LUISS Guido Carli
LIBERA UNIVERSITÀ INTERNAZIONALE DEGLI STUDI SOCIALI

Department of Economics and Finance
Course of Study in Banking and Financial Institutions

Econometrics for Finance

TWO NEW TESTS FOR THE NUMBER OF COMMON DYNAMIC FACTORS

SUPERVISOR
Prof. Federico Carlini

CANDIDATE
Antonio Maria Biasco
735851

CO-SUPERVISOR
Prof. Porchia Paolo

Academic Year 2021/2022

Two New Tests for the Number of Common Dynamic Factors

Antonio Maria Biasco

July 2022

Table of contents

Preface

> If you hit a wrong note, then
> make it right by what you play
> afterwards
>
> *Joe Pass*

Following closely the research of Professors Federico Carlini (LUISS Guido Carli, Rome, Italy) and Mirco Rubin (EDHEC Business School, Nice, France) I had the opportunity to discover and understand a new working method that will lead to determine completely new academic results in the econometric field. This dissertation is basically a paper based on what I have seen develop over the past six months in which I am extremely proud to have participated and contributed through small tasks and further research with the purpose of offering a simpler approach to this topic, also for all those who believe they do not have sufficient skills to approach the world of econometrics. That's why to make things as simple and complete as possible for the reader, inside the paper there will be presented completely innovative codes written by the Professors themselves and by the undersigned, which refer exclusively to the programming through the Matlab software. This choice has been adopted in order to make the reader participate in what has been discovered, also through his own personal exercise.

The work is structured to retrace in a sequential manner all the stages that I have followed to get first to understand the work that was being done and then participate actively trying to give my personal contribution. Each step has been illustrated with the aim of making the paper as readable as possible, starting from the first sections of review and introduction to the topics, until we get to the last, more technical and complex, but not for this reason more difficult to understand (provided that one has followed carefully the discussion and has made the appropriate personal insights). In this regard, for all those who want to learn more about this topic any additional information can be further explored by comparing Carlini and Rubin, *New Tests and Estimators for Common Dynamic Factors* (2022)[1]. Reading the latter is strongly recommended, since the birth of this dissertation is closely related to the writing of the paper itself.

Finally, a special thanks goes to the Professor of my Econometrics for Finance course and Supervisor, Federico Carlini for having involved me in such a new and stimulating project and for the availability he always showed me in clarifying all my doubts. This project seemed to me to start under a certain initial skepticism mainly due to the fear of not feeling up to such a specific task. However, the patience and attention I was given were priceless, as well as what encouraged me not to give up in the most difficult moments and to continue on this challenging path.

[1] At the time of this publication (July 2022) the paper itself may not yet have been published.

Preliminary concepts

An initial overview of the world of econometrics cannot be separated from an introduction to fundamental statistical concepts such as inference. Statistical inference is a process of data analysis widely used for estimating a parameter or determining whether a claim about a population has been made. Indeed, beyond existing data, this procedure uses any sample of data to determine the parameters' value and it also avoids observing the parameters themselves, while at the same time being influenced by chance.
Hence, the main aim of statistical inference is to learn population features from samples; population features are represented by parameters, while sample features are represented by statistics.
For the study of the characteristics of the above samples, we attempt to model and reduce dispersive sets of observations into synthetic mathematical models that can approximate trends and variations in an immediate way: in this case these are called statistical models and they provide representation of complex phenomena that generate data.
The main features of a statistical model are:

- it contains mathematical formulas that describe the relationship between random variables and parameters.
- it is built making assumptions about random variables and sometimes parameters.
- his general form is made as follows: $dependentvariable = model + residuals$.
- dependent variable should account for most of the variability in the data.
- residuals represent lack-of-fit, or rather parts of the data that are not explained by the model.

Inferential statistical analysis derives properties of a population by hypotheses testing and deriving estimates. Estimation represents the manner or process by which a model learns and determines the value (or range of potential values) of a population parameter based on fitted data. As we said, statistical inference also involves hypothesis testing, which may be described as assessing an idea about a population based on a sample.
Confidence intervals around the mean have important practical implications because they convey the meaning of the results and generally require assumptions about the nature of the population. In addition, hypothesis testing allows to qualify decisions with a certain degree of confidence, hereafter, it is important to know the shape of the probability distribution from which the samples were drawn.
Subsequently, in the course of this analysis we will deal with a large panel of observations, which are data that contain observations across different cross sections over time. Panel data retain some of the typical properties of time series data and cross-sectional data: like time series data, panel data consist of observations collected periodically in chronological order, like cross-sectional data, panel data contain observations on a group of people. This is the reason why panel data contain more information, more variability, and higher

efficiency[2] than pure time series or cross-sectional data. Indeed, panels can also minimize estimation bias caused by aggregating groups into a single time series. Consequently, it can be stated that panel data are a collection of quantities collected by multiple people, compiled at uniform time intervals, and organized chronologically. Examples of panels include individuals, countries, and companies, as well as investment portfolios and global market indices. To represent both individual and temporal observations, panel data typically refers to groups with index i and time with index t. For example, a panel data observation y_{it} relates to all individuals $i = 1, \ldots, N$ across all time periods $t = 1, \ldots, T$.

Considering what has just been said, another crucial aspect we will be dealing with over the entire research is that it is usually difficult and time-consuming to work with huge data sets, so scientists had to develop a solution that would allow them to work faster as long as the same characteristics of the initial dataset were maintained. Principal Component Analysis (PCA) is the process of computing principal components and using them to make changes to the data base. Principal components are defined as the linear combinations (among all linear combinations) of the x variables with the greatest variance. They take into account the maximum possible variance in the data, making PCA widely used in business and science for exploratory data analysis and building predictive models. However, the fundamental concept is that the primary use of PCA is for dimensionality reduction, by projecting each data point onto only the first few principal components to obtain low-dimensional data while preserving as much variation in the data as possible. In this regard, PCA aims to find the directions of maximum variance in high-dimensional data and project them onto a new subspace which has less dimensions than the original subspace. This is why principal components may be identified as the orthogonal axes of the new subspace and they can be interpreted as directions of maximum variance. To simplify, this technique can help provide a low-dimensional picture of the raw data: using PCA, complex and messy datasets can be reduced to a simplified set of useful information with minimal effort. This approach (often used in conjunction with other Maximum Likelihood Estimation (MLE)[3] models) has helped data scientists achieve impressive results in time series forecasting, data compression, and data visualization tasks. In finance, it is commonly used for financial risk analysis, exploration and dynamic trading strategies, statistical arbitrage, and stock price forecasting. Generally, forecasting stock prices is based on the assumption that market patterns repeat over time and that prices are always associated with a number of macroeconomic and fundamental variables, such as book-to-market ratios and earnings yields, which may be used for forecasting purposes. There is also extensive research showing that price forecasts can be generated from historical price/return data, and this particular research introduces a robust method for predicting stock price values based on covariance information.

Typically, MLE experts use traditional ML estimators to approximate the covariance matrix, or in some special cases, empirical covariances. However, both methods fail when the dimensionality of the matrix is

[2] Efficiency is defined as the ratio of the theoretical minimum variance of the estimator to the actual variance. The metric ranges from 0 to 1. An estimator with an efficiency of 1 is called an "efficient estimator". The efficiency of a given estimator depends on the population.

[3] Method of estimating the parameters of an assumed probability distribution, given some observed data. This is achieved by maximizing a likelihood function so that, under the assumed statistical model, the observed data is most probable.

huge compared to the number of variables. It can be shown how the principal components are nothing but the eigenvectors of the variance-covariance matrix that reports the data. Therefore, the principal components are often computed by eigendecomposition of the data covariance matrix or single value decomposition of the data matrix. PCA is the simplest type of true eigenvector-based multivariate analysis, and is related to factor analysis. Factor analysis typically involves more domain-specific assumptions about the underlying structure, and solves for the eigenvectors of a slightly different matrix. To make it clearer, PCA can be thought of as fitting a p-dimensional ellipsoid to the data, with each axis of the ellipsoid representing a principal component. If one axis of the ellipsoid is small, then the variance along that axis is also small. To find the axes of the ellipsoid, we need first center the values of each variable in the dataset to zero by subtracting the mean of the variable's observations from each value.

To summarize, the above is only a brief introduction to the topics that will be covered in this thesis. In a sense, they cover a wide range of useful tools for understanding increasingly complex topics. In this regard, the approach taken in dealing with each aspect is to proceed step by step, trying to make sure that each new topic is well understood before moving on to the next.

1 Introduction

After this general overview describing the main terms and topics that will be encountered along this research, the final goal of the work will be to propose two new sequential testing procedures for the number q of dynamic factors in a large dimensional dynamic factor models, as well as to present a series of simulations that can provide further empirical evidence regarding the results obtained. The testing procedures are based on two new tests for the rank of the residual covariance matrix of the VAR model estimated on principal component estimators of r static factors obtained from a large panel of observations from the dynamic factor model. The rank of the VAR residuals' covariance matrix is tested by deriving the asymptotically Gaussian distributions of the sum of (i) its smallest $r - q$ eigenvalues, and (ii) the largest $r - q$ canonical correlations between the estimated factors and their lagged values. The canonical directions associated to the $r - q$ smallest canonical correlations allow us to define an easily implementable estimator of the common dynamic factors themselves, and to derive its asymptotic properties. We will see that the asymptotic results hold for relative convergence rates of N, T more general than those required by Onatski (2009). These tests provide solutions to two unsolved problems mentioned in Bai and Ng (2007), in particular they are two examples of tests of rank for small positive semidefinite matrices[4], for which no theory has been developed prior to the paper written by Carlini and Rubin (2022).

Since the paper already provides the answers to the questions raised by Bai and Ng, this dissertation will have the task of offering a more scholastic and less technical vision of the whole procedure, referring for further details to the paper itself, as well as offer, as further support, a series of simulations through which the test will be ultimately implemented, to support the correctness and adequacy.

In this regard, in Section 2 we begin to look at how the model should be constructed, taking up a framework previously developed by Bai and Ng (2002). As we previously said, a number of these researches highlighted a persistent identification problem within their models, which is described and analyzed through the aid of applying the same directions provided within the paper, implemented in Matlab codes. Solutions to these problems are presented within Section 3, where the test developed by Carlini and Rubin is reported, in addition to which simulation results for different types of datasets are reported within Section 4. Finally, conclusions will be drawn in Section 5.

[4] Matrices where all eigenvalues are non-negative

2 Building the model

2.1 Data generating process

A Data Generating Process (DGP) is a conceptual framework that describes the steps involved in generating an outcome. This section aims to explain the concepts of statistical inference, to provide a general overview on what a DGP actually is and to describe how our model is going to be made.

