

UNIVERSITÀ DI PISA  
SCUOLA SUPERIORE DI STUDI UNIVERSITARI E DI  
PERFEZIONAMENTO SANT'ANNA



Master of Science in ECONOMICS

TESI DI LAUREA MAGISTRALE

# Identifying shocks in Structural Vector Autoregressions and Factor Models

an Application to Agent-Based Models Validation

**Supervisor**

Prof. Alessio MONETA

**Candidate**

Daniele COLOMBO

Academic Year 2020-2021

# Abstract

Recent developments in the VAR literature have demonstrated that it is possible to identify structural shocks by using only the distribution of reduced-form shocks and taking advantage of the information provided by its higher-order moments, making shock identification possible by relying solely on the assumptions of independence and non-Gaussianity of the structural shocks. However, the identification schemes proposed so far, which are rooted in independent component analysis, rely on additional assumptions to solve the indeterminacy of the permutation and scaling of the causal relations that this computational technique entails. After an overview of some popular identification strategies, this work introduces NGSi, a data-driven algorithm capable of performing shock identification without relying on such auxiliary assumptions. The key idea on which it is based is that it can be inferred from the data which assumptions are likely to hold and the most appropriate (and precise) identification scheme can be implemented accordingly. The performance of the algorithm is then assessed in a wide variety of settings via an extensive simulation study. Furthermore, this work proposes a new method to empirically validate simulation models that generate artificial time series data comparable with real-world data. The approach, which is based on the comparison of the causal structures estimated from the artificial and the real-world data, extends previous research by exploiting structural factor models, which, compared to standard SVARs, allow to consider a larger informative set, thereby leading to a more comprehensive validation assessment. This methodology is able to address both the problem of evaluating theoretical simulation models against the data and the problem of comparing different models in terms of their empirical reliability. Finally, an application of the validation procedure to the agent-based macroeconomic model proposed by Dosi et al. (2015) is provided.

**Keywords:** Data-driven shock identification, dynamic factor models, model validation, agent-based models, time-series analysis.

**JEL Codes:** C32, C38, C52.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Validation method</b>	<b>7</b>
2.1	Dataset uniformity and analysis of ABM properties . . . . .	7
2.2	The Vector Autoregression Model . . . . .	10
2.2.1	Impulse response structural analysis . . . . .	13
2.3	The Factor Augmented VAR model . . . . .	16
2.3.1	Estimation of the static factors . . . . .	17
2.4	The Dynamic Factor Model . . . . .	19
2.5	Validation assessment . . . . .	22
<b>3</b>	<b>Identification strategy</b>	<b>26</b>
3.1	The recursive identification scheme . . . . .	28
3.2	Identification via independent component analysis . . . . .	31
3.2.1	A simple ICA identification algorithm . . . . .	33
3.2.2	VAR-LiNGAM . . . . .	34
3.3	Testing the identification algorithms against different mixing matrices . . . . .	36
3.3.1	Simulation results . . . . .	41
3.4	NGSI: a data-driven algorithm for shock identification . . . . .	46
3.4.1	The case of Gaussian residuals . . . . .	56
3.5	Summary of the section . . . . .	57
<b>4</b>	<b>Application to the K+S model by Dosi et al. (2015)</b>	<b>60</b>
4.1	The model . . . . .	60
4.2	The data . . . . .	61
4.2.1	Analysis of the factors . . . . .	64
4.3	Dataset uniformity and ABM properties . . . . .	67
4.4	VAR validation results . . . . .	69
4.5	FAVAR validation results . . . . .	75
4.6	DFM validation results . . . . .	80
<b>5</b>	<b>Conclusions</b>	<b>82</b>
<b>A.</b>	<b>Assessment of the identification schemes</b>	<b>93</b>
<b>B.</b>	<b>Performance of the NGSI and related algorithms</b>	<b>95</b>
<b>C.</b>	<b>NGSI algorithm: an R implementation</b>	<b>97</b>
<b>D.</b>	<b>“K+S” model time series</b>	<b>100</b>
<b>E.</b>	<b>Parametrization of the “K+S” model</b>	<b>101</b>
<b>F.</b>	<b>Factor analysis: tables and figures</b>	<b>102</b>
<b>G.</b>	<b>Impulse Response Functions</b>	<b>106</b>

# List of Tables

1	Expectations on performance of identification schemes . . . . .	38
2	Fractions of correctly estimated recursive orderings. . . . .	52
3	Fraction of big diagonal instances . . . . .	53
4	Fraction of VARs displaying Gaussianity . . . . .	57
5	Equilibrium and ergodicity tests results . . . . .	68
6	Augmented Dickey-Fuller tests. . . . .	71
7	Normality tests on VAR residuals . . . . .	71
8	VAR validation results . . . . .	75
9	Normality tests on FAVAR residuals . . . . .	77
10	FAVAR validation results . . . . .	80
11	DFM validation results . . . . .	81
A.12	Total squared error . . . . .	93
A.13	Percentage of correct sign . . . . .	93
A.14	Percentage of close-to-correct size . . . . .	93
A.15	Percentage of correct contemporaneous relations . . . . .	94
A.16	Overall similarity results . . . . .	94
A.17	Assessment of similarity measure . . . . .	94
A.18	VAR-LiNGAM estimation of recursiveness and variable ordering . . .	94
B.19	Benchmark algorithm performance . . . . .	95
B.20	Refined algorithm performance . . . . .	95
B.21	NGSI algorithm performance . . . . .	96
B.22	Detection of structure $f$ results . . . . .	96
B.23	NGSI algorithm performance, Gaussian residuals . . . . .	96
D.24	Variables obtained from the “K+S” model . . . . .	100
E.25	Model parametrization . . . . .	101
F.26	Load of real-world series on first $r$ factors . . . . .	102
F.27	Load of agent-based series on first $r$ factors . . . . .	103

## List of Figures

1	Equilibrium and ergodicity tests . . . . .	10
2	Time series used in VAR validation . . . . .	69
3	VAR residuals distribution . . . . .	72
4	VAR forecast error variance decompositions. . . . .	74
5	Estimated latent factors . . . . .	76
6	FAVAR residuals distribution . . . . .	78
7	FAVAR forecast error variance decompositions. . . . .	79
C.8	Estimation via NGSI, 1 . . . . .	97
C.9	Estimation via NGSI, 2 . . . . .	97
C.10	Estimation via NGSI, 3 . . . . .	98
C.11	Estimation via NGSI, 4 . . . . .	98
C.12	Estimation via NGSI, 5 . . . . .	99
C.13	Estimation via NGSI, 6 . . . . .	99
F.14	Estimated static factors . . . . .	104
F.15	Variance explained by the factors . . . . .	104
F.16	Byplot of the first two static factors . . . . .	105
F.17	Explanatory power of the factors in each series . . . . .	105
G.18	VAR impulse responses . . . . .	106
G.19	FAVAR impulse responses . . . . .	107
G.20	DFM impulse responses . . . . .	108

# 1 Introduction

The ambition of this work is twofold. On the one hand, it contributes to the VAR literature by investigating the ways in which shock identification can be performed in a fully data-driven fashion, relying on a minimal set of assumptions, and by proposing a novel identification framework. On the other hand, it contributes to the literature on the validation of macroeconomic models by developing a validation procedure based on structural factor models, which can be identified by means of our proposed identification algorithm.

Ever since the Vector Autoregressive model has been introduced in the macroeconomic literature (Sims 1980), each and every implementation of this econometric technique has been met with the problematic nature of shock identification. Indeed, to obtain a structural representation of the economic mechanisms that they try to capture with structural VARs, researchers have frequently found themselves in a position in which they have to rely on a number of assumptions, mostly derived from economic theory or based on a description of the economic system which is often overly simplified. Moreover, most of these assumptions cannot be tested, leaving broad discretion in the choice of the shock identification strategy.

Nonetheless, recent developments in the VAR literature have demonstrated that it is possible to identify structural shocks by using only the distribution of reduced-form shocks and taking advantage of the information provided by its higher-order moments, making shock identification possible without the kinds of assumptions traditionally used in the literature (see e.g. Hyvärinen et al. 2010, Moneta et al. 2013 and Lanne et al. 2017). Yet, these approaches, which have placed emphasis on data-driven algorithms rooted in independent component analysis, do rely on a set of less theory-driven but nonetheless heavy assumptions, such as independence and non-Gaussianity of the structural shocks. Furthermore, a crucial issue when recurring

to independent component analysis is given by the indeterminacy of the permutation and scaling of the causal relations, which is typically solved by introducing additional assumptions. Building on these contributions, in the present work we investigate the ways in which shock identification can be performed in a fully data-driven fashion, without relying on such auxiliary assumptions. Moreover, we inquire whether the non-Gaussianity assumption (conveniently being testable) is likely to hold in real-world applications.

Once the identification task can be performed in a credible manner, structural VARs become versatile tools that can be employed in a wide variety of settings, which is the reason why they have become one of the most popular econometric analysis instruments when it comes to dealing with time series data (Kilian and Lütkepohl 2017), and recent contributions have also seen them being employed for the validation of macroeconomic models. Indeed, Guerini and Moneta (2017) have proposed a validation procedure which focuses on the estimation and comparison of the causal structures underlying the real-world time series and the causal structures embedded in the model under validation. Compared to the mere ex-post ability to reproduce a number of stylized facts, often used as main validation routine (see e.g. Fernández-Villaverde et al. 2016 and Lamperti et al. 2018), this constitutes a significant improvement in the validation procedure, since a good matching between the causal structures incorporated in the model and the causal structures underlying the real-world data can provide a better support to the policy statements drawn from the model. Furthermore, this procedure offers a solution to both the issues of comparing different models and of validating a given model against the empirical data.

However, a critical aspect of this methodology regards the choice of the variables to include in the VAR and whether these can be considered a sufficient information

set to recover the relevant causal structures. Indeed, a well-known problem in the traditional specification of VAR models is that only a small amount of variables can be directly included in the model, as the number of parameters that need to be estimated rapidly increases with the number of variables. As a consequence, the choice of what variables to consider is somewhat subjective and it constraints the researcher to exploits a thinner informative set than that of Central Banks and policy-makers.

In their seminal work, Bernanke et al. (2005) proposed a Factor-Augmented Vector Autoregression (FAVAR) approach to overcome the problem of including a large amount of information in a VAR model. This method allows to account for a large part of the information contained in the data in a parsimonious way and it facilitates the task of choosing which variables to include in the model, since a big part of the information is summarized by the factors. In addition, the factor approach sometimes allows to get a more precise measure of given quantities which have a clear theoretical definition but cannot be distinctly observed in reality. For instance, Boivin and Giannoni (2006) employ factor models to directly deal with measurement errors while estimating DSGE models for which a one-to-one correspondence between the theoretical concepts and the actual observed variables does not exist. For these reasons, Giannone et al. (2006) and Lippi (2019) put forward that a structural factor model should be the natural approach to use for the validation of macroeconomic models. Since the common components can be interpreted as a cleaner version of the variables that should be considered for structural analysis, hence free of measurement error, the factor approach allows for the recovery of structural shocks that are not contaminated by non-corresponding shocks (contamination which is instead possible in the case of simple VARs).

This work addresses this problem by building on the contribution of Guerini and

Moneta (2017) and expanding their validation procedure to a factor-based approach. We here focus on the validation of agent-based models, for which an application is also presented. Nevertheless, our methodology can be easily generalized to any simulation model able to generate enough time series to justify the use of a factor approach.

Over the last decade, Agent-Based Models (ABMs) have emerged as a possible alternative to the RBC/DSGE paradigm in macroeconomics, as an attempt to overcome some of the seemingly unrealistic assumptions that characterize the agents' behavior in this class of models. The ABM approach, which has rapidly gained attention in recent years, proposes to model the macroeconomic structure as an emergent property arising from the interaction of heterogeneous and bounded rational economic agents. However, a serious methodological issue that ABMs experience today is their ambiguous relationship with the empirical evidence. As a matter of fact, any macroeconomic model that attempts to represent real-world phenomena must be empirically reliable and this is particularly important if its aim is that of informing decision makers. Indeed, Agent-Based Models can be thought of as artificial economies that can be used as laboratories to conduct policy experiments. It is therefore a very relevant task to assess the degree to which they are able to represent real-world mechanisms that give rise to observable phenomena. A common approach to validate ABMs is that of evaluating a model's ex-post ability to reproduce a set of stylized facts, that is, a set of robust statistical properties of the real-world data (Fagiolo and Roventini 2008). Indeed, calibration and replication of statistical properties have been the most common methods employed by the ABM community in order to link the model to the data (Windrum et al. 2007), and they are also resorted to for other macroeconomic models, such as DSGEs. However, as pointed out by Brock (1999), this is not a sufficiently severe test since stylized facts

are “unconditional objects”, or statistical properties of a stationary process, which do not convey information on the underlying data generating mechanism. To put it in simpler terms, given an outcome of the model, there might be more than one underlying data generating process able to replicate it and compatible with the data. That is, replication does not necessarily imply explanation, and in this context a more robust validation procedure is required.

The first part of the work will outline our proposed validation procedure, based on the comparison of the causal structures embedded in the model under validation with the ones found in the real-world data. This is done on three different levels, corresponding to three different structural models, which we introduce: the Vector Autoregressive model, the Factor Augmented Vector Autoregressive model and the Dynamic Factor model.

The second part will be entirely dedicated to the problem of shock identification. After an overview of some popular identification strategies, we will test the performance of each of these when different assumptions on the structure of the matrix of contemporaneous relations hold. The results of this extensive simulation study will serve as a basis for the implementation of NGSi, a data-driven algorithm that performs shock identification by inferring from the data which assumptions are likely to hold and accordingly applying the most appropriate identification scheme. Then, after having defined some appropriate metrics, we will test the performance of the algorithm in a wide variety of settings.

The third part will provide an application of the proposed validation procedure to the “K+S” agent-based model by Dosi et al. (2015). Prior to the validation step, we will present the model and study the properties of the agent-based and the real-world datasets by examining the factors that can be extracted from each.

Finally, the last section will be dedicated to the conclusions that can be drawn

about the validation of the model and the potential of our identification approach, as well as the further development our investigation points to. Seven appendices follow with additional figures, tables and technical details.

## 2 Validation method

In this section, we describe our proposed validation procedure, which compares the causal structures embedded in the agent-based model with the ones found in the real-world data. It is here presented for agent-based models but it is easily generalizable to any macroeconomic model able to generate a large amount of time series.

The procedure is composed of several steps. We begin by applying some suitable transformations to the agent-based data (denoted  $V_{AB}$ ) in order to make it directly comparable to the real-world data (denoted  $V_{RW}$ ). We then analyze the emergent properties of the series produced by the simulated model, notably equilibrium and ergodicity. Then, we proceed to the core of the validation procedure, consisting of the estimation of a reduced-form model and the identification of its structural form by means of NGSF, a data-driven algorithm which will be introduced in section 3. Finally, we compare the estimated causal structures found in the agent-based and in the real-world data by means of suitable distance measures.

We repeat the exercise three times, implementing three different models. We first perform a benchmark validation by means of a standard VAR model, replicating the exercise carried out by Guerini and Moneta (2017), with minor differences. We then extend the framework to factor models by first implementing a Factor Augmented VAR and then a Dynamic Factor Model, which constitute “more severe” validation procedures. We also study the properties of the agent-based and the real-world datasets by performing an in-depth analysis of the factors that can be extracted from these datasets.

### 2.1 Dataset uniformity and analysis of ABM properties

The first step of our proposed validation method consists of rendering the agent-based and the real-world datasets comparable. In general, the number of time series

in each of the two datasets (respectively denoted  $K_{RW}$  and  $K_{AB}$ ) and their length, that is, the number of observations in the series (respectively denoted  $T_{RW}$  and  $T_{AB}$ ), might differ. Furthermore, for the agent-based data, we dispose of  $M$  Monte Carlo simulations while for the real-world dataset we dispose of a single realization of the data generating process. We therefore have that the dimensions of the two datasets are:

$$\begin{cases} \dim(V_{RW}) = 1 \times K_{RW} \times T_{RW} \\ \dim(V_{AB}) = M \times K_{AB} \times T_{AB} \end{cases}$$

This means that in practice, for the real world we observe only one realization of the variables of interest for a period of length  $T_{RW}$  while for the simulated data we have  $M$  Monte Carlo realizations of the variables, for a period of length  $T_{RW}$ .

It often holds true that  $T_{AB} > T_{RW}$ , as for agent-based models we can possibly generate an infinity of observations. As it is customary in the agent-based literature (see e.g. Caiani et al. 2016, Dosi et al. 2019 and Fagiolo et al. 2020), we discard the initial  $T_{AB} - T_{RW}$  observations of the simulated data. In addition to making the datasets uniform, this practice has the advantage of avoiding the risk of capturing the effects present in the transient period, during which the model might not yet display its true dynamic but still be dependant on the choice of the initial conditions. This is beneficial in particular if the process is ergodic, which heuristically means that it is asymptotically independent. That is, that two distant observations are almost independently distributed. In other words, ergodic processes with different initial conditions, which in agent-based models corresponds to different random seeds, have asymptotically convergent properties, since the process will eventually “forget” the past (Grazzini 2012; Windrum et al. 2007).

