for the hedonic algorithm. In such case we are not so much interested in finding the correct parameters of the underlying hedonic process but we want to obtain a good model for prediction.

To validate forecasts we may use several indexes according to different priorities. We point to an initial performance index and a measure of forecast precision. This is due to the twofold nature of our forecast analysis. Approximated estimated parameters can give low performances in initial periods and differently the best performances many periods ahead. In fact, if the system accepts structural changes a parameter not adapt today can become optimal in the future periods. Hence, an index for first day performances is good for an evaluation of the estimation performances of hedonic algorithm. Differently, if forecast precision is minimized via root mean squared errors

We define the following indexes:

– the one-day-ahead relative absolute error (ODAE), the relative error of the model returned the next day when we found the actual price. It is good that ODAE do not exceed a fixed value selected by the agent otherwise it means that our model fails. When the real time updating of the data goes day by day all the other forward performances are not so important as ODAE. To allow comparisons between different products we normalize them using the nominal product price. We define the index such that:

$$\mathrm{ODAE}(i,t) = \frac{|y_{i,t+1} - \hat{y}_{i,t+1}|}{np_i}, \tag{5.6}$$

for $i = 1, \ldots, n$ and $t = s, \ldots, T,$. Here the values np_i are the product nominal prices obtained by a sum of nominal component costs and assembly cost as in:

$$np_i = AssCost_i + \sum_{i=1}^{numParts} NomPartCost_{i,j}, \tag{5.7}$$

where $NomPartCost_{i,j}$ is the nominal cost of the j-th part for good i, $numParts$ is the number of parts needed to make the good i, and $AssCost_i$ is the cost of manufacturing the good i. A nominal component cost is defined as the reference price for an individual component known from each agent at the beginning of the game. They are necessary because in this way, we can compare performances for different products in the supply chain. Reasonable values for ODAE in many applications will depend on the largeness of the estimation window of the model. The longer is the series of prices the effective is the performances of the model;

– the one-day-ahead relative positive error (ODAE$^+$) returned the next day when we found the actual price given by:

$$\mathrm{ODAE}^{+}(i,t) = \frac{(y_{i,t+1} - \hat{y}_{i,t+1})^{+}}{np_i}, \tag{5.8}$$

where $(x)^{+} = max(0, +x)$, which indicates when the forecast values are under the actual values. At the same way, the one-day-ahead relative positive error, ODAE^{-} si defined as:

$$\mathrm{ODAE}^{-}(i,t) = \frac{(y_{i,t+1} - \hat{y}_{i,t+1})^{-}}{np_i}, \tag{5.9}$$

where $(x)^{-} = max(0, -x)$

- the root squared (prediction) relative error computed for the h period using an estimation window of t periods, respect on the prediction values $\hat{y}_{i,t+h}$ provided by one of the eight model estimated:

$$\mathrm{MSE}(i,t,h) = \left(\frac{y_{i,t+h} - \hat{y}_{i,t+h}}{np_i}\right)^2, \tag{5.10}$$

  for $i = 1, \ldots, n$, $t = s, \ldots, T$, and $h = 1, \ldots, H$, and $H = 40$. The MSE gives an idea of performance after h days ahead, where h spans from one to forty days ahead;

- to determine the accuracy of the model in more experiments, we can average across N_G simulations the RMSE, such that:

$$\mathrm{RMSE}(h) = \sqrt{\frac{\sum_{g=1}^{N_G}\sum_{t=s}^{T-h}\sum_{i=1}^{n}\left(\frac{y_{i,t+h}^{(g)} - \hat{y}_{i,t+h}^{(g)}}{np_i}\right)^2}{n \cdot (T-s-h) \cdot N_G}}, \tag{5.11}$$

  where g is the index for the simulation, and N_G is the total number of examined simulations. Similarly to the previous paragraph, np_i represents the nominal price of the product with index i $(i = 1, \ldots, n)$;

- another standard measure of forecast precision is given by the mean absolute percent error (MAPE) over all games, periods and products:

$$\mathrm{MAPE}(h) = \frac{\sum_{g=1}^{N_G}\sum_{t=s}^{T-h}\sum_{i=1}^{n}\left(\frac{\left|y_{i,t+h}^{(g)} - \hat{y}_{i,t+h}^{(g)}\right|}{y_{i,t+h}^{(g)}}\right)}{n \cdot (T-s-h) \cdot N_G}. \tag{5.12}$$

  It is dependent on the ahead period h, from one to H. MAPE is a percent measure and it is comparable amongst the models.

The RMSE is an integral component in statistical models. It is used in many forecast errors evaluations. The advantage of that measure is that its scale is the same as the forecast data. Thus, errors reported by the root mean square error are representative of the size of an average error. The presence of outliers should affect the values of MSE and RMSE. An useful correction may be the use of the median instead of the mean to evaluate a median absolute error. But in this case, the measure does not maximize the available information on the errors. It is more robust than RMSE but not efficient. In our formulation, the MSE and RMSE consider normalized errors given by:

$$norm\ err(i,t+h|y_1,\ldots,y_t)^{(g)} = \frac{|y_{i,t+h}^{(g)} - \hat{y}_{i,t+h}^{(g)}|}{np_i}, \tag{5.13}$$

whereas, MAPE considers percentage errors defined as:

$$perc\ err(i,t+h|y_1,\ldots,y_t)^{(g)} = \frac{|y_{i,t+h}^{(g)} - \hat{y}_{i,t+h}^{(g)}|}{y_{i,t+h}^{(g)}}. \tag{5.14}$$

The last measure, MAPE, is not scale dependent adjust for the product price using a percentage error given by the rate between prediction error and observed price. Many others indexes can be built and used for measure the forecast precision. We want only to mention the symmetric mean absolute percent error (sMAPE), which considers also the forecast values in the denominator of (5.12). It usually runs from zero to one hundred, while the MAPE is not limited.

When making forecasts, one would of course like to measure the uncertainty of the predictions. To this end, we have introduced the RMSE, the standard deviation of the forecast errors. Unfortunately, normality tests in the first application showed a reluctance to accepting the Gaussian assumption. In this way, we are forced to use empirical methodologies for the computation of $(1-\alpha)\%$ prediction intervals. The use of a set of agent-based supply chain simulations is very useful for the computation of the variability in predictions. In our application we can avoid the use of the Gaussian assumption, and we will compute forecast limits by simulation.

5.2 Autoregressive models including hedonic values

Now, we describe a set of multiple models that takes into account the hedonic prices for the individual components one at a time. First of all, we consider the hedonic multivariate model (DHMM), as in the previous chapter, such that:

$$\begin{aligned} \mathbf{z}_t &= \mathbf{\Phi}\mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t \\ \mathbf{y}_t &= \mathbf{D}\mathbf{z}_t + \boldsymbol{\nu}_t \\ \boldsymbol{\nu}_t &\sim MVN(0, \tilde{\mathbf{\Sigma}}_{\boldsymbol{\nu}}) \\ \boldsymbol{\varepsilon}_t &\sim MVN(0, \tilde{\mathbf{\Sigma}}_{\boldsymbol{\varepsilon}}) \\ \mathbf{z}_0 &\sim MVN(\tilde{\boldsymbol{\mu}}_0, \tilde{\mathbf{\Sigma}}_0) \end{aligned} \tag{5.15}$$

where every matrix is assumed non stochastic at time t, the white noise processes $\boldsymbol{\varepsilon}$ and $\boldsymbol{\nu}$ are independent. Also the initial state $\mathbf{z}_0$ is independent of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\nu}$.

Specifically, each of the two algorithms set out to estimate the unknown model parameters $\mathbf{\Theta} = \{\mathbf{\Phi}, \mathbf{\Sigma}_{\boldsymbol{\nu}}, \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}, \mu_0, \mathbf{\Sigma}_0\}$ together with the implicit component prices $\mathbf{z}_1, \ldots, \mathbf{z}_t$ in order to obtain product price forecasts of h periods ahead $\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+H}$, where H is the forecast horizon. For each day t, after some start up period of $t_0 - 1$ days, the procedure begins with identifying the available product price information up to and including period t and initializing the parameters to be estimated. It then starts a loop to maximize the joint likelihood of the model parameters and the state variables using an expectation-maximization (EM) approach. This EM-approach consists of two steps: a Kalman-filter operation and a maximum likelihood optimization. The Kalman-filter operation imputes values for the latent implicit component prices over time, $\mathbf{z}_1^{(j)}, \mathbf{z}_2^{(j)}, \ldots, \mathbf{z}_t^{(j)}$, for given estimates of the model parameters $\mathbf{\Theta}^{(j-1)}$. The subsequent maximization step determines new estimates for the model parameters $\mathbf{\Theta}^{(j)}$ given the imputed implicit component prices using maximum likelihood. The sequence of finding implicit component prices and estimating model parameters is repeated until some stopping criterion is met.

In the *Algorithm 1.A*, see figure 4.6, a stopping criterion based on the incremental change in the estimated parameters is suggested. In the *Algorithm 1.B*, see figure 4.7, a stopping criterion based on the number of iterations is suggested. In the latter case, we will explore the consequences for the forecast performance of putting a cap on the number of iterations regardless convergence. Upon convergence or stopping, the available estimates and implicit prices are used to forecast future product prices and implicit component prices. We use the output values of each algorithm to compare the gains of *Algorithm 1.A* with respect to *Algorithm 1.B*. In spite of its intuitive logic, we have already seen that the proposed algorithms have serious drawbacks when applied in dynamic real-time contexts. For instance, in the simulations of TAC SCM, they reported convergence times of one to over fifteen minutes, and convergence failures in several cases. As a consequence, the timely availability of product and component price forecasts is seriously challenged at arbitrary instances. For this reason we opt for different methodology to assign a value to the unknown parameters.

We advanced three options:

– the first is based on the likelihood ratio test to discriminate the convergence of the Kalman filter. We use the *Algorithm LR* with the following stopping rule:

$$\mathbf{n}^{(2)}\left(\mathbf{\Theta}^{(j)},\mathbf{\Theta}^{(j-1)}\right) = \\ -2\left\{\ln L_Y(\mathbf{\Theta}^{(j)}) - \min\left\{\ln L_Y(\mathbf{\Theta}^{(1)}),\cdots,\ln L_Y(\mathbf{\Theta}^{(j-1)})\right\}\right\} < \delta_2. \tag{5.16}$$

Note the change in the specification of the stopping rule. This time the likelihood value for an outer iteration of KF+EM is compared not only to the previous value, but it is compared with the minimum value reached. In that way, the presence of a jump does not break the algorithm research. Then, the value of δ_2 is used to the calculate the value of $p = 1 - \chi^2_{\delta_2,1}$. If the value of p is greater than 0.95 we accept the hypothesis of identical parameters in two close iterations.

– second, an extreme simplification of the model as we introduced in section 3.1, in the system (3.1). Obvious changes are the restriction of the transition matrix $\mathbf{\Phi}$ to an identity matrix implying that component prices reveal random walk behavior, and the assumption of diagonal covariance matrices for the noises. The design plus noise model is defined by:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{D}\mathbf{z}_t + \sigma_v\mathbf{I}, \\ \mathbf{z}_t &= \mathbf{z}_{t-1} + \sigma_u\mathbf{I}, \end{aligned} \tag{5.17}$$

where $t = 1,\ldots,T$, and σ_v, σ_u estimated by historical data;

– the third option is the estimation of diagonal entries of the transition matrix as in (3.3). We estimated the matrix $\mathbf{F}$ via *Algorithm 1.A* output for a set of historical data (the thirty training games). The technique used depend on the specific application. For instance, assuming a decreasing behavior for the implicit prices, the agent may assume a matrix $\mathbf{F}$ with all elements 0.99. Obviously, diversifications of $\mathbf{F}$ for component type is also possible.

We recall that in the noise model, the transition matrix is equal to $\mathbf{I}$, whereas in the diagonal one, it is equal to $\mathbf{F}$. Both models speed up the calculations, but their impact on estimation and forecasting performance is unknown. In fact, to limit the time spent on the Kalman-filter operations, we use a simple prediction algorithm for the determination of implicit component prices. If the expectation step is limited to predicting, like in the *Algorithm 2* and in the *Algorithm 3*, then this would save time considerably. Furthermore, in those we want to substitute historical parameters to the estimation procedure, not only for avoiding divergence problems, but also for testing stability hypothesis for the hedonic evaluations. The replacement of the transition matrix with an identity matrix means that each component evaluation is not dependent on the other. Value around unit are typical of stable systems.

Summarizing, we have five hedonic price vectors, extracted via five algorithms, *Algorithm 1.A*, *Algorithm 1.B*, *Algorithm LR*, *Algorithm 2*, and *Algorithm 3*.

Last operation of each of the algorithms is to forecast future prices via the relations:

$$\hat{\mathbf{y}}_{t+h} = \mathbf{D}\hat{\mathbf{z}}_{t+h}, \quad \text{and} \quad \hat{\mathbf{z}}_{t+h} = \hat{\Phi}^h \hat{\mathbf{z}}_t, \quad h = 1, \dots, H, \tag{5.18}$$

where both $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{z}}_t$ are provided by the algorithm, and H is a value previously chosen.

In the next subsection we will describe another way to use the hedonic prices for the accomplishment of a forecast framework.

5.2.1 Multiple autoregressive hedonic models (MAHR)

Instead of considering a multivariate model with all the components together, we can opt for simplest bivariate models which each include a hedonic variable. We opt for a univariate model to which we will include hedonic information. The interpretation of this model is given by the following assumption: in certain periods a component affects the prices more than expected value. In DHMM product prices changes are only induced by hedonic vector of evaluations. We weak this assumption assuming that there exists a link with historical prices. The advantages of our new model are:

- it considers the co-dependencies between product price and the hedonic evaluation of a component included in one of the product;
- it simplifies the multivariate DHMM reducing the number of variables;
- it avoids the assumption about false interrelations amongst the product prices as in multivariate case.