In statistics and empirical science, a Data generating process is the actual process of "generating" data of interest. When it comes to research, it is often the case that scientists cannot rely on pre-specified data models, neither they don't know the real data-generating model. However, these real models are believed to have observable consequences. These series are the population distribution of the data, so the aim would be to try to give them a representation using mathematical functions. The model that is going to be considered in this analysis is analogous to the Data generating process considered by Amengual and Watson (2007) and Bai and Ng (2007). We now consider an equivalent method for expressing the same DGP that simplifies the derivation of the static test distribution for the number q of common dynamic shocks. Here the model for generating our dataset will be described as follows by the equations. These consist in a two-step approach, where thanks to the dynamic factor f_t estimate by a vector autoregressive process (VAR(1)), we are allowed to determine the shape of the static factor model y_t.

$$y_t = \breve{\Lambda}\breve{f}_t + \varepsilon_t \tag{2.1}$$

$$\text{where} \qquad \varepsilon_t \sim iid(0, I_q)$$

$y_t = [y_{1t}, \ldots, y_{Nt}]'$ is the N-dimensional vector collecting the observations for N individuals, $\breve{f}_t$ is the r-dimensional vector of latent static factors, which can be estimated by classical Principal Component Analysis (PCA), with r finite, positive and possibly small ($r = 2$ in this case), such that $r \ll N, T$ and $\varepsilon_t = [\varepsilon_{1t}, \ldots, \varepsilon_{Nt}]'$ is the N-dimensional vector of weakly correlated error terms, with $t = 0, 1, \ldots, T$.

$$\breve{f}_t = \breve{\Phi}\breve{f}_{t-1} + \breve{v}_t \tag{2.2}$$

$$\text{where } \breve{v}_t \sim iid(0, \breve{\Sigma}_v) \qquad\qquad \breve{\Sigma}_v \geq 0$$

It is also important to take into account $\breve{\Sigma}_v$ as a semi-definite positive variance-covariance matrix of the estimator and $\breve{\Phi}$ as a pre-specified $r \times r$ matrix.

As discussed earlier, the model was constructed by only considering one delay as suggested by the definition of a VAR(1), which means that each endogenous variable is explained by its lagged, or past, value and, in case of higher order VAR, by the lagged values of all other endogenous variables in the model. VAR models generalize the single variable (univariate) autoregressive model by allowing multivariate time series. They are characterized by their order, which refers to the number of previous time periods that the model will use.

In the case under analysis, since we are dealing with a VAR(1), the VAR would model the output of each time instant as a linear combination of the last observation of the input factor. Thus, in general, a p-order VAR refers to a VAR model that includes delays for the last p time periods. A pth-order VAR is referred to as "VAR(p)" and sometimes called "a VAR with p lags". Hence, the same process can be replicated by including a finite p number of lags $\breve{f}_t$ into the dynamic factor model, so as to obtain a VAR(p) process.
Moreover, $\breve{\Lambda} = [\breve{\lambda}_1, \ldots, \breve{\lambda}_N]'$ is the $N \times r$ matrix of factor loadings and the innovations $\breve{v}_t$ can be represented as $\breve{v}_t = G * \eta_t$. G is an $r \times q$ full column rank matrix, with $1 < q \leq r$ and the q-dimensional vector of "dynamic factors shocks" or "primitive shocks" η_t is such that $\eta_t \sim iid(0, I_q)$. When $q < r$, the errors' $r \times r$ covariance matrix $\breve{\Sigma}_v := E(\breve{v}_t \breve{v}_t') = G I_q G' = GG'$ has reduced to rank q.
A final noteworthy aspect is that $\breve{\Lambda}$ should be taken as a random matrix although it is going to be kept fixed in the following simulations.

Here follows the MATLAB code:

```
N=100;
r=2;
T=1000;

Phi=[.5,-.2;.1,.3];
v=randn(r,T);
Lambda=randn(N,r);
e=randn(N,T);

F=zeros(r,T);

for t=2:T;

    F(:,t)=Phi*F(:,t-1)+v(:,t);
    y(:,t)=Lambda*F(:,t)+e(:,t);

end
```

As a result, what we obtain by running these lines of code is:

$$\underset{(N \times T)}{y_t} = \underset{(N \times r)}{\breve{\Lambda}} \times \underset{(r \times T)}{\breve{f}_t} + \underset{(N \times T)}{\varepsilon_t}$$

Where, transposing this final collected outcome y_t', the definitive dataset y_t results to be written in the typical $(T \times N)$ form. This model is equivalent to a restricted version of the *Generalized dynamic factor*

model with q "dynamic factors" introduced in the seminal work of Forni, Hallin, Lippi, and Reichlin (2000), and is analogous to the model considered by Amengual and Watson (2007) and Bai and Ng (2007).

Once we understand how to obtain the previous panel of data, the main purpose of this paper seeks to develop two new tests for the rank q of the covariance matrix $\breve{\Sigma}_v$, when the r factors $\breve{f}_t$ are estimated by Principal Component Analysis as in Stock and Watson (2002) and Bai and Ng (2002), applied to the large (i.e. with $N, T \to \infty$) panel of individual observations y_t. Consistent estimation procedures for the number q of dynamic factors based on Information Criteria (IC)[5] have been derived by relying on the rate of convergence of the PC estimator $\hat{f}_t$ of $\breve{f}_t$ by Amengual and Watson (2007), Bai and Ng (2007) and Breitung and Pigorsch (2013). Differently from their procedures, we can directly test the rank $q \leq r$ of the residual covariance (or correlation matrix) of a VAR model estimated on $\hat{f}_t$.

2.2 Bai and Ng comparison

Bai and Ng developed their studies on the econometric theory of large-scale factor models. First they focused on determining the number of factors r, an unsolved problem in the rapidly growing literature on multifactor models, and later on the number of shocks driving economic volatility q. In their first paper that we will address in this thesis, *Determining the Number of Factors in Approximate Factor Models*, which dates back to 2002, they set the rate of convergence of factor estimates to allow for consistent estimates of r. They then proposed some panel criteria and showed that the number of factors can be estimated consistently using this criterion. The theory was developed in the context of large cross sections N and a large time dimension T. Simulations showed that the proposed criterion of good finite-sample attributes exists in many panel data configurations encountered in practice.

In contrast, in their second paper, they addressed a common but untested assumption in macroeconomic analysis which states that the number of shocks that determine economic volatility, q, is usually small. In their research, they tried to relate q to the number of dynamic factors in a large panel of data and proposed a method to determine q without estimating the dynamic factor. They first estimated the VAR in r static factors, where the factors are obtained by applying the principal components method to a large data set, then calculated the eigenvalues of the covariance or residual correlation matrix. Finally, they checked whether the eigenvalues satisfied the asymptotic contraction bound. This led to reflection on sampling error. Before going ahead, we briefly remember the Bai and Ng's model in *Determining the Number of Primitive Shocks in Factor Models* as the source of an unsolved problem to date.

Unlike the work completed by Bai and Ng, in this paper we will show the determination of procedures that can asymptotically test the number of estimated factors in two parallel but different ways. Our analysis will mainly focus on the estimate procedure they conducted, and this will be replicated in the following lines of

[5] An information criterion is a measure of the quality of a statistical model. To do this it takes into account the degree to which the model fits the data and the complexity of the model. Information criteria are used to compare alternative models fitted to the same data set. All things being equal, a model with a lower information criterion is superior to a model with a higher value.

code. To do so, we will take advantage of what has been done before, generating a DGP and getting its principal components as we previously discussed. Although the following example does not represent the actual method we discussed earlier, since the model is static and not dynamic (f does not depend on any other variable), this may still help us to better understand the procedure to go through to build a PCA.

```
clear

N=1000;
T=200;

eps=randn(N,T)*0.1
Lambda=rand(N,1)-0.5
f=randn(1,T)

y=(Lambda*f+eps)'

[eigvect,eigval]=eig((y-mean(y))*(y-mean(y,2))')

first_eigvect=eigvect(:,1)

F_hat=first_eigvect*sqrt(T)

figure
plot([f',F_hat])

Lambda_hat_transposed=(F_hat'*y)/T
Lambda_hat=Lambda_hat_transposed'

lambda_comparison=[Lambda,Lambda_hat]
```

The approach just described through the use of Matlab code shows values centered first on the mean of the columns and then on the mean of the rows, so that these transformed values are used instead of the original observations for each variable. This procedure represents a key step that will be taken up once the comparison with the method adopted by Bai and Ng is discussed. With this example we wanted to provide a basic concrete method for using and implementing a PCA for the purpose of estimating parameters.
Continuing with this procedure, the next step should be to calculate the variance-covariance matrix of the

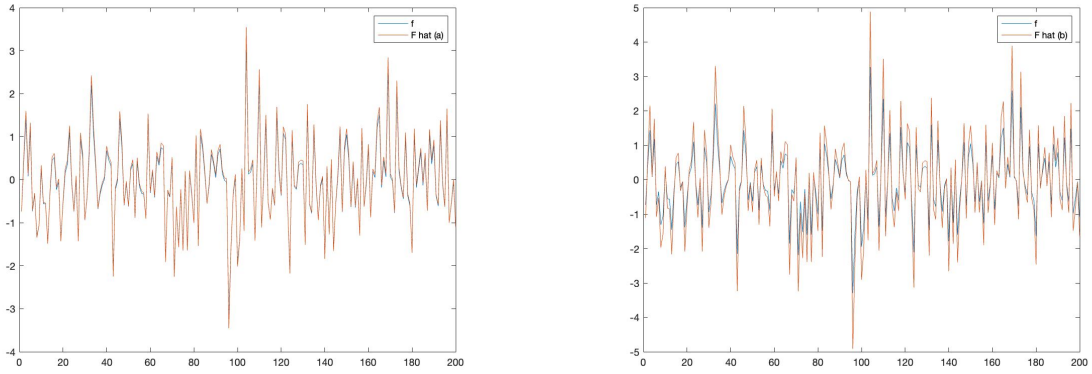
data and to compute the corresponding eigenvalues and eigenvectors of this matrix. This can be obtained by simply multiplying the centered data by the columns and rows. Thus, it will then be sufficient to normalize each of the orthogonal eigenvalues, transforming them into unit vectors. This basic choice transforms the covariance matrix into a diagonalized form, with the diagonal elements representing the variance of each axis. The percentage of variance represented by each eigenvector can be calculated by dividing the eigenvalue corresponding to that eigenvector by the sum of all eigenvalues.

At this point it should be clear to understand why PCA is defined as an orthogonal linear transformation that transforms the data into a new coordinate system such that a scalar projection of the data places the largest variance in the first coordinate (called the first principal component), the second largest variance on the second coordinate and so on.

2.2.1 Identification problem

As we can see by running a couple of times the same lines of code, starting by the very first DGP, the shape of the chart is likely to change. As a result, sometimes the estimated parameters' line turns out to be specular with respect to the one assumed as true. This trend is due to the so-called parameter identification problems, which are defined as problems that may arise when several estimators satisfy the maximization of the objective function (in our case we will see that it is the minimization of the variance of errors). This fact is closely related to unidentifiability in statistical models and econometrics, when a statistical model has more than one set of parameters that produce the same observed distribution, it means that multiple parameterizations are equivalent in terms of observations.

Hence, a parameter is identifiable if it can be unambiguously determined using the available data. This problem is well known in the context of simultaneous equations, but it is not unique to this case.

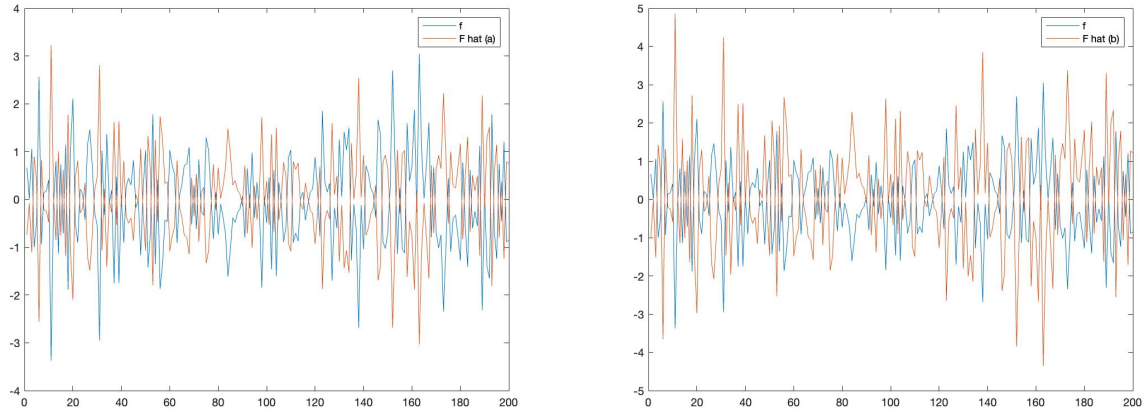


(a) Estimate method a) **(b)** Estimate method b)

Figure 1. Graphical comparison between two different estimation methods according to Bai and Ng.

