Ollscoil na hÉireann, Corcaigh

**National University of Ireland, Cork**

University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

# Application of Cointegrated State-Space Models to Financial and Economic Data

Thesis presented by

**Miaad Hamad Al-qurashi**, BSc, MA

for the degree of

**Doctor of Mathematics**

**University College Cork**

**School of Sciences, Engineering, and Food Science**
**Dept. Mathematics**

Head of Department: Prof. Bernard Hanzon

Supervisor: Prof Bernard Hanzon

2021

# Contents

# List of Figures

# List of Tables

This is to certify that the work I am submitting is my own and has not been submitted for another degree, either at University College Cork or elsewhere. All external references and sources are clearly acknowledged and identified within the contents. I have read and understood the regulations of University College Cork concerning plagiarism

*Miaad Hamad Al-qurashi*

# Abstract

In the dynamic stochastic modeling of financial and economic time series, the concept of cointegration plays an important role. It refers to the existence of long-term equilibrium relations between variables in a dynamic environment. The cointegration theory posits that, in a non-stationary environment, the long-term equilibrium relations will show up as stationary relations between certain variables. In the context of linear models, this translates into the existence of so-called *cointegrating linear relations* and corresponding *cointegrating vectors*.

The error correction model was introduced by Engle and Granger (1987) for estimating models exhibiting cointegration. However, it was not until Johansen (1988, 1991) that a rigorous methodology was proposed for estimation and analysis of cointegrated models in a multivariate dynamic setting. The iconic research by Johansen sets out the Vector Error Correction Model (VECM) approach to cointegration, by applying maximum likelihood estimation. The VECM is based on the Vector Autoregressive (VAR) model. The VAR model has several shortcomings for cointegration studies. Several authors have explored estimation of cointegration using VARMA and state-space models, e.g. Ribarits and Hanzon (2014b,a), Yap and Reinsel (1995), Lütkepohl and Claessen (1997), Poskitt (1994, 2006), Bauer and Wagner (2002), Kascha and Trenkler (2011), among others.

This research study focuses on the discrete-time state-space model as well as on a continuous-discrete time state-space model. This research study proposes a parameterization and an estimation method that translates the cointegration property into a low-rank constraint on a resulting likelihood optimization. First, the study partially optimizes the likelihood function. The resulting criterion is a function of eigenvalues of a matrix due to the low-rank constraint. This raises the challenge of calculating the derivatives of the parameterized matrix eigenvalues. The research study addresses this problem by applying the envelope theorem. The study applies a gradient method to optimize the remaining model parameters using a new parametrization. This parameterization has bounded and numerically stable parameters. The method is illustrated by some simulated examples and examples involving stock price index data, oil price data, and more. Comparisons are made against a classical VECM, and in many cases, the state-space model yields better results.

In the name of Allah, Most Gracious, Most Merciful.

# Acknowledgements

I want to take this moment as an opportunity to recognize the invaluable assistance that you all provided during my study. I would like to thank all those exemplary human beings, whom I have come across during this journey, whose lives have inspired me and from whom I have learned to live life.

Firstly, I want to express my deepest gratitude to my supervisor Prof. Bernard Hanzon for the continuous support of my PhD. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me throughout my research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD. study. My sincere thanks also goes to prof. Niall O'Sullivan who has provided me a helpful data, and give me guidance especially while my supervisor on Sabbatical.

I want to express my appreciation to the Saudi ministry of education for the scholarship opportunity and financial support. I would also like to say a heartfelt thank you to the Saudi embassy in Ireland for helping in all the ways to take care of me during this challenging period.

I cannot forget the contribution of my sister and friend, Cidalia who has been a constant reservoir of cheers at all those times when I needed it the most. It's hard to describe the importance of Cidalia in my life. She has been a source of constant support and inspiration and her confidence on me always boosted my morale in all possible ways.

Last but not least, my deepest gratitude goes to my beloved parents and also to my sisters and brother for their endless love, prayers, and encouragement, They are the ones who have made this thesis a reality by having faith in me and for instilling in me the courage to strive and achieve higher goals in life. Also not forgetting my lovely husband; Fahad who has been by my side throughout this PhD., living every single minute of it, and without whom, I would not have had the courage to embark on this journey in the first place and making it possible for me to complete what I started. Grateful thanks to him for his love and care, for always believing in me and encouraging me to follow my dreams, for his invaluable advice and feedback on my research and for always being so supportive of my work and for helping me enormously, especially with the mammoth task of doing the final formatting and printing of this thesis, I love you so much. To those who indirectly contributed to this research, your kindness means a lot to me. Thank you very much.

# Introduction

*Time series analysis* is a sub-field in statistics with applications in several fields such as economics, finance, engineering, natural sciences, climate sciences, and many others. However, many concepts that lie at the core of the recent literature have been developed by applied researchers who have aimed to address their field's specific practical needs.

Over the last three decades, the concept of *cointegration* has emerged as an essential concept emerging in economics and finance. It put the following idea forwards: in a non-stationary environment, the long-term equilibrium relations will show up as stationary relations between certain variables, leading in the context of linear dynamical models to the existence of *cointegrating linear relations* and *cointegrating vectors*.

Listed below are but a few of many instances of cointegration between various economic and financial variables.

- The Permanent Income Hypothesis (PIH) implies cointegration between consumption and income.
- Money demand models imply cointegration between interest rates, nominal income, and prices (Friedman and Kuttner 1992).
- Growth theory models imply cointegration between income, consumption, and investment.
- The Fisher equation implies cointegration between nominal interest rates and inflation.
- Purchasing Power Parity (PPP) implies cointegration between nominal exchange rate and, domestic, and foreign prices.

Econometric time series are often characterized by non-stationarity, such as trends in mean and variance. While individual time series might show these patterns, the economic theory might suggest static long-run equilibrium relationships between the variables under consideration. The cointegration theory's main idea is that although the long-run relationship is probably not satisfied at every point of time, the potential deviation from the equilibrium relationship is small and stationary, compared to non-stationary individual variables.

Since as early as the mid-1920s, statisticians have investigated the existence of stochastic trends and cycles and their detrimental effects on the regression analysis of time series (see Yule 1926). However, it was not until the mid-1970s when interest in the subject experienced a revival on academic grounds when Granger and Newbold (1974) did further work on the concept of *spurious regression* which had already been put forward by Yule (1926). This definition refers to the case where ordinary least squares estimates point to a linear relationship that does not hold and is driven by the non-stationarity of the variables. There is no universal way to address this problem, and cointegration methodologies emerged as one of the potential solutions. In the early 1980s, the academic discussion was complemented by the debate of trend-stationarity versus unit root non-stationarity, mainly raised by Nelson and Plosser (1982).

The early literature on spurious regression and the scientific need to develop methods to analyze non-stationary time series, paved the way for the groundbreaking research in cointegration theory by Granger (1981) and especially by Engle and Granger (1987). As documented by Boswijk et al. (2010), Engle and Granger (1987) was already the best cited time series econometric paper of all time by 2010; moreover, the relevance of their scientific contribution is well put into historical context by Boswijk et al. (2010).

Although the methodology by Engle and Granger (1987) has continued to gain prominence both in academic and policy spheres as the *two-step approach to cointegration*, it is only applicable to bi-variate time series. A related methodology emerged within the same research agenda: the *multivariate approach to cointegration*. This second approach was developed originally by Johansen (1988, 1991) whose work allowed researchers to study cointegration phenomena within a multivariate framework. More specifically, Johansen's approach can be categorized as a special case within the general model class of vector autoregressive models (VAR) that were popularized in macro-economic by Sims (1980), and is known in the literature as the vector error correction model (VECM). Three decades after their origin, both the *two-step* and the *multivariate* approaches have remained as the most widely used methodologies in dealing with integrated time series. The two-step approach falls into the category of non-parametric modeling. Contributions pursuing this approach include Bierens (1997), Park and Phillips (1988, 1989), Phillips and Hansen (1990), Saikkonen (1991), Sims et al. (1990), Stock and Watson (1988), Poskitt (1994). The multivariate approach falls into the category of parametric modelling, and although Johansen (1988, 1991) had the most prominent contribution with his VECM, other authors also contributed to this strand by considering Vector-Autoregressive Moving Average (VARMA) frameworks, e.g. Yap and Reinsel (1995), Lütkepohl and Claessen (1997), and Poskitt (2006). The difference between both types of analysis is that the non-parametric one tests and estimates the cointegrating vectors while treating all the remaining characteristics of the data generating process as nuisance parameters, while the parametric approach models cointegration within a fully specified model class. A feasible way of modeling economic and financial non-stationary series is using integrated processes: A $p$ dimensional process $(y_t \mid t \in \mathbb{N})$ is called *integrated of order* $k$ or $I(k)$ if $\Delta^k y_t = (1-z)^k y_t$ is the smallest number for which $k$ is stationary and $k = (0, 1.2, \ldots)$, where $z$ denotes the lag operator, i.e. $z(y_t \mid t \in \mathbb{N}) = (0, y_1, y_2, \ldots)$. In the series, an integrated process of order one appears to be most common.

The VAR approach is still the most widely used methodology in dealing with cointegrated $I(1)$-processes. Nonetheless, there is a potential challenge in applying the state-space cointegration framework as developed e.g., by Bauer and Wagner (2002), and Ribarits and Hanzon (2014b,a). Designed as a generalisation of the VAR approach developed by Johansen to a state-space setting, it holds many advantages over the more widely used VAR approach. Just a few of the advantages of the state-space approach over it's VAR counterpart mentioned by Ribarits and Hanzon (2014b,a) include.

- State-space models offer a larger and more flexible model class, containing VAR models as special cases.
- If the data generating process (DGP) is a VAR process, but one does not observe all the output components, then the resulting model can, generally speaking, no longer be described by a VAR model, but can still be described by a linear state-space model.
- State-space models are heavily used in continuous time modeling, especially in financial applications.
- If the DGP has a fixed time delay $t_{\max}$ and if observations are taken at frequency $N$ per unit of time, the maximal delay in the VAR equation will be

$t_{\max} \times N$. Therefore if $N$ is large, then the VAR model will have a large number of parameters (as the number of parameters grows with $p^2 \times t_{\max} \times N$) and will not be parsimonious process. The state-space model does not have this problem.

- The state-space model can be formulated in discrete-time, continuous-time and continuous-discrete time, where the latter refers to a continuous-time model that is observed at discrete points in time. The continuous-discrete formulation also allows for irregularly timed observations, which would appear to be relevant in high frequency financial markets.

Furthermore, if the model has a large value of $p$, then it could result in default values in VAR, as shown by Tschernig et al. (2013). This is because there are too many parameters involved and the model will provide an empirical average which deviates from the expected value. Hence, using a state-space model will provide an average closer to the expected value when compared against VAR models. In the context of present research study, it may be good to note that ARMA and ARIMA models are equivalent to the state-space models, every ARMA or ARIMA model can be rewritten as a state-space model and every state-space model can be rewritten as an ARMA or ARIMA model (see e.g. Hartl and Weigand 2018). Usage of state space models in econometrics has a longer history and can be traced back to Yap and Reinsel (1995), Lütkepohl and Claessen (1997), Aoki and Havenner (1991), Hannan and Deistler (1988), Heij et al. (1997), Harvey (1989), among others. Notwithstanding, it should be pointed out that the recent approach Ribarits and Hanzon (2014b,a) embodies one of the most recent the developments in the area of state-space cointegration modeling. Our study contributes to the literature as it formulates a discrete time state-space model and a continuous-discrete time state-space model that embodies cointegration.

As done by Ribarits and Hanzon (2006), our study presents a generalization of VAR cointegration to the state-space framework for cointegration. The mathematical pillars of the maximum likelihood estimator of VECM models proposed by Johansen (1988, 1991) can be summarized as follows:

- First, the likelihood maximization problem is reduced to a low-rank regression problem by *concentrating out* several linear parameters in the vector autoregressive representation.
- Second, the low-rank regression problem is solved.

- Third, following the concentration step, we will find the optimum of a quotient of determinants of matrices that are quadratic in terms of an unknown matrix $\beta$. This optimum is then obtained through the solution of a *generalized* eigenvalue-eigenvector problem.

Here, we present a state-space error correction model approach that is an alternative to the celebrated VECM, under the following analogous ideas:

- First, two of the three matrices in the state-space representation are fixed so that the constrained likelihood maximization can be reduced to a low-rank regression by concentrating out the third state-space matrix. These fixed matrices are then optimized.
- Second, the low-rank regression problem is solved.
- Third, after the concentration step, it is needed to find the optimum of a quotient of determinants of matrices that are quadratic in terms of an unknown matrix $\beta$ (similar to the VECM approach listed previously), which needs to be determined so that an optimum is reached through the solution of a *reduced rank* eigenvalue-eigenvector problem.
- Fourth, we will repeat the steps to obtain a better fit for our parameter as it will have many starting points.

Contrary to the VECM case, substituting the optimal parameter values (in the previous 4 steps) into the likelihood function generates a criterion function that still depends on some unknown parameters. The number of parameters is much smaller. While Ribarits and Hanzon (2014a) used a local parameterization technique known as *DDLC*, however, we will use a different approach by applying a generalized Schur parameters technique. In that sense, our work deviates from the original procedure by Ribarits and Hanzon (2014a) in order to assure the following advantages: a finite interval for each parameter, a numerically stable parameterization and a reduced number of parameters. This is achieved by using input-normal sub-diagonal pivot structures (SDPS) and parameterizing these using angles, which help us to greatly simplify the derivatives and find a starting point for gradient search. Furthermore, the approach followed in this research corrects an error in Ribarits and Hanzon (2014a); see the second chapter for further details.

It is important to remark that the calculation of the gradient of the concentrated likelihood function (if it exists), can be done by calculating the gradient of the likelihood function before *concentration* at the parameter point that emerges from the concentration step. Furthermore, even if the concentrated likelihood function cannot be differentiated, the *adapted gradient-based search algorithm* (proposed by Ribarits and Hanzon (2014a)) which operates only on the remaining (after concentration) free parameters, is ensured to avoid getting stuck at any point other than a critical point of the original likelihood function. Consequently, the local optimum of the likelihood function can be found by the use of a gradient-type search method. The main implication is of this is that, at least when using sufficiently many well dispersed starting points, the maximum likelihood estimator can be found.

The contents of this thesis are arranged as follows:

In chapter 1, we will review the vector autoregression models and cointegration to a deeper extent. This chapter provides some background reading to the concepts of multivariate time series analysis, which are central to the research question at hand; such concepts also contextualize the cointegration methodology's scientific relevance.

In chapter 2, we will propose a parameterization and estimation method that maps the concept of cointegration into a low-rank constraint. The difficulties of calculating the derivatives of the parameterized matrix eigenvalues are addressed by applying the envelope theorem. After that, a gradient method (gradient search algorithm) is derived to optimize the remaining parameters of the model. Concretely, this chapter presents the *cointegrated discrete-time state-space error correction model* and a corresponding maximum likelihood estimation procedure.

In chapter 3, we will apply a similar approach like the one outlined above but now applied to a continuous-time state-space model with a finite number of observations taken at irregularly spaced time points. We propose a parameterization and estimation method using a continuous state-space model which translates cointegration dynamics into a low-rank constraint on a resulting likelihood optimization. This approach utilizes parameters as having a particular value for potentially only an infinitesimally short amount of time. Between any two points in time there are an infinite number of other points in time. Thus time is viewed as a continuous variable. This approach's essential advantage lies in the explicit incorporation of the low-rank requirement while addressing the issue of eigenvalue derivative through the application of the envelope theorem.

We need to make some approximations, which make that the method is mainly suitable for high frequency.

In chapter 4, we will describe the way in which the model is simulated using the parameter values. The chapter will assess the simulation study to compare the performance of the state-space error correction model (SSECM) with that of the vector error correction model (VECM). First, we will assess a discrete-time in both cases, and we will evaluate them based on several statistical tests such as information criteria, the goodness of fit, forecast accuracy (RMSE) and Bayes factor, it turns out that the SSECM outperforms the VECM. Subsequently, we will assess a simulation study in continuous-time where we will perform a comparison between the state-space framework and the actual model, as neither VAR nor VECM methodologies have been extended to continuous-time applications.

In chapter 5, we will perform an application of the method to three data-sets: individual stock prices of six companies, three crude oil prices, and stock prices of six geographical areas. This exercise allows us to inspect the cointegration analysis results under the standard VECM approach and the SSECM. In several cases, the latter yields better results, particularly in relation to the number of parameters and cointegrating vectors in high-frequency data.

Our study contributes to the literature by presenting a new parameterization method for the maximum likelihood estimation of SSECM models. It allows for cointegration. In addition to the theoretical development, the estimation procedure is applied to three data sets and compared against the VECM. Although the discrepancies regarding the number of cointegration relationships are not drastic between both methods, the SSECM had lower information criterion values and it reduced the number of parameters by half (compared to the VECM) in all cases. It is also shown that the state-space framework has several advantages in terms of model selection indicators and is particularly preferable over the VECM when the number of $p$ is large. Furthermore, the state-space approach allows for continuous-discrete simulation and offers more flexibility to develop further research.

# Chapter 1

# Cointegration: Overview of Models and Development

This section aims to review the underlying fundamentals, estimation, and limitations behind VAR models and VECM models. Also, the section will highlight the theoretical aspects of the vector time series model and cointegration analysis.

## 1.1 Economic Models

Following Wooldridge (2012) and Greene (2003), *econometrics* could be broadly defined as a field that combines economic theory, statistics, and mathematics. Although it is not a requirement, most econometric analyses depart from an *economic model* that guides the underlying empirical strategy (i.e., the period of study, variable selection, a qualitative statement about causal effects, and others). According to Wooldridge (2012), an economic model consists of a mathematical equation that describes relationships; academic economists are well-known for their model building efforts to specify a vast number of economic and social phenomena. These models can be thought of as a formal statement of a theoretical proposition. Nonetheless, they are also an implication of reality and by no means can encompass all the random aspects of economic phenomena. Consequently, in addition to economic variables, they often incorporate stochastic elements to account for the randomness.

Microeconomics consists of the analysis of cross-sectional and panel data, focusing on individual consumers, firms, and decision-makers. In contrast, macroeconomics is involved with the analysis of time series data of aggregated data

(Greene 2003). This particular research will focus on time series. Time series data analysis vary depending on the variables; for instance, some financial data sets have missing variables values such as exchange rates, or stock prices. These are designated *high frequency data*. Mathematically, data analysis for high frequency data sets will depend on the extent of relationships between deterministic variables (a deterministic variable is one whose value is known with certainty.) However, some of these variables are not deterministic, a time-series data with missing value is analysed by using stochastic dynamic model.

The context of this research is one of time series analysis. Most data in macroeconomics and finance come in the form of time series, which means researchers have to evaluate a set of repeated observations of the same variable, such as a stock return or oil prices in our context.

## 1.2 Time Series Regression

The field of time series econometrics has evolved as a sub-field of econometrics with its own vast body of literature and numerous applications in the areas of macroeconomics and finance. As expected, it also draws on many elements from traditional time series analysis coming from other disciplines such as statistics and engineering Hannan and Deistler (1988).

The time series econometrics domain has gained so much depth that econometrics textbooks such as Greene (2003), Wooldridge (2012), among others, only cover the fundamental elements of time series regression. The vast number of model classes in this area require dedicated books such as Lütkepohl (2005) or Lütkepohl and Krätzig (2004). A more recent approach to the same problem uses additive models more familiar to statisticians. In this approach, the observed data are assumed to result from sums of series, each with a specified time series structure; for example, in economics, assume a series is generated as the sum of trend, a seasonal effect, and error. The state-space model that results is then treated by making judicious use of the celebrated Kalman filters, developed originally for estimation and control in space applications, see Boswijk et al. (2010). This section will briefly describe the essential elements of time series models, but for a more in-depth introduction, the reader should consult the specialized literature.

Following Greene (2003), a standard time series model explains a variable $y_t$ in terms of contemporaneous (or lagged) dependent variables $x_t$, and could potentially include lagged values of $y_t$ as a regressor. The disturbances or innovations (i.e. model residuals) are denoted by $\varepsilon_t$. For example, a linear time series with one scalar explanatory variable $x_t$ and an autoregression with maximal delay one is of the form:

$$y_t = \beta_1 + \beta_2 x_t + \beta_3 y_{t-1} + \varepsilon_t,$$

where $t \in T$, $T \subseteq \mathbb{Z}$ the time set, $T = \mathbb{Z}$ or $T = \{t_0, t_0 + 1, t_0 + 2, \ldots\}$ for some $t_0 \in \mathbb{Z}$. A time series process $\{y_t\}_{t \in T}$ is a sequence of observable variables characterized by its time ordering and its correlation between the values in the sequence. According to Greene (2003), the signature element of such time series process is that empirically, the data generating process produces exactly one realization of the sequence. Therefore certain assumptions, such as e.g. time-invariance of the time series, are required to make estimation possible.

The area of time series modelling is very large. Therefore for an in-depth review of different modelling frameworks and their applications, we refer to the literature see, e.g., Lütkepohl and Krätzig (2004). Within the more extensive time series econometrics literature, vector autoregression (VAR) emerged in macroeconomics as a much-used framework for multivariate analysis with time-series data.

## 1.3 VAR Models

### 1.3.1 VAR Models and Other Macro-econometric Methods

VAR models in macroeconomics were introduced by Sims (1980). Since their introduction they quickly became the standard method in empirical macroeconomics and finance. Before 1980, the dominant approach relied on Dynamic Simultaneous Equations (DSEM) models. The Cowles Commission was important in the development of DSEMs for research in economics, led by Nobel Prize laureate Lawrence Klein, whose work dealt with linking Keynesian economic theory to statistical methods. The development of DSEMs was closely related to the policy arena, and the models grew larger (in numbers of equations and parameters) as economists aimed for greater realism. However, they relied on *implausible* assumptions (Sims 1980). VARs emerged as a better alternative, and DSEMs became much less prevalent in academic research (see Lütkepohl 2013, Kilian 2013), although they are still included in the set of models run by some central banks.

Along with the development of VAR models, the rise of *rational expectations* in macroeconomics fostered the development of a Dynamic Stochastic General Equilibrium (DSGE) model in the 1980s. DSGEs require explicit specification regarding the economy's micro-structure, under the premise that model structures should be invariant to policy interventions. Economic agents optimise their objective functions in expectation subject to individual and aggregate resource constraints while considering other agents' behaviour rationally.

Separate bodies of literature emerged for VAR and DSGE methodologies, the latter as the theoretical framework and the former as the empirical counterpart. Although VARs emerged as a theory-free method, the last decades' research agenda has focused on including economic theory elements through the form of restrictions. Nowadays, both alternatives are included in the toolbox of industrial researchers and policymakers, as complementary methodologies, for forecasting and macroeconomic policy analysis.

### 1.3.2 Structural Form and Reduced Form

Once a VAR has passed the model testing stage, it can be used for inference applications. VARs have proved to be useful for forecasting purposes (Lütkepohl 2013), especially those which incorporate Bayesian elements to the estimation. Almost every central bank uses VARs in their forecasting models (including the Federal Reserve System's Board of Governors, 2014). By plugging in the Granger-causality approach (Granger 1969) into VARs, they could also be used to assess causality from one variable to another (Lütkepohl 2013).

Probably the most critical application of VARs to real-world policy problems is structural analysis. Unlike standard regressions, SVARs allows the researcher to evaluate relationships across variables within a multivariate, dynamic setting where it is possible to check how a variable responds to a shock in the system at and throughout different periods. The first alternative is *impulse response functions*, which allow us to check the responses on a given variable following an impulse (shock) in another one; impulse responses are widely used in macroeconomics to compare them to the results generated by dynamic stochastic equilibrium models (Kilian 2013) Note: Generally speaking The DSEM model is a combination of the state-space model with the dynamic factor model, merged with full SEM functionality, e.g. in addition to the core extensions of two-level and cross-classified modeling as well as categorical variables. A second alternative is the *forecast error variance decomposition*, which allow to investigate how the variance of each variable is explained by the individual variance of the remaining variables in the model (Lütkepohl 2005). The third tool is the *historical decomposition*, which unlike the previous two alternatives (they describe average movements in the data and represent unconditional expectations), allows the researcher to quantify how much of the variable's historical fluctuations are explained by structural shocks (Kilian 2013).

Generally speaking, the main difference between VARs (reduced form) and SVARs (structural form) is the fact that the latter incorporates restrictions on the parameters, i.e., identification (or identifiability) strategies that allow the econometricians to estimate causal effects with meaningful economic interpretation. Kilian and Lütkepohl (2017) provides the most up-to-date textbook on SVAR models, covering different estimators, identification methods, and practical advice on working with such models for macroeconomic policy analysis.

There is not an unified structural form for VAR models nor a single method to estimate them (see Kilian and Lütkepohl (2017)), however, here we consider the most general specification based on Kilian (2013):

$$B_0 y_t = B_1 y_{t_1} + \cdots + B_K y_{t_K} + w_t,$$

where

$$y_t^{\text{new}} = \begin{pmatrix} y_t \\ x_t \end{pmatrix}$$

is a $p \times 1$ vector of variables, presumed to be zero mean in the base case. $B_i$'s have a dimension of $p \times p$. $w_t$ is a $p \times 1$ vector assumed to be white noise. Furthermore it is assumed that the elements of $w_t$ are mutually uncorrelated and have clear interpretations in relation to an underlying economic theory and that the model variables are driven by $p$ different shocks such that their variance-covariance matrix $\Sigma_w$ has full rank. The model presented above can be expressed in so called reduced form as:

$$y_t = B_0^{-1} B_1 y_{t_1} + \cdots + B_0^{-1} B_K y_{t_K} + B_0^{-1} w_t,$$

where $B_0^{-1} B_1 = A_1$, $B_0^{-1} B_K = A_K$ and $B_0^{-1} w_t = u_t$ assuming that $B_0$ is invertible. The covariance matrix of the structural errors $\mathbb{E}(w_t w_t') \equiv \Sigma_w = I_p$ such that the reduced-form error covariance matrix is given by $\mathbb{E}(u_t u_t') \equiv \Sigma_u = B_0^{-1} B_0^{-1\prime}$. Going from the structural form to the reduced form -and vice versa- requires knowledge about matrix $B_0$ which contains the instantaneous relationships among the variables included in the model, or about its inverse $B_0^{-1}$, the structural impact multiplier matrix. Considering that $u_t = B_0^{-1} w_t$, the mutually correlated reduced form innovations $u_t$ can be expressed as weighted averages of the mutually uncorrelated structural innovations $w_t$, with the elements of $B_0^{-1}$ serving as weights. Much of the structural-form VAR (SVAR) literature has focused on different methods to recover estimates of $B_0^{-1}$, therefore see Kilian and Lütkepohl (2017) for a textbook treatment on that matter.

Analogous to Lütkepohl (2005) and Lütkepohl (2013), a VAR of order $K$ takes the following form:

$$y_t = A_1 y_{t-1} + \cdots + A_K y_{t-K} + u_t, \tag{1.1}$$

where $y_t = (y_{1t}, \ldots, y_{pt})'$ is a data vector containing a $p$-tuple of observed time series, $A_i$ $(i = 1, \ldots, K)$ are $(p \times p)$ fixed coefficient matrices and $u_t = (u_{1t}, \ldots, u_{pt})'$ is a $p$-dimensional white noise innovation with $E(u_t) = 0$ and covariance matrix $E(u_t u_t') = \Sigma_u$.

VARs can be estimated using ordinary least squares (OLS), maximum likelihood (ML) or Bayesian methods. The VAR in equation (1.1) can be rewritten in compact form as

$$y_t = \left[A_1, \ldots, A_K\right] Z_{t-1} + u_t, \tag{1.2}$$

where $Z_{t-1} = (y_{t-1}', \ldots, y_{t-K}')' \in \mathbb{R}^{Kp}$. As is well-known, e.g., see Lütkepohl (2005), model parameters can be estimated efficiently by OLS. The estimator is:

$$\left[\widehat{A_1}, \ldots, \widehat{A_K}\right] = \left(\sum_{t=1}^{N} y_t Z_{t-1}'\right)\left(\sum_{t=1}^{N} Z_{t-1} Z_{t-1}'\right)^{-1}. \tag{1.3}$$

Note that this only works if $\sum_{t=1}^{N} Z_{t-1} Z_{t-1}'$ is invertible, so $N \geq Kp$ has to hold. For any Gaussian process $y_t$, where $u_t \sim N(0, \Sigma_u)$, the OLS estimator from equation (1.3) is equivalent to the ML estimator, conditional on the initial pre-sample values $(y_0, \ldots, y_{K-1})$.

Oftentimes, reduced-form VARs are believed to be *theoretical*, based on their lack of econometric identifiability (they don't rely on any parameter restriction to recover economic shocks; see Kilian (2013)) and their disconnectedness from standard macroeconomic theory during the process of econometric specification and estimation. Hence, VARs are mainly used for forecasting rather than policy analysis. In the forecasting domain, reduced-forms VARs have benefited from the development in Bayesian methods and the introduction of priors has become a common practice following seminal work by Litterman (1984) and Doan et al. (1986). In this work we will focus on unconstrained maximum likelihood estimation methods (some remarks will be made in chapter 2 section 2.3 about allowing for constraints in the general state-space model).

### 1.3.3 VAR Lag Length and Model Validation

As outlined by Lütkepohl (2005), the normal VAR workflow starts with specifying and testing a reduced-form model (to check whether the VAR represents the data generating process of the variable adequately) before proceeding to actual statistical inference. So we try to model our time series using (1.1) to decide on lag length used.

The order $K$ of the VAR is equal to the highest lag in the model. This is also called *the lag length*. Here we discuss two approaches to determine the lag length. The first approach is to use the LR (likelihood ration) test for parameter restrictions. The LR statistic (Lütkepohl 2005) for testing a value $K$ against $K+1$ has the following form:

$$LR(K) = N\Big[\log\Big(\det\big(\widehat{\Sigma}_u(K)\big)\Big) - \log\Big(\det\big(\widehat{\Sigma}_u(K+1)\big)\Big)\Big], \tag{1.4}$$

where

$$\widehat{\Sigma}_u(K) = N^{-1}\sum_{t=1}^{N}\hat{u}_t\hat{u}_t' \tag{1.5}$$

is the ML estimator of the residual covariance matrix for the value $K$. Here $\hat{u}_t$ is the estimate of $u_t$.

A second approach to determine the lag length are information criteria for lag order selection. Following Lütkepohl (2005), those criteria have the general form:

$$C(K) = \log\Big(\det\big(\widehat{\Sigma}_u(K)\big)\Big) + c_N\varphi(K). \tag{1.6}$$

Once again, the residual covariance matrix is given by equation (1.5), based on LS residuals $\hat{u}_t$, $\varphi(K)$ is a function of the order $p$ that penalizes larger VAR orders, and $c_N$ is a weight that may depend on sample size. Information criteria are based on the premise that there is a trade-off between the improved fit of the VAR model, as $K$ increases, and the parsimony of the model. This allows to search for the value of a lag order which balances competing objectives between model fit and parsimony. The most commonly used information criteria for VARs are the Akaike (AIC), Hannan-Quinn (HQC) and Schwarz (SIC), all of them following a similar structure to equation (1.6).

Once a lag length is determined, the assumption that the residuals are uncorrelated need to be tested. The residual auto-correlation is tested by means of the Breusch-Godfrey LM test (Breusch 1978, Godfrey 1978). The null hypothesis for the test states that all residual auto-covariances are zero. In contrast, the alternative hypothesis states that at least one auto-covariance is different from zero. This test is often accompanied by plots of the auto-correlation function and the partial correlation function. Assume a VAR model for the error vector $u_t = D_1 u_{t-1} + \cdots + D_h u_{t-h}$ which will result in ARMA model which is as follow:

$$\hat{u}_t = A_1 Y_{t-1} + \cdots + A_K Y_{t-K} + D_1 \hat{u}_{t-1} + \cdots + D_h \hat{u}_{t-h} + \varepsilon_t. \tag{1.7}$$

The above equation represents a ARMA model which is equivalent to a state-space model. ARMA model provides parsimonious representations of linear data generation processes. The setup for these processes in the presence of stationary and cointegrated variables is considered.

We will check for autocorrelation in the above model. Once, it has been checked that there is no auto-correlation, we will check for the assumptions of normality. However, if there is autocorrelation then at least one of the $D_i$, $i = 1, \ldots, h$ will be nonzero. Normality is checked by means of a multivariate version of the Jarque-Bera test (Jarque and Bera 1987), although normality is not a required condition for most VAR-based statistical procedures. The validity of the assumptions of time introduce can also be investigated. Structural changes are tested by using the so-called Chow-test (Chow 1960). In section 1.5 the question whether the time series is stationary be addressed.

### 1.3.4 Limitations of the VAR Modelling

As most econometric model classes, VARs also have drawbacks. Here we will outline the main two problems highlighted in the literature.

Probably the most well-known problem with VARs is related to the *the curse of dimensionality*. Given that in VARs each variable is explained as a function of its own lags and the lags of the remaining variables included in the data-set, the number of parameters could grow fast if the lag order is increased or if the number of variables grows and consequently, the model is prone to become over-parameterized. Literature addresses the issue of dimensionality by suggesting potential solutions. Sims (1980), Litterman (1984) and Doan et al. (1986) were among the early proponents of the introduction of Bayesian methods to allow for parameter shrinkage, and indeed, this has has been one of the most active research areas in the domain of VAR models over the past three decades.

Another well-known issue is the problem of *non-fundamentalness* in VAR models, initially put forward by Lippi and Reichlin (1994) and subsequently expanded by many others. The issue highlights that econometricians do not have all the information that economic agents have (i.e. households, firms, and policymakers). Particularly, the expectations of such agents may be formed in such a way that they are not based on past information from the variables in the econometric model, but rather incorporate additional information outside the model to which econometricians do not have access. Consequently, the forecast errors governing the data generating process cannot be equated with the reduced-form innovations of the VAR model estimated by the econometricians, and as a result, the causal effects (economic shocks) are *non-fundamental*. As outlined by Kilian and Lütkepohl (2017), this problem has resulted into a strand of literature that focuses on working with larger data-sets, which is possible either by augmenting VAR models with factors (such as the popular factor-augmented VAR) or the development of dynamic factor models which are becoming increasingly popular these days. Here the state-space models applied within the study play an important role.

## 1.4 Spurious Regression

A critical review of literature shows that Yule (1926) was the first statistician to address the topic of spurious regression, where two non-stationary time series could show a high correlation along with an apparent causality, which might yield flawed results due to unseen or invalidated elements, such as serial correlation. Decades later, econometricians became widely aware of spurious regression problems in time series models and a vast strand of literature emerged on both model specification (e.g. Granger and Newbold 1974, Phillips et al. 1986) and unit root detection (e.g. Dickey and Fuller 1979, Kwiatkowski et al. 1992).

## 1.5 Stationarity and Non-stationarity

Lütkepohl and Krätzig (2004) explained that a stochastic process $y_t$ is said to be stationary if it has time-invariant first and second moments, namely if $E(y_t) = \mu$ for all $t \in T$ where $T$ is the time set, and if $E[(y_t - \mu_y)(y_{t-h} - \mu_y)] = \gamma_h$ for all $t \in T$ and all integers $h$ such that $t - h \in T$. The first condition means that all members of a stationary stochastic process have the same constant mean, therefore a time series generated by a stationary process must fluctuate around a constant mean and does not have a trend. The second condition implies that the variances are also time-invariant because, for $h = 0$, the variance $\sigma_y^2 = E[(y_t - \mu_y)^2] = \gamma_0$ does not depend on $t$; likewise, the covariances $E[(y_t - \mu_y)(y_{t-h} - \mu_y)] = \gamma_h$ do not depend on $t$ but on the distance in time $h$ of the two members of the process.

When the conditions explained above are violated, it is said that the time series is *non-stationary*. Many macroeconomic variables are assumed to be either $I(1)$ or $I(n)$ for some $n \in N$. The *order of integration* $I(n)$ refers to how many times a series should be differenced minimally to become stationary. Being $I(n)$, $n > 0$, means that the series is non-stationary of a very particular type, namely such that differencing n times gives a stationary sequence. This is not the case for a general non-stationary series. But in macroeconomics it is found empirically that many variables are of this type. Hence, a stationary stochastic process is $I(0)$ because it does not need to be first differenced. Further, when no differencing transformations are applied to a time series, it is said to be in *level form*.

The stationarity could be detected by visual inspection (since non-stationary time series do not fluctuate around a stable mean and usually have a trend) or more formally, by using unit root test. Among the most popular tests are the ADF test (Dickey and Fuller 1979) and the KPSS test (Kwiatkowski et al. 1992), however, Lütkepohl and Krätzig (2004) provide a more extensive overview of different tests to assess stationarity of time series.

The ADF framework tests the null that the series has a unit root, against the alternative of stationarity. The time series is non-stationary when the test statistic is higher (in absolute value) than the selected critical value, and otherwise the series has no unit root. If a DGP is assumed to be an AR(K) process, it is integrated when $\alpha(1) = 1 - \alpha_1 - \cdots - \alpha_K = 0$. The test is:

$$\Delta y_t = \phi y_{t-1} + \sum_{j=1}^{K-1} \alpha_j^* \Delta y_{t-j} + u_t, \tag{1.8}$$

where $\phi = -\alpha(1)$ and $\alpha_j^* = -(\alpha_{j+1} + \cdots + \alpha_K)$. Namely, the test is based on the t-statistic of the coefficient $\phi$ of the regression above, which can be estimated through standard OLS. As well, the KPSS framework tests that the time series is stationary as a null, against the alternative of unit root. If the test statistic is below the selected critical value, the null hypothesis is rejected. The test has the following statistic:

$$\text{KPSS} = \frac{1}{T^2} \sum_{t=1}^{T} \frac{S_t^2}{\widehat{\sigma}_\infty^2}, \tag{1.9}$$

where $S_t = \sum_{j=1}^{t} \hat{w}_j$ with $\hat{w}_j = y_t - \overline{y}$, and $\widehat{\sigma}_\infty^2$ is an estimator of long run variance process $z_t$, namely

$$\widehat{\sigma}_\infty^2 = \lim_{t \to \infty} T^{-1} \operatorname{Var}\left( \sum_{t=1}^{T} z_t \right). \tag{1.10}$$

This test departs from a DGP $y_t = x_t + z_t$, where $x_t$ is a random walk and $z_t$ is a stationary process.

## 1.6 Cointegration and Integrated Time Series

Most of the macroeconomic series are fairly smooth, moving with local patterns or with long swings, yet the swings are not standard. It is this relative perfection that makes them unsatisfactory for examination with standard statistical methods, which expects data to have "stationarity." In the context of macroeconomics and finance, many series do not have this property and can be classified "integrated" or, "non-stationary." However, when presented as changes or rates of returns, these series show up nearer to being stationary. Statistical methods to evaluate a single integrated series had been proposed already by Box and Jenkins (1970), yet the joint examination of data sets of such series was missing a significant element. It would appear the (weighted) difference between a couple of integrated series or more generally a non-trivial linear combination of a couple of integrated series can be stationary, and this property is known as *cointegration*.

Besides, distinguishing between growth and cycles is fundamental in macroeconomics. One can define economic growth as the time-varying steady state, or the permanent component, and cycles as deviations from the steady state, or the transitory component. One may interpret the permanent and transitory components as the natural rate and gap, respectively, though some economists may disagree with such interpretation, in which case one can consider the permanent component of the natural rate. If shocks affecting the two components differ, then policy prescriptions for promoting growth and stabilizing cycles differ. Thus, it is useful to decompose economic fluctuations into the two components. One such model is the multivariate Beveridge-Nelson (BN) decomposition model Boswijk et al. (2010), which decomposes a multivariate series into a random walk permanent component and a transitory component, assuming a linear time series model such as a VAR model or a vector error-correction model (VECM) for the differenced series. However, there is one drawback of using such a model. If there is a large reduction in the long-run growth rate, a forecasting model that fails to account for it will keep anticipating faster growth than actually occurs after the break, leading to a persistently negative estimate of the output gap based on the (BN) decomposition.

The concepts of spurious regression, unit roots and stationarity lie at the core of the cointegration debate and literature. Boswijk et al. (2010) provide an in-depth review of the historical context in which cointegration emerged. The next lines are based on their work. Engle and Granger (1987) is one of the most cited time series econometrics papers, but the concept had already been introduced by Granger (1981). There are several factors behind the success of cointegration, according to Boswijk et al. (2010). First, it managed to combine different strands of literature into a single framework in an elegant manner. Second, large-scale macro-econometric models had less forecasting power than simple time series models (e.g. AR, ARIMA), although the latter lacked economic theory. Third, the presence of stochastic trends in macroeconomic time series motivated an urgency for novel econometric tools to analyze such data in a proper way. Fourth, the availability of macroeconomic data increased around those years, and therefore many researchers started to explore the existence of long-run equilibrium relationships. Fifth, a key element allowed the practical implementation of the method: the increase of computing facilities and software tools to conduct cointegration analysis.

As documented by Boswijk et al. (2010), after the introduction of the influential Box-Jenkins methodology (ARIMA) in the early 1970s, it delivered high quality forecasts in despite of being a rather simple framework and the interest in time series analysis -in economics- was boosted. ARIMA models had a better forecasting power than most large-scale macro-econometric model at the time, famous for containing several dozens of equations. Separately, the unit root literature took off with papers like Dickey and Fuller (1979) and many others as mentioned in the previous section. Moreover, such academic circumstances were accompanied by three remarkable technological facts: the market entry of affordable personal computers in the beginning of 1980s, the availability of reliable quarterly macroeconomic data for most western countries, and the explosion in statistical packages and matrix programming languages made available for personal computers.

In that scenario, cointegration models satisfied an urgent need of a small-scale framework, incorporating stochastic trend data, useful for forecasting macroeconomic data. Of course, the development of the cointegration literature (similarly to VAR) was not always smooth and it took years to extend the initial proposition by Engle and Granger (1987).

## 1.7 Vector Error Correction Model

If two variables are non-stationary and first-order integrated $I(1)$ hence stationary, they might still yield a stationary linear combination, which is known as cointegration. In such scenario, the regression involving those variables is not spurious and could provide insights about long-run relationships between them Wooldridge (2012).

There are two approaches to cointegration and the corresponding error correction model. Engle and Granger (1987) proposed the two-step univariate approach which we are not addressing here. The second and probably the most popular error correction approach is the one proposed by Johansen (1988, 1991) under a VAR scheme. Recall the VAR with $K$ lags in equation (1.1). It can be rewritten in vector error correction (VEC) form as:

$$\Delta y_t = \Pi y_{t-1} + \sum_{i=1}^{K-1} \Gamma_i \Delta y_{t-i} + u_t, \tag{1.11}$$

where

$$\Pi = \sum_{j=1}^{j=K} A_j - I_K, \quad \text{and} \quad \Gamma_i = -\sum_{j=i+1}^{j=K} A_j, \quad \text{and} \quad \Delta y_t = y_t - y_{t-1}.$$

As the reader might see by comparison to equation (1.1), a VEC is equivalent to a VAR model.

The work by Engle and Granger (1987) shows that if the variables $y_t$ are integrated of order 1, the matrix $\Pi$ has rank $0 \leq r < p$, where $r$ is the number of linearly independent cointegrating relations. If the variables are cointegrated, $0 < r < p$ and comparison of (1.11) with (1.1) shows that a VAR in first differences is poorly specified because it omits the lagged level term $\Pi y_{t-1}$. The study aims to prove this fact in section 2.2, which also discusses state-space models.

### 1.7.1 VEC Estimation

This study considers a VEC model without deterministic terms that can be generalized to state space models see chapter 2 and section 2.2:

$$\Delta y_t = \Pi y_{t-1} + \Gamma_1 \Delta y_{t-1} + \cdots + \Gamma_{K-1} \Delta y_{t-K+1} + u_t, \tag{1.12}$$

Where $y_t$ is a process of dimension $p$, $\text{rk}(\Pi) = r$ with $0 \leq r < p$ so that if $r > 0$, then $\Pi = \alpha\beta'$, where $\alpha$ and $\beta$ are $(p \times r)$ matrices with $\text{rk}(\alpha) = \text{rk}(\beta) = r$. All other symbols follow conventional notations, namely, $\Gamma_j$ $(j = 1, \ldots, K-1)$ are $(p \times p)$ parameter matrices and $u_t \sim (0, \Sigma_u)$. $y_t$ is assumed to be $I(1)$. The VEC could be also written in matrix notation as:

$$\Delta Y = \Pi Y_{-1} + \Gamma \Delta X + U. \tag{1.13}$$

In matrix notation,

$$\Delta Y := \left[\Delta y_1, \ldots, \Delta y_T\right],$$
$$Y_{-1} := \left[y_0, \ldots, y_{T-1}\right],$$
$$\Gamma := \left[\Gamma_1, \ldots, \Gamma_{K-1}\right],$$

$$\Delta X := \left[\Delta X_0, \ldots, \Delta X_{T-1}\right] \quad \text{with} \quad \Delta X_{T-1} := \left[\begin{array}{c} \Delta y_{t-1} \\ \vdots \\ \Delta y_{t-K+1} \end{array}\right],$$

$$U := \left[u_1, \ldots, u_T\right].$$

The OLS estimator (Lütkepohl 2005) for a VEC is given by:

$$\left[\Pi : \Gamma\right] = \left[\Delta Y Y_{-1}' : \Delta Y \Delta X'\right] \left[\begin{array}{cc} Y_{-1} Y_{-1}' & Y_{-1} \Delta X' \\ \Delta X Y_{-1}' & \Delta X \Delta X' \end{array}\right]. \tag{1.14}$$

The white noise covariance matrix is:

$$\Sigma_u := (T - p_K)^{-1} \left(\Delta Y - \Pi Y_{-1} - \Gamma \Delta X\right)\left(\Delta Y - \Pi Y_{-1} - \Gamma \Delta X\right)'. \tag{1.15}$$

However, if one imposes $\mathrm{rk}(\Pi) = r$ and assumes $y_t$ is Gaussian, the VEC can estimated by ML methods as in Johansen (1988). The log-likelihood function for a sample of size $T$ is:

$$\begin{gathered} \ln l = -\frac{pT}{2} \ln 2\pi - \frac{T}{2} \ln |\Sigma_u| - \\ -\frac{1}{2} \mathrm{tr}\left[\left(\Delta Y - \alpha\beta' Y_{-1} - \Gamma \Delta X\right)' \Sigma_u^{-1} \left(\Delta Y - \alpha\beta' Y_{-1} - \Gamma \Delta X\right)\right]. \end{gathered} \tag{1.16}$$

The asymptotic properties of both OLS and ML estimators are fully available in chapter 7 of Lütkepohl (2005) book. The order of the VEC is always one less than the underlying VAR. However, the number of parameters is the same (that is if we count all the entries of $\Pi$ as parameters).

### 1.7.2 Johansen Cointegration Test

Determining the cointegration rank is a key step in cointegration modelling. In the context of VEC models formal methods are provided by Johansen (1988, 1991), namely, the *trace test* and the *maximum eigenvalue test*. Let $r$ be the rank of matrix $\Pi$ (the number of cointegrating vectors), the *trace statistic* allows to test whether the rank of $\Pi$ is $r_0$, against the alternative hypothesis where $r_0 < \mathrm{rank}(\Pi) \leq p-1$, where $p-1$ is the maximum number of possible cointegrating vectors. If the null is rejected, then the next null is that $\mathrm{rank}(\Pi) = r_0 + 1$ and the alternative hypothesis is that $r_0 + 1 < \mathrm{rank}(\Pi) \leq p-1$. The likelihood ratio test statistic is:

$$LR(r_0, n) = -T \sum_{i=r_0+1}^{n} \ln(1 - \lambda_i). \tag{1.17}$$

The *maximum eigenvalue statistic* tests whether the largest eigenvalue $\lambda_1$ is zero against the alternative that it is positive while the next largest eigenvalue is zero. If the rank of $\Pi$ is zero, then the largest eigenvalue is zero and there is no cointegration; on the other hand, if the largest eigenvalue is different from 0, the rank of the matrix is at least one so these is a cointegrating vector and there might be more cointegrating vectors. The same rationale applies to test the second and third largest eigenvalues. The test statistic is:

$$LR(r_0, r_0 + 1) = -T \ln(1 - \lambda_{r_0+1}). \tag{1.18}$$

### 1.7.3 Practical Issues in VEC Modelling

VEC offers two distinct advantages compared to VAR and their significance will vary depending on the objectives of the researcher. First, there are gains in terms of statistical efficiency when the VEC is correctly specified. Second, such model facilitates the imposition of restrictions on the long-run effects or structural shocks in a VAR.

The practical workflow follows a logical method. First, the econometrician must verify the potential existence of unit roots in the time series that will be used in the model, in particular to determine the order of integration of the series. If the series are found to be non-stationary (i.e. they have a unit root as it was assumed from the outset that the series is integrated for some values of n, i.e., $I(n)$), it is advised not to use a straight forward regression model, see Yule (1926), as it would yield spurious and statistically unreliable results. A popular approach to work with non-stationary series models relies on exploring cointegration relationships. Both the two-step (Engle and Granger 1987) and the multivariate Johansen (1988, 1991) approach to cointegration serve the purpose of inspecting long-term relationships test for cointegration. If the variables are found to be cointegrated, it means that they yield non-spurious results despite their lack of stationarity. If the variables are not cointegrated, they could be used in a VAR under the assumptions put forward by Sims et al. (1990), Doan et al. (1986), but they should not be used in standard regressions. Regarding the steps outlined above, there is not a unique way to proceed. At all of the stages described, the researcher's judgement plays a significant role when deciding which approach to follow, how to deal with non-stationarity, and which methodology to select. Section 4 will find cases in which cointegration modelling with state-space work better than the standard VECM.