As in the case of DHMM, our model is more attractive if and only if the hedonic values are estimated via an optimal design matrix, and in on line version it may suffer of short time of computations. We assume that our prices depend on past values but also hedonic values such that:

$$y_{i,t}^{(j)} = \beta_{1,i}^{(j)} y_{i,t-1} + \cdots + \beta_{p,i}^{(j)} y_{i,t-p} + \alpha_i^{(j)} \hat{z}_{jt} + \epsilon_{i,t}^{(j)}, \tag{5.19}$$

for $t = 1, \dots, T$ and $j = 1, \dots, m,$. Here, $(\hat{z}_{1t}, \dots, \hat{z}_{mt})^t = \hat{\mathbf{z}}_{\mathbf{t}}$ are the estimates of hedonic prices for components defined in (5.15). Differently from simple autoregressive model, the model in (5.19) includes the hedonics evaluations for components assembled into the product, which takes the place of the constant term of intercept regression. For all this reason, we call it multiple autoregressive model, MAR(p). Theoretically, via the equations given in (5.19) we state there is a link between the prices and the hedonic evaluations of characteristics of products established in 5.15. One time, an agent estimates the $\hat{\mathbf{z}}_{\mathbf{t}}$ vector value he can plug it into one of the m forecast model to improve performances. There are m (one for each component) MAR(p) models, that we indicate by MAR(p)$_j$, with $j = 1, \dots, m$. In this way we can have a multiple estimation of the product value considering a different components history and hence, the complete picture of future developments. One time estimated the coefficients in (5.19), we can use the following relation given by:

$$\hat{y}^{(j)}_{i,t+h} = \hat{\alpha_i}^{(j)} \left(\sum_{l=1}^{h} \hat{\beta}^{(j,l-1)}_{1l,i} \hat{z}_{j,(t+h+1-l)} \right) + \hat{\beta}^{(j,h)}_{11,i} + y_{i,t} + \cdots + \hat{\beta}^{(j,h)}_{1p,i} y_{i,t-p}, \tag{5.20}$$

where $\hat{\beta}^{(j,h)}_{1l,i}$ denote the $(1, l)$ element of $\mathbf{B}^h_{i,j}$, the h-th power of the matrix:

$$\mathbf{B}_{i,j} \equiv \begin{bmatrix} \hat{\beta}^{(j)}_{11,i} & \hat{\beta}^{(j)}_{12,i} & \dots & \hat{\beta}^{(j)}_{1p-1,i} & \hat{\beta}^{(j)}_{1p,i} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}. \tag{5.21}$$

Values of hedonic prices and parameters come from the output of the *Algorithm 1.A.*

Obviously, we must choose a criteria to select the most reliable between the univariate model (AR), the m bivariate models, the VAR model, and the DHMM model to predict the actual price of one product. In Figure 5.1, we represented the space of forecast models limited by basic standard models where our hedonic models are positioned. In that space, many researchers have already tested other models including latent, factors, and principal component models (Stock & Watson, 2002; De Sarbo *et al.*, 1987). In this work, we explore the advantages of a combination technique using all previous models, with the idea to use all the determinants of the price of the product linked to one, two, or all components, to forecast future prices.

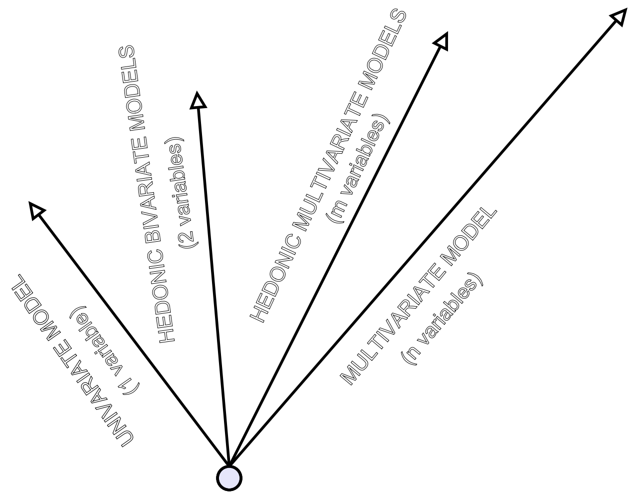

Figure 5.1: Our hedonic models spans from univariate and multivariate autoregressive models to test different hypothesis of co-dependencies.

5.2.2 Forecast combination model

We have shown a set of forecast methodologies. How an agent can weight the multiple forecast information to point in the right direction to predict prices? When a researcher tests a model in univariate or multivariate context is easy to obtain results which show a misspecified modeling because basic assumptions can vary during the life of the process, or the shape of the noise is not so regular. Then, he can opt for a combination model, a technique for increasing the performances of multiple models (Diebold & Pauly, 1987). Instead to determine the best forecast model we want to build a framework including the standard and hedonic techniques together.

In which way must we combine the set of forecasts? A weight can be assigned to every prediction value. There are many ways to find a vector of weights. In Bates & Granger (1969) the authors show five different ways to set the value of weights. They outline a methodology for unbiased forecast based on the variance of errors in the combined forecast. They restricted the analysis to two sets of forecasts, and their future works are directed towards operational ways to extend it for more than two forecasts.

In our approach, we follow a new regression–based methodology, for biased and unbiased sets of forecasts. We obtain a real time estimate of the weights at time t based on the forecast values in (5.2), (5.5), and (5.20) regressed on the actual values of the previous days. For each ahead period $h = 1, \ldots, H$, a linear regression model updates day by day the weights. After $t_0 + s$ days the forecast models start to provide sufficient predictors for the $k = 1, \ldots, H$ ahead periods, and the combination model can calculate the bs coefficients for each period h. In fact, we have to wait for s days over the initial t_0 for an output of combination model.

Although, longer time means more observations in the combination forecast regression. For instance, in our application we opt for a value of $s = 7$. For this reason, we fill the lack of predictors in the first days with the forecasts of AR model. But it causes a strange tail in the RMSE plot of the combination model, which near the RMSE of AR model in the last forecast ahead periods.

Our framework includes multiple indicators of market trend that could substitute the standard univariate and multivariate autoregressive estimators. The former in non stationary markets suffers from the defect of the reversion to the mean, and the latter is often too much complicated to accomplish the optimal performance. We test a model with a combination of predictions obtained by previous ones. It is a real-time model estimating the weights day by day during the standard game. We define $NumMod$ the number of models which are included in the combination model. They are the AR, the VAR, and the hedonic MAHR_j models. The choice to eliminate the hedonic forecasts coming from (5.15) is made to avoid the problem of initial period failure of DHMM, and to consider only non overlapping methodologies from the point of view of the variables. In fact, hedonic variables are all included in the single MAR_j models, $j = 1, \dots, m$.

In the on line version, we calculate every day the weights b of our new model using the previous forecast results of each of the $NumMod$ models as independent variables and the actual previous prices as dependent variable. The inclusion of the VAR model was made after the necessary period of time for the estimation of the multivariate model. In this way, $NumMod = m + 2$. To estimate the coefficients (weights) at the generic day $t = t_0 + s + d$, where $d = 0, \dots, T - t_0 - s$, we have available s forecast predictions of each l model. Let

$$\hat{y}^{(l)}_{i,u,h}\,, \text{ for } \quad u = t-1, t-2, \dots t_0, \quad h = 1, \dots, H, \tag{5.22}$$

the forecast calculated at the day $u < t$, for the ahead period h, through the autoregressive model l, for the product i. The set of forecasts are sufficient to obtain at t the weights through the h linear regression models:

$$y_{i,t-k+1} = b_{0,i,h} + \sum_{l=1}^{NumMod} b_{l,i,h}\hat{y}^{(l)}_{i,t-k,h}, \tag{5.23}$$

for $k = 1, \dots, t - t_0$. Here the response variable is the vector of product prices. The estimates are the $i \times h$ vectors of weights. Substitute the estimates in:

$$\hat{y}^{(combi)}_{i,t+h} = \hat{b}_{0,i,h} + \sum_{l=1}^{NumMod} \hat{b}_{l,i,h}\hat{y}^{(l)}_{i,t,h}, \tag{5.24}$$

and we obtain the forecast product prices according to a machine learning algorithm. Note, we have to wait until the period $t \geq t_0 + h$ to obtain the estimates of the weights for the h model from (5.24). Differently, when $t < t_0 + h$ we assume the combination model forecasts equal to the AR univariate model:

$$\hat{y}_{i,t+k} = \hat{y}^{(AR)}_{1,t+k}. \tag{5.25}$$

Finally, in the next section we list some indicators of forecast performances for all the $NumMod$ models we examined in the previous subsections plus the combination model.

5.2.3 How to measure performances in our framework?

Although RMSE(h) is an average of errors, we must pay attention to undervalued performance results of a model with not so low RMSE(h). For instance, if the index for the AR(3) model is lower than the index of the MAR$(3)_1$ model, we may have in any cases that the latter performed better than the first one in many cases. We will judge a model useful and applicable if its index show similar results to the best one model for different values of h. Thus, we may assign one point when MSE(i, t, h) is larger than the same index for another model for each $i = 1, \ldots, n$, $t = s, \ldots, T$, and each $h = 1, \ldots, H$ in every simulation observed. Collecting all the points we obtain a new index. It measures the relative efficacy of a model and hence the proximity of the same to another model in a period of $T - s$ estimation windows. Thus, we define a score function:

$$Q_{\frac{mdl1}{mdl2},h} = \sum_{t=s}^{T-h} \sum_{i=1}^{n} \sum_{g=1}^{NG} \sum_{d=h}^{H} I \left(\begin{array}{c} \text{MSE}(i,t,d)_{g,mdl1} \\ \text{MSE}(i,t,d)_{g,mdl2} \end{array} \right), \tag{5.26}$$

where $mdl1$ and $mdl2$ are two forecast models and the I function assign one if RSME(i, t, d) for one model is better then the RSME(i, t, d) for another model for the exact day, product, simulation, and forecast period. We may consider points when root mean squared error is simply lower than other or to give points for differences of 5% or 10%. Values of Q give us a measure of performance over a large number of cases and of the variability of the results. If we change the parameter h we are interested to compare the model performances h periods ahead. For instance, $Q_{\text{AR/VAR},5}$, compares dynamic performance of AR and VAR models considering from the sixth day ahead predictions until the H-th one.

Now, we introduce another new methodology to compare the performances of several interchangeable models for forecasting. Under the assumption of clairvoyance of the researcher, we can build a forecast precision index for a set of models (standard and not), which uses the best performance in each period, switching from one model to another according to the lowest residuals. In this way, we obtain "at posteriori" a lower bound for the RMSE of each model. Then, we have an index measuring the potential of multiple forecast technique respect to standard one. If the lower bound for the RMSE is very distant from the best model of those examined, we may opt for a combination model to improve performances like in the previous sub section. We call the RMSE for the ideal combination model RMSE_{Bottom} and we use it to prove the advantages of combination models. We have:

$$\text{RMSE}_{Bottom}(h) = \sqrt{\frac{\sum_{g=1}^{N_G}\sum_{t=s}^{T-h}\sum_{i=1}^{n}\min_{g,mdl} MSE(i,t,h)}{n\cdot(T-s-h)\cdot N_G}}. \tag{5.27}$$

We have the following definitions:

Definition1 *Potential of a set of forecast models* The area between the RMSE_{Bottom} and the lower RMSE among the models examined.

Definition 2 *Forecast precision value of the combination model* The area between the RMSE_{Combi} and the lower RMSE among the models examined.

Summarizing, we have examined the ideal framework to observe and forecast the trends of product prices when products are essentially assembled parts. Hedonic prices can be extracted via multiple methodologies which we divided in two families: algorithm that estimates in the mean time hedonic prices and parameters through likelihood inference, and algorithm that estimates hedonic prices using known parameters. A set of forecast values is available, and a technique to combine them is the model in (5.24). A manufacturer-agent has the opportunity to use the best model for decision processes. In the next section, we show an application of our forecast framework in TAC SCM.

5.3 Experimental Analysis for Real Time Algorithms

To analyze results of hedonic framework including multiple forecast models we continue to use data from the multi agent simulation of the computer market supply chain, TAC SCM. In this application, data to test structural combination model are extracted from an archive and consists of 85 games. We used 30 games for graphics and for training of algorithms, and the remaining 50 games for measure the performances. A box-and-whisker graph shows the variability of the larger set of games used in the final application. The representation is very brighten because we can deduce the usual behavior of product prices in TAC SCM. In the first days of the game, they have a value near to the nominal prices defined in (5.7).

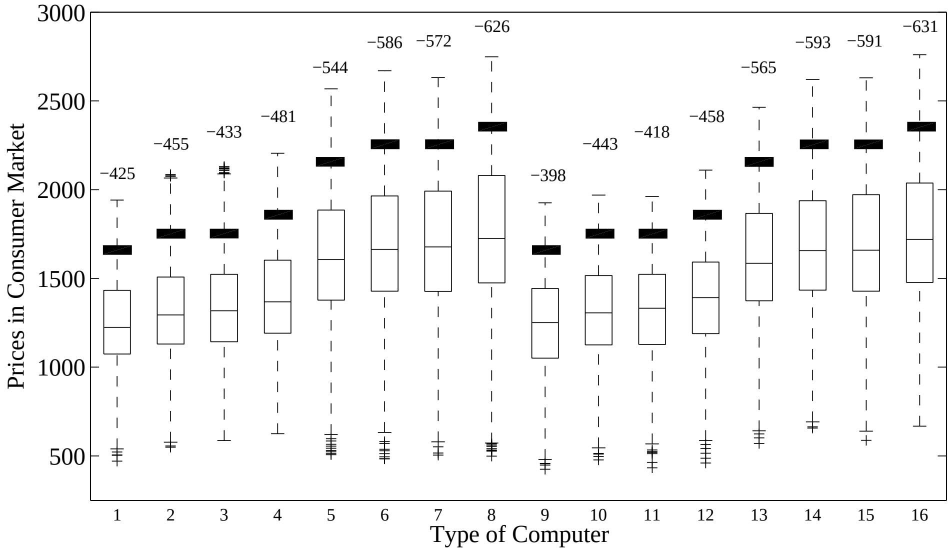

Figure 5.2: Box–and–whisker plot of product prices in 30 games in each period (1–217 days for a total of 6510 prices for each computer). Large bars represent nominal prices, crosses outliers, and thin bars are the medians. At the top of each box plot there is the difference between median and nominal cost for every PC.