In the case just shown by the two figures it is possible to see that the approximation of the data provided by the estimator is very precise, therefore the estimator is efficient. By running again the same code seen

previously it is likely to see one or both figures with one of the two lines represented in a specular way than the other. This is due to rotations that affect the estimation of parameters and that fall within the scope of the so-called identification problems.



(a) Estimate method a) **(b)** Estimate method b)

Figure 2. Mirrored graphs due to identification problems

Therefore, as stated just now, an identification problem refers to whether numerical estimates of the structural equation parameters can be obtained from the estimated reduction coefficients. If this is possible, we say that the particular equation is deterministic[6]. If this is not possible, we say that the equation under consideration is unidentified or underidentified. To try to make things as simple as possible here is an example:

$$y = \Lambda f + \varepsilon \qquad\qquad y = -\Lambda(-f) + \varepsilon$$

As we can easily see from observing the previous two equations, both lead to the same result regardless of the sign of the inputs given. Algebraically this occurs because of the mathematical rule that the product of two negative numbers returns a positive number, however in econometrics the sign of the factor and loadings matrix are not negligible. This is why we introduced the problem of parameter identification as the situation that occurs when the value of one or more parameters in an economic model cannot be determined from the observable variables.

This problem is closely related to non-identifiability in statistics and econometrics, which occurs when a statistical model with more than one set of parameters produces the same distribution of observations, meaning that multiple parameterizations are observable equivalent. An identified equation can be precisely identified or overidentified. It should be accurately identified whether the unambiguous value of the structural parameter can be get. Overidentification is said to exist when more than one value is available for some parameters of the structural equation. In order to solve the problem of identification a new estimation procedure can

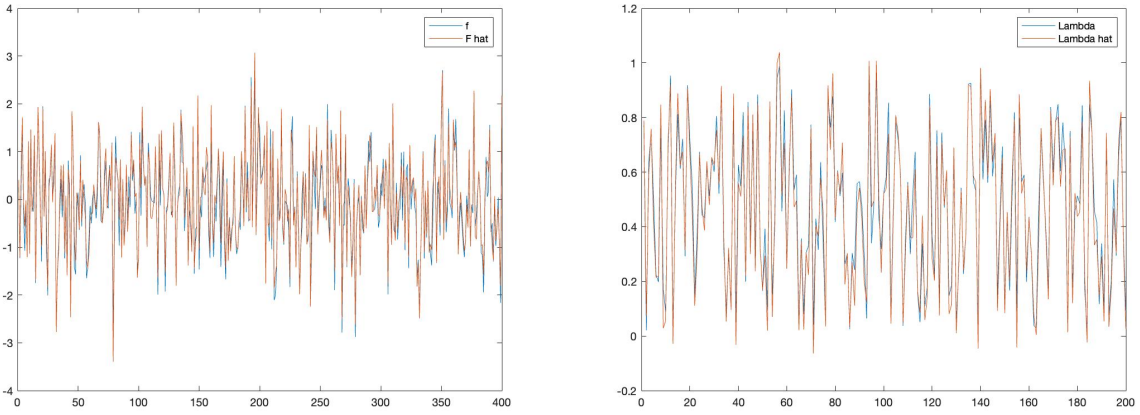
[6] A deterministic equation is an equation that governs the motion of a dynamical system and does not contain terms corresponding to random forces.

be developed thanks to the use of Monte Carlo simulations. This approach allows us to avoid any risk of finding ourselves in the previous situation in which a rotation of factors would, in some cases, distort the estimation of the parameters under analysis.

```
%% 1) Data generation
clear
N=200;
T=400;
M=1000; %Monte Carlo replication

rng(10);
Lambda=rand(N,1)-0,5;
rng(1000);
f=randn(1,T);
f=f/std(f,0);
Com_comp=Lambda*f; %common components matrix

%% 2) Bai and Ng estimation
for j=1:M

   display(j)
   rng(1000*j+55);
   eps=randn(N,T);
   y=(Com_comp+eps)'; %(T × N) matrix

   [eigvect,eigval]=eig((y-mean(y,1))*(y-mean(y,2))'); %no need to store every result in the workspace
   first_eigvect=(eigvect(:,1));

   f_hat1,j=first_eigvect*sqrt(T);
   Lambda_hat1,j=((f_hat1,j'*y)/T)';
end

%% 3) Bias estimation
f_hat_matrix=cell2mat(f_hat); %'cell2mat' shows cell arrays in a single matrix
Lambda_hat_matrix=cell2mat(Lambda_hat);
```

```
S=(f_hat_matrix(2,:)>0);

f_hat_matrix(:,S)=-1*f_hat_matrix(:,S);
Lambda_hat_matrix(:,S)=-1*Lambda_hat_matrix(:,S);

figure
plot([f',f_hat_matrix(:,1)])
legend("f","F hat")

figure
plot([Lambda,Lambda_hat_matrix(:,1)])
legend("Lambda","Lambda hat")

BiasL=mean(Lambda_hat_matrix,2)-Lambda; %Bias formula
StdL=std(Lambda_hat_matrix,0,2);
Avg_BiasL=mean(BiasL);
Avg_StdL=mean(StdL);

Bias_f=mean(f_hat_matrix,2)-f';
Std_f=std(f_hat_matrix,0,2);
Avg_Bias_f=mean(Bias_f);
Avg_Std_f=mean(Std_f);

save('Results_MC')
```

As a results what we obtain are the following graphs:



(a) f estimate **(b)** Λ estimate

Figure 3. Parameter's estimation through Monte Carlo simulation

2.2.2 Number of primitive shocks in vector autoregression

In the last part of this section devoted to the work of Bai and Ng, the focus will be on estimating the minimum number of primitive shocks in vector autoregression. As mentioned earlier, the analyses conducted over the years by the two researchers have focused on the assumption that in macroeconomics all economic fluctuations are driven by a small number of shocks. In this regard, scientific evidence has shown, in most cases, that the number of shocks involved in the approximation of a model rarely exceeds the value of four. However, the question arises: how to quantify the exact number of primitive shocks? The answer to this question can be found within Bai and Ng's *Determining the Number of Primitive Shocks in Factor Models* (2007)[7]. The objective of the article is to estimate the number of common factors in a dynamic factor model, without estimating a dynamic factor model. Surprisingly, as far as the relevant literature is concerned, few tests can formally estimate what the exact value of q is. One assertion commonly recognized as valid is to put $q = 2$, since there is no formal test for the number of dynamic factors. Dynamic factor models aggregate information on large amounts of data with only a few factors. What is important to note is that the rank of the spectrum[8] of q dynamic factors is always q. Since $r < \infty$ static factors can be dynamically correlated, the spectrum of $r \geq q$ static factors has a reduced rank. According to this we can see that this rank is actually q, the number of dynamic factors.

What will follow in the pages immediately ahead is a Matlab-implemented analysis of what is described in the paper itself, whose correctness and clarity has been confirmed by our empirical evidence[9].

```
%% Section 1
```

[7] Resuming the original notation of Bai and Ng, the number of shocks will also be denoted by the letter q in Carlini and Rubin (2022)

[8] Spectral analysis concerns the process of determining the frequency content of a time-continuous signal in the discrete time domain

[9] The entire procedure was accomplished by following step by step the instructions in the paper

```
clear

N=100;
r=4;
q=2;
T=200;

rng(10);
eps=rand(q,T)-0.5;
R=rand(r,q);
Rank_R=rank(R);
u=R*eps;

B=[-0.1 0.3 0.2 0.4;-0.3 0.9 0.6 1.2;0.8 0.7 0.2 0.5;0.4 0.35 0.1 0.25];
Rank_B=rank(B);
%we have constructed this matrix so that it has rank = q

v=inv(B)*u %weird values though

Sigma_eps=(eps*eps')/T;
Sigma_u=(u*u')/T; %Sigma_u is also equal to R*Sigma_eps*R'
Rank_Sigma_u=rank(Sigma_u);

a=rand(q,r);
A=a'*a;
[eigvect_A,eigval_A]=eig(A);
Rank_A=rank(A);

%Sort eigenvalues (and eigenvectors) from largest to smallest
[eig_A,index_eiden_sort]=sort(diag(eigval_A,'descend');
eigval_A=sort(diag(eigval_A),'descend');
eigvect_A = eigvect_A(:,index_eigen_sort);

%% Section 2
```

```
c=eigval_A;
k=sum(c<0.0000000001); %we assume these values equal to zero | k = 'number of eigval = 0'
D_1k=sqrt((c(k+1,1))^2/sum(c.^2));
D_2k=sqrt(sum((c(k+1:r,1).^2)) / (c.^2));

%% Section 3

beta=eigvect_A;

A_prime=zeros(r,r);
for j=1:r

    A_j{j,1}=c(j,1).*beta(:,j)*beta(:,j)'
    A_prime=A_prime+A_j{j,1}

end
A_prime_k=zeros(r,r);
for w=1:k

    A_i{w,1}=c(w,1).*beta(:,w)*beta(:,w)'
    A_prime_k=A_prime_k+A_i{w,1}

end
%If we compare A with both A prime(s) we discover that the three matrices are basically the
same thing

A_prime_k1=zeros(r,r);
for w=1:k+1

    A_i{w,1}=c(w,1).*beta(:,w)*beta(:,w)'
    A_prime_k1=A_prime_k1+A_i{w,1}

end

d_k=reshape(A_prime_k,[r*r,1]);
d_0=reshape(A,[r*r,1]);
d_k1=reshape(A_prime_k1,[r*r,1]);

D_1k_prime=norm(d_k1-d_k)/norm(d_0);
```

```
D_2k_prime=norm(d_k-d_0)/norm(d_0);

%As we excpected the values of D_1k and D_2k are extremely close to D_1k_prime and D_2k_prime

%% Section 4

for z=1:j

    Trace_beta{z}=trace(beta(:,z)*(beta(:,z))')
    Norm_beta{z}=(norm(beta(:,z)))^2

end

%By checking these values out we can confirm what has been stated into the paper, so trace
= norm = 1

d_0_sq=reshape(A,[r*r,1])'*reshape(A,[r*r,1]);
Trace_A=trace(A'*A);
Norm_A=norm(A^2);

%d_0 squared is exactly equal to the trace of A, while the norm of A is slightly different
from their values
```

2.3 Canonical correlation analysis

Before introducing the actual test, however, it is necessary to understand one last concept that is absolutely necessary to remember in order to proceed with this analysis.

Canonical correlation analysis (CCA) is a multivariate statistical technique that can be used to analyze the correlation between two sets of data. In statistics, CCA is a method of obtaining information from a cross-covariance matrix. If we have two random variable vectors $X = (X_1, \ldots, X_n)$ and $Y = (Y_1, \ldots, Y_m)$ and there is a correlation between the variables, then canonical correlation analysis finds all those linear combinations of X and Y that have the highest correlation with each other.

In more formal terms, given two column vectors $X = (x_1, \ldots, x_n)'$ and $Y = (y_1, \ldots, y_m)'$ of random variables, the cross-variance $\Sigma_{XY} = \mathrm{cov}(X, Y)$ might be defined as an $n \times m$ matrix whose (i, j) entry is the covariance $\mathrm{cov}(x_i, y_j)$. In practice the aim is to estimate the covariance matrix based on sampled data from X and Y (i.e. from a pair of data matrices).