## 1.8 State-Space Models

A state-space model is an overall model that includes an entire class of unbelievable instances in much a similar way as linear regression does. It is also known as a dynamic linear model which was presented by Kalman et al. (1960) and Kalman and Bucy (1961). The model was used in the setting of tracking a nominal trajectory in space, where the state equation characterizes the movement equations for the position or velocity of a spacecraft with area $x_t$ and the data $y_t$ to reflect information which can be observed to perform maximum likelihood estimation. In spite of the fact that the filter was originally considered as a technique for use in aerospace-related examination, the model has been applied to modeling data from economics (Harrison and Stevens 1976, Harvey and Todd 1983, Shumway and Stoffer 1982), medical sciences (Jones 1984) and the environmental sciences (Shumway 1988). A treatment of time series investigation dependent on the state-space model is the research by Durbin and Koopman (2001). Also, study by Douc et al. (2014) presented findings relevant to non-linear state-space models. Generally, the state-space model is described by two processes. To start with, there is a latent or hidden procedure $x_t$ called the state process. The state process is expected to be a Markov process; this implies the future $\{x_s; s > t\}$, and past $\{x_s; s < t\}$, are independent contingent on the present, $x_t$. The subsequent condition is that the observations, $y_t$ are conditionally independent given the states $x_t$. This implies the correlation among the observations is created by states. In the following chapters, the study will focus on incorporating the cointegration into the state-space model. This technique proposes an estimation method using parameterization that translates the cointegration property into a low-rank constraint on a resulting likelihood optimization problem. Hence, this study will highlight the advantages of using the state-space cointegrated models in the field of economics research.

# Chapter 2

# The Discrete-Time State-Space Error Correction Model

In this chapter we will discuss some of the fundamental concepts used in the study of economic and financial time series. The co-integration rank, generally denoted as $r$ in this thesis, represents the number of independent linear combinations satisfying stationarity property. Our set of economic time series will then be said co-integrated of rank $r$, see Engle and Granger (1991). In general, most economic and financial time series are different stationary. Over many decades econometricians have been interested in developing models to study economic time series behaviors, see Boswijk et al. (2010). The most common linear time series models are driven by a stochastic white noise process, that if often taken to be a Gaussian white noise process. This is also what we will do in this thesis. Within time series modeling the VAR and VARMA models play a special role. VAR models are utilized in academic and functional settings to investigate the connections among several economic variables. VAR models are a particular case of the VARMA model, which is suitable for multivariate analysis and utilizes the moving average along with the VAR structure. While VAR models remain the most widely used model for cointegration studies, there is a growing interest in cointegrated VARMA models. Existing specification and estimation procedures for co-integrated VARMA models can be found in Yap and Reinsel (1995), Lütkepohl and Claessen (1997), and also Poskitt (2006). Common to these papers is the use of the reverse "Echelon-Form", a set of parameter restrictions that make sure the remaining coefficients are identified for the likelihood function. A related but different approach uses so-called "scalar component" representations originally proposed by Tiao and Tsay (1989) and embedded in a complete estimation procedure by Athanasopoulos and Vahid (2008). While both structures can be quite parsimonious representations of a given process, they can display relatively complex structures. Cooley and Dwyer (1998) claim that modeling macroeconomic time series systematically as the underlying economic theory does not justify pure VARs. A comparison of structural identification using VAR, VARMA and state-space representations is provided by Kascha and Mertens (2009). They estimate vector autoregressive moving average (VARMA) and state-space models, which are not misspecified, using simulated data and compare true with estimated impulse responses of hours worked to a technology shock, and they find few gains from using VARMA models.

However, the VARMA form is very convenient for the interpretation of the results. The state-space model is equivalent to VARMA as the estimated state-space model can constantly be rewritten as a VARMA model and any VARMA model can always be rewritten in state-space form. Compared to the state-space model, the VARMA model, we think, is not a good starting point for parameter estimation as it has several drawbacks (a) one needs to estimate a, possibly significant, number of discrete parameters, related to the maximal delays in the VARMA equations, while in the cointegrated state-space model the state-space dimension (MacMillan degree) and the cointegrating rank are the only two discrete parameters that have to be determined (estimated). (b) as has been shown in Hanzon (1993), the VARMA parameters can take on huge values (theoretically they can take arbitrarily large values) even though the corresponding impulse response coefficients are bounded. Therefore, one has to search over unbounded parameter space. In the approach followed in this thesis, we can reduce the search to one over a parameter space that is bounded (the parameters are angles). Therefore, our approach become very interesting as the state-space model can constantly be rewritten as a VARMA model.

State-space equations can specify any linear time-invariant finite dimensional Gaussian time series process on the one hand or by VARMA equations on the other hand. In this thesis, we work with the state-space equations, but using established methods, these can be transformed into VARMA equations and vice versa. So one would perhaps expect that maximum likelihood estimation would give identical results whether one uses the state-space version or the VARMA version of the process. In practice, this is, not entirely true, because to apply the maximum likelihood estimation procedure, one has to determine a number of discrete parameters: in the case of state-space, one has to determine to the dimension of the state-space (also known as the order or the MacMillan degree);

in the case of the VARMA representation one has to determine the maximal delays in the AR and the MA part, and these delays will have to be determined in the context of a particular canonical form for the VARMA model (such as the echelon form). The result is that the likelihood function is maximized over different subsets of models when one used state-space on the one hand and VARMA on the other. However, if the discrete parameters (MacMillan degree in the case of state-space and the delay parameters in the case of VARMA) are estimated correctly, both approaches should give the same answers asymptotically. One can conclude that one method to obtain a maximum likelihood estimate in the VARMA form is to compute a maximum likelihood estimator for the state-space model and represent the result in the VARMA form. This also holds for co-integration models.

This chapter introduces State-Space Error correction models (henceforth SSECM) as a specific type within the general class of linear state-space models. For this purpose, we introduce a unique representation of the class of observationally equivalent linear state-space models. An example for such a representation in the context of VARMA models is the echelon canonical form (see Boswijk et al. (2010)). The advantage of this form is that the non-stationary and stationary parts are decoupled and can be transformed separately. The difficulty with the echelon forms is that they are often not suitable for parameterization for likelihood optimization. SSECMs are novel development in the field of time series econometrics, initially introduced by Ribarits and Hanzon (2014a,b). We will also use SSECMs but we use totally different parameterizationsn compared to those used in Ribarits and Hanzon (2014b,a). They are based on the (input normal) canonical forms of Hanzon et al. (2009) which hav good properties for likelihood optimization.

A linear state-space model can be taken to be observable and controllable. The theory of controllability plays a vital role in the design of modern control systems. It has been developed for a large part in the area of Control Theory, but also in the context of Time Series models (Bierens 1997). Parameterization is the process of deciding and defining the parameters necessary for a complete or relevant specification of a model. It is primarily a mathematical process involving identifying a complete set of effective coordinates of the system or model. However, we are working with controllable models. A controllable one can replace a model that is not controllable with a smaller state-space dimension, which is equivalent to its input-output behavior. Indeed, a state-space model is minimal if and only if it's controllable and observable (see Hannan and Deistler (1988)).

The literature and established results on state-space models will be helpful to study State-Space Error, correction models. The co-integration property translates into a low-rank condition on an appropriate matrix. We will strive to find the maximum likelihood estimator by maximizing the likelihood function. First, we will partially optimize the likelihood function analytically. The resulting criterion is a function of eigenvalues of a matrix due to the low-rank constraint. Mirroring the Johansen approach, Ribarits and Hanzon (2014b,a) we reduce the constrained non-linear likelihood optimization of the parameters of the SSECM to a low-rank generalized eigenvalue-eigenvector problem. Hence, there are several steps to be taken to first solve this partial optimization problem. For completeness, we will follow the steps in Ribarits and Hanzon (2014a,b). In the following sections, we outline the basic concepts and theorems, providing more detail. Then, after the partial optimization steps have been completed, we will use a search method to optimize the remaining model parameters. We will apply the envelope theorem to calculate the derivatives of the parameterized matrix eigenvalues, which will allow us to use a gradient method. We apply a new parameterization that we mentioned earlier Hanzon et al. (2009). This has a minimal number of parameters, each of which is taken from a finite interval. An overview of the sections in this chapter is now as follows:

First, section 2.1 will introduce the SSECM as a specific type state-space model.

After, in section 2.2, we will explain the proposed state-space error correction model and show that a co-integrated state-space model can be rewritten in SSECM form and how the structure of the SSECM can be exploited to set up a strategy to minimize the likelihood function. There is a new, concise proof of the critical rank condition in the error correction model. Hence, in section 2.3, we end up with the partially optimized ("concentrated") likelihood in terms of a number of remaining parameters. Here, we follow the approach of Ribarits and Hanzon (2014b,a) closely.

Then, in section 2.4, we will present the parameterization that is obtained by considering in a particular form, namely the so-called input-normal sub-diagonal pivot structure. The parameterization of the models that is used here is entirely different from that in Ribarits and Hanzon (2014b,a).

Finally, in section 2.5, a gradient method (gradient search algorithm) is derived to optimize the remaining parameters of the model. An error concerning the use of the envelope theorem in Ribarits and Hanzon (2014b,a) is detected and corrected in this section.

## 2.1 Specification of the State-Space Model with Cointegration

A textbook treatment of state-space models is provided in chapter 18 of Lütkepohl (2005). The state-space models have been widely used in system theory, engineering, and physical sciences. State-space models are based on the general idea that an observed (multiple) time series depends on a possibly unobserved state which follows a stochastic process, and such relationship is described by the *observation* or *measurement equation*. A basic model is given by $y_t = H_t z_t + v_t$, where $y_t$ is a vector of observable time series, $z_t$ is the unobserved state, $H_t$ is the *measurement matrix* that depends on the period of time $t$, and $v_t$ is the *observation error* oftentimes assumed to be a noise process. The *state vector* is generated as $z_t = B_{t-1} z_{t-1} + w_{t-1}$, which is often called the *transition equation* since it describes the transition of the state vector from period $t-1$ to $t$. $B_t$ is coefficient matrix that may depend on $t$, and $w_t$ is an error process.

The aim of this section is to introduce State-Space Error correction models (henceforth SSECM), as a specific type within the State-Space domain. SSECMs are a novel development in the field of time series econometrics, initially introduced by Ribarits and Hanzon (2014a,b).Imagine the following linear, time invariant, discrete time state-space system for $t = t_0, t_0+1, t_0+2, \ldots$:

$$x_{t+1} = Ax_t + B\varepsilon_t, \quad x_{t_0} = x_0, \tag{2.1}$$
$$y_t = Cx_t + \varepsilon_t. \tag{2.2}$$

Here, $x_t$ is a $n$-dimensional unobserved state vector. $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{p\times n}$ are parameter matrices. $y_t$ is the observed $p$-dimensional output. Additionally, $(\varepsilon_t)_{t\in\mathbb{N}}$ is a $p$-dimensional Gaussian discrete time white noise process with $\mathbb{E}\varepsilon_t = 0$ and $\mathbb{E}\varepsilon_t\varepsilon_t' = \delta_{s,t}\Sigma$ where $\delta_{s,t} = 1$ for $s = t$ and $\delta_{s,t} = 0$ otherwise. Often, this model is based on the following assumptions:

(i) $(A, B, C)$ is minimal (if not it can be replaced by a minimal model, i.e. one with minimal $n$).

(ii) The eigenvalues of $A$ are strictly within the unit circle or at one.

(iii) The algebraic multiplicity of the eigenvalue one of $A$ is equal to its geometric multiplicity, $c$.

(iv) The eigenvalues of $(A - BC)$ are strictly within the unit circle, which is called the *strict minimum phase* assumption.

As is well known Ribarits and Hanzon (2014a,b) such a state-space system is *minimal* if and only if;

$$\mathrm{rk}\left[B, AB, \ldots, A^{n-1}B\right] = \mathrm{rk}\left[C', A'C', \ldots, (A')^{n-1}C'\right]' = n.$$

## 2.2 The Cointegrated Discrete-Time State-Space Error Correction Model

Given that the assumptions (i)-(iv) discussed in the previous section hold, A square matrix $A$ is *asymptotically stable* if $\mid \lambda_{\max}(A) \mid< 1$, where $\lambda_{\max}(A)$ denotes an eigenvalue of $A$ of maximum modulus. $A$ is *stable* if the sequence of matrices $\{A^k\}_{k=0}^{\infty}$ is bounded. The assumptions (ii) and (iii) imply that $A$ is stable. The multiplicity $c$ is equal to the number of *common trends* in the model. If $c$ satisfies $1 \leq c \leq p-1$, then it is within the class of cointegrated $I(1)$ models. The number $r := p - c$ is called the *rank* of cointegration, namely, the number of linearly independent cointegrating vectors. This terminology is explained below:

By rewriting (2.2) as (2.8) below and substituting for $\varepsilon_t$ in (2.1) the study obtains the inverse state-space model (with $y_t$ as inputs and $\varepsilon_t$ as outputs), as follows:

$$y_t = CA^{t-t_0}x_0 + \sum_{j=1}^{t-t_0} CA^{j-1}B\varepsilon_{t-j} + \varepsilon_t. \tag{2.3}$$

For simplicity, in this study we will assume that $x_0 = 0$, and we will define $\varepsilon_t = 0$, $y_t = 0$, for all $t$ with $t < t_0$. Let $z$ denote the lag operator acting on the (infinite) time series so that $zy_t = y_{t-1}$, then (2.3) can be written as:

$$y_t = k(z)\varepsilon_t, \tag{2.4}$$

where

$$k(z) = \sum_{j=1}^{\infty} CA^{j-1}Bz^j + I. \tag{2.5}$$

Interpreting $z$ as a complex number with $|z|$ sufficiently small, one obtains the formula

$$k(z) = Cz(I - zA)^{-1}B + I. \tag{2.6}$$

As this is a rational matrix function, it is actually well-defined for all $z \in \mathbb{C}$, $z^{-1} \notin \sigma(A)$. Here, as well as in the following, the spectrum of $A$ is denoted by $\sigma(A)$. The matrix $k(z)$ is a $p \times p$ is a matrix of rational functions called *the transfer matrix or transfer function* corresponding to the state-space model. Let $\overline{k}(z) := k(z)^{-1} = z\overline{C}(I - z\overline{A})^{-1}\overline{B} + I$, where $(\overline{A}, \overline{B}, \overline{C})$ can be obtained by inverting the state-space model to obtain:

$$x_{t+1} = (A - BC)x_t + By_t, \tag{2.7}$$

$$\varepsilon_t = -Cx_t + y_t, \tag{2.8}$$

where

$$\overline{A} = (A - BC),$$
$$\overline{B} = B,$$
$$\overline{C} = -C.$$

Then

$$\varepsilon_t = \overline{k}(z)y_t = \sum_{j=1}^{\infty} \overline{C}(\overline{A})^{j-1}\overline{B}y_{t-j} + y_t. \tag{2.9}$$

As shown by Ribarits and Hanzon (2014a), it is known that the geometric multiplicity of $c$ of 1 as an eigenvalue of $A$ can be read off from the rank of $\overline{k}(1)$, which is in fact $\text{rank}(\overline{k}(1)) = r = p - c$. We will give a short alternative proof below (see Theorem 1) .

An equivalent statement is that there must exist a pair of full column rank $p \times r$ matrices $\alpha$ and $\beta$, such that $\overline{k}(1) = \overline{C}(I - \overline{A})^{-1}\overline{B} + I = -\alpha\beta'$. By defining

$$\widetilde{k}(z) := I + \frac{\overline{k}(z) - \overline{k}(1)z}{(z - 1)},$$

one finds $\widetilde{k}(0) = 0$ and the equation

$$\varepsilon_t = \overline{k}(z)y_t = \left[\widetilde{k}(z)(z-1) + \overline{k}(1)z - (z-1)\right]y_t$$

can be rewritten in the *error correction form*:

$$\Delta y_t = \Pi y_{t-1} + \widetilde{k}(z)\Delta y_t + \varepsilon_t, \tag{2.10}$$

where $\Pi = \bar{k}(1)$ is a matrix of rank $r$.

As $\widetilde{k}(0) = 0$, one finds that except for $\varepsilon_t$, the terms on the right-hand side of (2.10) are known at time $t-1$; therefore, such an equation could be considered a regression equation where coefficients $\alpha$, $\beta$ and those determining $\widetilde{k}$ have to be estimated.

From the state-space matrices for $\widetilde{k}(z)$, the error correction form in equation (2.10) can be rewritten in the *state-space error correction form* (SSECM) as shown in (Ribarits and Hanzon 2014a):

$$\begin{aligned}
&\Delta y_t = \Pi y_{t-1} + \overline{CA}(I-\overline{A})^{-1}\tilde{x}_t + \varepsilon_t,\\
&\tilde{x}_{t+1} = \overline{A}\tilde{x}_t + \overline{B}\Delta y_t,\\
\text{s.t.}\quad &-\Pi = \overline{C}(I-\overline{A})^{-1}\overline{B} + I_p, \quad \text{(SSECM)},
\end{aligned} \tag{2.11}$$

where $\overline{A} := (A - BC), \overline{B} := B$, and $\overline{C} := -C$ and $\tilde{x}_t = \Delta x_t$. Here all variables with time index less than $t_0$ are declared to be zero for the consistency of the notation.

**Lemma 1.** *Let a state-space model follow the assumptions as outlined, with $c$ denoting the multiplicity of 1 as an eigenvalue of $A$, and $r$ the number of linearly independent cointegrating relations of the process $y_t$. Then $c \leq \min(n,p)$, $r = p - c$, and*

$$\text{rk}\left[\overline{C}(I-\overline{A})^{-1}\overline{B} + I_p\right] = r.$$

The lemma follows directly from the following (general) theorem.

**Theorem 1.** *Assume that $(A,B,C)$ is minimal, and $(I-A+BC)$ is invertible. Then: $I-A$ has rank $n-c \Longleftrightarrow \text{rk}(-C(I-A+BC)^{-1}B + I_p) = p - c (= r)$.*

*Proof.* This was shown in Ribarits and Hanzon (2014a,b). Here, we will provide an alternative and much shorter, direct proof. Note that (using partitioned matrices in an obvious way)

$$\begin{bmatrix} I-A+BC & B \\ C & I_p \end{bmatrix} \times \begin{bmatrix} I_n & 0 \\ -C & I_p \end{bmatrix} = \begin{bmatrix} I-A & B \\ 0 & I_p \end{bmatrix},$$

and

$$\begin{bmatrix} I-A+BC & B \\ C & I_p \end{bmatrix} \times \begin{bmatrix} I_n & -(I-A+BC)^{-1}B \\ 0 & I_p \end{bmatrix}$$
$$= \begin{bmatrix} I-A+BC & 0 \\ C & -C(I-A+BC)^{-1}B+I_p \end{bmatrix}.$$

It follows that the co-rank $c$ of $I-A$ is equal to that of $-C(I-A+BC)^{-1}B+I_p$ and both are equal to the co-rank of

$$\begin{bmatrix} I-A+BC & B \\ C & I_p \end{bmatrix},$$

as $I-A+BC$ is invertible. It follows that $\mathrm{rk}(I-A) = n-c \iff \mathrm{rk}(-C(I-A+BC)^{-1}B+I_p) = p-c (= r)$. □

It is follows that the SSECM exhibits cointegration with cointegration rank $r$ if and only if $\mathrm{rk}(\Pi) = r = p-c$. As is well-known $\mathrm{rk}(\Pi) = r \leq p$ if and only if $\Pi = \alpha\beta'$, for some appropriate full rank $p \times r$ matrices $\alpha$ and $\beta$. Hence, the following SSECM is obtained for a cointegrated state-space model with $r$ cointegrating relations:

$$\begin{aligned} \Delta y_t &= \alpha\beta' y_{t-1} + \overline{C}\overline{A}(I-\overline{A})^{-1}\tilde{x}_t + \varepsilon_t, \\ \tilde{x}_{t+1} &= \overline{A}\tilde{x}_t + \overline{B}\Delta y_t, \\ \text{s.t.} \quad -\alpha\beta' &= \overline{C}(I-\overline{A})^{-1}\overline{B} + I_p, \quad \text{(SSECM)}. \end{aligned} \tag{2.12}$$

**Remark 1.** *In Ribarits and Hanzon (2014a), an analogous result was obtained in the context of reformulating the model in so-called error correction form. This form was popularized by Engle and Granger (1987). However, the proof given here is different, much shorter and more direct than in Ribarits and Hanzon (2014b). Even in those cases in which an error-correction formulation of the model is unknown, such as in the continuous-discrete model to be studied in chapter 3, the theorem 1 can still be used to translate the cointegration property into a low rank property. See chapter 3 for details.*

**Remark 2.** *The SSECM is a generalization of the well-known error correction formulation in the vector-autoregressive framework.*

**Remark 3.** *It is an illustrative exercise to count the number of free parameters using the representation in terms of the SSECM: There are $n^2 + 2np$ entries in $(\overline{A}, \overline{B}, \overline{C})$ and $2p(p-c)$ entries in $\alpha$ and $\beta$, which makes a total of $n^2 + 2np + 2p(p-c)$ parameters. However, one can apply a state transformation of the form $\hat{x}_t = T\tilde{x}_t$, $T \in Gl(n)$, to obtain an observational equivalent SSECM with $(T\overline{A}T^{-1}, T\overline{B}, \overline{C}T^{-1})$ instead of $(\overline{A}, \overline{B}, \overline{C})$; hence, it is needed to subtract $n^2$ degrees of freedom to determine the dimension of the class of observational distinct models, giving a total of $2np + 2p(p-c)$.*

One can write any VAR model as a state-space model. Consider a $p$-dimensional VAR of order $K$, $y_t = \Pi_1 y_{t-1} + \cdots + \Pi_K y_{t-K} + \varepsilon_t$, which could be written in a more convenient form as $a(z)y_t = \varepsilon_t$, where $a(z) = I - \Pi_1 z - \cdots - \Pi_K z^K$. The *transfer function* $k(z)$ from $(\varepsilon_t)$ to $(y_t)$ is given by $k(z) = a(z)^{-1}$. The state vector $x_t = (y'_{t-1}, \ldots, y'_{t-K})'$ is defined to obtain the following state-space representation:

$$
\begin{aligned}
x_{t+1} &= \begin{pmatrix} \Pi_1 & \Pi_2 & \cdots & \Pi_{k-1} & \Pi_k \\ I_p & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & I_p & 0 \end{pmatrix} x_t + \begin{pmatrix} I_p \\ 0 \\ \vdots \\ 0 \end{pmatrix} \varepsilon_t, \\
y_t &= \left(\Pi_1 \; \Pi_2 \ldots \Pi_{K-1} \; \Pi_K\right) x_t + \varepsilon_t.
\end{aligned}
\tag{2.13}
$$

The state-space representation of a VAR, as in equation (2.13), is widely known as the matrix *companion form of the VAR model*. The inverse transfer function is given by $\overline{k}(z) = a(z)$ and the state-space representation is:

$$
\begin{aligned}
x_{t+1} &= \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ I_p & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & I_p & 0 \end{pmatrix} x_t + \begin{pmatrix} I_p \\ 0 \\ \vdots \\ 0 \end{pmatrix} y_t, \\
\varepsilon_t &= -\left(\Pi_1 \; \Pi_2 \ldots \Pi_{K-1} \; \Pi_K\right) x_t + y_t,
\end{aligned}
\tag{2.14}
$$

where $\varepsilon_t = -\Pi_1 x_t^{(1)} - \cdots - \Pi_k x_t^{(k)} + y_t$.

**Lemma 2.** *The state-space representations in* (2.13) *and* (2.14) *are minimal if and only if* $\operatorname{rk}\Pi_k = p$.

*Proof.* See Ribarits and Hanzon (2014b,a). □

**Remark 4.** *A VEC model can be similarly rewritten as a SSECM, see Ribarits and Hanzon (2014b,a). The foregoing theorem can then also be applied to show that* $\mathrm{rk}(\Pi) = r$*, as was shown in Engle and Granger (1987) by a more complicated argument.*

## 2.3 Maximum Likelihood Estimation of the SSECM

Ribarits and Hanzon (2014b,a) developed the SSECM outlined in the last section in such a way that it can be viewed as a generalization of Johansen's VECM taken into a state-space setting. In doing so several parameters can be optimized analytically. Also, they hoped that such similarities between the methodologies would encourage practitioners to explore state-space modelling as an interesting alternative to the current VAR-based approach. Mirroring the Johansen approach, Ribarits and Hanzon (2014b,a) reduce the constrained non-linear likelihood optimization of the parameters of the SSECM to a low-rank generalized eigenvalue-eigenvector problem. This is achieved by sequentially *concentrating out* a number of the model parameters with respect to the optimization problem, namely $(\alpha, \beta, \overline{C}, \Sigma)$, in a similar fashion to Johansen's approach for VECM. This eventually leaves only the parameters $(\overline{A}, \overline{B})$ (and $\beta_2$ if $n < p$) which require a different method of optimization that is treated in section 2.4. Following this *concentrating out* of parameters $\Sigma(\sigma)$, $\overline{C}$, the resulting reduced rank generalized eigenvalue-eigenvector problem can be solved analytically, assuming that $(\overline{A}, \overline{B})$ (and $\beta_2$ if $n < p$) are known or have been determined. Similar notation to Johansen (1995) is adopted, where: $Z_{0t} = \Delta y_t \in \mathbb{R}^{p\times 1}$, $Z_{1t} = y_{t-1} \in \mathbb{R}^{p\times 1}$ and $Z_{2t} = \tilde{x}_t \in \mathbb{R}^{n\times 1}$. Finding the Gaussian MLE for the parameters of the SSECM involves solving the following non-linear constrained optimisation problem: $\min L(\alpha, \beta, \overline{A}, \overline{B}, \overline{C}, \sigma)$ where

$$L\big(\alpha, \beta, \overline{A}, \overline{B}, \overline{C}, \sigma\big) = \log\det\Sigma + \frac{1}{T}\sum_{t=1}^{T}\Big(Z_{0t} - \alpha\beta' Z_{1t} - \overline{C}\overline{A}(I-\overline{A})^{-1}Z_{2t}\Big)'\Sigma^{-1}\cdot \Big(Z_{0t} - \alpha\beta' Z_{1t} - \overline{C}\overline{A}(I-\overline{A})^{-1}Z_{2t}\Big) \tag{2.15}$$

subject to

$$\overline{C}\Big[(I-\overline{A})^{-1}\overline{B}\Big] = -(I_p + \alpha\beta').$$

**Remark 5.** *The constraint is derived from equation (2.12).*

The optimisation problem can be addressed by performing a series of concentration steps whereby the original constrained optimisation problem of $L(\alpha,\beta,\overline{A},\overline{B},\overline{C},\Sigma)$ is successively reduced to an unconstrained optimisation problem of $L^{4c}$ involving only $(\overline{A},\overline{B})$ (and $\beta_2$ if $n<p$) as:

$$L^{4c}(\overline{A},\ \overline{B}) = \log\det S_{00} + \log\prod_{i=1}^{r}(1-\lambda_i) + p. \tag{2.16}$$

In this equation, $\lambda_1,\ldots,\lambda_r$ denote the $r$ largest eigenvalues of the generalized eigenvalue problem $\det(\lambda S_{11} - S_{10}S_{00}^{-1}S_{01}) = 0$, where $S_{11}$ and $S_{10} = S_{01}'$, will be defined in (2.15). Depending only on $(\overline{A},\ \overline{B})$ and $\beta_2$, we will define empirical variance-covariance matrices obtained from regression residuals in the course of the step-wise concentration process. That said, for a given $(\overline{A},\ \overline{B})$ and $\beta_2$, the set of global optimized parameters $(\widehat{\alpha},\ \widehat{\beta},\ \hat{C},\ \widehat{\Sigma})$ of the original likelihood function $L$ can be explicitly obtained such that the remaining unconstrained optimisation problem involves $(\overline{A},\ \overline{B})$ and $\beta_2$ solely.

### 2.3.1 Reformulation of the Likelihood Optimization Problem

One can set $R = (I_n - \overline{A})^{-1}\overline{B} \in \mathbb{R}^{n\times p}$ to obtain $\overline{C}R = -(I_p + \alpha\beta')$ (if $n \geq p$) or in vectorized form $(R'\otimes I_p)\operatorname{vec}\overline{C} = -\operatorname{vec}(I_p + \alpha\beta')$. Matrix $R$ depends on $(\overline{A},\overline{B})$ only.

To be able to handle the restrictions $\overline{C}R = -(I_p + \alpha\beta')$ we would like $R$ to have full row rank. However, this is not always the case:

- If $n > p$ ($R$ is a tall matrix), then $R$ has full column rank $p$ only if $\overline{B}$ has full column rank $p$.
- If $n = p$ ($R$ is a square matrix), then the number of restrictions is equal to the number of entries in $\overline{C}$ and the rank condition is again satisfied if $\overline{B}$ has full column rank $p = n$.
- If $n < p$ ($R$ is a broad matrix), then there are more restrictions than entries in $\overline{C}$ and consequently, the rank condition $\operatorname{rk}(R'R) = p$ will not be satisfied (for any set of $(\overline{A},\ \overline{B})$ pairs) anymore.

For all three cases, the constrained optimization problem in equation (2.15) can be reformulated as follows:

$\min L(\alpha_1, \beta_1, \alpha_2, \beta_2, \overline{A}, \overline{B}, \overline{C}, \sigma)$ where

$$\begin{aligned} L\big(\alpha_1, \beta_1, \alpha_2, \beta_2, \overline{A}, \overline{B}, \overline{C}, \sigma\big) &= \log\det\Sigma(\sigma) \\ +\frac{1}{T}\sum_{t=1}^{T}&\Big(Z_{0t}-\alpha_1\beta_1'Z_{1t}-\alpha_2\beta_2'Z_{1t}-\overline{CA}(I-\overline{A})^{-1}Z_{2t}\Big)'\Sigma(\sigma)^{-1} \\ &\cdot\Big(Z_{0t}-\alpha_1\beta_1'Z_{1t}-\alpha_2\beta_2'Z_{1t}-\overline{CA}(I-\overline{A})^{-1}Z_{2t}\Big), \end{aligned} \tag{2.17}$$

subject to

$$[\overline{C}, \alpha_2]\widetilde{R} = -(I_p + \alpha_1\beta_1'),$$

with

$$\widetilde{R} = \begin{bmatrix} (I_n - \overline{A})^{-1}\overline{B} \\ \beta_2' \end{bmatrix} \in \mathbb{R}^{(n+k)\times p}, \quad n+k \geq p \quad \text{and } \mathrm{rk}\widetilde{R} = p,$$

where

$$(\alpha_1, \alpha_2) = \alpha, \quad \text{and} \quad (\beta_1', \beta_2') = \beta',$$

where $\alpha_1 \in \mathbb{R}^{p\times(r-k)}$, $\beta_1 \in \mathbb{R}^{p\times(r-k)}$, $\alpha_2 \in \mathbb{R}^{p\times k}$ and $\beta_2 \in \mathbb{R}^{p\times k}$, $\alpha_2$ and $\beta_2$ only appear in case $k > 0$ where $k = \max(0, \min(p-n, r))$ and $r = \mathrm{rk}(\alpha\beta') = p - c$.

Restricting the parameter set $\Theta$ for $(\alpha_1, \beta_1, \alpha_2, \beta_2, \overline{A}, \overline{B}, \overline{C}, \sigma)$ to the set where $\widetilde{R}$ has full column rank leaves an open and dense subset $\Theta^*$ of $\Theta$. Along with the continuity property of the likelihood function, this implies that the infimum over $\Theta$ coincides with the infimum over $\Theta^*$.

**Remark 6.** *Some special cases*

(i) *If $(\alpha_1, \beta_1) = \varnothing$, then $r = k$ and*

$$\big[\overline{C}, \alpha_2\big]\widetilde{R} = -I_p, \quad \widetilde{R} \in \mathbb{R}^{(n+r)\times p}.$$

(ii) *If $(\alpha_2, \beta_2) = \varnothing$, then $k = 0$ and*

$$\big[\overline{C}\big]\widetilde{R} = -\big(I_p + \alpha_1\beta_1'\big), \quad \widetilde{R} \in \mathbb{R}^{n\times p}.$$

The optimisation problem will be solved in various steps. For completeness, we repeat the steps in Ribarits and Hanzon (2014a,b).

### 2.3.2 Concentrating Out $[\overline{C}, \alpha_2]$

For concentrating out $\overline{C}$, similar steps were used as Hanson, however, in this study, we have used a slightly different approach to estimate the bounded parameters. For instance, in the equation below, the second order moment matrix of the corresponding vector is denoted by $Z_{it}$ and $Z_{jt}$, $i, j \in \{0,\ 1,\ 2\}$ by:

$$M_{ij} = \frac{1}{T}\sum_{t=1}^{T} Z_{it}Z'_{jt}. \tag{2.18}$$

**Theorem 2.** *Assume $(\alpha_1, \beta_1, \beta_2, \overline{A}, \overline{B}, \sigma)$ is such that $\widetilde{R}$ satisfies the rank condition in equation (2.17), $(\overline{A}, \overline{B})$ is controllable and $\beta_2 \in \mathbb{R}^{p\times k}$, $k \leq p$, is of full column rank. Furthermore, let*

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} > 0$$

*hold true. Then the unique global constrained minimizer $[\widehat{\overline{C}}, \widehat{\alpha}_2]$ solving*

$$\min_{[\overline{C},\alpha_2]} L\Big(\alpha_1,\ \beta_1,\ \alpha_2,\ \beta_2, \overline{A}, \overline{B}, \overline{C},\ \sigma\Big)$$

*is independent of $\sigma$ ,and $[\widehat{\overline{C}}, \widehat{\alpha}_2]$ is given by:*

$$\begin{aligned}[\widehat{\overline{C}}, \widehat{\alpha}_2] = \bigg(\Big[M_{02}(I-\overline{A})^{-1'}\overline{A'}, M_{01}\beta_2\Big] \\ -\alpha_1\beta_1'\Big[M_{12}(I-\overline{A})^{-1'}\overline{A'}, M_{11}\beta_2\Big]\bigg)H_{11} - (I_p + \alpha_1\beta_1')H_{21},\end{aligned} \tag{2.19}$$

*where*

$$\begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} = \begin{pmatrix} \overline{M} & \overline{R} \\ \overline{R}' & 0_p \end{pmatrix}^{-1}, \tag{2.20}$$

*and*

$$\overline{M} = \overline{A}(I_n - \overline{A})^{-1} M_{22}\big((I_n - \overline{A})^{-1}\big)'\overline{A'},$$

*and*

$$\overline{R} = (I_n - \overline{A})^{-1}\overline{B}.$$

*Note that $H_{11} \in \mathbb{R}^{n\times n}$, $H_{12} \in \mathbb{R}^{n\times p}$, $H_{21} \in \mathbb{R}^{p\times n}$ and $H_{22} \in \mathbb{R}^{p\times p}$.*

*Let the matrices $\widetilde{Z}_{1t}$, $\widetilde{Z}_{2t}$, $R_{0t}$ and $R_{1t}$ be specified as follows:*

$$\widetilde{Z}_{1t} = \beta_2' Z_{1t} \in \mathbb{R}^{k\times 1}, \tag{2.21}$$

$$\widetilde{Z}_{2t} = \overline{A}(I-\overline{A})^{-1} Z_{2t} \in \mathbb{R}^{n\times 1}, \tag{2.22}$$

$$R_{0t} = Z_{0t} - \left(\left[M_{02}(I-\overline{A})^{-1'}\overline{A}', M_{01}\beta_2\right]H_{11} - H_{21}\right)\begin{pmatrix}\widetilde{Z}_{2t}\\ \widetilde{Z}_{1t}\end{pmatrix} \in \mathbb{R}^{p\times 1}, \tag{2.23}$$

$$R_{1t} = Z_{1t} - \left(\left[M_{12}(I-\overline{A})^{-1'}\overline{A}', M_{11}\beta_2\right]H_{11} + H_{21}\right)\begin{pmatrix}\widetilde{Z}_{2t}\\ \widetilde{Z}_{1t}\end{pmatrix} \in \mathbb{R}^{p\times 1}, \quad t = 1,2,\dots,T. \tag{2.24}$$

*Inserting $\widehat{\mathcal{C}} = [\widehat{\overline{C}}, \widehat{\alpha}_2]$ into the likelihood function in equation (2.17), results in:*

$$\begin{aligned} L^{1c}\big(\alpha_1, \beta_1, \beta_2, \overline{A}, \overline{B}, \sigma\big) &:= L\big(\alpha_1, \beta_1, \widehat{\alpha}_2, \beta_2, \overline{A}, \overline{B}, \widehat{\overline{C}}, \sigma\big)\\ &= \log\det\Sigma + \operatorname{tr}\left\{\frac{1}{T}\sum_{t=1}^{T}\Sigma^{-1}\left(Z_{0t} - \alpha_1\beta_1' Z_{1t} - \widehat{\overline{C}}\left[\begin{array}{c}\widetilde{Z}_{2t}\\ \widetilde{Z}_{1t}\end{array}\right]\right)\big(\cdot\big)'\right\}\\ &= \log\det\Sigma + \frac{1}{T}\sum_{t=1}^{T}(R_{0t} - \alpha_1\beta_1' R_{1t})'\Sigma^{-1}(R_{0t} - \alpha_1\beta_1 R_{1t}). \end{aligned} \tag{2.25}$$

*Proof.* See Ribarits and Hanzon (2014b,a). □

The dot in the second bracket in the second row of (2.25) indicates that the same term as in the first bracket appears.

**Remark 7.** *Note that:*

- *If $\overline{M}$ is invertible, then usage of the well-known partitioned matrix inversion formula yields*

$$\begin{pmatrix} H_{11} & H_{12}\\ H_{21} & H_{22}\end{pmatrix} = \begin{pmatrix} \overline{M}^{-1} - \overline{M}^{-1}\overline{R}(\overline{R}'\overline{M}^{-1}\overline{R})^{-1}\overline{R}'\overline{M}^{-1} & \overline{M}^{-1}\overline{R}(\overline{R}'\overline{M}^{-1}\overline{R})^{-1}\\ (\overline{R}'\overline{M}^{-1}\overline{R})^{-1}\overline{R}'\overline{M}^{-1} & -(\overline{R}'\overline{M}^{-1}\overline{R})^{-1}\end{pmatrix}.$$

- *If $p \le n$ then $\widetilde{Z}_{1t}$ does not appear nor do $\alpha_2$ and $\beta_2$.*

**Remark 8.** *This concentration step with respect to $\overline{C}$ and $\alpha_2$ corresponds to the concentration step with respect to $\psi = (\Gamma_1, \dots, \Gamma_{k-1}, \Phi)$ in Johansen's (1995) vector-autoregressive cointegration framework. In such framework, however, there are no relations between $\alpha\beta'$ and the parameters in $\Gamma_i$, $i = 1, 2, \dots, k-1$.*

### 2.3.3 Concentrating Out $\alpha_1$

Let the second order moment matrix corresponding to $R_{it}$ and $R_{jt}$, $i, j \in \{0, 1\}$ be given by:

$$S_{ij} = \frac{1}{T}\sum_{t=1}^{T} R_{it}R'_{jt} \in \mathbb{R}^{p\times p}. \tag{2.26}$$

The following theorem assumes that $\beta_1' S_{11}\beta_1 > 0$ where $\beta_1' S_{11}\beta_1 \in \mathbb{R}^{(r-k)\times(r-k)}$. Note that $r-k < r < p$. The assumption is correct if $S_{11} > 0$ and $\beta_1$ has full column rank.

**Theorem 3.** *Let $(\beta_1, \beta_2, \bar{A}, \bar{B},\ \sigma)$ be given such that $\beta_1' S_{11}\beta_1 > 0$. Then the unique global minimizer solving $\min_{\alpha_1} L^{1c}(\alpha_1, \beta_1, \beta_2, \bar{A},\ \bar{B},\ \sigma)$ is independent of $\sigma$ and is given by:*

$$\widehat{\alpha}_1 = S_{01}\beta_1\left(\beta_1' S_{11}\beta_1\right)^{-1}. \tag{2.27}$$

*Inserting $\widehat{\alpha}_1$ in equation (2.25) results into:*

$$\begin{aligned} L^{2c}\left(\beta_1, \beta_2, \bar{A}, \bar{B},\ \sigma\right) &:= L^{1c}\left(\widehat{\alpha}_1, \beta_1, \beta_2, \bar{A}, \bar{B},\ \sigma\right) \\ = \log\det\Sigma + \frac{1}{T}\sum_{t=1}^{T}&\Big(R_{0t} - S_{01}\beta_1(\beta_1' S_{11}\beta_1)^{-1}\beta_1' R_{1t}\Big)'\Sigma^{-1}(\,\cdot\,). \end{aligned} \tag{2.28}$$

*Proof.* See Ribarits and Hanzon (2014b,a). □

### 2.3.4 Concentrating Out $\sigma$

**Theorem 4.** *Let $(\beta_1, \beta_2, \bar{A}, \bar{B})$ be given such that:*

$$\frac{1}{T}\sum_{t=1}^{T}\Big(R_{0t} - S_{01}\beta_1(\beta_1' S_{11}\beta_1)^{-1}\beta_1' R_{1t}\Big)(\,\cdot\,)' > 0.$$

*Then the unique global minimizer solving $\min\limits_{\sigma:\Sigma>0} L^{2c}(\beta_1, \beta_2, \bar{A}, \bar{B},\ \sigma)$ is given by:*

$$\Sigma(\widehat{\sigma}) = S_{00} - S_{01}\beta_1\left(\beta_1' S_{11}\beta_1\right)^{-1}\beta_1' S_{10}. \tag{2.29}$$

*Inserting $\Sigma(\widehat{\sigma})$ into equation (2.28) yields:*

$$\begin{aligned} L^{3c}\left(\beta_1, \beta_2, \bar{A}, \bar{B}\right) &:= L^{2c}\left(\beta_1, \beta_2, \bar{A}, \bar{B}, \widehat{\sigma}\right) \\ = \log\det&\Big[S_{00} - S_{01}\beta_1(\beta_1' S_{11}\beta_1)^{-1}\beta_1' S_{10}\Big] + p. \end{aligned} \tag{2.30}$$

*Proof.* See Ribarits and Hanzon (2014b,a). □

### 2.3.5 Concentrating Out $\beta_1$

**Lemma 3.** *Let $M$ and $N$ be positive semi-definite symmetric matrices of dimension $p \times p$ such that $Nx = 0$ implies $Mx = 0, x \in \mathbb{R}^p$. Let $\operatorname{rk}(N) = \tilde{n} \leq p$ and $\operatorname{rk}(M) = m \leq p$, then $m \leq \tilde{n}$. Let $\beta \in \mathbb{R}^{p \times r}$, $r \leq m$. The function*

$$f(\beta) = \frac{\det \beta' M \beta}{\det \beta' N \beta}, \quad \det \beta' N \beta \neq 0 \tag{2.31}$$

*is maximised as follows:*

- *Solve the eigenvalue problem $\det(pI - N) = 0$ for the eigenvalues $p_1 \geq \cdots \geq p_{\tilde{n}} > p_{\tilde{n}+1} = \cdots = p_p = 0$ and a corresponding matrix of orthogonal eigenvectors $W = [w_1, \ldots, w_p]$. Define the matrix:*

$$C = [w_1, \ldots, w_{\tilde{n}}] \begin{pmatrix} \sqrt{\frac{1}{p_1}} & & \\ & \ddots & \\ & & \sqrt{\frac{1}{p_{\tilde{n}}}} \end{pmatrix} \in \mathbb{R}^{p \times \tilde{n}} \quad \textit{such that } C'NC = I_{\tilde{n}}.$$

- *Solve the eigenvalue problem $\det(\lambda I - C'MC) = 0$ for the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_{\tilde{n}} \geq 0$ and a matrix of orthogonal eigenvectors $V = [v_1, \cdots, v_{\tilde{n}}] \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$. Define the matrix $D = [d_1, \ldots, d_{\tilde{n}}] = CV \in \mathbb{R}^{p \times \tilde{n}}$ such that $D'MD = \operatorname{diag}(\lambda_1, \ldots, \lambda_{\tilde{n}})$. Assuming that $\lambda_r > \lambda_{r+1}$, a solution of the maximisation problem is given by $\widehat{\beta} = [d_1, \ldots, d_r] \in \mathbb{R}^{p \times r}$ and the maximum value is $f(\widehat{\beta}) = \prod_{i=1}^{r} \lambda_i$. The set of all solutions, which are orthogonal to the null space for $N$, are obtained by $\widehat{\beta} T_r, T_r \in$* $\operatorname{Gl}(r)$.

*Proof.* See Ribarits and Hanzon (2014b,a). □

**Theorem 5.** *Let $r < p$ and let $(\beta_2, \overline{A}, \overline{B})$ be given such that $S_{00} > 0$. Let $S_{11} = WPW' \neq 0$ be an eigenvalue decomposition of $S_{11}$, where $W = [w_1, \ldots, w_p] \in \mathbb{R}^{p \times p}$ is an orthogonal matrix, $P = \operatorname{diag}(p_1, \ldots, p_{p-k}, 0, \ldots, 0) \geq 0$ (in decreasing order), $k \in \{0, 1, .., p-1\}$ and*

$$C := \left[w_1, \ldots, w_{p-k}\right] \operatorname{diag}\left(p_1^{-\frac{1}{2}}, \ldots, p_{p-k}^{-\frac{1}{2}}\right) \in \mathbb{R}^{p \times (p-k)}, \quad \textit{where} \quad p - k > 0.$$

*Furthermore, let* $C'S_{10}S_{00}^{-1}S_{01}C = V\Lambda V'$ *be an eigenvalue decomposition of* $C'S_{10}S_{00}^{-1}S_{01}C$*, where* $V = [v_1, \dots, v_{p-k}] \in \mathbb{R}^{(p-k)\times(p-k)}$*,* $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_{p-k}) \geq 0$ *(in decreasing order) and let* $\lambda_{r-k} > \lambda_{r-k+1}$*. Then all global minimizers solving* $\min_{\beta_1} L^{3c}(\beta_1, \beta_2, \overline{A}, \overline{B})$ *in equation (2.30), which are orthogonal to the null of space* $S_{11}$*, are given by:*

$$\widehat{\beta}_1 = C\big[v_1, \dots, v_{r-k}\big] \cdot \widetilde{T}, \quad \widetilde{T} \in \mathrm{Gl}(r-k). \tag{2.32}$$

*The likelihood function* $L^{4c}(\beta_2, \overline{A}, \overline{B})$ *is obtained as:*

$$L^{4c}(\beta_2, \overline{A}, \overline{B}) := L^{3c}(\widehat{\beta}_1, \beta_2, \overline{A}, \overline{B}) = \log\det S_{00} + \log\prod_{i=1}^{r-k}(1-\lambda_i) + p. \tag{2.33}$$

*Proof.* See Ribarits and Hanzon (2014b,a). ☐

Here, we will summarize all the steps involved. Let observations $\{y_1, \dots, y_T\}$ be given. Assuming that $\tilde{x}_0 = 0$ in the SSECM, one can compute $Z_{0t}$, $Z_{1t}$ and $Z_{2t}$. Additionally, let $(\beta_2, \overline{A}, \overline{B})$ be given such that the conditions specified in Theorem 2 and Theorem 5 above are satisfied. Then the global minimizers of the likelihood function as a function of $(\alpha_1, \alpha_2, \beta_1, \overline{C}, \sigma)$ simultaneously are determined as follows:

- Solve the eigenvalue problems in Theorem 5 to obtain the set of minimizers:

$$\widehat{\beta}_1 = C[v_1, \dots, v_{r-k}] \cdot \widetilde{T}, \quad \widetilde{T} \in \mathrm{Gl}(r-k). \tag{2.34}$$

  All the quantities involved in equation (2.34) depend only on $Z_{0t}$, $Z_{1t}$ $Z_{2t}$ and $(\beta_2, \overline{A}, \overline{B})$.

- Insert $\widehat{\beta}_1$ in equation (2.29) to obtain:

$$\Sigma(\widehat{\sigma})|_{\beta_1=\widehat{\beta}_1} = S_{00} - S_{01}\widehat{\beta}_1\big(\widehat{\beta'}_1 S_{11}\widehat{\beta}_1\big)^{-1}\widehat{\beta'}_1 S_{10}. \tag{2.35}$$

- Similarly, insert $\widehat{\beta}_1$ in equation (2.27) to obtain

$$\widehat{\alpha}_1|_{\beta_1=\widehat{\beta}_1} = S_{01}\widehat{\beta}_1\big(\widehat{\beta'}_1 S_{11}\widehat{\beta}_1\big)^{-1}. \tag{2.36}$$

- Finally, insert $(\widehat{\alpha}_1, \widehat{\beta}_1)$ into equation (2.19) to obtain $[\widehat{\widehat{C}},\ \widehat{\alpha}_2]|_{\alpha_1=\widehat{\alpha}_1,\beta_1=\widehat{\beta}_1}$ as:

$$\Big(\Big[M_{02}(I-\overline{A'})^{-1}\overline{A'},\ M_{01}\beta_2\Big]-\widehat{\alpha}_1\widehat{\beta'}_1\Big[M_{12}(I-\overline{A'})^{-1}\overline{A'}, M_{11}\beta_2\Big] \\ \vdots -(I_p+\widehat{\alpha}_1\widehat{\beta'}_1)\Big)\begin{pmatrix} H_{11} \\ H_{21} \end{pmatrix}. \tag{2.37}$$

## 2.4 Parameterization of Stable $[\overline{B} \mid \overline{A}]$ Pairs and $\beta_2$

The parameterization to be presented here deviates from the original work by Ribarits and Hanzon (2014a) and has a number of advantages: first, each parameter lies in a finite interval $(-\pi/2, \pi/2)$; second, the parameterization is numerically stable; third, the total number of parameters are smaller. In section 2.4.1, the study will describe the case $n \geq p$. In section 2.4.2 further information is provided about how to find the MLE for $\beta_2$ in case $n < p$. In both cases there is no analytic solution available. Thus, the study will use a gradient search algorithm for the optimization. The first step in the process is to specify a parameterization for $[\overline{B} \mid \overline{A}]$ pairs and later on, for $\beta_2$.