Conversely, the availability of many Monte Carlo realizations is not an issue but an advantage, since it allows the pairwise comparison of each run with the unique

empirical realization. The results of these comparisons can then be averaged over all the Monte Carlo realizations to obtain a validation outcome robust to the variability of the data generating process entailed in the model. Finally, the magnitude of the time series is harmonized by applying suitable transformations to the variables. Since the concern of most ABMs is the replication of stylized facts, such as distributions and variations but not levels, the scale of the time series is not perceived as an issue in general by the ABM community. However, in our application, this may cause comparability issues with respect to the real-world counterpart and it is therefore necessary to rescale most variables by applying appropriate transformations.

After the transient period has been discarded, as pointed out in Grazzini (2012), in order to be a good proxy of the data, the model must be in a statistical equilibrium state in which the properties of the analyzed series are constant. In particular, the series (or a transformation of them) must have distributional properties that are time-independent. We further require that the model series are ergodic, that is that the simulated observations are random draws from a multivariate stochastic process. Since we dispose of  $M$  Monte Carlo realizations, these assumptions can be tested directly. This is done by collecting in an  $M \times T$  matrix all the observations of a given variable. The  $1 \times T$  rows are called *samples* and the  $M \times 1$  columns are called *ensembles*.  $F_t(Y_k)$  denotes the empirical cumulative distribution function of an ensemble while  $F_m(Y_k)$  indicates the empirical cumulative distribution function of a sample.

As illustrated in Fig.1, ergodicity and statistical equilibrium are tested by performing a series of Kolmogorov-Smirnov iterative tests on the following hypotheses:

$$F_i(Y_k) = F_j(Y_k), \quad \text{for } i, j = 1, \dots, T \quad i \neq j \quad (2.1.1)$$

$$Y^k = \left( \begin{array}{c|c|c|c} y_{1,1} & y_{1,2} & \cdots & y_{1,T} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ y_{M,1} & y_{M,2} & \cdots & y_{M,T} \end{array} \right)$$

$$Y^k = \left( \begin{array}{c|c|c|c} y_{1,1} & y_{1,2} & \cdots & y_{1,T} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ y_{M,1} & y_{M,2} & \cdots & y_{M,T} \end{array} \right)$$

Figure 1: *The elements of comparison when testing for statistical equilibrium (above) and ergodicity (below).*

$$F_i(Y_k) = F_j(Y_k), \quad \text{for } i = 1, \dots, T \quad j = 1, \dots, M \quad (2.1.2)$$

Prior to performing the ergodicity and equilibrium tests we stationarize the variables via appropriate transformations. These can be chosen following the procedure proposed by M. McCracken and Ng (2020). This proceeds by first deciding which variables to treat in levels and which variables to treat in log-levels and transforming them accordingly. Then, stationarity is iteratively tested by means of Augmented Dickey-Fuller tests for unit roots and each series is differentiated as long as the test rejects the null hypothesis of stationarity. The number of differentiations necessary to reach stationarity is taken as the estimate of the order of integration of each series. We consider a 95% confidence level.

## 2.2 The Vector Autoregression Model

After having rendered the agent-based and the real-world data uniform and having analyzed the emergent properties of the agent-based series, we can proceed to the estimation of the first model we consider: the VAR. We first estimate the reduced-

form model and then, after the identification step, the structural model, obtaining the structural coefficients and the impulse response functions. In section 2.5 we will then illustrate how these can be used to validate the agent-based model.

A Vector Auto Regression model is a model where each variable is allowed to be influenced by its lagged values, along with the lagged values of all other variables (up to lag  $p$ ), and an error term. This can be expressed by the following system of equations:

$$\begin{aligned}
y_{1,t} &= \phi_{11,1}y_{1,t-1} + \phi_{12,1}y_{2,t-1} + \cdots + \phi_{1k,1}y_{k,t-1} + \cdots + \phi_{1k,p}y_{k,t-p} + u_{1,t} \\
y_{2,t} &= \phi_{21,1}y_{1,t-1} + \phi_{22,1}y_{2,t-1} + \cdots + \phi_{2k,1}y_{k,t-1} + \cdots + \phi_{2k,p}y_{k,t-p} + u_{2,t} \\
&\dots \\
y_{k,t} &= \phi_{k1,1}y_{1,t-1} + \phi_{k2,1}y_{2,t-1} + \cdots + \phi_{kk,1}y_{k,t-1} + \cdots + \phi_{kk,p}y_{k,t-p} + u_{k,t}
\end{aligned} \tag{2.2.1}$$

which can be conveniently summarized by the expression

$$y_t = \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \cdots + \Phi_p y_{t-p} + u_t \quad \text{for } t = 1, \dots, T, \tag{2.2.2}$$

where  $k$  is the number of variables,  $y_t$  is a  $(k \times 1)$  vector of variables,  $p$  is the number of lags  $u_t$  is a  $(k \times 1)$  vector of reduced-form residuals (or shocks) and  $\Phi_1, \dots, \Phi_p$  are  $(k \times k)$  matrices of reduced-form coefficients. This can be equivalently rewritten in matrix form as

$$Y = L(Y)\Phi'_1 + L^2(Y)\Phi'_2 + \cdots + L^p(Y)\Phi'_p + U, \tag{2.2.3}$$

where  $L$  is the lag operator and the  $Y$  and  $U$  matrices are obtained by stacking the  $y_t$  and  $u_t$  vectors in the following way:

$$Y := \begin{bmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_T \end{bmatrix}, \quad U := \begin{bmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_T \end{bmatrix}.$$

This is called reduced-form VAR(p) model and there are several criteria to estimate the optimal number of lags, such as the Aikaike Information criterion (AIC), the Bayesian information criterion (BIC) and the Hannan Quinn criterion (HQ). Estimates of the  $\Phi$  and  $U$  matrices can be easily obtained via ordinary least squares. When performing structural analysis, we are interested in answering questions of the kind “how are the other variables affected by a shock in a given variable, *all else constant*?”. However, answering these types of questions is not possible simply with the reduced form coefficients and shocks since we have that the latter are not orthogonal, that is, we have  $E[u_t u'_t] \neq I_k$ .

For this reason, we model the VAR as a “structural VAR” by means of the *matrix of contemporaneous relations*  $\Gamma_0$ , which we multiply to both sides to get

$$\begin{aligned} \Gamma_0 y_t &= \Gamma_0 \Phi_1 y_{t-1} + \Gamma_0 \Phi_2 y_{t-1} + \cdots + \Gamma_0 \Phi_p y_{t-p} + \Gamma_0 u_t && \text{for } t = 1, \dots, T \quad (2.2.4) \\ &= \Gamma_1 y_{t-1} + \Gamma_2 y_{t-1} + \cdots + \Gamma_p y_{t-p} + w_t \end{aligned}$$

where  $w_t = \Gamma_0 u_t$  are the “structural shocks” and  $\Gamma_0$  is such that these are orthogonal, that is, we have  $E[w_t w'_t] = I_k$ .  $\Gamma_1, \dots, \Gamma_p$  are called matrices of “structural” coefficients. Eq.2.2.4 can be equivalently rewritten in matrix form as

$$Y \Gamma'_0 = L(Y) \Gamma'_1 + L^2(Y) \Gamma'_2 + \cdots + L^p(Y) \Gamma'_p + W \quad (2.2.5)$$

If we take a closer look at the expression  $\Gamma_0 y_t$ , we can appreciate what we mean by “contemporaneous relations”: when a variable changes value, at the same time

the value of the other variables changes by that same amount multiplied by the respective entries of  $\Gamma_0$ . This matrix is also referred to as *un-mixing matrix*, while its inverse  $\Gamma_0^{-1}$  is the *mixing matrix*.

However, there exist infinite matrices that render the  $w_t$ 's orthogonal. The problem of finding the appropriate  $\Gamma_0$  (and hence also finding the matrices  $\Gamma_1, \dots, \Gamma_p$ ) is called the *identification problem*, which we will address in section 3, where we propose a data-driven algorithm able to retrieve the  $\Gamma_0$  matrix. In matrix form, the structural residuals can be retrieved as  $W = U\Gamma_0'$ , with

$$W := \begin{bmatrix} w'_1 \\ w'_2 \\ \vdots \\ w'_T \end{bmatrix}$$

### 2.2.1 Impulse response structural analysis

Once the VAR has been *identified*, it is possible to recover the impulse response functions. Eq.2.2.4 can be rewritten compactly in vector form as

$$\Gamma(L) \begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \dots \\ y_{k,t} \end{bmatrix} = w_t, \quad (2.2.6)$$

where  $\Gamma(L) = \Gamma_0 - \Gamma_1 L - \dots - \Gamma_p L^p$  is a  $k \times k$  matrix operator and  $w_t$  is a  $k$ -dimensional vector of structural shocks, such that  $\Sigma_w := E[w_t w_t']$  is diagonal and full rank, that is  $\text{Rank}(\Sigma_w) = k$ . Eq.2.2.6 represents the structural form of the VAR that we want to estimate. As we have pointed out, the problem is that the structural model cannot be estimated directly, because the contemporaneous variables are endogenous and

their estimation would be subject to bias. What we can estimate, however, is the reduced form of the VAR,

$$\Phi(L) \begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \dots \\ y_{k,t} \end{bmatrix} = u_t, \quad (2.2.7)$$

which is the same as Eq.2.2.2, where  $\Phi(L) = I_k - \Phi_1 L - \Phi_2 L^2 - \dots - \Phi_p L^p$ , with  $p$  being the number of lags. The covariance matrix of the reduced form residuals  $\Sigma_u = E[u_t u_t']$  is in general non-diagonal. As pointed out above, this implies that we cannot study the effect of a shock on a variable without movements in the others, since the reduced-form residuals are mutually correlated. In other words, we cannot recover the structural shocks directly from the reduced form.

However, the reduced form can be interpreted as a linear transformation of the structural model. Such transformation is represented by the *un-mixing matrix*  $\Gamma_0$ , and in particular by its inverse  $\Gamma_0^{-1}$ . Indeed, if we pre-multiply both sides of Eq.2.2.6 by  $\Gamma_0^{-1}$ , we can rewrite  $\Phi(L)$  as

$$\Phi(L) = \Gamma_0^{-1} \Gamma(L) = \underbrace{\Gamma_0^{-1} \Gamma_0}_{I_k} - \underbrace{\Gamma_0^{-1} \Gamma_1}_{\Phi_1} L - \dots - \underbrace{\Gamma_0^{-1} \Gamma_p}_{\Phi_p} L^p, \quad (2.2.8)$$

and the reduced-form residuals as a linear combination of the mutually uncorrelated structural shocks:

$$u_t = \Gamma_0^{-1} w_t. \quad (2.2.9)$$

Therefore, if  $\Gamma_0$  is known, we can recover the structural model by simply pre-multiplying both sides of Eq.2.2.7 by  $\Gamma_0$ , which describes the simultaneous relations between the variables of the model. In this way, the response of each variable to the

shocks, i.e. the impulse responses, are obtained as

$$\Psi_i = \frac{\partial(y_{t+i})'}{\partial w'_t}, \quad i = 0, 1, 2, \dots, H, \quad (2.2.10)$$

which is a  $k \times k$  matrix of impulse response functions, with  $H$  being the maximum temporal horizon of the shock propagation and  $\Psi_0$  representing the contemporaneous responses to the shocks. Following Kilian and Lütkepohl (2017), we consider the companion form

$$\mathbf{Y}_t = \Phi \mathbf{Y}_{t-1} + U_t, \quad (2.2.11)$$

where

$$\mathbf{Y}_t := \begin{bmatrix} (y_t)' \\ \vdots \\ (y_{t-p+1})' \end{bmatrix}, \quad \Phi := \begin{bmatrix} \Phi_1 & \Phi_2 & \dots & \Phi_{p-1} & \Phi_p \\ I_k & 0 & \dots & 0 & 0 \\ 0 & I_k & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I_k & 0 \end{bmatrix}, \quad U_t := \begin{bmatrix} u_t \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

and  $k$  is the total number of variables included in the model. The IRFs are obtained as

$$\Psi_i = J \Phi^i J' \Gamma_0^{-1}, \quad (2.2.12)$$

where  $J := [I_k, 0_{k \times k(p-1)}]$ . Therefore, provided that  $\Gamma_0$  is known, we can use a consistent estimate of the reduced-form coefficients  $\Phi$  (obtained for instance via least-squares estimation) to retrieve the IRFs.

Once the structural coefficients and the impulse responses have been retrieved for both the real-world and the agent-based data, it is possible to compare them, as explained in section 2.5, to gauge to what extent the agent-based model is validated.

### 2.3 The Factor Augmented VAR model

A well-known problem in the traditional specifications of VAR models is that only a small amount of variables can be directly included in the model, as the number of parameters to estimate rapidly increases with the number of included variables. As a consequence, the choice of the variables to include is somewhat subjective and exploits a thinner informative set than that of, for example, Central Banks and policy-makers.

In their seminal work, Bernanke et al. (2005) proposed a Factor-Augmented Vector Autoregression (FAVAR) approach to overcome the problem of including a large amount of information in a VAR model. This method allows to account for a large part of the information contained in the data in a parsimonious way and it facilitates the problem of choosing which variables to include in the model, since a big part of the information is summarized by the factors. Moreover, the factor approach sometimes allows to get a more precise measure of given quantities which have a clear theoretical definition but cannot be distinctly observed in reality. For instance, Boivin and Giannoni (2006) employ factor models to directly deal with measurement errors while estimating DSGE models for which a one-to-one correspondence between the theoretical concepts and the actual observed variables does not exist.

Since a critical aspect of the methodology regards the choice of the variables to include in the VAR, a factor-based approach is ideal for our validation procedure. Indeed, the problem of choosing which variables to consider in the model is overshadowed since a big part of the information is included via the factors. It can therefore be considered a more “global” validation approach, which takes into account all the time series produced by the agent-based model, instead of being uniquely focused on a given subset of variables.

We consider a FAVAR model where  $v$  variables are directly included as “observed”

factors  $Y_{1,t}, \dots, Y_{v,t}$ . By adding the first, say  $z$  (in most applications, as in Bernanke et al. 2005, simply one or two) “unobserved” or latent static factors  $F_{1,t}, \dots, F_{z,t}$ , we get the structural FAVAR model

$$\Gamma(L) \begin{bmatrix} F_{1,t} \\ \dots \\ F_{z,t} \\ Y_{1,t} \\ \dots \\ Y_{v,t} \end{bmatrix} = w_t, \quad (2.3.1)$$

where  $\Gamma(L) = \Gamma_0 - \Gamma_1 L - \dots - \Gamma_p L^p$  is a  $(z+v) \times (z+v)$  matrix operator and  $w_t$  is a  $(z+v)$ -dimensional vector of structural shocks, as in Eq.2.2.6. From a computational point of view, the reduced-form estimation and the structural analysis can then be performed in the same way as for the VAR model, with the main difference being the interpretation of the results. That is, equations 2.2.2 to 2.2.12 are still valid and it suffices to substitute  $y_{1,t}, \dots, y_{k,t}$  with  $F_{1,t}, \dots, F_{z,t}, Y_{1,t}, \dots, Y_{v,t}$ .

### 2.3.1 Estimation of the static factors

Naturally, a necessary step for the implementation of a FAVAR model is the estimation of the static factors. In what follows, we will denote by  $Y_t$  the variables of interest that we directly include in the model, by  $X_t$  all the  $N$  variables in the dataset and by  $F_t$  the static factors. We consider two different procedures for the estimation of the static factors. To begin with, both require that the time series are made stationary, by applying suitable transformations, and re-scaled before estimating the factors by principal component analysis. When the aim is studying the properties of a dataset by analyzing the factors that can be extracted from it, as we

will do in section 4.2.1, we apply PCA directly to  $X_t$ :

$$X_t = \Lambda F_t + e_t, \quad (2.3.2)$$

where  $X_t$  is  $N \times 1$ ,  $F_t$  is the  $r \times 1$  vector of relevant factors, where  $r$  is the optimal number of static factors, and  $\Lambda$  is the  $N \times r$  loading matrix.  $\hat{\Lambda}$  and  $\hat{F}_t$  are the PC estimators of  $\Lambda$  and  $F_t$ . Several criteria have been proposed to determine the optimal number of static factors  $r$  (see e.g. Bai and Ng 2002, Onatski 2010, Kapetanios 2010, Ahn and Horenstein 2013 and Reijer et al. 2020).

Conversely, when we want to estimate the factors to include in the FAVAR, as in section 4.5, we employ the two-step procedure proposed by Hae Hwang (2009) as an alternative to the iterative procedure by Stock and Watson (2005). This takes into account the information given by the “observable” factors  $Y_t$  and ensures orthogonality of  $Y_t$  and  $F_t$ . In this case, we use the notation  $\tilde{X}_t$  instead of  $X_t$  to stress that the observable factors are treated separately from the rest of the variables. If  $N$  is the total number of variables in the full dataset and  $v$  is the number of observable factors, we will have that  $\tilde{X}_t$  is  $(N - v) \times 1$ . Furthermore, we have that  $F_t$  will be  $z \times 1$ , since, in general, a different number of factors than  $r$  can be included in the FAVAR.