Canonical correlation analysis seeks vectors a ($a \in \mathbb{R}^n$) and b ($b \in \mathbb{R}^m$) such that the random variables $a^T X$ and $b^T Y$ maximize the correlation $\rho = \mathrm{corr}(a^T X, b^T Y)$.

The random variables $U = a^T X$ and $V = b^T Y$ are the first pair of canonical variables. Then looking for the vectors that maximize the same correlation, provided they should be uncorrelated with the first pair of canonical variables; this gives the second pair of canonical variables. Repeating the same procedure is allowed up to $\min\{m, n\}$ times.

$$(a', b') = \underset{a,b}{\mathrm{argmax}}\, \mathrm{corr}(a^T X, b^T Y) \tag{2.3}$$

Canonical correlation analysis can be used to model the correlation between two datasets in two ways:

- By focusing on dependencies and regression models for two datasets: dataset y as a function of dataset x.
- By focusing on examining the relationship between two datasets without specifying any datasets as dependent or independent variables. In this case, we can compare it with methods like PCA or factor analysis.

There are two key concepts to understand in canonical correlation analysis:

- canonical variable
- canonical correlation

The first core concept of canonical correlation analysis is the concept of canonical variables. Of course, datasets are difficult to model at once because they contain a large number of variables. Analyzing the correlations between all the variables in a dataset is already a challenge, so now the task is more complicated because we need to distinguish the correlation between the variables and the correlation between the two datasets. A canonical variable is a linear combination of variables from one of the datasets. If we have to deal

with different numbers of variables in our datasets, we can have as many pairs of canonical variables as there are variables in the smallest dataset. Typical variables are defined by the model. They are chosen as the linear combinations of the (original) variables with the greatest possible correlation. Since this correlation is measured between canonical variables, we simply call it Canonical Correlation.

A final useful comparison that can be quickly observed is between canonical correlation analysis and principal component analysis, which we have previously discussed. PCA is a method of finding linear combinations (called principal components) in a dataset with the goal of maximizing the set of changes explained by these principal components. While PCA focuses on finding the linear combination that explains the largest variance in the datasets, canonical correlation analysis focuses on finding the linear combination that explains the largest correlation in the two datasets.

3 The test

3.1 Introduction

Previously, the main concepts have just been introduced in order to fully understand what will be described in this section. From now on we will be looking at how to set up our model in order to show some results that will be the subject of our further analysis. However, this section will not exactly describe the test developed in the paper by Carlini and Rubin (2022), since it is not the object of this dissertation to report in detail what is already available from the reading of the paper itself. Besides a brief introduction to the model and its structure, in this section we will see and try to understand what is the idea behind the research without going into the more technical and complex aspects. For any further discussion, the last sub-section of this chapter will be entirely dedicated to the coding aspect, so that the reader can see how much work went into implementing all the ideas and giving them a concrete form.

3.2 The model

As seen above, the model follows a pattern as described for example by the two equations below:

1. $y_t = \Lambda f_t + \varepsilon_t \qquad\qquad t = 1, ..., T$
2. $y_{t-1} = \Lambda f_{t-1} + \varepsilon_{t-1} \qquad\qquad t = 1, ..., T$

where we have to keep in mind that $f_t = \Phi f_{t-1} + v_t$ and $v_t = G\,\eta_t$, with $\mathrm{V}(\eta_t) = \mathrm{I}$.

$$\begin{array}{ccccccccc} f_t & = & \Phi & \times & f_{t-1} & + & G & \times & \eta_t \\ (r \times 1) & & (r \times r) & & (r \times 1) & & (r \times q) & & (q \times 1) \end{array}$$

Before proceeding, it is also good to remember two fundamental properties to provide a different approach that will be useful to take an extra step towards the idea of the problem:

1. $G_\perp \in ker(G)$ where $G_\perp$ is an $r \times (r - q)$ matrix, with $q \leq r$
2. $G_\perp{}'G = 0$

Through this transformation, including the $G_\perp$ matrix, our equation results to be written as follows:

$$G_\perp{}' f_t = G_\perp{}' \Phi\, G_\perp G_\perp{}' f_{t-1} + G_\perp{}' G\, \eta_t$$

By exploiting the second property above, it is possible to get rid of the last term since $G_\perp{}'G = 0$ and therefore rewrite the regression in a better form, in which no error term appears:

$$\begin{array}{ccccc} g_t^\star & = & \Phi^\star & \times & g_{t-1}^\star \\ (r-q) \times 1 & & (r-q) \times (r-q) & & (r-q) \times 1 \end{array}$$

Through parameter rotation we are now ready to determine how many canonical correlations of (f_t, f_{t-1})[10] are equal to 1, establishing an unbiased and perfectly predictable correlation. As it will seen later on, one of the coming tasks will be to enumerate exactly how many $\text{corr}(g_t^\star, g_{t-1}^\star) = 1$. This result will represent the one of the key concepts of the entire paper. Starting from the size of the matrix Φ, looking at its r coefficient, and subtracting this value for the size $r - q$ of the transformation obtained by the rotation of f_t with G_t, it is possible to obtain in a trivial way the value of q representing the size of the matrix $G * \eta_t$, as well as the number of shocks present in the system Σ_v.

On a simpler note, given for example $r = 3$ and $r - q = 1$, we calculate that q is equal to 2. This example is not entirely coincidental; in fact, as assumed earlier, one of the assumptions made in analyzing this type of model is that the number of observable shocks is small and usually exactly equal to two.

However, in this case, the parameter estimate is not exactly f_t, but $\hat{f}_t$ that corresponds to the product of f_t with an orthogonal matrix that we will call Q, such that $Q\,Q' = Q'\,Q = \text{I}$. As a result, what we get in the model is a v_t that is not exactly a v_t, but a rotated object $\tilde{v}_t$ due to the presence of the orthogonal matrix Q. At the same time it should be noted that this rotation does not affect the eigenvalues of the variance-covariance matrix of the $\tilde{\Sigma}_v$ matrix which are exactly equal to a "rescaling"[11] of the Σ_v matrix, or rather they are exactly equal to those of the Σ_v matrix. Consequently, the rotation does not directly impact the test, or rather the rotation only impacts the moment one chooses to perform the "right" test, where the "right" test is one that has v_t in the form written as G for η_t where $G = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. So as to have the shock above and 0 below. This structure is dictated by the expected outcome of the test. To formalize:

$$Q\,f_t = \Phi\,Q\,f_{t-1} + \tilde{v}_t \qquad\qquad \tilde{\Sigma}_v \sim \Sigma_v$$

Therefore, the idea of the test is to calculate how many eigenvalues are exactly equal to zero in Σ_v, and consequently to test them. An alternative but complementary approach would be the one discussed above, which is to test how many canonical correlations between $\hat{f}_t$ and $\hat{f}_{t-1}$ are exactly equal to one. The number of eigenvalues equal to zero, i.e., the number of dynamic shocks within the system Σ_v, will be represented by q, while the number of canonical correlations equal to one by $r-q$, pointing out that the $r-q$ value represent the number of rotations concerning f_t and f_{t-1} that have the highest level of correlation and therefore are perfectly predictable.

In a nutshell, the core part of the work conducted by Carlini and Rubin will be to prove the one-to-one parameter identification problem between the values just discussed.

3.2.1 Eigenvalues of the innovations' covariance matrix of the factor VAR

Assuming a sequence of r eigenvalues, ordered from the first to the ℓ-th one, let σ_ℓ^2 be the ℓ-th largest eigenvalue of the VAR covariance innovations' matrix $\breve{\Sigma}_v = GI_qG'$. Since $\breve{\Sigma}_v$ has reduced rank q, this means

[10] or as in the case under consideration $(g_t^\star, g_{t-1}^\star)$.

[11] In this context this term takes on the meaning of changing the scale or size of the values in the matrix.

that the smallest $r-q$ eigenvalues of $\breve{\Sigma}_v$ must equal zero, while its largest q eigenvalues are strictly positive, that is:

$$\sigma_1^2 \geq \sigma_2^2 \geq ... \geq \sigma_q^2 > \sigma_{q+1}^2 = \sigma_{q+2}^2 = ... = \sigma_r^2 = 0 \tag{3.1}$$

An equivalent way of expressing the Data generating process may be useful to i) simplify the derivation of the distribution of the test statistics for the number q of common dynamic shocks, and ii) allow to introduce an alternative identification strategy for q along with the common dynamic factors themselves. Let W_v be the (r,r) matrix collecting the eigenvectors associated to the ordered eigenvalues σ_ℓ^2, with $\ell = 1, ..., r$, of $\breve{\Sigma}_v$, and let $\breve{\Sigma}_v := diag(\sigma_1^2, ..., \sigma_q^2, 0, ..., 0)$ be the (r,r) diagonal matrix collecting the ordered eigenvectors of $\breve{\Sigma}_v$. Then

$$\breve{\Sigma}_v W_v = W_v \Sigma_v, \tag{3.2}$$

with $W_v' W_v = W_v W_v' = I_r$. Then, by multiplying both sides of equation (2.2) by W_v, what we obtain is: $W_v' \breve{f}_t = W_v' \breve{\Phi} W_v W_v' \breve{f}_{t-1} + W_v' \breve{v}_t$. By defining $f_t := W_v' \breve{f}_t$ and $\Lambda = [\lambda_1', ..., \lambda_N']' := \breve{\Lambda} W_v$, which implies $\lambda_i = W_v' \breve{\lambda}_i$ for $i = 1, ..., N$, the DGP for the observable variables y_t in the equation (2.1) can be equivalently written as:

$$y_t = \Lambda f_t + \varepsilon_t. \tag{3.3}$$

By also defining $\Phi := W_v' \breve{\Phi} W_v$, the DGP for the factors can be written as:

$$f_t = \Phi f_{t-1} + v_t, \tag{3.4}$$

where the innovation vector $v_t := W_v' \breve{v}_t$ is such that:

$$v_t \sim iid(0, \Sigma_v), \tag{3.5}$$

with $\Sigma_v := V(v_t) = W_v' \breve{\Sigma}_v W_v$. Equation (3.5) and the definition of Σ_v imply:

$$v_t = [v_{Ht}', v_{Lt}']' = [v_{Ht}', 0_{q \times 1}]', \tag{3.6}$$

and the model (3.4) can be re-written as:

$$\begin{bmatrix} f_{Ht} \\ f_{Lt} \end{bmatrix} = \begin{bmatrix} \Phi_{HH} & \Phi_{HL} \\ \Phi_{LH} & \Phi_{LL} \end{bmatrix} \begin{bmatrix} f_{Ht-1} \\ f_{Lt-1} \end{bmatrix} + \begin{bmatrix} v_{Ht} \\ 0 \end{bmatrix}. \tag{3.7}$$

Equations 3.3 and 3.7 show that under the reduced rank assumption of made for matrix $\breve{\Sigma}_v$, there exist a specific rotation of the factors $\breve{f}_t$, given by the eigenvectors of $\breve{\Sigma}_v$ itslef (i.e. $f_t := W_v' \breve{f}_t$) such that the bottom q-dimensional subvector f_{Lt} of the rotated factors f_t is a linear combination of the *lagged* values of both upper subvector f_{Ht} and the lower subvector f_{Lt} of f_t itself.

3.2.2 Canonical correlations between factors and their lagged values

Given the special result obtained by the shape of the innovations in model (3.7), this also implies that there exist $r-q$ linear combinations of f_t (resp. $\breve{f}_t$) which are perfectly correlated with other $r-q$ linear combinations of f_{t-1} (resp. $\breve{f}_{t-1}$) or, equivalently that there exist q canonical correlations between f_t (resp. $\breve{f}_t$) and f_{t-1} (resp. $\breve{f}_{t-1}$) which are exactly equal to one.