### 2.4.1 Parameterization of Stable $[\overline{B} \mid \overline{A}]$ Pairs

The state-space model allows for a choice basis for the state-space. The freedom of basis choice can be used to put $(\bar{A}, \bar{B})$ in a suitable form. In this section, we will present findings from papers by Hanzon et al. (2006, 2007, 2009). It will use a particular form for the $[\overline{B} \mid \overline{A}]$ pairs, namely the so-called input-normal SDPS form which will be explained in this section.

The original problem optimization problem can be transformed to an (unrestricted) optimization problem where $[\overline{B} \mid \overline{A}]$ is *input-normal* and has a particular pivot structure which guarantees that the pair is controllable. $[\overline{B} \mid \overline{A}]$ *input-normal* means that the rows of the matrix $[\overline{B} \mid \overline{A}]$ are orthonormal. Advantages are that:

1. The method will be numerically stable.
2. It is possible to find a good coverage of the parameter space as it is a subset of a closed and bounded set.

#### 2.4.1.1 The Input Normal Form

**Definition 1.** The input normal form of $[\overline{B}\,|\,\overline{A}]$ is characterized by the fact that the rows of the partitioned matrix $[\overline{B}\,|\,\overline{A}]$ are pairwise orthogonal and have a unit length:

$$\left[\overline{B}\,|\,\overline{A}\right]\left[\overline{B}\,|\,\overline{A}\right]^{T}=I_n. \tag{2.38}$$

#### 2.4.1.2 Controllability Matrix

**Definition 2.**

1. For a given pair $[\overline{B}\,|\,\overline{A}]$, the partitioned matrix $K=[\overline{B},\ \overline{AB},\ldots,\overline{A}^{n-1}\overline{B}]$ is called the controllability matrix.
2. The input pair $[\overline{B}\,|\,\overline{A}]$ is said to be controllable provided $K$ has full row rank $n$.

**Theorem 6.** *If $(A,\ B)$ is controllable and $A$ is asymptotically stable, then there exists $T\in GL_n$ such that $(TAT^{-1},\ TB)$ is input normal.*

See Hanzon et al. (2006, 2007, 2009).

#### 2.4.1.3 Pivot Structures

**Definition 3.** Assume $n$ is a positive integer. Consider a vector defined as:

$$v=\Big(v(1),\ v(2),\ldots,\ v(n)\Big)'\in R^n. \tag{2.39}$$

The vector $v$ is called a pivot-$k$ vector if $v(k)>0$ and $v(j)$ with $j>k$ are all zero for some positive integer $k$, $k\le n$.

**Definition 4.** An $(n\times r)$ matrix $M$, $r\ge n$ will be said to have a full pivot structure $J=(j_1,j_2,\ldots,j_n)$ provided for each $k\in\{1,\ 2,\ldots,\ n\}$ the column with index $j_k$ of $M$ is a pivot-$k$ vector.

**Example 1.**

$$M=\begin{bmatrix} * & * & * & * & * & * & + & * \\ + & * & * & * & * & * & 0 & * \\ 0 & * & * & * & + & * & 0 & * \\ 0 & * & + & * & 0 & * & 0 & * \\ 0 & * & 0 & * & 0 & + & 0 & * \end{bmatrix},$$

where $*$ denotes an arbitrary number and $+$ denotes a (strictly) positive number, has a full pivot structure $J = (7,1,5,3,6)$.

#### 2.4.1.4 Sub-Diagonal Pivot Structures (SDPS)

**Definition 5.** Consider the partitioned matrix $[B \mid A]$ in $\mathbb{R}^{n\times(p+n)}$; we say this structure has a sub-diagonal pivot structure if

(1) $[B\,|\,A]$ has a full pivot structure

(2) the prescribed pivot columns of $A$ have the property:

that a column with pivot at position $k$ has column number, $m < k$ in $A$ (hence column number $j_k = p+m < p+k$ in $[B\,|\,A]$ for each $k \in (1,2,\dots,n)$).

**Example 2.**

$$[B\,|\,A] = \left[\begin{array}{c:c} \underbrace{\begin{matrix} + & * \\ 0 & * \\ 0 & + \\ 0 & 0 \end{matrix}}_{B_{n\times p}} & \underbrace{\begin{matrix} * & * & * & * \\ + & * & * & * \\ 0 & * & * & * \\ 0 & + & * & * \end{matrix}}_{A_{n\times n}} \end{array}\right],$$

has a sub-diagonal pivot structure;

**Example 3.**

$$[B\,|\,A] = \left[\begin{array}{c:c} \underbrace{\begin{matrix} + & * \\ 0 & * \\ 0 & + \\ 0 & 0 \end{matrix}}_{B_{n\times p}} & \underbrace{\begin{matrix} * & * & * & * \\ + & * & * & * \\ 0 & + & * & * \\ 0 & 0 & + & * \end{matrix}}_{A_{n\times n}} \end{array}\right],$$

has a sub-diagonal pivot structure;

**Example 4.**

$$[B\,|\,A] = \left[\begin{array}{c:c} \underbrace{\begin{matrix} + & * \\ 0 & * \\ 0 & + \\ 0 & 0 \end{matrix}}_{B_{n\times m}} & \underbrace{\begin{matrix} * & * & * & * \\ * & * & + & * \\ 0 & * & 0 & * \\ * & * & 0 & + \end{matrix}}_{A_{n\times n}} \end{array}\right]$$

does not have a sub-diagonal pivot structure.

So in $A$ matrix in SDPS form

$\longrightarrow$ the diagonal has just $*$'s

$\longrightarrow$ any $+$ comes under the $*$'s on the diagonal and not above

$\longrightarrow$ under any $+$ one can only have zeros.

**Theorem 7.** *If $[\overline{B} \mid \overline{A}]$ has a sub-diagonal pivot structure then the pair $(\overline{A}, \overline{B})$ is controllable (Hanzon et al. 2009).*

Indeed, the two controllable pairs $(A_1, B_1)$ and $(A_2, B_2)$ are called equivalent if and only if there exists a non-singular matrix $T$ such that $(TA_1T^{-1}, TB_1) = (A_2, B_2)$.

**Proposition 1.** *Let the pair $(\overline{B}, \overline{A})$ be controllable and $A$ asymptotically stable. There exists an equivalent input normal pair with sub-diagonal pivot structure (Hanzon et al. 2009).*

#### 2.4.1.5 Orthogonal Hessenberg Matrix

Now note that as $[\overline{B} \mid \overline{A}]$ is in input normal form, so $\overline{B}\overline{B}' + \overline{A}\overline{A}' = I$, we can extend $[\overline{B} \,|\, \overline{A}]$ to a square orthogonal matrix $R = \begin{bmatrix} R_{11} & R_{12} \\ \overline{B} & \overline{A} \end{bmatrix}$. Here $[R_{11} \,|\, R_{12}]$ is determined up to multiplication by an orthogonal matrix on the left. Applying a state-space transformation we can bring $R$ into a particular form. Before doing that we need a theorem as given below.

**Definition 6.** A matrix $\widehat{R} = (\widehat{R}_{ij})_{ij} \in R^{(n+1)(n+1)}$ is said to be upper Hessenberg if $\widehat{R}_{ij} = 0$ for $j+1 < i$. Likewise, $\widehat{R}$ is said to be *positive p-upper Hessenberg* if $\widehat{R}_{ij} = 0$ for $i > j + p$ and $\widehat{R}_{ij} > 0$ for $i = j + p$.

**Theorem 8.** *Consider a matrix $\widehat{R} \in R^{(n+p)(n+p)}$. The following statements are equivalent*

- *$\widehat{R}$ is positive p-upper Hessenberg and orthogonal.*
- *$\widehat{R}$ is equal to the following product of matrices of size $(p + n) \times (p + n)$:*

$$\widehat{R} = \Gamma_n \Gamma_{n-1} \ldots \Gamma_2 \Gamma_1,$$

*where for $k = 1, \ldots, n$:*

$$\Gamma_k = \begin{bmatrix} I_{n-k} & 0 & 0 \\ 0 & V_k & 0 \\ 0 & 0 & I_{k-1} \end{bmatrix},$$

*with $V_k$ an orthogonal $(p+1) \times (p+1)$ matrix block (hence $V_k \times V_k' = I_{p+1}$) specified as*

$$V_k = \begin{bmatrix} v_k & I_p - \sqrt{\left(1 - \|v_k\|^2\right)} \dfrac{v_k v_k'}{\|v_k\|^2} \\ \sqrt{1 - \|v_k\|^2} & -v_k' \end{bmatrix}, \tag{2.40}$$

*where $v_k \in R^p$ and has length $0 < \|v_k\| < 1$, $k = 1, 2, \ldots, n$, or*

$$V_k = \begin{bmatrix} 0 & I_p \\ 1 & 0 \end{bmatrix}, \tag{2.41}$$

*which we associate with $v_k = 0$.*

*Proof.* See Hanzon et al. (2006, 2007, 2009). □

Summarizing results from Hanzon et al. (2006, 2007, 2009), we can make the following remarks:

**Remark 9.**

- *The entries of a positive p-upper Hessenberg matrix which are prescribed to be positive are called the pivots of the matrix. The pivot positions are the corresponding row positions in the matrix. Allowing for column permutations of a positive p-upper Hessenberg matrix, one gets matrices $R^\pi$ with more general pivot positions. Here $\pi$ represents the column permutation used. Now we will say that such a matrix is in SDPS (sub-diagonal pivot structure) form if all the pivots in $R^\pi_{22}$ have their positions in the strict lower triangular part of $R^\pi_{22}$. Take $[\overline{B} \mid \overline{A}] = [0 \mid I]R^\pi = [R^\pi_{21} \mid R^\pi_{22}]$ to find that all the pivots of $R^\pi$ are also in $[\overline{B} \mid \overline{A}]$ and $R^\pi$ is in SDPS form if and only if the pivot positions in $\overline{A}$ are all in the strict lower triangular part of $\overline{A}$. If this is the case, it can be said that $[\overline{B} \mid \overline{A}]$ is in SDPS form.*
- *Any reachable pair $[\overline{B} \mid \overline{A}]$, with $\overline{A}$ asymptotic stable, can be brought into at least one input-normal SDPS form by a state-space transformation. Note that such a*

*state-space transformation will result in a similar state-space transformation for $\widehat{\widehat{C}}$; also, such a transformation does not change the value of $L^{4c}$.*

- *If $[\overline{B} \mid \overline{A}]$ is in SDPS form there is a column permutation $\pi^{-1}$ which brings its first n columns into the form of an upper triangular matrix with positive elements on its diagonal.*
- *Applying the same column permutation to $R^{\pi}$ gives an orthogonal p-upper Hessenberg matrix R.*
- *$[\overline{B} \mid \overline{A}]$ can be obtained as the last n rows of $R^{\pi}$. Therefore, parameterization of orthogonal p-upper Hessenberg matrices also gives rise to a parameterization of input-normal $[\overline{B} \mid \overline{A}]$ pairs in any given SDPS form.*
- *Actually the case in which the pivots in $[\overline{B} \mid \overline{A}]$ are to be on the diagonal of the first n columns of $[\overline{B} \mid \overline{A}]$ is generic meaning that if one would restrict to this case, we will still cover almost all points in the parameter space (an open and dense set). Therefore, in the case studies, this form has been used.*

#### 2.4.1.6 Parameterization of $[\overline{B} \mid \overline{A}]$ Using Angles

The study will parameterize orthogonal positive p-upper Hessenberg matrices as follows: use a well-known parameterization of unit vectors in $\mathbb{R}^p \times \mathbb{R}_{>0}$ using $p$ angles $\theta_{1,k}, \theta_{2,k}, \ldots, \theta_{p,k}$ where all $\theta_{i,k}$ are restricted to $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ to ensure positivity of the last entry. To parameterize a unit vector in $\mathbb{R}^p \times \mathbb{R}_{>0}$, define recursively, for each $k = \{1, 2, \ldots, n\}$,

$$u(p,k) := \begin{bmatrix} \sin(\theta_{p,k}) \\ \cos(\theta_{p,k}) \end{bmatrix} \in \mathbb{R}^2,$$
$$u(h-1,k) := \begin{bmatrix} \sin(\theta_{h-1,k}) * u(h,k) \\ \cos(\theta_{h-1,k}) \end{bmatrix} \in \mathbb{R}^{p-h+2} \quad \text{for} \quad h = p, p-1, \ldots, 2.$$

Note that $\|u(h,k)\| = 1$ for each $h = p, p-1, \ldots, 2, 1$, and

$$\sqrt{1 - \|v_k\|^2} = \cos(\theta_{1,k}) > 0, \quad as \quad -\frac{\pi}{2} < \theta_{1,k} < \frac{\pi}{2},$$

hence

$$\|v_k\|^2 = 1 - \cos^2(\theta_{1,k}) = \sin^2(\theta_{1,k}) \Longrightarrow \|v_k\| = |\sin(\theta_{1,k})| < 1.$$

Now let

$$u(1,k) := \begin{bmatrix} v_k \\ \sqrt{1-\|v_k\|^2} \end{bmatrix} \in \mathbb{R}^{p+1}.$$

Then

$$u(2,k) := \frac{v_k}{\|v_k\|} \in \mathbb{R}^p$$

is the normalised version of $v_k$. Using this, $V_k$ can be viewed as a function $V_k(\theta_k)$ of $\theta_k = (\theta_{1,k}, \theta_{2,k}, \ldots, \theta_{p,k})'$. Note that the case $\theta_{1,k} = 0$ corresponds to $\|v_k\| = 0$. Now use

$$\Gamma_k = \begin{bmatrix} I_{n-k} & 0 & 0 \\ 0 & V_k(\theta_k) & 0 \\ 0 & 0 & I_{k-1} \end{bmatrix},$$

and

$$R = \Gamma_n(\theta_n)\Gamma_{n-1}(\theta_{n-1})\ldots\Gamma_2(\theta_2)\Gamma_1(\theta_1)$$

to obtain $[\overline{B}\,|\,\overline{A}]$ as $[\overline{B}\,|\,\overline{A}] = [0\,|\,I]R$ as a function of $\theta_n, \theta_{n-1}, \ldots, \theta_1$.

This is a novel approach and could also be useful for other estimation problems and applications of the work developed by Hanzon et al. (2006, 2007, 2009). Advantages of this approach emerge especially when doing gradient calculations. This is because (i) the derivative of $V_k(\theta_k)$ can be obtained straight forwardly, as $V_k(\theta_k)$ can be expressed just in terms of simple expressions in terms of sine and cosine functions of $\theta_{i,k}$, $i = 1, 2, \ldots, p$; and (ii) the derivatives of $R$ w.r.t. $\theta_{i,k}$ only require differentiation of $\Gamma_k$, as for every $\hat{k} \neq k$ $\Gamma_{\hat{k}}$ does not depend on $\theta_k$.

### 2.4.2 Parameterization of $\boldsymbol{\beta_2}$

Since, the study only requires $\beta_2 \in \mathbb{R}^{p\times k}$ (with $k = p - n$ so $k < p$) up to right multiplication by $k \times k$ invertible matrices, the column space of $\beta_2$ is invariant. Alternatively, one can say that any two choices for the $\beta_2$ matrix with the same column space are equivalent. One restriction that can be imposed on $\beta_2$ is column ortho-normality. This is similar to the input normality form of $[\overline{B}\,|\,\overline{A}]$. But to ensure that we get a unique or canonical representation for every equivalence class of $\beta_2$, further restrictions need to be imposed as we did earlier for the input normal form. A situation that has similarity to $[\overline{B}\,|\,\overline{A}]$ is this: The matrix $[\overline{B}\,|\,\overline{A}]$ has more columns than rows (so is broad) and so has $\beta_2'$. So we can parameterize $\beta_2'$ with angle parameters in a similar way as we did for $\overline{A}$ and $\overline{B}$. The study uses $\phi$ instead of $\theta$ to denote these angles to avoid confusion.

Recall that $[\overline{B} \mid \overline{A}]$ was obtained as $[\overline{B} \mid \overline{A}] = [0 \mid I]R$ which is a sub-matrix of the product

$$R = \Gamma_n(\theta_n)\Gamma_{n-1}(\theta_{n-1})\cdots\Gamma_2(\theta_2)\Gamma_1(\theta_1);$$

similarly, one obtains $\beta_2'$ as $\beta_2' = [0 \mid I]R_{\beta_2}$ which is a $k \times p = k \times (k+n)$ sub-matrix of the product $R_{\beta_2} = \Gamma_n(\phi_n)\Gamma_{n-1}(\phi_{n-1})\cdots\Gamma_2(\phi_2)\Gamma_1(\phi_1)$. Here each $\Gamma_i(\phi_i)$ is a block diagonal matrix of dimension $p \times p$ depending on a parameter vector $\phi_i$, $i = 1, 2, \ldots, p-k$:

$$\Gamma_i(\phi_i) = \begin{bmatrix} I_{p-k-i} & 0 & 0 \\ 0 & V_i(\phi_i) & 0 \\ 0 & 0 & I_{i-1} \end{bmatrix}. \tag{2.42}$$

Here $V_i(\phi_i)$ is a matrix of dimensions $(k+1) \times (k+1)$ produced from $k$ values in the interval $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. This approach is similar to the one we took for $V_k(\phi_k)$ as a matrix of dimension $(p+1) \times (p+1)$ produced from $p$ values in the interval $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, see the section above.

## 2.5 A Gradient Algorithm

Here, consider the case $n \geq p$. (The case $n < p$ is handled similarly, but involving $\overline{A}$, $\overline{B}$ and $\beta_2$). No closed form solution is known to exist for optimising the likelihood function with respect to $(\overline{A}, \overline{B})$. The pair $(\overline{A}, \overline{B})$ will be parameterized (locally) using a parameter vector $\theta \in \mathbb{R}^d$, where $d(n, p)$ is the dimension of the parameter space. The study will use a gradient algorithm, hence it needs to compute the gradient given by

$$\frac{\partial L^{4c}\left(\overline{A}(\theta), \overline{B}(\theta)\right)}{\partial \theta} \in \mathbb{R}^d.$$

An issue of differentiation can arise as the criterion function may not be open for differentiation at some points as it involves eigenvalues. Let

$$\nabla := \frac{\partial L^{1c}\left(\widehat{\alpha}, \widehat{\beta}, \overline{A}(\theta), \overline{B}(\theta), \widehat{\Sigma(\sigma)}\right)}{\partial \theta} \in \mathbb{R}^d. \tag{2.43}$$

The following theorem is used:

**Theorem 9 (Envelope Theorem).**

1. *Provided that $L_{4c}(\theta)$ can be differentiated at point $\theta$, then $\nabla$ coincides with the gradient vector of $L_{4c}$ at point $\theta$. This is called the envelope theorem.*
2. *Provided that $\nabla \neq 0$ at $\theta$ we can improve the value of $L_{4c}$ by making an appropriately small step in the direction of $-\nabla$.*
3. *Provided $\nabla = 0$ we have reached a critical point of the function $L^{1c}$.*

*Proof.* See Ribarits and Hanzon (2014a) and Wooldridge (2012). □

**Remark 10.** *In relation to (1), even if at a certain point $L^{4c}(\theta)$ can be differentiated, it should be noted that this involves differentiation of eigenvalues and eigenvectors, along with the use of the chain rule, which will be a tedious task. Using the theorem above this can be avoided.*

**Remark 11.** *In relation to (2), if $L^{4c}(\theta)$ is not differentiable at $\theta$, taking up the suggested modified gradient $\nabla$ will guarantee that the original likelihood function value will decrease provided a small step size is chosen.*

A gradient search can now be implemented to find a local optimum of the likelihood from a given starting point. This will allow us to choose many starting points for $(\overline{A}, \overline{B})$ to find a parameter point with a high likelihood value, as a good approximation of the MLE. More specifically, the gradient algorithm can be outlined as follows:

**Step 1.** To initialise the gradient algorithm, choose a starting point in the parameter space. In the algorithm many starting points will be tried, and starting points will be chosen such that a good covering of the parameter space is obtained. This is possible as the parameter space is bounded.

**Step 2.** Next, compute the vector of partial derivatives of the likelihood function $L^{1c}(\alpha, \beta, \overline{A}(\theta), \overline{B}(\theta), \sigma)$ with respect to $(\theta)$ by evaluated at $\beta = \widehat{\beta}$, $\alpha = \widehat{\alpha}$, $\sigma = \widehat{\sigma}$. These are obtained by global minimisation as outlined previously and we obtain the resulting vector of partial derivatives $\nabla$.

**Step 3.** For the case where $\nabla \neq 0$, the algorithm takes a small incremental step in the direction of $-\nabla$ to lower the value of of likelihood function. In case $\nabla = 0$ the algorithm stops as a critical point is reached.

**Remark 12.** *The method is iterative in each iteration the above steps will be carried out.*

**Remark 13.** *The approach followed here circumvents an error in Ribarits and Hanzon (2014a) as the optimization problem in terms of $L^{1c}$ (rather than $L$) is an unconstrained optimization problem to which the envelope theorem can be applied. The envelope theorem applies to unconstrained optimization.*

The partial derivative of the likelihood function $L^{1c}\left(\alpha, \beta, \overline{A}, \overline{B}, \sigma\right)$ with respect to $(\theta_{i,k})$ is written as:

$$\frac{\partial L^1}{\partial \theta_{i,k}} = \frac{\partial L}{\partial \theta_{i,k}} + \sum_{t=1}^{T} \left(\frac{\partial L}{\partial \overline{C}}\right)' \times \frac{\partial \widehat{\overline{C}}}{\partial \theta_{i,k}}. \tag{2.44}$$

The partial derivative of $\dfrac{\partial L}{\partial \overline{C}}$ is written as follows:

$$\frac{\partial L}{\partial \overline{C}} = \frac{2}{T} \sum_{t=1}^{T} \left(-\overline{A}(I-\overline{A})^{-1} Z_{2t}\right)\left(Z_{0t} - \alpha\beta' Z_{1t} - \overline{CA}(I-\overline{A})^{-1} Z_{2t}\right)' \Sigma^{-1}. \tag{2.45}$$

Indeed, the study finds the partial derivative of $\dfrac{\partial L}{\partial \theta_{i,k}}$ which is as follow:

$$\frac{\partial L}{\partial \theta_{i,k}} = \frac{1}{T} \sum_{t=1}^{T} 2 \frac{\partial M_t}{\partial \theta_{i,k}} \Sigma^{-1} M_t. \tag{2.46}$$

Where $M_t = -\widehat{\overline{C}}\overline{A}(I_n - \overline{A})^{-1} Z_{2t}$. In the context of the results that are coming next, it is simpler to write $A_0 := \overline{A}(I_n - \overline{A})^{-1}$ so that $M_t = -\widehat{\overline{C}} A A_0 Z_{2t}$. Next, find $\dfrac{\partial M_t}{\partial \theta_{i,k}}$ as follows:

$$\frac{\partial M_t}{\partial \theta_{i,k}} = -\frac{\widehat{\overline{C}}}{\partial \theta_{i,k}} A_0 Z_{2,t} - \widehat{\overline{C}} \left( \frac{\partial A_0}{\partial \theta_{i,k}} Z_{2,t} + A_0 \frac{\partial Z_{2,t}}{\partial \theta_{i,k}} \right). \tag{2.47}$$

Further, note that:

$$\begin{aligned} \frac{\partial A_0}{\partial \theta_{i,k}} &= \frac{\partial \overline{A}}{\partial \theta_{i,k}} (I_n - \overline{A})^{-1} - \overline{A}(I_n - \overline{A})^{-1} \frac{\partial \overline{A}}{\partial \theta_{i,k}} (I_n - \overline{A})^{-1} \\ &= (I_n - A_0) \frac{\partial \overline{A}}{\partial \theta_{i,k}} (I_n - \overline{A})^{-1}. \end{aligned} \tag{2.48}$$

Next step is to find the partial derivative of the $Z_{2,t}$ series, which is computed through $Z_{2,t} = \overline{A}Z_{2,t-1} + \overline{B}Z_{0,t-1}$. As $Z_{0,t}$ does not depend on $\theta_{i,k}$, we obtain:

$$\frac{\partial Z_{2,t}}{\partial \theta_{i,k}} = \frac{\partial \overline{A}}{\partial \theta_{i,k}} Z_{2,t-1} + \overline{A}\frac{\partial Z_{2,t-1}}{\partial \theta_{i,k}} + \frac{\partial \overline{B}}{\partial \theta_{i,k}} Z_{0,t-1}. \tag{2.49}$$

Likewise, define;

$$\begin{aligned} \frac{\partial M_{j,2}}{\partial \theta_{i,k}} &= \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial Z_{j,t}}{\partial \theta_{i,k}} Z'_{2,t} + Z_{j,t}\left(\frac{\partial Z_{2,t}}{\partial \theta_{i,k}}\right)'\right), \quad \text{and} \\ \frac{\partial M_{2,j}}{\partial \theta_{i,k}} &= \frac{\partial M'_{j,2}}{\partial \theta_{i,k}}, \quad j \in (0,1,2). \end{aligned} \tag{2.50}$$

It should be noted that the series $Z_{0,t}$ and $Z_{1,t}$ are data series and are not dependent on $\theta_{i,k}$; therefore, its partial derivative terms have a value of 0.

Recall the formula (2.19) for $\widehat{\overline{C}}$, and obtain

$$\begin{aligned} \frac{\partial \widehat{\overline{C}}}{\partial \theta_{i,k}} &= \left(\frac{\partial M_{02}}{\partial \theta_{i,k}} - \alpha\beta'\frac{\partial M_{12}}{\partial \theta_{i,k}}\right) A'_0 H_{11} \\ &+ (M_{02} - \alpha\beta' M_{12})\left(\frac{\partial A'_0}{\partial \theta_{i,k}} H_{11} - A'_0\frac{\partial H_{11}}{\partial \theta_{i,k}}\right) - (I_p + \alpha\beta')\frac{\partial H_{21}}{\partial \theta_{i,k}}. \end{aligned} \tag{2.51}$$

In the next step, we compute the parameter derivatives of $H_{j\ell}$, $j,l = 1,\ 2$ which is needed above. The matrices $H_{j\ell}$ is defined by the following equation:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} \overline{M} & \overline{R} \\ \overline{R}' & 0 \end{bmatrix}^{-1}, \tag{2.52}$$

where $\overline{M} = \overline{A}(I_n - \overline{A})^{-1} M_{22}((I_n - \overline{A})^{-1})'\overline{A'} = A_0 M_{22} A'_0$ and $\overline{R} = (I_n - \overline{A})^{-1}\overline{B}$. With the help of the chain rule, one can write down the following expression:

$$\begin{bmatrix} \frac{\partial H_{11}}{\partial \theta_{i,k}} & \frac{\partial H_{12}}{\partial \theta_{i,k}} \\ \frac{\partial H_{21}}{\partial \theta_{i,k}} & \frac{\partial H_{22}}{\partial \theta_{i,k}} \end{bmatrix} = -\begin{bmatrix} \overline{M} & \overline{R} \\ \overline{R}' & 0 \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \overline{M}}{\partial \theta_{i,k}} & \frac{\partial \overline{R}}{\partial \theta_{i,k}} \\ \frac{\partial \overline{R}'}{\partial \theta_{i,k}} & 0 \end{bmatrix} \begin{bmatrix} \overline{M} & \overline{R} \\ \overline{R}' & 0 \end{bmatrix}^{-1}. \tag{2.53}$$

The partial derivatives of $\overline{M}$ and $\overline{R}$ are written in the form of the following equations:

$$\frac{\partial \overline{M}}{\partial \theta_{i,k}} = \frac{\partial A_0}{\partial \theta_{i,k}} M_{22} A_0' + A_0 \frac{\partial M_{22}}{\partial \theta_{i,k}} A_0' + A_0 M_{22} \frac{\partial A_0'}{\partial \theta_{i,k}}, \tag{2.54}$$

and

$$\frac{\partial \overline{R}}{\partial \theta_{i,k}} = -(I_n - \overline{A})^{-1} \frac{\partial \overline{A}}{\partial \theta_{i,k}} (I_n - \overline{A})^{-1} \overline{B} + (I_n - \overline{A})^{-1} \frac{\partial \overline{B}}{\partial \theta_{i,k}}. \tag{2.55}$$

Finally, calculate $\dot{\overline{A}}, \dot{\overline{B}}$ and $\dot{\tilde{x}}_t$ where the dot represents the differentiation with respect to $(\theta_{i,k})$. Firstly, consider the calculations of $\dot{\tilde{x}}_t$. Extended prediction error filters are used to calculate partial derivatives of $\tilde{x}_t$ with respect to the entries in $\theta_{i,k}$ that were derived from the state equation by means of differentiation:

$$\begin{pmatrix} \tilde{x}_{t+1} \\ \dot{\tilde{x}}_{t+1} \end{pmatrix} = \begin{pmatrix} \overline{A}(\theta_{i,k}) & 0 \\ \dot{\overline{A}} & \overline{A}(\theta_{i,k}) \end{pmatrix} \begin{pmatrix} \tilde{x}_t \\ \dot{\tilde{x}}_t \end{pmatrix} + \begin{pmatrix} \overline{B}(\theta_{i,k}) \\ \dot{\overline{B}}(\theta_{i,k}) \end{pmatrix} \Delta y_t. \tag{2.56}$$

Now to find the derivative of matrix $R$ we need to get the derivative of matrix $\Gamma_k$ where we can write:

$$\overline{\Gamma}_{i,k} = \begin{cases} \Gamma_k, & \text{if} \quad i \neq k \\ \partial \Gamma_k / \partial \theta_{i,k}, & \text{if} \quad i = k \end{cases}. \tag{2.57}$$

Therefore we need to find the derivative of matrix $V_k(\theta_k)$, by follow the following steps:

$$\frac{\partial}{\partial \theta_{i,k}} \big( u(p,k) \big) = \begin{bmatrix} \cos(\theta_{p,k}) \\ -\sin(\theta_{p,k}) \end{bmatrix} \in \mathbb{R}^2,$$

$$\frac{\partial}{\partial \theta_{i,k}} \big( u(h-1,k) \big) = \begin{cases} (h \neq k) \Rightarrow \begin{bmatrix} \sin(\theta_{h-1,k}) * \frac{\partial}{\partial \theta_{i,k}} \big[ u(h,k) \big] \\ 0 \end{bmatrix} \\ (h = k) \Rightarrow \begin{bmatrix} \cos(\theta_{h-1,k}) u(h,k) + \sin(\theta_{h-1,k}) \frac{\partial \big( u(h,k) \big)}{\partial \theta_{i,k}} \\ -\sin(\theta_{h-1,k}) \end{bmatrix} \end{cases}$$

$$\frac{\partial \big( u(2,k) \big)}{\partial \theta_{i,k}} = \frac{\partial \left( \frac{v_k}{\|v_k\|} \right)}{\partial \theta_{i,k}}$$

$$\frac{\partial\left(u(1,k)\right)}{\partial\theta_{i,k}}=\frac{\partial\left(\frac{v_k}{\sqrt{1-\|v_k\|^2}}\right)}{\partial\theta_{i,k}}.$$

Furthermore, note that

$$\begin{aligned}&\frac{\partial}{\partial\theta_{i,k}}\left[I-\left(1-\sqrt{1-\|v_k\|^2}\right)\left(\frac{v_k}{\|v_k\|}\right)\left(\frac{v_k}{\|v_k\|}\right)^T\right]\\ &\Longrightarrow\frac{\partial}{\partial\theta_{i,k}}\left[I-\left(1-\sqrt{1-\|v_k\|^2}\right)\left(u(2,k)\right)\left(u(2,k)\right)^T\right]\\ =-&\frac{\partial}{\partial\theta_{i,k}}\left(\sqrt{1-\|v_k\|^2}\right)\left(u(2,k)\right)\left(u(2,k)\right)^T+\left(1-\sqrt{1-\|v_k\|^2}\right)\times\\ &\left[\frac{\partial\left(u(2,k)\right)}{\partial\theta_{i,k}}\left(u(2,k)\right)^T+\left(u(2,k)\right)\left(\frac{\partial\left(u(2,k)\right)}{\partial\theta_{i,k}}\right)^T\right].\end{aligned}$$

Hence, one can express the partial derivatives matrices of $R=\Gamma_n\Gamma_{n-1}\dots\Gamma_2\Gamma_1$ in the following form:

$$\frac{\partial R}{\partial\theta_{i,k}}=\prod_{k=1}^{n}\overline{\Gamma}_{i,k}. \tag{2.58}$$

$\overline{A}$ and $\overline{B}$ are simply sub-matrices of $R$. So a similar fact is true for the partial derivatives of $\partial\overline{A}/\partial\theta_{i,k}$, $\partial\overline{B}/\partial\theta_{i,k}$ and $\partial\overline{R}/\partial\theta_{i,k}$. The partial derivatives of $\overline{A}$ and $\overline{B}$ are obtained by taking out the corresponding sub-matrices of the partial derivatives of $R$.

A flowchart of the algorithm used is given in Figure 2.1. It describes how to optimize $L^c=L^{4c}$. Note that to compute $\nabla L^c$ Theorem 9 ("Envelope Theorem") was used.

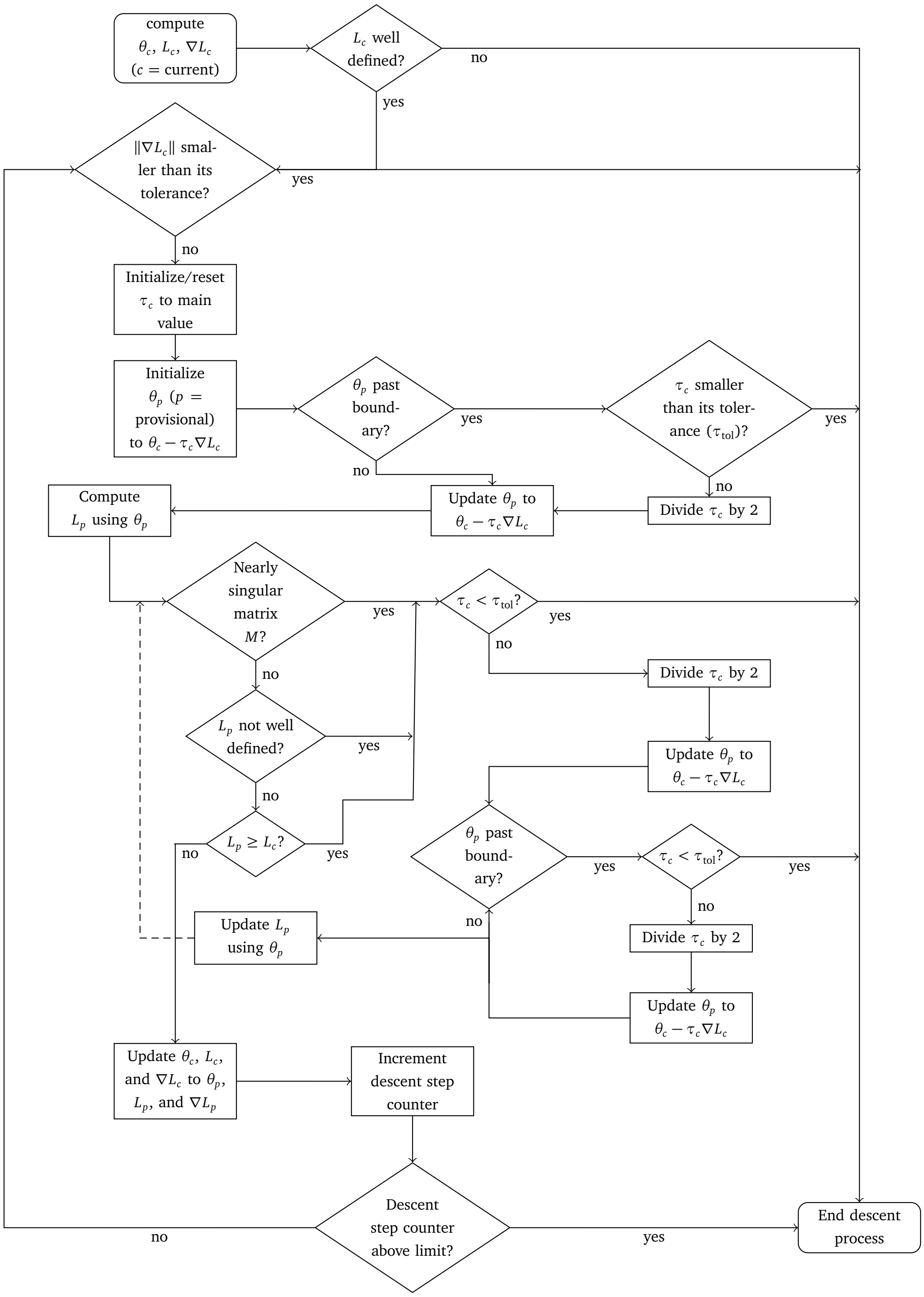



Figure 2.1: A Gradient Search Algorithm

# Chapter 3

# Continuous-time State-Space Model Observed at Irregular Time Points

The goal of this chapter is to calibrate a continuous-time state-space error correction model. The model will utilize a finite number of irregularly dated observations. In this chapter, we try to replicate the method set out in the previous chapter for the case of irregularly spaced observation times. The method again relies on using a particular parametrization of the class of models considered. We apply likelihood optimization to obtain a good estimate for the model. By using this method, the research study aims to parameterize the model. Using our parameterization, a low-rank regression problem will be generated as the parameters are concentrated out sequentially. The model follows an approach in which cointegration is converted into a low-rank constraint on a resulting likelihood optimization problem. The resulting criterion is a function of eigenvalues of a matrix because of the reduced-rank constraint. This results in the problem of computing the derivatives of the parameterized eigenvalues, which is addressed by applying the envelope theorem (which is formulated in the previous chapter). This is followed by the gradient descent method to optimize the remaining model parameters. This approach is significant as it allows for explicit incorporation of the reduced-rank requirement while addressing the eigenvalue derivative issue by applying the envelope theorem.

## 3.1 Specification of the Cointegrated Continuous-time State-Space Model Observed at Irregular Time Points

Consider the following continuous-time stochastic linear state-space model from Hanzon and Ribarits (2008):

$$dX_t = AX_t\,dt + BdW_t, \tag{3.1}$$
$$dY_t = CX_t dt + dW_t, \tag{3.2}$$

where $A$ is assumed to be continuous-time stable, and each element of the spectrum of $A$ is either zero or lies in the open left half plane, and the algebraic and geometric multiplicities of 0 as an eigenvalue of $A$ coincide and are denoted by $c$. $c$ is the number of common trends, and the number $r := p - c$ is called the cointegration rank, namely, the number of linearly independent cointegrating vectors. If $1 \le c \le p-1$, this is a cointegration model of class $I(1)$ for continuous time. $W_t$ denotes a Wiener process with positive definite co-variance matrix $\Sigma$ so that $W_t - W_s$ has a normal distribution with a zero mean value and a co-variance matrix $(t-s)\Sigma$ if $s < t$, which can be rewritten as $W_t - W_s \sim N(0, \Sigma(t-s))$.

Equation (3.1) has the following solution:

Let $\widehat{W}(t-t_1) = W(t) - W(t_1)$ then

$$X_t = e^{A(t-t_1)}X_{t_1} + \int_{t_1}^{t} e^{A(t-s)}Bd\widehat{W}_{s-t_1} \sim N\big(\mu(t_1,t), \Omega(t_1,t)\big), \tag{3.3}$$
$$= e^{A(t-t_1)}X_{t_1} + B\int_{t_1}^{t} d\widehat{W}_{s-t_1} + \int_{t_1}^{t} (t-s)\left[\frac{e^{A(t-s)} - I}{t-s}\right]Bd\widehat{W}_{s-t_1}, \tag{3.4}$$

where $\mu(t,t_1) = e^{A(t-t_1)}X_{t_1}$ and

$$\Omega_x(t,t_1) = B\int_{t_1}^{t} \Sigma ds + \int_{t_1}^{t} \left[e^{A(t-s)} - I\right]B\Sigma B'\left[e^{A'(t-s)} - I\right]ds. \tag{3.5}$$

Solve $\Omega_x(t,t_1)$ using the linear $ds$ matrix equation.

Let's solve the first term of (3.5), replacing $t$ by $t_1 + \Delta t$, $\Delta t > 0$, to get:

$$B\int_{t_1}^{t} \Sigma ds = B\Sigma\Delta t.$$

Next, solve the second term of (3.5). Let $u = t - s$, $0 < u < t - t_1$, to get

$$\int_0^{t-t_1} \Big[e^{Au} - I\Big] B\Sigma B' \Big[e^{A'u} - I\Big] du$$

$$f_{\Delta t} = \int_0^{\Delta t} \Big[e^{Au} - I\Big] B\Sigma B' \Big[e^{A'u} - I\Big] du.$$

As you can easily seen $f_{\Delta t}$ is $O(\Delta t)^3$. Therefore, the equation (3.5) has the final solution:

$$\Omega_x(t, t_1) = B\Sigma\Delta t + O(\Delta t)^2.$$

Now, solve equation (3.2) above using equation (3.3)

$$\begin{aligned} Y_t - Y_{t_1} &= \int_{t_1}^t dY_t = C \int_{t_1}^t X_\tau d\tau + \int_{t_1}^t d\widehat{W}_{\tau - t_1} \\ &= C \left[ \int_{t_1}^t e^{A(\tau - t_1)} X_{t_1} d\tau + \int_{t_1}^t \int_{s=t_1}^{\tau} e^{A(\tau - s)} B d\widehat{W}_{s-t_1} d\tau \right] + \int_{t_1}^t d\widehat{W}_{\tau - t_1}. \end{aligned} \tag{3.6}$$

Let's solve the first term of the right-hand side of the (3.6), taking $\Delta t = t - t_1$

$$\begin{aligned} C \int_{t_1}^t e^{A(\tau - t_1)} X_{t_1} d\tau &= C A^{-1} e^{A(\tau - t_1)} X_{t_1} \Big]_{t_1}^t \\ &= C \Big(A^{-1} e^{A(t-t_1)} - A^{-1}\Big) X_{t_1} \\ &= C A^{-1} \Big(e^{A\Delta t} - I\Big) X_{t_1} \\ &= C A^{-1} \left( \left[ I + A\Delta t + A^2 \frac{\Delta t^2}{2!} + \cdots \right] - I \right) X_{t_1} \\ &= C \left( \Delta t + A \frac{\Delta t^2}{2!} + \cdots + A^{k-1} \frac{\Delta t^k}{k!} + \cdots \right) X_{t_1} \\ &= C X_{t_1} \Delta t + O(\Delta t^2). \end{aligned}$$

Next, solve the second term of the right-hand side of the (3.6)

$$\begin{aligned} C \int_{t_1}^t \int_{t_1}^{\tau} e^{A(\tau - s)} B d\widehat{W}_{s-t_1} d\tau \quad & t_1 < s < \tau < t \\ = C \int_{t_1}^t \Big[A^{-1} e^{A(\tau - s)} B\Big]_s^t d\widehat{W}_{s-t_1} & \\ = C \int_{t_1}^t \Big(A^{-1} e^{A(t-s)} B - A^{-1} B\Big) d\widehat{W}_{s-t_1} & \end{aligned}$$

$$= C\int_{t_1}^{t} A^{-1}\left(e^{A(t-s)} - I\right)Bd\widehat{W}_{s-t_1}$$

$$= C\int_{t_1}^{t} A^{-1}\left[\left(I + A(t-s) + A^2\frac{(t-s)^2}{2!} + \cdots\right) - I\right]Bd\widehat{W}_{s-t_1}.$$

If $|t-s| \leq \Delta t$ so that $(t-s)^k$, $k \geq 2$ terms can be put into $O(\Delta t^2)$, then this quantity is approximately equal to

$$= C\int_{t_1}^{t}(t-s)Bd\widehat{W}_{s-t_1} + O(\Delta t^2).$$

Consequently

$$\int_{t_1}^{t} dY_t = CX_t\Delta t + C\int_{t_1}^{t}(t-s)Bd\widehat{W}_{s-t_1} + \int_{t_1}^{t} d\widehat{W}_{s-t_1} + O(\Delta t^2)$$
$$= CX_t\Delta t + \int_{t_1}^{t}\left(C(t-s)B + 1\right)d\widehat{W}_{s-t_1} + O(\Delta t^2)$$

the expression follows a normal distribution with a zero mean value and covariance matrix

$$\Omega_y(t,t_1) = \int_{t_1}^{t}\left[C(t-s)B + I\right]\Sigma\left[C(t-s)B + I\right]^T ds.$$

Note that, by considering that

$$\frac{\partial}{\partial s}\left(-\frac{1}{3}(t-s)^3\right) = (t-s)^2.$$

The following solution is achieved

$$\begin{aligned}\Omega_y(t,t_1) = \Omega_y(\Delta t) &= -\frac{C}{3}(\Delta t)^3 B\Sigma B^T C^T + \Delta t\Sigma + O(\Delta t^2)\\ &-\frac{1}{2}\left[\left(CB\Sigma + (CB\Sigma)'\right)\right](\Delta t)^2 = \Delta t\Sigma + O(\Delta t^2).\end{aligned} \tag{3.7}$$

Therefore, after solving equations (3.1) and (3.2), the following two expressions are derived (neglecting $O(\Delta t^2)$):

$$X_{t+\Delta t} = e^{A\Delta t}X_t + B\sqrt{\Delta t}\varepsilon_t, \tag{3.8}$$

$$Y_{t+\Delta t} - Y_t = CX_t\Delta t + \sqrt{\Delta t}\varepsilon_t, \tag{3.9}$$

where $\varepsilon_t \sim N(0, \Omega(\Delta t))$, and $\Omega(\Delta t) = \Sigma + O(\Delta t)$. After introducing $t_1, t_2, \ldots$ such that $t_1 < t_2 < \cdots$ and $\Delta t_j = t_{j+1} - t_j$. Consider equations (3.8) and (3.9) at time $t_j$ and let $Z_{t_j} = Y_{t_{j+1}} - Y_{t_j}$, get;

$$X_{t_{j+1}} = e^{A\Delta t_{j+1}} X_{t_j} + B\sqrt{\Delta t_{j+1}}\varepsilon_{t_j}, \tag{3.10}$$
$$Z_{t_j} = CX_{t_j}\Delta t_{j+1} + \sqrt{\Delta t_{j+1}}\varepsilon_{t_j}, \tag{3.11}$$

where $\varepsilon_{t_j} \sim N(0, \Omega(\Delta t_{j+1}))$ for all $j$ and $X_0 = 0$. Note, however, that due to the factor $\Delta t_{j+1}$, the system is now time dependent unlike the one in Ribarits and Hanzon (2014b,a). Equations (3.10) and (3.11) look like a discrete-time state-space system as in Ribarits and Hanzon (2014b,a), where $X_{t_j}$ is an $n$-dimensional unobserved state vector, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{p\times n}$ are parameter matrices, and $Z_{t_j}$ is an observed $p$-dimensional output. Additionally, $\varepsilon_{t_j}$ is a $p$-dimensional (weak) white noise process. Particularly, $\mathbb{E}\varepsilon_{t_j} = 0$ and $\mathbb{E}\varepsilon_{t_j}\varepsilon'_{t_j} = \Sigma$ if the $O(\Delta t_j)$ term is ignored in the co-variance matrix of $\varepsilon_{t_j}$. Find the inverse $\{Z_{t_j}\} \to \{\varepsilon_{t_j}\}$ in state-space form, which is easy to find using $\sqrt{\Delta t_{j+1}}\varepsilon_{t_j} = Z_{t_j} - \Delta t_{j+1} C X_{t_j}$ based on equation (3.11), and by using equation (3.9):

$$X_{t_{j+1}} = \left[e^{A\Delta t_{j+1}} - BC\Delta t_{j+1}\right] X_{t_j} + BZ_{t_j}, \tag{3.12}$$
$$\sqrt{\Delta t_{j+1}}\varepsilon = -C\Delta t_{j+1} X_{t_j} + Z_{t_j}. \tag{3.13}$$

Use $e^{A\Delta t_{j+1}} = I + A\Delta t_{j+1} + O(\Delta t_{j+1})^2$ and approximate $e^{A\Delta t_{j+1}}$ by $I + A\Delta t_{j+1}$. This is an important step for our approach to work, and obtain:

$$X_{t_{j+1}} = \left[I + (A - BC)\Delta t_{j+1}\right] X_{t_j} + BZ_{t_j}, \tag{3.14}$$
$$\sqrt{\Delta t_{j+1}}\varepsilon = -C\Delta t_{j+1} X_{t_j} + Z_{t_j}. \tag{3.15}$$

Use the following notation:

$$\begin{aligned} \overline{A}_c &:= A - BC, & \overline{B}_c &:= B =: \overline{B}_d, \\ \overline{C}_c &:= -C =: \overline{C}_d, & \overline{A}_{d,j+1} &:= I + \overline{A}_c\ \Delta t_{j+1}. \end{aligned} \tag{3.16}$$

Using this notation, one finds

$$X_{t_{j+1}} = \overline{A}_{d,j+1}\, X_{t_j} + \overline{B}_d Z_{t_j}, \tag{3.17}$$
$$\sqrt{\Delta t_{j+1}}\varepsilon_{t_j} = \overline{C}_d \Delta t_{j+1} X_{t_j} + Z_{t_j}. \tag{3.18}$$

## 3.2 Maximum Likelihood Estimation for the Model

To estimate the model parameters, we will use a maximum likelihood approach. To maximize the likelihood initially we follow Ribarits and Hanzon (2014a), and optimize w.r.t. the parameters $\alpha, \beta, \overline{C_c}$ and $\sigma$ where $\Sigma = \Sigma(\sigma)$ is parameterized by the parameter vector $\sigma$. The partially optimized (or "concentrated") likelihood that is obtained after partial optimization by $\overline{C_c}$ is denoted as $L^{1c}(\alpha, \beta, \overline{A_c}, \overline{B_c}, \sigma)$. If $n \geq p$ the likelihood function that is obtained after further partial optimization by $\alpha, \beta, \sigma$ is denoted by $L^{4c}(\overline{A_c}, \overline{B_c})$. If $n < p$ it will be necessary to partition $\alpha$ and $\beta$ as $\alpha = [\alpha_1, \alpha_2]$, $\beta = [\beta_1, \beta_2]$. Then the likelihood function that is obtained after partial optimization by $\alpha, \beta, \sigma$ will be denoted by $L^{4c}(\overline{A_c}, \overline{B_c}, \beta_2)$. This leaves the optimization w.r.t. $\overline{A_c}, \overline{B_c}$ and in case $n < p$, $\beta_2$. For that, we have to follow an approach different to Ribarits and Hanzon (2014a). Using Theorem 1, we can replace the cointegration condition $\mathrm{rk}(A) = n - c$ by $\mathrm{rk}(\overline{C}_c(\overline{A}_c)^{-1}\overline{B}_c + I_p) = p - c (= r)$. This equivalent to the existence of $\alpha, \beta \in \mathbb{R}^{p \times (p-c)}$ both of full rank, such that

$$\exists \alpha, \beta \quad \text{such that} \quad \alpha\beta' = \overline{C}_c(\overline{A}_c)^{-1}\overline{B}_c + I_p. \tag{3.19}$$

Recall that the co-variance matrix of $\varepsilon_{t_j}$ is denoted by $\Sigma$ and let $\Sigma$ be unrestricted, except for the requirement that $\Sigma > 0$. Assume there are no cross-restrictions between the co-variance parameter $\Sigma$ and the parameters for $(\overline{A}_c, \overline{B}_c, \overline{C}_c)$, $\alpha$ and $\beta$.