Thus, as in Bernanke et al. (2005), we can write

$$\tilde{X}_t = \Lambda_f F_t + \Lambda_y Y_t + e_t. \quad (2.3.3)$$

To estimate  $\Lambda_f, F_t, \Lambda_y$  we proceed as follows:

Step 1: Estimate  $\Lambda_y$  via least-squares as the coefficients of a linear regression of  $\tilde{X}_t$  on  $Y_t$

$$\hat{\Lambda}_y = (Y'Y)^{-1}Y'\tilde{X}; \quad (2.3.4)$$

Step 2: Estimate  $\Lambda_f$  and  $F_t$  via PCA on the residuals of the previous regression,

$$\tilde{X}_t - \hat{\Lambda}_y Y_t = \Lambda_f F_t + e_t, \quad (2.3.5)$$

to obtain  $\hat{\Lambda}_f$  and  $\hat{F}_t$ .

## 2.4 The Dynamic Factor Model

A more sophisticated model based on macroeconomic factors is the dynamic factor model (DFM), in which it is modeled that the factors have effect on  $X_t$  through their lags too. The dynamic model is more realistic but harder to estimate than the static model (Barigozzi and Luciani 2019). However, its estimation can be accomplished by going through an equivalent static model, as illustrated below. The DFM is often deemed better than the FAVAR for structural analysis and it has been argued that it should be considered the natural tool for the validation of macroeconomic models (Lippi 2019). Furthermore, it also represents an advantage over structural VAR or FAVAR models where the researcher has to take a stance on the variables to include which, in turn, determine the number of shocks (Breitung and Eickmeier 2006). The comparison of the causal relations estimated by means of a DFM constitutes the third and final approach of our validation procedure.

As in the previous section, we denote the static factors  $F_t$  while we now indicate the dynamic factors with  $f_t$ . Following Doz and Fuleky (2020), we consider the state-space representation of the model:

$$X_t = \Lambda_0 f_t + \Lambda_1 f_{t-1} + \cdots + \Lambda_s f_{t-s} + e_t, \quad (2.4.1)$$

$$f_t = A_1 f_{t-1} + A_2 f_{t-2} + \cdots + A_p f_{t-p} + u_t, \quad (2.4.2)$$

where  $A_1, \dots, A_p$  are  $N \times q$  matrices and  $f_t$  is a vector of  $q$  stationary factors. Eq.2.4.1 and Eq.2.4.2 are called *measurement* and *state* equations respectively.  $q \leq r$  is the optimal number of dynamic factors and, as for the number of static factors, there exist many criteria in the literature to estimate it (see e.g. Bai and Ng 2007, Amengual and Watson 2007 and Onatski 2009).

Note that from the dynamic Eq.2.4.1, if we let  $F_t = [f_t', f_{t-1}', \dots, f_{t-s}']'$  and  $\Lambda = [\Lambda_0, \Lambda_1, \dots, \Lambda_s]$ , we can recover its static equivalent given by Eq.2.3.2. Furthermore, as we did for the standard VAR, we can cast the state equation in companion form in the following way:

$$\mathbf{F}_t = \mathbf{A}\mathbf{F}_{t-1} + U_t, \quad (2.4.3)$$

where

$$\mathbf{F}_t := \begin{bmatrix} (f_t)' \\ \vdots \\ (f_{t-p+1})' \end{bmatrix}, \quad \mathbf{A} := \begin{bmatrix} A_1 & A_2 & \dots & A_{p-1} & A_p \\ I_q & 0 & \dots & 0 & 0 \\ 0 & I_q & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I_q & 0 \end{bmatrix}, \quad U_t := \begin{bmatrix} u_t \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

and (as we do for the rest of this section) we have assumed for semplicity that  $p = s$ .

The simplest way to estimate the model is the two-step procedure proposed by Reichlin et al. (2005) and Forni et al. (2009):<sup>1</sup>

Step 1: The static factors and the loadings are estimated by PC, obtaining  $\hat{F}_t$  and  $\hat{\Lambda}_t$ ;

Step 2: An estimate of the  $A_1, \dots, A_p$  matrices is obtained by estimating a VAR on the estimated factors

---

<sup>1</sup>Other, more sophisticated, ways to estimate a DFM include Doz et al. (2011), Doz et al. (2012) and Barigozzi and Luciani (2019).

$$\hat{F}_t = A_1 \hat{F}_{t-1} + \dots + A_p \hat{F}_{t-p} + u_t. \quad (2.4.4)$$

However, in this way, we only obtain reduced-form coefficients. When performing structural analysis, we are interested in recovering the impact of  $q$  structural shocks, which we denote  $w_t$ , that drive the common factors and impact the variables in  $X_t$ . That is, we look for the impulse response functions

$$\Psi_i = \frac{\partial(X_{t+i})'}{\partial w_t'}, \quad i = 0, 1, 2, \dots, H, \quad (2.4.5)$$

which are  $N \times q$  matrices, with  $H$  being the maximum temporal horizon of the shock propagation and  $\Psi_0$  representing the contemporaneous responses to the shocks.

It is clear that with the above procedure we obtain estimates of  $F_t$  that are  $r \times 1$ . However, since  $\hat{F}_t$  includes estimates of the lagged factors, some of the VAR equations are identities (at least asymptotically) and, therefore, the rank of the covariance matrix of the residuals  $\hat{\Sigma}_u$  is  $q \leq r$ , as  $N \rightarrow \infty$  (Breitung and Eickmeier 2006). Nonetheless, we can obtain an estimate of the actual  $q \times 1$  innovations of the dynamic factors as  $\hat{u}_t = \hat{G}\hat{\eta}_t$ , where the  $r \times q$  matrix  $\hat{G}$  is the matrix of the  $q$  eigenvectors associated to the  $q$  largest eigenvalues of  $\hat{\Sigma}_u$ . These estimates can then be used to identify the structural shocks  $w_t$  that drive the common factors, by means of suitable identification strategies:

$$\eta_t = \Gamma_0^{-1} w_t, \quad (2.4.6)$$

which is the equivalent of Eq.2.2.9.

Hence, the IRFs are

$$\Psi_i = \Lambda J \mathbf{A}^i J' G \Gamma_0^{-1}, \quad (2.4.7)$$

where  $J := [I_r, 0_{r \times r(p-1)}]$ . Therefore, provided that  $\Gamma_0^{-1}$  is known, we can use a consistent estimate of the reduced-form coefficients  $\mathbf{A}$  to retrieve the IRFs. Note that, in this model, similar objects to the structural coefficients of VAR and FAVAR models can be computed as  $\Lambda A_1 G \Gamma_0^{-1}, \dots, \Lambda A_p G \Gamma_0^{-1}$ .

Note that in the DFM, the sign of the IRFs is undetermined as the  $\Lambda$  and  $G$  matrices are estimated up to their sign, an issue which also reflects on the confidence bands. To compute the confidence intervals for the DFM IRFs, we perform a wild bootstrap on the VAR of the factors (see Eq.2.4.2) and then proceed to the computation of the IRFs at every new iteration according to Eq.2.4.7. However, as matrix  $G$ , the matrix of  $q$  eigenvectors, has to be recomputed at every new bootstrap iteration, it is possible for the sign of its columns to flip for some of the iterations, since eigenvectors are estimated with PCA up to their sign. This results in “wide” confidence bands that are likely to show symmetry about zero.

Since both the IRF and the structural coefficients matrices have dimensions  $N \times q$  and  $N$  is large, it is customary to select only those entries corresponding to the impact of the first dynamic innovations on some particular variables of interest in  $X_t$ .

## 2.5 Validation assessment

After the causal structures embedded in the real-world and in the agent-based data have been recovered, either with a VAR, FAVAR or DFM model, the validity of the model under validation is assessed by comparing these causal structures.

After the identification procedure, we have estimated the structural matrices  $\Gamma_i^{RW}$  for  $i = 0, \dots, p_{RW}$  and  $\Gamma_i^{AB}$  for  $i = 0, \dots, p_{AB}$  and the structural impulse responses  $\Psi_i^{RW}$  for  $i = 0, \dots, H$  and  $\Psi_i^{AB}$  for  $i = 0, \dots, H$ .<sup>2</sup> As a first validation assessment, we employ the three similarity measures proposed by Guerini and Moneta (2017):

---

<sup>2</sup>Note that we are now using the notation of section 2.2, thereby indicating with  $\Gamma_1, \dots, \Gamma_p$  all the structural coefficients.

a *sign-based* similarity measure,  $\Omega^{sign}$ , which compares the signs of the estimated structural coefficients; a *size-based* measure,  $\Omega^{size}$  which compares the size of the causal effects; and a *conjunction* measure,  $\Omega^{conj}$ , which jointly compares the sign and the size of the causal effects. For completeness, we will compute these measures on the reduced-form coefficients also.

As a preliminary step we define  $p_{min} = \min\{p_{RW}, p_{AB}\}$  and select only the structural residuals for  $i = 0, \dots, p_{min}$ . The sign-based similarity measure is built by means of the indicator function

$$\omega_{i,jk}^{sign} = \begin{cases} 1 & \text{if } \text{sign}(\gamma_{i,jk}^{RW}) = \text{sign}(\gamma_{i,jk}^{AB}) \\ 0 & \text{if } \text{sign}(\gamma_{i,jk}^{RW}) \neq \text{sign}(\gamma_{i,jk}^{AB}) \end{cases}$$

where  $j$  and  $k$  are the row and column indexes of the structural coefficients. In practice, this assigns 1 if the agent-based model captures the same direction of causality of the real-world estimates and 0 otherwise. The sign-based similarity measure is therefore defined as

$$\Omega^{sign} = \frac{(\sum_{i=1}^{p_{min}} \sum_{j=1}^k \sum_{k=1}^k \omega_{i,jk}^{sign})}{k^2 p_{min}}. \quad (2.5.1)$$

The second similarity measure tries to capture the extent to which the size of the estimates in the simulated data are similar to the size of the estimates in the real-world data. The indicator function in this case is

$$\omega_{i,jk}^{size} = \begin{cases} 1 & \text{if } \gamma_{i,jk}^{AB} \in [\gamma_{i,jk}^{RW} - 2\sigma(\gamma_{i,jk}^{RW}), \gamma_{i,jk}^{RW} + 2\sigma(\gamma_{i,jk}^{RW})] \\ 0 & \text{if } \gamma_{i,jk}^{AB} \notin [\gamma_{i,jk}^{RW} - 2\sigma(\gamma_{i,jk}^{RW}), \gamma_{i,jk}^{RW} + 2\sigma(\gamma_{i,jk}^{RW})] \end{cases}$$

and the size-based similarity measure is computed as

$$\Omega^{size} = \frac{(\sum_{i=1}^{p_{min}} \sum_{j=1}^k \sum_{k=1}^k \omega_{i,jk}^{size})}{k^2 p_{min}}. \quad (2.5.2)$$

Finally, the conjunction measure evaluates the extent to which the model represents both the direction and the magnitude of the causal relationships entailed in the real-world data. The indicator function is

$$\omega_{i,jk}^{conj} = \begin{cases} 1 & \text{if } \text{sign}(\gamma_{i,jk}^{RW}) = \text{sign}(\gamma_{i,jk}^{AB}) \wedge \gamma_{i,jk}^{AB} \in [\gamma_{i,jk}^{RW} - 2\sigma(\gamma_{i,jk}^{RW}), \gamma_{i,jk}^{RW} + 2\sigma(\gamma_{i,jk}^{RW})] \\ 0 & \text{if else} \end{cases}$$

and the conjunction similarity measure is computed as

$$\Omega^{conj} = \frac{(\sum_{i=1}^{p_{min}} \sum_{j=1}^k \sum_{k=1}^k \omega_{i,jk}^{conj})}{k^2 p_{min}}. \quad (2.5.3)$$

In addition to the similarity measures proposed by Guerini and Moneta (2017), we introduce a new measure which looks at when the agent-based IRFs lie inside the confidence bands computed for the real-world IRFs. Since most agent-based models are built to perform policy exercises, we believe this to be the most appropriate similarity measure to be used as a validation tool. Indeed, impulse response functions allow to look at the effects of a shock in a variable on other variables, all else equal, from which it is possible to draw policy recommendations. We call this the *irf* similarity measure and it is based on the indicator function

$$\omega_{i,jk}^{irf} = \begin{cases} 1 & \text{if } \gamma_{i,jk}^{AB} \in [{}^{5\%}\gamma_{i,jk}^{RW}, {}^{95\%}\gamma_{i,jk}^{RW}] \\ 0 & \text{if } \gamma_{i,jk}^{AB} \notin [{}^{5\%}\gamma_{i,jk}^{RW}, {}^{95\%}\gamma_{i,jk}^{RW}] \end{cases}$$

where the 5% and 95% superscripts denote the confidence bands of the impulse responses, and it results in the similarity measure computed as

$$\Omega^{irf} = \frac{(\sum_{i=1}^H \sum_{j=1}^k \sum_{k=1}^k \omega_{i,jk}^{irf})}{k^2 H}, \quad (2.5.4)$$

where  $H$  denotes the chosen horizon for the impulse response functions.

All four similarity measures are bounded between  $[0, 1]$ , which allows for an easy interpretation of the results.

### 3 Identification strategy

The R code relative to this section, as well as additional details and comments, are available in the form of Rmarkdowns knitted to html at the following link:

<https://drive.google.com/drive/folders/18OqK69Prpv3wvRDyREyrR1-BudPrOXQB?usp=sharing>

The problematic nature of shock identification in Vector Autoregressive models seems to have afflicted each and every implementation of this econometric technique since its introduction in the macroeconomic literature (Sims 1980). For the purpose of obtaining a structural representation of the economic mechanisms that they try to capture with structural VARs, researchers have often found themselves in a position in which they have to rely on a number of assumptions, mostly derived from economic theory or based on a description of the economic system which is overly simplified. However, most of these assumptions cannot be tested, leaving broad discretion in the choice of the shock identification strategy. Recent developments in the VAR literature, which have placed emphasis on data-driven algorithms, have demonstrated that it is possible to identify structural shocks by using only the distribution of reduced-form shocks and taking advantage of the information provided by its higher-order moments, making shock identification possible without the kinds of assumptions traditionally used in the literature (see e.g. Hyvärinen et al. 2010, Moneta et al. 2013 and Lanne et al. 2017). Yet, these approaches, rooted in independent component analysis, do rely on a set of less theory-driven but nonetheless heavy assumptions, such as independence and non-Gaussianity of the structural shocks (the latter of which conveniently being testable). Furthermore, a crucial issue when recurring to independent component analysis is given by the indeterminacy of the permutation and scaling of the columns of the mixing matrix and the structural shocks, which is typically solved by introducing additional assumptions, such as the acyclic structure of the causal relations.

In this section, we introduce the NGSi algorithm (short for *non-Gaussian shock identification*). This data-driven algorithm is able to perform VAR identification relying solely on the assumptions of independence and non-Gaussianity of the structural shocks. Indeed, this ICA-based algorithm does not require additional assumptions on the structure of the matrix of contemporaneous causal relations as it is able to infer it from the data and accordingly apply the most appropriate identification scheme. With an approach more typical of Data Science rather than Economics, it combines independent component analysis, some of the intuitions behind VAR-LiNGAM (Shimizu et al. 2006, Moneta et al. 2013), and the traditional recursive identification scheme, to achieve shock identification in a general setting. Following this approach, we perform an extensive simulation study to test the performance of the algorithm under different specifications of the mixing matrix. We will show that the NGSi algorithm performs well (“correctness” ranging from 55 to 100%) across a wide variety of data generating processes, where we vary the number of variables, the number of lags and the structure of the mixing matrix.

We first review the standard recursive identification scheme via Cholesky decomposition and how VAR identification can be achieved via independent component analysis. We introduce MaxDiag, a simple identification scheme based on ICA, and we review VAR-LiNGAM. By combining these techniques, we show that it is possible to obtain five different identification algorithms. In an extensive simulation study (which serves as a basis for the development of the NGSi algorithm), we test these algorithms against different structures of the mixing matrix, which in turn imply different assumptions on the contemporaneous relations among the variables. We do this by artificially generating data according to a S-VAR process where we vary the structure of the mixing matrix. We then estimate the VAR by OLS and use the reduced-form residuals to obtain estimates of the mixing matrix via the consid-

ered identification algorithms, and, after having defined some convenient metrics, we study under which conditions each identification scheme is able to retrieve such matrix correctly (and how precisely). To the best of our knowledge, this is the first study of this kind in the VAR literature.