For this specific demonstration we leave it to the reader to consult *New Tests and Estimators for Common Dynamic Factors* (2022) by Carlini and Rubin under the heading *Proposition 1.* In a nutshell, what *Proposition 1* shows is that the number of dynamic factors q, and the dynamic factor space spanned by f_{Ht}, can be identified from the canonical correlations and canonical variables between $\breve{f}_t$ and $\breve{f}_{t-1}$, or equivalently f_t and f_{t-1}. Specifically, the factor space dimensions q, $r-q$, and the dynamic factors f_{Ht} (up to a rotation) are identifiable from information that can be inferred by i) performing PCA to identify the static factors f_t, and ii) by performing a canonical correlation analysis on the panels of $\breve{f}_t$ and $\breve{f}_{t-1}$. Indeed, PCA on the panel of observations y_t allows to identify f_t and f_{t-1} up to (the same) one-to-one linear transformation. The latter indeterminacy does not prevent identifiability of dynamic factors f_{Ht} from *Proposition 1*, due to the invariance of canonical correlations and canonical variables under linear one-to-one transformations of vectors f_t and f_{t-1}.

3.3 The code

The test now being discussed is the result of extremely meticulous work carried out by Professors Carlini and Rubin whose proof and evidence from the econometric point of view can be compared by reading their paper. One of the objectives of this dissertation is to understand how the arguments discussed so far have been applied and have led to such a complex result. In this regard, not all aspects of the test will be explained in detail, rather a discussion will be provided about the idea behind the entire paper.

Going into detail, what has not yet been said is that the main test is actually composed of the two tests discussed so far (as well as a possible further test which will be mentioned later on). The structure of the code in fact provides that the two tests are performed separately but at the same time so that we can exploit the same DGP and thus compare the relative outputs. Regarding the Data generating process, the code below follows almost exactly what has already been described in section (2.1), where, however, we can see that it is represented by a single function ('f_DGP.m').

```
function [mY] = f_DGP(mG0,mPhi0,lambdast,sigma,iQ,iR,iN,iT)
mF=zeros(iR,iT+1);
mV=randn(iQ,iT+1);
for t=2:iT+1
    mF(:,t)= mPhi0*mF(:,t-1)+mG0*mV(:,t);
```

```
end
mF=(mF'-mean(mF'))';
mLambda=lambdast*randn(iN,iR);
mEps=randn(iN,iT+1)*sigma;
mEps=(mEps'-mean(mEps'))';
mY=mLambda*mF+mEps;

end
```

As we can see from its structure, the code here shows all the main features of a function even if its inputs are not present. In fact they will be displayed later, at the beginning of the main code script, precisely 'main_MC_size_power.m'.

```
clc
clear
close all
mG0=[1 0; 0 1; 0 0];
%mG0=[1; 0; 0];
mPhi0=[0.9 0.5 -0.2; -0.4 0.7 0.1; 0.1 0.2 0.6];
lambdast=1;
iN=30;
iT=100;
iQ=2;
iR=3;
nMC=2000;
sigma=0.4;

Test_Th2=zeros(nMC,1);
Test_Th3=zeros(nMC,1);
Test_Th4=zeros(nMC,1);

parfor s=1:nMC
display(s)
mY=(f_DGP(mG0,mPhi0,lambdast,sigma,iQ,iR,iN,iT));
mY=mY';
r=3;
```

```
q=iQ;
Test_Th4(s)=f_TEST_Th4(mY,r,q,iN,iT);
r=3;
q=iQ;
Test_Th3(s)=f_TEST_Th3(mY,r,q,iN,iT);
r=3;
q=iQ;
Test_Th2(s)=f_TEST_Th2(mY,r,q,iN,iT);
end

histogram(Test_Th4,20,'Normalization','pdf')
hold on
y = -6:0.1:6;
mu = 0; sigma = 1; f = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
plot(y,f,'LineWidth',1.5)
hold off

ResultsTh2 = [mean(Test_Th2), median(Test_Th2), std(Test_Th2,0), iqr(Test_Th2)]
ResultsTh3 = [mean(Test_Th3), median(Test_Th3), std(Test_Th3,0), iqr(Test_Th3)]
ResultsTh4 = [mean(Test_Th4), median(Test_Th4), std(Test_Th4,0), iqr(Test_Th4)]

save('N_30_T_100_MCstatistics')
```

This is essentially the program.

Its structure begins specifying a series of variables that need to come maintained constant, or others that will be made to vary according to the simulations carried out like it will be seen later on. Matrices G_0 and Φ should remain constant as long as the size of r and q remain as this first case ($r = 3$, $q = 2$). In addition there are some other parameters including Λ^* we assume to be equal to one. For example, the standard deviation of the error, σ_ε, could have also been 0.5 either any value in the range (0, 1), but let us assume it is equal to 0.4. The other inputs, preceded by the lowercase letter i, identify q as the number of columns in G_0, its number of rows r and the size of the panel of data $T \times N$ that will be processed each time through every simulation, whose number is described by 'nMC' which stands for the number of Monte Carlo iterations. Since the number of zeros in the autocovariance matrix of v_t, which basically represents the shocks of the VAR, is exactly equal to the number of canonical correlations which are equal to one, this implies that whatever result is tested will always be something of the form $r - q$. As a result what we get is the 'My' panel of data that needs to be fixed right away because 'My' is exactly inside the 'f_DGP.m'

function. Once the different y's have been generated, it is possible to perform the different tests that are proposed just below, scrolling the code.

```
function [Tstat] = f_TEST_Th2(x,iR,iQ,iN,iT)
%x2= zscore(x);
iX=x*(x')/(iT+1); % compute eigenvctros form T X T matrix obtained form standardized data:
each T.S. is demeaned and divided by its st.dev computed in time series
[eigvR_uns, eigR_mat_uns] = eigs(iX,iR);
% Sort eigenvalues (and eigenvectors) from largest to smallest
[eigR, index_eigen_sort] = sort(diag(eigR_mat_uns),'descend');
eigvR = eigvR_uns(:,index_eigen_sort);

Fhat=sqrt(iT+1)*eigvR;
Lhat=1/(iT+1)*(x)'*Fhat;
epsHat = x- Fhat*Lhat';
%S=[mLambda0*inv(mLambda0(end-1:end,end-1:end)),Lhat*inv(Lhat(end-1:end,end-1:end))]
Fhat_t=Fhat(2:end,:);
Fhat_tm1=Fhat(1:(end-1),:);

Phi_hat=Fhat_t'*Fhat_tm1/(Fhat_tm1'*Fhat_tm1);
V_hat= Fhat_t'-Phi_hat*Fhat_tm1';
Sigv_hat=1/iT*V_hat*(V_hat');

[eig_VECT,eig_Sigv_hat]=eigs(Sigv_hat);
[eig_Sigv_hat_sorted,index_Sv]=(sort(diag(eig_Sigv_hat),'descend'));
eig_VECT = eig_VECT(:,index_Sv);

%Phi_hat=V'*Phi_hat*V
%Gamma=zeros(iN,1);
%for i=1:iN
    %Gamma(i)=1/iT*sum(epsHat(2:end,i));
%end
Phi_hat=eig_VECT'*Phi_hat*eig_VECT;
Lhat=Lhat*eig_VECT;
Gamma=cov(epsHat(2:end));
Gamma=diag(diag(Gamma));
```

```
Sigu_hat=iN*((Lhat'*Lhat)/((Lhat'*Gamma*Lhat)/(Lhat'*Lhat)));


L=iQ+1:iR;

H=1:iQ;


B_Uhat=Phi_hat(L,H)*Sigu_hat(H,H)*Phi_hat(L,H)'+...
    +Phi_hat(L,L)*Sigu_hat(L,H)*Phi_hat(L,H)'+...

    Phi_hat(L,H)*Sigu_hat(H,L)*Phi_hat(L,L)'+...

    Sigu_hat(L,L)+...

    Phi_hat(L,L)*Sigu_hat(L,L)*Phi_hat(L,L)';

SUzero=Phi_hat(L,H)*Sigu_hat(H,H)*Phi_hat(L,H)'+...
    +Phi_hat(L,L)*Sigu_hat(L,H)*Phi_hat(L,H)'+...

    Phi_hat(L,H)*Sigu_hat(H,L)*Phi_hat(L,L)'+...

Sigu_hat(L,L)+...
      Phi_hat(L,L)*Sigu_hat(L,L)*Phi_hat(L,L)';

SUone=-Phi_hat(L,H)*Sigu_hat(L,H)'-Phi_hat(L,L)*Sigu_hat(L,L)';

Omega_U1=2*trace((SUzero*SUzero'+2*SUone*SUone')); %it needs to be‘2*’


Tstat= iN*sqrt(iT)*(Omega_U1)^(-1/2)*(sum(eig_Sigv_hat_sorted(L))-1/iN*trace(B_Uhat));


end
```

The just displayed Theorem 2 concerns the test based on going to determine whether the last or smallest eigenvalues of the variance-covariance matrix have value of zero.

```
function [Tstat] = f_TEST_Th3(x,iR,iQ,iN,iT)

iX=x*(x')/(iT+1); % compute eigenvctros form T X T matrix obtained form standardized data:
each T.S. is demeaned and divided by its st.  dev computed in time series

[eigvR_uns, eigR_mat_uns] = eigs(iX,iR); % Sort eigenvalues (and eigenvectors) from largest
to smallest

[eigR, index_eigen_sort] = sort(diag(eigR_mat_uns),‘descend’);

eigvR = eigvR_uns(:,index_eigen_sort);


Fhat=sqrt(iT+1)*eigvR;

Lhat=1/(iT+1)*(x)'*Fhat;

epsHat = x-Fhat*Lhat';
```

```
%S=[mLambda0*inv(mLambda0(end-1:end,end-1:end)),Lhat*inv(Lhat(end-1:end,end-1:end))]
Fhat1=Fhat(2:end,:);
Fhat2=Fhat(1:(end-1),:);

Phi_hat=Fhat1'*Fhat2/(Fhat2'*Fhat2);
V_hat= Fhat1'-Phi_hat*Fhat2';
Sigv_hat=1/iT*V_hat*(V_hat');

[V1,eig_Sigv_hat]=eigs(inv(Fhat1'*Fhat1)*(Fhat1'*Fhat2)*inv(Fhat2'*Fhat2)*Fhat2'*Fhat1);
[eig_Sigv_hat, index_eigen_sort3[ = sort(diag(eig_Sigv_hat),'descend');
eigvR = V1(:,index_eigen_sort3);
%eig_Sigv_hat=diag(eig_Sigv_hat);

[eig_VECT,eig_Sigv_hat2]=eigs(Sigv_hat);
[eig_Sigv_hat_sorted,index_Sv]=(sort(diag(eig_Sigv_hat2),'descend'));
eig_VECT = eig_VECT(:,index_Sv);

%Phi_hat=V'*Phi_hat*V
%Gamma=zeros(iN,1);
%for i=1:iN
    %Gamma(i)=1/iT*sum(epsHat(2:end,i));
%end
Phi_hat=eig_VECT'*Phi_hat*eig_VECT;
Lhat=Lhat*eig_VECT;
Gamma= cov(epsHat(2:end));
Gamma= diag(diag(Gamma));
Sigu_hat=iN*((Lhat'*Lhat)/((Lhat'*Gamma*Lhat)/(Lhat'*Lhat)));

L=iQ+1:iR;
H=1:iQ;

B_Uhat= Sigu_hat(L,L)+...
```

```
    Phi_hat(L,H)*Sigu_hat(H,H)*Phi_hat(L,H)'+...
    +Phi_hat(L,L)*Sigu_hat(L,H)*Phi_hat(L,H)'+...
    Phi_hat(L,H)*Sigu_hat(H,L)*Phi_hat(L,L)'+...
    Phi_hat(L,L)*Sigu_hat(L,L)*Phi_hat(L,L)';

SUzero=Sigu_hat(L,L)+...

    Phi_hat(L,H)*Sigu_hat(H,H)*Phi_hat(L,H)'+...
    +Phi_hat(L,L)*Sigu_hat(L,H)*Phi_hat(L,H)'+...
    Phi_hat(L,H)*Sigu_hat(H,L)*Phi_hat(L,L)'+...
    Phi_hat(L,L)*Sigu_hat(L,L)*Phi_hat(L,L)';

SUone=-Phi_hat(L,H)*Sigu_hat(L,H)'-Phi_hat(L,L)*Sigu_hat(L,L)';
%-Sigu_hat(L,H)*Phi_hat(L,H)';%-Sigu_hat(L,L)*Phi_hat(L,L)';
Omega_U1=2*trace(SUzero*SUzero'+2*SUone*SUone'); % it needs to be 2*

Tstat= iN*sqrt(iT)*(0.25*Omega_U1)^(-1/2)*(sum(sqrt(eig_Sigv_hat(1:iR-iQ)))-iR+iQ+1/(2*iN)*trace(B_Uhat))

end
```

Theorem 3 here above regards, instead, the second test. That one concerns the number of canonical correlation exactly equal to one.