Equation (3.18) will be rewritten as follows:

$$\varepsilon_{t_j} = \overline{C}_c \sqrt{\Delta t_{j+1}} X_{t_j} + \frac{1}{\sqrt{\Delta t_{j+1}}} Z_{t_j}. \tag{3.20}$$

Let $F_{1j} := Z_{t_j}$ and $F_{2j} := X_{t_j}$. Suppose, there are a finite number of observations $Y_t$ at times $t = t_0, t_1, t_2, \ldots, t_T, t_0 < t_1 < t_2 < \cdots < t_T$, and hence, one knows $Z_t$ at times $t_1, t_2, \ldots, t_T$. The likelihood function can then be written as:

$$P\big(\{Z_{t_j}\}; \overline{A}_c, \overline{B}_c, \overline{C}_c, \alpha, \beta, \Sigma\big) = \frac{1}{2\pi\sqrt{\det(\Sigma)^T}} e^{-\sum_j \frac{1}{2}\varepsilon'_{t_j}\Sigma^{-1}\varepsilon_{t_j}}. \tag{3.21}$$

Hence the scaled negative log likelihood function is $L$, as per the data:

$$\begin{aligned} L(\alpha,\beta,\overline{A}_c,\overline{B}_c,\overline{C}_c,\Sigma) &= -\frac{2}{T}\log P\Big(\{Z_{t_j}\};\overline{A}_c,\overline{B}_c,\overline{C}_c,\alpha,\beta,\Sigma\Big) = \\ &= \text{Constant} + \log\det(\Sigma) + \frac{1}{T}\sum_j \varepsilon'_{t_j}\Sigma^{-1}\varepsilon_{t_j}. \end{aligned} \tag{3.22}$$

Now by using equation (3.20)

$$\begin{aligned} &L(\alpha,\beta,\overline{A}_c,\overline{B}_c,\overline{C}_c,\Sigma) = \text{Constant} + \log\det(\Sigma) \\ &+\frac{1}{T}\sum_j\left(\overline{C}_c\sqrt{\Delta t_{j+1}}F_{2j} + \frac{1}{\sqrt{\Delta t_{j+1}}}F_{1j}\right)\Sigma^{-1}(\cdot)', \end{aligned} \tag{3.23}$$

where the dot in the second bracket indicates that the same term as appeared in the first bracket.

### 3.2.1 Concentrating Out $[\overline{C}_c,\, \alpha_2]$

Considering the restriction in equation (3.19), one can write $R := (-\overline{A}_c)^{-1}\overline{B}_c \in \mathbb{R}^{n\times p}$ to obtain

$$\left[\overline{C}_c,\alpha_2\right]\begin{bmatrix} R \\ \beta_2' \end{bmatrix} = I_p - \alpha_1\beta_1'.$$

It is important to note that matrix $R$ depends on $(\overline{A}_c,\overline{B}_c)$ only. Let $\overline{A}_c$,$\overline{B}_c$, $\alpha_1$, $\beta_1$, $\Sigma$ remain fixed and optimize with respect to $\left[\overline{C}_c,\alpha_2\right]$.

**Theorem 10.** *Let $(\overline{A}_c,\overline{B}_c,\alpha_1,\beta_1,\Sigma)$ be given, and let*

$$M_{12} = \left(\frac{2}{T}\right)\sum_j F_{1j}F'_{2j}, \quad M_{22} = \left(\sum_j \Delta t_{j+1}F_{2j}F'_{2j}\right),$$

*and assume $M_{22}$ invertible, and let $P = R^T M_{22}^{-1}R + \beta_2\beta_2' \in \mathbb{R}^{p\times p}$ and $Q = (\beta_2'P^{-1}\beta_2)^{-1}\beta_2'P^{-1} \in \mathbb{R}^{k\times p}$. Let $\left[\widehat{\overline{C}_c},\widehat{\alpha_2}\right]$ be obtained by optimizing $L$ as given by (3.23) under the linear constraint (3.19), then $\widehat{\overline{C}_c}$ is given by:*

$$\widehat{\overline{C}_c} = \left[\frac{-T}{2}M_{12} + \left[I_p - \alpha_1\beta_1 + \frac{T}{2}M_{12}M_{22}^{-1}R\right]\left[P^{-1} - P^{-1}\beta_2 Q\right]R^t\right]M_{22}^{-1}, \tag{3.24}$$

*and $\widehat{\alpha_2}$ is given by*

$$\widehat{\alpha_2} = \left[I_p - \alpha_1\beta_1 + \frac{T}{2}M_{12}M_{22}^{-1}R\right]Q'. \tag{3.25}$$

*Proof.* As (3.23) is quadratic in $\overline{C}_c$ and the constraint (3.19) is linear in $\overline{C}_c$, the first order conditions of the constrained optimization problem are linear and can be solved in a straight forward manner. The details are described in the following steps.

**Step 1.** Let the Lagrangian $V$ be given by

$$V = L - tr\left[\Lambda\left[\left[\overline{C}_c, \alpha_2\right]\begin{bmatrix} R \\ \beta_2' \end{bmatrix} - I_p - \alpha_1\beta_1'\right]'\right].$$

**Step 2.** Solve for $\overline{C}_c$ using the Lagrangian multiplier matrix $\Lambda \in \mathbb{R}^{\mathbb{p}\times\mathbb{p}}$ and using the first order conditions ($FOC$) of the Lagrangian wrt $\overline{C}_c$

$$\frac{\partial V}{\partial \overline{C}_c} = \Sigma^{-1}\left(\frac{2}{T}\right)\sum_j\left[\overline{C}_c\Delta t_{j+1}F_{2j}F_{2j}' + F_{1j}F_{2j}'\right] - \Lambda R^T = 0$$

multiply by $\Sigma$

$$\begin{aligned}
&\Longleftrightarrow \left(\frac{2}{T}\right)\sum_j\left[\overline{C}_c\Delta t_{j+1}F_{2j}F_{2j}' + F_{1j}F_{2j}'\right] - \Sigma\Lambda R^T = 0\\
&\Longleftrightarrow \left(\frac{-2}{T}\right)\overline{C}_c\sum_j\Delta t_{j+1}F_{2j}F_{2j}' = \left(\frac{-2}{T}\right)\sum_j F_{1j}F_{2j}' - \Sigma\Lambda R^T\\
&\Longleftrightarrow \overline{C}_c = -\frac{T}{2}\left[\left(\frac{-2}{T}\right)\sum_j F_{1j}F_{2j}' - \Sigma\Lambda R^T\right]\left[\sum_j\Delta t_{j+1}F_{2j}F_{2j}'\right]^{-1}.
\end{aligned}$$

**Step 3.** Substitute the formula found for $\overline{C}_c$ in the constraint. This gives a linear equation in terms of $\Lambda$ and $\alpha_2$

$$\frac{\partial V}{\partial \Lambda} = \left[\overline{C}_c \vdots \alpha_2\right]\begin{bmatrix} R \\ \beta_2' \end{bmatrix} - (I_p - \alpha_1\beta_1') = 0$$

$$\frac{\partial V}{\partial \Lambda} =$$

$$\left[-\frac{T}{2}\left[\left(\frac{2}{T}\right)\sum_j F_{1j}F_{2j}' - \Sigma\Lambda R^T\right]\left[\sum_j\Delta t_{j+1}F_{2j}F_{2j}^T\right]^{-1} \vdots \alpha_2\right]\overbrace{\begin{bmatrix} R \\ \beta_2' \end{bmatrix}}^{\tilde{R}} - (I_p - \alpha_1\beta_1') = 0.$$

**Step 4.** Write down the *FOC* of the Lagrangian w.r.t. $\alpha_2$. This gives a linear equation in terms of lambda

$$\frac{\partial V}{\partial \alpha_2} = -\Lambda\beta_2 = 0.$$

**Step 5.** Solve the joint linear equations obtained in step 3 and 4 to obtain an expression for $\Lambda$ and $\alpha_2$.

Let $M_{12} = \left(\frac{2}{T}\right)\sum_j F_{1j}F'_{2j}$ and $M_{22} = \left(\sum_j \Delta t_{j+1}F_{2j}F'_{2j}\right)$. Then we obtain:

$$\Longleftrightarrow -\frac{T}{2}M_{12}M_{22}^{-1}R + \frac{T}{2}\Sigma\Lambda R^T M_{22}^{-1}R + \alpha_2\beta_2^T - (I_p - \alpha_1\beta_1') = 0.$$

From $-\Lambda\beta_2 = 0$ it follows that $\left(\frac{T}{2}\Sigma\Lambda\right)\beta_2 = 0$. Let $\widetilde{\Lambda} = \frac{T}{2}\Sigma\Lambda$. Therefore,

$$\begin{gathered}
\Longleftrightarrow -\frac{T}{2}M_{12}M_{22}^{-1}R + \widetilde{\Lambda}R^T M_{22}^{-1}R + \alpha_2\beta_2^T - (I_p - \alpha_1\beta_1') = 0 \\
\text{and} \quad \widetilde{\Lambda}\beta_2 = 0 \Longleftrightarrow \\
\Longleftrightarrow \widetilde{\Lambda}R^T M_{22}^{-1}R + \alpha_2\beta_2^T = I - \alpha_1\beta_1 + \frac{T}{2}M_{12}M_{22}^{-1}R \\
\text{and} \quad \widetilde{\Lambda}\beta_2 = 0 \Longleftrightarrow \\
\Longleftrightarrow [\widetilde{\Lambda} \vdots \alpha_2]\begin{bmatrix} R^T M_{22}^{-1}R & \beta_2 \\ \beta_2^T & 0 \end{bmatrix} = \left[I - \alpha_1\beta_2 + \frac{T}{2}M_{12}M_{22}^{-1}R \vdots 0\right].
\end{gathered}$$

We obtain the following expressions, assuming that the bordered matrix is invertible:

$$[\widetilde{\Lambda} \vdots \alpha_2] = \left[I - \alpha_1\beta_1 + \frac{T}{2}M_{12}M_{22}^{-1}R \vdots 0\right]\begin{bmatrix} R^T M_{22}^{-1}R & \beta_2 \\ \beta_2^T & 0 \end{bmatrix}^{-1}$$

$$\begin{bmatrix} R^T M_{22}^{-1}R & \beta_2 \\ \beta_2' & 0 \end{bmatrix} = \begin{bmatrix} R^T M_{22}^{-1}R + \beta_2\beta_2' & \beta_2 \\ \beta_2' & 0 \end{bmatrix} \times \begin{bmatrix} I & 0 \\ -\beta_2' & I \end{bmatrix} \Longrightarrow$$

$$\begin{bmatrix} R^T M_{22}^{-1}R & \beta_2 \\ \beta_2' & 0 \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ \beta_2' & I \end{bmatrix}\begin{bmatrix} R^T M_{22}^{-1}R + \beta_2\beta_2' & \beta_2 \\ \beta_2' & 0 \end{bmatrix}^{-1}.$$

Let $P = R^T M_{22}^{-1} R + \beta_2 \beta_2'$, where $M_{22}$ is symmetric and invertible. Use the Partitioned inverse matrix as follows:

$$\begin{bmatrix} R^T M_{22}^{-1} R + \beta_2 \beta_2' & \beta_2 \\ \beta_2' & 0 \end{bmatrix}^{-1} = \begin{bmatrix} P^{-1} - P^{-1}\beta_2(\beta_2' P^{-1}\beta)^{-1}\beta_2^T P^{-1} & \left[\left(\beta_2' P^{-1}\beta_2\right)^{-1}\beta_2' P^{-1}\right]' \\ \left(\beta_2' P^{-1}\beta_2\right)^{-1}\beta_2' P^{-1} & -\left(\beta_2' P^{-1}\beta_2\right)^{-1} \end{bmatrix}$$

$$\text{Therefore, } \begin{bmatrix} R^T M_{22}^{-1} R & \beta_2 \\ \beta_2' & 0 \end{bmatrix}^{-1} =$$

$$= \begin{bmatrix} I_r & 0 \\ \beta_2' & I_{p-n} \end{bmatrix} \begin{bmatrix} P^{-1} - P^{-1}\beta_2(\beta_2' P^{-1}\beta)^{-1}\beta_2^T P^{-1} & \left[\left(\beta_2' P^{-1}\beta_2\right)^{-1}\beta_2' P^{-1}\right]' \\ \left(\beta_2' P^{-1}\beta_2\right)^{-1}\beta_2' P^{-1} & -\left(\beta_2' P^{-1}\beta_2\right)^{-1} \end{bmatrix}$$

$$\begin{bmatrix} R^T M_{22}^{-1} R & \beta_2 \\ \beta_2' & 0 \end{bmatrix}^{-1} = \begin{bmatrix} P^{-1} - P^{-1}\beta_2(\beta_2' P^{-1}\beta)^{-1}\beta_2^T P^{-1} & \left[\left(\beta_2' P^{-1}\beta_2\right)^{-1}\beta_2' P^{-1}\right]' \\ \left(\beta_2' P^{-1}\beta_2\right)^{-1}\beta_2' P^{-1} & I_{p-n} - \left(\beta_2' P^{-1}\beta_2\right)^{-1} \end{bmatrix}.$$

Let $Q = \left(\beta_2' P^{-1}\beta\right)^{-1}\beta_2' P^{-1}$, therefore, the last equation can be written as:

$$\begin{bmatrix} R^T M_{22}^{-1} R & \beta_2 \\ \beta_2' & 0 \end{bmatrix}^{-1} = \begin{bmatrix} P^{-1} - P^{-1}\beta_2 Q & Q' \\ Q & I_{p-n} - \left(\beta_2' P^{-1}\beta_2\right)^{-1} \end{bmatrix}.$$

It now follows that,

$$\left[\widetilde{\Lambda} \vdots \alpha_2\right] = \left[I_p - \alpha_1\beta_1 + \frac{T}{2} M_{12} M_{22}^{-1} R \vdots 0\right] \times \begin{bmatrix} P^{-1} - P^{-1}\beta_2 Q & Q' \\ Q & I_{p-n} - \left(\beta_2' P^{-1}\beta_2\right)^{-1} \end{bmatrix}$$

and so,

$$\begin{aligned} qd\widetilde{\Lambda} &= \left(I_p - \alpha_1\beta_1 + \frac{T}{2} M_{12} M_{22}^{-1} R\right)\left[P^{-1} - P^{-1}\beta_2 Q\right], \\ \alpha_2 &= \left(I_p - \alpha_1\beta_1 + \frac{T}{2} M_{12} M_{22}^{-1} R\right) Q. \end{aligned}$$

Note that this proves (3.25).

**Step 6.** As $\Lambda = \frac{2}{T}\Sigma^{-1}\widetilde{\Lambda}$, therefore, we obtain:

$$\Lambda = \frac{2}{T}\Sigma^{-1}\left(I_p - \alpha_1\beta_1 + \frac{T}{2} M_{12} M_{22}^{-1} R\right)\left[P^{-1} - P^{-1}\beta_2 Q\right].$$

Substitute the expression found for $\Lambda$ into the formula for $\overline{C}_c$ found in step 2, then we find that $\widehat{\overline{C}_c}$ is given by:

$$\widehat{\overline{C}_c} = \Big[\frac{-T}{2}M_{12} + \big[I_p - \alpha_1\beta_1 + \frac{T}{2}M_{12}M_{22}^{-1}R\big]\big[P^{-1} - P^{-1}\beta_2 Q\big]R^t\Big]M_{22}^{-1}.$$

This proves equation (3.24).

**Step 7.** The study found $\widehat{\overline{C}_c}$ and $\widehat{\alpha_2}$.

□

Substituting $\widehat{\overline{C}_c}$ into the likelihood function as given in (3.23), one obtains:

$$\begin{aligned}
L^{1c}\big(\alpha,\beta,\overline{A_c},\overline{B_c},\Sigma\big) &:= L\big(\alpha,\beta,\overline{A_c},\overline{B_c},\widehat{\overline{C}_c},\Sigma\big)\\
&= \log\det(\Sigma) + \operatorname{tr}\left\{\frac{1}{T}\sum_{j=0}\left(\widehat{\overline{C}_c}\sqrt{\Delta t_{j+1}}F_{2j} + \frac{1}{\sqrt{\Delta t_{j+1}}}F_{1j}\right)\Sigma^{-1}(\cdot)'\right\}\\
&= \log\det(\Sigma) + \frac{1}{T}\sum_{j=0}\Big[\Big[\frac{-T}{2}M_{12} + \Big[I_p - \alpha_1\beta_1 + \frac{T}{2}M_{12}M_{22}^{-1}R\Big]\times \qquad (3.26)\\
&\quad \big[P^{-1} - P^{-1}\beta_2 Q\big]R^t\Big]M_{22}^{-1}\Big]\times\left(\sqrt{\Delta t_{j+1}}F_{2j} + \frac{1}{\sqrt{\Delta t_{j+1}}}F_{1j}\right)\Sigma^{-1}(\cdot)'\\
&= \log\det(\Sigma) + +\frac{1}{T}\sum_{j=0}\big(R_{0j} - \alpha_1\beta_1' R_{1j}\big)\Sigma^{-1}(\cdot)',
\end{aligned}$$

where, the dot in the second pair brackets indicates that the same term as in the first pair brackets appears and where

$$\begin{aligned}
R_{0j} &= \frac{-T}{2}M_{12}M_{22}^{-1}\sqrt{\Delta t_{j+1}}F_{2j} + (I + \frac{T}{2}M_{12}M_{22}^{-1}R')R_{1j},\\
R_{1j} &= \Big[\Big[\big[P^{-1} - P^{-1}\beta_2 Q\big]R^t\Big]M_{22}^{-1}\Big]\sqrt{\Delta t_{j+1}}F_{2j}, \quad j = 1,2.
\end{aligned} \tag{3.27}$$

### 3.2.2 Concentrating Out $\alpha_1$

In this step, the study will partially optimize $\alpha$. Denote the product moment matrices corresponding to $R_{it}$ and $R_{gt}$, where $i, g \in \{0,1\}$ by:

$$S_{ig} = \frac{1}{T}\sum_{t=1}^{T} R_{it}R'_{gt}. \tag{3.28}$$

The study assumes that $\beta' S_{ii}\beta > 0$, which is typically no restriction of generality as argued by Ribarits and Hanzon (2014a).

Therefore, by Ribarits and Hanzon (2014a), we find $\widehat{\alpha} = S_{01}\beta(\beta' S_{11}\beta)^{-1}$.

After inserting $\widehat{\alpha}$ into $L^{1C}$, get:

$$\begin{aligned} L^{2c}(\beta,\overline{A_c},\overline{B_c},\sigma) &:= L^{1c}(\widehat{\alpha},\beta,\overline{A_c},\overline{B_c},\Sigma) \\ = \log\det(\Sigma) &+ \frac{1}{T}\sum_{j=1}^{T}\Big(R_{0j} - S_{01}\beta(\beta' S_{11}\beta)^{-1}\beta R_{1j}\Big)\Sigma^{-1}(\cdot)'. \end{aligned} \tag{3.29}$$

### 3.2.3 Concentrating Out Σ

By using theorem 4.3 in Ribarits and Hanzon (2014a), let $(\beta,\overline{A_c},\overline{B_c})$ be given such that:

$$\frac{1}{T}\sum_{j=1}^{T}\Big(R_{0j} - S_{01}\beta(\beta' S_{11}\beta)^{-1}\beta R_{1j}\Big)(\cdot)' > 0. \tag{3.30}$$

Then the unique global minimizer solving $\min\limits_{\Sigma>0} L^{2c}(\beta,\overline{A_c},\overline{B_c},\Sigma)$ is given by:

$$\widehat{\Sigma} = S_{00} - S_{01}\beta\big(\beta' S_{11}\beta\big)^{-1}\beta S_{10}. \tag{3.31}$$

Inserting $\widehat{\Sigma}$ into equation (3.29) yields:

$$\begin{aligned} L^{3c}(\beta,\overline{A_c},\overline{B_c}) :&= L^{2c}(\beta,\overline{A_c},\overline{B_c},\widehat{\Sigma}) \\ &= \log\det\Big[S_{00} - S_{01}\beta(\beta' S_{11}\beta)^{-1}\beta' S_{10}\Big] + p. \end{aligned} \tag{3.32}$$

### 3.2.4 Concentrating Out $\beta_1$

First note that $\beta = [\beta_1 : \beta_2]$ and, therefore, we can write $L^{3c}$ as a function of $\beta_1$, $\beta_2$, $\overline{A_c}$ and $\overline{B_c}$. By following theorem 5 and lemma 3 in chapter 2 section 2.3.5,we get:

$$\widehat{\beta_1} = C\big[v_1,\ldots,v_{r-k}\big]\cdot\widetilde{T}, \quad \widetilde{T}\in \mathrm{Gl}(r-k). \tag{3.33}$$

The likelihood function $L^{4c}(\beta_2,\overline{A_c},\overline{B_c})$ is obtained as:

$$L^{4c}(\beta_2,\overline{A_c},\overline{B_c}) = L^{3c}(\widehat{\beta_1},\beta_2,\overline{A_c},\overline{B_c}) = \log\det S_{00} + \log\prod_{i=1}^{r-k}(1-\lambda_i) + p. \tag{3.34}$$

## 3.3 Parameterization of stable $[\overline{B_c} \mid \overline{A_c}]$ pairs

Next, we will optimize $[\overline{B_c} \mid \overline{A_c}]$. There is no analytic solution available. We will use a gradient search type algorithm for the optimization. $\overline{A_c}$ is a continuous-time asymptotically stable matrix, i.e $\sigma(\overline{A_c})$ is a subset of the open LHP. All matrices with this property can be written as $\Phi(\overline{A_\mu})$ where $\Phi$ represents the linear fractional transformation given by $\Phi(Z) = \dfrac{1+z}{1-z} = s$ and $\overline{A_\mu}$ is discrete-time asymptotic stable. We will use a particular parameterization approach to parameterize the pairs $[\overline{B_c} \mid \overline{A_\mu}]$. The parameterization is done in the same way as in section 2.4.1.5.

## 3.4 A Gradient Algorithm

Finally, we obtains the so-called concentrated likelihood $L^{4c} = L^{4c}(\overline{A_c}, \overline{B_c})$. Let $\overline{A_c} = \overline{A_c}(\theta)$ and $\overline{B_c} = \overline{B_c}(\theta)$ are parameterized by a vector $\theta \in \mathbb{R}^d$. Now, use the gradient

$$\frac{\partial L^{4c}(\overline{A_c}(\theta), \overline{B_c}(\theta))}{\partial(\theta)}.$$

If this exists, it is equal to

$$\frac{\partial L^{1c}\left(\widehat{\alpha}, \widehat{\beta}, \overline{A_c}(\theta), \overline{B_c}(\theta), \widehat{\Sigma}\right)}{\partial(\theta)},$$

see Theorem 9 in chapter 2, section 2.5. Here $L^{1c}$ is the concentrated likelihood obtained after optimizing with respect to $\left[\overline{C_c}, \alpha_2\right]$ only. A gradient algorithm can then be implemented to find a local optimum of the likelihood from a given starting point for $(\overline{A_c}, \overline{B_c})$ to find the most likely one relevant to the output time series. Also, it is to be noted that to optimize the chance that the global optimum of the likelihood is found, many starting points should be used, distributed rather evenly over the set of all possible input-normal pairs. The fact that all the (angular) parameters are taken from a finite interval makes this task easier.

Thus, this chapter formulated the state-space model in discrete time and continuous time where the latter refers to a continuous-time model that is observed at several points in a discrete time. The continuous-discrete formulation allows for irregularly timed observations. These appear to be relevant in high frequency financial markets. Also, this chapter highlights the benefits of applying the state-space model; as neither VAR nor VECM methodologies have been extended to continuous-time applications as far as the present author is aware. Chapter 4 will build on the findings of chapter 3. Simulations with the continuous-discrete time model have been carried out and and are treated in chapter 4. Also, it will present a case for simulation in continuous-time to draw a comparison between the state-space framework and the true model. The chapter will also present a case for discrete-time model to differentiate between the state-space model and the VAR model.

# Chapter 4

# Simulation Study

In this chapter, we will utilize generated data to address the issue of model selection. Moreover, a simulation study is conducted to compare the performance of the state-space Error Correction Model (SSECM) against the VAR Error Correction Model(VECM). It is important to recall that the VECM can be considered as a particular case within the class of vector auto-regressive (VAR) models. As a first step, the study discusses a discrete-time series in both the SSECM and the VECM case, and evaluates their performance based on several statistical benchmarks; including, information criteria, the goodness of fit, forecast accuracy (measured in terms of root mean square error), and Bayes factor. SSECM outperforms VECM in several cases that we consider. After that, a simulation study in continuous-discrete time is presented where we compare the estimated the cointegration state space against a true model, as neither VAR nor VECM frameworks have been extended to address the continuous-discrete time applications.

## 4.1 Generation of Cointegrated Models and Corresponding Time Series

The earlier chapters provided an overview of the VECM and SSECM, their underlying assumptions, and the maximum likelihood estimation of their parameters. Based on the discussion, this section focuses on testing the accuracy of the maximum likelihood estimates. This will be done by generating some simulation models and associated time series. These time series will then be used to test the model estimation procedures and to compare the outcomes of VECM and

SSECM. There are two cases which need to be considered regarding the dimensions $(n, p)$ of the parameter matrices, (1) $n \geq p$ and (2) $n < p$. Each of these cases has different implications for the dimension of the parameter space on which $L$ is defined, and in the second case, $\alpha_2$ and $\beta_2$ will appear (see Ribarits and Hanzon 2014a,b).

## Generation of the Angles $\theta_k$

The section 2.4 highlighted that the matrices $(\overline{A}, \overline{B})$ are parameterized by $np$ angles $\theta_{i,k}$, $i = 1, 2, \ldots, p$, and $k = 1, 2, \ldots, n$. To generate angles, the study will use the Monte Carlo (MC) method. The uniform distribution is used on $(-\pi/2, \pi/2)$ to generate each of the angles.

## Choice of $(\alpha, \beta)$

Now consider the cases (1) and (2). First consider case (1), here $\alpha$ and $\beta$ are randomly generated using a standard Gaussian distribution (in *Matlab* by using the built-in function *randn*). Finding $\alpha$ and $\beta$ with $\mathrm{rk}(\alpha\beta') < r$ has probability zero, so this is unlikely to occur, as $\alpha$ and $\beta$ are both tall or square and $n \times p$, $\mathrm{rk}(\alpha\beta') = r$ iff $\mathrm{rk}(\alpha) = r$ and $\mathrm{rk}(\beta) = r$. But as for a tall or square $n \times r$ matrix to be of rank less than $r$ all the $r \times r$ submatrices have to have determinant zero, this is a condition that has Lebesgue measure zero. The same holds for the condition that $\mathrm{rk}(\alpha) \leq r-1$ and/or $\mathrm{rk}(\beta) \leq r-1$). To be safe, check the value for $\mathrm{rk}(\alpha\beta') = r$ and if not eliminate this choice. In case(2), $\alpha_2$, $\beta_2$ appear and similarly here $\alpha_2$ is randomly generated using a standard Gaussian distribution and $\beta_2$ is generated similarly to $(\overline{A}, \overline{B})$, compare section 2.4.3 and the previous paragraph.

## The Choice of $\overline{C}$

First consider case(1), so $n \geq p$. Given that $\alpha$, $\beta$, $\overline{A}$, $\overline{B}$ have been selected, one has to choose $\overline{C}$ in such a way that the following equation holds see (2.15):

$$\overline{C}(I - \overline{A})^{-1}\overline{B} = -I_p - \alpha\beta'.$$

As $R$ is selected to be in $p$-upper Hessenberg form $\overline{B}$ will be of full rank $p$ and so will $(I - \overline{A})^{-1}\overline{B}$. Therefore, $(I - \overline{A})^{-1}\overline{B}$ will have at least $p$ independent rows. One can assume the last $p$ rows are linearly independent (if not: permute the rows, this corresponds to a state-space transformation). Partition $\overline{C}$ as $\overline{C} = [\overline{C}_1 : \overline{C}_2]$, where $\overline{C}_1$ is a matrix with dimensions equal to $(p \times (n-p))$, and $\overline{C}_2$ is a matrix with dimensions $(p \times p)$. Let $L := (I - \overline{A})^{-1}\overline{B}$ and partition $L$ as follows:

$$L = \begin{bmatrix} L_1 \\ \cdots \\ L_2 \end{bmatrix}, \tag{4.1}$$

where $L_2$ is a non-singular matrix with dimensions $(p \times p)$. One can then write :

$$[\overline{C}_1 \vdots \overline{C}_2] \begin{bmatrix} L_1 \\ \cdots \\ L_2 \end{bmatrix} = -I_p - \alpha\beta' \tag{4.2}$$

$$\Longleftrightarrow \overline{C}_1 L_1 + \overline{C}_2 L_2 = (-I_p - \alpha\beta') \tag{4.3}$$

$$\Longleftrightarrow \overline{C}_2 L_2 = -\overline{C}_1 L_1 - I_p - \alpha\beta'. \tag{4.4}$$

As $L_2$ is non-singular, one finds:

$$\overline{C}_2 = (-\overline{C}_1 L_1 - I_p - \alpha\beta')(L_2)^{-1}. \tag{4.5}$$

So, one can select any $\overline{C}_1$; the study uses the standard Gaussian distribution to generate the elements of $\overline{C}_1$. In the end, the matrix $\overline{C}$ is given a value that is in accordance with the partitioned matrix $[\overline{C}_1 \vdots \overline{C}_2]$.

Hence, the choice of $(\alpha, \beta, \overline{A}, \overline{B}, \overline{C})$ and the corresponding triple $A = \overline{A} + \overline{BC}$, $B = \overline{B}$, $C = -\overline{C}$ should meet the four assumptions that form the basis of the SSECM as described in section 2.1.

In case(2), given that $\alpha_1$, $\alpha_2$, $\beta_1$, $\beta_2$, $\overline{A}$, $\overline{B}$ have been selected, one has to choose $\overline{C}$ in such a way that the following equation holds:

$$[\overline{C} \vdots \alpha_2] \begin{bmatrix} (I - \overline{A})^{-1}\overline{B} \\ \beta_2' \end{bmatrix} = -I_p - \alpha\beta'. \tag{4.6}$$

However, we follow a similar process as was done case(1) to find matrix $\overline{C}$, by let

$$L = \begin{bmatrix} (I - \overline{A})^{-1}\overline{B} \\ \beta_2' \end{bmatrix},$$

and partition $L$ similar to equation (4.1). Then, we will use equation (4.5) to find matrix $\overline{C_2}$, and by selecting any $\overline{C_1}$ as was done in case(1). Finally, the matrix $\overline{C}$ is determined as the partitioned matrix $[\overline{C}_1 \vdots \overline{C}_2]$.

### Generation of Time Series

To generate the time series, the study utilizes an initial state $\tilde{x}_0$ and equires the innovations sequence $\{\epsilon_t\}_{t=0}^{T}$. The study will draw the elements of innovations from a Gaussian distribution with zero mean and co-variance matrix $\Sigma > 0$ and here, choose $\Sigma = I$. The $\tilde{x}_0$ are generated by a standard Gaussian. The next step is to compute the time series $\{y_t\}_{t=0}^{T}$. Here, the study uses the model (13) and also obtain $\Delta y_t \in \mathbb{R}^{p \times r}$, $y_{t-1} \in \mathbb{R}^{p \times r}$ and $\tilde{x}_t \in \mathbb{R}^{p \times r}$ to be stored in $Z_{0t}$, $Z_{1t}$ and $Z_{2t}$ respectively for each $t \in \{0, 1, 2, \ldots, T\}$.

## 4.2 Model Selection

### 4.2.1 VECM Model

Model selection in the VAR setting was done as follows: The lag order $k$ has been determined using the Akaike information criterion, as is known from Johansen's theory performing the trace test. Having determined a lag estimate $k_{AIC}$, the classical Johansen trace test has been used to determine the number $r = p - c$ of cointegrating relations. That can be done by simulation of the $c \times c$ random matrix see (Johansen 1988):

$$\left(\int_0^1 \mathbf{W} d\mathbf{W}'\right)' \left(\int_0^1 \mathbf{W}\mathbf{W}' ds\right)^{-1} \left(\int_0^1 \mathbf{W} d\mathbf{W}'\right). \tag{4.7}$$

Here $W(t)$ is a $c$-dimensional standard Brownian motion. Simulation was performed replacing the Brownian motion by a Gaussian random walk with 400 steps and involved 6000 replications. For further details of how the simulation is done see the section below.

#### 4.2.1.1 Asymptotic Distribution of the Cointegration Rank Test Statistics (Trace Test)

To find asymptotic distribution of the cointegration rank test statistics (Trace Test), the study assumes a $p$-dimensional time series. It is interested in testing the null hypothesis $H_0$ for cointegrating rank

$$H_0 : r = r_0 \quad \text{against} \quad H_1 : r_0 < r \leq p.$$

The likelihood ratio statistic $\lambda_{LR}(r_0, p) = \mathrm{tr}(\mathcal{D})$ is the *Johansen trace statistic* (Note: here $\lambda$ allows us to define an anonymous function inside another function. It helps us estimate the cointegrating vectors and restrictions can be tested using standard inference ($\lambda$). Asymptotic theory shows that estimated rank is consistent in the sense that the probability that the estimated rank is not equal to the true rank equals the size of tests, whereas the probability that the estimated rank is too small vanishes). The test statistic has the asymptotic distribution equal to that of trace of random matrix $\mathcal{D}$, given by

$$\mathcal{D} = \left(\int_0^1 \mathbf{W} d\mathbf{W}'\right)' \left(\int_0^1 \mathbf{W}\mathbf{W}' ds\right)^{-1} \left(\int_0^1 \mathbf{W} d\mathbf{W}'\right),$$

where $\mathbf{W} := \mathbf{W}_{p-r_0}(s)$ stands for a $(p-r_0)$-dimensional standard Wiener process.

The above statistic ($\mathcal{D}$) can be computed by estimating the distribution of the statistics based on Monte Carlo sampling. We want to compute (aspects of) the distribution of this random variable. Namely, one can generate samples $\lambda_j$, $j = 1, \ldots, M$. After samples, construct an empirical distribution function; and consequently, compute the tables of probabilities based on a histogram. The procedure for generating a single sample of $\lambda_j$ is described below. This is based essentially on generating a single approximate samples of the matrix $\mathcal{D}$, say $\mathcal{D}^j$, that itself requires the generation of a single sample path of a Wiener process $\mathbf{W}_{p-r_0}(s)$ in the interval $[0, 1]$.

More precisely, the latter is achieved after approximating $[0, 1]$ using a partition with a time step $h$ that results in $N =: 1/h$ intervals, so $h =: 1/N$ and we take $N \in \mathbb{N}$. Next, the study generated increments

$$d\mathbf{W}(s_i) \approx \Delta\mathbf{W}(s_i) = \sqrt{h}\mathbf{Z}^i,$$

where $\mathbf{Z}^i \sim \mathcal{N}(0, \mathbf{I}_{K-r_0})$ are uncorrelated standard Gaussian variables, for $i = 1, \ldots, N$. The sample path $\mathbf{W}$ is then computed as

$$\mathbf{W}(s_i) := \sum_{r=1}^{i} \Delta\mathbf{W}(s_r).$$

The integrals, involved in $\mathcal{D}$ are then approximated as

$$\int_0^1 \mathbf{W}d\mathbf{W}' \approx \sum_{i=1}^{N} \mathbf{W}(s_i)\Delta\mathbf{W}(s_i)',$$

and

$$\int_0^1 \mathbf{W}\mathbf{W}'ds \approx \sum_{i=1}^{N} \mathbf{W}(s_i)\mathbf{W}(s_i)'h.$$

Repeating this procedure, generate samples

$$\mathcal{D}^j = \left[\sum_{i=1}^{N} \mathbf{W}(s_i)\Delta\mathbf{W}(s_i)'\right]' \left[\sum_{i=1}^{N} \mathbf{W}(s_i)\mathbf{W}(s_i)'h\right]^{-1} \left[\sum_{i=1}^{N} \mathbf{W}(s_i)\Delta\mathbf{W}(s_i)'\right],$$

$j = 1, \ldots, M$ and compute the trace $\lambda^j = \mathrm{tr}\left\{\mathcal{D}^j\right\}$ from which the histogram follows below. Our table presents 1000 observations for the deviations from Johansen (1988) at different confidence intervals as shown below:

Table 4.1: Quantiles of the Asymptotic Distribution of the Cointegration Rank Test Statistic

| Dim\ % | 2.5% | 5% | 10% | 50% | 90% | 95% | 97.5% |
|---|---|---|---|---|---|---|---|
| **1** | 0.0015 | 0.0056 | 0.0231 | 0.5936 | 2.9689 | 4.0984 | 5.2706 |
| **2** | 1.6165 | 1.9955 | 2.5470 | 5.4757 | 10.4569 | 12.3079 | 14.0016 |
| **3** | 7.1194 | 8.0190 | 9.1614 | 14.4037 | 21.7588 | 24.2586 | 26.5106 |
| **4** | 16.6143 | 18.0606 | 19.8591 | 27.3848 | 37.0136 | 40.1696 | 43.0435 |
| **5** | 30.1801 | 32.1422 | 34.5877 | 44.3538 | 56.1665 | 59.9041 | 63.2475 |

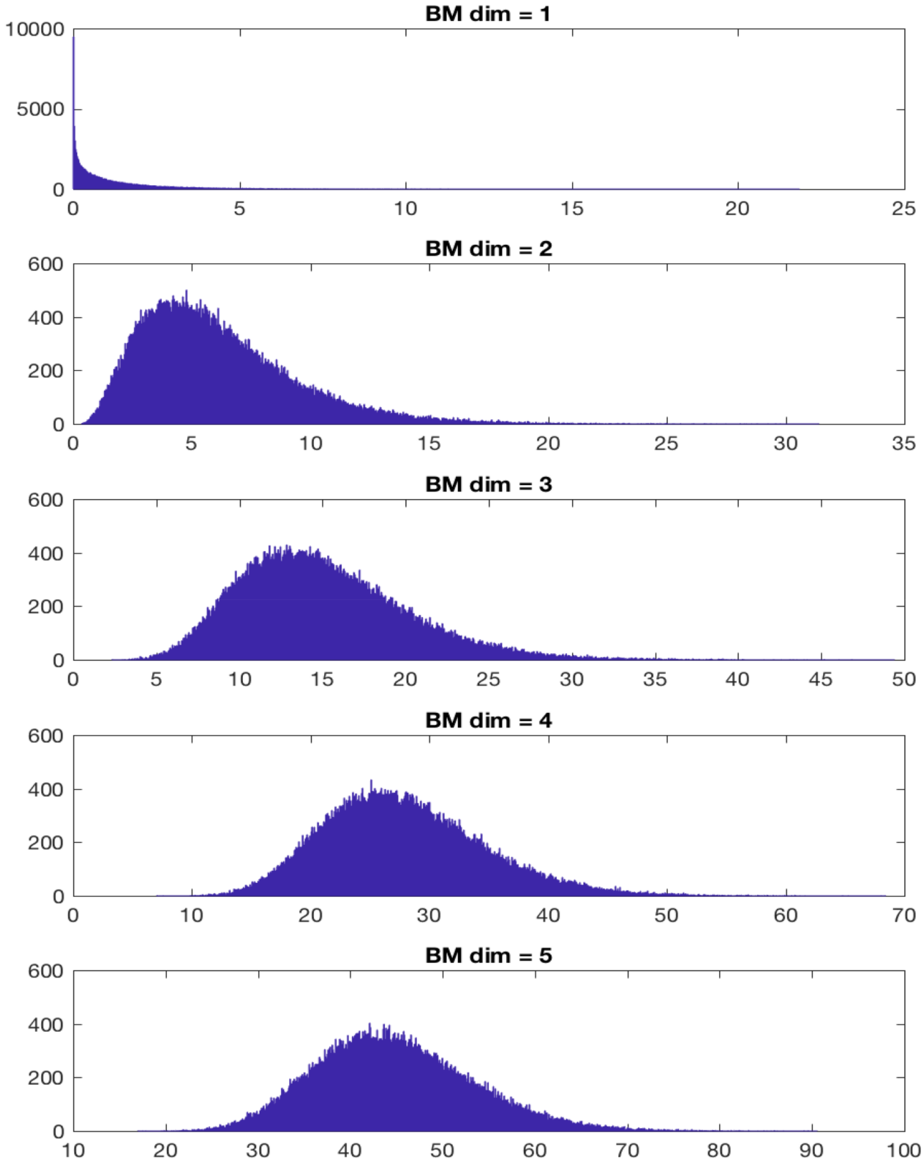



Figure 4.1: Histogram of the Asymptotic Distribution of the Cointegration Rank Test Statistic (Trace Test)

### 4.2.2 State-Space Models

There are several ways of determining $n$. One such method is the likelihood ration test that presented in section 1.7.2 to find the trace test of state-space model. Then, study used the equation (4.7) to find the critical value and used it to compare it with trace test. The study rejects the test if the the trace result above of the critical value, since $r$ can be determined by using $c = p - r$. The study uses equation (4.7) because the state-space model is $VAR(\infty)$ such that

$$y_t = \sum_{1}^{\infty} H_K y_{t-K},$$

and the impulse response coefficients $H_K$ converge to zero exponentially (as $\bar{A}$ is asymptotic stable). Therefore, one can approximately neglect the tail of the summation. Hence

$$y_t \approx \sum_{1}^{K} H_K y_{t-K},$$

for some sufficiently large $K$ and for $t - t_0$ sufficiently large, where $t_0$ is the time at which the state equation starts, so the time for which $x_{t_0}$ is the initial state. This last model is a VAR model to which to Johansen test can be applied. Therefore we claim the Johansen test can also be applied in our case by good approximation. An alternative way of determining $n$ is by using it as a sample in calculation runs as shown below: The study uses the VECM software to determine a cointegrated VAR model. This gives an estimated of $r = p - c$, using the classical trace test by Johansen (1988). The resulting VEC model is readily obtained as well, and one can then transform it back to a conventional VAR, which in turn can be easily written in state-space form $A_{joh}$, $B_{joh}$, $C_{joh}$. It is important to consider that the state-space representation of a VAR is widely known as the *companion form of the VAR model*. Matrix $A_{joh}$ has 1 as $c$-fold eigenvalue. By applying a suitable state transformation of the form:

$$\left(\tilde{A}_{joh},\ \tilde{B}_{joh},\ \tilde{C}_{joh}\right) = \left(TA_{joh}T^{-1},\ TB_{joh},\ C_{joh}T^{-1}\right),$$

it is possible to block- diagonalize $A_{joh}$ such that $\tilde{A}_{joh} = \mathrm{diag}(I_c,\ \tilde{A}^{st}_{joh})$ where $|\ \lambda_{\max}(\tilde{A}^{st}_{joh})\ | < 1$. To determine $n$, it is possible to follow the *balance and truncate* approach, which should be applied to the stable subsystem $(\tilde{A}^{st}_{joh},\ \tilde{B}^{st}_{joh}, \tilde{C}^{st}_{joh})$ where $B_{joh}$ and $C_{joh}$ are partitioned is accordance with partitioning of $\tilde{A}_{joh}$, such that $B_{joh} = (*\tilde{B}^{st}_{joh})$, $C_{joh} = (*, \tilde{C}^{st}_{joh})$. Again, an information-type criterion would serve to consider the number of singular values that differ significantly from zero as the estimated order of the stable subsystem. By adding $c$, one can obtain the estimated system order $n$.

## 4.3 Discrete-time Simulation Study (case 1: $p \leq n$)

In this section, the study will compare the classical VECM approach with the SSECM approach in case 1 when $p \leq n$.

### 4.3.1 Model Estimation

In this section, the study will provide simulation results and model selection result. It considers four state-space systems $(A, B, C)$, with $p = 3$ and with state dimension $n = 4$. These were generated with the techniques described in section 4.2.2. Note that, $c \leq p$ due to the minimality of $(A, B, C)$. Here the four systems have $c = 2$ and the eigenvalues of the $A$ matrix in a minimal state-space representation ($\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$) are as given in Table 4.2.

Table 4.2: The Eigenvalues of the $A$-matrix in a Minimal State-Space Representation of the Data Generating Models.

| Model | $T$ | $c$ | Sigma(A) |
|---|---|---|---|
| 1 | 100 | 2 | $(0.04, 0.86, 1, 1)$ |
| 2 | 500 | 2 | $(-0.84, 0.42, 1, 1)$ |
| 3 | 1000 | 2 | $(-0.63, 0.59, 1, 1)$ |
| 4 | 2000 | 2 | $(0.68, 0.76, 1, 1)$ |

Simulation data comprising $T = 100$, $T = 500$, $T = 1000$ and $T = 2000$ output observations $(y_1; \ldots; y_T)$ have been generated. For each of the four cases, the study will find a good estimate of the data generating process (DGP). First, it will have to select the discrete parameters $K$, $n$ , $r$ as discussed in section above. A VECM has been estimated for these data sets. As seen in Table 4.3 below, the Akaike information criteria suggests different lag orders depending on the size of $T$. Namely, the appropriate lag length is 3 for 100 observations, 4 for 500 observations, 5 for 1000 observations, and 6 for 2000 observations.

Table 4.3: Lag Order Selection Results

| Lag order | $T = 100$ | $T = 500$ | $T = 1000$ | $T = 2000$ |
|---|---|---|---|---|
| | Model AIC | Model AIC | Model AIC | Model AIC |
| 1 | 1166.42 | 5829.63 | 11175.62 | 23837.17 |
| 2 | 904.23 | 4550.69 | 9223.90 | 18514.42 |
| 3 | 896.81 | 4502.07 | 8972.52 | 17524.29 |
| 4 | 907.25 | 4482.94 | 8850.93 | 17284.29 |
| 5 | 915.31 | 4505.12 | 8806.46 | 17091.35 |
| 6 | 934.99 | 4531.76 | 8806.47 | 17061.69 |
| 7 | 964.98 | 4564.11 | 8829.23 | 17090.44 |
| 8 | 989.19 | 4586.89 | 8845.92 | 17141.12 |
| 9 | 1012.02 | 4621.58 | 8878.39 | 17196.86 |
| 10 | 1036.86 | 4656.20 | 8908.21 | 17237.04 |
| 11 | 1056.44 | 4692.27 | 8945.44 | 17291.59 |
| 12 | 1080.63 | 4732.20 | 8973.71 | 17343.27 |

An SSECM model was also estimated for the same data sets. Here it was assumed that the order is 4(= the correct order) for all data sets. However, the study will now present the results of estimating the state-space model with order $n = 3$, using the data generated by the fourth model, and with $T = 2000$ observations. To compare the outcomes with those that can be obtained using the true model, the data is divided into two data sets each with the number of observations $T = 1000$. The study estimates the state-space model once with order $n = 4$, and once more with order $n = 3$ using the first set. Table 4.4 shows that the sample co-variance matrix of the $\hat{\epsilon}_t$ $(t = 1, \ldots, T)$ that was found with order $n = 4$ are close to Identity compare to the sample co-variance matrix of the $\hat{\epsilon}_t$ with order $n = 3$.

Table 4.4: Sample Co-variance Matrices of $\hat{\epsilon}_t$ from State-Space Data Generating Processes with Order $n = 4$ and Order $n = 3$

| $n = 4$ | | | $n = 3$ | | |
|---|---|---|---|---|---|
| 0.9995 | 0.0007 | 0.0002 | 0.9427 | 0.0407 | 0.0318 |
| 0.0001 | 0.9934 | 0.0004 | 0.0023 | 0.9591 | 0.0003 |
| 0.0004 | 0.0003 | 0.9987 | 0.0067 | 0.0019 | 0.9101 |

The Trace Cointegration Test was used to evaluate the number of cointegrating relations $(r)$ between $\{y_t\}_t$. Table 4.5 shows the results of applying the Cointegration Test to the generated data. The results show that the null hypothesis of no cointegration relationships ($r = 0$) can be rejected at a significance level of 95%, giving strong evidence for the presence of cointegration. The null hypothesis of at most 1 cointegrating relation (rank $r \leq 1$ against the alternative $r > 1$) can not be rejected at a significance level of 95%.