### 3.1 The recursive identification scheme

A common approach to solve the identification problem (see section 2.2) is to impose a sufficient number of restrictions to the entries of  $\Gamma_0$  in order to recover the unconstrained ones from the estimates  $\hat{\Phi}_1, \dots, \hat{\Phi}_p, \hat{\Sigma}_u$ . In particular, it is customary to assume that the simultaneous relationships between the variables are *acyclic*, meaning that if we have three variables  $A, B$  and  $C$ , if  $A$  influences  $B$  (i.e. if variable  $B$  responds to a shock in  $A$ ), then variable  $B$  can only influence  $C$  and  $C$  has no contemporaneous influence on the previous variables, even if it reacts to shocks in  $A$  and  $B$ . This assumption imposes that there are no contemporary feedbacks in the system and that there exists a precise causal ordering of the variables. In practice, this is equivalent to imposing that  $\Gamma_0$  is lower triangular, given a particular ordering of the variables. By doing so,  $\Gamma_0^{-1}$  can be unambiguously identified through the Cholesky factorization of  $\hat{\Sigma}_u$  and the particular contemporaneous ordering is usually chosen by relying on prior economic knowledge. This technique has perhaps been the most popular way to identify a structural VAR models, as the Cholesky factorization of the variance-covariance matrix of reduced-form residuals is an efficient and straightforwardly implementable way to “orthogonalize” the reduced-form errors, that is, to disentangle  $w_t$  from the reduced-form innovations  $u_t$ . However, it must be stressed that this identification scheme is built upon the a priori imposition of a whole causal chain with a rigid, recursive causation order, deriving from the computational restriction imposed by the Cholesky factorization.

Recall (see Eq.2.2.4 and successive) that we want to recover  $w_t$  such that  $E[w_t w_t'] = I_k$ . If we substitute  $w_t = \Gamma_0 u_t$  to get  $E[w_t w_t'] = E[\Gamma_0 u_t u_t' \Gamma_0'] = \Gamma_0 E[u_t u_t'] \Gamma_0'$ , we have that the middle element of this expression,  $E[u_t u_t']$ , can be estimated by the sample variance of the reduced-form residuals,  $\hat{\Sigma}_u$ . The Cholesky factorization is a decomposition of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its upper triangular transpose. Since  $\Sigma_u$  is a variance-covariance matrix, it is indeed Hermitian and positive-definite and it can therefore be decomposed as  $P_k P_k'$  with  $P_k$  being lower triangular. Therefore, if we take  $P_k$  as our choice for  $\Gamma_0^{-1}$ , that is we have  $\Gamma_0 = P_k^{-1}$ , we get  $E[w_t w_t'] = \Gamma_0 E[u_t u_t'] \Gamma_0' = \Gamma_0 \Gamma_0^{-1} \Gamma_0'^{-1} \Gamma_0' = I_k$ , which is exactly what we are looking for. Therefore, the matrix  $\Gamma_0$  can be estimated as the inverse of the lower triangular factor (hence also lower triangular) of the Cholesky decomposition of the variance-covariance matrix of reduced-form residuals.

However, as remarked above, the choice of a lower triangular  $\Gamma_0$  introduces a recursive structure among the variables. Consider for example a VAR with three variables where the variables are ordered simply as  $[y_1 \rightarrow y_2 \rightarrow y_3]$ , we would have

$$\Gamma_0 y_t = \Gamma_1 y_{t-1} + \dots + \Gamma_p y_{t-p} + \Gamma_0 u_t, \quad (3.1.1)$$

and in matrix form

$$\begin{aligned} \begin{bmatrix} a & 0 & 0 \\ b & c & 0 \\ d & e & f \end{bmatrix} \begin{bmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \end{bmatrix} &= \Gamma_1 \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ y_{3,t-1} \end{bmatrix} + \dots + \Gamma_p \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ y_{3,t-p} \end{bmatrix} + \begin{bmatrix} a & 0 & 0 \\ b & c & 0 \\ d & e & f \end{bmatrix} \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ u_{3,t} \end{bmatrix} \\ &= \Gamma_1 \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ y_{3,t-1} \end{bmatrix} + \dots + \Gamma_p \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ y_{3,t-p} \end{bmatrix} + \begin{bmatrix} w_{1,t} \\ w_{2,t} \\ w_{3,t} \end{bmatrix}, \end{aligned} \quad (3.1.2)$$

which, multiplying both sides by  $\Gamma_0^{-1}$ , becomes

$$\begin{aligned}
\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \end{bmatrix} &= \Phi_1 \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ y_{3,t-1} \end{bmatrix} + \cdots + \Phi_p \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ y_{3,t-p} \end{bmatrix} + \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ u_{3,t} \end{bmatrix} \\
&= \Phi_1 \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ y_{3,t-1} \end{bmatrix} + \cdots + \Phi_p \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ y_{3,t-p} \end{bmatrix} + \begin{bmatrix} a & 0 & 0 \\ b & c & 0 \\ d & e & f \end{bmatrix}^{-1} \begin{bmatrix} w_{1,t} \\ w_{2,t} \\ w_{3,t} \end{bmatrix}.
\end{aligned} \tag{3.1.3}$$

This means that a structural shock in  $y_{1,t}$  (that is, a change in  $w_{1,t}$ ) entails a contemporaneous effect on  $y_{1,t}$ ,  $y_{2,t}$  and  $y_{3,t}$ ; a shock in  $y_{2,t}$  entails a contemporaneous effect on  $y_{2,t}$  and  $y_{3,t}$  but not on  $y_{1,t}$ , which is affected only after one lag (through  $\Gamma_1 y_{2,t-1}$ ); a shock in  $y_{3,t}$  entails a contemporaneous effect only on  $y_{3,t}$  itself, as  $y_{2,t}$  and  $y_{3,t}$  are affected only after the first lag. Hence  $y_{1,t}$  can be referred to as a “slow moving variable” (since it takes time to react to the shocks in the other variables), while  $y_{3,t}$  can be referred to as a “fast moving variable.” Since under this scheme the diagonal elements of  $P_k$  correspond to the square roots of the diagonal elements of the variance-covariance matrix  $\Sigma_u$ , we have that  $\Gamma_0$  is chosen so that the structural shocks represent one standard deviation of the time series of reduced-form shocks.

Since the Cholesky identification scheme can correctly retrieve the matrix  $\Gamma_0$  only if the true structure is indeed recursive and the ordering of the variables is specified correctly, this approach is problematic for a number of reasons. As Kilian and Lütkepohl (2017) put it, the credibility of an approach that imposes a recursive causal architecture without any clear order of the variables in mind is undermined in the first place. Furthermore, this is aggravated by the fact that the number of possible orderings grows with the factorial of the number of variables, and, finally, even if all

the permutations lead to the same impulse responses, this does not prove that every identification strategy is bound to lead to the same results. It simply shows that all recursive identifications provide the same results, but it gives no evidence that the model should be recursive in the first place. This is why this approach has inspired a series of critic contributions to the literature that take explicit aim at the fact that it seems to be built on the (often quite misled) confidence in the data's ability to speak for themselves but which in practice relies on a set of assumptions that are extremely difficult to justify within real-world applications (Cooley and LeRoy 1985).

### **3.2 Identification via independent component analysis**

In recent years, we have observed the flourishing of a new strand of literature that bases the identification of structural VARs on the assumption of non-Gaussianity of the shocks (Gourieroux, Monfort, et al. 2014; Moneta et al. 2013). Indeed, if the data are non-Gaussian, which is not uncommon in many econometric studies (see for example Lanne and Lütkepohl 2010 and Lanne and Saikkonen 2013), it is possible to exploit higher-order statistics of the reduced-form residuals to identify the VAR. Under the additional assumption of independence of the shocks (instead of simple orthogonality) the identification can be obtained by applying independent component analysis.

Independent component analysis (ICA) is a popular computational method for separating a multivariate signal into additive subcomponents. Given a data matrix  $X$ , the aim is to factor it as  $X = SA$ , where  $A$  is a matrix of coefficients and  $S$  is a matrix whose columns are independent. The interpretation that is often given is that the columns of  $S$  are unobserved independent source signals that are linearly combined together into the observed signals  $X$ . Under this generative model, the observed signals in  $X$  will tend to be more Gaussian than the source components (in

$S$ ) due to the Central Limit Theorem. Indeed, when applying ICA, a key assumption is that the source signals are in fact non-Gaussian (at most one column of  $S$  is allowed to be Gaussian), since otherwise it would not be possible to decompose  $X$  correctly and ICA would not retrieve the original, correct,  $S$ . A computationally efficient algorithm that performs independent component analysis is “fastICA”, which proceeds as follows:

1. The data are centered by subtracting the mean of each column of the data matrix  $X$ ;
2. The data matrix is then “whitened” (that is, transformed into a new set of variables which are uncorrelated and which have unitary variance) by projecting the data onto its principal component directions. This gives  $XK$ , where  $K$  is a matrix of eigenvectors of  $X$ ;
3. Finally, the algorithm estimates a matrix  $Q$  such that  $XKQ = S$ , under the constraint that  $Q$  is orthonormal. Among the infinite possible such matrixes,  $Q$  is chosen in order to maximize the non-Gaussianity of  $S$ , as measured by its Negentropy. This iterative step is done by initially setting a random  $Q$  and adjusting it to maximize Negentropy at each iteration until convergence is reached. We have that our desired un-mixing matrix is given by  $A^{-1} = KQ$ .<sup>3</sup>

If we take  $X = U$ ,  $S = W$  and  $A = (KQ)^{-1} = \Gamma_0'^{-1}$  (from which  $\Gamma_0^{-1} = A'$ ), we can see that our identification problem fits perfectly in this framework: we have  $X = SA = S(KQ)^{-1}$  which is equivalent to  $U = W\Gamma_0'^{-1}$ .

As stated above, we have that the two pivotal assumptions of ICA regarding the source signals are their independence (which fits well with the SVAR framework, where the structural shocks are required to be orthogonal) and their non-Gaussianity.

---

<sup>3</sup>For further details see Hyvärinen (1999) and Hyvärinen and Oja (2000).

This second assumption can be tested prior to performing the decomposition by testing Gaussianity (for example with the Jarque-Bera, Shapiro-Wilk or Shapiro-Francia tests) on the reduced form residuals  $u_t$ , thank to the Lévy-Cramér theorem which states that, given some independent random variables, their sum is normally distributed if and only if each of them is individually normally distributed.

Finally, it must be pointed out that, since  $S$  is unobserved, ICA identifies the matrix  $A$  up to the scale, sign and permutation of its rows. Hence,  $\Gamma_0^{-1}$  is identified up to the scale, sign and permutation of its columns. This issue, while often disregarded in signal processing, is very relevant to the solution of the identification problem in SVARs. A rescaling of the columns of  $\Gamma_0^{-1}$  can be interpreted simply as a rescaling of the structural shocks and it is not of great concern. Contrarily, a wrong sign and permutation of the columns of  $\Gamma_0^{-1}$  implies a wrong matching of the variables to their contemporaneous relations (and hence a wrong matching to their structural lagged relations), therefore invalidating the whole identification. To solve the permutation and sign issues some refinements of ICA specific for VAR applications have been proposed.

### 3.2.1 A simple ICA identification algorithm

We now propose a first simple criterion to solve the indeterminacies of the ICA estimation.

Since in the SVAR setting the  $\Gamma_0$  matrix represents the contemporaneous relations among variables, it could seem reasonable to expect that the biggest element in each row should be the diagonal element. In other words, we could expect that each variable reacts more to its own shock rather than to the shocks of the other variables. If the biggest element in each row of  $\Gamma_0$  is the diagonal one, the same holds for  $\Gamma_0^{-1}$ . We can therefore solve the permutation issue by setting the columns of  $\Gamma_0^{-1}$  according to this criterion. As for the sign issue, for better interpretability, we require that

the diagonal element of each column is positive. We can do this simply by switching the sign of each column for which the diagonal element is negative. Furthermore, we normalize the estimated matrix so that the variance-covariance matrix of the estimated structural shocks is the identity matrix. In the rest of this section we will refer to this identification algorithm as “MaxDiag”.

We are well aware that this simple algorithm alone cannot constitute a solution to the identification problem, as in many settings the assumption of “big diagonal” (that is, that each variable reacts more to its own shock rather than to the shocks of the other variables) is implausible. Nonetheless, it will prove useful in building the NGSF algorithm.

### 3.2.2 VAR-LiNGAM

An identification algorithm that relies on ICA that has enjoyed success is the VAR-LiNGAM algorithm, first introduced by Shimizu et al. (2006) and developed by Moneta et al. (2013). It relies on the assumption of a recursive structure and it is able to infer the ordering of the variables in a data-driven fashion.

Unlike the previous algorithm, VAR-LiNGAM works with the matrix  $\Gamma_0$  and not  $\Gamma_0^{-1}$ , however this is of course not a substantive difference, as it only entails that the indeterminacy is that of the permutation and sign of the rows instead of that of the columns. In addition to the previous algorithm, it also standardizes the rows of the  $\Gamma_0$  matrix so as to have all ones on the diagonal. Then, it looks for the row permutation that gives the matrix which is closest as possible to lower triangular. The row ordering that gives such matrix is taken as the order of the recursive structure and the upper part of the matrix is pruned (coefficients set to zero) so that the resulting matrix is actually lower triangular. Finally, the rows are set back to their original order. In detail, the steps of the algorithm are the following:

1. Perform fastICA (or another suitable ICA algorithm) on the matrix of residuals  $U$  and get an initial estimate of  $\Gamma_0$ ;
2. Among all the possible row permutations of  $\Gamma_0$ , choose the one that maximizes the elements on the diagonal. This is done in practice by calculating the sum of the absolute value of the reciprocal of the elements of the diagonal;
3. Divide all rows by the corresponding diagonal element to obtain ones on the diagonal;
4. Take  $G_0 = I_k - \Gamma_0$ , which is required to be strictly lower triangular and, in general, with  $k^2 - k$  non-zero entries. The aim is to find the combination of  $\frac{k^2-k}{2}$  coefficients (to be put in the upper triangular part of the matrix) that when summed give the lowest possible quantity. To do this, try all the possible  $Z_k$  permutation matrices and compute the sum of the upper triangular part of  $Z_k G_0 Z_k'$ . Choose the  $Z_k$  matrix that gives the “smallest” upper triangular part;
5. Prune (set to zero) the upper triangular elements of  $Z_k G_0 Z_k'$ . The idea behind this step and the previous is that we want to set to zero the minimal amount of coefficients of  $\Gamma_0$  that induce a recursive structure. Furthermore, we prefer to prune coefficients that are already small to begin with;
6. Get back to  $G_0^* = Z_k' Z_k G_0^* Z_k' Z_k$ , where  $G_0^*$  is the pruned version of  $G_0$ ;
7. Get  $\Gamma_0^* = I_k + G_0^*$ ;
8. Reorder the  $\Gamma_0$  matrix according to the row permutations found in step 2 and invert the matrix to get  $\Gamma_0^{-1}$ .

It must be noted that, because of the pruning, the estimated  $\Gamma_0^{-1}$  matrix does not render the structural coefficients completely independent. This shortcoming is

more severe the less the “true”  $\Gamma_0^{-1}$  is lower triangular, which entails larger pruning. Conveniently, in applications it is straightforward to check the magnitude of the pruning and eventually insert a warning as additional output of the algorithm should the pruning be too big.<sup>4</sup>

### 3.3 Testing the identification algorithms against different mixing matrices

We have presented three schemes to estimate the  $\Gamma_0^{-1}$  matrix, which can simply be treated as algorithms: the Cholesky decomposition of the variance-covariance matrix of the reduced-form residuals, MaxDiag and VAR-LiNGAM. As we have seen, a crucial issue with the Cholesky scheme is given by the indeterminacy in the permutation of the columns. However, since the VAR-LiNGAM algorithm is able to retrieve the recursive structure of the mixing matrix (when it exists), we can use it to inform the other two algorithms. Indeed, as a byproduct, VAR-LiNGAM gives an estimate of the recursive ordering of the variables and it is straightforward to obtain a fourth algorithm by ordering the variables according to the order estimated by VAR-LiNGAM and then running a standard Cholesky decomposition. Since the same can be done with the MaxDiag algorithm by permuting the columns accordingly, we end up with five identification algorithms.

The problem in real-world VAR applications is that the  $\Gamma_0$  matrix is never observed and therefore the researcher has to rely on assumptions on its structure (that is, on the magnitude and sparseness of its coefficients) to justify the use of a given identification strategy. However, it is possible to assess the performance of each identification scheme by means of a simulation study. Indeed, we compare how well each

---

<sup>4</sup>In this particular application we follow Moneta et al. (2013) so as to get a warning if the sum of the squared pruned coefficients divided by the sum of all the squared coefficients of the un-pruned matrix is  $> 0.05$ .

of the five algorithms estimates the  $\Gamma_0^{-1}$  matrix under its different possible specifications, by artificially generating the mixing matrix and the corresponding VAR data several times and testing how well (on average) each identification scheme performs. We will investigate the performance of the five algorithms when the matrix is lower triangular, when it has a different recursive structure and when it does not have a recursive structure. Furthermore, we will distinguish the cases when the biggest element (in absolute value) in each column is the one relative to the contemporaneous relation of a variable with itself and when this is not the case for at least one variable. In practice, this is obtained by setting the biggest element in each column on the main diagonal before permuting the columns. For brevity, we refer to this latter classification as “big diagonal” versus “small diagonal” (even though it could arguably be referred to more clearly as big *contemporaneous relation* versus small *contemporaneous relation* of each variable with itself).<sup>5</sup>

We now discuss why these are the exhaustive dimensions that we need to explore in our simulation study by briefly going over the identification algorithms we want to test. A priori, we expect the following:

- **Cholesky.** Imposes a lower triangular structure and works well both when the diagonal elements of the mixing matrix are “big” or “small”. When the true structure is recursive but not lower triangular, to use this scheme it is necessary to specify the ordering of the variables, either based on economic theory or as estimated by the VAR-LiNGAM algorithm. It fails to identify the VAR when the structure of the mixing matrix is not recursive.
- **MaxDiag.** Does not impose a particular structure: it works well both with a recursive and a non-recursive structure of the mixing matrix. It is always able to estimate correctly the coefficients of the mixing matrix, however, to correctly

---

<sup>5</sup>Examples of big diagonal and small diagonal mixing matrices are given in Appendix C. (figures C.8, C.9, C.10 and C.11, C.12, C.13 respectively).

solve the permutation and sign indeterminacies it requires the diagonal of the true mixing matrix to be “big”.