A final useful result that the test provides is Theorem 4, which takes its cue from the paper by Andreou, Gagliardini, Ghysels, and Rubin, *Inference in Group Factor Models with an Application to Mixed Frequency Data* (2019). Here follows the code as well:

```
function [Tstat] = f_TEST_Th4(mY,iR,iQ,iN,iT)

x1= zscore(mY(2:end,:));
x2= zscore(mY(1:end-1,:));
%x1=x1';
%x2=x2';
%x=zscore(mY); % (T+1) x N
iX=(x1*(x1'))/(iT); % compute eigenvectors form T X T matrix obtained form standardized data:
each T.S. is demeaned and divided by its st.dev computed in time series
[eigvR_uns, eigR_mat_uns] = eigs(iX,iR);
% Sort eigenvalues (and eigenvectors) from largest to smallest
[eigR, index_eigen_sort] = sort(diag(eigR_mat_uns),'descend');
```

```
eigvR = eigvR_uns(:,index_eigen_sort);

iX2=(x2*(x2'))/(iT); % compute eigenvectors form T X T matrix obtained form standardized data:
each T.S. is demeaned and divided by its st.dev computed in time series
[eigvR_uns2, eigR_mat_uns2] = eigs(iX2,iR);
% Sort eigenvalues (and eigenvectors) from largest to smallest [eigR2, index_eigen_sort2] =
sort(diag(eigR_mat_uns2),'descend');
eigvR2 = eigvR_uns2(:,index_eigen_sort2);

Fhat1=sqrt(iT)*eigvR;
Fhat2=sqrt(iT)*eigvR2;

% Fhat'*Fhat/(iT+1)
% Lhat=1/(iT+1)*(mY)'*Fhat;
% epsHat = mY-Fhat*Lhat';
Lhat1=1/(iT)*(x1')*Fhat1;
epsHat1 = x1- Fhat1*Lhat1';

Lhat2=1/(iT)*(x2')*Fhat2;
epsHat2=x2-Fhat2*Lhat2';
%S=[mLambda0*inv(mLambda0(end-1:end,end-1:end)),Lhat*inv(Lhat(end-1:end,end-1:end))]
%Fhat_t=Fhat(2:end,:);
%Fhat_tm1=Fhat(1:(end-1),:);

%Phi_hat=Fhat_t'*Fhat_tm1/(Fhat_tm1'*Fhat_tm1);
%V_hat= Fhat_t'-Phi_hat*Fhat_tm1';
%Sigv_hat=1/iT*V_hat*(V_hat');

[V1,eig_Sigv_hat]=eigs(inv(Fhat1'*Fhat1)*(Fhat1'*Fhat2)*inv(Fhat2'*Fhat2)*Fhat2'*Fhat1);
[eig_Sigv_hat, index_eigen_sort3] = sort(diag(eig_Sigv_hat),'descend');
eigvR = V1(:,index_eigen_sort3);
eig_Sigv_hat=diag(eig_Sigv_hat);

Fhatc1=(eigvR(:,1:iR-iQ)'*Fhat1')';
Fhats1=(eigvR(:,iR-iQ+1:end)'*Fhat1')';
```

```
[V2,eig_Sigv_hat2]=eigs(inv(Fhat2'*Fhat2)*(Fhat2'*Fhat1)*inv(Fhat1'*Fhat1)*Fhat1'*Fhat2);
[eig_Sigv_hat2, index_eigen_sort4] = sort(diag(eig_Sigv_hat2),'descend');
eigvR2 = V1(:,index_eigen_sort4);
eig_Sigv_hat2=diag(eig_Sigv_hat2);

Fhatc2=(eigvR(:,1:iR-iQ)'*Fhat2')';
Fhats2=(eigvR(:,iR-iQ+1:end)'*Fhat2')';

Lambdac1=x1'*Fhatc1/iT;
Lambdac2=x2'*Fhatc1/iT;

xi1=x1-Fhatc1*Lambdac1';
xi2=x2-Fhatc1*Lambdac2';

iX=(xi1*(xi1'))/(iT); % compute eigenvectors form T X T matrix obtained form standardized data:
each T.S. is demeaned and divided by its st.dev computed in time series
[eigvR_uns, eigR_mat_uns] = eigs(iX,iQ);
% Sort eigenvalues (and eigenvectors) from largest to smallest
[eigR, index_eigen_sort] = sort(diag(eigR_mat_uns),'descend');
eigvR = eigvR_uns(:,index_eigen_sort);

iX2=(xi2*(xi2'))/(iT); % compute eigenvectors form T X T matrix obtained form standardized
data:  each T.S. is demeaned and divided by its st.dev computed in time series
[eigvR_uns2, eigR_mat_uns2] = eigs(iX2,iQ);
% Sort eigenvalues (and eigenvectors) from largest to smallest
[eigR2, index_eigen_sort2] = sort(diag(eigR_mat_uns2),'descend');
eigvR2 = eigvR_uns2(:,index_eigen_sort2);

Fhats1=sqrt(iT)*eigvR;
Fhats2=sqrt(iT)*eigvR2;

Fhat1=[Fhatc1,Fhats1];
Fhat2=[Fhatc1,Fhats2];

Lambda1=x1'*Fhat1/(Fhat1'*Fhat1);
Lambda2=x2'*Fhat2/(Fhat2'*Fhat2);
```

```
epshat1=x1-Fhat1*Lambda1';
epshat2=x2-Fhat2*Lambda2';

%F1'*F2*inv(F2'*F2)
%Su11=iN*inv(Lambda1'*Lambda1)*0.9^2;
%Su22=iN*inv(Lambda2'*Lambda2)*0.9^2;

%Gamma12=epsHat1'*epsHat2/iT;
Gamma2=diag(diag((epshat2-mean(epshat2))'*(epshat2-mean(epshat2))/iT));
Gamma1=diag(diag((epshat1-mean(epshat1))'*(epshat1-mean(epshat1))/iT));
Gamma12=diag(diag((epshat1(1:end-1,:)-mean(epshat1(1:end-1,:)))'*(epshat2(2:end,:)-mean(epshat2(2:end,:)

Su11=iN*inv(Lambda1'*Lambda1)*Lambda1'*Gamma1*Lambda1*inv(Lambda1'*Lambda1);
Su22=iN*inv(Lambda2'*Lambda2)*Lambda2'*Gamma2*Lambda2*inv(Lambda2'*Lambda2);
Su12=iN*inv(Lambda2'*Lambda2)*Lambda2'*Gamma12*Lambda1*inv(Lambda1'*Lambda1);
Su21=Su12';

Omega=(Su11(1:iR-iQ,1:iR-iQ)+Su22(1:iR-iQ,1:iR-iQ))^2+Su21(1:iR-iQ,1:iR-iQ)^2+Su12(1:iR-iQ,1:iR-iQ)^2;
%Su12=iN*inv(Lambda1'*Lambda1)*Lambda1'*Gamma12*Lambda2*inv(Lambda2'*Lambda2);
%Su21=iN*inv(Lambda2'*Lambda2)*Lambda2'*Gamma12'*Lambda1*inv(Lambda1'*Lambda1);
%Phi_hat=V'*Phi_hat*V
%Gamma=zeros(iN,iN);
%for i=1:iN-1
%Gamma(i,i+1)=0.81;
%Gamma(i+1,i)=0.81;
%end
%Gamma= cov(epsHat(2:end));
%Gamma= %cov(epsHat(1:end-1,:),epsHat(2:end,:));

%Gamma= (diag(Gamma));
Sigu_hat=Su11(1:iR-iQ,1:iR-iQ)+Su22(1:iR-iQ,1:iR-iQ);
%+2*Su12(1:iR-iQ,1:iR-iQ);%Su12(1,1)-Su21(1,1);%-Su12(1,1)-Su21(1,1);%-Su12(1,1)-Su21(1,1);

%B_Uhat=2*Sigu_hat(1,1);%-Sigu_hat*Phi_hat';
```

```
%SUzero=Sigu_hat(2,2)+Phi_hat(2,1)*Sigu_hat(1,1)*(Phi_hat(2,1)')+...
        %+Phi_hat(2,2)*Sigu_hat(2,1)*(Phi_hat(2,1)')+Phi_hat(2,1)*Sigu_hat(1,2)*(Phi_hat(2,2)')+...

        %+Phi_hat(2,2)*Sigu_hat(2,2)*(Phi_hat(2,2)');


%Omega_U1=2*trace(SUzero*(SUzero')+SUone*(SUone'));

Tstat= iN*sqrt(iT)*((0.5*trace(Omega))^(-0.5))*(...
       sum(sqrt(diag(eig_Sigv_hat(1:iR-iQ,1:iR-iQ))))-iR+iQ +(trace(Sigu_hat))/(2*iN));


end
```

Theorem 4 represents only a further test analogous to the main two just treated, developed as a support, since it was the first one to provide the hoped-for results.

4 The simulations

As the final section of this dissertation, a series of simulations will now be performed, useful for understanding the results to which these lead, as well as the degree of accuracy of both tests. This represents, actually, the main section of the entire thesis, as the latter was conceived as a supporting and complementary work to the research conducted by Professors Carlini and Rubin.

As we could previously see, the approach to the analysis will be carried out through the Monte Carlo simulation method. A Monte Carlo simulation is a model that predicts the likelihood of various outcomes in the presence of intervening random variables. The basis of Monte Carlo simulation is that the probability of different outcomes cannot be determined due to random variable interference. It involves assigning multiple values to an indeterminate variable to produce multiple outcomes, which are then averaged to obtain an estimate. Monte Carlo simulation takes a variable with uncertainty and assigns it a random value. The model is then run and it provides a result. This process is repeated over and over, while assigning many different values to the variable in question. Once the simulation is complete, the results are averaged to provide an estimate. Therefore, Monte Carlo simulations focus on repeating a random sample over and over to obtain a specific result. These types of simulations are widely used in finance, engineering as in econometrics since they help explain the effects of risk and uncertainty in prediction and forecasting models.