Medians are very distant from product nominal prices. The distance represents the drop of a product price from initial reference price (e.g. for the Pintel computer of the kind eight the average drop is about 26.6%, and for IMD computer of the kind nine is about 19.9%). The drop of prices is due to the competitiveness of both markets. Suppliers and manufacturers tend to decrease production costs through an optimal allocation of resources and productive factors. Computer market is a conventional example for the phenomenon of that drop of prices during the shelf life of product. Variability of product prices is represented by dotted vertical lines in figure 5.2. There is the tendency to low the product prices under the average prices which causes no skewness in their distributions. In the last days of the game, the agents sell out and empty their inventories. Maybe product prices reach out the minimum levels in that period. All this factors affects the forecasts. The differences amongst types of product are relevant also when we average many simulation results.

5.3.1 Results for standard models

In our univariate characterization, we set $p = 3$ for each kind of product price process since the partial auto-correlation function of the time series showed high values until this lag for all of them.

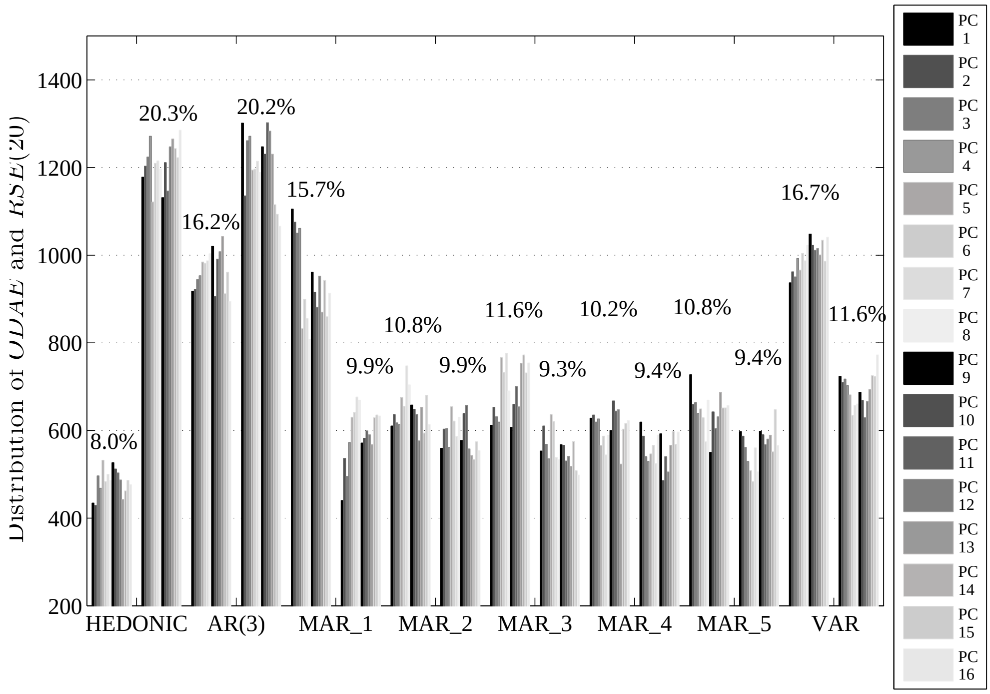

Figure 5.3: Number of cases with minimum average one-step-ahead (first series of columns) and root square relative errors at 18th day (second series of columns) for different models in 50 games and for 199 periods (18-217). Hedonic prices are estimated via *Algorithm 1.A*. Percentages are calculated over all the products

We estimated the univariate model, AR(3) for every window after eighteen periods of the game. In fact, from the first days of the game AR(3) model is not consistent since values are so few to estimate it correctly. Agent should prefer a simple AR(1) model in that case. After the first 10 days AR(3) starts to work and its performances showed robustness and good prediction. But for the output we will show only results after the 18th day because this is the first day of estimation of the multivariate model. All values of t-Student for estimated parameters say that coefficients are not null and for this we did not show them. All the R^2 shows an high value around 0.90-0.98.

The ODAE(i, t) and the MSE$(i, t, 20)$ for the univariate standard model gives optimal results as well as VAR and MAR_1 (see figure 5.3). Differently from MAR_1 (winning ODAE model for computers with ID 1, 2, 3 and 4) and VAR (winning ODAE model for computers with ID 8, 9, 10, 11, 14, 15, and 16), AR(3) is the best ODAE model for computers with ID 5, 6, 7, 12 and 13. The performances after 20 days, measured by an average of MSE$(i, t, 20)$ over the 199 periods of the game, are completely different because hedonic model substitutes the MAR_1 model, whereas AR(3) maintains the second position. Further, AR(3) becomes the best autoregressive model also for computer with ID 1, 3 and 4. Figure 5.4 shows the

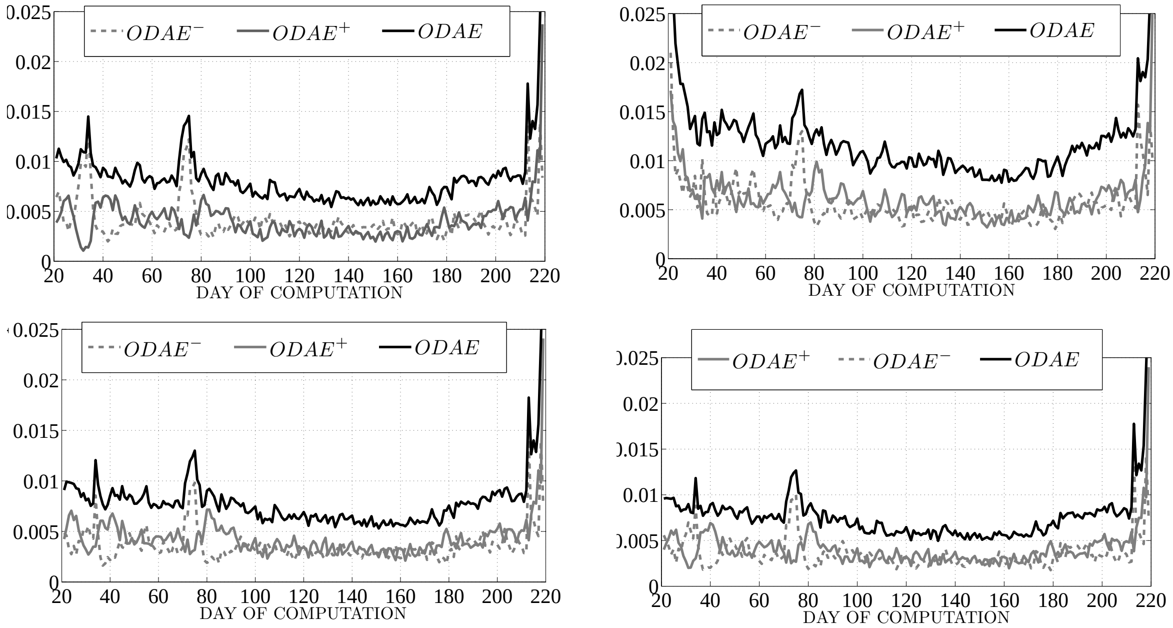

Figure 5.4: Average ODAE performances for AR, VAR, MAR_1, and MAR_3 model in 50 games and for 199 periods (from day 18 until the day 217).

average ODAE performances during the periods of a game. The graph is useful to understand if there are differences during specific periods of a game.

Multivariate models performs better after a certain number of days since they need a large number of data to estimate correctly the coefficients. Univariate and bivariate models have good results also in initial phase. Same graphs show three periods of radical change in prices, when the $ODAE^-$ doubles its value: around the 34th day, the 54th day, and at the end of the game. Risks of all the models are higher for over-estimation than under-estimation. Prices increase or decrease in those periods out of normal, and we find a periodic stochastic volatility in the central phase of the game. It is due on the periodic market information which update agent strategies every 20 days. Comparing figures 5.2 and 5.4, we may deduce that volatility is quite predictable with our models after 2 months of game, until the last days, when it becomes higher than ever. Anyway, hedonic information is able to improve the univariate model of 30% in those periods. The score function (see Table 5.1) between AR(3) and other models shows robustness of univariate model in each period of the game. Table refers to root square relative errors for all types of computer. Furthermore, AR(3) is the best model in initial periods since it improves of 15% one time over six results of the other models.

In our application, VAR(1) model behaves so well for ODAE to be selected the first model in our framework but not in the first phase of the game. The disadvantages are: the complicated lecture of the dynamic multipliers of matrix $\mathbf{\Pi}$ that should be lay in a ball of the unit value, but this rarely happens; the weakness of the model at the last periods of the game.

The one-day-ahead relative error for VAR is increasing after the 180 days (see Figure 5.4). This is due on the anomalous behavior of the agents that tend to empty their warehouses and price the products without consider its mean historical value and usual co-dependencies.

The RMSE shows how multivariate models performs optimally in the initial and middle periods. But we should not use only VAR model to predict prices in that period. In fact, the score function (see Table 5.1) between VAR and other models shows many times (20% for HM and 25% for AR) it is better to use the latter.

5.3.2 Results for models including hedonic values

We show forecast results for each of the five hedonic specifications (*Algorithm 1.A*, *Algorithm 1.B*, *Algorithm LR*, *Algorithm 2*, and *Algorithm 3*) and the five models, MAR$(3)_{1-5}$, under the assumption that future developments of components valuations follow the estimated transition matrices $\hat{\mathbf{\Phi}}$. In standard models future forecasting valuation of computer prices tends toward a stabilization due on the mean reverting effect. Thus, our agent could fall in the error to bet in a future stability of product prices. Differently, the hedonic forecasting values follow a different mean reverting effect given by (2.9).

Firstly, all the algorithm rules and steps are well examined in the previous paragraphs except the *Algorithm 3*, which needs of a methodology to estimate an alternative transition matrix, $\mathbf{F}$. In this application we use a set of thirty training games to assign diagonal entries, $\phi_1, \ldots, \phi_5$. Through *Algorithm 1.A* we estimate the thirty transition matrices $\Phi_{T,g}$, where $T = 220$, the last day in which agent sell products, and g is the number of the game. Then, we compute the mean matrix with respect to all the games:

$$\bar{\mathbf{\Phi}}_T = \begin{bmatrix} 0.9934 & 0.0286 & -0.0008 & 0.0328 & 0.0447 \\ 0.0008 & 0.9748 & -0.0027 & -0.0033 & 0.0007 \\ 0.0022 & -0.0135 & 0.9836 & 0.0347 & -0.0216 \\ 0.0033 & -0.0037 & 0.0044 & 0.9291 & 0.0039 \\ 0.0030 & -0.0026 & 0.0026 & 0.0205 & 0.9159 \end{bmatrix}. \tag{5.28}$$

Actually, it is very similar to a diagonal matrix, but for diagonalize it we calculated the generic one day ahead rate using the initial values for components:

$$\tilde{\mathbf{z}}_1 = \bar{\mathbf{\Phi}}_T \cdot \mathbf{z}_0, \tag{5.29}$$

Table 5.1: $Q_{\frac{mdl1}{mdl2},1}$ in each period of forecast. Pts 0 if MSE(i, t, h) is lower for the first model respect than second model. Pts α if MSE(i, t, h) is lower of α for the first model respect than second one

Period	AR/HM		AR/MAR$_1$		AR/VAR	
	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)
21–60	59.3	13.9 (7.9)	54.6	12.3(7.2)	63.5	20.5(13.0)
61–100	58.7	10.1(5.1)	52.5	6.7(3.1)	59.1	11.3(5.7)
101–140	56.7	5.5(2.4)	52.4	3.3(1.2)	55.8	6.1(2.6)
141–180	60.9	7.9(3.3)	54.3	5.8(2.4)	54.4	7.5(3.1)
181–210	63.6	4.7(1.6)	56.3	2.9(1.0)	60.5	5.4(1.9)
21–210	59.4	8.8 (4.3)	53.8	2.4(1.1)	58.5	2.9(1.4)
	MAR$_1$/HM		MAR$_1$/AR		MAR$_1$/VAR	
	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)
21–60	55.6	4.3 (2.2)	45.1	5.0(2.8)	58.2	15.6(10.2)
61–100	57.4	4.5(2.3)	47.5	3.6(0.9)	56.5	7.8(3.8)
101–140	56.5	2.9(1.2)	47.5	1.6(0.2)	54.6	4.1(1.8)
141–180	56.8	3.7(1.5)	45.7	2.0(0.4)	50.8	3.7(1.4)
181–210	61.2	2.5(0.8)	43.7	0.6(0.0)	55.8	3.7(1.1)
21–210	57.1	3.7 (1.7)	46.1	2.8(1.0)	55.1	7.3(3.9)
	HM/AR		HM/MAR$_1$		HM/VAR	
	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)
21–60	40.4	5.1 (2.8)	40.9	2.9(1.7)	53.0	14.8(9.6)
61–100	41.3	3.6(0.9)	41.4	1.2(0.4)	49.5	5.8(2.6)
101–140	43.3	1.9(0.2)	43.2	1.0(0.3)	46.8	3.3(1.5)
141–180	39.1	2.3(0.6)	42.6	1.5(0.5)	44.2	2.6(1.0)
181–210	36.4	0.7(0.0)	38.6	1.1(0.3)	45.7	2.9(0.9)
21–210	40.5	3.0 (1.0)	41.7	1.6(0.7)	48.1	6.2(3.3)
	VAR/HM		VAR/MAR$_1$		VAR/AR	
	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)	Pts 0	Pts 5%(10%)
21–60	45.7	8.0 (4.5)	40.8	7.4(4.4)	36.3	4.7(2.7)
61–100	49.5	5.0(2.2)	43.0	3.3(1.4)	40.9	3.2(0.7)
101–140	52.8	2.7(1.1)	45.2	1.5(0.6)	44.2	1.6(0.2)
141–180	55.2	3.4(1.3)	48.8	2.6(0.8)	45.5	2.4(0.5)
181–210	54.0	2.3(0.7)	44.0	1.5(0.3)	39.5	0.7(0.0)
21–210	51.1	4.5 (2.1)	44.4	3.5(1.6)	41.5	2.7(0.9)

Table 5.2: Total average, maximum and minimum times for the estimation and forecast of the hedonic prices from $t = 18, \ldots, 217$. Daily operation times for the estimation of the hedonic price.