- **VAR-LiNGAM.** Imposes a recursive structure and works well both when the diagonal elements of the mixing matrix are “big” or “small”. To retrieve the recursive ordering of the variables, it exploits the sparseness of the mixing matrix as estimated via ICA. Since it always imposes a recursive structure, it fails when the true structure of the mixing matrix is not recursive. Conveniently, when running, it also informs if the pruning which is being performed is large, which is an indication that the true structure might not be recursive.

These a priori expectations are summarized in Tab.1, with the checkmarks indicating that the algorithm performs the identification correctly under a given structure of the  $\Gamma_0^{-1}$  matrix.

$\Gamma_0^{-1}$		Cholesky	Cholesky + order	MaxDiag	MaxDiag + order	VAR_LiNGAM
Big diagonal	Lower triangular	✓	✓	✓	✓	✓
	Other recursive structure	✗	✓	✗	✓	✓
	Not recursive	✗	✗	✓	✗	✗
Small diagonal	Lower triangular	✓	✓	✗	✗	✓
	Other recursive structure	✗	✓	✗	✗	✓
	Not recursive	✗	✗	✗	✗	✗

Table 1: *A priori expectations on the performance of the five identification algorithms.*

At this point the crucial question is how to assess the identification schemes during the simulation study, that is, how to compare the true  $\Gamma_0^{-1}$  with the estimate  $\hat{\Gamma}_0^{-1}$ .<sup>6</sup> To this aim we define five distance measures:

- Total squared error (TSE): the sum of the squared difference of each entry of the true and the estimated matrices. We can interpret this measure as the overall precision in the estimate.

---

<sup>6</sup>We adopt a simple normalization to make the true and the estimated mixing matrices readily comparable. That is, we rescale the columns of the matrices (and possibly flip their sign) so that the entries corresponding to the contemporaneous relation of each variable with itself take on the value 1.

- Percentage of correct sign: how many entries of the estimated matrix have (in percentage terms) the same sign as the corresponding entries of the true matrix;
- Percentage of close-to-correct size: how many entries of the estimated matrix are (in percentage terms) less than two standard deviations apart from the corresponding entries of the true matrix. The standard deviations are calculated on all the entries of the true matrix. This can be seen as an additional measure of estimate precision: while the TSE looks at how precise the overall estimate of the matrix is, this measure looks at the percentage of entries of the matrix can be considered “precise”;
- Percentage of correct contemporaneous relations: how many contemporaneous relations are correctly identified (in percentage terms). In practice, given the normalization we adopted, this corresponds to checking where the unitary entries are located in the estimated matrix with respect to where they are located in the true matrix;
- Overall similarity. This final distance measure tries to capture what is generally considered important in the estimation of the  $\Gamma_0^{-1}$  matrix. In most cases, when implementing a SVAR model, we are not interested in retrieving the exact coefficients of the contemporaneous relations matrix, but rather on estimating a matrix that is “qualitatively” similar to the true one. By this, we mean that we want the sign of the entries of the matrix to be correct and that we want them to be “close enough” in magnitude to their real counterpart.

It can be considered a sort of “conjunction measure” of some of the other four measures and we will take it as an indicator of overall correct identification. It can take on values 0 and 1 and it is implemented in the following way:

1. Classify as “not correct” any entry of the estimated matrix that does not lie within two standard deviations of its true counterpart;
2. Then, among the remaining entries, actually classify as “correct” only those that have either the same sign or that are very close (in size) to their true counterpart and classify as “not correct” any other entry.<sup>7</sup> The idea here is that if the true and estimated coefficients are both very small, it does not matter if their sign differs.
3. Finally, we classify as “overall correct” an estimate of  $\Gamma_0^{-1}$  in which at most one entry has individually been classified as “not correct” in steps 1 and 2.

In addition to these five measures, we will pay special attention to the additional info produced by the VAR-LiNGAM algorithm. In this respect, we do two things: the first is counting how many times (on average) the ordering of the variables is estimated correctly; the second is looking at the warning with which it signals whether the pruning being made is judged to be “too big” and therefore that the true mixing matrix might not be recursive.

To test the performance of the five algorithms, we artificially generate data from a SVAR process of which we store the true mixing matrix (labeled according to its structure) used for each new data generation and compare it to the estimates given by the algorithms by means of the distance measures. We perform the reduced-form estimation by OLS and we use the estimated reduced-form residuals to obtain an estimate of the mixing matrix with each identification algorithm. For each new data generation, the reduced form coefficients and the entries of the matrix of contemporaneous relations are randomly sampled (from a normal and a uniform distributions respectively) and the stationarity of the process is ensured by discarding all those

---

<sup>7</sup>In this application we set a threshold of 0.1 to the allowed difference in size.

simulations where unit roots are displayed and replacing them with new simulations. We here consider reduced-form residuals that follow a uniform distribution.

### 3.3.1 Simulation results

We report the results of the simulation study in Appendix A. Tables A.12 through A.16 report the computed distance measures: for each relevant structure of the mixing matrix  $\Gamma_0^{-1}$  we average the result obtained for each of 500 Monte Carlo simulations. For simplicity, for the rest of this section, we will refer to the considered structures of the mixing matrix as structures a) to f). For brevity, we only present the results obtained considering a data generating process of a SVAR with three variables and three lags. Nonetheless, the main results can be generalized to different numbers of variables and/or lags.

- a) **Lower triangular  $\Gamma_0^{-1}$  with big diagonal.** We have that the Cholesky and MaxDiag are the most precise identification methods in this case, both under the TSE and the size measures. As for the sign measure, we have that the two MaxDiag and MaxDiag + order algorithms suffer from the fact that they do not set any coefficient to zero (even though some might be very close to zero). Given this structure of the mixing matrix, Cholesky and MaxDiag always identify correctly the contemporaneous relations. Finally, the Cholesky and MaxDiag always output a matrix that is considered “correct” according to our measure, while the other algorithms instead fail in those few (8% in this simulation) cases when the ordering of the variables estimated by VAR-LiNGAM is not the correct one.
- b) **Other recursive structure  $\Gamma_0^{-1}$  with big diagonal.** In this case we can see that only the algorithms that exploit the ordering estimated by the VAR-LiNGAM perform well. This is because the Cholesky and the MaxDiag algo-

rithms always output a lower triangular matrix and a matrix with the biggest elements on the diagonal respectively. Since in this case the true  $\Gamma_0^{-1}$  matrix is recursive but not lower triangular, this can be seen as a wrong specification of the recursive ordering of the variables. However, it must be noted that since the VAR-LiNGAM does not always retrieve the correct recursive ordering, it is possible for the other three algorithms to also fail. Nonetheless, we have that, for every case in which the VAR-LiNGAM estimates the order correctly, the matrices estimated by the three algorithms informed by it are considered “correct” by our distance measure.

- c) **Not recursive  $\Gamma_0^{-1}$  with big diagonal.** We can see that in this case the MaxDiag algorithm is the one that performs best by far. This is because the Cholesky and VAR-LiNGAM always impose a recursive structure, therefore setting to zero some coefficients which might in general be different from zero. Since the diagonal elements are bigger than the other elements, the MaxDiag algorithm is able to correctly solve the permutation and sign issues. It must be noted that the Cholesky algorithm seems to identify all the contemporaneous relations correctly, however, this simply depends on the column normalization we have adopted and should not be interpreted as a “merit” of the algorithm. We also show that the VAR-LiNGAM algorithm fails in retrieving the correct ordering of the variables (its guess rate is close to that of a random guess, which is 1/6, as there are 6 possible permutations with three variables). Nonetheless, it is able to correctly signal in the vast majority of cases that the actual structure is not recursive (see Tab.A.18, second column).
- d) **Lower triangular  $\Gamma_0^{-1}$  with small diagonal.** Since the structure is recursive, the Cholesky and VAR-LiNGAM algorithms perform well. However, since the biggest element of each row is not on the diagonal, the MaxDiag does not

solve correctly the permutation and sign indeterminacies. It must be noted that VAR-LiNGAM can sometimes fail to retrieve the correct ordering of the variables, which is what makes the measure obtained for the Cholesky + order and the MaxDiag + order algorithms respectively slightly worse than Cholesky and MaxDiag. It is striking how big the TSE is for the three algorithms based on independent component analysis. This has to do with the adopted normalization, since some coefficients might result in being multiplied by very small quantities thus giving rise to a very big squared error. This is why just very few wrong orderings estimated by VAR-LiNGAM are enough to make the TSE very large. However, by checking the other similarity measures, we can confirm that the VAR-LiNGAM is actually estimating the mixing matrix correctly most of the times.

- e) **Other recursive structure  $\Gamma_0^{-1}$  with small diagonal.** As in the case of a big diagonal, the order retrieved by the VAR-LiNGAM algorithm is effective in enhancing the Cholesky scheme. However, it does not work well for MaxDiag in general, since this fails in identifying the correct permutation of its columns as it always sets them so as to have the biggest entries on the diagonal. Therefore, since in this case the diagonal is “small”, the MaxDiag and the MaxDiag + order algorithms will in general not work. Similarly to case b), VAR-LiNGAM shows a very high TSE, even though it is correctly estimating the mixing matrix in 98% of the instances.
- f) **Not recursive  $\Gamma_0^{-1}$  with small diagonal.** We have that in this case all the algorithms fail to correctly estimate the  $\Gamma_0^{-1}$  matrix, and, as expected, the correct ordering is not retrieved by the VAR-LiNGAM algorithm, and, as in the analogous case of a big diagonal, it is close to a random guess. As in the case of a big diagonal, the correct contemporaneous relations identified by the

Cholesky and MaxDiag algorithms are due to the way in which we constructed the mixing matrix and should not be interpreted as a “merit” of the algorithms. Nonetheless, VAR-LiNGAM is able to signal in roughly two-thirds of the cases that the structure is not recursive.

We now look at the results obtained for two of the distance measures we have used in more detail. The total sum of squared residuals (see Tab.A.12) is an important metric since it gives insight on the precision of our estimates of the mixing matrix. That is, given two algorithms that both estimate the mixing matrix “correctly” given a particular structure, by looking at the TSE we can gauge which among the two better approximates the true matrix. An important aspect is represented by what degree of similarity, in terms of total squared distance, we should consider as “good” performance. By comparing the TSE and the a priori expectations table (Tab.1), in this application a threshold around 0.8 or 0.9 seems reasonable. It strikes out that the VAR-LiNGAM, contrarily to our expectations, seems to show a problem in terms of TSE with respect to the d) and e) structures. As we have seen, this can be attributed to the failure in retrieving the true recursive ordering in some of the Monte Carlo runs. When the Cholesky (Cholesky + order) and the MaxDiag (MaxDiag + order) algorithms perform a correct estimation (according to our overall correctness measure), Cholesky (Cholesky + order) is always at least as precise as MaxDiag (MaxDiag + order) and VAR-LiNGAM is the least precise. This is true both in terms of TSE and in terms of the “size” measure (see Tab.A.12 and Tab.A.14).

We then look at our measure of overall estimate correctness (see Tab.A.16). We can see that it reproduces very well our table of a priori expectations. Furthermore, it does so with a “large margin”, in the sense that not only we can find a threshold that classifies all the entries exactly as in that table (TRUEs corresponding to checkmarks), but we can also find a very large confidence band and still obtain the

same result. That is, we can classify as a check mark every entry of the table  $\geq 0.92$  and as a cross mark every entry  $\leq 0.22$  and still perfectly reproduce the table of a priori expectations (see Tab.A.17 in comparison with Tab.1). This means that no entry takes on values that are between these thresholds, which indicates that our measure captures well what we commonly mean when we say that a particular identification scheme “estimates the mixing matrix correctly given its structure.” This result allows us to confidently use the same measure to test the NGS algorithm, as explained in the next section.

Finally, we summarize the additional information provided by the VAR-LiNGAM algorithm in Tab.A.18. That is, in percentage terms, how many times the correct order is retrieved and how many times a message that signals the recursiveness of the structure is given. We can see that the performance of the VAR-LiNGAM, in terms of identifying the correct ordering of the variables, is well above 90% when the structure is recursive and it is close to random when it is not. In terms of correctly detecting recursiveness it is close to 100%, giving however roughly 30% of false positives when the diagonal is “small” (while giving a false positives rate of only about 5% when the diagonal is “big”).

In summary, we have that the best algorithm to use is the Cholesky (possibly refined by the ordering estimated with VAR-LiNGAM) in all cases except when we have a non-recursive structure. When we have a non-recursive structure and a “big” diagonal we should use MaxDiag, while if we have both a non-recursive structure and a “small” diagonal, none of the algorithms considered will retrieve the mixing matrix correctly. Our best guess would still be MaxDiag, since it estimates the entries of the matrix correctly, but we would have no way to solve the permutation and sign issues in a fully correct manner. Therefore, if we knew beforehand the structure of the mixing matrix we would proceed as follows:

- a) first-best: use Choleky; second-best: use MaxDiag; third-best: use Cholesky + order (this is third-best since the order might not be estimated correctly by VAR-LiNGAM).
- b) first-best: use Cholesky + order; second-best: use MaxDiag + order.
- c) first-best: use MaxDiag.
- d) first-best: use Choleky; second-best: use Cholesky + order (again, second-best since the order might not be estimated correctly).
- e) first-best: use Cholesky + order; second-best: use VAR-LiNGAM.
- f) best try: use MaxDiag.

Note that, as a preliminary step, we should always check that the reduced-form residuals are non-Gaussian (or that at most one of them is) since otherwise the algorithms based on ICA cannot be used reliably.

### 3.4 NGSi: a data-driven algorithm for shock identification

In this part of the section, we present NGSi (short for *Non-Gaussian Shock Identification*), a data-driven algorithm that performs VAR identification, and we show its performance across a wide variety of data generating processes. This algorithm is innovative in that it is able to infer the structure of the  $\Gamma_0^{-1}$  matrix and accordingly apply the most appropriate identification scheme, thereby not requiring any of the a priori assumptions on the structure (such as the recursiveness of the contemporaneous relations or the magnitude of the relations of each variable with itself, that is, the elements on the diagonal of the mixing matrix) which are instead necessary when applying the identification schemes that we have presented so far.

Similarly to what we did in section 3.3, to test the performance of our algorithm, we generate data from a VAR process which we restrict to be stationary, we store the mixing matrix (labeled according to its structure) used for each new data generation, we estimate a VAR by OLS from the generated data and use the reduced form residuals to estimate the mixing matrix via NGSF. We finally check the performance of the algorithm by comparing the true and the estimated  $\Gamma_0^{-1}$  matrices via the “correctness” distance measure. We do this for a very general set of data generating processes, varying both the number of variables and the number of lags. We here work in a setting where the reduced-form residuals are non-Gaussian. However, the algorithm is designed to test the Gaussianity of the residuals and to warn if more than one vector of residuals is found to be Gaussian, in which case, as we show, it can fail to retrieve the correct mixing matrix, since it is rooted in independent component analysis. The generalization of the algorithm to the case of Gaussian residuals is left for future research and a brief discussion on the topic will be given at the end of this section.

We begin by presenting a simple benchmark algorithm, which is at the core of NGSF, and we will then illustrate a series of ideas that will help shaping the final algorithm. The benchmark algorithm uses the Cholesky + order scheme when VAR-LINGAM signals that the true structure is recursive, and MaxDiag otherwise. That is:

### **BENCHMARK ALGORITHM**

#### **1. Test Gaussianity:**

- if more than one vector of residuals is Gaussian, WARN and STOP.
- if at most one vector of residuals is Gaussian proceed,

## 2. Test recursiveness:

- if the estimated structure is non-recursive, use MaxDiag and STOP.
- else use Cholesky + order and STOP.

As might be expected, a first problem with this simple algorithm is that it is subject to the possible failure of VAR-LiNGAM in recovering the true ordering of the variables when the true mixing matrix is of the type a), b), d) or e). However, in cases a) and d), this needs not be the case since the true ordering is simply the ordering of input of the data. In other words, the true structure is simply lower triangular. Therefore, if we were able to infer when we are in case a) and d), distinguishing them from cases b) and e) respectively, we would be able to significantly improve our algorithm. This can be done by comparing the outputs of the different identification schemes. In fact, when Cholesky and MaxDiag give the same result, the only possibility is that the true structure is of the type a) (see Tab.1). The implementation of a similar criterion distinguishing case d) from e) is left for future research.