Table 4.5: Results of the VECM Trace Cointegration Test and State-Space Trace Cointegration Test

| Rank (number of r) ($T = 100$) | $r = 0$ | $r = 1$ | $r = 2$ |
|---|---|---|---|
| VECM Trace Statistic | 45.1627 | 6.9073 | 1.5334 |
| State Space Trace Statistic | 47.3746 | 6.8836 | 1.6548 |
| Critical Value(own code) | 24.2586 | 12.3079 | 4.0984 |
| $p$-value | 0.0010 | 0.5867 | 0.3175 |
| Rank (number of r) ($T = 500$) | $r = 0$ | $r = 1$ | $r = 2$ |
| VECM Trace Statistic | 188.5833 | 5.1530 | 0.3686 |
| State Space Trace Statistic | 99.6435 | 6.9356 | 0.76489 |
| Critical Value(own code) | 24.2586 | 12.3079 | 4.0984 |
| $p$-value | 0.0010 | 0.5914 | 0.2082 |
| Rank (number of r) ($T = 1000$) | $r = 0$ | $r = 1$ | $r = 2$ |
| VECM Trace Statistic | 260.0482 | 3.6864 | 0.1984 |
| State Space Trace Statistic | 199.847 | 4.97547 | 0.2574 |
| Critical Value(own code) | 24.2586 | 12.3079 | 4.0984 |
| $p$-value | 0.0010 | 0.6626 | 0.3653 |
| Rank (number of r) ($T = 2000$) | $r = 0$ | $r = 1$ | $r = 2$ |
| VECM Trace Statistic | 106.1460 | 11.3374 | 0.0018 |
| State Space Trace Statistic | 97.6578 | 9.7469 | 0.0035 |
| Critical Value(own code) | 24.2586 | 12.3079 | 4.0984 |
| $p$-value | 0.0010 | 0.5727 | 0.3732 |

This thesis used a parametric model to study cointegration.However, a non-parametric model can also be used to study cointegration. For instance, Bierens (1997) proposed consistent co-integration tests using a non-parametric approach. His approach is similar to Johansen's method, who estimated the co-integrating vectors and derived them using maximum likelihood estimation from the eigenvectors. Bierens (1997)' study was conducted analogously to Johansen's tests, including parametric restrictions on the co-integrating vectors. Likewise, a non-parametric model free approach to cointegration can be found in Poskitt (2012). Unlike classical tests, non-parametric tests make only few assumptions about the data and are appropriate when the data distribution is non-normal. Therefore, to use a non-parametric approach, one needs to evaluate future literature and investigate a non-parametric approach to estimating the co-integrating rank (to adapting to our model). In present study, we investigated cointegration using State Space model and a normally distributed data. We expected our method to work well with a non-parametric approach as well as non-normal distributed data; hence, we aim to investigate it in our future work related to cointegration.

Next, the study assesses the MLE of the remaining (continuous) parameters. This will be treated below.

### 4.3.2 Comparison Between the Outcomes of VECM, SSECM and the True Model

In this section, state-space process will be utilized to set up 4 different model using different $\theta$. The objective of the study is to compare the true model and the results of MLE parameters using state-space model. The study will also compare the cointegration vector generated using state-space with the one generated by using 4 different models. We want to study deviation between the true $\theta$ parameter with different models to establish the advantage of using state-space model and compare it with the outcome of VECM. The Table 4.6 presents our findings for the $\theta$ for all 4 models that we generated.

Table 4.6: True Theta from the State-Space Data Generating Process in Different Models (using the different $\theta$).

| | | | | |
|---|---|---|---|---|
| Model 1 | −0.27261 | −0.91946 | −0.30682 | −0.63637 |
| | 0.63547 | 0.13784 | 0.19223 | 0.29043 |
| | −0.67585 | −0.63024 | −0.44679 | 0.35666 |
| Model 2 | 0.67615 | 0.53792 | −0.22352 | −0.64732 |
| | −0.19115 | −0.88425 | −0.67453 | −0.38491 |
| | −0.69002 | 0.82776 | −0.38541 | −0.02017 |
| Model 3 | 0.17817 | 0.1566 | 0.40346 | −0.88662 |
| | 0.2011 | 0.61957 | 0.47747 | 0.15329 |
| | −0.88202 | −0.30162 | −0.47865 | 0.82501 |
| Model 4 | 0.73248 | −0.40799 | 0.53523 | −0.6209 |
| | 0.43598 | −0.78821 | 0.35164 | 0.91431 |
| | −0.47761 | 0.66522 | 0.86352 | 0.06545 |

Further, in Table 4.7 the study compares the cointegrating vector in each of the four generated time series, as estimated by VECM and SSECM respectively, with their true value. The results were relatively close.

Table 4.7: Cointegration Vector

| **T = 100** (Model 1) | **True model** | **VECM ($k = 2$)** | **SSECM ($n = 4$)** |
|---|---|---|---|
| **Cointegration Vector** | 2.49<br>−2.03<br>−0.82 | 2.44<br>−2.01<br>−0.85 | 2.51<br>−2.03<br>−0.80 |
| **T = 500** (Model 2) | **True model** | **VECM ($k = 3$)** | **SSECM ($n = 4$)** |
| **Cointegration Vector** | 0.12<br>2.66<br>−0.43 | 0.14<br>2.45<br>−0.40 | 0.17<br>2.64<br>−0.44 |
| **T = 1000** (Model 3) | **True model** | **VECM ($k = 4$)** | **SSECM ($n = 4$)** |
| **Cointegration Vector** | 0.50<br>−2.01<br>2.03 | 0.54<br>−2.05<br>2.02 | 0.51<br>−2.09<br>2.03 |
| **T = 2000** (Model 4) | **True mode** | **VECM ($k = 5$)** | **SSECM ($n = 4$)** |
| **Cointegration Vector** | −4.79<br>3.84<br>2.27 | −4.83<br>3.79<br>2.38 | −4.74<br>3.88<br>2.25 |

Table 4.8 shows the comparison between the estimated VECM and SSECM using four different models or observations. The study found out three characteristics that hold across different sample sizes. First, the log likelihood estimates are similar between both model classes. Second, the number of parameters grows fast in the VECM but stays the same in the SSECM. As a general comment, it is well-known that models under the VAR class are prone to become over-parameterized. Third, the BIC information criterion is lower for the SSECM, which favors the SSECM over the VECM in terms of model selection. Note that, the study used the same true parameters as in Table 4.6. The study chose 1000 starting points to search for state space MLE. The best results (our optimal likelihood values) are presented in Table 4.8.

Table 4.8: Comparison of Johansen's VAR Error Correction Model and Estimated SSECM for $T = 100$, $T = 500$, $T = 1000$, and $T = 2000$.

| $T = 100$ (Model 1) | **VECM ($k = 2$)** | **SSECM ($n = 4$)** |
|---|---|---|
| **Cointegration rank** | 1 | 1 |
| **Effective sample size** | 97 | 99 |
| **Total Parameters** | 29 | 26 |
| **BIC** | 944.72 | 927.74 |
| **Log likelihood** | 390.10 | 389.85 |
| $T = 500$ (Model 2) | **VECM ($k = 3$)** | **SSECM ($n = 4$)** |
| **Cointegration rank** | 1 | 1 |
| **Effective sample size** | 496 | 499 |
| **Total Parameters** | 38 | 26 |
| **BIC** | 4542.25 | 4453.35 |
| **Log likelihood** | 2132.33 | 2131.63 |
| $T = 1000$ (Model 3) | **VECM ($k = 4$)** | **SSECM ($n = 4$)** |
| **Cointegration rank** | 1 | 1 |
| **Effective sample size** | 995 | 999 |
| **Total Parameters** | 47 | 26 |
| **BIC** | 8875.59 | 8664.33 |
| **Log likelihood** | 4249.76 | 4228.09 |
| $T = 2000$ (Model 4) | **VECM ($k = 5$)** | **SSECM ($n = 4$)** |
| **Cointegration rank** | 1 | 1 |
| **Effective sample size** | 1994 | 1999 |
| **Total Parameters** | 56 | 26 |
| **BIC** | 17149.75 | 17023.13 |
| **Log likelihood** | 8331.37 | 8327.83 |

Next, we will be using the same $\theta$ to distinguish the effects of model parameters based on sample size (including 100, 750, and 2000 observations) in each model. For consistency and comparison, we have also performed a Monte Carlo replication to ascertain the sampling properties of the suggested procedures and examined the accuracy of our estimation. The exact process is repeated 10000 times for generating different $\epsilon_t$ series data of 100, 750, and 2000 observations in each model. The goal of repeating the process was to develop the best estimate for MLE by applying SSECM to time series. We estimated the average value and compared the result with the actual $\theta$ value at a significance level of 95% and the standard deviation as presented in Table 4.10. When we compare the parameter for true $\theta$ to model parameters, there is little difference as the number of observations increase. However, $T = 2000$ observations yielded a better result. As we can see, the difference between the $\theta$ used for the true model and the $\theta$ of the MLE for state-space is slightly small for $T = 100$, some of the estimates are close to the true $\theta$ but not for all. For example, the first 3 models, not all of the estimates are significantly close to the true $\theta$. However, when we increase $T = 2000$, then we see that all of the estimate are significantly close to the true $\theta$, indicating a better fit model.

Next, we have Table 4.9, where the MLE parameter estimates show an improved value when compared to actual $\theta$ parameters as the $T$ increases. The results show that our method is reliable and consistent, as shown in Table 4.10 for Model 4 or $T = 2000$. The standard deviation values are minimal, which indicates that there is less variance or statistical difference and the values are same, more or less at different times. Also, Table 4.11 shows the results of comparing the co-integrating vector; the result shows better value of co-integrating vector when compared with the actual model as $T$ increases as well as the standard deviation at a significance level of 95%.

Table 4.9: Monte Carlo study: (a) The average value of MLE theta in different epsilon time series

| | True Theta from the State-Space Data Generating Process | | | | The average value of MLE theta in different epsilon time series | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # Observation | | | | | $T = 100$ | | | | $T = 750$ | | | | $T = 2000$ | | | |
| Model 1 | -0.27261 | -0.91946 | -0.30682 | -0.63637 | -0.2729 | -0.90577 | -0.55418 | -0.63177 | -0.27201 | -0.91921 | -0.30641 | -0.6364 | -0.27265 | -0.91950 | -0.30677 | -0.63632 |
| | 0.63547 | 0.13784 | 0.19223 | 0.29043 | 0.63686 | 0.12472 | 0.19041 | 0.28629 | 0.63537 | 0.13792 | 0.17211 | 0.29055 | 0.63551 | 0.13776 | 0.19241 | 0.29041 |
| | -0.67585 | -0.63024 | -0.44679 | 0.35666 | -0.67811 | -0.65876 | 0.74522 | 0.35999 | -0.67593 | -0.63757 | -0.41178 | 0.35621 | -0.67589 | -0.63021 | 0.44655 | 0.35697 |
| Model 2 | 0.67615 | 0.53792 | -0.22352 | -0.64732 | 0.67447 | 0.53756 | -0.21316 | -0.64655 | 0.67610 | 0.53784 | -0.22337 | -0.64712 | 0.676112 | 0.53790 | -0.22355 | -0.64736 |
| | -0.19115 | -0.88425 | -0.67453 | -0.38491 | -0.18683 | -0.88439 | -0.68239 | -0.38503 | -0.19270 | -0.88458 | -0.67428 | -0.38477 | -0.19112 | -0.88423 | -0.67456 | -0.38493 |
| | -0.69002 | 0.82776 | -0.38541 | -0.02017 | -0.68596 | 0.82781 | -0.37584 | -0.020162 | -0.69014 | 0.82771 | -0.38544 | -0.02035 | -0.69000 | 0.82774 | -0.38545 | -0.02016 |
| Model 3 | 0.17817 | 0.1566 | 0.40346 | -0.88662 | 0.17897 | 0.1576 | 0.40376 | -0.87492 | 0.17821 | 0.1563 | 0.40349 | -0.88667 | 0.17819 | 0.1566 | 0.40346 | -0.88661 |
| | 0.2011 | 0.61957 | 0.47747 | 0.15329 | 0.2018 | 0.61665 | 0.47353 | 0.15370 | 0.2009 | 0.61951 | 0.47750 | 0.15333 | 0.2010 | 0.61957 | 0.47745 | 0.15330 |
| | -0.88202 | -0.30162 | -0.47865 | 0.82501 | -0.8755 | -0.30439 | -0.47891 | 0.82511 | -0.88202 | -0.30162 | -0.47864 | 0.82500 | -0.88202 | -0.30163 | -0.47867 | 0.82503 |
| Model 4 | 0.73248 | -0.40799 | 0.53523 | -0.6209 | 0.73248 | -0.40725 | 0.53498 | -0.6207 | 0.73242 | -0.407103 | 0.53518 | -0.6212 | 0.73248 | -0.40797 | 0.53520 | -0.6209 |
| | 0.43598 | -0.78821 | 0.35164 | 0.91431 | 0.43353 | -0.78796 | 0.35132 | 0.91422 | 0.43591 | -0.78816 | 0.35169 | 0.91429 | 0.435100 | -0.78825 | 0.35164 | 0.91435 |
| | -0.47761 | 0.66522 | 0.86352 | 0.06545 | -0.47632 | 0.66501 | 0.86378 | 0.06549 | -0.47756 | 0.66519 | 0.86348 | 0.06543 | -0.47763 | 0.66528 | 0.86356 | 0.06549 |

Table 4.10: Monte Carlo study: (b) Standard deviation results

| # Observation | $T = 100$ | | | | $T = 750$ | | | | $T = 2000$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model 1 | 0.00099 | 0.00075 | 0.00085 | 0.00094 | 0.00073 | 0.00038 | 0.00065 | 0.00044 | 0.00032 | 0.00016 | 0.00047 | 0.00015 |
| | 0.00116 | 0.00084 | 0.00044 | 0.00094 | 0.00105 | 0.00042 | 0.00037 | 0.00080 | 0.00091 | 0.00038 | 0.00011 | 0.00072 |
| | 0.00081 | 0.00095 | 0.00088 | 0.00092 | 0.00064 | 0.00071 | 0.00086 | 0.00070 | 0.00040 | 0.00062 | 0.00073 | 0.00067 |
| Model 2 | 0.00063 | 0.00060 | 0.00131 | 0.00076 | 0.00038 | 0.00049 | 0.00094 | 0.00047 | 0.00019 | 0.00021 | 0.00087 | 0.00038 |
| | 0.00034 | 0.00059 | 0.00099 | 0.00118 | 0.00019 | 0.00042 | 0.00087 | 0.00106 | 0.00001 | 0.00026 | 0.00064 | 0.00093 |
| | 0.00052 | 0.00037 | 0.00111 | 0.00049 | 0.00030 | 0.00014 | 0.00091 | 0.00028 | 0.00019 | 0.00003 | 0.00072 | 0.00011 |
| Model 3 | 0.00093 | 0.000109 | 0.00055 | 0.00037 | 0.00073 | 0.00091 | 0.00048 | 0.00021 | 0.00056 | 0.00077 | 0.00023 | 0.00007 |
| | 0.00082 | 0.00034 | 0.00056 | 0.00093 | 0.00067 | 0.00017 | 0.00040 | 0.00073 | 0.00042 | 0.00009 | 0.00028 | 0.00061 |
| | 0.00042 | 0.00085 | 0.00069 | 0.00122 | 0.00022 | 0.00069 | 0.00048 | 0.00096 | 0.00017 | 0.00051 | 0.00027 | 0.00081 |
| Model 4 | 0.00036 | 0.00048 | 0.00061 | 0.00090 | 0.00016 | 0.00020 | 0.00051 | 0.00077 | 0.00009 | 0.00012 | 0.00041 | 0.00062 |
| | 0.00063 | 0.00058 | 0.00074 | 0.00068 | 0.00050 | 0.00032 | 0.00052 | 0.00044 | 0.00037 | 0.00013 | 0.00021 | 0.00016 |
| | 0.00047 | 0.00089 | 0.00033 | 0.00027 | 0.00033 | 0.00062 | 0.00019 | 0.00015 | 0.00011 | 0.00039 | 0.00001 | 0.00003 |

Table 4.11: Monte Carlo study: (c) Cointegrating vector with their true model

| | True | $T = 100$ | | $T = 750$ | | $T = 2000$ | |
|---|---|---|---|---|---|---|---|
| | mode | SSECM(MC) | SD | SSECM(MC) | SD | SSECM(MC) | SD |
| Model 1 | 2.49 | 2.58 | 0.00012 | 2.54 | 0.00009 | 2.5 | 0.00002 |
| | −2.03 | −2.15 | 0.00017 | −2.07 | 0.00007 | −2.01 | 0.00005 |
| | −0.82 | −0.86 | 0.00009 | −0.85 | 0.00004 | −0.8 | 0.00001 |
| Model 2 | 0.12 | 0.22 | 0.00013 | 0.19 | 0.00008 | 0.14 | 0.00004 |
| | 2.66 | 2.62 | 0.00006 | 2.71 | 0.00009 | 2.65 | 0.00003 |
| | −0.43 | −0.44 | 0.00007 | −0.46 | 0.00017 | −0.42 | 0.00001 |
| Model 3 | 0.5 | 0.61 | 0.00016 | 0.54 | 0.00012 | 0.51 | 0.00008 |
| | −2.01 | −2.07 | 0.00013 | −2.04 | 0.00004 | −2.02 | 0.00003 |
| | 2.03 | 2.06 | 0.00007 | 2.01 | 0.00003 | 2.05 | 0.00001 |
| Model 4 | −4.79 | −4.83 | 0.00011 | −4.75 | 0.00008 | −4.8 | 0.00009 |
| | 3.84 | 3.79 | 0.00015 | 3.81 | 0.00012 | 3.83 | 0.00008 |
| | 2.27 | 2.31 | 0.00009 | 2.26 | 0.00007 | 2.28 | 0.00005 |

For further validation our model, we are comparing the VECM and SSECM by using two methods as shown in Table 4.12. The two methods are a) MLE estimation, and b) least-squares method. We have taken Model 4 in several time series (100, 750 and 200). In Model 4, the theta estimates were far better when compared with other 3 models because of less deviation between estimated and true $\theta$; hence, a higher probability of providing a better $\theta$ for the model. The table below shows that the estimates are similar for a normally distributed data set. The result indicates that VECM estimates for both MLE and Least Squares Method (LSM) provide similar forecasts to the SSECM model as there are more variables in VECM. Next, in Table 4.13, we select Model 4 as well for the reason stated above to compare the MLE and LSM estimates and actual values for the parameter vector $\theta$. The findings suggest that the MLE parameters are closer to the actual theta parameters compared to LSM but not that different. The two methods are only likely to show significantly different statistical characteristics for smaller sample sizes. However, as the sample size increases to $T = 2000$, MLE parameters are closer to the actual $\theta$ parameters as well as in LSM. The deviance becomes less and the parameter becomes closer to the true $\theta$ with an increase in the sample size. In conclusion, the $\theta$ from the true model is closer to the $\theta$ estimates of MLE and LSM models as the time increases indicating that for smaller sizes the estimates or parameters for LSM and MLE are asymptotically equivalent.

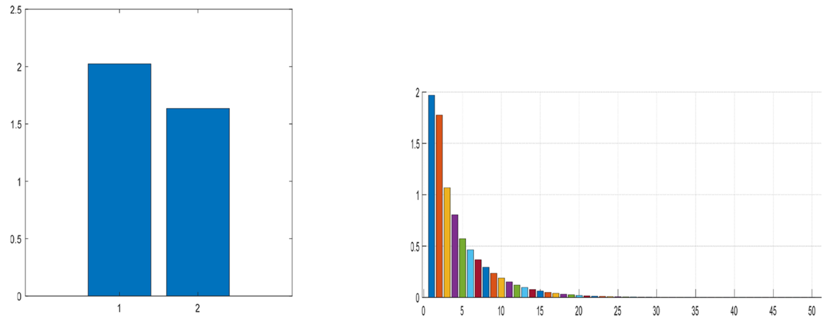
Table 4.12: Comparing the MLE with LSM in both models VECM and SSECM.

| Model 4 | $T = 100$ | | $T = 750$ | | $T = 2000$ | |
|---|---|---|---|---|---|---|
| | VECM | SSECM | VECM | SSECM | VECM | SSECM |
| MLE | 8324.65 | 8328.89 | 8332.86 | 8328.07 | 8331.37 | 8327.83 |
| Least squares metho | 8391.06 | 8385.82 | 8397.74 | 8388.02 | 8398.48 | 8388.12 |

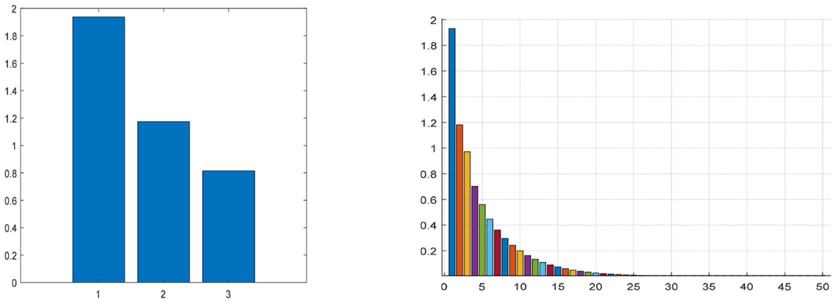
Table 4.13: Comparing the theta parameter vector for each model (MLE and LSM) with true theta.

| Model 4 | $T = 100$ | $T = 750$ | $T = 2000$ | True Theta from the State-Space Data Generating Process |
|---|---|---|---|---|
| LSM_theta | $\begin{matrix} 0.73282 & -0.40703 & 0.53465 & -0.62001 \\ 0.43331 & -0.78783 & 0.35147 & 0.91460 \\ -0.47694 & 0.66537 & 0.86322 & 0.06524 \end{matrix}$ | $\begin{matrix} 0.73255 & -0.407111 & 0.53520 & -0.6204 \\ 0.43588 & -0.78829 & 0.35161 & 0.91437 \\ -0.47769 & 0.66530 & 0.86355 & 0.06540 \end{matrix}$ | $\begin{matrix} 0.73250 & -0.40794 & 0.53518 & -0.6206 \\ 0.435105 & -0.78830 & 0.35166 & 0.91439 \\ -0.47767 & 0.66531 & 0.86355 & 0.06542 \end{matrix}$ | |
| MLE_ theta | $\begin{matrix} 0.73248 & -0.40725 & 0.53498 & -0.6207 \\ 0.43353 & -0.78796 & 0.35132 & 0.91422 \\ -0.47632 & 0.66501 & 0.86378 & 0.06549 \end{matrix}$ | $\begin{matrix} 0.73242 & -0.407103 & 0.53518 & -0.6212 \\ 0.43591 & -0.78816 & 0.35169 & 0.91429 \\ -0.47756 & 0.66519 & 0.86348 & 0.06543 \end{matrix}$ | $\begin{matrix} 0.73248 & -0.40797 & 0.53520 & -0.6209 \\ 0.435100 & -0.78825 & 0.35164 & 0.91435 \\ -0.47763 & 0.66528 & 0.86356 & 0.06549 \end{matrix}$ | $\begin{matrix} 0.73248 & -0.40799 & 0.53523 & -0.6209 \\ 0.43598 & -0.78821 & 0.35164 & 0.91431 \\ -0.47761 & 0.66522 & 0.86352 & 0.06545 \end{matrix}$ |

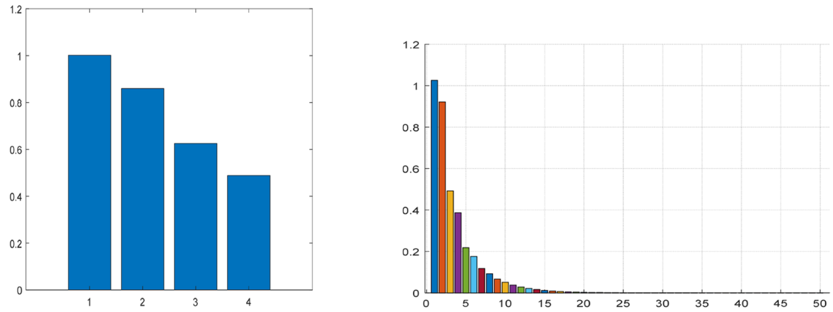
Indeed, we compare between the delay profiles for VECM and SSECM that were estimated by the norm squared of the coefficient matrices as presented in Figure 4.2. It is clear that the difference between these quantities for VECM and SSECM is small wherever VECM has a nonzero value. Also, the rest of these quantities converge to zero in the SSECM case. Here by delay profile of a system $\{u_t\} \rightarrow \{y_t\}$, we mean the sequence of norm-squared values for inputs response matrices of the inverse system $\{y_t\} \rightarrow \{u_t\}$.



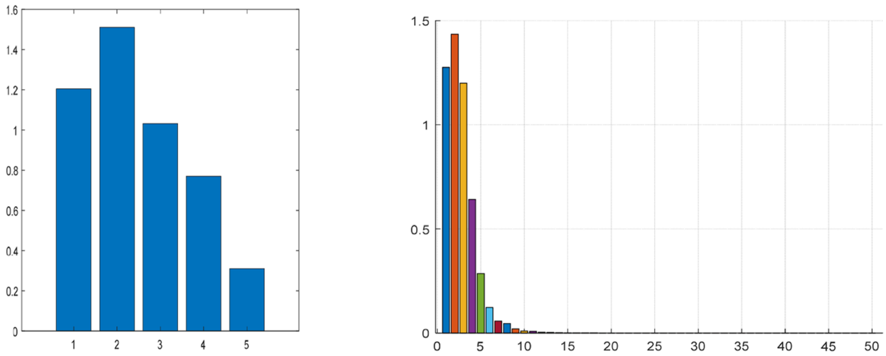
(a) $T = 100$



(b) $T = 500$



(c) $T = 1000$



(d) $T = 2000$

Figure 4.2: The Delay Profiles; the Figures on the Left Display the Delay Profiles for the VECM.The Figures on the Right Display the Delay Profiles for the SSECM.

For further comparison, the study used the data set with 2000 observations. These models were compared based on three Information Criteria (AIC, BIC, and HQC) and Bayes factors (see, Poskitt and Tremayne (1987) Atukeren (2005)), used as the model selection tools. These can be used to compare any models fitted to the same data. For illustrative purposes, traditional fit measures ($R^2$ and the root mean squared error – RMSE) were also used to compare the SSECM with the traditional VECM.

Figure 4.3 and Table 4.14 show the results of using the information criteria to compare the SSEC and VEC models. In the case of the BIC, this criterion suggests that SSEC is the model that minimizes the information loss; in contrast, the lower values of both the AIC and the HQC point to the traditional VEC as the preferred model.

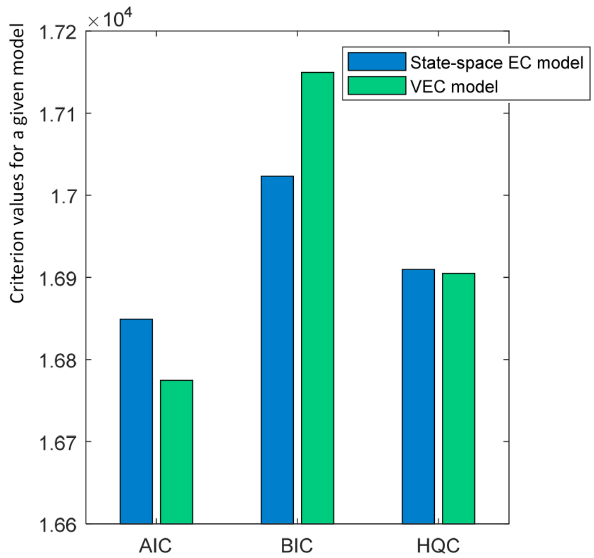



Figure 4.3: Comparison Between the SSEC and the VEC Model Using Three Information Criteria (AIC, BIC, and HQC)

Table 4.14: Comparison Between the SSEC and the VEC Model Using Three Information Criteria (AIC, BIC, and HQC)

| | AIC | BIC | HQC |
|---|---|---|---|
| SSVEC | 16849.1 | 17023.3 | 16909.6 |
| VEC | 16774.7 | 17149.7 | 16904.9 |

Berger and Pericchi (1996) suggested that model selection should have a Bayesian basis. This is because non-Bayesian methods (Poskitt and Tremayne 1987) – based on the introduction of *ad hoc* penalties for model complexity – have difficulty incorporating the "Occam's razor" principle. It refers to the notion that if two models explain data equally well, then the simpler model is to be preferred. Bayes factors do this "Occam's razor" comparison automatically (Atukeren 2005). In the Bayesian approach, given an information set $\mathbf{\Phi}$, which includes the data set and (possibly) other additional information, a measure of evidence in favor of a model $\mathcal{M}_0$ is the posterior odds of model $\mathcal{M}_0$ versus the other model, i.e. the posterior probability of the additional model $\mathcal{M}_1$ divided by the posterior probability of the first model:

$$\mathbb{p}(\mathcal{M}_1|\mathbf{\Phi})/\mathbb{p}(\mathcal{M}_0|\mathbf{\Phi}).$$

Based on the posterior probabilities $\mathbb{p}(\mathcal{M}_1|\mathbf{\Phi})$, $\mathbb{p}(\mathcal{M}_0|\mathbf{\Phi})$ and the prior probabilities $\mathbb{p}(\mathcal{M}_0)$, $\mathbb{p}(\mathcal{M}_1)$, a commonly used measure of evidence in favor of $\mathcal{M}_0$ is the Bayes factor, defined by (Wasserman 2000):

$$\mathcal{B}_{01} = \frac{\mathbb{p}(\mathcal{M}_1|\mathbf{\Phi})}{\mathbb{p}(\mathcal{M}_0|\mathbf{\Phi})} \div \frac{\mathbb{p}(\mathcal{M}_1)}{\mathbb{p}(\mathcal{M}_0)},$$

which is the posterior odds in favor of $\mathcal{M}_1$ divided by the prior odds in favor of $\mathcal{M}_1$. The Schwarz criterion (or BIC) gives a rough approximation to the algorithm of the Bayes factor $\mathcal{B}_{01}$. This approximation avoids the need to elicitate the prior distributions of $\mathbb{p}(\mathcal{M}_1)$ and $\mathbb{p}(\mathcal{M}_0)$ (Kass and Raftery 1995). Following Wagenmakers (2007):

$$\mathcal{B}_{01} \approx \exp\left(\frac{\Delta \mathrm{BIC}_{10}}{2}\right),$$

and the table below can be used to take a decision about the evidence in favor of $\mathcal{M}_0$ (see Kass and Raftery 1995):

Table 4.15: Approximate Minimum Bayes Factor $\mathcal{B}_{01}$ and Probability in Favor of the Model $\mathcal{M}_0$ with Evidence

| Bayes factor $\mathcal{B}_{01}$ | Probability in favor of the model $\mathcal{M}_0$ | Evidence |
|---|---|---|
| 1-3 | .50-.75 | Weak |
| 3-20 | .75-.95 | Positive |
| 20-150 | .95-.99 | Strong |
| $> 150$ | $> .99$ | Very strong |

In the case of the comparison of the model SSEC ($\mathcal{M}_0$) against the VEC model ($\mathcal{M}_1$),

$$\Delta \mathrm{BIC}_{10} = \mathrm{BIC}(\mathcal{M}_1) - \mathrm{BIC}(\mathcal{M}_0) = 17149.7 - 17023.3 = 126.4,$$

and thus $\mathcal{B}_{01} \approx 2.8 \times 10^{27}$, providing very strong evidence in favor of the SSEC model when compared to the simple VEC model.

For illustration purposes, model comparison based on measures of goodness of fit ($R^2$ and RMSE) were also used to compare the SSEC model and the VEC model. Also, the number of observations to work with the model were taken to be $T = 2000$. Nonetheless, care must be taken when comparing models with these measures, since the ordinary $R^2$ and RMSE do not penalize model complexity, i.e. an artificially good fit (a high $R^2$ and low RMSE) can be obtained with a complex model that over-fits the data. The $R^2$ and RMSE statistics suggest that a better fit for $y_{1,t}$ and $y_{2,t}$ are obtained with the SSEC model, since the $R^2$ of the SSEC is slightly higher than the $R^2$ of the VEC model for the variables $y_{1,t}$ and $y_{2,t}$ of the vector $\mathbf{y}_t$. The RMSE of the SSEC model is also lower than the RMSE of the VEC model for the variables $y_{1,t}$ and $y_{2,t}$. Only in the case of the variable $y_{3,t}$ of the vector $\mathbf{y}_t$ the $R^2$ of the VEC model is slightly higher and the RMSE is lower as clear as shown in Table 4.16

Table 4.16: Comparison Between the SSECM and the VECM Based on the Measures of the Goodness of Fit ($R^2$ and RMSE).

| | Ordinary $R^2$ statistic | | Root mean squared error (RMSE) | |
|---|---|---|---|---|
| | SSEC | VEC | SSEC | VEC |
| $y_{1,t}$ | .9861 | .9444 | 13.854 | 27.711 |
| $y_{2,t}$ | .2788 | .2492 | 40.835 | 41.656 |
| $y_{3,t}$ | .9207 | .9357 | 45.871 | 41.334 |

## 4.4 Discrete-time Simulation Study (case 2: $p > n$)

In this section, we will compare the classical VECM approach with the SSECM approach in case 2 when $p > n$.

### 4.4.1 Model Estimation

In this section, we will present estimation results for case $p > n$ and note that $\beta_2$ here will appear as presented in section 2.4.3. Now, consider one state-space system $(A, B, C)$ with $p = 6$ and state dimension $n = 2$. Here, the system with $c = 1$ and the eigenvalues of the $A$ matrix in a minimal state-space representation $(\lambda_1, \lambda_2)$ are studied and results given in Table 4.17.

Table 4.17: The Eigenvalues of the $A$-matrix in a Minimal State-Space Representation of the Generating Model for the Case 2

| | $c = 1$ |
|---|---|
| Sigma(A) | (1,0.9964) |

Simulation data comprising $T = 2000$ output observations $(y_1; \ldots; y_T)$ have been generated.

A VECM model has been estimated for this data set. Table 4.18 shows that the Akaike information criteria suggested an optimal lag order of two for the VECM. An SSEC model was also estimated for the same data set. Here, it was assumed that the order is 2 (= the correct order).

Table 4.18: Lag Order Selection Results for the Discrete-time Simulation Study

| Lag order | Model AIC |
|---|---|
| 1 | 37387.33 |
| 2 | 34667.35 |
| 3 | 34801.74 |
| 4 | 34880.89 |
| 5 | 35106.32 |
| 6 | 35306.33 |
| 7 | 35505.73 |
| 8 | 35735.94 |
| 9 | 35956.61 |
| 10 | 36161.21 |
| 11 | 36384.93 |
| 12 | 36591.28 |

The Trace Cointegration Test was used to evaluate the number of cointegrating relations (if any) between $\{y_t\}_t$. Table 4.19 shows the results of applying the Cointegration Test to the generated data. The results show that the null hypothesis of no cointegrating relationships ($r = 0, 1, 2, 3, 4$) can be rejected at a significance level of 95%, giving strong evidence for the presence of cointegration. The null hypothesis of at most 5 cointegrating relation can not be rejected at a significance level of 95%. This is in line with the true model as $r = 5$.

Table 4.19: Results of the VECM Trace Cointegration Test and State-Space Trace Cointegration Test

| Rank (r) ($T = 2000$) | $r = 0$ | $r = 1$ | $r = 2$ | $r = 3$ | $r = 4$ | $r = 5$ |
|---|---|---|---|---|---|---|
| VECM Trace Statistic | 197.0159 | 150.9098 | 131.8520 | 102.6618 | 96.365 | 0.4383 |
| State Space Trace Statistic | 109.746 | 99.6473 | 94.758 | 93.847 | 90.647 | 0.5748 |
| Critical Value (own code) | 83.3133 | 59.6193 | 39.9665 | 24.1994 | 12.2744 | 4.0969 |
| $p$-value | 0.0010 | 0.0010 | 0.0010 | 0.0010 | 0.0010 | 0.6849 |

### 4.4.2 Comparison Between the Outcomes of VECM, SSECM and the True Model

We compared the true model with estimation results as in Table 4.20. The differences between the true $\theta$ and the MLE $\theta$ is found to be small. Furthermore, Table 4.21 shows the cointegration vector estimates are similar for both model classes and the true model.

Table 4.20: True Theta Parameter Values from the State-Space Data Generating Process and the Corresponding ML Estimator Values.

| True theta | | MLE of theta | |
|---|---|---|---|
| −0.72142 | 0.63845 | −0.74763 | 0.65674 |
| −0.56782 | 0.68991 | −0.57263 | 0.68888 |
| 0.64503 | 0.12418 | 0.70465 | 0.10334 |
| −0.04211 | 0.53771 | −0.03546 | 0.49378 |
| 0.69631 | 0.18104 | 0.65798 | 0.19943 |
| 0.16815 | 0.11382 | 0.18563 | 0.11187 |

Table 4.21: Cointegration Vector

| Model | Cointegration Vector | | | | |
|---|---|---|---|---|---|
| True model | −0.023 | 0.781 | 1.761 | −4.131 | −3.761 |
| | 1.471 | −2.192 | −0.569 | 0.981 | 4.233 |
| | −3.152 | 1.835 | 2.013 | −2.637 | −2.812 |
| | 2.934 | −0.113 | −0.401 | 5.191 | 1.585 |
| | 0.099 | −0.231 | −3.091 | 0.080 | 0.940 |
| | −0.383 | 0.085 | 0.158 | −0.001 | −0.075 |
| VECM | −0.029 | 0.788 | 1.766 | −4.137 | −3.764 |
| | 1.501 | −2.193 | −0.573 | 0.978 | 4.238 |
| | −3.158 | 1.831 | 2.016 | −2.633 | −2.814 |
| | 2.937 | −0.117 | −0.404 | 5.189 | 1.588 |
| | 0.091 | −0.230 | −3.096 | 0.083 | 0.939 |
| | −0.384 | 0.089 | 0.153 | −0.003 | −0.074 |
| SSECM | −0.022 | 0.783 | 1.760 | −4.132 | −3.761 |
| | 1.470 | −2.190 | −0.568 | 0.981 | 4.235 |
| | −3.155 | 1.833 | 2.013 | −2.638 | −2.814 |
| | 2.936 | −0.114 | −0.402 | 5.192 | 1.584 |
| | 0.097 | −0.231 | −3.093 | 0.081 | 0.941 |
| | −0.381 | 0.084 | 0.157 | −0.001 | −0.076 |

We compared the SSECM and VECM. Table 4.22 shows, once again, that the SSECM performs better than the VECM. With similar sample size, the SSECM has a remarkably lower number of parameters, lower BIC and yields a similar log likelihood. It's clear to see that the number of parameters in the VECM grows very fast, while the SSECM is not prone to become over-parameterized. It should be said that the larger the number of estimated parameters, the larger the uncertainty in the estimation. By the end, we compared the delay profiles for VECM and SSECM.

Table 4.22: Comparison of the Estimated VAR Error Correction Model and the Estimated SSECM for $T = 2000$

| $T = 2000$ | **VECM ($k = 1$)** | **SSECM ($n = 2$)** |
|---|---|---|
| **Cointegration rank** | 5 | 5 |
| **Effective sample size** | 1997 | 1998 |
| **Total Parameters** | 92 | 20 |
| **BIC** | 34984.62 | 34165.72 |
| **Neg. Log likelihood** | 17060.32 | 16988.53 |

We compared the delay profiles for the VECM and the SSECM. The findings showed the coefficient matrices as presented in Figure 4.4. It is clear that the difference between these quantities for the VECM and the SSECM is small wherever VECM has a nonzero value. Also, the rest of these quantities converge to zero in the SSECM case. Here, by delay profile of a system $\{u_t\} \to \{y_t\}$, one means the sequence of norm-squared values for inputs response matrices of the inverse system $\{y_t\} \to \{u_t\}$.

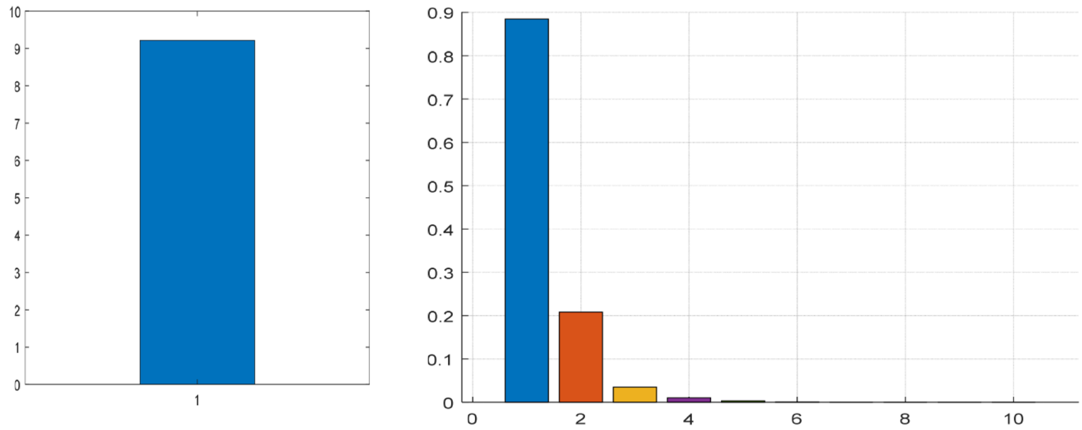



Figure 4.4: The Delay Profiles; the Figure on the Left Display the Delay Profile for the VECM; the One on the Right Display the Delay Profile for the SSECM.

## 4.5 Continuous-time Simulation Study (case 1: $p \leq n$)

### 4.5.1 Model Estimation

In this section, we will present estimation results for case $p \leq n$. We want to model a continuous-discrete time system. We will use a continuous-time stat-space system $(A_c, B_c, C_c)$ parameterized by a corresponding (artificial) discrete-time state-space system $(A_\mu, B_\mu, C_\mu)$ with $p = 3$ and state dimension $n = 4$.. Here we study the system with $c = 2$ and the eigenvalues of the $A_\mu$ matrix in a minimal state-space representation $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ are given in Table 4.23. Simulation data comprising $N = 2000$ output vector observations $(y_{t_1}; \ldots; y_{t_N})$ have been generated.

The Trace Cointegration Test was used to evaluate the number of cointegrating relations (if any) between $\{y_{t_j}\}_{j=1}^{N}$. Table 4.24 shows the results of applying the Cointegration Test to the generated data. The results show that the null hypoth-

Table 4.23: The Eigenvalues of the $A_\mu$-matrix and the (continuous-time) $A_c$-matrix in a Minimal State-Space Representation of the Generating Models

| | $c = 2$ |
|---|---|
| Sigma($A_\mu$) | (0.6069,0.9991,1,1) |
| Sigma($A_c$) | $(-3.2579, -0.0185, 0, 0)$ |

esis of no cointegrating relationships ($r = 0$) can be rejected at a significance level of 95%, giving strong evidence for the presence of cointegration. The null hypothesis of at most one cointegrating relation (rank $r \leq 1$ against the alternative $r > 1$) can not be rejected at a significance level of 95%. This is in line with the under lying true model, for which $r = 1$.

Table 4.24: Results of the State-Space Trace Cointegration Test

| Rank (r) ($T = 2000$) | $r = 0$ | $r = 1$ | $r = 2$ |
|---|---|---|---|
| State Space Trace Statistic | 37.786 | 9.6433 | 1.7458 |
| Critical Value (own soft) | 24.1994 | 12.3079 | 4.0984 |

### 4.5.2 Comparison Between the Outcomes of SSECM and the True Model

We compared the true model with estimation results as in Table 4.25. The differences between the true $\theta$ and the MLE $\hat{\theta}$ can be seen to be small. The method

Table 4.25: True Theta Parameter Values from the State-Space Data Generating Process and the Corresponding ML Estimator Values for Comparing the SSECM for $\lambda = 159$.

| True theta | | | | MLE theta | | | |
|---|---|---|---|---|---|---|---|
| −0.58911 | 0.83562 | 0.84378 | −0.08856 | −0.55785 | 0.88785 | 0.83371 | −0.0854 |
| 0.58591 | 0.32556 | −0.17776 | 0.33675 | 0.54439 | 0.39932 | −0.176823 | 0.35379 |
| 0.80842 | −0.24065 | −0.11531 | −0.065841 | 0.83291 | −0.23463 | −0.11833 | −0.06894 |

was repeated several times with different values of $\lambda$ where

$$\Delta t_j = -\log\left(1 - \frac{U_j}{\lambda}\right),$$

$U \sim U(0,1)$ and $j = 1, \ldots, N$. However, with large value of $\lambda$, e.g. $\lambda = 159$, better comparison was seen between the $\theta$ and the $\hat{\theta}$. The neg.log likelihood in the optimum was 8147.72 and BIC equal to 16548 with total number of parameters

29 and the effective sample size is 1987. Furthermore, Table 4.26 shows that the cointegration vector estimates are similar to the cointegration vector in the true model.

Table 4.26: Cointegration Vector

| True model | SSECM |
|---|---|
| $-0.85$ | $-0.83$ |
| $2.59$ | $2.57$ |
| $-1.74$ | $-1.76$ |

## 4.6 Continuous-time simulation study (case 2: $p > n$)

### 4.6.1 Model Estimation

In this section, we will present estimation results for case $p > n$ and note that $\beta_2$ here will appear as presented in section 2.4.2. Consider one state-space system $(A_\mu, B_\mu, C_\mu)$ with $p = 6$ and state dimension $n = 2$. Here, we study the system with $c = 1$ and the eigenvalues of the $A_\mu$ matrix in a minimal state-space representation $(\lambda_1, \lambda_2)$ are given in Table 4.27. Simulation data comprising $N = 2000$ output observations $(y_{t_1}; \ldots; y_{t_N})$ have been generated.

Table 4.27: The Eigenvalues of the $A_\mu$-matrix and the (continuous-time) $A_c$-matrix in a Minimal State-Space Representation of the Generating Models for the case 2.

| | $c = 1$ |
|---|---|
| Sigma$(A_\mu)$ | $(-0.0533, 1)$ |
| Sigma$(A_c)$ | $(-1.6942, 0)$ |

The Trace Cointegration Test was used to evaluate the number of cointegrating relations (if any) between $\{y_{t_j}\}_{j=1}^N$. Table 4.28 shows the results of applying the Cointegration Test to the generated data. The results show that the null hypothesis of no cointegrating relationships ($r = 0, 1, 2, 3, 4$) can be rejected at a significance level of 95%, giving strong evidence for the presence of cointegration. The null hypothesis of at most 5 cointegrating relation can not be rejected at a significance level of 95%. This is in line with the true model as $r = 5$.

Table 4.28: Results of the State-Space Trace Cointegration Test for case 2

| Rank (r) ($T = 2000$) | $r = 0$ | $r = 1$ | $r = 2$ | $r = 3$ | $r = 4$ | $r = 5$ |
|---|---|---|---|---|---|---|
| State Space Trace Statistic | 109.746 | 99.6473 | 94.758 | 93.847 | 90.647 | 0.5748 |
| Critical Value (own soft) | 83.3133 | 59.6193 | 39.9665 | 24.1994 | 12.2744 | 4.0969 |

### 4.6.2 Comparison Between the Outcomes of the SSECM and the True Model

The study compared the true model with estimation results as in Table 4.29. The differences between the true $\theta$ and the MLE $\hat{\theta}$ can be seen to be small. However, when $\lambda$ equals to 173 ( which is the reference value), we get the better comparison between the $\theta$ and the $\hat{\theta}$.

Table 4.29: True Theta Parameter Values from the State-Space Data Generating Process and the Corresponding ML Estimator Values for case 2 where $\lambda = 173$

| True theta | | MLE theta | |
|---|---|---|---|
| −0.72142 | 0.63845 | −0.74763 | 0.65674 |
| −0.56782 | 0.68991 | −0.57263 | 0.68888 |
| 0.64503 | 0.12418 | 0.70465 | 0.10334 |
| −0.04211 | 0.53771 | −0.03546 | 0.49378 |
| 0.69631 | −0.18104 | 0.65798 | −0.19943 |
| 0.16815 | 0.11382 | 0.18563 | 0.11187 |

The neg.log likelihood in the optimum was 18146.33 and BIC equal to 36140.45 with total number of parameters 20 and the effective sample size is 1998. Furthermore, Table 4.30 shows the cointegration vector estimates are similar for the model and the true model.

Table 4.30: Cointegration Vector

| True model | | | | | SSECM | | | | |
|---|---|---|---|---|---|---|---|---|---|
| −1.55 | −2.34 | 1.65 | 1.01 | 5.03 | −1.52 | −2.33 | 1.66 | 1.04 | 6.06 |
| −0.001 | −0.18 | 2.47 | 3.73 | 1.72 | −0.003 | −0.20 | 2.45 | 3.76 | 1.69 |
| 2.03 | 1.87 | −0.56 | −0.77 | −4.22 | 2.04 | 1.91 | −0.58 | −0.73 | −4.18 |
| 0.10 | 4.01 | −0.97 | −1.12 | −2.19 | 0.12 | 4.03 | −0.99 | −1.11 | −2.19 |
| −1.91 | −3.29 | 1.33 | 0.27 | 0.98 | −1.94 | −3.31 | 1.37 | 0.30 | 0.96 |
| 1.12 | 0.36 | −4.02 | −4.18 | −1.11 | 1.15 | 0.37 | −4.05 | −4.21 | −1.12 |

In conclusion, this chapter estimated the discrete-time simulated data and compared the findings to estimation results using the VAR model. The main findings of this chapter are as follow:

- First,there is a similarity between the log likelihood estimates for both model classes
- Second, VECM has a higher number of parameters compared to SSECM
- Third, the BIC information criteria are much lower in several cases for the SSECM, which favours such model class in terms of model selection
- Fourth, SSECM outperforms, if not reformulate the VECM regarding model selection criteria including goodness of fit, forecast accuracy (RMSE) and Bayes factor.
- Fifth, there is a clear difference between the delay profiles of VECM and the SSECM. Unlike the VECM case, the quantities converge rather smoothly to zero in the SSECM case.