	Total Performances			Daily Performances		
	Avg Est. Time (min)	Max Est. Time (min)	Min Est. Time (min)	Avg Est. Time (sec)	Max Est. Time (sec)	Min Est. Time (sec)
Alg1.A	16	24	8	3.36	92.90	0.32
Alg1.B	12	14	6	2.76	61.09	0.32
Alg2-3	5	7	4	0.04	0.12	0.01

and then, we calculate the m rates z_{1j}/z_{0j}. We obtain the vector of dynamic multipliers [0.998 1.000 0.993 1.009 0.999] which is the diagonal of the transition matrix $\mathbf{F}$ to be used in the *Algorithm 3*.

All the algorithm performances about time of computations for hedonic models are showed in table 5.2, for the testbed of 50 games. We see how the estimation of the parameters requires at least the 50% of the total time spent by the forecasting framework. Daily performances of *Algorithm 1.A* depend on the period of estimation. Strangely, we have tested that estimation windows in the interval $(30 - 60)$ require more iterations (and therefore time) with respect to other windows. It is interesting to note the good performances of *Algorithm 1.B*, which estimates parameters by a fixed number of iterations. Obviously, the prediction algorithms are much faster and stable than *Algorithm 1.A* and *Algorithm 1.B*, in terms of time.

In Figure 5.4 we omitted the values for the DHMM since it provides always not good performances in the first period of forecast. This is an effect of the unpredictable scarce performances in some games due on the short estimation window, divergence of the Kalman filter and on the high volatility. The same defects are not so evident if we consider one of the simplest models MAR_j. Until the half of the forecast window VAR model is the best model in sense of average ODAE. Differently, after this period hedonic models and AR start to give optimal predictions (see figure 5.3 for MSE_{20}). Figure 5.4 shows the average ODAE over the product types. The MAR_1 and AR models are more robust than VAR model. In fact, the latter is better only for specific types of products, and when it is averaged its performances are not so good. Also MAR_1 model has good ODAE performances. It considers the base product hedonic price in the autoregressive equation and it is the best model for computers with ID 1-4.

A comparison between pure hedonic formulations is given in figure 5.5. We tested the models across time series of increasing length from the 18th until the 217th day. From the graph is evident the good performance of *Algorithm 1.A* with respect to other algorithms at least in the first ten days. It is due to the estimation method of parameters which is more exact than likelihood ratio. The result is more than unexpected, and it confirms previous results in the first application of chapter two. The motivation for the inefficiency of the likelihood ratio method is not the lack of normality in the data (residuals). The main reason lays in the high dimensionality of the likelihood, which complicates the maximization. The use of the stopping rule based on the nearness of transition matrix leads to an identification of parameters, which are more efficient than parameters estimated via *Algorithm LR*.

The difference between the *Algorithm 1.A* and *Algorithm 1.B* are due to the greater number of iterations required from the first algorithm. Anyway, noise model reaches out

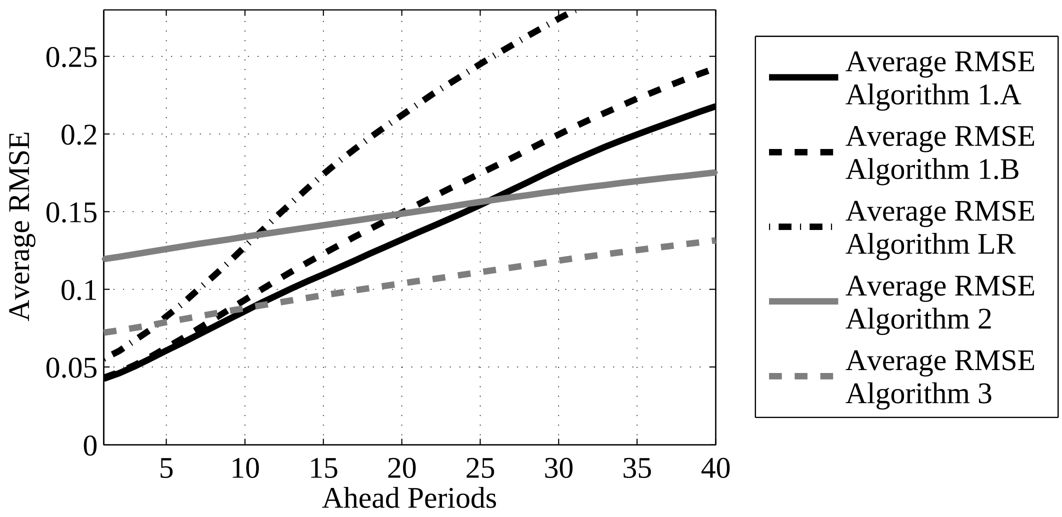

Figure 5.5: Average root mean squared relative error for all PCs in 50 games computed off line from 18th day until the 217th. The values are normalized by the nominal price of each computer, and hence comparable.

every algorithm without an estimation procedure but assuming independence in the hedonic price process and the same price for each component for all the game. Finally, the *Algorithm 3* provides optimal performances after the 10th day. Normally, hedonic prices for a part tend to diminish during the game, overall for the base computer. Through the estimated diagonal matrix **F**, we are able to predict the best sequence of product prices for the next tenth, eleventh, . . . , forty th days.

Figure 5.6 shows off line performance results for all models in 50 games with forecast windows of 40 periods compared to the other models, using the index RMSE(h) as in (5.11). In the following we omit the other hedonic algorithms, and analyze the output of the sole *Algorithm 1.A*. We see how AR(3) performances are the best ones amongst the single autoregressive models. Our hedonic model has a strangely behavior during the first ahead days of forecast since its estimates some times are not sharp. Although, the same values are quite similar for AR(3) and MAR$(3)_m$, which are not affected by identification problems. It confirms the hypothesis of good performances of bivariate models not so different respect than multivariate ones. Hedonic information may improve the forecasts in several situations as in the middle-final days of the game. Both bivariate and multivariate hedonic models improve forecasts around the 1-2% respect than standard autoregressive models, corresponding in an average absolute value difference of 20-50 per unit produced. Hence, a simple inclusion of hedonic information may improve the forecast price agent framework. We see how all models improve with the increasing time series length. After 150 days, the performances are quite similar except first days forecast of multivariate models. Our agent should prefer to use a

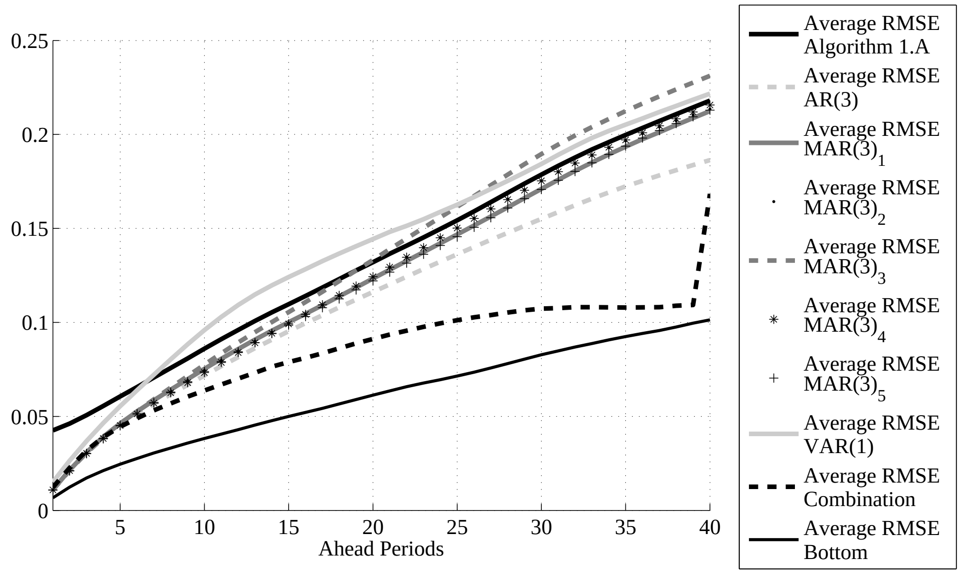

Figure 5.6: Average root mean squared relative error for all PCs in 50 games computed off line from 18th day until the 217th. The values are normalized by the nominal price of each computer, and hence comparable

different model depending on the period of the game and on the ahead days of forecasts. Thus, our hedonic multivariate model performs very well in certain games and periods and it is better then VAR(1) until the last days of the game. Failures of the model are due on the short time of computations in on line conditions, the larger number of parameters to be estimated than other models and on the non-observability of hedonic variables. The larger the number of variables is in the model the larger probability to have forecast errors.

In Table 5.1, points obtained from HM are compared to AR, VAR and MAR_j scores. The good performances of hedonic model, when algorithm achieves to estimate perfectly the component implicit price behavior, pass from 40% to 43.6% in the middle periods. In this period the performances of HM are the highest of the game since agent strategies do not affect the volatility of the prices.

In figure 5.7 the MAPE is represented for all the models as alternative measure to RMSE for comparing performances. The linearity of that measure affects all the plots in the figure except the combination model. Results for average MAPE are quite similar to the average RMSE. It means that differences between errors in (5.13) and (5.14) are relevant. Since normalize prices usually are higher than actual prices, RMSE is lower than MAPE for all the models except the combination model. Conclusions for MAPE are the same as RMSE analysis.

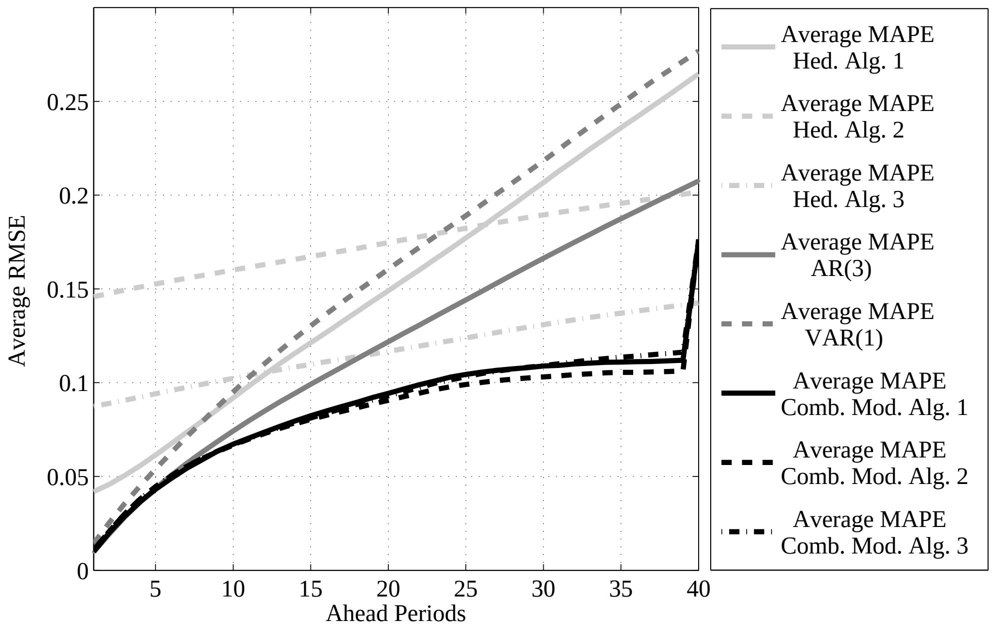

Figure 5.7: Average mean absolute percent error for all PCs in 50 games computed off line from 18th day until the 217th

5.3.3 The combined models results

In on line forecasting modules for product prices, the choice for a combination of forecast based on the previous performances is somewhat compulsory. From previous results it is clear that each model can contribute to increase forecast performances depending on the periods, agent strategies, volatility and forecast window. Figure 5.6 shows results for RMSE_{Bottom} and the combination models calculated for the same games, computers, and periods. The large distance between RMSE for AR(3) (the best model among those observed) and the RMSE_{Bottom} suggests us to implement the combination model given in (5.24) in any case.

We tested it in the same games obtaining the RMSE represented in figure 5.6 with a dashed black line. The strange tail of the plot for this model is due on the substitution of the predictions with the values provided by the autoregressive univariate model (AR(3)), when $t < t_0 + s + h = 18 + 7 + h = 25 + h$. We conclude that starting from the 10th ahead period it is convenient to use the combination model instead of AR(3). The result is not surprising because the model performances do not include a large part of initial days. The first 40 ahead period forecast is ready after 65 days, whereas the other models provide the same prediction after $t_0 = 18$ days. But in any case, forecast results improve of 50% with the combination technique. From the analysis of the coefficients b depending on the period t, the product i, and the ahead period h, it is clear that every sub-model contributes to reach the optimal performances.

The potential of the set of forecast models included in the combination one is very large, like the value of the combination model itself.

5.3.4 Conclusions About Forecast Framework Application

We examined an application of the dynamic multivariate hedonic model to estimate parameters and implicit prices in the supply chain computer markets. Our methodology for parameter estimation is based on the second contribution of the thesis, the stopping rule number one. We have demonstrated that it outperforms the likelihood ratio rule. Although it does not reach out standard forecast , it provides the hedonic information useful for other models.