Furthermore, we have no way of distinguishing cases c) and f). This is a crucial issue, since when the true mixing matrix is of the f) kind, it is likely that our identification will fail. Therefore, we refine our algorithm by introducing a warning when the mixing matrix is identified as f). Such warning is given when the  $\Gamma_0^{-1}$  matrix as estimated by MaxDiag has entries off the diagonal that are bigger (in absolute value) than the corresponding entries on the diagonal of each column. In practice, this aspect of the algorithm can be tuned (to improve the accuracy in giving the warning) to account for entries that are lower but very close to the ones on the diagonal.<sup>8</sup> To show that this idea is actually grounded we perform a small Monte

---

<sup>8</sup>In this particular application we tune the algorithm to give a warning when the largest off-diagonal entry of the matrix estimated via MaxDiag is  $> 0.90$ . We do this to optimize the true positives versus false positives trade-off.

Carlo study where we generate data from 100 VAR processes with a mixing matrix of the type c) and 100 of the type f) and check when the warning is given correctly.<sup>9</sup> We observe only 1% of c) instances wrongly labeled as f) instances (false positives) and 65% of f) instances correctly labeled as f) instances (true positives).

This can be considered a somewhat good result, however, by performing a quick error analysis, we observe that in fact the mislabeling occurs when the algorithm does not correctly classify the structure as non-recursive, rather than when it is unable to distinguish case c) from case f). This is coherent with the result of the first part of this section, where we showed that in case f) the non-recursive nature of the mixing matrix is correctly identified roughly 68% of the times (see Tab.A.18). Indeed, if we look at only those instances where the structure has been correctly labeled as non-recursive, the result is way more encouraging: we have that 90% of the f) instances are correctly labeled as f) instances.

We implement these ideas in the following refined algorithm:

### **REFINED ALGORITHM**

#### **1. Test Gaussianity:**

- if more than one vector of residuals is Gaussian, WARN and STOP.
- if at most one vector of residuals is Gaussian proceed,

#### **2. Test recursiveness:**

- if the estimated structure is non-recursive, use MaxDiag and STOP.  
Test whether likely to be in case f) rather than in case c) and WARN if case f) is identified.

---

<sup>9</sup>We here use three variables and three lags. We generalize this result at the end of the section (see Tab.B.21).

- if the estimated structure is recursive proceed,

### 3. Compare methods' estimates:

- if can infer that case a) is likely use Cholesky and STOP.
- else use Cholesky + order and STOP.

We report the performance of the benchmark and refined algorithms in Tab.B.19 and Tab.B.20.

The first thing that we notice is that the number of lags does not influence substantially the performance of the two algorithms. This in general (but not always) worsens just slightly when the number of lags is increased. Secondly, we have that the refined algorithm outperforms the benchmark algorithm if the mixing matrix is of type a) but not if it is of type b). This is due to the fact that in the former we introduced the possibility of directly labeling the structure as a), therefore increasing the probability of correctly labeling cases a) (true positives) but also of incorrectly labeling cases b), d) or e) as cases a) (false positives). The false positives issue is particularly severe when there are only two variables. However, when the number of variables is increased, it diminishes and ultimately only concerns b) instances, with false positives in d) and e) instances disappearing. This suggests that the refinement which compares the estimates obtained with Cholesky and simple ICA, which allows to directly skip to case a), should be used only if the number of variables is  $\geq 5$ .

A possible objection to the use of this refinement is that, in real-world applications, b) instances are likely to be more common than a) instances, which can be seen as a “lucky” case that simply depends on the ordering with which the variables are collected and arranged prior to applying the algorithm. In other terms, out of all the possible permutations of the variables, only one would be of the a) type while all the others would be of the b) type. Therefore, if the ordering with which the

data is fed to the algorithm were simply random, it would be inconvenient to favor the correct identification of a) instances over b) instances. This issue is particularly severe when the number of variables is high. However, in most applications in which the true mixing matrix is thought to be recursive and with a big diagonal, the econometrician is likely to have at least an idea (driven by theoretical considerations) of which might be the correct ordering of the variables. The refinement, by increasing the probability of correctly identifying a) instances, can be seen as a way to incorporate prior belief (in a somewhat Bayesian fashion) on the ordering of the variables. This is obtained simply by feeding the data to the algorithm following the order which is thought to be more plausible. Instead, the benchmark algorithm is more convenient when the VAR identification is carried out in a fully a-theoretical fashion. Finally, the correct signaling that we might be in the presence of a structure of the mixing matrix of the type f) oscillates around 45 to 70% if we have at least three variables (see Tab.B.22). Almost no false positives are given, only if the true structure is of type c) some (albeit few) false positives are possible. We recall that, as illustrated above, given the method we are using to identify possible f) instances, this 30 to 55% rate of non-signaling is mostly due to the fact that the structure is not correctly identified as non-recursive rather than to the inability of distinguishing between cases c) and f).

Interestingly, the idea of comparing the outcome of the different identification schemes can be used to devise a different procedure to estimate the recursive ordering of the variables, instead of using the order estimated by VAR-LiNGAM. Indeed, as stressed above, we have that the  $\Gamma_0^{-1}$  matrix estimated by the Cholesky and by the MaxDiag schemes can be similar only when the true mixing matrix is of the a) kind. To gauge when the two estimates are similar we can as usual use the “correctness” measure. To estimate the ordering of the variables, we can therefore try all the

permutation of the variables and choose the one for which the estimates given by the Cholesky and the MaxDiag schemes are “the same” (that is, judged similar by the distance measure). In theory, there should be only one such ordering, however, in practice the “correctness” measure might output a 1 for more than one permutation. If this is the case, we can use either the TSE or the size similarity measures to decide which should be our final estimate (it will be the one giving the lowest tse or the biggest size measure respectively). If these still do not lead to a unique choice, we pick solutions that are suggested by both measures. Ultimately, if the estimate is still not unique, we pick a permutation at random among the remaining. Of course, it is possible (albeit unlikely) that no permutation is judged as possibly correct to begin with. When this is the case we simply revert to the use of VAR-LiNGAM. We can compare the performance of this method for finding the recursive ordering (in both its versions, TSE and size), to that of VAR-LiNGAM. The results are reported in Tab.2.

Number of variables	Number of lags	Big Diagonal						Small Diagonal					
		Lower triangular			Other recursive structure			Lower triangular			Other recursive structure		
		TSE	Size	VAR-LiNGAM	TSE	Size	VAR-LiNGAM	TSE	Size	VAR-LiNGAM	TSE	Size	VAR-LiNGAM
2	1	0.98	0.98	0.98	0.98	0.98	0.98	0.50	0.50	1.00	0.50	0.50	1.00
	2	0.98	0.98	0.98	0.98	0.98	0.98	0.50	0.50	1.00	0.50	0.50	1.00
	3	0.98	0.98	0.98	0.98	0.98	0.98	0.50	0.50	1.00	0.50	0.50	1.00
3	1	0.96	0.96	0.96	0.96	0.96	0.96	0.92	0.92	1.00	0.92	0.92	0.98
	2	0.98	0.98	0.96	0.98	0.98	0.96	0.92	0.92	0.96	0.94	0.94	0.98
	3	0.98	0.98	0.98	0.98	0.98	0.98	0.94	0.94	0.96	0.94	0.94	0.96
4	1	0.92	0.92	0.94	0.94	0.94	0.96	0.92	0.92	0.92	0.94	0.94	0.94
	2	0.92	0.92	0.94	0.92	0.92	0.92	0.96	0.96	0.96	0.98	0.98	0.98
	3	0.94	0.94	0.96	1.00	1.00	0.98	0.92	0.92	0.92	0.96	0.96	0.96
5	1	0.80	0.80	0.84	0.80	0.80	0.88	0.84	0.84	0.84	0.80	0.80	0.88
	2	0.82	0.82	0.88	0.80	0.80	0.90	0.80	0.80	0.80	0.80	0.80	0.80
	3	0.82	0.82	0.82	0.84	0.84	0.82	0.80	0.80	0.80	0.78	0.79	0.80

Table 2: *Fractions of correctly estimated recursive orderings.*

The first thing to notice is that this alternative way to find the recursive ordering of the variables should be used only when we are in cases a) or b) but not when we are in cases d) or e). This is as expected since when the diagonal is “small”, the Cholesky and the MaxDiag schemes do not give similar estimates. The two methods, TSE and size, always show the same performance. Finally, we have that the performance of

the new method introduced in this section (for a) and b) instances) is comparable to that of VAR-LiNGAM, with a slight increase in gain when the number of variables is  $\geq 4$  and the number of lags is  $\geq 3$ . Given these results, if we can find an effective way of distinguishing cases a) or b) from d) or e), we can marginally ameliorate our identification algorithm by using this alternative estimation of the ordering of the variables when the structure is of type a) or b) and when the number of variables is  $\geq 4$  and the number of lags is  $\geq 3$ . One possible way of distinguishing cases a) and b) (“big diagonal”) from d) and e) (“small diagonal”), is simply by checking if the biggest coefficient in each column of the estimate of the mixing matrix given by Cholesky is on the diagonal.

The performance of this last method in distinguishing the case of a big diagonal from that of a small diagonal when the structure is recursive is reported in Tab.3.

Number of variables	Number of lags	Big Diagonal		Small Diagonal	
		Lower triangular	Other recursive structure	Lower triangular	Other recursive structure
4	1	1.00	0.99	0.03	0.29
	2	1.00	0.99	0.03	0.29
	3	1.00	0.99	0.03	0.29
5	1	1.00	0.96	0.00	0.08
	2	1.00	0.96	0.00	0.08
	3	1.00	0.95	0.00	0.08
6	1	1.00	0.93	0.00	0.02
	2	1.00	0.93	0.00	0.02
	3	1.00	0.93	0.00	0.03

Table 3: *Fraction of instances labeled as “big diagonal” by looking at the diagonal of the Cholesky estimate. Average over 500 Monte Carlo iterations.*

This method is effective in labeling correctly cases a) or b), however it shows some false positives, that is, it wrongly labels some “small diagonal” instances as “big diagonal”. Nonetheless, this problem tends to disappear when the number of variables grows.

Wrapping up all of these results, we can now present the final NGSi algorithm, which runs differently depending on the number of variables: when  $k < 5$  the benchmark algorithm is applied (with the addition of warning when a possible f) structure

is detected), when  $k \geq 5$ , the refinement to identify case a) is used and the methods just illustrated are employed to estimate the recursive ordering of the variables (but only when the structure is identified as either a) or b), that is, in the case of a “big diagonal”, otherwise VAR-LiNGAM is used). Therefore, the NGSi algorithm works as follows:

### **NGSI ALGORITHM**

#### **1. Test Gaussianity:**

- if more than one vector of residuals is Gaussian, WARN and STOP.
- if at most one vector of residuals is Gaussian proceed,

#### **2. Test recursiveness:**

- if the estimated structure is non-recursive, use MaxDiag and STOP.  
Test whether likely to be in case f) rather than in case c) and WARN if case f) is identified.
- if the estimated structure is recursive proceed,

#### **3. Count the number of variables:**

- if there are less than five variables use Cholesky + order and STOP.
- if there are five or more variables proceed,

#### **4. Check comparisons:**

- if Cholesky similar to MaxDiag use Cholesky and STOP.
- else proceed,

### 5. Run Cholesky and check diagonal:

- if big diagonal identified use Cholesky + alternative order and STOP.
- if small diagonal identified use Cholesky + VAR-LiNGAM order and STOP.

We report the performance of the NGS algorithm in Tab.B.21. As we can see, it combines the best of the benchmark and the refined algorithm, showing marginal improvements with respect to both. We have that the performance is outstanding when there are up to four variables (“correctness” ranging from 86% to 100%) for all possible structures of the mixing matrix except f). In general, the more the variables and the worse the performance of our algorithm, as it is natural since there are more entries to be estimated in the  $\Gamma_0^{-1}$  matrix. This can be due either to the inability in recovering the correct ordering of the variables (when the structure is recursive) or the inability in recovering the correct coefficients of the mixing matrix (when the structure is not recursive). Nonetheless, the performance can be gauged satisfactory in most situations, at least up to six variables considered and when we are not in the presence of a structure of the f) type. Once again, we can see that the number of lags does not have a significant influence on the performance of the algorithm, or, to put it in different terms, it impacts the performance way less than the number of variables.

Since f) is the only structure of the mixing matrix with which the algorithm often fails, we look at how well NGS signals that an f) structure is likely and we report the results in Tab.B.22. When there are only two variables the signaling is rather poor, however, this is not a big problem since with two variables NGS performs rather well (“correctness” equal to 71%) even in case f). The best signaling rate is shown with three to four variables: 60 ~ 70%. With more variables the correct warning drops to 45 ~ 65%. However, almost no mislabeling as f) (false positives)

is shown when the true structure is not of the f) type. There is likely to be space for improvement regarding the efficacy of this warning, which is left for future research.

As a final note, since in our implementation the NGSi algorithm relies on the fastICA algorithm to perform the preliminary ICA decomposition, it is good practice to run the algorithm several times and check that it always converges to the same (or a set of similar) solutions. Indeed, as illustrated in section 3.2, since fastICA relies on a random initialization, when the algorithm is run various times it is possible for the resulting mixing matrices to differ, thereby leaving the identification undetermined.

Appendix C. provides six examples (one for each structure of the mixing matrix) of VAR identification via NGSi. The algorithm has been implemented in R.

### 3.4.1 The case of Gaussian residuals

The main shortcoming of the NGSi algorithm is that it does not generalize to the case of Gaussian reduced-form residuals. This is because, as underlined numerous times, independent component analysis fails if more than one vector of residuals is Gaussian, therefore leading to wrong estimation of the mixing matrix in cases c) and f), and general inability to distinguish between recursive and non-recursive structure as well as inability to find the true recursive ordering of the variables via VAR-LiNGAM. We show this by modifying our algorithm so that it does not stop when the residuals are Gaussian and by evaluating its performance under Gaussianity. The results are reported in Tab.B.23, where it is shown that the algorithm performs rather poorly when the reduced-form residuals are indeed Gaussian.

However, an interesting question is “how actually likely is it, in real-world applications, for more than one vector of reduced-form residuals to show Gaussianity?” To try to give a tentative answer, we run a subset of all the possible VARs obtainable with the FRED-MD and the FRED-QD datasets and test the Gaussianity of their reduced-form residuals. We do this by selecting 10000 combinations of variables at

random, estimating a VAR by OLS and testing Gaussianity via the Jarque-Bera test. In Tab.4, we report the fraction of VARs thus obtained for which for more than one vector of residuals the null hypothesis of Gaussianity is not rejected at the 90% confidence level as well as the fraction of the total possible combinations that the 10000 combinations that we check represent. We repeat the exercise for 2 to 6 variables. Prior to performing this test, we impute the missing values in the FRED-MD and the FRED-QD datasets via the EM algorithm, as described in section 4.2.

Number of variables	FRED-MD		FRED-QD	
	Fraction of Gaussian VARs	Fraction of total combinations	Fraction of Gaussian VARs	Fraction of total combinations
2	0.000	1.000	0.003	0.358
3	0.000	0.035	0.011	0.005
4	0.000	0.001	0.030	0.000
5	0.000	0.000	0.047	0.000
6	0.000	0.000	0.289	0.000

Table 4: *Fraction of VARs that display more than one Gaussian vector of residuals. Samples of 10000 VARs.*

Although the results for  $\geq 4$  variables cannot be judged robust, we can see that there is a clear trend. When using monthly data, we do not find any combination of variables for which the reduced-form residuals display Gaussianity, while for quarterly data this fraction increases with the number of variables. Up to five variables, there are very few possible VARs that display Gaussianity. This is a good result for the NGS algorithm, as it means that its main drawback is unlikely to be problematic in practice in real-world applications.

### 3.5 Summary of the section

In this section, we have presented NGS, a data-driven algorithm that performs VAR identification, and we have shown its performance across a wide variety of data generating processes. This algorithm is innovative in that it is able to infer the structure of the  $\Gamma_0^{-1}$  matrix and accordingly apply the most appropriate identification scheme, thereby not requiring the a priori assumption (such as the recursiveness of

the contemporaneous relations or the magnitude of the relations of each variable with itself, that is, the elements on the diagonal of the mixing matrix) that are instead necessary when applying some of the identification schemes common in the literature. It only relies on the assumptions of shock independence and non-Gaussianity (the latter being testable).

To assess the performance of our algorithm, we have performed an extensive simulation study where we have artificially generated data according to a S-VAR process where we controlled the structure of the mixing matrix. We then estimated the resulting VAR by OLS and used the reduced-form residuals to obtain estimates of the mixing matrix via NGSJ. Finally, we checked the performance of the algorithm by comparing the true and the estimated  $\Gamma_0^{-1}$  matrices via a distance measure appropriately defined. We have done this for a general set of data generating processes, changing both the number of variables and the number of lags.

We have shown that our algorithm performs very well (“correctness” ranging from 55% to 100%) across different number of variables, different numbers of lags and different structures of the mixing matrix. The only exception is when the  $\Gamma_0^{-1}$  matrix is non-recursive structure and with a “small diagonal”, as we do not dispose yet of an identification scheme able to correctly estimate the mixing matrix in this case. Nonetheless, the algorithm is able to signal correctly roughly 45% to 70% of the time, depending on the number of variables, whether we are in the presence of such a mixing matrix.