4.1 Simulation Design

A further necessary step in describing the setup of simulations concerns the structure of model and matrices to be taken into account. We simulate the observation $y_{i,t}$ for $i = 1, ..., N$, $t = 1, ..., T$ from the following factor model:

$$y_{i,t} = \breve{\lambda}_i' \breve{f}_t + \varepsilon_{i,t} \tag{4.1}$$

The r-dimensional vector f_t follows the VAR(1) process:

$$\breve{f}_t = \breve{\Phi} \breve{f}_{t-1} + \breve{v}_t, \qquad \text{and} \qquad \breve{v}_t = G\eta_t,$$

where $\breve{\Phi}$ is an (r,r) autoregressive matrix. We consider the following two designs for the factors' DGPs:

- Design 1: we set $r = 3$, and simulate the $q = 2$ primitive shocks $\eta_t \sim i.i.d.N(0, I_q)$. Moreover, we set:

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad \text{and} \qquad \Phi = \begin{bmatrix} 0.9 & 0.5 & -0.2 \\ -0.4 & 0.7 & 0.1 \\ 0.1 & 0.2 & 0.6 \end{bmatrix}.$$

As an assumption to ensure a high level of significance we consider 2000 MC replications for each MC design[12].

[12] we have mentioned the term 'each' because within Carlini and Rubin's paper there is an additional design called *Design 2* that involves a structure of different r and q ($r = 5$, $q = 3$).

4.2 Results and Asymptotic Gaussian distribution

Once the main code has finished its calculations, the output that come out will be collected in three different arrays, each labeled 'ResultsTh-' depending on the number of the referenced theorem. The values displayed in these arrays are the mean, the median, the standard deviation and interquartile range[13] of the empirical distribution of the recentered and standardized statistics in Theroem 2 and 3, respectively. The asymptotic distribution of the statistics is always $N(0,1)$ and has interquartile ≈ 1.35. Every statistics needs to be computed by changing $(N,\ T)$ every time. Empirical distributions are obtained by recomputing the statistics with 2000 MC simulations for each Deisgn/DGP.

For example, from the results we have obtained through this last simulation, increasing more and more the size of our panel of data, it is clear how the test is close to the maximum level of accuracy, as evidenced by the following tables:

	1	2	3	4
1	0.1414	0.1214	1.0102	1.4177

(a) Theorem 2 with $N = 2000$ and $T = 2000$

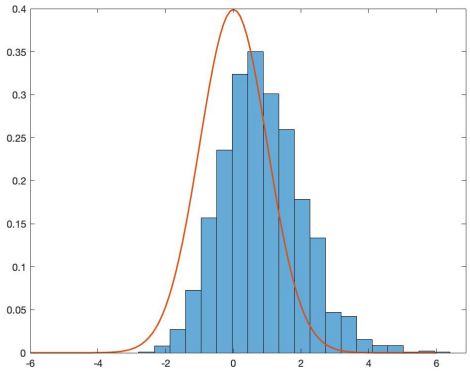
	1	2	3	4
1	-0.1291	-0.1089	1.0098	1.4169

(b) Theorem 3 with $N = 2000$ and $T = 2000$

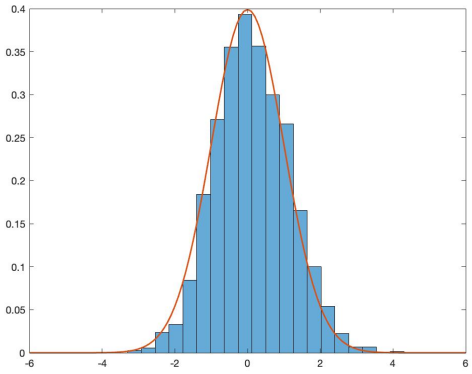
Through the use of Monte Carlo simulations, it was possible to obtain an extremely accurate result. The advantage of this method of approach is just that one that to the increase of the number of simulation the result will tend asymptotically to one known distribution. In order to observe better how much just said it is sufficient to execute a run to the code brought back in section 3.3 in order to observe like in the case in which $N = 30$ and $T = 100$ the result seems to be however correct but a lot less precise.

Once it is clear how the simulation designs are made, it is time to figure out what to expect from the results of these tests. As we've seen by scrolling through the code the goal now should be to compare each histogram that comes out of the simulation to a known distribution, which in our case is represented by a standard Gaussian distribution. To be more precise, the statistics referenced by the bar graph will be denoted $\tilde{\xi}(q)$ and $\tilde{\xi}_c(q)$ depending on whether they are tests conducted from Theorems 2 and 3, respectively. In the figure below we present two cases of empirical distributions for each of the two statistics, computed under the null hypothesis of $q = 3$ primitive shocks from data simulated by the DGP from Design 1 (i.e., the one that had $q = 1$ dynamic factors), overlaid on an asymptotic representation of the standardized Gaussian distribution. What is immediately noticeable is that for small sample sizes such as $N = 50$ and $T = 100$ the two graphs do not match exactly. As the sample size increases as in the case of $N = 1000$ and $T = 2000$, the almost perfect correspondence between the statistics obtained from the empirical simulations of $\tilde{\xi}(q)$ (resp. $\tilde{\xi}_c(q)$) and those of an $N(0,1)$ is evident.

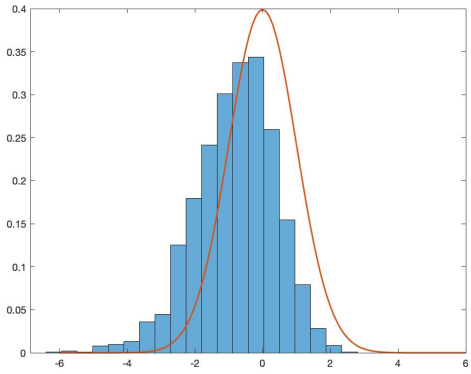
[13] Interquartile range is a measure of statistical dispersion, which is the spread of the data. It is defined as the difference between the 75th and 25th percentiles of the data



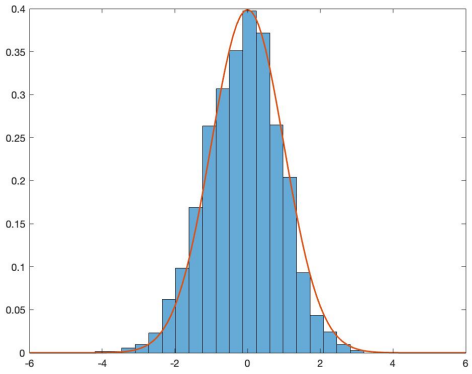
(a) $\tilde{\xi}(q), N = 50$ and $T = 100$



(b) $\tilde{\xi}(q), N = 1000$ and $T = 2000$



(c) $\tilde{\xi}_c(q), N = 50$ and $T = 100$



(d) $\tilde{\xi}_c(q), N = 1000$ and $T = 2000$

Figure 5. Sample distribution of the test statistic $\tilde{\xi}(q)$ and $\tilde{\xi}_c(q)$ for Design 1

The rest of the results obtained within each simulation can be seen in what is referred to as Table II in the paper by Carlini and Rubin (2022), which is also shown below for completeness of information.

TABLE I - Finite sample distribution of the test statistics $\tilde{\xi}(q)$ and $\tilde{\xi}_c(q)$ in Theorems 2 and 3

		Design 1 ($r = 3, \quad q = 2$)							
		Plug-in: Th. 2				*Plug-in: Th. 3*			
N	T	m.	med.	std.	iqr.	m.	med.	std.	iqr.
30	100	1.37	1.23	1.40	1.85	-1.33	-1.20	1.40	1.85
30	200	1.69	1.62	1.35	1.77	-1.67	-1.60	1.35	1.76
30	300	1.99	1.96	1.32	1.81	-2.00	-1.96	1.33	1.82
50	100	0.90	0.80	1.20	1.62	-0.85	-0.75	1.19	1.62
50	200	0.98	0.93	1.17	1.58	-0.96	-0.91	1.17	1.58
50	300	1.17	1.14	1.16	1.61	-1.15	-1.13	1.16	1.61
100	100	0.71	0.66	1.16	1.53	-0.65	-0.61	1.15	1.52
100	200	0.57	0.51	1.12	1.55	-0.53	-0.47	1.11	1.54
100	300	0.61	0.56	1.10	1.42	-0.58	-0.53	1.10	1.42
200	100	0.68	0.59	1.18	1.56	-0.61	-0.52	1.17	1.55
200	200	0.46	0.42	1.07	1.44	-0.42	-0.38	1.07	1.43
200	300	0.44	0.37	1.05	1.43	-0.41	-0.34	1.05	1.43
500	100	1.06	0.84	1.53	1.80	-0.99	-0.77	1.52	1.79
500	200	0.46	0.38	1.12	1.50	-0.41	-0.34	1.12	1.49
500	300	0.32	0.27	1.05	1.44	-0.29	-0.24	1.05	1.43
500	1000	0.26	0.22	1.01	1.42	-0.24	-0.20	1.01	1.41
500	2000	0.32	0.30	1.03	1.39	-0.32	-0.29	1.03	1.39
1000	1000	0.19	0.17	1.02	1.34	-0.17	-0.15	1.02	1.34
1000	2000	0.16	0.14	0.99	1.38	-0.15	-0.13	0.99	1.38
2000	1000	0.20	0.22	0.99	1.31	-0.18	-0.20	0.99	1.31
2000	2000	0.14	0.12	1.01	1.42	-0.13	-0.11	1.01	1.42

An important observation to note is how, although the two tests converge asymptotically to the same values, they do so from opposite "directions". Comparing, in fact, the mean and median values, it is immediate to note that the observed values are of different sign and as the sample is increased these tend to have almost opposite values. This occurs because in the former case the test structure has zero as the lower bound of the variance, below which it is not possible to fall. In the second case, however, our interest falls on canonical correlations whose upper bound is one. That said, it is evident to see that the two distributions are identical, but changed in sign. In contrast, what has just been said is not reflected when comparing the values within the standard deviation and interquantile range columns. On the contrary, the values of the latter differ very little between the tests implemented according to Theorem 2 and 3, providing further empirical evidence of what good has been achieved in terms of results.