The assumptions about hedonic prices may help in forecast future product prices as we have shown for noise and diagonal models. The latter provides the best results in the framework for medium/long predictions. When the markets show the same pattern for trends in many succeeding years, very simple hypothesis can help the agent. Furthermore, the versatility of the prediction algorithms solves the parameter identification problem for computation times.

The complete framework includes standard models and hedonic ones under different assumptions. We do not wish to select the best model. Our goal is to give an agent all the information under different assumptions. The researcher has all the instruments to verify on line the correctness of the algorithm via mean squared error performances. Using hedonic variables in every decision process he can improve it, like in the forecast module case. Finally, we have presented the combination model, a new on line technique to improve the future forecast precision using previous results about it.

In the future work, we want to study the opportunity to estimate shorter pattern of hedonic prices under stationarity assumptions for each of them. As we have seen in the supply chain markets, multiple product price series often have not stability and hence stationarity for more than few weeks. More precisely, Kalman filter technique have showed that longest series cause estimation problems, because the assumptions on stability are violated. We will point to regime detection under the hedonic state space model. In that way, we can further improve the estimation methodology of parameters, and to find the centroid parameters of each regime.

Chapter 6

Conclusions and Future Works

Throughout this thesis, the hedonic information benefits in supply chain markets were showed. The type of that supply chain is two tiers with parts provided by suppliers assembled in products by manufacturers. Customer demands are independent and possibly segmented. The best example of similar environment is the computer market, while many other examples may be done.

6.1 Research Contributions and Results

In the sequel of the last chapter, we prefer discuss the implications of the results of the thesis by listing its contributions. As we have seen in the introduction, five contributions are given to the research

6.1.1 Research Contribution 1 - The hedonic model and its specification

Firstly, the hedonic model and correspondent algorithm were showed. We start from the inclusion of the hedonic price in a dynamic mapping for product prices. The design matrix, the transition matrix, the initial vector of hedonic prices, the covariance matrices for noise, disturbances, and initial distribution, form the hyperparameter of the hedonic model. Each of them requires initial assumption for the identification. An application in a set of ten games in TAC SCM was offered. It shows the interpretation of the hyperparameter.

In the second chapter we extend the static concept of hedonic price in dynamic sense. We introduced a state space model representation. In the third chapter, we listed several alternative hedonic models. Each of them can improve the analysis of the markets and the decision processes of the manufacturer-agent.

6.1.2 Research Contribution 2 - An algorithm for the hedonic model for state space models in high dimensionality

In the standard model of time series analysis, the identification of the parameters enjoys of the property of the monotonicity of the log-likelihood. Differently, when the researcher faces the state space model specification, without any knowledge about the parameters, he is obliged to implement another convergence criteria (second contribution). The latter can not be based on the multivariate Gaussian likelihood, like in the standard methodology. When the dimension of the space of variables is too much high, the maximization of the likelihood can not be stopped according to the likelihood ratio test. It is preferable a stopping rule for Expectation-Maximization plus Kalman filter (EM+KF) algorithm based on the nearness between transition matrices.

In the second chapter we list the code steps for the *Algorithm 1*, the basic algorithm for EM+KF. We introduced the problems link to the procedure and the relative solutions in literature. The first application in TAC SCM shows a comparison of time performances and the properties of the model. In the third chapter some solutions to avoid the estimation of the hyperparameter of the state space model are listed. In the fourth chapter, a detailed analysis of the behavior of the Kalman filter in the nearing of the convergence to the solution is given. We presented two types of innovative algorithm with respect to the literature (Shumway & Stoffer, 2006). The first one, *Algorithm 1.A* is the best algorithm which minimizes the distance between two transition matrices in close iterations. The second one, *Algorithm 1.B* offers an approximated solution but it is quicker than the first one.

6.1.3 Research Contribution 3 - A complete specification of a framework for forecast product prices in a dynamic multivariate process for heterogeneous supply chain markets

When an agent wants to forecast future prices for the goods in a product variety of dimension n sharing m parts he can use a module based on the dynamic multivariate hedonic model. Hedonic price measure the preference of the customer for the part/characteristic of the product. Hence, under several assumptions, hedonic prices may be differently evaluated an to contribute at forecast analysis. For instance, we can assume that the hedonic prices for a CPU in the computer market follow a multivariate random walk as in the Noise model in (3.1). Otherwise, we can opt for the estimation of the transition matrix like in the *Algorithm 3*. Furthermore, we included in the framework the conventional autoregressive models like AR and VAR models. The univariate model achieves to improve the short term forecasts of product prices. Differently, hedonic prices improve the long term predictions.

6.1.4 Research Contribution 4 - An on line forecast combination model in which weights are estimated via linear regression on the previous performances

Research contribution three leads to the construction of a combination model where all the predictions are regressed according to the previous performances. The on line combination model is an extension of the model in Bates & Granger. We extended it in the hedonic sense with the inclusion of implicit prices. Results show an increasing in the performances similar to other advanced techniques in (Ketter *et al.*, 2009). Furthermore, innovative concepts of forecast analysis are introduced for a complete and clear picture of the supply chain, from the point of view of product prices.

6.2 Future Works

We presented a dynamic multivariate hedonic model to explain and forecast prices of heterogeneous products sharing common components. We showed one way to use hedonic information in a dynamic supply chain, in forecast analysis. In our future research, we want to test connections between procurement prices and hedonic prices. In such circumstance, hedonic values can be used also as a predictor of unobservable component prices. So, we can consider them as predictors of both supply chain market prices. Secondly, it is worthwhile to integrate our base model with the Markov regime switching approach to cope with structural changes in the model hyper-parameter. Furthermore, the methodology can be generalized to the state-space models in high-dimensioned cases. With the same stopping rule we can detect break-points and regimes for improving the knowledge of the markets.

Obviously, our algorithm could be modified and implemented in several ways. For instance, considering that our model should ultimately be useful for real-time forecasting, the algorithm might be suppleted with exogenous information.

Appendix A

Discrete-Time Systems and Kalman Filter

In this appendix, we will give a mathematical description of the Kalman filter starting from the theory of the discrete-time systems. The material presented is taken by (Simon, 2006). We define a linear discrete-time system as:

$$\mathbf{z}_t = \mathbf{\Phi}_{t-1}\mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_{t-1}, \quad t \in N, \tag{A.1}$$

where $\boldsymbol{\varepsilon}_t$ is Gaussian zero-mean white noise with covariance $\mathbf{\Sigma_t}$. How does the mean and the covariance of the state $\mathbf{z}_t$ change with time? Taking the expectation of right side of equation (A.1) we have for the mean:

$$\bar{\mathbf{z}}_t = E(\mathbf{z}_t) = \mathbf{\Phi}_{t-1}\bar{\mathbf{z}}_{t-1}, \tag{A.2}$$

and for the covariance:

$$\mathbf{P}_t = E[(\mathbf{z}_t - \bar{\mathbf{z}}_t)(\mathbf{z}_t - \bar{\mathbf{z}}_t)^T] = \mathbf{\Phi}_{t-1}\mathbf{P}_{t-1}\mathbf{\Phi}_{t-1}^T + \mathbf{\Sigma_t}. \tag{A.3}$$

This is called a discrete-time Lyapunov equation, or a Stein equation. It is well known in control theory for discrete-time systems. Last two equation are the fundamental in the derivation of the Kalman filter. The following theorem of stability, whose proof can be found in the book of Kailath, 2000, gives the conditions under which the discrete-time Lyapunov equation has a steady-state solution with $\mathbf{\Phi}$ and $\mathbf{\Sigma}$ constant. We resume the property of stable systems:

1. a stable $\mathbf{\Phi}$ has eigenvalues $\lambda_i(\mathbf{\Phi})$ less than one in magnitude;

2. a unique solution $\mathbf{\Phi}$ for (A.3) exists if and only if $\lambda_i(\mathbf{\Phi})\lambda_j(\mathbf{\Phi}) \neq 1, \forall i, j$. The matrix

solution is symmetric and may be stable or not;

3. in the case of stable transition matrix the covariance matrix can be written as

$$\mathbf{P} = \sum_{i=0}^{\infty} \mathbf{\Phi}^i \mathbf{\Sigma} (\mathbf{\Phi}^T)^i. \tag{A.4}$$

4. if $\mathbf{\Phi}$ is stable and $\mathbf{\Sigma}$ is positive (semi)definite, then the unique solution $\mathbf{P}$ is symmetric and positive (semi)definite.

The solution of the linear system of equation (A.1) is given by:

$$\mathbf{z}_t = \mathbf{\Phi}_{t,0}\mathbf{z}_0 + \sum_{i=0}^{t-1} \mathbf{\Phi}_{t,i+1}\boldsymbol{\varepsilon}_i, \tag{A.5}$$

where each matrix $\mathbf{\Phi}_{t,i}$ is defined as:

$$\mathbf{\Phi}_{t,i} = \begin{cases} \mathbf{\Phi}_{t-1}\mathbf{\Phi}_{t-2}\dots\mathbf{\Phi}_i & \text{if } t > i \\ \mathbf{I} & \text{if } t = i \\ \mathbf{0} & \text{if } t < i \end{cases} \tag{A.6}$$

If $\mathbf{z}_0$ and the series of disturbances, $\{\boldsymbol{\varepsilon}_i\}$ are unknown but are Gaussian random variables, then the state variable $\mathbf{z}_t$ is itself a Gaussian random variable, $\mathbf{z}_t \sim \text{MVN}(\bar{\mathbf{z}}_t, \mathbf{P}_t)$.
Suppose we have more linear constraint in our discrete-time system:

$$\begin{aligned} \mathbf{z}_t &= \mathbf{\Phi}_t\mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_{t-1} \\ \mathbf{y}_t &= \mathbf{H}_t\mathbf{z}_t + \boldsymbol{\nu}_t \\ \boldsymbol{\varepsilon}_t &\sim \text{MVN}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\varepsilon}_t}) \\ \boldsymbol{\nu}_t &\sim \text{MVN}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\nu}_t}) \\ E[\boldsymbol{\varepsilon}_t {\boldsymbol{\nu}_t}^T] &= 0 \end{aligned} \tag{A.7}$$

To estimate the state $\mathbf{z}_t$ based on the availability of the noisy measurements $\{\mathbf{y}_t\}$ we consider four different estimates:

- $\hat{\mathbf{z}}_t^+ = E[\mathbf{z}_t|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t]$ = *a posteriori* estimate;
- $\hat{\mathbf{z}}_t^- = E[\mathbf{z}_t|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{t-1}]$ = *a priori* estimate;
- $\hat{\mathbf{z}}_{t|T} = E[\mathbf{z}_t|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T]$ = smoothed estimate for $T > t$;
- $\hat{\mathbf{z}}_{t|T} = E[\mathbf{z}_t|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T]$ = predicted estimate for $T < t - 1$.

For each of upper estimates we can define the relative covariance $\mathbf{P}_t^+$, $\mathbf{P}_t^-$, and $\mathbf{P}_{t|T}$. As we have seen in the chapter two, the formula for the discrete-time Kalman filter, given in (A.12)–(A.19) estimates all those values and the relative covariance matrices.

From another point of view, the Kalman filter may be viewed as the vector minimizing the quantity:

$$\min E[(\mathbf{z}_t - \hat{\mathbf{z}}_t)^T \mathbf{S}_t (\mathbf{z}_t - \hat{\mathbf{z}}_t)], \tag{A.8}$$

where the matrix $\mathbf{S}_t$ is a positive definite weighting matrix. Under the assumption of Gaussian zero-mean, uncorrelated disturbances the estimates provided by the Kalman filter are the solution of (A.8).

A.1 Kalman Prediction

We now consider the state space model in (A.7), and we derive the prediction recursions. Our target is to estimate the projection of the process $\mathbf{z}_t$ onto the linear space spanned by the random vectors of the product prices, $\mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_t}$, which we denoted by $\hat{\mathbf{z}}_t^-$, the *a priori* estimates, or the prediction values.

A.2 Kalman Filtering and Smoothing

The Kalman filter algorithm starts with an estimator of the prediction value, the expected value of the current period implicit prices conditional on available product prices:

$$\mathbf{z_t^{t-1}} = E(\mathbf{z_t} | \mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_{t-1}}), \tag{A.9}$$

and the variance–covariance matrix of the errors:

$$\mathbf{P_t^{t-1}} = E\left[\left(\mathbf{z_t} - \mathbf{z_t^{(t-1)}}\right)\left(\mathbf{z_t} - \mathbf{z_t^{(t-1)}}\right)' \mid \mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_{t-1}}\right].$$

To compute filtered values and their variance-covariance matrix we start from the following quantities:

$$\mathbf{e}_t = \mathbf{y}_t - E\left(\mathbf{y}_t | \, \mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_{t-1}}\right) = \mathbf{y_t} - \mathbf{D}\mathbf{z_t^{t-1}},$$

where $\mathbf{e}_t$ is known as the *innovation* or *measurement residual.* Through the joint conditional distribution of $\mathbf{z}_t$ and $\mathbf{e}_t$ given by:

$$\begin{pmatrix} \mathbf{z}_t \\ \mathbf{e}_t \end{pmatrix} | \mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_{t-1}} \sim N\left(\begin{bmatrix} \mathbf{z}_t^{t-1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_t^{t-1} & \mathbf{P}_t^{t-1}\mathbf{D}' \\ \mathbf{D}\mathbf{P}_t^{t-1} & \mathbf{\Sigma}_t \end{bmatrix}\right),$$

we can derive the filtering value $\mathbf{z}_t^t$.
Considering the density function of the disturbances in (2.9) and (2.10), the Kalman filter is a Bayesian updating algorithm, where every value is dependent on the previous one until the initial condition. Using all available price information, a smoothed estimator of the expected current period states is obtained through:

$$\mathbf{z_t^T} = E(\mathbf{z_t} \mid \mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_T}). \tag{A.10}$$

The estimator of the variance-covariance matrix of the errors in (2.9) is defined as:

$$\mathbf{P_t^s} = E[(\mathbf{z_t} - \mathbf{z_t^s})(\mathbf{z_t} - \mathbf{z_t^s})' \mid \mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_s}] \tag{A.11}$$

both for the filter and the smoother, taking $s = t$ and $s = T$ respectively. For the filtering relations with $\mathbf{z_0^0} = \boldsymbol{\mu_0}$ and $\mathbf{P_0^0} = \mathbf{\Sigma_0}$, for $t = 1, \ldots, T$, we compute the following quantities:

- Predicted values of implicit prices:

$$\mathbf{z_t^{t-1}} = \mathbf{\Phi} \mathbf{z_{t-1}^{t-1}}. \tag{A.12}$$

  The recursion begins with $\mathbf{z_1^0}$ which denotes a forecast of $\mathbf{z_1}$ based on the initial value of $\boldsymbol{\mu_0}$. In subsequent steps the prediction values will be based on observations of $\mathbf{y}$, using (A.9).