In contrast, the results of the continuous-discrete time model suggest that the model has several advantages over conventional discrete-time series and VAR models. For example, data with unequal time intervals can be treated efficiently, since the model parameters of the dynamical system model are not affected by the measurement process. The continuous-discrete state space model is a combination of continuous time dynamics (stochastic differential equations, SDE) and discrete-time noisy measurements. As is well-known, the crucial point is the implementation of the nonlinear constraints directly. An inspection of the empirical distribution of the ML estimates reveals that the parameter values are nearer to normality.

Furthermore, our findings established that for a higher p-value, VAR is not an appropriate model, as shown by Tschernig et al. (2013). The findings suggest that state-space models are better as estimation models with higher $p$-values instead of VAR models. Also, Ribarits and Hanzon (2014b,a) offered a logical explanation in favouring the state-space models over vector-auto-regressive methodologies. These reasons includes statistical parsimony and flexibility for different applications.

Furthermore, our study compared a true model with estimated SSECM results. The difference between estimates in the true model and the one used in the maximum likelihood estimation for the state-space model is small. When the number of observations equals 100, the estimates are not close to the true (values of the

$\theta$ parameters); however, when the same value is used with 2000 observations, it yields a better result.We also conducted a state-space estimation with order $n = 3$ and $n = 4$, using the same data set and found the sample co-variance matrix of epsilon $t$. The results were very close, as the result of $n = 3$ was slightly larger than of $n = 4$.

Lastly, the chapter presented a simulation study in continuous time. The chapter also compared the state-space framework and the true mode, as neither VAR nor VECM methodologies have been extended to continuous-time applications. Similar results were obtained for the SSECM cointegration vector and the values of the $\theta$ parameters.

# Chapter 5

# Application of the Study to Real Data

This study proposed a framework for the analysis of state-space models with discrete-time and continuous-time parameters. Discrete-time state-space models provide the same type of linear difference relationship between the inputs and outputs as the linear ARMA model, but are rearranged such that there is only one delay in the expressions. Each ARMA model can be (re-)written in state-space form and vice versa. Using a cointegration exercise, this chapter will show that the proposed method works well for several different data.

The objective of this chapter is to evaluate the data sets and carry out a cointegration analysis. The analysis will be useful in comparing the results of the VECM and SSECM. The analysis will comprise of estimates which will be tested for stationarity. The stationarity of a time series can be tested using standard software. This study used the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test using statistical software. The tests were applied to distinctive data sets across different time frames, and all tests showed non-stationarity.

Indeed, in this chapter we will build on the findings of the previous chapter. In the last chapter, we estimated the discrete-time model and compared the findings with the results from the VAR model. In this chapter, we will apply our findings to real-world data.

## 5.1 An Application to the Oil prices

This section comprises the estimation technique, applied to time series data consisting of the daily *Oil prices*. This concerns the prices of West Texas, Brent, and Dubai for the period 6/22/1992 to 6/22/2012. Figure 5.1 shows daily oil prices. A clear non-stationary pattern is observed in all price indices.

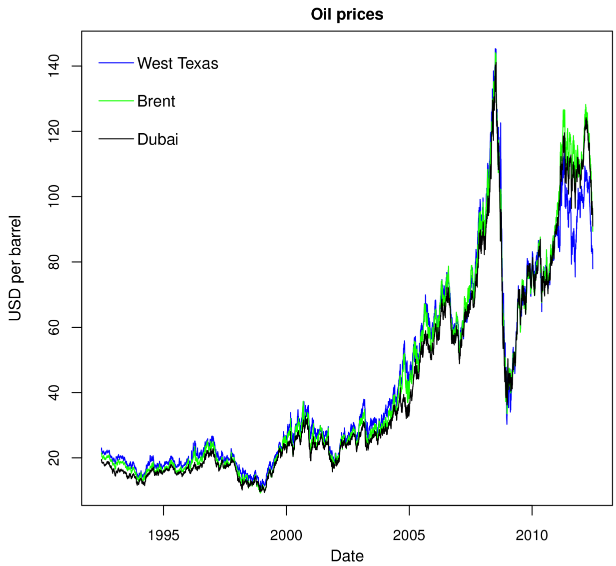



Figure 5.1: Daily Oil Prices (West Texas, Brent, and Dubai)

### 5.1.1 Cointegration Vector Analysis

The Trace Cointegration Test was used to evaluate the number of cointegrating relations among the three time series. Table 5.1 shows the results of applying the Cointegration Test to the three time series. We did not assume any intercepts or trends in the testing model. The results show that the null hypothesis of no cointegrating relationships ($r = 0$) can be rejected at a significance level of 95%, giving strong evidence for the presence of cointegration. The null hypothesis of at most 1 cointegrating relation (rank $r \leq 1$ against the alternative $r > 1$) can be rejected at a significance level of 95%. The null hypothesis of at most 2 cointegrating relation (rank $r \leq 2$ against the alternative $r > 2$) can not be rejected at a significance level of 95%. We chose 95% confidence interval as it is generally used and provides a range of values that guarantees to be within the true mean of the populace. Also, usually with large samples, this confidence interval provides precise values. This research study also has large sample values; hence, a 95% confidence interval provides precision and is narrower for a large sample compared to a 90% confidence interval.

Table 5.1: Results of the Trace VECM Cointegration Test and Trace State-Space Cointegration Test

| Rank ($r$) | VECM Trace Statistic | State-Space Trace Statistic | Critical Value (own code) | Critical Value (Johansen Matlab code) | $p$-value |
|---|---|---|---|---|---|
| $r = 0$ | 87.1830 | 90.324 | 83.3133 | 83.9402 | 0.0010 |
| $r \leq 1$ | 63.3510 | 68.658 | 59.6193 | 60.0623 | 0.0336 |
| $r \leq 2$ | 0.0111 | 1.0027 | 39.9665 | 40.1751 | 0.9316 |

Table 5.2 shows Brent as the weighted average of the other two in the context of the cointegrating relation. Similarly, West Texas is approximately a weighted average of other two.

Table 5.2: Estimated Beta Vector

| Company | West Texas | Brent | Dubai |
|---|---|---|---|
| VECM | 0.0809 | −0.5710 | 0.5001 |
| | −0.1913 | 0.0657 | 0.1211 |
| SSECM | 0.0751 | −0.5845 | 0.5100 |
| | −0.1862 | 0.0741 | 0.1133 |

The financial crisis of 2008 has highlighted the macroeconomic volatility across time series data. We took the oil price as an example of reflection as we do not have more time to test the financial crisis of 2008 to the rest of our real data. First, we divided our data into two time periods, i.e., oil prices before and after 2008 to compare the co-integration vector and the parameters by comparing the co-integration vector estimated by VECM and SSECM, respectively. Table 5.3 and 5.4 shows the results that there is not much variation between the parameter values before or after 2008, and they are more or less identical. The results indicate that Oil and equity price movements were highly correlated earlier. However, after 2008 they were insignificantly correlated dependent on the sample size. It is clear from Figure 5.1 that the pre-2008 period shows a strong positive association.

Table 5.3: Estimated Beta Vector after 2008

| Company | West Texas | Brent | Dubai |
|---|---|---|---|
| VECM | 0.0784 | −0.5608 | 0.4931 |
| | −0.2001 | 0.0605 | 0.1198 |
| SSECM | 0.0747 | −0.5786 | 0.5112 |
| | −0.1903 | 0.0733 | 0.1140 |

Table 5.4: Estimated Beta Vector before 2008

| Company | West Texas | Brent | Dubai |
|---|---|---|---|
| VECM | 0.0804 | −0.5708 | 0.5017 |
| | −0.1915 | 0.0655 | 0.1209 |
| SSECM | 0.0753 | −0.5823 | 0.5101 |
| | −0.1858 | 0.0742 | 0.1130 |

In comparison, the co-integration between oil prices was stronger before 2008. There are two possible reasons for this: a) the model used a large sample size resulting; b) each of these associations is linked with some important macroeconomic variable also influencing the financial crisis of 2008. It implies that economic shocks in a developed market have a spillover effect.

The research study compared the MLE parameters as shown in Table 5.5. The parametric before 2008 was closer to MLE parameters over all the data sets than parametric after 2008 as we have more observation before 2008. Indeed, the difference was not that big.

Table 5.5: Comparing the MLE parameters before and after 2008 with all the data sets

| MLE of theta with effect sample size $N = 5219$ | | | |
|---|---|---|---|
| 0.39047 | 0.60218 | −0.20148 | −0.55812 |
| −0.24194 | 0.57261 | 0.42981 | 0.78439 |
| -0.47317 | −0.13721 | −0.83758 | 0.28945 |
| **MLE of theta after 2008, with effect sample size $N = 1305$** | | | |
| 0.28736 | 0.59776 | -0.18835 | -0.63651 |
| −0.38746 | 0.66458 | 0.47465 | 0.87563 |
| −0.40013 | −0.19846 | −0.94673 | 0.33745 |
| **MLE of theta before 2008,with effect sample size $N = 3914$** | | | |
| 0.40013 | 0.62013 | −0.20746 | −0.54872 |
| −0.23634 | 0.57132 | 0.43121 | 0.76035 |
| -0.46917 | −0.13652 | −0.83883 | 0.28912 |

### 5.1.2 VECM & SSECM Maximum Likelihood Estimation and Analysis

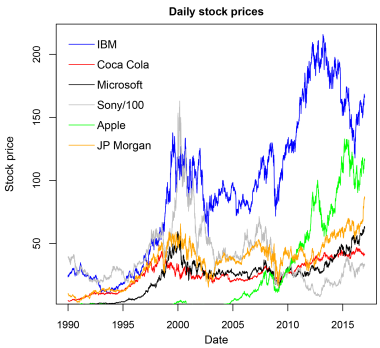
The comparison of these models was performed using an information criterion. Table 5.6 shows the results of using the BIC to compare the SSECM against the VECM, SSECM is the model that minimizes the information criterion.

Table 5.6: Results of VECM with Lags $k = 3$ and Effective Sample Size $N = 5216$ & SSECM with $n = 4$ and Effective Sample Size $N = 5219$

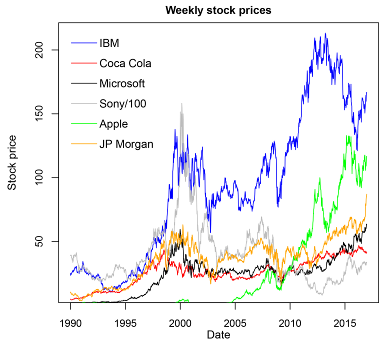
| Model | Log likelihood | BIC | Total Parameters |
|---|---|---|---|
| VECM | 19037.57 | 38471.11 | 41 |
| SSECM | 19053.34 | 37884.12 | 26 |

## 5.2 An Application to the Stock Prices of the Six Companies

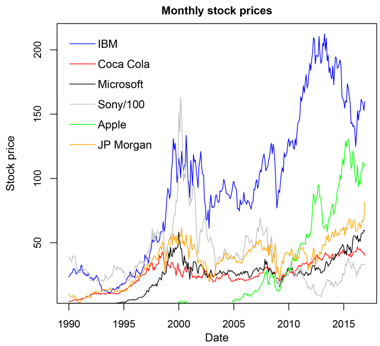
In this section, the study will analyze a time series of stock prices for the following six companies: *(Apple US\$, Coca-Cola US\$, Sony JPY, Microsoft US\$, IBM US\$, JP Morgan US\$)* from 1/1/1990 to 12/30/2016. Daily, weekly and monthly series are available over this time period. Figure 5.2 shows the (monthly, weekly, daily) prices. A clear non-stationary pattern is observed in the all price indices.




(a) Daily price




(b) Weekly price




(c) Monthly price

Figure 5.2: Stock Prices of the Six Companies (Apple US, Coca-Cola US, Sony JPY, Microsoft US, IBM US, and JP Morgan US)

### 5.2.1 Cointegration Vector Analysis

The Trace Cointegration Test was used to evaluate the number of cointegrating relations between the six time series. Table 5.7 shows the results of applying the

Table 5.7: Results of the Cointegration Test: Monthly Data

| Rank ($r$) | VECM Trace Statistic | State-Space Trace Statistic | Critical Value (own code) | Critical Value (Matlab code) | $p$-value |
|---|---|---|---|---|---|
| $r = 0$ | 98.6509 | 99.7527 | 83.3133 | 83.9402 | 0.0036 |
| $r \leq 1$ | 56.1723 | 55.9487 | 59.6193 | 60.0623 | 0.1021 |
| $r \leq 2$ | 28.2257 | 27.9665 | 39.9665 | 40.1751 | 0.4811 |
| $r \leq 3$ | 14.2671 | 14.0564 | 24.1994 | 24.2747 | 0.5459 |
| $r \leq 4$ | 2.9764 | 3.0211 | 12.2744 | 12.3206 | 0.8506 |
| $r \leq 5$ | 0.1122 | 0.023 | 4.0969 | 4.1302 | 0.7942 |

Cointegration Test to the monthly series of the prices. A VECM(1) model was used for each of the three frequencies considered (daily, weekly, monthly) as this turned out to be the best choice. The results show that the null hypothesis of no cointegrating relationships ($r = 0$) can be rejected at a significance level of 95%, giving strong evidence for the presence of cointegration. The null hypothesis of one cointegrating relation (rank $r \leq 1$) can not be rejected at a significance level of 95%. The weekly data have the same result as monthly data as shown in Table 5.8.

Table 5.8: Results of the Cointegration test: Weekly Data

| Rank ($r$) | VECM Trace Statistic | State-Space Trace Statistic | Critical Value (own code) | Critical Value (Matlab code) | $p$-value |
|---|---|---|---|---|---|
| $r = 0$ | 102.3697 | 101.6599 | 83.3133 | 83.9402 | 0.0016 |
| $r \leq 1$ | 56.7770 | 58.0098 | 59.6193 | 60.0623 | 0.0535 |
| $r \leq 2$ | 28.2214 | 28.0116 | 39.9665 | 40.1751 | 0.4813 |
| $r \leq 3$ | 13.3774 | 14.1223 | 24.1994 | 24.2747 | 0.6109 |
| $r \leq 4$ | 2.3191 | 2.2931 | 12.2744 | 2.3206 | 0.9232 |
| $r \leq 5$ | 0.0044 | 0.0137 | 4.0969 | 4.1302 | 0.9569 |

In the case of the daily data Table 5.9 shows that the null hypothesis of no cointegrating relationships ($r = 0$) can be rejected at a significance level of 95%, also the null hypothesis of one cointegrating relation (rank $r \leq 1$) can be rejected at a significance level of 95% as well. The null of 2 cointegrating relationship can not be rejected at a significance level 95%. The research study has applied the Trace State-Space Cointegration Test to the monthly series of the price indices as well as weekly and daily. The results show that there is one cointegrating relationship in monthly and weekly data and there are two cointegrating relationship in daily.

Table 5.9: Results of the Cointegration Test: Daily Data

| Rank ($r$) | VECM Trace Statistic | State-Space Trace Statistic | Critical Value (own code) | Critical Value (Matlab code) | $p$-value |
|---|---|---|---|---|---|
| $r = 0$ | 99.4625 | 100.2331 | 83.3133 | 83.9402 | 0.0032 |
| $r \leq 1$ | 69.7550 | 69.1008 | 59.6193 | 60.0623 | 0.0431 |
| $r \leq 2$ | 28.7444 | 28.9889 | 39.9665 | 40.1751 | 0.4534 |
| $r \leq 3$ | 13.1240 | 13.3321 | 24.1994 | 24.2747 | 0.6293 |
| $r \leq 4$ | 2.1405 | 2.0550 | 12.2744 | 12.3206 | 0.9392 |
| $r \leq 5$ | 0.0003 | 0.0042 | 4.0969 | 4.1302 | 0.9885 |

Table 5.10 seems to suggest that in cointegrating relations the IBM company and JP Morgan stock prices are approximately a weighted average of the stock prices two other companies Microsoft and Coca-Cola.

Table 5.10: Estimated Beta Vector

| Company | monthly | | weekly | | daily | | | |
|---|---|---|---|---|---|---|---|---|
| | VECM | SSECM | VECM | SSECM | VECM | | SSECM | |
| Apple | 0.0085 | 0.0088 | 0.0011 | 0.0014 | 0.0036 | 0.0019 | 0.0038 | 0.0021 |
| Coca-Cola | 0.1030 | 0.1033 | −0.1250 | −0.1253 | −0.1301 | 0.02560 | −0.1304 | 0.2531 |
| Sony | 0.0002 | 0.0002 | −0.0000 | −0.0000 | −0.000 | 0.000 | −0.000 | 0.000 |
| Microsoft | 0.1518 | 0.1520 | −0.1644 | −0.1646 | −0.1668 | 0.3172 | −0.1670 | 0.3146 |
| IBM | −0.0037 | −0.0038 | 0.0069 | 0.0070 | 0.0077 | −0.06953 | 0.0078 | −0.06972 |
| JP Morgan | −0.1932 | −0.1934 | 0.1854 | 0.1856 | 0.1817 | −0.50612 | 0.1820 | −0.50608 |

### 5.2.2 VECM & SSECM Maximum Likelihood Estimation and Analysis

A VECM(1) model was used for each of the three frequencies considered (daily, weekly, monthly). This can be explained theoretically from the fact that stock price movements depend on present but not on past stock prices. The findings show that the VEC model gives better BIC values than the SSEC model. But in this particular case the number of parameters in the VEC model is only marginally larger than the number in the state-space model. This would probably be different if the prices of more asset categories were modeled simultaneously (i.e. if p would be larger).

Table 5.11: Results of VECM with Lags $k = 1$ and Effective Sample Size $N = 324$ & SSECM with $n = 5$ and Effective Sample Size $N = 321$: Monthly Data.

| Model | Log Likelihood | BIC | Total Parameters |
|---|---|---|---|
| VECM | 1534.12 | 2773.46 | 51 |
| SSECM | 1571.49 | 2900.64 | 42 |

Table 5.12: Results of VECM with Lags $k = 1$ and Effective Sample Size $N = 1418$ & SSECM with $n = 5$ and Effective Sample Size $N = 1415$: Weekly Data.

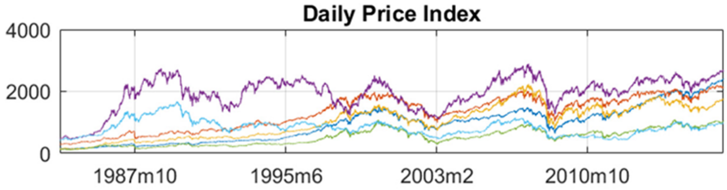
| Model | Log Likelihood | BIC | Total Parameters |
|---|---|---|---|
| VECM | 10954.42 | 21539.06 | 51 |
| SSECM | 11321.55 | 22346.65 | 42 |

Table 5.13: Results of VECM with Lags $k = 1$ and Effective Sample Size $N = 7045$ & SSECM with $n = 5$ and Effective Sample Size $N = 7042$: Daily Data.

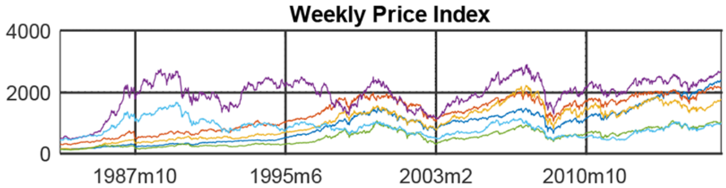
| Model | Log likelihood | BIC | Total Parameters |
|---|---|---|---|
| VECM | 18685.11 | 37000.47 | 51 |
| SSECM | 18703.01 | 37108.77 | 42 |

## 5.3 An Application to the Stock Prices Index in Six Countries

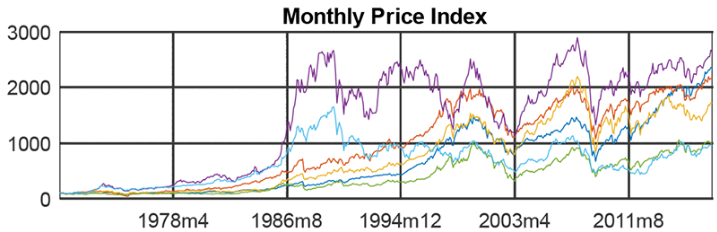
In this section, the study will analyze a time series of the six stock market *price indices* of MSCI (Morgan Stanley Capital International) for USA, UK, Europe, Pacific, German and Japan series from 1/1970 to 9/2017. Daily, weekly and monthly series are available over this time period. Figure 5.3 shows the (monthly, weekly, daily) price indices. A clear non-stationary pattern is observed in all the price indices.




(a) Daily Prices




(b) Weekly Prices




(c) Monthly Price

Figure 5.3: Stock Prices Index in the Six Countries (USA, UK, Europe, Pacific, Germany, and Japan)

### 5.3.1 Cointegration Vector Analysis

The Trace Cointegration Test was used to evaluate the number of cointegrating relations between six stock market *price indices*. Table 5.14 shows the results of applying the Cointegration Test to the monthly series of the price indices. The results show that the null hypothesis of no cointegrating relationships ($r = 0$) can be rejected at a significance level of 95%, giving strong evidence for the presence of cointegration, both with the VECM and the SSECM. The null hypothesis of at most 1 cointegrating relation (rank $r \leq 1$) against the alternative $r > 1$) can be rejected at a significance level of 95% both with VECM and SSECM, while the null hypothesis of 2 cointegrating relation (rank $r \leq 2$) can not be rejected at a significance level of 95%.

In the case of the weekly and daily data, the study revealed a discrepancy between the results of the VECM and the SSECM. In the case of VECM in the weekly data of the stock price indices, the null of no cointegration can not be rejected at a significance level 95% (Table 5.15), and similarly in the case of the daily data

Table 5.14: Results of the VECM Cointegration Test and State-Space Cointegration Test: Monthly Data

| Rank ($r$) | VECM Trace Statistic | State-Space Trace Statistic | Critical Value (own soft) | Critical Value (Matlab soft) | $p$-value |
|---|---|---|---|---|---|
| $r = 0$ | 103.5031 | 106.1023 | 83.3133 | 83.9402 | 0.0010 |
| $r \leq 1$ | 63.0875 | 97.4543 | 59.6193 | 60.0623 | 0.0273 |
| $r \leq 2$ | 26.8450 | 25.1703 | 39.9665 | 40.1751 | 0.5549 |
| $r \leq 3$ | 13.4345 | 11.5253 | 24.1994 | 24.2747 | 0.6067 |
| $r \leq 4$ | 6.9885 | 7.9973 | 12.2744 | 12.3206 | 0.3772 |
| $r \leq 5$ | 1.1299 | 1.4328 | 4.0969 | 4.1302 | 0.4529 |

of stock prices, the null of no cointegration can not be rejected at conventional significance levels (Table 5.16). However, in the case of VECM the effect of time period is more prominent than for SSECM as SSECM establishes cointegration in these high frequency series. In the case of the State-Space Cointegration Test applied to the monthly series of the price indices as well as weekly and daily, the results show that there are two cointegrating relationships in monthly, and one cointegrating relationship in weekly and daily data. So, for weekly and daily data, the VAR method using the VECM trace test does not find cointegration at 95% level as compared to the state-space method.

Table 5.15: Results of the VECM Cointegration Test and State-Space Cointegration Test: Weekly Data

| Rank ($r$) | VECM Trace Statistic | State-Space Trace Statistic | Critical Value (own soft) | Critical Value (Matlab soft) | $p$-value |
|---|---|---|---|---|---|
| $r = 0$ | 81.4376 | 92.7863 | 83.3133 | 83.9402 | 0.0749 |
| $r \leq 1$ | 49.0909 | 57.9323 | 59.6193 | 60.0623 | 0.3214 |
| $r \leq 2$ | 20.6354 | 22.0739 | 39.9665 | 40.1751 | 0.8726 |
| $r \leq 3$ | 10.1878 | 10.1561 | 24.1994 | 24.2747 | 0.8412 |
| $r \leq 4$ | 4.8096 | 7.0035 | 12.2744 | 12.3206 | 0.6315 |
| $r \leq 5$ | 0.3031 | 1.0967 | 4.0969 | 4.1302 | 0.7302 |

Table 5.16: Results of the VECM Cointegration Test and State-Space Cointegration Test: Daily Data

| Rank ($r$) | VECM Trace Statistic | State-Space Trace Statistic | Critical Value (own soft) | Critical Value (Matlab soft) | $p$-value |
|---|---|---|---|---|---|
| $r = 0$ | 76.6185 | 89.8564 | 83.3133 | 83.9402 | 0.1509 |
| $r \leq 1$ | 43.8313 | 52.8313 | 59.6193 | 60.0623 | 0.0537 |
| $r \leq 2$ | 18.6842 | 20.0098 | 39.9665 | 40.1751 | 0.9362 |
| $r \leq 3$ | 9.3888 | 9.0544 | 24.1994 | 24.2747 | 0.8892 |
| $r \leq 4$ | 4.4033 | 5.6009 | 12.2744 | 12.3206 | 0.6789 |
| $r \leq 5$ | 0.7159 | 0.9001 | 4.0969 | 4.1302 | 0.5917 |

Table 5.17: Estimated Beta Vector

| Company | monthly | | | | weekly | | daily | |
|---|---|---|---|---|---|---|---|---|
| | VAR | | State space | | VAR | State space | VAR | State space |
| USA | −0.0054 | 0.0087 | −0.0055 | 0.0084 | −0.0052 | −0.0053 | −0.0050 | −0.0051 |
| UK | −0.0040 | 0.0022 | −0.0042 | 0.0027 | −0.0043 | −0.0045 | −0.0047 | −0.0049 |
| Europe | −0.0005 | 0.0001 | −0.0007 | 0.0001 | −0.0001 | −0.0003 | −0.0000 | −0.0001 |
| Pacific | −0.0010 | 0.0031 | −0.0009 | 0.0029 | −0.0009 | −0.0007 | −0.0011 | −0.0009 |
| German | 0.0114 | −0.0128 | 0.0111 | −0.0126 | 0.0115 | 0.0114 | 0.0113 | 0.0112 |
| Japan | 0.0020 | −0.0014 | 0.0023 | −0.0017 | 0.0015 | 0.0011 | 0.0020 | 0.0021 |

As can be seen from Table 5.15 and 5.16, in the context of the state-space model, the cointegrating vector in weekly and daily data is found at 95% significance level. However, in the VAR model the cointegrating vector in weekly and daily data is found at 80% significance level, not at 95%. Table 5.17 seems to suggest that in cointegrating relations the German index and Japan stock price index are approximately a weighted average of the USA and UK stock price index.

### 5.3.2 VECM & SSECM Maximum Likelihood

A VECM(1) model was used for monthly data set and VECM(2) for each of the two other frequencies considered (daily, weekly) as this turned out to be the best choice. The research study applied the same cointegration rank on VECM and SSECM for fair comparison of likelihood and BIC. See Table 5.18 for results using monthly data. In the case of the BIC, this criterion suggests that VECM model minimizes the information criterion. In fact, the BIC results are not very different between VECM and SSECM and the same holds for the Log likelihood results. Table 5.12 and Table 5.13 show the results of the weekly and the daily series, here SSECM model minimizes the information criterion in both the data sets.

Table 5.18: Results of VECM with Lags $k = 1$ and Effective Sample Size $N = 572$ & SSECM with $n = 5$ and Effective Sample Size $N = 575$: Monthly Data.

| Model | Log Likelihood | BIC | Total Parameters |
|---|---|---|---|
| VECM | 2374.46 | 4425.58 | 51 |
| SSECM | 2383.79 | 4441.58 | 42 |

Table 5.19: Results of VECM with Lags $k = 2$ and Effective Sample Size $N = 2489$ & SSECM with $n = 5$ and Effective Sample Size $N = 2492$: Weekly Data.

| Model | Log Likelihood | BIC | Total Parameters |
|---|---|---|---|
| VECM | 11954.48 | 24580.44 | 86 |
| SSECM | 11961.37 | 24251.18 | 42 |

Table 5.20: Results of VECM with Lags $k = 2$ and Effective Sample Size $N = 11933$ & SSECM with $n = 5$ and Effective Sample Size $N = 11936$: Daily Data.

| Model | Log Likelihood | BIC | Total Parameters |
|---|---|---|---|
| VECM | 22836.82 | 45753.32 | 86 |
| SSECM | 22841.73 | 45289.51 | 42 |

# Conclusions and Further Research

The research approach used in this study allows for cointegration with a novel parametrization to the problem of maximum likelihood estimation of a state-space model in error correction form.This does not hold true for the continuous-discrete model for which we do not have an error correction form. We proposed a parameterization and estimation model that translates the cointegration property into a low-rank constraint on a resulting likelihood optimization. The concept of cointegration is central to the analysis of financial and economic time series, which are usually non-stationary. In particular, the VECM approach by Johansen (1988, 1991) falls within the broader category of vector autoregressive (VAR) models that are well-known to be prone to overparameterization (the number of parameters grow with the number of lags and variables). Besides, the number of parameters are limited in terms of data generating process representation. In comparison, state-space models are a more flexible model class, which, in this case, allows for a comprehensive study of cointegration among time series, particularly within a continuous-time framework.

The research applied the estimation procedure to discrete-time simulated data and compared the estimation results using the VAR model. The main findings resulting from this simulation are as follow:

- First, the log likelihood estimates are similar between both model classes;
- Second, VECM has a higher number of parameters compared to SSECM[1];
- Third, the BIC information criteria are much lower for the SSECM, which favors such model class regarding model selection;
- Fourth, SSECM outperforms VECM in other model selection criteria including goodness of fit, forecast accuracy (RMSE) and Bayes factor.
- Fifth, there is a clear difference between the delay profile between the VECM and the SSECM. In the delay profile for SSECM, the quantities go to zero rather smoothly compared to VECM.

[1]VAR-based models are well known to be prone to overparameterization and the curse of dimensionality.

Moreover, Ribarits and Hanzon (2014b,a) outlined several reasons favouring the state-space models over vector-autoregressive methodologies regarding statistical parsimony and flexibility for different applications.

Furthermore, the study compared an actual model with SSECM results. The difference between $\theta$ used in the true model and the one used in the maximum likelihood estimation for the state-space model is small. The estimates deviate from the values in the true model when the number of observations equals 100 and the estimates are not close to the true $\theta$; however, when the same value for $\theta$ is used with 2000 observations, it yields better results. In fact, the study conducted a state-space estimation with order $n = 3$ and $n = 4$, using the same set. The results yield the sample covariance matrix of $\varepsilon_t$. The results were very close when taking the negative log likelihood, as the result of $n = 3$ was slightly larger than of $n = 4$.

The estimation procedure is applied to some real data sets as well, and compared to estimation results using the VAR model. In some of the cases the state-space model showed a better performance. It is expected that for high frequency data in which the DGP has an intrinsic delay the state-space model will do better than VAR model as it will be more parsimonious in such cases. The study applied the method to three datasets: oil prices, stock indices of six companies and stock indices of six countries.

- For the first application, it studied the time series using daily data from 1992 to 2012 for three types of crude oil: West Texas Intermediate, Brent and Dubai. Although both model classes yield similar results, the SSECM had a lower information criterion. Also, the SSECM model yielded lower values for half of the parameters when compared against the VECM. Specifically, the latter had 41 parameters, while the former had only 26. This makes the SSECM not only more parsimonious but also less uncertain as the estimation involves less parameter to be estimated. In the cointegration relationship, the Brent crude is approximately a weighted average of the other two crude types, and the same applies to the West Texas Intermediate.
- For the second application, daily, weekly and monthly data was considered for six equity prices (Apple, Coca-Cola, Sony, Microsoft, IBM and JP Morgan) over the period 1990–2016. The results indicated that there is one cointegrating relationship at monthly and weekly frequency, and two cointegration relationships at daily frequency. In the cointegrating vector, the stock prices of IBM and JP Morgan are a weighted average of the stock prices of Microsoft and Coca cola. When the integration order is different in time series, the correla-

tion metric and even the basic regression outputs are spurious Furthermore, the VECM model has lower information criterion value than the SSECM in this case.

- In a third application,the study analyzed stock market indices for six geographical areas (USA, UK, Europe, Germany, the Pacific and Japan) at monthly, weekly and daily frequencies over the period 1970–2017. Although at monthly frequency the SSECM and VECM coincided regarding the number of cointegrating relations, there is a discrepancy at the remaining frequencies. For weekly and daily data, there is no cointegration with the VECM but only in the SSECM, which suggests that the latter is effective for high frequency data. Overall, the indices for Germany and Japan are approximately a weighted average of the indices for USA and the UK in the cointegrating relationship. Information criteria and log likelihoods are similar across both model classes but the number of parameters in the VECM were higher than in the SSECM in all cases.

Summarizing, the study found many cases in which the state-space model could be preferred over the VAR model:

- For example, BIC is better for SSECM in several cases (mainly due to the fact that SSECM uses fewer parameters).
- Other outcome, if both models find the same cointegration vectors they are typically more or less the same for VECM as for SSECM. Indeed, for given significance level VECM does not always find the cointegration vectors that are found by the SSECM.
- Another result, is when p is large ($n < p$). If p is too big the VECM can no longer be estimated while SSECM can, as long as the order n is not too big.

The study findings significantly contribute to future research on this topic. The study has presented an approach to modelling time series which exhibit cointegration and have irregular observation times. The approach uses a continuous time state evolution model. In order to find the maximum likelihood estimator of the model an approximation has been used that assumes small time step lengths. The approximation allows researchers to reduce the likelihood optimization problem to an optimization problem over asymptotically stable reachable pairs. For this, a gradient-type search algorithm is used, based on a parametrization of input-normal pairs which is known to be numerically stable. The study applied a continuous-time state-space model with cointegration to simulated data, and compared the result against a true model, where the the SSECM cointegration vector and the $\theta$ parameter has similar results[2].

Further research could be carried out,

(i) To apply a continuous time state-space model with the cointegration to real data.

(ii) To study the more general case of missing data.

(iii) To improve the approximation in case the time steps are not sufficiently small.

(iv) To combine this with investigations into trading (waiting) times distributions, see (Wasserman 2000, Mainardi et al. 2000a,b, Sabatelli et al. 2002).

(v) To investigate the case in which $A$ has an eigenvalue at zero in the predictor model.

(vi) Using Quasi Monte Carlo to our approach to generate starting points.

(vii) Find alternative gradient-type search methods like for instance the Gauss-Newton algorithm.

In conclusion, VARs (both in reduced and structural form) have become standard tools for macroeconomic forecasting and policy analysis. The model is supported by for its practicality by academic and policy economists worldwide. The SSECM is a novel approach in the arena of time series econometrics. SSECM effectiveness over VARs could be summarised as follows:

- State-space models represent a larger and more flexible model class, where VARs are integrated as special case within a broader scope.
- If the researcher is not able to observe all the output components of DGP in a VAR process, then it is generally impossible to describe such a process via a VAR model, while it can still describe by a linear state-space model.
- If the DGP is a state-space process and not itself a VAR process, then the order estimate for the VAR approximation will have infinite value when the sample size tends to infinity. Also, the study used the information criterion for estimating the parameters as there were a large number of parameters.
- The state-space approach also allows for a continuous-discrete formulation of the model which is investigated in this thesis. In such a model the time points

[2] We do not present a result for the VECM, as this framework has not been extended to continuous-time which in turn gives strong advantage to the state space framework.

at which observations are taken do not have to be equally spaced which can be relevant for high frequency financial data.

Substantially, the flexibility of state-space model makes it possible to treat such cases and further research along these lines is expected. Also, the study would like to mention two situations in which the VAR method might not work:

- If the frequency of the observations is high, while there is an intrinsic delay in the system, one would expect the lag lengths in the VAR model to be large, requiring a large number of parameters.
- If $p$ is large, the number of VAR-parameters will be large as the number of VAR parameters grows quadratically with $p$. In such cases the state space model would still produce estimates as long as the order $n$ is not chosen too large.

This study has outlined the advantages of the SSECM. Based on these advantages, one has to acknowledge that VAR framework has become the dominant empirical workhorse because of its balance between explanatory power and simplicity: VARs (including VECs) are built upon regression analysis, which in turn has been proven to be computationally unproblematic and statistically reliable. In contrast, state-space modelling is complicated and computationally intensive, given that it involves numerical optimisation problems over high-dimensional parameter spaces; thus, it could profit from further research. Also, VARs can be easily implemented in a wide range of software tools (including pre-coded routines), which is accompanied by a lot of documentation; SSECMs in turn represents promising work rather than an established method so far.

# References


Aoki, M. and Havenner, A.: 1991, State-space modeling of multiple time series, *Econometric Reviews* **10**(1), 1–59.

Athanasopoulos, G. and Vahid, F.: 2008, Varma versus var for macroeconomic forecasting, *Journal of Business & Economic Statistics* **26**(2), 237–252.

Atukeren, E.: 2005, Measuring the strength of cointegration and granger-causality, *Applied Economics* **37**(14), 1607–1614.

Bauer, D. and Wagner, M.: 2002, Estimating cointegrated systems using subspace algorithms, *Journal of Econometrics* **111**(1), 47–84.

Berger, J. O. and Pericchi, L. R.: 1996, On the justification of default and intrinsic bayes factors, *Modelling and Prediction Honoring Seymour Geisser*, Springer, pp. 276–293.

Bierens, H. J.: 1997, Nonparametric cointegration analysis, *Journal of Econometrics* **77**(2), 379–404.

Bos, T. and Newbold, P.: 1984, An empirical investigation of the possibility of stochastic systematic risk in the market model, *Journal of Business* pp. 35–41.

Boswijk, H. P., Franses, P. H. and van Dijk, D.: 2010, Cointegration in a historical perspective, *Journal of Econometrics* **158**(1), 156–159.

Box, G. E. and Jenkins, G. M.: 1970, *Time Series Analysis, Forecasting, and Control*, Holden Day: San Francisco.

Breitung, J.: 2002, Nonparametric tests for unit roots and cointegration, *Journal of econometrics* **108**(2), 343–363.

Breusch, T. S.: 1978, Testing for auto-correlation in dynamic linear models, *Australian Economic Papers* **17**(31), 334–355.

Chow, G. C.: 1960, Tests of equality between sets of coefficients in two linear regressions, *Econometrica: Journal of the Econometric Society* **28**(3), 591–605.

Collins, D. W., Ledolter, J. and Rayburn, J. D.: 1987, Some further evidence on the stochastic properties of systematic risk, *The Journal of Business* **60**(3), 425–48.

Cooley, T. F. and Dwyer, M.: 1998, Business cycle analysis without much theory a look at structural vars, *Journal of econometrics* **83**(1-2), 57–88.

Dickey, D. A. and Fuller, W. A.: 1979, Distribution of the estimators for autoregressive time series with a unit root, *Journal of the American statistical association* **74**(366a), 427–431.

Doan, T., Litterman, R. B., Sims, C. A. et al.: 1986, Forecasting and conditional projection using realistic prior distribution, *Technical report*, Staff Report from the Federal Reserve Bank of Minneapolis.

Douc, R., Moulines, E. and Stoffer, D.: 2014, *Nonlinear time series: Theory, Methods and Applications with R examples*, Boca Raton: CRC press.

Durbin, J. and Koopman, S. J.: 2001, Time series analysis by state-space methods, *Technical report*, Oxford University Press.

Engle, R. F. and Granger, C. W.: 1987, Co-integration and error correction: representation, estimation, and testing, *Econometrica: Journal of the Econometric Society* pp. 251–276.

Engle, R. and Granger, C.: 1991, Long-run economic relationships: Readings in cointegration, *in* R. F. Engle and C. W. J. Granger (eds), *Advanced Texts in Econometrics*, Oxford University Press.

Friedman, B. M. and Kuttner, K. N.: 1992, Money, income, prices, and interest rates, *The American Economic Review* pp. 472–492.

Gilbert, P. D.: 1993, Estimation and reduction of state space models: An improvement on var methodology.

Godfrey, L. G.: 1978, Testing against general autoregressive and moving average error models when the regressors include lagged dependent variables, *Econometrica: Journal of the Econometric Society* **46**(6), 1293–1301.

Goodwin, G. C. and Payne, R. L.: 1977, Dynamic system identification: experiment design and data analysis, *Mathematics in Science and Engineering* **136**.

Granger, C. and Newbold, P.: 1974, Spurious regressions in econometrics, *Journal of Econometrics* **2**(2), 111–120.

Granger, C. W.: 1969, Investigating causal relations by econometric models and cross-spectral methods, *Econometrica: Journal of the Econometric Society* pp. 424–438.

Granger, C. W.: 1981, Some properties of time series data and their use in econometric model specification, *Journal of Econometrics* **16**(1), 121–130.

Granger, C. W. and Weiss, A. A.: 1983, Time series analysis of error-correction models, *Studies in Econometrics, Time Series, and Multivariate Statistics*, Elsevier, pp. 255–278.

Greene, W. H.: 2003, *Econometric Analysis*, Prentice-Hall.

Hannan, E. J. and Deistler, M.: 1988, *The statistical theory of linear systems*, John Wiley, New York.

Hanzon, B.: 1993, On the closure of several sets of arma and linear state space models with a given structure, *in* Brillinger, Caines, Geweke, Parzen, Rosenblatt and Taqqu (eds), *New Directions in Time Series Analysis*, Springer, pp. 239–253.

Hanzon, B., Olivi, M. and Peeters, R.: 2009, Subdiagonal pivot structures and associated canonical forms under state isometries, *IFAC Proceedings Volumes* **42**(10), 1620–1625.

Hanzon, B., Olivi, M. and Peeters, R. L.: 2006, Balanced realizations of discrete-time stable all-pass systems and the tangential schur algorithm, *Linear Algebra and its Applications* **418**(2-3), 793–820.

Hanzon, B., Peeters, R. L. and Olivi, M.: 2007, Canonical lossless state-space systems: Staircase forms and the schur algorithm, *Linear Algebra and its Applications* **425**(2-3), 404–433.

Hanzon, B. and Ribarits, T.: 2008, Co-integration in continuous and discrete time–the state-space error-correction model, *5th World Congress of the Bachelier Finance Society. London, UK*.

Harrison, P. J. and Stevens, C. F.: 1976, Bayesian forecasting, *Journal of the Royal Statistical Society: Series B (Methodological)* **38**(3), 205–228.

Hartl, T. and Weigand, R.: 2018, Approximate state-space modelling of unobserved fractional components, *arXiv preprint arXiv:1812.09142* .

Harvey, A. C.: 1989, *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press.

Harvey, A. C. and Todd, P.: 1983, Forecasting economic time series with structural and box-jenkins models (with discussion), *Journal of Business & Economic Statistics* **1**(4), 299–315.

Heij, C., Schumacher, J., Hanzon, B. and Praagman, C.: 1997, System dynamics in economic and financial models.

Jarque, C. M. and Bera, A. K.: 1987, A test for normality of observations and regression residuals, *International Statistical Review/Revue Internationale de Statistique* **55**(2), 163–172.

Johansen, S.: 1988, Statistical analysis of cointegration vectors, *Journal of Economic Dynamics and Control* **12**(2-3), 231–254.

Johansen, S.: 1991, Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models, *Econometrica: Journal of the Econometric Society* **59**(6), 1551–1580.

Johansen, S.: 1995, *Likelihood-based inference in cointegrated vector autoregressive models*, Oxford University Press on Demand.

Jones, R. H.: 1984, Maximum likelihood fitting of arma models to time series with missing observations, *Technometrics* **22**(3), 389–395.

Kalman, R. and Bucy, R.: 1961, New results in linear filtering and prediction theory, *J. Fluids Eng* **83**(1), 95–108.

Kalman, R. E. et al.: 1960, A new approach to linear filtering and prediction problems, *Transactions of the ASME – Journal of Basic Engineering* **82**(1), 35–45.

Kascha, C. and Mertens, K.: 2009, Business cycle analysis and varma models, *Journal of Economic Dynamics and Control* **33**(2), 267–282.
**URL:** *https://www.sciencedirect.com/science/article/pii/S0165188908001012*

Kascha, C. and Trenkler, C.: 2011, Cointegrated varma models and forecasting us interest rates, *Available at SSRN 1957103* .

Kass, R. E. and Raftery, A. E.: 1995, Bayes factors, *Journal of the American Statistical Association* **90**(430), 773–795.

Kilian, L.: 2013, Structural vector autoregressions, *in* N. Hashimzade and M. Thornton (eds), *Handbook of Research Methods and Applications in Empirical Macroeconomics*, Edward Elgar Publishing, pp. 515–554.

Kilian, L. and Lütkepohl, H.: 2017, *Structural vector autoregressive analysis*, Cambridge University Press.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., Shin, Y. et al.: 1992, Testing the null hypothesis of stationarity against the alternative of a unit root, *Journal of Econometrics* **54**(1-3), 159–178.

Leusner, J., Akhavein, J. D. and Swamy, P. A.: 1997, Solving an empirical puzzle in the capital asset pricing model, *Washington, D.C.: Office of the Comptroller of the Currency* .

Lippi, M. and Reichlin, L.: 1994, Var analysis, non-fundamental representations, blaschke matrices, *Journal of Econometrics* **63**(1), 307–325.

Litterman, R.: 1984, Specifying var's for macroeconomic forecasting, *Federal Reserve Bank of Minneapolis Staff Report* **92**.

Ljung, L.: 1987, System identification: theory for the user, *PTR Prentice Hall, Englewood Cliffs, New Jerse* .

Lütkepohl, H.: 2005, *New Introduction to Multiple Time Series Analysis*, 1st ed. 2005. corr. 2nd printing 2007 edn, Springer-Verlag Berlin.

Lütkepohl, H.: 2013, Vector autoregressive models, *in* N. Hashimzade and M. Thornton (eds), *Handbook of Research Methods and Applications in Empirical Macroeconomics*, Edward Elgar Publishing Limited, pp. 139–164.

Lütkepohl, H. and Claessen, H.: 1997, Analysis of cointegrated varma processes, *Journal of Econometrics* **80**(2), 223–239.

Lütkepohl, H. and Krätzig, M.: 2004, *Applied Time Series Econometrics*, Cambridge University Press.

Mainardi, F., Raberto, M., Gorenflo, R. and Scalas, E.: 2000a, Fractional calculus and continuous-time finance, *Physica A: Statistical Mechanics and its Applications* **284**(14), 376–384.

Mainardi, F., Raberto, M., Gorenflo, R. and Scalas, E.: 2000b, Fractional calculus and continuous-time finance ii: the waiting-time distribution, *Physica A: Statistical Mechanics and its Applications* **287**(3-4), 468–481.

Mainardi, F., Raberto, M., Gorenflo, R. and Scalas, E.: 2000c, Fractional calculus and continuous-time finance ii: the waiting-time distribution, *Physica A: Statistical Mechanics and its Applications* **284**(14), 376–384.

Mainardi, F., Raberto, M., Gorenflo, R. and Scalas, E.: 2000d, Fractional calculus and continuous-time finance ii: the waiting-time distribution, *Physica A: Statistical Mechanics and its Applications* **287**(3-4), 468–481.

Nelson, C. R. and Plosser, C. R.: 1982, Trends and random walks in macroeconmic time series: some evidence and implications, *Journal of Monetary Economics* **10**(2), 139–162.

Park, J. Y. and Phillips, P. C.: 1988, Statistical inference in regressions with integrated processes: Part 1, *Econometric Theory* **4**(3), 468–497.

Park, J. Y. and Phillips, P. C.: 1989, Statistical inference in regressions with integrated processes: Part 2, *Econometric Theory* **5**(1), 95–131.

Pernebo, L. and Silverman, L.: 1982, Model reduction via balanced state space representations, *IEEE Transactions on Automatic Control* **27**(2), 382–387.

Phillips, P. C. and Hansen, B. E.: 1990, Statistical inference in instrumental variables regression with i (1) processes, *The Review of Economic Studies* **57**(1), 99–125.

Phillips, P. C. et al.: 1986, Understanding spurious regressions in econometrics, *Journal of Econometrics* **33**(3), 311–340.

Phillips, R. F.: 1991, A constrained maximum-likelihood approach to estimating switching regressions, *Journal of Econometrics* **48**(1-2), 241–262.

Poskitt: 2012, Strongly consistent determination of cointegrating rank via canonical correlations, *Journal of Business and Economics Statistics* **18**, 77–90.

Poskitt, D. S.: 1994, A note on autoregressive modeling, *Econometric Theory* **10**(5), 884–899.

Poskitt, D. S.: 2000, Strongly consistent determination of cointegrating rank via canonical correlations, *Journal of Business & Economic Statistics* **18**(1), 77–90.

Poskitt, D. S.: 2006, On the identification and estimation of non-stationary and cointegrated armax systems, *Econometric Theory* pp. 1138–1175.

Poskitt, D. and Tremayne, A.: 1987, Determining a portfolio of linear time series models, *Biometrika* **74**(1), 125–137.

Reinsel, G. C. and Ahn, S. K.: 1992, Vector autoregressive models with unit roots and reduced rank structure: Estimation. likelihood ratio test, and forecasting, *Journal of time series analysis* **13**(4), 353–375.

Ribarits, T. and Hanzon, B.: 2006, On a new approach to cointegration–the state-space error correction model, *62nd European Meeting of the Econometric Society, 27, Budapest*, Mimeo. 31 August 2007.

Ribarits, T. and Hanzon, B.: 2014a, The state-space error correction model: Definition, estimation and model selection, *Estimation and Model Selection (October 29, 2014)* .
**URL:** *https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2516529*

Ribarits, T. and Hanzon, B.: 2014b, The state-space error correction model: Simulations and applications, *Available at SSRN 2516560* .
**URL:** *https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2516560*

Sabatelli, L., Keating, S., Dudley, J. and Richmond, P.: 2002, Waiting-time distributions in financial markets, *The European Physical Journal B-Condensed Matter and Complex Systems* **27**(2), 273–275.