While describing how we built the NGSJ algorithm, we have presented a series of ideas, including:

- A method to estimate whether the true mixing matrix is recursive or not, based on the entity of the pruning performed by VAR-LiNGAM;
- An alternative to VAR-LiNGAM to estimate the recursive ordering of the vari-

ables, based on the comparisons of the estimates given by different identification schemes. We have shown that in the case of “big diagonal” and with at least five variables, we are able to marginally outperform VAR-LiNGAM. When the number of variables is less than five, the performance is instead comparable to that of VAR-LiNGAM;

- To distinguish structures that have a “big diagonal” from structures with a “small diagonal”, we have shown that it is effective to look at whether the largest coefficients (in absolute value) of the estimates given by the Cholesky or the MaxDiag algorithms (for the cases of recursiveness and non-recursiveness respectively) are on the diagonal.

There are a series of hyperparameters of the NGSi algorithm that can be tuned (as, for example, the pruning threshold and the threshold used to identify a big diagonal). We here have performed only an elementary tuning of these parameters, based on theoretical considerations and trial-and-error. A proper tuning, which is likely to enhance significantly the performance of the algorithm, is left for future research. We speculate that the best combination of hyperparameter values is likely to change with the number of variables, the number of lags and the true structure of the mixing matrix.

Several aspects of the NGSi algorithm are likely to be improvable, perhaps implementing new ideas similar to those we have illustrated throughout the section. Nonetheless, we believe that the main contribution of this work is that of introducing a new framework to approach the identification problem in VARs, more performance-oriented rather than aprioristically theory-driven, upon which future research can build.

## 4 Application to the K+S model by Dosi et al. (2015)

This section is dedicated to an application of the validation procedure presented in section 2 to the “K+S” model (Dosi et al. 2015), where the identification of the causal structures is carried out by means of the NGIS algorithm (see section 3).

### 4.1 The model

The Schumpeter meeting Keynes (“K+S”) model by Dosi et al. (2015) is an extension of two previous versions of evolutionary agent-based models: Dosi et al. (2010) and Dosi et al. (2013). Its distinguishing characteristic is that it combines Schumpeterian theories of firm-specific, endogenous innovations with typical Keynesian features of demand-generation. Such framework aims at investigating the mutual interaction of supply-side factors with demand-side factors and their effect in the short and in the long run, and in particular of public policies. The model is able to reproduce a long list of stylized facts and, from the data generated from it, it is possible to build a cross-correlation table close to the one usually computed with the US observed data.

The model is made up of heterogeneous economic agents, divided into consumption-good firms, capital-good firms, consumers/workers, commercial banks, a Central Bank and the public sector. In the supply side of the economy, firms in the capital-good industry produce heterogeneous machines using labor as their only input. They introduce innovations or imitate competitors to augment labor productivity or to reduce costs, by investing a fraction of their revenues in research activity. Consumption-good firms buy capital machines and employ them together with workers to produce a homogeneous good. They plan their production and inventories on the basis of the expected demand, which is formed in a backward-looking manner, and invest in new machines when their capital stock is insufficient or obsolete. Suppliers advertise their machines’ prices and productivity to a restricted number of

firms, so that the capital goods market is characterized by imperfect information. Consumption-good firms can invest using internal or external resources. However, capital markets are modeled as being imperfect. Indeed, banks are unable to allocate credit optimally due to imperfect access to information about the creditworthiness of the applicants. This implies that the financial structure of firms matters (external funds are more expensive than internal ones) and firms may be credit rationed. The maximum amount of credit that a firm can receive is given by a loan-to-value ratio, while the cost of credit depends on the interest rate set by the Central Bank and on the creditworthiness of the firm. As a consequence, firms prefer to use internal resources, when available, instead of banks' loans. Commercial banks are heterogeneous in the number of clients and in other characteristics. They maintain both a mandatory and a strategical buffer and supply credit to consumption-goods firms according to the value of their equity and to their level of financial fragility. A bank goes bankrupt when its net worth falls below zero. In that case, the government intervenes and bails out the bank. Finally, consumption is given by the sum of the wages of employed workers and the subsidies that the government provides to unemployed workers. In addition to providing subsidies and saving banks, the government spends resources for repaying debts, and it collects taxes from firms, consumers and banks.

## 4.2 The data

We extract  $K_{AB} = 60$  time series from the “K + S” model, listed in Tab.D.24. We run  $M = 100$  Monte Carlo simulations of the model for which we employ the original parametrization used in Dosi et al. (2015), which we report in Tab.E.25 The unique difference across the 100 Monte Carlo replications is the random seed, with all the other parameters kept constant throughout. Each simulation is run for  $T_{AB} = 600$

time periods, so that we have  $\dim(V_{AB} = 100 \times 60 \times 600)$  and our dataset is balanced.

As a real-world counterpart against which we validate the model, we use the FRED-QD dataset proposed by M. McCracken and Ng (2020).<sup>10</sup> As in the original paper, for our analysis we use the September 2019 vintage. This contains  $K_{RW} = 248$  quarterly frequency series relative to the U.S. economy dating back to 1959:Q1 and it is benchmarked to previous work by Stock and Watson (2012).<sup>11</sup> We favour FRED-QD over FRED-MD (M. W. McCracken and Ng 2016) since it is more realistic to consider the agent-based data as quarterly data rather than monthly data (consider that the agent-based series are updated with every new iteration of the model, which would be unlikely for series such as wages or prices if it were monthly series). Furthermore, FRED-MD does not include Gross Domestic Product, Consumption, Investment, Government spending, and other macroeconomic series that come from the National Income and Product Accounts (NIPA). While most of the series are actually collected as quarterly series, some are higher-frequency series that have been aggregated up to the quarterly frequency. Finally, the factors that can be extracted from it have been shown to bear good predictive power also when used for diffusion index forecasting exercises. The data is provided in levels and has not been previously transformed in any way.

FRED-QD is not a balanced dataset as 34 series show missing values at the beginning of the sample for a total of 1826 missing data points. The reasons why these series have missing observations at the beginning of the sample are various, with most of them not existing back in 1959:Q1 and only becoming available in 1960:Q1. Since no outlier adjustment has been made to the raw data, we check for outliers in the transformed series and remove them prior to constructing the factors,

---

<sup>10</sup>The dataset is freely available at <http://research.stlouisfed.org/econ/mccracken/fred-databases/>.

<sup>11</sup>A description of the FRED-QD variables, together with the suggested transformations applicable, is given in the Appendix of M. McCracken and Ng (2020).

therefore treating them on a par with missing data. We define an outlier as an observation that deviates from the sample median by more than ten interquartile ranges. We then transform the data according to the transformations suggested in M. W. McCracken and Ng (2016), thereby losing two observations at the beginning of the sample.

To obtain a balanced dataset, we impute missing values using the EM algorithm presented in Stock and Watson (2002). The data is demeaned and standardized and all missing values are initialized to zero, the unconditional mean. A  $T \times r$  matrix of factors  $F = (f_1, \dots, f_T)'$  and a  $N \times r$  matrix of loadings  $\Lambda = (\lambda_1, \dots, \lambda_N)'$  are estimated with PCA, exploiting the  $\Lambda' \Lambda / N = I_r$  normalization. The missing values in series  $i$  at time  $t$  are then updated to  $\hat{\Lambda}'_i \hat{f}_t$ . At this point each variable is re-multiplied by the standard deviation and the mean is re-added, so that the values that were missing can now be treated as observations for series  $i$  at time  $t$ . This procedure is iterated until the estimates of the factors converge to a fixed value.

To apply the EM algorithm, we must first select the relevant number of factors  $r$ . To this aim, we make use of the  $IC_{p_2}$  criterion developed in Bai and Ng (2002) which pertains to the  $IC_p$  class of criteria but is shown to have better finite sample properties. The  $IC_{p_2}$  criterion is defined as

$$IC_{p_2}(r) = \ln(V(r, \hat{F}^r)) + r \left( \frac{N+T}{NT} \right) \ln C_{NT}^2,$$

where  $C_{NT}^2 = \min(N, T)$ ,  $V(r, \hat{F}^r) = \frac{1}{N} \sum_{i=1}^N \hat{\sigma}_i^2$ ,  $\hat{\sigma}_i^2 = \hat{e}_i' \hat{e}_i / T$  and  $\hat{e}_i$  is the  $T \times 1$  vector of estimated residuals deriving from the regression  $X_i = F^r \lambda_i^r + e_i$ .

Since the  $IC_{p_2}(r)$  value can only be computed on a balanced panel, it must be calculated after the imputation step via the EM algorithm, which, however, requires the specification of the number of factors  $r$ . Therefore, for coherence, when we compute  $IC_{p_2}(r)$  for  $r = 1, \dots, r_{max}$ , for every new value of  $r$ , we impute the missing

values using that same value in the EM step. With this procedure we find  $r = 7$  to be the optimal number of static factors.

After the preprocessing of the data, we dispose of a quarterly panel with no outliers, made up of 248 series spanning the period 1959:Q1 to 2019:Q4. However, for full comparability with Guerini and Moneta (2017), we consider up to 2014:Q2, implying a time series length  $T_{RW} = 222$ .

#### 4.2.1 Analysis of the factors

For a better understanding and comparison of the properties of the agent-based and the real-world datasets, we study the static factors that can be extracted from each. We begin by looking at the real-world dataset. We plot the first 7 factors (whose PCA estimate is obtained as the last iteration of the EM algorithm) in panel (a) of Fig.F.14. These account for roughly 47% of the variation in the whole dataset and the contribution of each individual factor is given in Fig.F.15, panel (a). Prior to estimating the factors, we have stationarized the series according to the transformation suggested by M. McCracken and Ng (2020).

Following M. W. McCracken and Ng (2016), to interpret the factors we regress each  $i$ -th series on the set of  $r = 7$  factors for  $k = 1, \dots, r$ . This gives  $R_i^2(k)$  for  $i = 1, \dots, 248$ . We then compute the incremental explanatory power of factor  $k$  for series  $i$  as  $mR_i^2(k) = R_i^2(k) - R_i^2(k-1)$ , for  $k = 2, \dots, r$  with  $mR_i^2(1) = R_i^2(1)$ . We finally calculate the average importance of each factor  $k$  as  $mR^2(k) = \frac{1}{N} \sum_{i=1}^N mR_i^2(k)$ . The results are reported in Tab.F.26, together with the ten series with the highest  $mR_i^2(k)$  for factor  $k$ . We then study the explanatory power and the main characteristics of each factor, by considering the  $mR_i^2(k)$  associated with each variable.

Factor 1 accounts for 20% of the variation in the data and it can be labeled as a real activity factor as it is mainly associated with variables related to aggregate employment (USGOOD, PAYEMS, MANEMP, DMANEMP) and industrial

production (IPMANSICS, INDPRO), which have on average an  $mR_i^2(1)$  of about 0.77-0.87. Factor 2 explains around 7.6% of the data variability and it is mainly related to term interest rates spreads, both of government bonds (T5YFFM) and corporate bonds (AAAFFM), and inventories (BUSINVx) as well as housing permits and starts (PERMIT, PERMITS, HOUST). It is therefore associated with the forward-looking variables in the dataset. Factor 3's explanatory power is about 6.7%, and it is clearly related to price indices, as all of the top ten variables associated with this factor are price indices, and it can therefore be interpreted as an inflation factor. The interpretation of the remaining factors is less clear. Factor 4, which explains around 4.1% of the data variability, appears to be a second employment-oriented factor. Together with factors 5 to 7, it also shows considerable correlation with earnings and productivity series. The last three factors jointly explain about 9% of the data variation and correlate with employment, money and credit and household balance sheet series respectively. These seven factors together explain 47.12% of the data variability.

We then perform the converse exercise and for each factor we look at which variables are associated with the highest loadings. The results for the first two factors are shown in Fig.F.16, panel (a). As it is natural, the variables that mostly contribute to the formation of the factors coincide with the ones whose variability is mostly explained by the factors. Finally, Fig.F.17, panel (a), plots  $R^2(7)$ , the fraction of variation in each series explained by the seven factors, for each ordered variable. We can see that the relative importance of the common components varies across the different series. The first seven factors explain over 0.5 of the variation in 117 series (49%) and between 0.25 and 0.5 of the variation in 69 series (29%).

We now turn to the agent-based dataset. For simplicity, all the tables and the figures relative to this section (reported in Appendix F.) refer to a single typical

agent-based realization. Nonetheless, they generalize (with only minor differences) to the other Monte Carlo realizations also. Prior to computing the factors, we transform the series according to the transformations reported in Tab.D.24, found with the procedure outlined at the end of section 2.1.<sup>12</sup> As in the real-world counterpart, the ICp2 criterion finds 7 as optimal number of static factors on a Monte Carlo mode (8.18 as a Monte Carlo average, with a standard deviation of 2.80). However, the first seven factors explain on average 65% (with a standard deviation of 1.3%) of the variability in the data, which is a larger proportion than in the real-world dataset. Finally, we compute the  $mRi^2(k)$ s, reported in Tab.F.27.

The interpretation of the agent-based factors is less clear than that of the real-world factors. However, we observe some distinct patterns. Factor 1, which accounts for 16,5% of the data variability, is clearly associated with price indices (like `cpi` and `d.cpi`) and interest rates (`r`, `r.bonds`), which all show an  $mR_i^2(1)$  well above 0.75. Factor 2 explains around 15.6% of the data variability and it is related to real activity series (like `GDP` and `Creal`) and to debt (like `Deb` and `DefonGDP`). Factor 3's explanatory power is about 11.8%, and it is related to employment (`U`, `LD`), also through firm failures (`next2`, `bankr_LD_tot`), in particular in the second sector. Factor 4, which explains around 7.2% of the data variability, is clearly related to the banking sector, as all of the top ten variables associated with this factor relate to bank balance sheets or to bank failures series (with the exception of `DebonGDP`). Factor 5 is also associated to real activity (`Ir`, `GDP`, `Creal`), while the interpretation of factors 6 and 7 (which jointly explain 7.8% of the variability in the data) is less clear. Fig.F.16, panel (b), shows the biplot of the first two agent-based factors. Fig.F.17, panel (b), shows the variation in each series explained by the seven factors, which

---

<sup>12</sup>We exclude the variables `Nb_act`, `count_inflation_target`, `Loan_profit_share`, `count_def_rec2`, `count_def2` and `count_compact2` as in at least one simulation they are constant for the whole observation period (or at least after having discarded the initial burn-in observations, see section 4.3).

varies across the series. On a Monte Carlo average, they explain more than 50% of the variance in 39 series (72%), with a standard deviation of 2.09, and between 25% and 50% in 11 series (20%), with a standard deviation of 2.77.

In summary, we have that the main factors are present in both the real-world and the agent-based data, although their relative importance differs (that is, they appear in different orders). The real activity factor is the first in the real-world case and the second (and fifth) in the agent-based case; the price indices factor is the third in the real-world case and the first in the agent-based case; the employment factor is the third in the real-world case and the fourth in the agent-based case. Finally, in the agent-based data, we do not have a factor specifically related to interest rates, unlike the second real-world factor, but it appears “mixed” with the prices factor. These results indicate that the main forces driving the real-world economy are also present in the model, although their relative importance might not be matched quite correctly.

### 4.3 Dataset uniformity and ABM properties

As explained in the previous section, we have

$$\begin{cases} \dim(V_{RW}) = 1 \times 248 \times 222 \\ \dim(V_{AB}) = 100 \times 60 \times 600 \end{cases}$$

To fulfill the dataset uniformity requirement for the AB-data, we collect the last  $T_{RW} = 222$  observations, therefore discarding the first 378 observations, getting rid of possible transients. We consider all  $M = 100$  Monte Carlo simulations, as we will compare each to the unique realization of the RW-data. It is clear that we do not have a one-to-one correspondence between the agent-based and the real-world variables, as not all real-world variables have a precise agent-based equivalent and

vice versa. This is because, as a simplification of reality, the “K + S” model does not generate all the variables in FRED-QD. Furthermore, not all the variables generated by the model have a precise real-world counterpart that is collected by statistical agencies. However, we do have a correspondence between the main variables that we will consider in the VAR and FAVAR validation steps: aggregate consumption, gross private investments, unemployment rate, gross domestic product, current price index and effective federal funds rate. As for the rest of the variables, we will consider them when estimating the factors and we will be mainly interested in understanding which type of series are the main macroeconomic drivers of the variability that we observe in the datasets, an exercise for which it is not necessary to have a precise one-to-one correspondence among agent-based and real-world variables.