- Variance-covariance matrix of predicted values:

$$\mathbf{P_t^{t-1}} = \mathbf{\Phi} \mathbf{P_{t-1}^{t-1}} \mathbf{\Phi}' + \mathbf{\Sigma}_\varepsilon \tag{A.13}$$

  for $t = 1$ we have $\mathbf{P_1^0} = \mathbf{\Sigma}_0$.

- Kalman gain:

$$\mathbf{K_t} = \mathbf{P_t^{t-1}} \mathbf{D}' [\mathbf{D} \mathbf{P_t^{t-1}} \mathbf{D}' + \mathbf{\Sigma}_\nu]^{-1}, \tag{A.14}$$

  which is a time-varying matrix used for updating filtered states.

- Filter values:

$$\mathbf{z_t^t} = \mathbf{z_t^{t-1}} + \mathbf{K_t} \mathbf{e}_t; \tag{A.15}$$

- Variance-covariance matrix of filtered values:

$$\mathbf{P_t^t} = [\mathbf{I} - \mathbf{K_t} \mathbf{D}] \mathbf{P_t^{t-1}}. \tag{A.16}$$

This $\mathbf{P_t^t}$ affects the distribution of estimated implicit prices.

Given a time series of n product prices, $\mathbf{Y_t} = \mathbf{y_1}, \ldots, \mathbf{y_t}$, equations (A.12)-(A.16) recursively generate estimates of the implicit prices $\mathbf{z_{t+1}}$.

Next, we consider the estimators for $\mathbf{z_t}$ based on the entire product price series $\mathbf{Y_T} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)'$, where $t \leq T$. For these smoothed estimators we compute:

- Smoothed gain

$$\mathbf{J_{t-1}} = \mathbf{P_{t-1}^{t-1}} \mathbf{\Phi}' [\mathbf{P_t^{t-1}}]^{-1}, \tag{A.17}$$

  time-varying matrices that measure the differences between filtered and smoothed values;

- Variance-covariance matrix of smoothed errors:

$$\mathbf{P_{t-1}^T} = \mathbf{P_{t-1}^{t-1}} + \mathbf{J_{t-1}}(\mathbf{P_t^T} - \mathbf{P_t^{t-1}})\mathbf{J'_{t-1}}; \tag{A.18}$$

- Smoothed values:

$$\mathbf{z_{t-1}^T} = \mathbf{z_{t-1}^{t-1}} + \mathbf{J_{t-1}}(\mathbf{z_t^T} - \mathbf{z_t^{t-1}}) \tag{A.19}$$

with $t = T, T-1, \ldots, 1$, and with $\mathbf{z_T^T}$ and $\mathbf{P_T^T}$ obtained via filter formulas (A.12)-(A.16).

Appendix B

The Expectation-Maximization Algorithm

One of the most popular methods for the specific class of models with hidden, missed, or latent values is the framework introduced by Dempster *et al.* (1977). In this appendix we rapidly highlight the framework. In the two steps of EM algorithm, the scope is to estimate the model parameter(s) for which the partially observed data are the most likely. In the expectation step, we compute the conditional expectation of the incomplete-data likelihood, where conditional refers to the partially observed data. In the maximization step, the conditional expectation of the likelihood is maximized under the assumptions that the hidden values are known, and we find the estimates for the parameters. This is possible under the Gaussian assumption, while for other distributions, exponential or not, it may be numerically infeasible to perform the maximization step.
The target of the algorithm is to maximize the incomplete-data likelihood:

$$L(\mathbf{\Theta}) = L(\mathbf{y};\mathbf{\Theta}) = \int_{\mathcal{X}(\mathbf{y})} f(\mathbf{x};\mathbf{\Theta})\mathbf{dx}, \tag{B.1}$$

with respect to the multi parameter $\mathbf{\Theta}$, for a family of density functions given by $f(\mathbf{x};\mathbf{\Theta})$, over the space generated by observed values. The function $f(\mathbf{x};\mathbf{\Theta})$ is the complete data likelihood, the likelihood of the hypothetical higher dimensional random variable. Here, X is the random variable including hidden (missed) values, and Y is the random variable of the observed values. Thus, there are "many-to-one" mappings from $\mathcal{X}$, the sample space of X, and $\mathcal{Y}$, the sample space of Y. It follows that $\mathcal{X}(\mathbf{y})$ is the subset of $\mathcal{X}$ determined by the equation $\mathbf{y} = \mathbf{y}(\mathbf{x})$.
For a given specification of $L(\mathbf{\Theta})$ there exists a family of specifications of complete data likelihoods $f(\mathbf{x};\mathbf{\Theta})$. We assume that the likelihood $L(\mathbf{\Theta})$ is positive, and the problem in

(B.1) is equivalent to maximizing the log-likelihood. If we take the logarithm of the complete data likelihood:

$$l(\mathbf{\Theta}) = \log L(\mathbf{\Theta}), \tag{B.2}$$

-and the probability density function, the conditional density of X given Y:

$$p(\mathbf{x}; \mathbf{\Theta}) = \frac{f(\mathbf{x}; \mathbf{\Theta})}{L(\mathbf{\Theta})}, \tag{B.3}$$

we can define the score function of EM algorithm the following integral:

$$Q(\mathbf{\Theta}|\mathbf{\Theta}^*) = \int \log f(\mathbf{x}; \mathbf{\Theta}) p(\mathbf{x}; \mathbf{\Theta}^*) \mathbf{dx}, \tag{B.4}$$

which results generally well defined for regular density functions. The score function is the expectation of the complete likelihood under the distribution of the incomplete data given by $p(\mathbf{x}; \mathbf{\Theta}^*)$. For instance, assume $\mathbf{\Theta}^{(0)}$ is the initial value for $\mathbf{\Theta}$. Then, in the first iteration, the expectation step requires the calculation of:

$$Q(\mathbf{\Theta}|\mathbf{\Theta}^{(0)}) = E\{\log L(\mathbf{\Theta})|\, \mathbf{y}, \mathbf{\Theta}^{(0)}\} = \int \log f(\mathbf{x}; \mathbf{\Theta}) p(\mathbf{x}; \mathbf{\Theta}^{(0)}) \mathbf{dx}. \tag{B.5}$$

The maximization step requires the maximization of $Q(\mathbf{\Theta}|\mathbf{\Theta}^{(0)})$ with respect to $\mathbf{\Theta}$. It provides a new value for the parameter, which we call $\mathbf{\Theta}^{(1)}$. Obviously, we have that:

$$Q(\mathbf{\Theta}^{(1)}|\mathbf{\Theta}^{(0)}) \geq Q(\mathbf{\Theta}|\mathbf{\Theta}^{(0)}), \quad \forall \quad \mathbf{\Theta}. \tag{B.6}$$

The algorithm continues the two steps procedure defined as follows:

- **E-Step.** Calculate:

$$Q(\mathbf{\Theta}|\mathbf{\Theta}^{(j)}) = E\{\log L(\mathbf{\Theta})|\, \mathbf{y}, \mathbf{\Theta}^{(j)}\} = \int \log f(\mathbf{x}; \mathbf{\Theta}) p(\mathbf{x}; \mathbf{\Theta}^{(j)}) \mathbf{dx}$$

.

- **M-Step.** Choose $\mathbf{\Theta}^{(j+1)}$ such that maximizes the score function $Q(\mathbf{\Theta}|\mathbf{\Theta}^{(j)})$.

The algorithm is repeated until the convergence of the incomplete-data likelihood. Hence, we compute, after the M-step, the difference:

$$L(\mathbf{\Theta}^{(j+1)}) - L(\mathbf{\Theta}^{(j)}), \tag{B.7}$$

and if it changes by a small value (prefixed) we break the procedure.
We can state that EM algorithm satisfies the following properties:

1. Any sequence $\{\mathbf{\Theta}^{(j)}\}$ increases the likelihood and $L(\mathbf{\Theta}^{(j)})$, and if bounded above, converges to some value L^*;

2. If Q is continuous in $\mathbf{\Theta}$ and $\mathbf{\Theta}^{(j)}$, then L^* is a stationary value of L. For the Gaussian distribution, the property holds in any case;

3. EM performances depend on the starting points;

4. if in addition to (2) or (3), for $j \to \infty$ we have $\left\|\mathbf{\Theta}^{(j+1)} - \mathbf{\Theta}^{(j)}\right\| \to 0$, then $\mathbf{\Theta}^{(j)}$ converges to a local maximum for Gaussian distribution

From the property of the logarithm we can further define:

$$H(\mathbf{\Theta};\mathbf{\Theta}^*) = Q(\mathbf{\Theta};\mathbf{\Theta}^*) - l(\mathbf{\Theta}) = -\int \log p(\mathbf{x};\mathbf{\Theta})p(\mathbf{x};\mathbf{\Theta}^*)\mathbf{dx}, \tag{B.8}$$

which is the entropy of the probability density function $p(\mathbf{x};\mathbf{\Theta}^*)$. Finally, we introduce the Kullback-Leibler divergence (or relative entropy) between the probability density functions $p(\mathbf{x};\mathbf{\Theta})$ and $p(\mathbf{x};\mathbf{\Theta}^*)$:

$$H(\mathbf{\Theta};\mathbf{\Theta}^*) - H(\mathbf{\Theta}^*;\mathbf{\Theta}^*) = -\int \log \frac{p(\mathbf{x};\mathbf{\Theta})}{p(\mathbf{x};\mathbf{\Theta}^*)} p(\mathbf{x};\mathbf{\Theta}^*)\mathbf{dx}, \tag{B.9}$$

Bibliography

Anderson, S.P., De Palma, A., & J.F., Thisse. 1992. *Discrete choice theory of product differentiation*. Massachusetts Institute of Technology.

Azzalini, A. 1996. *Statistical inference based on the likelihood*. London: Chapman and Hall.

Bartlett, M.S. 1954. A note on Multiplying Factors for Various Chi-Squared Approximations. *Journal of the Royal Statistical Society (B)*, **16**, 296–298.

Bates, J. M., & Granger, C. W. J. 1969. The Combination of Forecasts. *OR*, **20**(4), pp. 451–468.

Benisch, Michael, Greenwald, Amy, Grypari, Ioanna, Lederman, Roger, Naroditskiy, Victor, & Tschantz, Michael. 2004. Botticelli: A Supply Chain Management Agent Designed to Optimize under Uncertainty. *ACM Trans. on Comp. Logic*, **4**(3), 29–37.

Berndt, E.R., & Rappaport, N.J. 2001. Price and quality of desktop and mobile personal computers: A quarter-century historical overview. *American Economic Review*, **91**(2), 268–273.

Box, G.E.P., & Jenkins, G.M. 1976. *Time Series Analysis:forecasting and control*. rev. ed. San Francisco: Holden-Day.

Branco, Fernando. 1997. The design of multidimensional auctions. *The RAND Journal of Economics*, **28**(1), 63–81.

Brown, Robert G., Meyer, Richard F., & D'Esopo, D. A. 1961. The Fundamental Theorem of Exponential Smoothing. *Operations Research*, **9**(5), 673–687.

Caines, P.E. 1988. *Linear Stochastic Systems*. New York: Wiley.

Case, B., Pollakowski, H.O., & Wachter, S.M. 2003. On choosing among house price index methodologies. *Real Estate Economics*, **19**(3), 286–307.

Chanel, O., Gérard-Varet, L.A., & Ginsburgh, V. 1996. The relevance of hedonic price indices. The case of paintings. *Journal of Cultural Economics*, **20**, 1–24.

Chen, Xin, Sean, X. Zhou, & Youhua, F.Chen. 2011. Integration of Inventory and Pricing Decisions with Costly Price Adjustments. *Operations Research*, **59**(5), 1144–1158.

Cheng, Edwin T.C., Li, Jian, Wan, Johnny C.L., & Shouyang, Wang. 2010. *Postponement strategies in supply chain management*. Springer Science+Business Media.

Cheung, Yiuming, & Xu, Lei. 2003. Further studies on temporal factor analysis: comparison and Kalman filter-based algorithm. *Neurocomputing*, **50**, 87–103.

Chinloy, P.T. 1977. Hedonic price and depreciation indexes for residential housing: A longitudinal approach* 1. *Journal of Urban Economics*, **4**(4), 469–482.

Clay, K.B., Smith, M.D., & Wolff, E.D. 2004. Static and Dynamic Pricing in Online Markets. *Working Paper Series of Social Science Research Network*, 63–77.

Collins, John, Arunachalam, Raghu, Sadeh, Norman, Eriksson, Joakim, Finne, Niclas, & Janson, Sverker. 2005 (November). *The Supply Chain Management Game for the 2006 Trading Agent Competition*. Tech. rept. CMU-ISRI-05-132. Carnegie Mellon University, Pittsburgh, PA.

Collins, John, Ketter, Wolfgang, & Gini, Maria. 2008 (March). Architectures for Agents in TAC SCM. *Pages 7–12 of: AAAI Spring Symposium on Architectures for Intelligent Theory-Based Agents*.