4.3 Size and Power of both tests

What we want to find out now is whether the statistical accuracy rate of our tests can be considered significant or not. The most useful tool in this regard is the hypothesis testing procedure, which involves actually testing a hypothesis with the goal of finding evidence through probabilistic reasoning applied to a sample in order to say whether the hypothesis is true or false. Without dwelling further on the notional aspect, what was done was to calculate a series of indicators that in a direct and immediate way provide values capable of highlighting the goodness of the tests, so as to make comparable the level of accuracy of each. Specifically, these indicators belong to the field of application of the so-called size and power of a test. In statistics, the size of a test is the probability of falsely rejecting the null hypothesis H_0. That is, it is the probability of making a Type I error and it is usually represented by the Greek letter alpha (α). In other words, the size of a test is the probability of incorrectly rejecting the null hypothesis whether it is true.
To deal with it, critical alpha values at 1%, 5% and 10% were chosen for both testing Theorem 2 and 3, each time loading the results obtained from the different Monte Carlo simulations:

```
%% Size

clc
clear

load('N_2000_T_2000_MCstatistics')

cv1_Th2 = norminv(0.99)
cv1_Th3 = norminv(0.01)

cv2_Th2 = norminv(0.95)
cv2_Th3 = norminv(0.05)

cv3_Th2 = norminv(0.90)
cv3_Th3 = norminv(0.10)

% Number of times the null hypothesis is rejected under the null hypothesis
Size1_Th2 = sum(Test_Th2 > cv1_Th2)/nMC
Size1_Th3 = sum(Test_Th3 < cv1_Th3)/nMC

Size2_Th2 = sum(Test_Th2 > cv2_Th2)/nMC
Size2_Th3 = sum(Test_Th3 < cv2_Th3)/nMC
```

```
Size3_Th2 = sum(Test_Th2 > cv3_Th2)/nMC
Size3_Th3 = sum(Test_Th3 < cv3_Th3)/nMC
```

Once the procedure for calculating the different sizes is defined, the test power values can be obtained in a trivial way. The statistical power of a binary hypothesis test is the probability that the test correctly rejects the null hypothesis H_0 given an alternative hypothesis H_1 is true. It is usually expressed as $1-\beta$ and represents the probability of a "true positive" detection, which depends on the actual presence of the effect to be detected. Statistical power ranges from 0 to 1, and as the power of the test increases, the probability β of making a Type II error due to incorrectly not rejecting the null hypothesis decreases. Thus, the power of a test is the probability of correctly rejecting the null hypothesis whether it is false.
Before showing the results obtained (comparable also in what in the paper of Carlini and Rubin is reported as Table III) it is reported, as well as for the size, the Matlab code through which all the calculations have been carried out:

```
%% Power

clc
clear

load('N_2000_T_2000_MCstatistics')

cv1_Th2 = norminv(0.99)
cv1_Th3 = norminv(0.01)

cv2_Th2 = norminv(0.95)
cv2_Th3 = norminv(0.05)

cv3_Th2 = norminv(0.90)
cv3_Th3 = norminv(0.10)

% 1 - probability of accepting under the alternative hypothesis
Power2_Th2 = 1-sum(Test_Th2 < cv2_Th2)/nMC
Power2_Th3 = 1-sum(Test_Th3 > cv2_Th3)/nMC
```

It is possible to notice that in this specific case the power of the test has only been calculated taking into consideration the critical value 2, that is the one at 5%.

TABLE II - Empirical size and power of the tests of the number of dynamic factors q for Design 1

		Plug-in: Th. 2				*Plug-in: Th. 3*			
		size			*power*	*size*			*power*
N	T	1%	5%	10%	5%	1%	5%	10%	5%
30	100	0.23	0.39	0.49	1	0.23	0.38	0.48	1
30	200	0.29	0.49	0.60	1	0.29	0.48	0.60	1
30	300	0.38	0.59	0.70	1	0.39	0.59	0.70	1
50	100	0.12	0.25	0.35	1	0.11	0.24	0.33	1
50	200	0.12	0.28	0.38	1	0.12	0.27	0.37	1
50	300	0.16	0.33	0.46	1	0.15	0.38	0.45	1
100	100	0.09	0.20	0.30	1	0.08	0.19	0.28	1
100	200	0.06	0.17	0.27	1	0.06	0.17	0.26	1
100	300	0.07	0.17	0.25	1	0.07	0.16	0.25	1
200	100	0.09	0.20	0.29	1	0.08	0.19	0.27	1
200	200	0.04	0.14	0.22	1	0.04	0.13	0.21	1
200	300	0.04	0.13	0.21	1	0.04	0.13	0.20	1
500	100	0.17	0.29	0.37	1	0.16	0.27	0.35	1
500	200	0.06	0.15	0.21	1	0.05	0.13	0.20	1
500	300	0.03	0.11	0.18	1	0.03	0.10	0.17	1
500	1000	0.02	0.09	0.17	1	0.02	0.09	0.16	1
500	2000	0.03	0.10	0.19	1	0.03	0.10	0.19	1
1000	1000	0.02	0.08	0.15	1	0.02	0.08	0.15	1
1000	2000	0.02	0.07	0.13	1	0.01	0.07	0.13	1
2000	1000	0.02	0.07	0.14	1	0.02	0.07	0.14	1
2000	2000	0.01	0.07	0.14	1	0.01	0.07	0.14	1

In this latter case, we are under the alternative hypothesis (H_1) and we want to understand how many times we can reject the H_0. On the contrary before in the calculation of the size we were under the null hypothesis. To calculate the power of a test means to understand the probability, being under the alternative, how much we reject the null hypothesis. The result of 1, which comes out of 1 minus the probability of accepting under the alternative hypothesis, is an excellent value, hence the power is maximum and the test is extremely accurate.

All this evidence show that our statistics work more than well.

5 Conclusions

What has been described so far undoubtedly demonstrates the accuracy of the procedures that have been conducted as part of this analysis. The results that were provided are similar to those that can be found within Carlini and Rubin's paper *New Tests and Estimators for Common Dynamic Factors* (2022). In fact, as stated at the outset, the purpose of this dissertation was precisely to proceed in parallel with the above-mentioned work, as confirmed by the fact that this was done starting from the structure of the model and the DGP itself to the final section regarding simulations. It should also be pointed out that the entire procedure was based on a DGP that replicates, in an initially random manner, any type of dataset that can be identified through an autoregressive type model. This was made possible from the combination of a static factor model combined with a stationary dynamic VAR(1) model for f_t factors. The reasoning that has gone into this type of application ensures that the theses developed, as well as the way the model was reasoned and structured, allow the entire modality to be applied to any real-world dataset, enabling concrete hypotheses to be made and empirically tested for veracity.

In addition, the first part was developed taking into account the work conducted by Bai and Ng during their research on factor models. These provided a working basis on which hypotheses and models could be developed, having as the main reference the problems they encountered. In particular, we focused on two aspects relevant to the writing of the paper itself: on the one hand, identification problems were discussed, and on the other, an introduction to the estimation of the number of shocks in factor models was provided. The approach taken to overcome this type of problem was twofold. The code implemented through the use of Matlab involved two types of independent tests that led to the same results. The first method concerned the study of the eigenvalues of the covariance matrix of the VAR factor innovations. By testing for the presence of eigenvalues with a value of zero, it was possible to advance the hypothesis that the $\breve{\Sigma}_v$ matrix had reduced rank q, that is, the number of all those eigenvalues within it that had strictly positive values. We then introduced the concept of canonical correlation. The second approach was based precisely on identifying those canonical correlations between the factors f_t and their lagged values f_{t-1} that had a value exactly equal to one.

A final crucial aspect that was involved in the structure of our test was the Monte Carlo simulation analysis. The use of Monte Carlo simulations was essential in making it possible to assess the adequacy of the standard Gaussian asymptotic distributions of $\tilde{\xi}(q)$ and $\tilde{\xi}_c(q)$, in Theorems 2 and 3, respectively, to approximate their small-sample counterparts. For small sample sizes as $N = 50$, and $T = 100$, empirical distribution were not precisely describing what we would have expected. As the sample sizes grow to $N = 2000$, and $T = 2000$, the empirical distributions of $\tilde{\xi}(q)$ (resp. $\tilde{\xi}_c(q)$) have empirical mean and standard deviation of 0.14 and 1.01 (resp. -0.13 and 1.01), respectively, and almost perfectly overlaps with the asymptotic standard Gaussian distribution. All the simulated values were stored in Table II, which reports, for different sample sizes N and T, the values of the mean, median, standard deviation and interquartile range of the simulated distributions of $\tilde{\xi}(q)$ and $\tilde{\xi}_c(q)$ in Theorems 2 and 3, respectively when the data are simulated from the DGP in what

Carlini and Rubin in their paper chose as Design 1 ($r = 3$ and $q = 2$). As a final result, we had to test the consistency rate of the values reported in the latter table. To this end, Table III reports the empirical dimension and power of the one-sided test for the null hypothesis of primitive shocks q, based on the $\tilde{\xi}(q)$ and $\tilde{\xi}_c(q)$ statistics of Theorems 2 and 3. Empirical size was evaluated at the 1%, 5% and 10% significance levels: the null hypothesis of q = 2 for Design 1 is rejected when the value of the test statistic calculated on the simulated data is less than the 1%, 5% and 10% quantiles of the asymptotic distribution of the test statistic (which we assumed to be a standard Gaussian) for each MC simulation for Design 1. The empirical power of the test was calculated as the empirical frequency of rejection of the null hypothesis of $q = 1$ dynamic factors against the alternative of a number of common factors strictly greater than 1. Empirical power was evaluated for a test performed at a 5% significance level. That said, what has been demonstrated irrefutably reflects the hoped-for results with an incredible level of accuracy. This can only leave one with the knowledge and pride of having been able to contribute to an extremely innovative and experimental work for which no theory was developed before Carlini and Rubin's *New Tests and Estimators for Common Dynamic Factors.*

6 References

Amengual, D., and M. Watson (2007): "Consistent Estimation of the Number of Dynamic Factors in a Large N and T Panel", *Journal of Business and Economic Statistics*.

Andreou, E., P. Gagliardini, E. Ghysels and M. Rubin (2019): "Inference in Group Factor Models with an Application to Mixed-frequency Data", *Econometrica*.

Bai, J., and S. Ng (2002): "Determining the Number of Factors in Approximate Factor Models", *Econometrica*.

——— (2007): "Determining the Number of Primitive Shocks in Factor Models", *Journal of Business and Economic Statistics*.

Carlini, F., and M. Rubin (2022): "New Tests and Estimators for Common Dynamic Factors".

Carlini, F., and P. Gagliardini (2022): "Instrumental variables inference in a VAR model with dynamic latent factors".

Forni, M., M. Hallin, M. Lippi, and L. Reichlin (2000): "The generalized dynamic-factor model: Identification and estimation", *Review of Economics and Statistics*.

Lewbel, A. (2019): "The Identification Zoo - Meanings of Identification in Econometrics".

Onatski, A. (2009): "Testing Hypotheses About The Number of Factors in Large Factor Models" *Econometrica*.

Stock, J., and M. Watson (2002): "Forecasting Using Principal Components From a Large Number of Predictors" *Journal of the American Statistical Association*

7 Summary

In this dissertation, a new working method is basically described with regard to the econometric field of dynamic factor models. The analysis conducted was carried out entirely with the aim of providing support for the work that has been referenced throughout, namely Carlini and Rubin, *New Tests and Estimators for Common Dynamic Factors* (2022). The initial aim of the two Professors was to solve a problem that had been recurring in the econometric literature for years and provide a solution through the research of new tests that would allow precisely to test factor estimation in the above-mentioned models. This dissertation sought to provide a simplified approach to the same topics covered in the paper, as the ultimate goal of the entire work was not so much to replicate the work conducted by both Professors but to take advantage of the research conducted to carry out a large number of simulations that could confirm the veracity of what was stated in the paper itself. Before reaching this point, however, a period of study and introduction to such a topic was necessary, which took the form of a series of gradually more and more challenging preparatory learning tasks. This part was accompanied by reading and comparing with a series of papers among which the series of research conducted by Bai and Ng certainly played a prominent role. Their papers were of great help in terms of approach and interpretation of the topic, as well as fundamental, thanks to the findings reported in them, to the writing of Carlini and Rubin's paper. In one of the first sections of this paper, on the basis of a couple of them, a comparison and analysis of the problems encountered by Bai and Ng themselves to which Carlini and Rubin were able to find an entirely innovative solution was addressed. Such a solution is basically represented by one test, or rather two tests. In fact, although the code to run is unique, it contains within it the implementation of two different tests that eventually asymptotically arrive at virtually the same results. These two tests focus in a parallel manner on the study of two concepts on which their respective analyses are structured. Specifically we are referring to the number of eigenvalues exactly equal to zero and consequently a reduction in the rank of the variance-covariance matrix Σ_v on the one hand, and the number of canonical correlations exactly equal to one on the other. Once introduced and explained through appropriate equations and codes, the tests in question (implemented through the use of Matlab) are ready to be used within subsequent simulations with the aim of asymptotically deriving the series of outputs from them.