Saikkonen, P.: 1991, Asymptotically efficient estimation of cointegration regressions, *Econometric theory* **7**(1), 1–21.

Shumway, R. H.: 1988, Applied statistical time series analysis., *Englewood Cliffs, NJ: Prentice-Hall Series in Statistics.* .

Shumway, R. H. and Stoffer, D. S.: 1982, An approach to time series smoothing and forecasting using the em algorithm, *Journal of time series analysis* **3**(4), 253–264.

Sims, C. A.: 1980, Macroeconomics and reality, *Econometrica: Journal of the Econometric Society* **48**(1), 1–48.

Sims, C. A., Stock, J. H. and Watson, M. W.: 1990, Inference in linear time series models with some unit roots, *Econometrica* **58**, 113–144.

Stock, J. H. and Watson, M. W.: 1988, Testing for common trends, *Journal of the American statistical Association* **83**(404), 1097–1107.

Tiao, G. C. and Tsay, R. S.: 1989, Model specification in multivariate time series, *Journal of the Royal Statistical Society: Series B (Methodological)* **51**(2), 157–195.
**URL:** *https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1989.tb01756.x*

Tschernig, R., Weber, E. and Weigand, R.: 2013, Long-run identification in a fractionally integrated system, *Journal of Business & Economic Statistics* **31**(4), 438–450.

Uschmajew, A. and Vandereycken, B.: 2020, Geometric methods on low-rank matrix and tensor manifolds, *Handbook of variational methods for nonlinear geometric data* pp. 261–313.

Wagenmakers, E.-J.: 2007, A practical solution to the pervasive problems of p-values, *Psychonomic Bulletin & Review* **14**(5), 779–804.

Wainwright, K. and Chiang, A.: 1967, Fundamental methods of mathematical economics, 3rd edn.(1984).

Wasserman, L.: 2000, Asymptotic inference for mixture models by using data-dependent priors, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**(1), 159–180.

Wooldridge, J. M.: 2012, Introductory econometrics: A modern approach, (boston: Cengage learning), pp. 1–16.

Yap, S. F. and Reinsel, G. C.: 1995, Estimation and testing for unit roots in a partially non-stationary vector autoregressive moving average model, *Journal of the American Statistical Association* **90**(429), 253–267.

Yule, G. U.: 1926, Why do we sometimes get nonsense-correlations between time-series?–a study in sampling and the nature of time-series, *Journal of the Royal Statistical Society* **89**(1), 1–63.

# Appendices

# Appendix A

# Asymptotic Distribution of the Cointegration Rank Test Statistics (Trace Test)

```
function [alpha] = johansen_perc(dim,r,p)
assert((p>=0) && (p<= 100), 'Alpha is not a valid probability number.')
N_mc = 1000000; % Number of Monte Carlo samples
samples = johansen(dim, r, N_mc);
alpha = prctile(samples, p);
end
function [alphas] = johansen_validation
%assert((p>=0) && (p<= 100), 'Alpha is not a valid probability number.')
probs = [2.5, 5, 10, 50, 90, 95, 97.5];
alphas = zeros(5,7);
N_mc = 100000; % Number of Monte Carlo samples
fig = figure(1);
set(fig, 'position', [0, 0, 500, 3000])
for dim=2:6
    samples = johansen(dim, 1, N_mc);
    for i=1:7
    alphas(dim-1,i) = prctile(samples, probs(i));
    end
    subplot(5,1,dim-1);
    hist(samples, 1000);
    title('BM dim = '+string(dim-1));
end
%saveas(fig, 'histograms_dim_1to5.png');
end

function [sam] = johansen(dim, r, N)
assert((dim>= r) && (r>=0), 'Rank r can be at most equal to the number of
    time-series dim.')
h = 0.001;
M = 1000;
sam = zeros(N,1);
```

```
for i=1:N
    dW = sqrt(h) * randn(dim-r,M-1);
    W = zeros(dim-r, M);
    W(:,2:M) = cumsum(dW, 2);
    D0 = h * (W * W');
    D1 = W(:,1:M-1) * dW';
    sam(i,1) = trace(D1' * pinv(D0) * D1);
    %sam(i,1)
end
end
```

# Appendix B

# Generation of Cointegrated Models and Corresponding Time Series

data_gen

```
function [th,TH,beta2, yseries, dyseries] = data_gen (n_data)

n = 2;
p = 6;
%p>n so will appear beta2
 k=p-n; % alpha2 and beta2 belong to R^(p*k)
 r=5; %I(1) rank of (alpha*beta')
 %c=p-k
 c=2; % c=n so alpha1 and beta1 not appear

%{
This block uses randomization to select matrices bar_A, bar_B, and bar_C
so that the state space model with A = bar_A - bar_B * bar_C is stable. Based
on preliminary testing, stability is not guaranteed and so checking this
condition is necessary.

We allow the block to loop at each of the three main randomization steps:
-The selection of THETA values, which ultimately determine beta2
   -The selection of theta values, which ultimately determine bar_A and bar_B;
   -The selection of the vectors alpha and beta, which contribute to bar_C;
   -The selection of the matix/vector bar_C_1, which also contributes to bar_C.
Looping is limited to 10 iterations for the second and third steps, in case
previous steps happen to give values that aren't favorable to obtaining a stable
system.
%}

eig_tol = 0.000001;
```

```
found_stable = false;
while (~found_stable)  % loop for theta values
  % the th(j, k) effectively replace x(k), y(k), and z(k) from the
  % previous script, at least within the data generating process; they are
  % selected randomly
  th = genth_tol(); %The selection of theta values, which ultimately determine bar_A and bar_B
  TH = genth_tol_beta2();%The selection of THETA values, which ultimately determine beta2
  % construct the V_k's and then the realization matrix R and RB
  VB = getVB(TH); %for RB n x n
  V = getV(th); %for R p+1 x P+1
  %find orthogonal matrix RB to select beta2
  RB = eye(p);
  for i = 1:k
    RB =blkdiag(eye((k)-i), VB(:, :, i), eye(i-1)) * RB;
  end

  %find orthogonal matrix R to select bar_A and bar_B
  R = eye(n+p);
  for i = 1:n
    R = blkdiag(eye(n-i), V(:, :, i), eye(i-1)) * R;
  end
R;
RB;
   % select beta2 from RB
   beta2prime=RB(n+1:p,1:p);
   beta2=beta2prime';
  % select out bar_A and bar_B from R
  bar_A = R(p+1:n+p, p+1:n+p);
  bar_B = R(p+1:n+p, 1:p);
  % find the matrix R_tel
  Rtel= vertcat((eye(n)-bar_A)^(-1)*bar_B,beta2prime);
  counter1 = 1;
  while (~found_stable && counter1<=10)  % loop for alpha and beta
    % randomly select alpha and beta and prepare to generate bar_C
%      alpha1 = randn(p,1);
%      beta1 = randn(p,1);
%alpha1 and beta1 not appear cos c=n
Cbaralpha2=-eye(p)*Rtel^(-1);
alpha2=Cbaralpha2(1:p,n+1:p);
alphabetaprime = alpha2 * beta2prime;

    counter2 = 1;
    while (~found_stable && counter2<=10)  % loop for bar_C_1
      % randomly select bar_C_1, then determine bar_C_2 and bar_C

      bar_C  =Cbaralpha2(1:p,1:n);
      eigvalA = eig(bar_A - bar_B * bar_C)
      % the eigenvalue size check is for the condition that all eigenvalues
      % of A = bar_A - bar_B * bar_C are less than or equal to 1 in absolute
      % value, allowing a small tolerance for machine rounding error
      eig_size_check = (max(abs(eigvalA)) < 1 + eig_tol);
      % the eigenvalue phase check is for the condition that the only
```

```
      % eigenvalue of A having absolute value 1 is 1 itself, again allowing a
      % small tolerance for machine rounding error
      phaseok = @(l) ((abs(l) < 1 - eig_tol) || (real(l) > 1 - eig_tol));
      eig_phase_check = all(arrayfun(phaseok, eigvalA));
      % if both checks pass, we've found a stable system
      if (eig_size_check && eig_phase_check)
        found_stable = 1;
      end
      counter2 = counter2 + 1;
    end
    counter1 = counter1 + 1;
  end
end

%{
The next section generates data for the state-space error correction model.

This begins with a 'burn-in' period, which can be adjusted with the n_burn
variable. The 'burn-in' data are not recorded for later use.

After the 'burn-in' period the data is recorded. The amount of data that gets
recorded can be adjusted with the n_data variable.

The white noise process ep has independent components that are drawn from the
standard normal distribution. This means that the variance-covariance matrix
Sigma for ep is simply the identity matrix. We could obtain other Sigmas by
multiplying the current ep by any fixed p-by-p matrix.
%}

n_burn = 2;
 n_data = 2000;
% initialize and generate the burn-in data
xt = zeros(n, 1);
dy = zeros(p, 1);
y = zeros(p, 1);
for i = 1:n_burn
  new_y = y + dy;
  new_xt = bar_A * xt + bar_B * dy;
  ep = randn(p, 1);
  new_dy = alphabetaprime * new_y + bar_C * bar_A * ((eye(n) - bar_A) \ new_xt) + ep;
  xt = new_xt;
  dy = new_dy;
  y = new_y;
end
% initialize and generate the data to be recorded; note that
%  * xtseries(:, i) = xt_(i)
%  * dyseries(:, i) = dy_(i)
%  * yseries(:, i)  = y_(i-1)
% in order to align subscripts with how they are used in the Z_j's below
xtseries = zeros(n, n_data);
```

```
dyseries = zeros(p, n_data);
yseries = zeros(p, n_data);
xtseries(:, 1) = xt;
dyseries(:, 1) = dy;
yseries(:, 1) = y;
for i = 2:n_data
  yseries(:, i) = yseries(:, i-1) + dyseries(:, i-1);
  xtseries(:, i) = bar_A * xtseries(:, i-1) + bar_B * dyseries(:, i-1);
  ep = randn(p, 1);
  dyseries(:, i) = alphabetaprime * yseries(:, i) + bar_C * bar_A * ((eye(n)
      - bar_A) \ xtseries(:, i)) + ep;
end

end
```

## GenerateExcelFiles

```
clear all;
close all;
clc

[trueth,THRUETH,beta2, data , dyseries] = data_gen(2000); % the length of
    data is p=3 by 2000 samples
size(trueth);
size(data);
size(THRUETH);
size(beta2);
save DATA_FILE data trueth
csvwrite('gener_data_2000.csv',data')
csvwrite('gener_theta.csv',trueth)
csvwrite('gener_thetaBeta2.csv',THRUETH)
csvwrite('gener_Beta2.csv',beta2)
```

## enth_tol

```
% Function to generate a random matrix of theta values, essentially from the
% range -pi/2 to pi/2; the output matrix is 3x4
function thetas = genth_tol()
  n = 2;
  p = 6;
  thetas = zeros(p, n);
  thetas(1, :) = unifrnd(-pi/2+0.001, pi/2-0.001, 1, n);
  thetas(2:p, :) = unifrnd(-pi/2, pi/2, p-1, n);
  thetas=0.6*thetas;
end
```

## genth_tol_beta2

```
% Function to generate a random matrix of theta values, essentially from the
% range -pi/2 to pi/2; the output matrix is 2x1 (to generated beta2
function Thetas = genth_tol_beta2()
  n = 2;
```

```
  p = 6;
  k=p-n;
  Thetas = zeros(n, k);
  Thetas(1, :) = unifrnd(-pi/2+0.001, pi/2-0.001, 1, k);
  Thetas(2:n, :) = unifrnd(-pi/2, pi/2, n-1, k);
  Thetas=0.6*Thetas;
end
```

## getV

```
% DESCRIPTION: Function to construct the matrices V_k (k = 1 to 6) from a
% 3x4 matrix containing theta coordinate values
%
% Input = thmtx  (6x6 matrix of theta coordinates)
%
% Output = V     (6x6x.. multidimensional array that holds V_1 through V_6,
%                 with V_k corresponding to V(:, :, k) )

function V = getV(thmtx)
  n = 2;
  p = 6;
  for k = 1:n
    th1k = thmtx(1, k);  % abbreviations to reduce the number of parentheses
    th2k = thmtx(2, k);
    th3k = thmtx(3, k);
    th4k = thmtx(4, k);
    th5k = thmtx(5, k);
    th6k = thmtx(6, k);


v(:,:,k)=[sin(th1k)*sin(th2k)*sin(th3k)*sin(th4k)*sin(th5k)*sin(th6k);sin(
    th1k)*sin(th2k)*sin(th3k)*sin(th4k)*sin(th5k)*cos(th6k);sin(th1k)*sin(
    th2k)*sin(th3k)*sin(th4k)*cos(th5k);sin(th1k)*sin(th2k)*sin(th3k)*cos(
    th4k);sin(th1k)*sin(th2k)*cos(th3k);sin(th1k)*cos(th2k)];
u(:,:,k)= cos(th1k);
vu(:,:,k)=eye(p)-(1-u(:,:,k))*((v(:,:,k)*v(:,:,k)')/sin(th1k)^2);
V(:, :, k) = horzcat(vertcat(v(:,:,k), u(:,:,k)), vertcat(vu(:,:,k), v(:,:,k
    )'));

  end
end
```

## getVB

```
% DESCRIPTION: Function to construct the matrices VB_k (k = 1 to 3) from a
% 1x3 matrix containing theta coordinate values
%
% Input = thmtx  (1x3 matrix of theta coordinates)
%
% Output = VB     (2x2 multidimensional array that holds V_1 through V_3,
%                 with V_k corresponding to V(:, :, k) )

function VB = getVB(thmtx)
```

```
  p = 6;
  n=2;
  K=p-n;
  for k = 1:K
    th1k = thmtx(1, k);  % abbreviations to reduce the number of parentheses
    th2k = thmtx(2, k);

    VB(:, :, k) = [sin(th1k)*sin(th2k), 1-(1-cos(th1k))*(sin(th2k)^2), -(1-cos(th1k))*(sin(th2k)*cos(th2k)); sin(th1k)*cos(th2k), -(1-cos(th1k))*(sin(th2k)*cos(th2k)), 1-(1-cos(th1k))*(cos(th2k)^2); cos(th1k), -sin(th1k)*sin(th2k), -sin(th1k)*cos(th2k)];
   %VB(:, :, k) = [sin(th1k), 1-(1-cos(th1k)); cos(th1k), -sin(th1k)]
  end
end
```

## getVBnn

```
% DESCRIPTION: Function to construct the matrices VB_k (k = 1 to 3) from a
% 1x3 matrix containing theta coordinate values
%
% Input = thmtx  (1x3 matrix of theta coordinates)
%
% Output = VB     (2x2 multidimensional array that holds V_1 through V_3,
%                 with V_k corresponding to V(:, :, k) )

function VB = getVB(thmtx)
  p = 3;
  n=2;
  K=p-n;
  for k = 1:K
    th1k = thmtx(1, k);  % abbreviations to reduce the number of parentheses
    th2k = thmtx(2, k);

    VB(:, :, k) = [sin(th1k)*sin(th2k), 1-(1-cos(th1k))*(sin(th2k)^2), -(1-cos(th1k))*(sin(th2k)*cos(th2k)); sin(th1k)*cos(th2k), -(1-cos(th1k))*(sin(th2k)*cos(th2k)), 1-(1-cos(th1k))*(cos(th2k)^2); cos(th1k), -sin(th1k)*sin(th2k), -sin(th1k)*cos(th2k)];
   %VB(:, :, k) = [sin(th1k), 1-(1-cos(th1k)); cos(th1k), -sin(th1k)]
  end
end
```

## vecmSummary

```
% DESCRIPTION: A function that prints a summary of the fitted vector error
% correction model, including estimated parameters, to a file
%
% Input = data      (matrix of data)
%         mles      (cell array of structs with fitted VEC model information)
%         cirank    (the cointegration rank to use for the model)
%         fileout   (what to name the output file)
%
% Output = none

```

```
function vecSummary(data, mles, cirank, fileout)
  nobs = size(data, 1);
  nvars = size(data, 2);
  mdlInfo = mles.(cirank+1);  % select the model with the right coint. rank
  nlags = length(mdlInfo.paramNames) - 2;  % this assumes the H2 model type
  effecSampSize = nobs - nlags - 1;

  % Get the log likelihood and BIC
  logLik = mdlInfo.rLL;
  cointParams = 2 * nvars * cirank - cirank^2;
  lagParams = nlags * nvars^2;
  covarParams = nvars * (nvars + 1) / 2;
  totalParams = cointParams + lagParams + covarParams;
  [aic, bic] = aicbic(logLik, totalParams, effecSampSize*nvars);

  % Organize the parameter estimates in order to print
  C = mdlInfo.paramVals.A * mdlInfo.paramVals.B';  % alphabetaprime term
  betaprime = mdlInfo.paramVals.B'; %betaprime cointegration vector
  prNames = mdlInfo.paramNames;
  prVals = struct2cell(mdlInfo.paramVals);
  paramNames = ['C'; prNames(3:nlags+2); 'Q'];  % assumes H2 again
  paramVals = [{C}; prVals(3:nlags+2); {mdlInfo.EstCov}];

  % Print model summary to file
  fileID = fopen(fileout, 'w');
  fprintf(fileID, 'VECTOR ERROR CORRECTION MODEL SUMMARY\n');
  fprintf(fileID, '-------------------------------------------------------------\n');
  fprintf(fileID, ' Model eq: dy_t = C*y_t-1');
  for i = 1:nlags
    fprintf(fileID, ' + B%u*dy_t-%u', i, i);
  end
  fprintf(fileID, ' + e_t\n');
  fprintf(fileID, '           E[e_t*e_t''] = Q (innovations covariance)\n');
  fprintf(fileID, ' Effective sample size: %u\n', effecSampSize);
  fprintf(fileID, ' Lag order: %u\n', nlags);
  fprintf(fileID, ' Cointegration rank: %u\n', cirank);
  fprintf(fileID, ' Log likelihood: %5.2f\n', logLik);
  fprintf(fileID, ' BIC: %4.2f\n', bic);
  fprintf(fileID, ' Total Parameter: %4.2f\n', totalParams);
  fprintf(fileID, ' Betaprime: %4.2f\n', betaprime);
  fprintf(fileID, '\n\n  *Parameter estimates*\n\n');
  for i = 1:length(paramNames)
    fprintf(fileID, [formatMtxOut(paramNames{i}, paramVals{i}) '\n\n']);
  end
  fclose(fileID);
end


% DESCRIPTION: A function that reformats how a matrix is displayed/printed
%
% Input = name   (the name or symbol to use for the matrix when printing)
%         value  (the matrix itself)
%
% Output = mtxString (string containing the reformatted matrix for printing)
```

```
function mtxString = formatMtxOut(name, value)
  startFormat = get(0, 'Format'); % find out Matlab's current display
      setting
  nrows = size(value, 1);
  padLength = length(name) + 5;

  format shortG; % change Matlab's display setting
  T = evalc('value'); % get displayed matrix as a character vector
  newlineIxs = find(T == repmat(char(10), 1, length(T)));
  mtxRows = {};
  % Convert the rows of the matrix, as displayed, to a cell array
  for i = 1:nrows
    mtxRows = [mtxRows, T(newlineIxs(i+2)+1:newlineIxs(i+3))];
  end
  % Pad the displayed rows, including the matrix name with the first row
  mtxRows{1} = ['    ', name, ' =', mtxRows{1}];
  mtxString = mtxRows{1};
  for i = 2:nrows
    mtxRows{i} = [repmat(' ', 1, padLength), mtxRows{i}];
    mtxString = [mtxString mtxRows{i}];
  end

  format(startFormat); % return display setting to starting value
end
```

# Appendix C

# Estimation of the Classical Model (State Space, VAR)

cointRank

```
% DESCRIPTION: A function that prints selected Johansen cointegration test
% information to a file
%
% Input = data      (matrix of data)
%         vecLag    (lag order to use in vector error correction model)
%         fileout   (what to name the output file)
%
% Output = mles     (cell array of structs with fitted VEC model information)

function mles = cointRank(data, vecLag, fileout)
  nobs = size(data, 1);
  nvars = size(data, 2);

  warning('off', 'econ:jcitest:RightTailStatTooBig'); % suppress warnings

  % Johansen cointegration tests - trace and maximum eigenvalue
  [hTr, pTr, statTr, cTr, mlesTr] = jcitest(data, 'model', 'H2', 'lags',
      vecLag, 'test', 'trace', 'display', 'off');
  [hME, pME, statME, cME, mlesME] = jcitest(data, 'model', 'H2', 'lags',
      vecLag, 'test', 'maxeig', 'display', 'off');
  mles = mlesTr; % we could equally well use mlesME as the return value

  % Print test results to file
  fileID = fopen(fileout, 'w');
  fprintf(fileID, 'JOHANSEN COINTEGRATION TEST RESULTS\n');
  fprintf(fileID, '-----------------------------------------------------------\n');
  fprintf(fileID, ' Effective sample size: %u\n', nobs-vecLag-1);
  fprintf(fileID, ' VEC model type: H2 (no deterministic trends or
      intercepts)\n');
  fprintf(fileID, ' Lags: %u\n', vecLag);
  fprintf(fileID, '\n\n  *Trace test information*\n\n');
```

```
  fprintf(fileID, '   Coint rank      Trace stat       p-value      Eigenvalue\
      n');
  for i = 1:nvars
    fprintf(fileID, '%1$10u %2$16.4f %3$13.4f %4$13.4f\n', i-1, statTr.(i),
        pTr.(i), mlesTr.(i).eigVal);
  end
  fprintf(fileID, '\n\n  *Maximum eigenvalue test information*\n\n');
  fprintf(fileID, '   Coint rank      Max eig stat      p-value      Eigenvalue
      \n');
  for i = 1:nvars
    fprintf(fileID, '%1$10u %2$17.4f %3$13.4f %4$13.4f\n', i-1, statME.(i),
        pME.(i), mlesME.(i).eigVal);
  end
  fclose(fileID);
end
```

## formatMtxOut

```
% DESCRIPTION: A function that reformats how a matrix is displayed/printed
%
% Input = name   (the name or symbol to use for the matrix when printing)
%         value  (the matrix itself)
%
% Output = mtxString (string containing the reformatted matrix for printing)

function mtxString = formatMtxOut(name, value)
  startFormat = get(0, 'Format');  % find out Matlab's current display
      setting
  nrows = size(value, 1);
  padLength = length(name) + 5;

  format shortG;  % change Matlab's display setting
  T = evalc('value');  % get displayed matrix as a character vector
  newlineIxs = find(T == repmat(char(10), 1, length(T)));
  mtxRows = {};
  % Convert the rows of the matrix, as displayed, to a cell array
  for i = 1:nrows
    mtxRows = [mtxRows, T(newlineIxs(i+2)+1:newlineIxs(i+3))];
  end
  % Pad the displayed rows, including the matrix name with the first row
  mtxRows{1} = ['    ', name, ' =', mtxRows{1}];
  mtxString = mtxRows{1};
  for i = 2:nrows
    mtxRows{i} = [repmat(' ', 1, padLength), mtxRows{i}];
    mtxString = [mtxString mtxRows{i}];
  end

  format(startFormat);  % return display setting to starting value
end
```

## gener_state_main

```
clear all;
```

```
close all;
clc;

tic
% Read in data
data = csvread('gener_data_2000.csv')';
trueth = csvread('gener_theta_2000.csv');
% GenerateExcelFiles

% load DATA_FILE
data=data(:,1:2000);


% THESE VALUES CAN BE ADJUSTED
% In testing, one starting point with 100 steps took an average of 4.5 secs
nStart = 100;      % number of start points for gradient descent
maxSteps = 100;    % maximum number of descent steps for each start point
rng(963791);       % seed for random number generator (reproducible results)


% Initialize the minimum L value
minL = Inf;


% Find the best theta using steepest descent
for i = 1:nStart

    th_curr = genth_tol();
    out_curr = oneDescent(th_curr, data, maxSteps);
    reasons{i} = out_curr.endReason;
    if (abs(out_curr.LEnd) < minL)
        minL = out_curr.LEnd;
        minth = out_curr.thetaEnd;
    end
    fprintf('Current start point: %d of %d,  LEnd= %f,  Lmin = %f,  NSteps =%d,  End Reason: %s\n',i,nStart,out_curr.LEnd, minL,out_curr.nSteps,out_curr.endReason)
    disp('Difference between true theta and actual best estimation of theta:')
    DIFF=abs(trueth - minth)
end
mdl = getStateInfo(minth, data);
rsn_table = tabulate(reasons);
nReasons = size(rsn_table, 1);


% Print full summary of results to file
fileID = fopen('gener_state_output.txt', 'w');
fprintf(fileID, 'STATE SPACE ERROR CORRECTION MODEL SUMMARY\n');
fprintf(fileID, '--------------------------------------------------------------------------------\n');
fprintf(fileID, ' Model eq: dy_t = a*b''*y_t-1 + Cbar*Abar*(I_n-Abar)^-1*x~_t + e_t\n');
fprintf(fileID, '           x~_t+1 = Abar*x~_t + Bbar*dy_t\n');
```

```
fprintf(fileID, '           E[e_t*e_t''] = Sigma\n\n');
fprintf(fileID, '      s.t.  -a*b'' = Cbar*(I_n-Abar)^-1*Bbar + I_p\n\n');
fprintf(fileID, ' Effective sample size: %u\n', size(data, 2) - 1);
fprintf(fileID, ' Cointegration rank: 1\n');
fprintf(fileID, ' Log likelihood: %5.2f\n', mdl.logLik);
fprintf(fileID, ' BIC: %4.2f\n\n', mdl.BIC);
fprintf(fileID, ' Totla parameter: %4.2f\n\n', mdl.TotalParameter);
fprintf(fileID, ' True theta from data generating process:\n\n');
fprintf(fileID, [formatMtxOut('theta', trueth) '\n']);
fprintf(fileID, '
    ------------------------------------------------------------\n\n');
fprintf(fileID, '  *Parameter estimates from steepest descent method*\n');
fprintf(fileID, '   (%1$u starting points; up to %2$u descent steps per
    starting point)\n\n', nStart, maxSteps);
fprintf(fileID, [formatMtxOut('theta', minth) '\n\n']);
fprintf(fileID, [formatMtxOut('a*b''', mdl.abp) '\n\n']);
fprintf(fileID, [formatMtxOut('Abar', mdl.Abar) '\n\n']);
fprintf(fileID, [formatMtxOut('Bbar', mdl.Bbar) '\n\n']);
fprintf(fileID, [formatMtxOut('Cbar', mdl.Cbar) '\n\n']);
fprintf(fileID, formatMtxOut('Sigma', mdl.Sigma));

for j = 1:nReasons
  fprintf(fileID, '  %s: %u (out of %u)\n', rsn_table{j, 1}, rsn_table{j,
      2}, nStart);

end
fclose(fileID);
type('gener_state_output.txt')
toc
```

## genth_tol

```
% Function to generate a random matrix of theta values, essentially from the
% range -pi/2 to pi/2; the output matrix is 3x4
function thetas = genth_tol()
  n = 4;
  p = 3;
  thetas = zeros(p, n);
  thetas(1, :) = unifrnd(-pi/2+0.001, pi/2-0.001, 1, n);
  thetas(2:p, :) = unifrnd(-pi/2, pi/2, p-1, n);
  thetas=0.6*thetas;
end
```

## getL

```
% DESCRIPTION: A function that calculates the concentrated (adjusted
    negative
% log) likelihood and gradient of a 3-variable data series for a 4
    dimensional,
% cointegration rank 1 state space error correction model based on a given
% set of concentrated coordinate values
%
% Input = thmtx    (3x4 matrix of theta coordinates)
```

```
%          data      (matrix of data; observations in columns, last at right)
%
% Output = L         (concentrated adjusted negative log likelihood)
%           Ldot     (gradient of L with respect to theta, as a 3x4 matrix)

function [L, Ldot] = getL(thmtx, y)
  det_tol = 0.01;  % tolerance for determinant checks
  n = 4;
  p = size(y, 1);
  r = 1;
  T = size(y, 2);
  dy = diff(y')';
  y = y(:, 1:T-1);  % drop the last y value, to match the shorter dy series

  % begin by constructing estimates of V, R, bar_A, and bar_B, and their
  % partial derivatives from the current theta estimates
  Vhat = getV(thmtx);
  Gamma_1 = blkdiag(eye(n-1), Vhat(:, :, 1), eye(1-1));
  Gamma_2 = blkdiag(eye(n-2), Vhat(:, :, 2), eye(2-1));
  Gamma_3 = blkdiag(eye(n-3), Vhat(:, :, 3), eye(3-1));
  Gamma_4 = blkdiag(eye(n-4), Vhat(:, :, 4), eye(4-1));
  Rhat = Gamma_4 * Gamma_3 * Gamma_2 * Gamma_1;


  bar_Ahat = Rhat(p+1:n+p, p+1:n+p);
  bar_Bhat = Rhat(p+1:n+p, 1:p);

  % calculate the partial derivatives of V, Gamma, R, bar_A, and bar_B
  Vdot = getVderiv(thmtx);
  Gamma_1dot_11 = blkdiag(zeros(n-1), Vdot(:, :, 1, 1), zeros(1-1));
  Gamma_1dot_21 = blkdiag(zeros(n-1), Vdot(:, :, 2, 1), zeros(1-1));
  Gamma_1dot_31 = blkdiag(zeros(n-1), Vdot(:, :, 3, 1), zeros(1-1));
  Gamma_2dot_12 = blkdiag(zeros(n-2), Vdot(:, :, 1, 2), zeros(2-1));
  Gamma_2dot_22 = blkdiag(zeros(n-2), Vdot(:, :, 2, 2), zeros(2-1));
  Gamma_2dot_32 = blkdiag(zeros(n-2), Vdot(:, :, 3, 2), zeros(2-1));
  Gamma_3dot_13 = blkdiag(zeros(n-3), Vdot(:, :, 1, 3), zeros(3-1));
  Gamma_3dot_23 = blkdiag(zeros(n-3), Vdot(:, :, 2, 3), zeros(3-1));
  Gamma_3dot_33 = blkdiag(zeros(n-3), Vdot(:, :, 3, 3), zeros(3-1));
   Gamma_4dot_14 = blkdiag(zeros(n-4), Vdot(:, :, 1, 4), zeros(4-1));
   Gamma_4dot_24 = blkdiag(zeros(n-4), Vdot(:, :, 2, 4), zeros(4-1));
   Gamma_4dot_34 = blkdiag(zeros(n-4), Vdot(:, :, 3, 4), zeros(4-1));

  Rdot_11 = Gamma_4 * Gamma_3 * Gamma_2 * Gamma_1dot_11;
  Rdot_21 = Gamma_4 *Gamma_3 * Gamma_2 * Gamma_1dot_21;
  Rdot_31 = Gamma_4 * Gamma_3 * Gamma_2 * Gamma_1dot_31;
  Rdot_12 = Gamma_4 * Gamma_3 * Gamma_2dot_12 * Gamma_1;
  Rdot_22 = Gamma_4 * Gamma_3 * Gamma_2dot_22 * Gamma_1;
  Rdot_32 = Gamma_4 * Gamma_3 * Gamma_2dot_32 * Gamma_1;
  Rdot_13 = Gamma_4 * Gamma_3dot_13 * Gamma_2 * Gamma_1;
  Rdot_23 = Gamma_4 * Gamma_3dot_23 * Gamma_2 * Gamma_1;
  Rdot_33 = Gamma_4 * Gamma_3dot_33 * Gamma_2 * Gamma_1;
   Rdot_14 = Gamma_4dot_14 * Gamma_3 * Gamma_2 * Gamma_1;
   Rdot_24 = Gamma_4dot_24 * Gamma_3 * Gamma_2 * Gamma_1;
   Rdot_34 = Gamma_4dot_34 * Gamma_3 * Gamma_2 * Gamma_1;
```

```

bar_Adot_11 = Rdot_11(p+1:n+p, p+1:n+p);
bar_Adot_21 = Rdot_21(p+1:n+p, p+1:n+p);
bar_Adot_31 = Rdot_31(p+1:n+p, p+1:n+p);
bar_Adot_12 = Rdot_12(p+1:n+p, p+1:n+p);
bar_Adot_22 = Rdot_22(p+1:n+p, p+1:n+p);
bar_Adot_32 = Rdot_32(p+1:n+p, p+1:n+p);
bar_Adot_13 = Rdot_13(p+1:n+p, p+1:n+p);
bar_Adot_23 = Rdot_23(p+1:n+p, p+1:n+p);
bar_Adot_33 = Rdot_33(p+1:n+p, p+1:n+p);
 bar_Adot_14 = Rdot_14(p+1:n+p, p+1:n+p);
 bar_Adot_24 = Rdot_24(p+1:n+p, p+1:n+p);
 bar_Adot_34 = Rdot_34(p+1:n+p, p+1:n+p);

bar_Bdot_11 = Rdot_11(p+1:n+p, 1:p);
bar_Bdot_21 = Rdot_21(p+1:n+p, 1:p);
bar_Bdot_31 = Rdot_31(p+1:n+p, 1:p);
bar_Bdot_12 = Rdot_12(p+1:n+p, 1:p);
bar_Bdot_22 = Rdot_22(p+1:n+p, 1:p);
bar_Bdot_32 = Rdot_32(p+1:n+p, 1:p);
bar_Bdot_13 = Rdot_13(p+1:n+p, 1:p);
bar_Bdot_23 = Rdot_23(p+1:n+p, 1:p);
bar_Bdot_33 = Rdot_33(p+1:n+p, 1:p);
 bar_Bdot_14 = Rdot_14(p+1:n+p, 1:p);
 bar_Bdot_24 = Rdot_24(p+1:n+p, 1:p);
 bar_Bdot_34 = Rdot_34(p+1:n+p, 1:p);

% set up Z_0 and Z_1, and then construct Z_2 and its derivative from Z_0,
% Z_1, and the current estimates for bar_A and bar_B and their derivatives
% (this uses the method of extended error prediction filters described in the
% Ribarits-Hanzon article)
Z_0 = dy;
Z_1 = y;
Z_2 = zeros(n, T-1);
Z_2dot_11 = zeros(n, T-1);
Z_2dot_21 = zeros(n, T-1);
Z_2dot_31 = zeros(n, T-1);
Z_2dot_12 = zeros(n, T-1);
Z_2dot_22 = zeros(n, T-1);
Z_2dot_32 = zeros(n, T-1);
Z_2dot_13 = zeros(n, T-1);
Z_2dot_23 = zeros(n, T-1);
Z_2dot_33 = zeros(n, T-1);
 Z_2dot_14 = zeros(n, T-1);
 Z_2dot_24 = zeros(n, T-1);
 Z_2dot_34 = zeros(n, T-1);

Z_2(:, 1) = bar_Bhat * Z_1(:, 1);
Z_2dot_11(:, 1) = bar_Bdot_11 * Z_1(:, 1);
Z_2dot_21(:, 1) = bar_Bdot_21 * Z_1(:, 1);
Z_2dot_31(:, 1) = bar_Bdot_31 * Z_1(:, 1);
Z_2dot_12(:, 1) = bar_Bdot_12 * Z_1(:, 1);
Z_2dot_22(:, 1) = bar_Bdot_22 * Z_1(:, 1);
```

```
Z_2dot_32(:, 1) = bar_Bdot_32 * Z_1(:, 1);
Z_2dot_13(:, 1) = bar_Bdot_13 * Z_1(:, 1);
Z_2dot_23(:, 1) = bar_Bdot_23 * Z_1(:, 1);
Z_2dot_33(:, 1) = bar_Bdot_33 * Z_1(:, 1);
 Z_2dot_14(:, 1) = bar_Bdot_14 * Z_1(:, 1);
 Z_2dot_24(:, 1) = bar_Bdot_24 * Z_1(:, 1);
 Z_2dot_34(:, 1) = bar_Bdot_34 * Z_1(:, 1);

for t = 2:T-1
  Z_2(:, t) = bar_Ahat * Z_2(:, t-1) + bar_Bhat * Z_0(:, t-1);
  Z_2dot_11(:, t) = bar_Adot_11 * Z_2(:, t-1) + bar_Ahat * Z_2dot_11(:, t
      -1) + bar_Bdot_11 * Z_0(:, t-1);
  Z_2dot_21(:, t) = bar_Adot_21 * Z_2(:, t-1) + bar_Ahat * Z_2dot_21(:, t
      -1) + bar_Bdot_21 * Z_0(:, t-1);
  Z_2dot_31(:, t) = bar_Adot_31 * Z_2(:, t-1) + bar_Ahat * Z_2dot_31(:, t
      -1) + bar_Bdot_31 * Z_0(:, t-1);
  Z_2dot_12(:, t) = bar_Adot_12 * Z_2(:, t-1) + bar_Ahat * Z_2dot_12(:, t
      -1) + bar_Bdot_12 * Z_0(:, t-1);
  Z_2dot_22(:, t) = bar_Adot_22 * Z_2(:, t-1) + bar_Ahat * Z_2dot_22(:, t
      -1) + bar_Bdot_22 * Z_0(:, t-1);
  Z_2dot_32(:, t) = bar_Adot_32 * Z_2(:, t-1) + bar_Ahat * Z_2dot_32(:, t
      -1) + bar_Bdot_32 * Z_0(:, t-1);
  Z_2dot_13(:, t) = bar_Adot_13 * Z_2(:, t-1) + bar_Ahat * Z_2dot_13(:, t
      -1) + bar_Bdot_13 * Z_0(:, t-1);
  Z_2dot_23(:, t) = bar_Adot_23 * Z_2(:, t-1) + bar_Ahat * Z_2dot_23(:, t
      -1) + bar_Bdot_23 * Z_0(:, t-1);
  Z_2dot_33(:, t) = bar_Adot_33 * Z_2(:, t-1) + bar_Ahat * Z_2dot_33(:, t
      -1) + bar_Bdot_33 * Z_0(:, t-1);
   Z_2dot_14(:, t) = bar_Adot_14 * Z_2(:, t-1) + bar_Ahat * Z_2dot_14(:, t
       -1) + bar_Bdot_14 * Z_0(:, t-1);
   Z_2dot_24(:, t) = bar_Adot_24 * Z_2(:, t-1) + bar_Ahat * Z_2dot_24(:, t
       -1) + bar_Bdot_24 * Z_0(:, t-1);
   Z_2dot_34(:, t) = bar_Adot_34 * Z_2(:, t-1) + bar_Ahat * Z_2dot_34(:, t
       -1) + bar_Bdot_34 * Z_0(:, t-1);
end

% calculate the M_ij matrices and their derivatives
M_02 = Z_0 * Z_2' / T;
M_11 = Z_1 * Z_1' / T;
M_12 = Z_1 * Z_2' / T;
M_21 = M_12';
M_22 = Z_2 * Z_2' / T;

M_02dot_11 = Z_0 * Z_2dot_11' / T;
M_02dot_21 = Z_0 * Z_2dot_21' / T;
M_02dot_31 = Z_0 * Z_2dot_31' / T;
M_02dot_12 = Z_0 * Z_2dot_12' / T;
M_02dot_22 = Z_0 * Z_2dot_22' / T;
M_02dot_32 = Z_0 * Z_2dot_32' / T;
M_02dot_13 = Z_0 * Z_2dot_13' / T;
M_02dot_23 = Z_0 * Z_2dot_23' / T;
M_02dot_33 = Z_0 * Z_2dot_33' / T;
 M_02dot_14 = Z_0 * Z_2dot_14' / T;
 M_02dot_24 = Z_0 * Z_2dot_24' / T;
```

```
 M_02dot_34 = Z_0 * Z_2dot_34' / T;

M_12dot_11 = Z_1 * Z_2dot_11' / T;
M_12dot_21 = Z_1 * Z_2dot_21' / T;
M_12dot_31 = Z_1 * Z_2dot_31' / T;
M_12dot_12 = Z_1 * Z_2dot_12' / T;
M_12dot_22 = Z_1 * Z_2dot_22' / T;
M_12dot_32 = Z_1 * Z_2dot_32' / T;
M_12dot_13 = Z_1 * Z_2dot_13' / T;
M_12dot_23 = Z_1 * Z_2dot_23' / T;
M_12dot_33 = Z_1 * Z_2dot_33' / T;
 M_12dot_14 = Z_1 * Z_2dot_14' / T;
 M_12dot_24 = Z_1 * Z_2dot_24' / T;
 M_12dot_34 = Z_1 * Z_2dot_34' / T;

M_22dot_11 = (Z_2dot_11 * Z_2' + Z_2 * Z_2dot_11') / T;
M_22dot_21 = (Z_2dot_21 * Z_2' + Z_2 * Z_2dot_21') / T;
M_22dot_31 = (Z_2dot_31 * Z_2' + Z_2 * Z_2dot_31') / T;
M_22dot_12 = (Z_2dot_12 * Z_2' + Z_2 * Z_2dot_12') / T;
M_22dot_22 = (Z_2dot_22 * Z_2' + Z_2 * Z_2dot_22') / T;
M_22dot_32 = (Z_2dot_32 * Z_2' + Z_2 * Z_2dot_32') / T;
M_22dot_13 = (Z_2dot_13 * Z_2' + Z_2 * Z_2dot_13') / T;
M_22dot_23 = (Z_2dot_23 * Z_2' + Z_2 * Z_2dot_23') / T;
M_22dot_33 = (Z_2dot_33 * Z_2' + Z_2 * Z_2dot_33') / T;
 M_22dot_14 = (Z_2dot_14 * Z_2' + Z_2 * Z_2dot_14') / T;
 M_22dot_24 = (Z_2dot_24 * Z_2' + Z_2 * Z_2dot_24') / T;
 M_22dot_34 = (Z_2dot_34 * Z_2' + Z_2 * Z_2dot_34') / T;

% check determinant of M_ij block matrix and display warning if necessary
M_det = det(vertcat(horzcat(M_22, M_21), horzcat(M_12, M_11)));
if (M_det < det_tol)
  disp('Warning: M_ij block determinant');
end

% calculate the H_ij matrices; we do this by inverting the RMblock matrix and
% then partitioning the result (see equation A.7 in the Ribarits-Hanzon
% article), which seems to give more accurate results than using the longer
% product and inverse forms used in the original script (and in equation
% A.8 of the article)
% also calculate the derivatives of the H_ij matrices (several intermediate
% derivatives are required for this)
IAinv = inv(eye(n) - bar_Ahat);
Rt = IAinv * bar_Bhat;
Apow = bar_Ahat * IAinv;
Mt = Apow * M_22 * Apow';
RMblock = vertcat(horzcat(Mt, Rt), horzcat(Rt', zeros(p)));
RMblockinv = inv(RMblock);
H_11 = RMblockinv(1:n, 1:n);
H_12 = RMblockinv(1:n, n+1:n+p);
H_21 = RMblockinv(n+1:n+p, 1:n);
```

```
Rtdot_11 = IAinv * (bar_Adot_11 * Rt + bar_Bdot_11);
Rtdot_21 = IAinv * (bar_Adot_21 * Rt + bar_Bdot_21);
Rtdot_31 = IAinv * (bar_Adot_31 * Rt + bar_Bdot_31);
Rtdot_12 = IAinv * (bar_Adot_12 * Rt + bar_Bdot_12);
Rtdot_22 = IAinv * (bar_Adot_22 * Rt + bar_Bdot_22);
Rtdot_32 = IAinv * (bar_Adot_32 * Rt + bar_Bdot_32);
Rtdot_13 = IAinv * (bar_Adot_13 * Rt + bar_Bdot_13);
Rtdot_23 = IAinv * (bar_Adot_23 * Rt + bar_Bdot_23);
Rtdot_33 = IAinv * (bar_Adot_33 * Rt + bar_Bdot_33);
Rtdot_14 = IAinv * (bar_Adot_14 * Rt + bar_Bdot_14);
Rtdot_24 = IAinv * (bar_Adot_24 * Rt + bar_Bdot_24);
Rtdot_34 = IAinv * (bar_Adot_34 * Rt + bar_Bdot_34);

Apdot_11 = (eye(n) + Apow) * bar_Adot_11 * IAinv;
Apdot_21 = (eye(n) + Apow) * bar_Adot_21 * IAinv;
Apdot_31 = (eye(n) + Apow) * bar_Adot_31 * IAinv;
Apdot_12 = (eye(n) + Apow) * bar_Adot_12 * IAinv;
Apdot_22 = (eye(n) + Apow) * bar_Adot_22 * IAinv;
Apdot_32 = (eye(n) + Apow) * bar_Adot_32 * IAinv;
Apdot_13 = (eye(n) + Apow) * bar_Adot_13 * IAinv;
Apdot_23 = (eye(n) + Apow) * bar_Adot_23 * IAinv;
Apdot_33 = (eye(n) + Apow) * bar_Adot_33 * IAinv;
 Apdot_14 = (eye(n) + Apow) * bar_Adot_14 * IAinv;
 Apdot_24 = (eye(n) + Apow) * bar_Adot_24 * IAinv;
 Apdot_34 = (eye(n) + Apow) * bar_Adot_34 * IAinv;

leftdot_11 = Apdot_11 * M_22 * Apow';
Mtdot_11 = leftdot_11 + Apow * M_22dot_11 * Apow' + leftdot_11';
leftdot_21 = Apdot_21 * M_22 * Apow';
Mtdot_21 = leftdot_21 + Apow * M_22dot_21 * Apow' + leftdot_21';
leftdot_31 = Apdot_31 * M_22 * Apow';
Mtdot_31 = leftdot_31 + Apow * M_22dot_31 * Apow' + leftdot_31';
leftdot_12 = Apdot_12 * M_22 * Apow';
Mtdot_12 = leftdot_12 + Apow * M_22dot_12 * Apow' + leftdot_12';
leftdot_22 = Apdot_22 * M_22 * Apow';
Mtdot_22 = leftdot_22 + Apow * M_22dot_22 * Apow' + leftdot_22';
leftdot_32 = Apdot_32 * M_22 * Apow';
Mtdot_32 = leftdot_32 + Apow * M_22dot_32 * Apow' + leftdot_32';
leftdot_13 = Apdot_13 * M_22 * Apow';
Mtdot_13 = leftdot_13 + Apow * M_22dot_13 * Apow' + leftdot_13';
leftdot_23 = Apdot_23 * M_22 * Apow';
Mtdot_23 = leftdot_23 + Apow * M_22dot_23 * Apow' + leftdot_23';
leftdot_33 = Apdot_33 * M_22 * Apow';
Mtdot_33 = leftdot_33 + Apow * M_22dot_33 * Apow' + leftdot_33';
leftdot_14 = Apdot_14 * M_22 * Apow';
Mtdot_14 = leftdot_14 + Apow * M_22dot_14 * Apow' + leftdot_14';
leftdot_24 = Apdot_24 * M_22 * Apow';
Mtdot_24 = leftdot_24 + Apow * M_22dot_24 * Apow' + leftdot_24';
leftdot_34 = Apdot_34 * M_22 * Apow';
Mtdot_34 = leftdot_34 + Apow * M_22dot_34 * Apow' + leftdot_34';

RMbidot_11 = -RMblockinv * vertcat(horzcat(Mtdot_11, Rtdot_11), horzcat(
    Rtdot_11', zeros(p))) * RMblockinv;
```

```
RMbidot_21 = −RMblockinv * vertcat(horzcat(Mtdot_21, Rtdot_21), horzcat(
    Rtdot_21', zeros(p))) * RMblockinv;
RMbidot_31 = −RMblockinv * vertcat(horzcat(Mtdot_31, Rtdot_31), horzcat(
    Rtdot_31', zeros(p))) * RMblockinv;
RMbidot_12 = −RMblockinv * vertcat(horzcat(Mtdot_12, Rtdot_12), horzcat(
    Rtdot_12', zeros(p))) * RMblockinv;
RMbidot_22 = −RMblockinv * vertcat(horzcat(Mtdot_22, Rtdot_22), horzcat(
    Rtdot_22', zeros(p))) * RMblockinv;
RMbidot_32 = −RMblockinv * vertcat(horzcat(Mtdot_32, Rtdot_32), horzcat(
    Rtdot_32', zeros(p))) * RMblockinv;
RMbidot_13 = −RMblockinv * vertcat(horzcat(Mtdot_13, Rtdot_13), horzcat(
    Rtdot_13', zeros(p))) * RMblockinv;
RMbidot_23 = −RMblockinv * vertcat(horzcat(Mtdot_23, Rtdot_23), horzcat(
    Rtdot_23', zeros(p))) * RMblockinv;
RMbidot_33 = −RMblockinv * vertcat(horzcat(Mtdot_33, Rtdot_33), horzcat(
    Rtdot_33', zeros(p))) * RMblockinv;
RMbidot_14 = −RMblockinv * vertcat(horzcat(Mtdot_14, Rtdot_14), horzcat(
    Rtdot_14', zeros(p))) * RMblockinv;
RMbidot_24 = −RMblockinv * vertcat(horzcat(Mtdot_24, Rtdot_24), horzcat(
    Rtdot_24', zeros(p))) * RMblockinv;
RMbidot_34 = −RMblockinv * vertcat(horzcat(Mtdot_34, Rtdot_34), horzcat(
    Rtdot_34', zeros(p))) * RMblockinv;

H_11dot_11 = RMbidot_11(1:n, 1:n);
H_11dot_21 = RMbidot_21(1:n, 1:n);
H_11dot_31 = RMbidot_31(1:n, 1:n);
H_11dot_12 = RMbidot_12(1:n, 1:n);
H_11dot_22 = RMbidot_22(1:n, 1:n);
H_11dot_32 = RMbidot_32(1:n, 1:n);
H_11dot_13 = RMbidot_13(1:n, 1:n);
H_11dot_23 = RMbidot_23(1:n, 1:n);
H_11dot_33 = RMbidot_33(1:n, 1:n);
H_11dot_14 = RMbidot_14(1:n, 1:n);
H_11dot_24 = RMbidot_24(1:n, 1:n);
H_11dot_34 = RMbidot_34(1:n, 1:n);

H_21dot_11 = RMbidot_11(n+1:n+p, 1:n);
H_21dot_21 = RMbidot_21(n+1:n+p, 1:n);
H_21dot_31 = RMbidot_31(n+1:n+p, 1:n);
H_21dot_12 = RMbidot_12(n+1:n+p, 1:n);
H_21dot_22 = RMbidot_22(n+1:n+p, 1:n);
H_21dot_32 = RMbidot_32(n+1:n+p, 1:n);
H_21dot_13 = RMbidot_13(n+1:n+p, 1:n);
H_21dot_23 = RMbidot_23(n+1:n+p, 1:n);
H_21dot_33 = RMbidot_33(n+1:n+p, 1:n);
H_21dot_14 = RMbidot_14(n+1:n+p, 1:n);
H_21dot_24 = RMbidot_24(n+1:n+p, 1:n);
H_21dot_34 = RMbidot_34(n+1:n+p, 1:n);

% calculate the R_0 and R_1 series and the S_ij matrices, as well as their
% derivatives
R_0 = Z_0 − (M_02 * Apow' * H_11 − H_21) * Apow * Z_2;
R_1 = Z_1 − (M_12 * Apow' * H_11 + H_21) * Apow * Z_2;
S_00 = R_0 * R_0' / T;
```

```
S_01 = R_0 * R_1' / T;
S_10 = S_01';
S_11 = R_1 * R_1' / T;

% check determinant of S_00 and display warning if necessary
if (det(S_00) < det_tol)
  disp('Warning: S_00 determinant');
end

% solve the eigenvalue problems, making sure to sort the eigenvalues in
% descending order; since the first eigenvalue problem involves discarding
% eigenvalues that are 0, and there could be rounding error, we make use
% of an eig_tol variable again (eigenvalues smaller than the tolerance are
% discarded as if they were equal to 0)
eig_tol = 0.000001;
S_11sym = (S_11 + S_11') / 2;  % ensure that S_11 is exactly symmetric
[W, P] = eig(S_11sym);
[Pdiag, perm1] = sort(diag(P), 'descend');
W = W(:, perm1);
p_2 = sum(Pdiag > eig_tol);
% if necessary, display warnings about several issues that could arise in
    the
% first eigenvalue problem
if (sum(Pdiag < 0) > 0)
  disp('Warning: Unexpected negative eigenvalue(s) of S_11');
end
if (sum(Pdiag > 0) - p_2 > 0)
  disp('Warning: Discarded small positive eigenvalue(s) of S_11');
end
if (sum(-diff(Pdiag) < eig_tol) > 0)
  disp('Warning: Possible repeat eigenvalue of S_11');
end
if (p_2 == 0)
  disp('Warning: No satisfactory eigenvalues of S_11');
end
C = W(:, 1:p_2) * diag(1 ./ sqrt(Pdiag(1:p_2)));
SC = S_01 * C;  % for the next line, note that SC' = C' * S_10
S_00i = inv(S_00);
M2 = SC' * S_00i * SC;
M2sym = (M2 + M2') / 2;  % ensure that M2 is exactly symmetric
[V2, Lambda] = eig(M2sym);
[Lambdadiag, perm2] = sort(diag(Lambda), 'descend');
V2 = V2(:, perm2);
% if necessary, display warnings about two issues that could arise in the
% second eigenvalue problem
if (Lambdadiag(1) - Lambdadiag(2) < eig_tol)
  disp('Warning: Possible repeat eigenvalue in second eigenvalue problem')
      ;
end
if (sum(Lambdadiag > eig_tol) == 0)
  disp('Warning: No satisfactory eigenvalues in second eigenvalue problem;
       proceeding with highest eigenvalue');
end
```

```
% construct our estimates for alpha, beta, bar_C, and Sigma; we assume that
% r = p-1 when estimating beta and alpha (i.e. that they are vectors); also
% check the positivity of two other determinants and display warnings if
% necessary
betahat = C * V2(:, r);
Mlike = R_0 - S_01 * betahat * betahat' * R_1;
Slike = Mlike * Mlike' / T;
if (det(Slike) < det_tol)
  disp('Warning: Slike determinant');
end
alphahat = S_01 * betahat;
albepr = alphahat * betahat';
Sigmahat = S_00 - albepr * S_10;
bar_Chat = (M_02 - albepr * M_12) * Apow' * H_11 - (eye(p) + albepr) * H_21;

% find the derivatives of the estimate of bar_C
bar_Cdot_11 = (M_02dot_11 - albepr * M_12dot_11) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_11' * H_11 + Apow' * H_11dot_11) - (eye(p) + albepr) * H_21dot_11;
bar_Cdot_21 = (M_02dot_21 - albepr * M_12dot_21) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_21' * H_11 + Apow' * H_11dot_21) - (eye(p) + albepr) * H_21dot_21;
bar_Cdot_31 = (M_02dot_31 - albepr * M_12dot_31) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_31' * H_11 + Apow' * H_11dot_31) - (eye(p) + albepr) * H_21dot_31;
bar_Cdot_12 = (M_02dot_12 - albepr * M_12dot_12) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_12' * H_11 + Apow' * H_11dot_12) - (eye(p) + albepr) * H_21dot_12;
bar_Cdot_22 = (M_02dot_22 - albepr * M_12dot_22) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_22' * H_11 + Apow' * H_11dot_22) - (eye(p) + albepr) * H_21dot_22;
bar_Cdot_32 = (M_02dot_32 - albepr * M_12dot_32) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_32' * H_11 + Apow' * H_11dot_32) - (eye(p) + albepr) * H_21dot_32;
bar_Cdot_13 = (M_02dot_13 - albepr * M_12dot_13) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_13' * H_11 + Apow' * H_11dot_13) - (eye(p) + albepr) * H_21dot_13;
bar_Cdot_23 = (M_02dot_23 - albepr * M_12dot_23) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_23' * H_11 + Apow' * H_11dot_23) - (eye(p) + albepr) * H_21dot_23;
bar_Cdot_33 = (M_02dot_33 - albepr * M_12dot_33) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_33' * H_11 + Apow' * H_11dot_33) - (eye(p) + albepr) * H_21dot_33;
bar_Cdot_14 = (M_02dot_14 - albepr * M_12dot_14) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_14' * H_11 + Apow' * H_11dot_14) - (eye(p) + albepr) * H_21dot_14;
bar_Cdot_24 = (M_02dot_24 - albepr * M_12dot_24) * Apow' * H_11 + (M_02 - albepr * M_12) * (Apdot_24' * H_11 + Apow' * H_11dot_24) - (eye(p) + albepr) * H_21dot_24;
```

```
bar_Cdot_34 = (M_02dot_34 - albepr * M_12dot_34) * Apow' * H_11 + (M_02 -
    albepr * M_12) * (Apdot_34' * H_11 + Apow' * H_11dot_34) - (eye(p) +
    albepr) * H_21dot_34;

% calculate the (modified negative log) likelihood using the original
    formula
M = Z_0 - albepr * Z_1 - bar_Chat * Apow * Z_2;
Sigmadet = det(Sigmahat);
Sigmainv = inv(Sigmahat);
MSiM = M .* (Sigmainv * M);
L = log(Sigmadet) + sum(MSiM(:)) / T;

% calculate the derivatives of the likelihood using the original formula
Mdot_11 = - bar_Cdot_11 * Apow * Z_2 - bar_Chat * (Apdot_11 * Z_2 + Apow *
     Z_2dot_11);
Mdot_21 = - bar_Cdot_21 * Apow * Z_2 - bar_Chat * (Apdot_21 * Z_2 + Apow *
     Z_2dot_21);
Mdot_31 = - bar_Cdot_31 * Apow * Z_2 - bar_Chat * (Apdot_31 * Z_2 + Apow *
     Z_2dot_31);
Mdot_12 = - bar_Cdot_12 * Apow * Z_2 - bar_Chat * (Apdot_12 * Z_2 + Apow *
     Z_2dot_12);
Mdot_22 = - bar_Cdot_22 * Apow * Z_2 - bar_Chat * (Apdot_22 * Z_2 + Apow *
     Z_2dot_22);
Mdot_32 = - bar_Cdot_32 * Apow * Z_2 - bar_Chat * (Apdot_32 * Z_2 + Apow *
     Z_2dot_32);
Mdot_13 = - bar_Cdot_13 * Apow * Z_2 - bar_Chat * (Apdot_13 * Z_2 + Apow *
     Z_2dot_13);
Mdot_23 = - bar_Cdot_23 * Apow * Z_2 - bar_Chat * (Apdot_23 * Z_2 + Apow *
     Z_2dot_23);
Mdot_33 = - bar_Cdot_33 * Apow * Z_2 - bar_Chat * (Apdot_33 * Z_2 + Apow *
     Z_2dot_33);
Mdot_14 = - bar_Cdot_14 * Apow * Z_2 - bar_Chat * (Apdot_14 * Z_2 + Apow *
     Z_2dot_14);
Mdot_24 = - bar_Cdot_24 * Apow * Z_2 - bar_Chat * (Apdot_24 * Z_2 + Apow *
     Z_2dot_24);
Mdot_34 = - bar_Cdot_34 * Apow * Z_2 - bar_Chat * (Apdot_34 * Z_2 + Apow *
     Z_2dot_34);

MSiMdot_11 = (2 * Mdot_11) .* (Sigmainv * M);
MSiMdot_21 = (2 * Mdot_21) .* (Sigmainv * M);
MSiMdot_31 = (2 * Mdot_31) .* (Sigmainv * M);
MSiMdot_12 = (2 * Mdot_12) .* (Sigmainv * M);
MSiMdot_22 = (2 * Mdot_22) .* (Sigmainv * M);
MSiMdot_32 = (2 * Mdot_32) .* (Sigmainv * M);
MSiMdot_13 = (2 * Mdot_13) .* (Sigmainv * M);
MSiMdot_23 = (2 * Mdot_23) .* (Sigmainv * M);
MSiMdot_33 = (2 * Mdot_33) .* (Sigmainv * M);
MSiMdot_14 = (2 * Mdot_14) .* (Sigmainv * M);
MSiMdot_24 = (2 * Mdot_24) .* (Sigmainv * M);
MSiMdot_34 = (2 * Mdot_34) .* (Sigmainv * M);

Ldot_11 = sum(MSiMdot_11(:)) / T;
Ldot_21 = sum(MSiMdot_21(:)) / T;
Ldot_31 = sum(MSiMdot_31(:)) / T;
```

```
  Ldot_12 = sum(MSiMdot_12(:)) / T;
  Ldot_22 = sum(MSiMdot_22(:)) / T;
  Ldot_32 = sum(MSiMdot_32(:)) / T;
  Ldot_13 = sum(MSiMdot_13(:)) / T;
  Ldot_23 = sum(MSiMdot_23(:)) / T;
  Ldot_33 = sum(MSiMdot_33(:)) / T;
  Ldot_14 = sum(MSiMdot_14(:)) / T;
  Ldot_24 = sum(MSiMdot_24(:)) / T;
  Ldot_34 = sum(MSiMdot_34(:)) / T;

  Ldot = [Ldot_11, Ldot_12, Ldot_13, Ldot_14; Ldot_21, Ldot_22, Ldot_23,
      Ldot_24; Ldot_31, Ldot_32, Ldot_33, Ldot_34];
end
```