Court, A.T. 1939. Hedonic Price Indexes with Automotive Examples. *Pages 98–119 of: Dynamics of Automobile Demand*. New York: General Motors.

Cromwell, J.B., Hannan, M.J., Labys, W.C., & Terraza, M. 1994. *Multivariate Tests for Time Series Models*. Sage University Paper.

De Sarbo, W. S., Rao, V.R., Steckel, J.H., Wind, J., & Colombo, R. 1987. A friction Model for describing and forecasting price changes. *Marketing Science*, **6**(4), 299–319.

Deistler, M., & Hannan, E.J. 1988. *The Statistical Theory of Linear Systems*. John Wiley and Sons, Inc.

DEMPSTER, A. P., LAIRD, N. M., & RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Stat. Soc.*, **Series B, 39**(1), 1–38.

DIEBOLD, F. X., & PAULY, P. 1987. Structural changes and the combination of forecasts. *Journal of Forecasting*, **6**, 21–40.

DONG, LINGXIU, KOUVELIS, PANOS, & TIAN, ZHONGJUN. 2009. Dynamic Pricing and Inventory Control of Substitute Products. *Manufacturing & Service Operations Management*, **11**(April), 317–339.

DU PREEZ, JOHANN, & WITT, F. STEPHEN. 2003. Univariate versus multivariate time series forecasting: an application to international tourism demand. *International Journal of Forecasting*, **19**, 435–451.

DURBIN, J., & KOOPMAN, S. J. 2001. *Time Series Analysis by State Space Methods*. Oxford University Press, New York.

ENGLE, ROBERT, & WATSON, MARK. 1981. A One-Factor Multivariate Time Series Model of Metropolitan Wage Rates. *Journal of the American Statistical Association*, **76**(376), pp. 774–781.

ENGLE, ROBERT F., LILIEN, DAVID M., & WATSON, MARK. 1985. A Dymimic model of housing price determination. *Journal of Econometrics*, **28**, 307–326.

FOGLIATTO, F. S., & DA SILVEIRA, G.J.C. ET AL. 2011. *Mass Customization*. Springer.

FORNI, MARIO, HALLIN, MARC, & REICHLIN, LUCREZIA. 2000. The Generalized Dynamic Factor Model: Identification and Estimation. *Review of Economics and Statistics*, 540–554.

GANDAL, N. 1994. Hedonic price indexes for spreadsheets and an empirical test for network externalities. *the RAND Journal of Economics*, **25**(1), 160–170.

GATTORNA, J.L. 1998. *Strategic supply chain alignment: best practice in supply chain management*. Fourth edn. Gower.

GHAHRAMANI, Z., & HINTON, G.E. 2001. Variational learning for switching state-space models. *Neural Computation*, **12**(4), 831–864.

GOLUB, G.H., & VAN LOAN, C.F. 1996. *Matrix computations*. Third edn. The John Hopkins university press.

GORDON, R.J. 1990. *The measurement of durable goods prices*. University of Chicago Press.

GREWAL, M.S., & ANDREWS, A.P. 2008. *Kalman Filtering: theory and practice using Matlab*. Second edn. Wiley.

GUPTA, D., & BENJAAFAR, S. 2004. Make-to-order, make-to-stock, or delay product differentiation? A common framework for modeling and analysis. *IEE Transactions*, **36**, 529–546.

HAMILTON, J.D. 1994. *Time Series Analysis*. Princeton University Press.

HARHOFF, D., & MOCH, D. 1997. Price indexes for PC database software and the value of code compatibility. *Research Policy*, **26**(4-5), 509–520.

HARTMAN, R.S. 1987. Product quality and market efficiency: The effect of product recalls on resale prices and firm valuation. *The Review of Economics and Statistics*, 367–372.

HARVEY, A.C. 1989. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.

JUSELIUS, KATARINA, & HENDRY, DAVID F. 2000. *Explaining Cointegration Analysis: Part II*. Discussion Papers 00-20. University of Copenhagen. Department of Economics.

KELLERHALS, B.P. 2001. *Financial Pricing Models in Continuos Time and Kalman Filtering*. Springer: Lecture Notes in Economics and Mathematical Systems.

KETTER, WOLFGANG, COLLINS, JOHN, GINI, MARIA, GUPTA, ALOK, & SCHRATER, PAUL. 2006. Strategic Sales Management Guided By Economic Regimes. *In:* VAN HECK ET AL., ERIC (ed), *Edited Volume of the 2nd Smart Business Network Initiative Discovery Event*. Springer Verlag.

KETTER, WOLFGANG, COLLINS, JOHN, GINI, MARIA, GUPTA, ALOK, & SCHRATER, PAUL. 2008. *Tactical and Strategic Sales Management for Intelligent Agents Guided By Economic Regimes*. ERIM Working paper, reference number ERS-2008-061-LIS.

KETTER, WOLFGANG, COLLINS, JOHN, GINI, MARIA, GUPTA, ALOK, & SCHRATER, PAUL. 2009. Detecting and Forecasting Economic Regimes in Multi-Agent Automated Exchanges. *Decision Support Systems*, **47**(4), 307–318.

KIEKINTVELD, CHRISTOPHER, MILLER, JASON, JORDAN, PATRICK R., CALLENDER, LEE F., & WELLMAN, MICHAEL P. 2009. Forecasting market prices in a supply chain game. *Electron. Commer. Rec. Appl.*, **8**(March), 63–77.

Kotz, Samuel, & Nadarajah, Saralees. 2000. *Extreme value distributions: theory and applications*. Vol. 2ed. Imperial College Press.

Lancaster, K.J. 1966. A new approach to consumer theory. *The Journal of Political Economy*, **74**(2), 132.

Lancaster, K.J. 1990. The economics of product variety: a survey. *Marketing Science*, **9**(3), 189–206.

Lee, J, Cho, Y, & Lee, C. 2006. Forecasting future demand for large-screen television sets using conjoint analysis with diffusion model. *Technological Forecasting and Social Change*, **73**(4), 362–376.

Lei, Xu. 1998. Bayesian Ying-Yang Dimension Reduction and Determination. *Journal of Computational Intelligence in Finance*, **6**(5), 6–18.

Lutkepohl, Helmut. 2005. *New Introduction to Multiple Time Series Analysis*. Springer-Verlag Berlin.

Mardia, K.V., Kent, J.T., & Bibby, J.M. 1979. *Multivariate Analysis*. Academic Press of Elsevier Science.

Mazzocchi, Mario, Delle Monache, Davide, & Lobb, Alexandra E. 2010. A structural time series approach to modelling multiple and resurgent meat scares in Italy. *Applied Economics*, **38**(14), 1677–1688.

McLachlan, Geoffrey J., & Krishnan, Thriyambakam. 1997. *The EM algorithm and extensions*. Wiley series in probability and Statistics.

Meese, R., & Wallace, N. 2003. Nonparametric estimation of dynamic hedonic price models and the construction of residential housing price indices. *Real Estate Economics*, **19**(3), 308–332.

Meng, Xiao Li, & Rubin, Donald B. 1991. Using EM to obtain asymptotic Variance–Covariance Matrices: the SEM algorithm. *Journal of the American Statistical Association*, **86**(416), 899–909.

Muellbauer, J. 1974. Household Production Theory, Quality, and the "Hedonic Technique". *The American Economic Review*, **64**(6), 977–994.

Palmquist, R.B. 1980. Alternative techniques for developing real estate price indexes. *The Review of Economics and Statistics*, **62**(3), 442–448.

Pardoe, David, & Stone, Peter. 2009. Adapting Price Predictions in TAC SCM. *Pages 30–45 of:* Collins, John, Faratin, Peyman, Parsons, Simon, Rodriguez-Aguilar, Juan A., Sadeh, Norman M., Shehory, Onn, & Sklar, Elizabeth (eds), *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*. Lecture Notes in Business Information Processing. Springer.

Parkes, David C., & Kalagnanam, Jayant. 2005. Models for iterative multiattribute procurement auctions. *Management Science*, **51**(3), 435–451.

Reis, H.J., & Silva, S. 2006. Hedonic prices indexes for new passenger cars in Portugal (1997-2001). *Economic Modelling*, **23**(6), 890–908.

Sadeh, Norman, Arunachalam, Raghu, Eriksson, Joakim, Finne, Niclas, & Janson, Sverker. 2003. TAC-03: A supply-chain trading competition. *AI Magazine*, **24**(1), 92–94.

Schneider, T., & Neumaier, A. 2001. A Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. Math. Software*, **27**, 58–65.

Schoonbeek, L. 1986. On the Robustness of the Dominant Eigenvalue of Dynamic Linear Econometric Models. *Computer and Operatives Research*, **13**(1), 47–52.

Schreyer, P. 2002. Computer price indices and international growth and productivity comparisons. *Review of Income and Wealth*, **48**, 15–31.

Serfling, R.J. 1980. *Approximation theorems in mathematical statistics*. Wiley, New York.

Shumway, R.H., & Stoffer, D.S. 1982. An approach to Time Series Smoothing and Forecasting using the EM Algorithm. *Journal of Time Series Analysis*, **3**(4), 143.

Shumway, R.H., & Stoffer, D.S. 2006. *Time Series Analysis and Its Applications*. Springer Science+Business Media.

Simchi-Levi, D, Kaminsky, P, & Simchi-Levi, E. 2003. *Designing and Managing the Supply Chain*. Vol. 2ed. McGraw-Hill.

Simon, Dan. 2006. *Optimal State Estimation*. Wiley.

Song, Jing-Sheng, & Zhao, Yao. 2009. The Value of Component Commonality in a Dynamic Inventory System with Lead Times. *Manufacturing & Service Operations Management*, **11**(July), 493–508.

Stadtler, H., & Kilger, C. 2008. *Supply Chain Management and Advanced planning.* Springer.

Stock, J.H., & Watson, M.W. 2002. Forecasting using Principal Components from a large number of predictors. *Journal of the American statistical association*, **97**(460), 1167–1179.

Triplett, J.E. 2006. *Handbook on hedonic indexes and quality adjustments in price indexes: special application to information technology products.* Publications de l'OCDE.

Unwin, T. 1999. Hedonic price indexes and the qualities of wines. *Journal of Wine Research*, **10**(2), 95–104.

Van Dalen, J., & Bode, B. 2004. Quality-corrected price indices: The case of the Dutch new passenger car market, 1990–1999. *Applied Economics*, **36**(11), 1169–1197.

van Dalen, Jan, Ketter, Wolfgang, Lucchese, Gianfranco, & Collins, John. 2010. A Kalman Filter Approach to Analyze Multivariate Hedonic Pricing in Dynamic Supply-Chain Markets. *Pages 57–66 of: Twelfth International Conference on Electronic Commerce (ICEC 2010).* ACM.

van Dalen, Jan, Ketter, Wolfgang, Lucchese, Gianfranco, & Collins, John. 2011. Forecasting prices in dynamic heterogeneous product markets using multivariate prediction methods. *In: Thirteenth International Conference on Electronic Commerce (ICEC 2011).* ACM.

Verhaegen, H.R., & Van Dooren, P. 1986. Numerical aspects of different Kalman filter implementations. *IEEE Transaction on Automatic Control*, **AC**(31), 907–917.

Wang, Zhaowen, Kuruoglu, Ercan E., Yang, Xiaokang, Xu, Yi, & Huang, Thomas S. 2011. Time Varying Dynamic Bayesian Network for Nonstationary Events Modeling and Online Inference. *IEEE Transactions on Signal Processing*, **59**(4), 1553–1568.

Watson, M.W., & Engle, R.F. 1983. Alternative Algorithms for the Estimation of Dynamic Factor, MIMIC and Varying Coefficient Regression Models. *Journal of Econometrics*, **23**, 385–400.

Wazed, M.A., Ahmed, S., & Yusoff, N. 2008. Commonality Models in Manufacturing Resources Planning: State–of–the–art and future directions. *European Journal of Scientific Research*, **23**(3), 421–435.

Wu, C. F. Jeff. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics*, **11**(1), 95–103.

Summary in English

Over the years, expert systems for decision processes in supply chain management and electronic commerce have experienced significant growth from the point of view of mathematical complexity. The reasons for this growth are the increasing of technological instruments for supporting the agents, and the low costs of hardware and software for companies and customers. Thus, an increasing of electronic negotiations was reported, a cause and an effect of the artificial intelligence underlying the innovative selling methodologies. An example is given by recommendation systems appeared in many websites. The multivariate statistic analysis of the customer preferences is the methodology for clustering the consuming patterns. In this way the software agents can offer a dedicated product to the customers.

First contribution of the thesis is the dynamic modeling of a hedonic process, as described in Chapter 2. The customer evaluation of single parts included in a good is assumed homogeneous in the market, like in the Lancaster model. In our model, hedonic price, or implicit price, for a component (e.g. the CPU of a home computer) is considered the average price in the market that a customer is available to spend for acquire the part/characteristic of the good. The customer evaluation of the part can not be clearly declared. In fact, it is usually latent and it can be evaluated only in the moment of the payment.

The concept of hedonic price is back in the literature when researchers proposed the quality adjustment based on implicit prices. Several economists proposed the estimation of the technological goods not more present in the market through hedonic regression. The hedonic estimation refers to single periods and products. This procedure for the construction of a price index, is named "quality adjustment". The latter was criticized and nowadays it is rarely applied (e.g. technological goods like computers, compact discs, and appliances) in few countries (e.g. US, Great Britain, Australia). In our hedonic model we extend the estimation methodology in the dynamic sense and taking into account the inter-dependencies between products and components. The estimation of the hedonic prices is advantageous for manufacturers (assemblers) for the decisions about product variety selection. In that case, they can select the best product to assemble for the consumer market. It is well known that a large product variety accomplishes for reaching more customer tastes with respect to small one. But the costs increase with the dimension of product variety. Optimization problems in the production planning, inventory management (Make-to-Order vs Make-to-Stock), postponement strategies, and scheduling, could be implemented considering hedonic prices in their formulations. The extraction of hedonic prices in a dynamic-multivariate system is linked to the estimation of the multiple parameters (hyperparameter) underlying the process like in a state-space model. The methodology is different from that used in

state-space models, which is based on the multivariate likelihood of the process. It consists in the second contribution of the thesis: the identification of the parameters in a state-space model for multivariate high dimensioned processes, both in input (n), and state variables (m). The conventional methodology was applied for bivariate state vectors and our extension represents an experimental benchmark for high dimension cases. The entire procedure for estimation of hedonic prices consists in two steps: the "Kalman filter-smoother" (KF) estimates the hedonic prices, and the "Expectation-Maximization" (EM) algorithm identifies the hyperparameter. Iterating the two procedures until the convergence has reached, the close-to-optimal solution is obtained. The more the estimation of parameters is close-to-optimal, the more the hedonic evaluations are meaningful in the market. In our methodology the stopping rule of the KF+EM is given by the absolute difference between the two transition matrices in close iterations. In high-dimensioned systems, when n and m are greater than two or three, that stopping rule outperforms the likelihood ratio test. The inconvenient is the behavior of the Kalman filter in the nearing of the solution. The problem may become ill conditioned if the Kalman gain is small and rapidly diverge from the close-to-optimal solution. The disadvantages of our technique refers also to the elongation of the computation times. But the augmentation of performances leads us to the use of our stopping rule.

Chapter 3 shows a set of alternative models for the base one. The "Noise model", which does not include the estimation of parameters, assumes stable and non Markovian the hedonic prices. The dynamic multipliers are all unitary in this case, and the transition matrix coincides with the identity matrix. Thus, the volatility of prices is explained only by disturbance in the markets. If we assume independent hedonic prices, thus a diagonal transition matrix, the dynamic multipliers represent the trend for the individual components in the customer market. The latter can be estimated from the historical data or supposed decreasing (increasing) according to the customer tastes. Other hedonic formulations are the lagged model and the premium model.

Chapter 4 analyzes the algorithm for the estimation of parameters in the base model, the *Algorithm 1*. The goal is the real time estimation of hedonic prices, hence a measure of the goodness of the estimation in an on line context. Two algorithms are designed: the first one with a variable number of iterations, whereas the second has a fixed number. A calibration is reached both for time of computations and precision. The algorithms will be tested in the chapter 5 in a framework for the extraction of hedonic prices, the identification of hyperparameter, and the forecasting of product prices in a supply chain. The framework includes also conventional autoregressive models, both univariate and multivariate. Coefficients for hedonic and standard forecast models are estimated day by day, and a combination model use them for a prediction (contribute 5).

Finally, chapter 5 outlines the forecast models for similar environments. Starting from AR and VAR models, the predictions for the next h (forty) days are given. In this way, an agent-manufacturer can negotiate with suppliers for resources and to optimize the production. Hedonic and standard predictions are combined in a real time forecast model. The latter increases performances and requires few seconds for an output.

The analysis of the stopping rule for the procedure is the larger contribute in the thesis. It can be extended in similar state space models for large values of n and m. Future works are focused on the detection of turning (break) points and regimes in a multivariate process with state variables. Furthermore, forecast methodology may be used as selector of the centroid hyper-parameters for regimes. The more performances are good for a hyper-parameter, the more that hyper-parameter must be included in the selection for centroids.

Sommario in lingua italiana

Negli ultimi anni, abbiamo assistito ad una crescita di sistemi esperti e algoritmi per i processi decisionali applicati alla gestione delle catene di rifornimento e nel commercio elettronico. Questa crescita riguarda anche la complessità di dette procedure, sia dal punto di vista tecnologico che matematico. Le ragioni per cui tale crescita è avvenuta risiedono nella diffusione di strumenti informatici nei processi tecnologici che coadiuvano sempre di più le scelte degli amministratori. Inoltre sulla crescita di strumenti di intelligenza artificiale ha influito sicuramente la diffusione a basso costo di hardware e software nelle case e negli uffici. Ciò ha portato anche alla crescita delle negoziazioni commerciali di tipo elettronico. Un tipico esempio è dato dai sistemi di raccomandazione per l'utente inseriti oramai in qualsiasi sito Internet per la vendita di prodotti. L'analisi multivariata delle preferenze del cliente permette di creare dei pattern di consumo e delle tipologie di utente, che a loro volta suggeriscono cosa acquistare o offrono promozioni.

Il primo contributo della tesi consiste nella modellizzazione dinamica di un processo edonico, descritta nel capitolo due. La valutazione da parte del cliente delle singole parti che compongono un prodotto può ipotizzarsi omogenea nel mercato come nel modello di Lancaster. Ovviamento il prezzo edonico, o implicito, di una componente (ad esempio la CPU di un computer) deve considerarsi come il prezzo equo nel mercato che un cliente è disposto a pagare per ottenere la singola componente/caratteristica. La valutazione del cliente di questa componente non necessariamente avviene nella realtà in modo esplicito. Spesso essa è di natura intrinseca e diventa manifesta solo al momento dell'acquisto. Qui il cliente è disposto a pagare un prezzo per una somma di caratteristiche collegate alle parti del prodotto.

L'utilità del prezzo edonico è tornata in auge quando è stata proposto l'utilizzo dello stesso nel calcolo dell'indice dei prezzi per particolari beni di natura tecnologica. Alcuni economisti proposero di stimare le variazioni dei prezzi, per quei prodotti non più presenti nel mercato, tramite una correzione degli stessi che prende il nome di "quality adjustment". La tecnica è appunto basata sulla stima dei prezzi impliciti mediante una semplice regressione di natura univariata e statica. Essa ha sollevato molti dubbi e perplessità, ed è tuttoggi applicata solamente in sporadici casi (beni tecnologici ad alta innovazione, i quali si rinnovano con velocità maggiore rispetto ad altri beni) ed in pochi paesi (tra i più importanti USA, Australia, Gran Bretagna). Nel nostro modello noi estendiamo le relazioni fra componenti e prodotti sia in ambito multivariato che considerando la dinamica nel tempo degli stessi.

L'analisi dei prezzi edonici risulta quindi molto utile ai singoli produttori (o sarebbe meglio dire assemblatori) per decidere quale prodotto della varietà prodotta risulta più apprezzato. È ovvio infatti che una produzione variegata riesce spesso a soddisfare molteplici bisogni ma con costi non del tutto trascurabili. Problemi di ottimo legati all'assemblaggio su ordinazione (Make-To-Order), alla posticipazione dell'assemblaggio come nel caso delle vernici, e alle politiche di magazzino per parti e componenti, sono solo alcuni fra i principali che il modello edonico potrebbe aiutare ad ottimizzare.

La ricerca di un modello edonico in ambito dinamico e multivariato porta alla stima dei parametri sottostanti il sistema. La metodologia è diversa da quella consueta basata sulla verosimiglianza della serie temporale dei prezzi dei prodotti ed è oggetto del contributo numero due: l'identificazione dei parametri in un modello stato-spazio, "state-space model", nel caso multivariato sia per le variabili di input, vettori di dimensione n, che per gli stati, vettori di dimensione m. La tecnica da noi scelta è già stata adottata in ambito bivariato da Shumway e Stoffer. Il nostro contributo lo studio per il caso di dimensioni elevate di questi vettori. L'intera metodologia prevede due fasi: la stima dei prezzi edonici tramite il "Kalman filter-smoother", e l'identificazione dei parametri tramite la massimizzazione del valore atteso della verosimiglianza, ovvero l'algoritmo Expectation-Maximization (EM). Iterando più volte questo procedimento si arriva alla convergenza verso la soluzione migliore. Più ci si avvicina alla stima migliore dei parametri e più i prezzi edonici corrisponderanno a quelli effettivi del mercato. Nel nostro caso individuiamo come regola di arresto della procedura, la vicinanza fra le due matrici di transizione ottenute in due iterazioni contigue. Essa, nel caso in cui sia n che m sono valori relativamente alti, si comporta meglio che la regola basata sul test del rapporto di verosimiglianza che usualmente viene scelto allo scopo. Il problema però può diventare mal condizionato se ci avviciniamo troppo alla soluzione del Kalman filter. Quindi parametri e prezzi edonici possono esser stimati non correttamente dalla regola di arresto da noi introdotta, e inoltre ritardare i tempi di stima.

Il capitolo tre provvede una serie di alternative per il modello edonico base analizzato nel capitolo precedente. Il 'Noise model', il modello che non prevede la stima dei parametri, assume stabili e non markoviani (quindi unitari i moltiplicatori dinamici) i prezzi edonici durante l'intera vita dei prodotti. In questo caso la matrice di transizione è una matrice identità, e la variabilità dei prezzi dei prodotti è spiegata solamente dai residui e quindi dalla volatilità del mercato. Se invece, supponiamo l'andamento dei prezzi edonici non correlato e decrescente nel tempo ci ritroviamo in un modello in cui la matrice di transizione assume l'aspetto di una matrice diagonale con elementi diversi da uno. Il cliente valuta l'utilità della singola componente decrescente nel tempo (valore di poco minore di uno), costante (valore unitario) o crescente (valore di poco maggiore di uno). In questo modo otteniamo il "Diagonal model" che, come il "Noise model" semplifica l'algoritmo alla sola stima dei prezzi edonici, evitando il problema pi difficile dell'identificazione dei parametri.

Questi ultimi infatti possono venir stimati o supposti noti. Altre estensioni per un modello edonico sono incluse nel capitolo tre, e riguardano un modello di stima dei parametri con lag maggiore di uno, e da un modello che considera anche i premi associati ai singoli prodotti e non solo i prezzi edonici.

Il capitolo quattro esplora in profondità cosa avviene nel caso dell'algoritmo principale, con regola di arresto data dalla differenza delle due matrici di transizione, l'*Algorithm 1*. L'obiettivo è la stima dei parametri in tempo reale, ovvero la formulazione di una regola di decisione della bontà di stima. Si ottengono due algoritmi differenti: il primo con un numero di iterazioni variabile, il secondo con un numero fisso. Si arriva così ad una corretta calibrazione dell'algoritmo sia per ottimizzare l'errore di previsione, che i tempi di stima. Entrambi verranno impiegati nel capitolo cinque in un software per la stima dei prezzi edonici, dei parametri e la previsione dei prezzi in una catena di approvvigionamento con componenti e prodotti. Tale software adotta modelli autoregressivi standard, sia univariati che multivariati. I loro parametri vengono stimati anche essi in tempo reale. Le stime fornite da tutti i singoli modelli analizzati vengono adottate per la costruzione di un "forecast combination model" (contributo 5).

Infine il capitolo cinque espone i modelli che possono venire utilizzati per seguire giorno dopo giorno la previsione di prezzi in un mercato con prodotti e componenti. Si parte dai modelli autoregressivi convenzionali, come l'AR ed il VAR. In questo caso si fornisce la metodologia per la stima dei loro coefficienti e delle previsioni relative agli h giorni futuri. Si arriva a considerare previsioni a lungo termine, fino a quaranta giorni in avanti. In questo modo l'agente-assemblatore può stipulare contratti e coordinare la produzione anche in base ai prezzi futuri previsti. La costruzione di un modello che sfrutta la combinazione delle previsioni è basata sulla stima di pesi delle singole performance. Esso non solo aumenta la precisione delle stime ma influisce molto poco in termine di tempi di calcolo.

Ovviamente il contributo più importante della tesi è dato dall'introduzione di un criterio di convergenza per l'algoritmo che fonde il *Kalman filter* e la procedura *Expectation-Maximization*. Ed in questo senso i nostri sforzi futuri saranno concentrati, sia estendendo il modello con l'inclusione di "break/turning points", sia individuando dei regimi nel processo, sia formalizzando l'accettazione dei parametri sottostanti il processo in base alla prestazione fornita nella previsione dei dati futuri. Abbiamo già sperimentato infatti che una selezione dei parametri stimati possa avvenire proprio grazie ad una analisi delle prestazioni degli stessi.

Vita

Gianfranco Lucchese 1970) was born in Rome, Italy. After getting his higher secondary school diploma, he started the study of Statistics and Demography at "La Sapienza", the oldest University of Rome. In the mean time he worked as gardener and dedicated a lot of time to the learning of the music (bass player and sound engineer). After two albums he abandoned definitely the world of music. At only four exams to the thesis, in the middle of 1999, he started to work for the secretarial staff of the President of Italian Republic as gardener. In fact, the Quirinale, the historical building in the center of Rome, where Italian President lives and its staff works, includes an ancient garden (dated 1490 a.C.) which requires a deep knowledge of the history of arts, and gardening techniques. In the four hectare Renaissance garden, Gianfranco learned the base of turf and greenhouse management, the art of floral adornment, and he specialized as garden guide for President guests. He provides interpretative services to Quirinale visitors through guided tours and special workshops for children, associations, and people with disabilities. While its creative, imaginative, and enthusiastic job, after eight years, he want to conclude his graduated studies under the direction of Prof. Enzo Orsingher, one of the most important luminary in Theory of Probability in Italy. He was fascinated by the world of the research. In 2008, he applied for a Ph.D course in 2009 at Bergamo, in Mathematics, Statistics, and Computer Science. Here, he spent 18 months at the Rotterdam School of Management as visiting Ph. D. student. Recently, works experiences in Statistics are focused in data analysis and modeling.

Since February 2012, he collaborates with University of Brescia, at Center for Multi Sectoral Services and Innovation Management, and with University of Bergamo, in the field of the Microfinance.

Permanent address: Via G. da Campione, 22
24124 Bergamo, Italy
glstatistica@gmail.com
http://www.unibg.it/struttura/struttura.asp?cerca=r_dmsia

This dissertation was typeset with LaTeX[1] by the author.

The code for the algorithms and tests was written in MATLAB@ by author.

[1] LaTeXis a document preparation system developed by Leslie Lamport as as a special version of Donald Knuth's TeXProgram