## getStateInfo

```
% DESCRIPTION: A function that calculates the concentrated (adjusted
    negative
% log) likelihood of a 3-variable data series for a 4 dimensional,
% cointegration rank p-1 state space error correction model based on a given
% set of concentrated coordinate values [does not calculate gradient]
%
% Input = thmtx      (3x4 matrix of theta coordinates)
%         data       (matrix of data; observations in columns, last at right
    )
%
% Output = mdlInfo   (Matlab structure containing the following fields)
%              Abar   (MLE for Abar matrix in model)
%              Bbar   (MLE for Bbar matrix in model)
%              Cbar   (MLE for Cbar matrix in model)
%              abp    (MLE for alpha*beta' matrix in model)
%              Sigma  (MLE for innovations covariance matrix in model)
%              L      (concentrated adjusted negative log likelihood)
%              logLik (log likelihood)
%              BIC    (Bayesian information criterion)
%              TP     (Total parameter)

function mdlInfo = getStateInfo(thmtx, y)
  det_tol = 0.01;  % tolerance for determinant checks
  n = 4;
  p = size(y, 1);
  r = 1;
  T = size(y, 2);
  dy = diff(y')';
  y = y(:, 1:T-1);  % drop the last y value, to match the shorter dy series

  % begin by constructing estimates of V, R, bar_A, and bar_B, and their
  % partial derivatives from the current theta estimates
  Vhat = getV(thmtx);
  for k = 1:n
    Gamma(:, :, k) = blkdiag(eye(n-k), Vhat(:, :, k), eye(k-1));
  end
  Rhat = Gamma(:, :, 4) * Gamma(:, :, 3) * Gamma(:, :, 2) * Gamma(:, :, 1);

```

```
bar_Ahat = Rhat(p+1:n+p, p+1:n+p)
bar_Bhat = Rhat(p+1:n+p, 1:p)

% set up Z_0 and Z_1, and then construct Z_2 and its derivative from Z_0,
% Z_1, and the current estimates for bar_A and bar_B and their derivatives
% (this uses the method of extended error prediction filters described in the
% Ribarits-Hanzon article)
Z_0 = dy;
Z_1 = y;
Z_2 = zeros(n, T-1);

Z_2(:, 1) = bar_Bhat * Z_1(:, 1);
for t = 2:T-1
  Z_2(:, t) = bar_Ahat * Z_2(:, t-1) + bar_Bhat * Z_0(:, t-1);
end

% calculate the M_ij matrices and their derivatives
M_02 = Z_0 * Z_2' / T;
M_11 = Z_1 * Z_1' / T;
M_12 = Z_1 * Z_2' / T;
M_21 = M_12';
M_22 = Z_2 * Z_2' / T;

% check determinant of M_ij block matrix and display warning if necessary
M_det = det(vertcat(horzcat(M_22, M_21), horzcat(M_12, M_11)));
if (M_det < det_tol)
  disp('Warning: M_ij block determinant');
end
% calculate the H_ij matrices; we do this by inverting the RMblock matrix and
% then partitioning the result (see equation A.7 in the Ribarits-Hanzon
% article), which seems to give more accurate results than using the longer
% product and inverse forms used in the original script (and in equation
% A.8 of the article)
% also calculate the derivatives of the H_ij matrices (several intermediate
% derivatives are required for this)
IAinv = inv(eye(n) - bar_Ahat);
Rt = IAinv * bar_Bhat;
Apow = bar_Ahat * IAinv;
Mt = Apow * M_22 * Apow';
RMblock = vertcat(horzcat(Mt, Rt), horzcat(Rt', zeros(p)));
 if ( RMblock < 1e-17)
  disp('Warning: RMblock block determinant');
end
RMblockinv = inv(RMblock);
H_11 = RMblockinv(1:n, 1:n);
H_12 = RMblockinv(1:n, n+1:n+p);
H_21 = RMblockinv(n+1:n+p, 1:n);

% calculate the R_0 and R_1 series and the S_ij matrices, as well as their
% derivatives
```

```
R_0 = Z_0 - (M_02 * Apow' * H_11 - H_21) * Apow * Z_2;
R_1 = Z_1 - (M_12 * Apow' * H_11 + H_21) * Apow * Z_2;
S_00 = R_0 * R_0' / T;
S_01 = R_0 * R_1' / T;
S_10 = S_01';
S_11 = R_1 * R_1' / T;

% solve the eigenvalue problems, making sure to sort the eigenvalues in
% descending order; since the first eigenvalue problem involves discarding
% eigenvalues that are 0, and there could be rounding error, we make use
% of an eig_tol variable again (eigenvalues smaller than the tolerance are
% discarded as if they were equal to 0)
eig_tol = 0.000001;
S_11sym = (S_11 + S_11') / 2;  % ensure that S_11 is exactly symmetric
[W, P] = eig(S_11sym);
[Pdiag, perm1] = sort(diag(P), 'descend');
W = W(:, perm1);
p_2 = sum(Pdiag > eig_tol);
C = W(:, 1:p_2) * diag(1 ./ sqrt(Pdiag(1:p_2)));
SC = S_01 * C;  % for the next line, note that SC' = C' * S_10
S_00i = inv(S_00);
M2 = SC' * S_00i * SC;
M2sym = (M2 + M2') / 2;  % ensure that M2 is exactly symmetric
[V2, Lambda] = eig(M2sym);
[Lambdadiag, perm2] = sort(diag(Lambda), 'descend');
V2 = V2(:, perm2);
V2first = V2(:, r);

% construct our estimates for alpha, beta, bar_C, and Sigma; we assume
    that
% r = 1 when estimating beta and alpha (i.e. that they are vectors)
betahat = C * V2first;
alphahat = S_01 * betahat;
albepr = alphahat * betahat';
Sigmahat = S_00 - albepr * S_10;
bar_Chat = (M_02 - albepr * M_12) * Apow' * H_11 - (eye(p) + albepr) *
    H_21;

% calculate the (modified negative log) likelihood using the original
    formula
M = Z_0 - albepr * Z_1 - bar_Chat * Apow * Z_2;
Sigmadet = det(Sigmahat);
Sigmainv = inv(Sigmahat);
MSiM = M .* (Sigmainv * M);
L = log(Sigmadet) + sum(MSiM(:)) / T;

% calculate the actual log likelihood and the BIC
logLik = -(T-1)/2 * (L + p * log(2*pi));
c = p - r;  % number of 'common trends'
mainParams = 2*n*p - c^2;
covarParams = p * (p + 1) / 2;
totalParams = mainParams + covarParams;
[aic, bic] = aicbic(logLik, totalParams, (T-1)*p);

```

```
  mdlInfo = struct('Abar', bar_Ahat, 'Bbar', bar_Bhat, 'Cbar', bar_Chat, 'abp', albepr,'Betaprim',betahat , 'Sigma', Sigmahat, 'L', L, 'logLik', logLik, 'BIC', bic ,'TotalParameter',totalParams);
end
```

## getV

```
% DESCRIPTION: Function to construct the matrices V_k (k = 1 to 4) from a
% 3x4 matrix containing theta coordinate values
%
% Input = thmtx  (3x4 matrix of theta coordinates)
%
% Output = V     (4x4x4 multidimensional array that holds V_1 through V_4,
%                 with V_k corresponding to V(:, :, k) )

function V = getV(thmtx)
  n = 4;
  for k = 1:n
    th1k = thmtx(1, k);  % abbreviations to reduce the number of parentheses
    th2k = thmtx(2, k);
    th3k = thmtx(3, k);
    V(:, :, k) = [sin(th1k)*sin(th2k)*sin(th3k), 1-(1-cos(th1k))*(sin(th2k)^2*sin(th3k)^2), -(1-cos(th1k))*(sin(th2k)^2*sin(th3k)*cos(th3k)), -(1-cos(th1k))*(sin(th2k)*sin(th3k)*cos(th2k)); sin(th1k)*sin(th2k)*cos(th3k), -(1-cos(th1k))*(sin(th2k)^2*sin(th3k)*cos(th3k)), 1-(1-cos(th1k))*(sin(th2k)^2*cos(th3k)^2), -(1-cos(th1k))*(sin(th2k)*cos(th2k)*cos(th3k)); sin(th1k)*cos(th2k), -(1-cos(th1k))*(sin(th2k)*sin(th3k)*cos(th2k)), -(1-cos(th1k))*(sin(th2k)*cos(th2k)*cos(th3k)), 1-(1-cos(th1k))*(cos(th2k)^2); cos(th1k), -sin(th1k)*sin(th2k)*sin(th3k), -sin(th1k)*sin(th2k)*cos(th3k), -sin(th1k)*cos(th2k)];
  end
end
```

## getVderiv

```
% DESCRIPTION: Function to construct the partial derivatives of the matrices
% V_k (k = 1 to 4) for a 3x4 matrix containing theta coordinate values; note
% that since V_k only depends on the kth column of theta coordinate values,
% each V_k only has three partial derivatives
%
% Input = thmtx  (3x4 matrix of theta coordinates)
%
% Output = Vdot  (4x4x4x4 multidimensional array that holds the partial
%                 derivatives, with Vdot(:, :, i, k) corresponding to the
%                 derivative of V_k with respect to theta_i,k)

function Vdot = getVderiv(thmtx)
  n = 4;
  for k = 1:n
    th1k = thmtx(1, k);  % abbreviations to reduce the number of parentheses
    th2k = thmtx(2, k);
    th3k = thmtx(3, k);
```

```
    Vdot(:, :, 1, k) = [cos(th1k)*sin(th2k)*sin(th3k), -sin(th1k)*sin(th2k)^2*sin(th3k)^2, -sin(th1k)*sin(th2k)^2*sin(th3k)*cos(th3k), -sin(th1k)*sin(th2k)*sin(th3k)*cos(th2k); cos(th1k)*sin(th2k)*cos(th3k), -sin(th1k)*sin(th2k)^2*sin(th3k)*cos(th3k), -sin(th1k)*sin(th2k)^2*cos(th3k)^2, -sin(th1k)*sin(th2k)*cos(th2k)*cos(th3k); cos(th1k)*cos(th2k), -sin(th1k)*sin(th2k)*sin(th3k)*cos(th2k), -sin(th1k)*sin(th2k)*cos(th2k)*cos(th3k), -sin(th1k)*cos(th2k)^2; -sin(th1k), -cos(th1k)*sin(th2k)*sin(th3k), -cos(th1k)*sin(th2k)*cos(th3k), -cos(th1k)*cos(th2k)];
    Vdot(:, :, 2, k) = [sin(th1k)*cos(th2k)*sin(th3k), -(1-cos(th1k))*(sin(2*th2k)*sin(th3k)^2), -(1-cos(th1k))*(sin(2*th2k)*sin(th3k)*cos(th3k)), -(1-cos(th1k))*(cos(2*th2k)*sin(th3k)); sin(th1k)*cos(th2k)*cos(th3k), -(1-cos(th1k))*(sin(2*th2k)*sin(th3k)*cos(th3k)), -(1-cos(th1k))*(sin(2*th2k)*cos(th3k)^2), -(1-cos(th1k))*(cos(2*th2k)*cos(th3k)); -sin(th1k)*sin(th2k), -(1-cos(th1k))*(cos(2*th2k)*sin(th3k)), -(1-cos(th1k))*(cos(2*th2k)*cos(th3k)), (1-cos(th1k))*(sin(2*th2k)); 0, -sin(th1k)*cos(th2k)*sin(th3k), -sin(th1k)*cos(th2k)*cos(th3k), sin(th1k)*sin(th2k)];
    Vdot(:, :, 3, k) = [sin(th1k)*sin(th2k)*cos(th3k), -(1-cos(th1k))*(sin(th2k)^2*sin(2*th3k)), -(1-cos(th1k))*(sin(th2k)^2*cos(2*th3k)), -(1-cos(th1k))*(sin(th2k)*cos(th3k)*cos(th2k)); -sin(th1k)*sin(th2k)*sin(th3k), -(1-cos(th1k))*(sin(th2k)^2*cos(2*th3k)), (1-cos(th1k))*(sin(th2k)^2*sin(2*th3k)), (1-cos(th1k))*(sin(th2k)*cos(th2k)*sin(th3k)); 0, -(1-cos(th1k))*(sin(th2k)*cos(th3k)*cos(th2k)), (1-cos(th1k))*(sin(th2k)*cos(th2k)*sin(th3k)), 0; 0, -sin(th1k)*sin(th2k)*cos(th3k), sin(th1k)*sin(th2k)*sin(th3k), 0];
  end
end
```

## oneDescent

```
% DESCRIPTION: Function that conducts one steepest descent search (from a given
% set of starting theta coordinate values) in L, the modified negative log
% likelihood for a 4 dimesional, cointegration rank 1 state space error
% correction model applied to a data set
%
% Input = th_start   (3x4 matrix of starting theta coordinates)
%         data       (matrix of data; observations in columns, last at right)
%         step_lim   (maximum number of steps to allow in the descent)
%
% Output = descinfo  (Matlab structure containing the following fields)
%            thetaStart (the starting theta coordinates for the descent)
%            thetaEnd   (the ending theta coordinates for the descent)
%            LStart     (the starting modified negative log likelihood)
%            LEnd       (the ending modified negative log likelihood
%            nSteps     (the number of steps taken in the descent)
%            totalDist  (the total length of all the descent steps taken)
%            displcmnt  (the Euclidean distance from thetaStart to thetaEnd)
%            endReason  (a string indicating why the descent stopped; possible
%                        values are 'critical point', 'theta boundary', 'no
%                        likelihood decrease', 'maximum number of steps', and
```

```
%                           'complex L at start')

function descinfo = oneDescent_orig(th_start, data, step_lim)
  n = 4;
  p = size(data, 1);  % dimension of the data series
  max_steps = step_lim;  % the largest number of steps to take in the descent

  % THESE VALUES CAN BE ADJUSTED
  grad_tol = 0.1;    % the tolerance for the norm of the gradient
  tau = 0.01;            % main step size, relative to the gradient
  tau_tol = 0.00001;     % smallest step size to consider (relative to gradient)

  % Initialize three logical values that, if true, will allow the current loop
  % to terminate: either a critical point is found (i.e. the size of the
  % gradient is small), the step size using tau has fallen below the
  % tolerance level, or the number of steps has reached the limit
  found_crit_pt = false;
  reached_tau_tol = false;
  reached_max_steps = false;

  % Initialize other values
  n_steps = 0;                % The number of steps taken in the descent
  total_dist = 0;             % The cumulative distance of all the steps

  % Initialize the current theta values and get the likelihood and gradient
  [Lik_start, grad_L_start] = getL(th_start, data);
  th_curr = th_start;
  Lik_curr = Lik_start;
  grad_L_curr = grad_L_start;

  % If the starting likelihood is a complex number, set values to exit
  % immediately (this can happen if Sigmahat has a negative determinant due
  % to rounding error)
  if ~isreal(Lik_start)
    bad_start = true;
    end_reason = 'complex L at start';
    Lik_curr = Inf;  % to avoid problems with likelihood comparisons
  else
    bad_start = false;
  end

  % do gradient descent until one of the conditions above is met
  while ~any([found_crit_pt, reached_tau_tol, reached_max_steps, bad_start])
    % First check if we're at a critical point, if so stop

    if (norm(grad_L_curr(:)) < grad_tol)
      found_crit_pt = true;
      end_reason = 'critical point';
      NORM=norm(grad_L_curr(:))
    % If not a critical point, start with our default step size (tau)
    else
```

```
      tau_curr = tau;
      th_provis = th_curr - tau_curr * grad_L_curr;
      % Check if the current step size will move past the boundary, and if so
      % reduce the step size by half (unless the step size has gone below
      % tau_tol)
      while pastbdy(th_provis) && (tau_curr > tau_tol)
        tau_curr = tau_curr / 2;
        th_provis = th_curr - tau_curr * grad_L_curr;
      end
      % If the step size has gone below tau_tol, stop
      if (tau_curr <= tau_tol)
        reached_tau_tol = true;
        end_reason = 'theta boundary';
      % Otherwise begin checking the likelihood of the provisional theta
      % value
      else
        info_provis = getStateInfo(th_provis, data);
        Lik_provis = info_provis.L;
        % Check if the provisional likelihood is >= the current likelihood or
        % complex (being complex means that Sigmahat had a negative determinant
        % due to rounding error); if so reduce the step size by half (unless
        % the step size has gone below tau_tol)
        while (Lik_provis >= Lik_curr || ~isreal(Lik_provis)) && (tau_curr > tau_tol)
          tau_curr = tau_curr / 2;
          th_provis = th_curr - tau_curr * grad_L_curr;
          % Repeat the boundary check, reducing the step size if necessary
          % and changing the value of reached_tau_tol if appropriate
          while pastbdy(th_provis) && (tau_curr > tau_tol)
            tau_curr = tau_curr / 2;
            th_provis = th_curr - tau_curr * grad_L_curr;
          end
          if (tau_curr <= tau_tol)
            reached_tau_tol = true;
            end_reason = 'theta boundary';
          end
          info_provis = getStateInfo(th_provis, data);
          Lik_provis = info_provis.L;
        end
        % If the step size has gone below tau_tol, stop
        if (tau_curr <= tau_tol)
          reached_tau_tol = true;
          end_reason = 'no likelihood decrease';
        % Otherwise, we've found an improvement on the current likelihood
        else
          % Update the current values of theta, likelihood, and gradient
          [Lik_provis, grad_L_provis] = getL(th_provis, data);
          th_curr = th_provis;
          Lik_curr = Lik_provis;
          grad_L_curr = grad_L_provis;
          % Update other values
```

```
            n_steps = n_steps + 1;
            total_dist = total_dist + tau_curr * norm(grad_L_curr(:));
            if (n_steps >= max_steps)
              reached_max_steps = true;
              end_reason = 'maximum number of steps';
            end
          end
        end
      end
    end

    % Gradient descent is now complete (from the given starting point)

    % Calculate the displacement
    displc_mtx = th_curr - th_start;
    displacement = norm(displc_mtx(:));

    % Create the return structure
    descinfo = struct('thetaStart', th_start, 'thetaEnd', th_curr, 'LStart',
        Lik_start, 'LEnd', Lik_curr, 'nSteps', n_steps, 'totalDist',
        total_dist, 'displcmnt', displacement, 'endReason', end_reason);
end
```

## pastbdy

```
% this function checks whether a matrix of theta values is outside of the
    domain
% under consideration (the domain being restricted to all theta values
% strictly between -pi/2 and pi/2)
function ispast = pastbdy(thmtx)
  ispast = any(any(thmtx >= pi/2 | thmtx <= -pi/2));
end
```

## R2rmse

```
function [r2, rmse] = R2rmse(y,f,varargin)
% Compute coefficient of determination of data fit model and RMSE
%
% [r2 rmse] = rsquare(y,f)
% [r2 rmse] = rsquare(y,f,c)
%
% RSQUARE computes the coefficient of determination (R-square) value from
% actual data Y and model data F. The code uses a general version of
% R-square, based on comparing the variability of the estimation errors
% with the variability of the original values. RSQUARE also outputs the
% root mean squared error (RMSE) for the user's convenience.
%
% Note: RSQUARE ignores comparisons involving NaN values.
%
% INPUTS
%   Y       : Actual data
%   F       : Model fit
%
```

```
% OPTION
%   C        : Constant term in model
%              R-square may be a questionable measure of fit when no
%              constant term is included in the model.
%   [DEFAULT] TRUE : Use traditional R-square computation
%             FALSE : Uses alternate R-square computation for model
%                     without constant term [R2 = 1 - NORM(Y-F)/NORM(Y)]
%
% OUTPUT
%   R2       : Coefficient of determination
%   RMSE     : Root mean squared error
%
% EXAMPLE
%   x = 0:0.1:10;
%   y = 2.*x + 1 + randn(size(x));
%   p = polyfit(x,y,1);
%   f = polyval(p,x);
%   [r2 rmse] = rsquare(y,f);
%   figure; plot(x,y,'b-');
%   hold on; plot(x,f,'r-');
%   title(strcat(['R2 = ' num2str(r2) '; RMSE = ' num2str(rmse)]))
%
% Jered R Wells
% 11/17/11
% jered [dot] wells [at] duke [dot] edu
%
% v1.2 (02/14/2012)
%
% Thanks to John D'Errico for useful comments and insight which has helped
% to improve this code. His code POLYFITN was consulted in the inclusion of
% the C-option (REF. File ID: #34765).

if isempty(varargin); c = true;
elseif length(varargin)>1; error 'Too many input arguments';
elseif ~islogical(varargin{1}); error 'C must be logical (TRUE||FALSE)'
else c = varargin{1};
end

% Compare inputs
if ~all(size(y)==size(f)); error 'Y and F must be the same size'; end

% Check for NaN
tmp = ~or(isnan(y),isnan(f));
y = y(tmp);
f = f(tmp);

if c; r2 = max(0,1 - sum((y(:)-f(:)).^2)/sum((y(:)-mean(y(:))).^2));
else r2 = 1 - sum((y(:)-f(:)).^2)/sum((y(:)).^2);
    if r2<0
    % http://web.maths.unsw.edu.au/~adelle/Garvan/Assays/GoodnessOfFit.html
        warning('Consider adding a constant term to your model') %#ok<WNTAG>
        r2 = 0;
    end
end
```

```

rmse = sqrt(mean((y(:) - f(:)).^2));
```

## run_compare

```
% ////////////////////////////////
% Mia's code: Comparing models
% v.17.09.17
% ////////////////////////////////
% This code:
%    (0) Estimates de SSEC and VEC models
%    (I) Compares the models with information criterion
%   (II) Compares the models with an approximation to Bayes factors
%  (III) Compares the models with goodness-of-fit measures
% ////////////////////////////////////////////////////////////////////
clear; clc %<<<<<< careful here, it will erase previous work
%% (0) Model estimation
% (1) Run state-space error-correction model
run gener_state_main
    % state-space stats:
    LogL_ssm = mdl.logLik; % Log-likelihood
       n_ssm = size(data, 2) - 1; % sample-size
       k_ssm = mdl.TotalParameter; % number of parameters
    % Information criterion for SS model:
    BIC_ssm = k_ssm*log(n_ssm) - 2*(LogL_ssm);
    AIC_ssm = 2*k_ssm - 2*(LogL_ssm);
    HQC_ssm = 2*k_ssm*log(log(n_ssm)) - 2*(LogL_ssm);
            IC_ssm = [BIC_ssm AIC_ssm HQC_ssm];
% (2) Run VEC model (for gener data)
run vec_main
    % VEC stats:
    [LogL_vec,n_vec,k_vec] = vecm_stats(gener_data, gener_mles,
        gener_coint_rank);
    % Information criterion for VEC model:
    BIC_vec = k_vec*log(n_vec) - 2*(LogL_vec);
    AIC_vec = 2*k_vec - 2*(LogL_vec);
    HQC_vec = 2*k_vec*log(log(n_vec)) - 2*(LogL_vec);
            IC_vec = [BIC_vec AIC_vec HQC_vec];
% ------------------------------------------------------------------------
%% (I)  Model comparison based on information criterion
    % The "best" model is the one which minimizes the AIC/BIC/HQC
    b = bar(categorical({'BIC','AIC','HQC'}),[IC_ssm' IC_vec']);
    ylim([1.66e+04 1.72e+04])
    b(1).FaceColor = [0 .5 .8];
    b(2).FaceColor = [0 .8 .5];
    legend('State-space EC model','VEC model','Location','bestoutside')
    % IC Table
    disp('--- IC for SSEC model ---')
        disp([' AIC = ',mat2str(AIC_ssm)])
        disp([' BIC = ',mat2str(BIC_ssm)])
        disp([' HQC = ',mat2str(HQC_ssm)])
    disp('--- IC for VEC model ---')
        disp([' AIC = ',mat2str(AIC_vec)])
        disp([' BIC = ',mat2str(BIC_vec)])
```

```
        disp(['  HQC =  ',mat2str(HQC_vec)])
    disp('AIC: Akaike IC, BIC: Bayesian IC, HQC: Hannan-Quinn IC')
%---------------------------------------------------------------------------
%% (II) Model comparision based on an approximation to Bayes factors
    BIC_M0 = BIC_ssm; % Base model
    BIC_M1 = BIC_vec; %
    Delta_BIC = (BIC_M1 - BIC_M0);
BF = exp(Delta_BIC/2);
PM = (1 + exp((-1/2)*Delta_BIC))^(-1);
disp([' Bayes factor for M0: ',mat2str(round(BF,2))])
disp([' Probability in favor of M0: ',mat2str(round(PM,2))])
%---------------------------------------------------------------------------
%% (III) Model comparison based on measures of goodness of fit
        y = data;
    % VEC fit:
    yfit_vec = VEC_fit(gener_data, gener_mles, gener_coint_rank,y);
    % Graph:
        figure
        subplot(3,1,1)
        plot(y(1,:),'b:'); hold on; plot(yfit_vec(1,:),'r:'); hold off;
            legend('Observed','Fitted','Location','bestoutside'); title('VEC
             fit: y_{1,t}')
        subplot(3,1,2)
        plot(y(2,:),'b:'); hold on; plot(yfit_vec(2,:),'r:'); hold off;
            legend('Observed','Fitted','Location','bestoutside'); title('VEC
             fit: y_{2,t}')
        subplot(3,1,3)
        plot(y(3,:),'b:'); hold on; plot(yfit_vec(3,:),'r:'); hold off;
            legend('Observed','Fitted','Location','bestoutside'); title('VEC
             fit: y_{3,t}')
    [r2_vec1, rmse_vec1] = R2rmse(y(1,1:end-7),yfit_vec(1,:),true(0));
    [r2_vec2, rmse_vec2] = R2rmse(y(2,1:end-7),yfit_vec(2,:),true(0));
    [r2_vec3, rmse_vec3] = R2rmse(y(3,1:end-7),yfit_vec(3,:),true(0));
    % SSEC fit
    run SSEC_fit
    % Graph:
        figure
        subplot(3,1,1)
        plot(y(1,:),'b:'); hold on; plot(yfit_ssec(1,:),'r:'); hold off;
            legend('Observed','Fitted','Location','bestoutside'); title('
            SSEC fit: y_{1,t}')
        subplot(3,1,2)
        plot(y(2,:),'b:'); hold on; plot(yfit_ssec(2,:),'r:'); hold off;
            legend('Observed','Fitted','Location','bestoutside'); title('
            SSEC fit: y_{2,t}')
        subplot(3,1,3)
        plot(y(3,:),'b:'); hold on; plot(yfit_ssec(3,:),'r:'); hold off;
            legend('Observed','Fitted','Location','bestoutside'); title('
            SSEC fit: y_{3,t}')
    [r2_ssec1, rmse_ssec1] = R2rmse(y(1,1:end-2),yfit_ssec(1,:),true(0));
    [r2_ssec2, rmse_ssec2] = R2rmse(y(2,1:end-2),yfit_ssec(2,:),true(0));
    [r2_ssec3, rmse_ssec3] = R2rmse(y(3,1:end-2),yfit_ssec(3,:),true(0));
    % Table with results
    disp('Ordinary R2 fit:')
```

```
    disp('          SSEC       VEC')
    disp(['y_{1,t}   ', mat2str(round(r2_ssec1,4)),'     ', mat2str(round(
        r2_vec1,4))])
    disp(['y_{2,t}   ', mat2str(round(r2_ssec2,4)),'     ', mat2str(round(
        r2_vec2,4))])
    disp(['y_{3,t}   ', mat2str(round(r2_ssec3,4)),'     ', mat2str(round(
        r2_vec3,4))])
    disp('RMSE:')
    disp('          SSEC       VEC')
    disp(['y_{1,t}   ', mat2str(round(rmse_ssec1,3)),'     ', mat2str(round(
        rmse_vec1,3))])
    disp(['y_{2,t}   ', mat2str(round(rmse_ssec2,3)),'     ', mat2str(round(
        rmse_vec2,3))])
    disp(['y_{3,t}   ', mat2str(round(rmse_ssec3,3)),'     ', mat2str(round(
        rmse_vec3,3))])
 % ////////////////////////////////////////////////////////////////////////
```

## unDiff

```
function y = unDiff(dy,y0)
y(1) = y0 + dy(1);
for t = 2:(length(dy)-1)
       x = y(t-1);
    y(t) = x + dy(t);
end
```

## varLagSelect

```
% DESCRIPTION: A function that prints lag order selection information to a
    file
%
% Input = data        (matrix of data)
%         fileout     (what to name the output file)
%
% Output = lagOrder   (the BIC-minimizing VAR lag order)

function lagOrder = varLagSelect(data, fileout)
  nobs = size(data, 1);
  nvars = size(data, 2);
  ndata = nobs * nvars;

  maxLag = 12;  % Only consider lags of up to 12; this can be adjusted
  warning('error', 'MATLAB:nearlySingularMatrix');  % change warning to
      handle

  % First fit VAR models only if the estimation doesn't result in a warning
      of
  % a nearly singular matrix (which could indicate an unreliable result)
  varLagBics = zeros(1, maxLag);
  for j = 1:maxLag
    varmodel = varm(nvars, j);
    varmodel.Constant = zeros(nvars, 1);  % no intercept term in model
    try
```

```
        estmodel = estimate(varmodel, data);
        results = summarize(estmodel);
        varLagBics(j) = results.BIC;
      catch ME
        if strcmp(ME.identifier, 'MATLAB:nearlySingularMatrix')
          varLagBics(j) = NaN;
        else
          rethrow(ME);
        end
      end
    end

    fileID = fopen(fileout, 'w');
    % If all the previous attempts resulted in a nearly singular matrix warning,
    % go ahead and fit all of them anyway, but issue a warning in the output file
    if all(isnan(varLagBics))
      fprintf(fileID, 'WARNING: All fitted models had calculations with nearly singular matrices.\n');
      fprintf(fileID, 'Results may be unreliable.\n\n\n\n');
      warning('off', 'MATLAB:nearlySingularMatrix');
      for j = 1:maxLag
        varmodel = varm(nvars, j);
        varmodel.Constant = zeros(nvars, 1);  % no intercept term in model
        estmodel = estimate(varmodel, data);
        results = summarize(estmodel);
        varLagBics(j) = results.BIC;
      end
    end

    minVarBic = min(varLagBics);
    lagOrder = find(varLagBics == minVarBic, 1);  % pick first minimum BIC

    % Print information to file
    fprintf(fileID, 'LAG ORDER SELECTION RESULTS\n');
    fprintf(fileID, '------------------------------------------------------------\n');
    fprintf(fileID, '   Lag order          Model BIC\n');
    for j = 1:maxLag
      fprintf(fileID, '%1$9u %2$19.2f\n', j, varLagBics(j));
    end
    fprintf(fileID, '\n\nA VAR model lag order of %u is recommended. This is equivalent to a VEC model\n', lagOrder);
    fprintf(fileID, 'lag order of %u.', lagOrder-1);
    fclose(fileID);
end
```

## VEC_fit

```
function yfit_vec = VEC_fit(gener_data, gener_mles, gener_coint_rank,y)
    dy1_t = diff(y(1,:));
    dy2_t = diff(y(2,:));
    dy3_t = diff(y(3,:));
```

```
    dy = [dy1_t; dy2_t; dy3_t];
VECfit = vecm_param(gener_data, gener_mles, gener_coint_rank);
    % Fitted parameters:
    C = VECfit.paramVals.A * VECfit.paramVals.B';
    B1 = VECfit.paramVals.B1;
    B2 = VECfit.paramVals.B2;
    B3 = VECfit.paramVals.B3;
    B4 = VECfit.paramVals.B4;
    B5 = VECfit.paramVals.B5;
    % Fitted model:
dyfit_vec =   C*y(:,2:end-5) + ...
            B1*dy(:,2:end-4) + ...
            B2*dy(:,3:end-3) + ...
            B3*dy(:,4:end-2) + ...
            B4*dy(:,5:end-1) + ...
            B5*dy(:,6:end);
    %
 yfit_vec(1,:) = unDiff(dyfit_vec(1,:),y(1,1));
 yfit_vec(2,:) = unDiff(dyfit_vec(2,:),y(2,1));
 yfit_vec(3,:) = unDiff(dyfit_vec(3,:),y(3,1));
```

## vec_main

```
% Import data

gener_data = csvread('gener_data_2000.csv');

% Produce output for generated data
gener_var_lag = varLagSelect(gener_data, 'gener_lag_output.txt');
gener_vec_lag = gener_var_lag - 1;
gener_mles = cointRank(gener_data, gener_vec_lag, 'gener_coint_output.txt');
gener_coint_rank = 1;  % based on information in gener_coint_output.txt file
vecmSummary(gener_data, gener_mles, gener_coint_rank, 'gener_vecm_output.txt');
```

## vecm_param

```
function VECmod = vecm_param(data, mles, cirank)
    nobs = size(data, 1);
    nvars = size(data, 2);
    mdlInfo = mles.(cirank+1);  % select the model with the right coint. rank
        VECmod = mdlInfo;
    nlags = length(mdlInfo.paramNames) - 2;  % this assumes the H2 model type

  % Organize the parameter estimates in order to print
  C = mdlInfo.paramVals.A * mdlInfo.paramVals.B';  % alphabetaprime term
  prNames = mdlInfo.paramNames;
  prVals = struct2cell(mdlInfo.paramVals);
  paramNames = ['C'; prNames(3:nlags+2); 'Q'];  % assumes H2 again
  paramVals = [{C}; prVals(3:nlags+2); {mdlInfo.EstCov}];
end
```

## vecm_stats

```
function [logLik ,n_vecm ,totalParams] = vecm_stats(data , mles , cirank)
    nobs = size(data , 1);
    nvars = size(data , 2);
    mdlInfo = mles.(cirank+1);  % select the model with the right coint.
        rank
    nlags = length(mdlInfo.paramNames) − 2;  % this assumes the H2 model
        type
  effecSampSize = nobs − nlags − 1;

  % Get the log likelihood and BIC
  logLik = mdlInfo.rLL;
  cointParams = 2 * nvars * cirank − cirank^2;
  lagParams = nlags * nvars^2;
  covarParams = nvars * (nvars + 1) / 2;
  totalParams = cointParams + lagParams + covarParams;
  n_vecm = effecSampSize*nvars;
  [aic , bic] = aicbic(logLik , totalParams , effecSampSize*nvars);

  % Organize the parameter estimates in order to print
  C = mdlInfo.paramVals.A * mdlInfo.paramVals.B';  % alphabetaprime term
  prNames = mdlInfo.paramNames;
  prVals = struct2cell(mdlInfo.paramVals);
  paramNames = ['C'; prNames(3:nlags+2); 'Q'];  % assumes H2 again
  paramVals = [{C}; prVals(3:nlags+2); {mdlInfo.EstCov}];
end
```

## vecmSummary

```
% DESCRIPTION: A function that prints a summary of the fitted vector error
% correction model, including estimated parameters, to a file
%
% Input = data      (matrix of data)
%         mles      (cell array of structs with fitted VEC model information)
%         cirank    (the cointegration rank to use for the model)
%         fileout   (what to name the output file)
%
% Output = none

function vecSummary(data , mles , cirank , fileout)
  nobs = size(data , 1);
  nvars = size(data , 2);
  mdlInfo = mles.(cirank+1);  % select the model with the right coint. rank
  nlags = length(mdlInfo.paramNames) − 2;  % this assumes the H2 model type
  effecSampSize = nobs − nlags − 1;

  % Get the log likelihood and BIC
  logLik = mdlInfo.rLL;
  cointParams = 2 * nvars * cirank − cirank^2;
  lagParams = nlags * nvars^2;
  covarParams = nvars * (nvars + 1) / 2;
  totalParams = cointParams + lagParams + covarParams;
```

```
  [aic, bic] = aicbic(logLik, totalParams, effecSampSize*nvars);

  % Organize the parameter estimates in order to print
  C = mdlInfo.paramVals.A * mdlInfo.paramVals.B';  % alphabetaprime term
  betaprime = mdlInfo.paramVals.B'; %betaprime cointegration vector
  prNames = mdlInfo.paramNames;
  prVals = struct2cell(mdlInfo.paramVals);
  paramNames = ['C'; prNames(3:nlags+2); 'Q'];  % assumes H2 again
  paramVals = [{C}; prVals(3:nlags+2); {mdlInfo.EstCov}];

  % Print model summary to file
  fileID = fopen(fileout, 'w');
  fprintf(fileID, 'VECTOR ERROR CORRECTION MODEL SUMMARY\n');
  fprintf(fileID, '
      ------------------------------------------------------------------------\n');
  fprintf(fileID, ' Model eq: dy_t = C*y_t-1');
  for i = 1:nlags
    fprintf(fileID, ' + B%u*dy_t-%u', i, i);
  end
  fprintf(fileID, ' + e_t\n');
  fprintf(fileID, '            E[e_t*e_t''] = Q (innovations covariance)\n');
  fprintf(fileID, ' Effective sample size: %u\n', effecSampSize);
  fprintf(fileID, ' Lag order: %u\n', nlags);
  fprintf(fileID, ' Cointegration rank: %u\n', cirank);
  fprintf(fileID, ' Log likelihood: %5.2f\n', logLik);
  fprintf(fileID, ' BIC: %4.2f\n', bic);
  fprintf(fileID, ' Total Parameter: %4.2f\n', totalParams);
  fprintf(fileID, ' Betaprime: %4.2f\n', betaprime);
  fprintf(fileID, '\n\n  *Parameter estimates*\n\n');
  for i = 1:length(paramNames)
    fprintf(fileID, [formatMtxOut(paramNames{i}, paramVals{i}) '\n\n']);
  end
  fclose(fileID);
end


% DESCRIPTION: A function that reformats how a matrix is displayed/printed
%
% Input = name    (the name or symbol to use for the matrix when printing)
%         value   (the matrix itself)
%
% Output = mtxString (string containing the reformatted matrix for printing)

function mtxString = formatMtxOut(name, value)
  startFormat = get(0, 'Format');  % find out Matlab's current display
      setting
  nrows = size(value, 1);
  padLength = length(name) + 5;

  format shortG;  % change Matlab's display setting
  T = evalc('value');  % get displayed matrix as a character vector
  newlineIxs = find(T == repmat(char(10), 1, length(T)));
  mtxRows = {};
  % Convert the rows of the matrix, as displayed, to a cell array
```

```
  for i = 1:nrows
    mtxRows = [mtxRows, T(newlineIxs(i+2)+1:newlineIxs(i+3))];
  end
  % Pad the displayed rows, including the matrix name with the first row
  mtxRows{1} = ['    ', name, ' =', mtxRows{1}];
  mtxString = mtxRows{1};
  for i = 2:nrows
    mtxRows{i} = [repmat(' ', 1, padLength), mtxRows{i}];
    mtxString = [mtxString mtxRows{i}];
  end

  format(startFormat); % return display setting to starting value
end
```

Appendix C