	Equilibrium	Ergodicity
$\Delta\log(C)$	92,1	96,1
$\Delta\log(I)$	95,5	91,2
U	89,8	90,4
$\Delta\log(Y)$	91,3	91,8
$\Delta\log(p)$	94,5	90,2
r	91,8	88,7
<i>All variables</i>	90,3	91,2

Table 5: *Percentages of non-rejection of statistical equilibrium and ergodicity (stationarized series).*

As explained in section 2.1, we check whether the ergodicity and equilibrium assumptions are supported by the data. Prior to performing the Kolmogorov–Smirnov tests, we stationarize the variables according to the transformations indicated in Tab.D.24. The results of the tests are presented in Tab.5, which reports the percentage of non-rejection of each pairwise comparison, for each stationarised series of interest. We also include a grand mean of all the 60 variables in the dataset, transformed according to the t-codes in Tab.D.24. For each series we have  $\frac{T \times (T-1)}{2} = 24531$  and  $T \times M = 22200$  pairwise comparisons respectively for the statistical equilibrium

and for the ergodicity tests. For all the series we have high values of non-rejection, which allows us to conclude that the assumptions about the model having reached a statistical equilibrium and producing ergodic series are reasonable.

#### 4.4 VAR validation results

Following Guerini and Moneta (2017), we consider aggregate consumption (C), gross private investments (I), unemployment rate (U), gross domestic product (Y), current price index (p) and effective federal funds rate (r). These correspond to PCECC96, GPDIC1, UNRATE, GDPC1, CPIAUCSL, FEDFUNDS and Creal, Ir, U, GDP, cpi, r in the real-world and the agent-based datasets respectively. We harmonize the magnitude of the time series by taking logs of the C, I, Y, p variables and expressing U and r in percentage terms. In doing this, we exclude three Monte Carlo simulations,  $m = \{55, 61, 89\}$ , from the agent-based dataset because in (at least) one period of the model, investment goes to zero, so that it cannot be log-transformed. We plot the variables in Fig.2.

(a) Real-world time series

(b) Typical agent-based time series (first Monte Carlo realization)

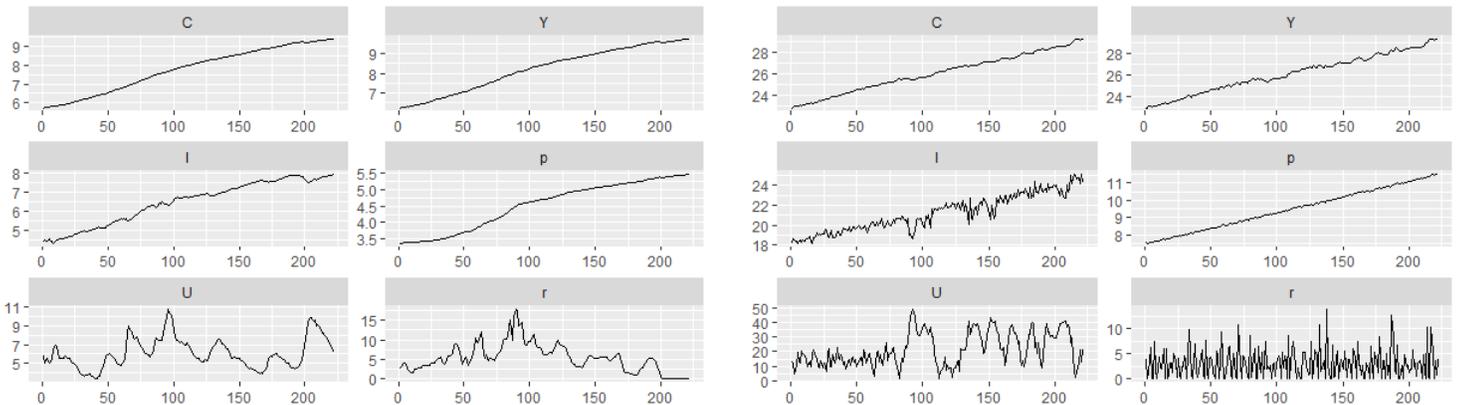


Figure 2: *Plot of the time series used in the VAR validation.*

We estimate the integration order of the variables adapting the procedure proposed by M. McCracken and Ng (2020) outlined at the end of section 2.1. That

is, we keep differentiating the variables as long as the augmented Dickey-Fuller test does not reject the null hypotheses of unit root and the number of differentiations necessary to reach stationarity is our estimate for the order of integration. We have that the test does not reject the null hypotheses of unit root in all of the real-world time series, with the estimated integration order being 2 for  $p$  and 1 for all the other variables. For the agent-based data, the evidence of the ubiquity of unit root is weaker: on a Monte Carlo average, the integration orders are 0.59 for  $C$ , 0.03 for  $I$ , 0.12 for  $U$ , 0.41 for  $Y$ , 0.10 for  $p$  and 0.00 for  $r$ . We display the results for the real-world data and a typical agent-based realization in Tab.6. We then select the number of lags using the Akaike Information Criterion (AIC) and the number of cointegrating relationships following the Johansen procedure. We obtain  $p_{RW} = 3$  and  $p_{AB} = 4$  as a Monte Carlo mode (3.89 on average with a standard deviation of 1.58) and 2 and 5 cointegrating relationships respectively for the real-world and the agent-based data (as a Monte Carlo mode). These differences do not create difficulties for the estimation of the structural models as they are reduced-form properties and they can be regarded simply as stylized facts not reproduced by the model.

We then proceed to the estimation of the reduced-form model by ordinary least squares and check that the VAR( $p$ ) process is stable. That is, we check if its reverse characteristic polynomial has no roots in or on the complex circle by looking at whether the eigenvalues of the estimated companion matrix have modulus less than one. We have that for the real-world VAR this condition is met, while 57% of the agent-based simulations display a unit root.

Tab.7 shows the results of the Shapiro-Wilk, Shapiro-Francia and the Jarque-Bera tests for normality on the VAR residuals, estimated from real-world data and from a typical Monte Carlo realization of the agent-based data. We have that for all the real-world residuals the tests reject the null hypothesis of normality. For the

(a) Real-world data

Variable	ADF p-value for levels	ADF p-value for 1st-differences	Critical level
C	0.99	0.01	0.05
I	0.84	0.01	0.05
U	0.05	0.01	0.05
Y	0.99	0.01	0.05
p	0.98	0.09	0.05
r	0.46	0.01	0.05

(b) Typical agent-based data (first Monte Carlo realization)

Variable	ADF p-value for levels	ADF p-value for 1st-differences	Critical level
C	0.17	0.01	0.05
I	0.01	0.01	0.05
U	0.01	0.01	0.05
Y	0.07	0.01	0.05
p	0.01	0.01	0.05
r	0.01	0.01	0.05

Table 6: *Augmented Dickey-Fuller tests.*

(a) Real-world data

Variable	Shapiro-Wilk p-value	Shapiro-Francia p-value	Jarque-Bera p-value
C	0.00	0.00	0.00
I	0.00	0.00	0.00
U	0.01	0.01	0.00
Y	0.01	0.00	0.00
p	0.00	0.00	0.00
r	0.00	0.00	0.00

(b) Typical agent-based data (first Monte Carlo realization)

Variable	Shapiro-Wilk p-value	Shapiro-Francia p-value	Jarque-Bera p-value
C	0.01	0.01	0.03
I	0.00	0.00	0.00
U	0.05	0.07	0.16
Y	0.01	0.01	0.09
p	0.00	0.00	0.00
r	0.14	0.07	0.08

Table 7: *Normality tests on the VAR residuals.*

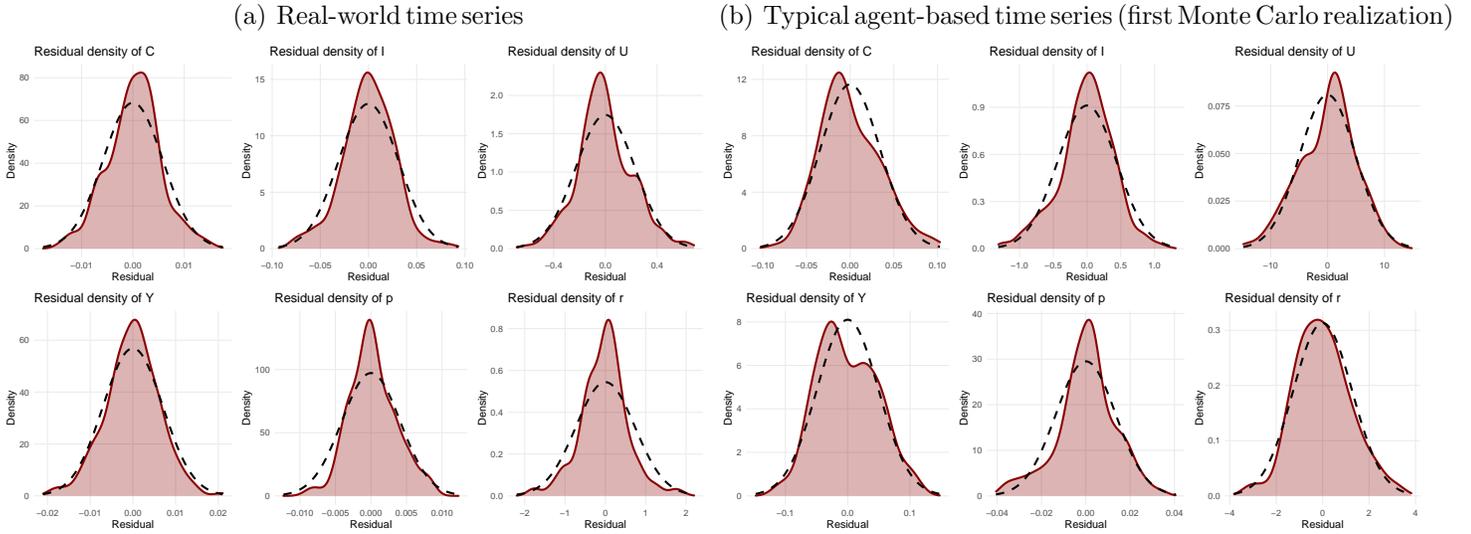


Figure 3: *VAR residuals distribution (solid red line) and normal distribution (dashed black line).*

agent-based residuals, at the 0.1 level of significance we have that for at most one residual the null cannot be rejected (funds rate or unemployment for the Shapiro-Wilk and the Jarque-Bera tests respectively). We report the empirical distributions of the same residuals in Fig.3. These results allow us to confidently employ the NGSi algorithm for identification (recall that at most one vector of residuals is allowed to show Gaussianity).

We thus perform the identification and check the convergence of NGSi by running the algorithm 1000 times with different seeds. We have that in the agent-based the convergence is exact while in the real-world case we obtain two  $\Gamma_0^{-1}$  matrices (respectively in the 79.7% and the 20.3% of the runs) which, however, differ only in one column and after the third decimal digit, so that the non-exact convergence is not of great concern. Both with the real-world and with the agent-based data NGSi detects a recursive structure and thus performs the ‘‘Cholesky + order’’ identification scheme. We have that the estimated ordering of the real-world variables is  $[p \rightarrow C \rightarrow Y \rightarrow I \rightarrow U \rightarrow r]$ , while in the the 97 agent-based simulations considered, we find 19 different orderings. The three most common, which account for 49% of the

orderings found, are  $[C \rightarrow p \rightarrow Y \rightarrow I \rightarrow U \rightarrow r]$ ,  $[C \rightarrow p \rightarrow Y \rightarrow I \rightarrow r \rightarrow U]$ ,  $[p \rightarrow C \rightarrow Y \rightarrow I \rightarrow r \rightarrow U]$ . We therefore have that the estimated  $\Gamma_0$ s can be seen as block recursive matrices, where  $p$ ,  $C$  and  $Y$  are “slow moving” variables, while  $U$  and  $r$  are “fast moving.” Even though this applies both for the real-world and the agent-based VARs, the exact real-world ordering is found in only 7 agent-based simulations. We then look at how often each variable is set in the same position as in the real-world counterpart: 28% for  $p$ , 35% for  $C$ , 60% for  $Y$ , 91% for  $I$ , 40% for  $U$  and 42% for  $r$ .

We then compute the impulse response functions as outlined in section 2.2.1 and pick  $H = 35$  as maximum horizon. We corroborate the identification of the structural VAR impulses by inspecting the forecast error variance decompositions (FEVD), shown in Fig.4. The panels show the percentage of forecast-error variance explained by the different impulses in the periods following an impulse of each type. If the forecast error variance of each variable can be largely explained by the impulse that is identified as that variable’s impulse, the identification is then likely to be correct. This is indeed the case for all our impulse responses, both real-world and agent-based, with the FEVD of consumption and prices being the most reliable in both cases.

After having completed the identification of the real-world and the agent-based VARs, we compute the four similarity measures as defined in section 2.5 across the 97 feasible Monte Carlo realizations of the “K+S” model. We report the mean and standard deviation of the similarity measures in Tab.8. For sake of completeness, we also report the similarity measures computed on the reduced-form coefficients, which, as can be seen, do not differ substantially from the “structural” similarity measures. The results related to the first similarity measure (sign-based, that is, concerning the direction of the causal relationship) suggest that the *Schumpeter*

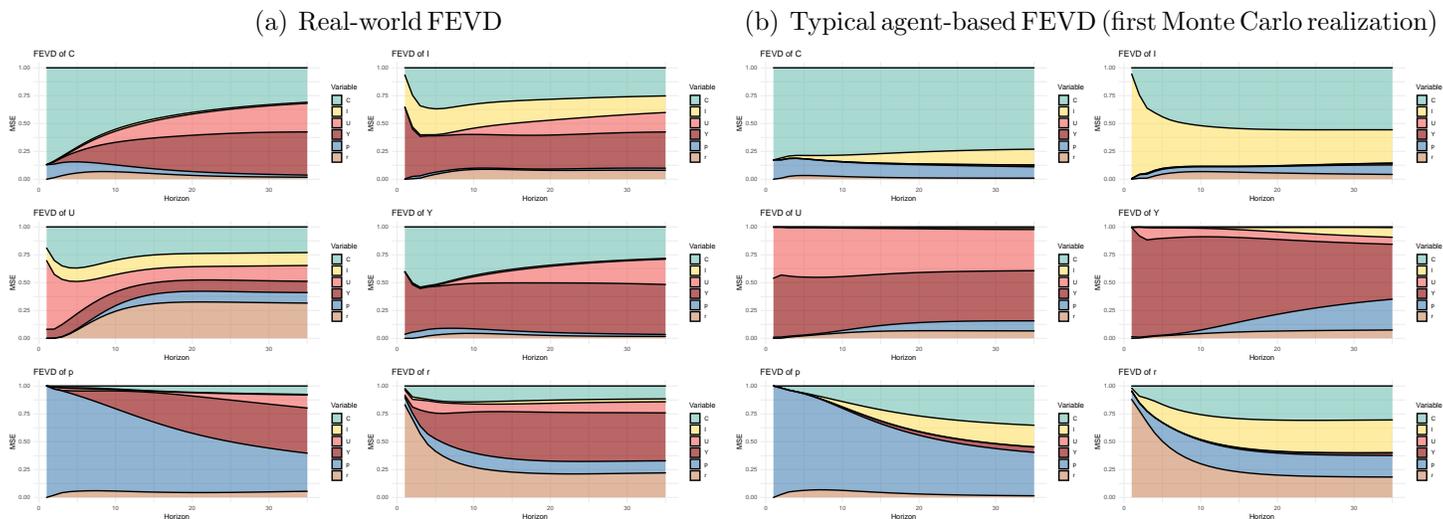


Figure 4: *VAR forecast error variance decompositions.*

*meeting Keynes* model is able to reproduce, on a Monte Carlo average, the 52% of the causal relations underlying the real-world data. Since the standard deviation is rather low, we can safely conclude that strong outliers are not present in either direction. The results related to the second similarity measure (size-based, that is, concerning the magnitude of the causal relations) show that a much higher proportion of causal relations are reproduced: 91% on a Monte Carlo average, with an even smaller standard deviation. Despite the fact that it might be intuitive to think that the size-based measure should be more stringent, this is not the case. This can be explained by the fact that the two standard deviation interval from the real-world parameter estimate might include the agent-based parameter estimate even when the two estimates have opposite sign. This is why, in our application, the third similarity measure (conjunction) is mainly driven by the sign-based measure. On a Monte Carlo average, we have that the agent-based model reproduces correctly 49% of the joint causal relations. Our results differ from the ones of Guerini and Moneta (2017), who find 79%, 79% and 67% for these three measures respectively and we speculate that this is due to the different estimation and identification procedures

employed.<sup>13</sup> Finally, we look at our proposed additional similarity measure. We have that the model generates, on a Monte Carlo average, impulse responses that lie for the 61% inside the confidence bands obtained for the real-world impulse responses. This result is shown in Fig.G.18. Overall, we consider these results as a positive indication for the model under validation, however, to reinforce this claim, it would be necessary to compare them with those coming from other models.

Similarity type	Reduced-form coefficients		Structural coefficients	
	$\mu$	$\sigma$	$\mu$	$\sigma$
sign-based	0.53	0.04	0.52	0.03
size-based	0.90	0.02	0.91	0.01
conjunction	0.49	0.04	0.47	0.03
irf			0.61	0.05

Table 8: *VAR validation results: mean and standard deviation of the four similarity measures across 97 Monte Carlo simulations.*

## 4.5 FAVAR validation results

We now proceed to the second validation of the model by exploiting a FAVAR approach. When using a VAR, we have focused only on six variables, while it can be argued that for a full validation of the model one should look at a larger set of series. FAVARs allow to consider a larger informative set by summarizing a large part of the information in the factors (see section 2.3).

Following Bernanke et al. (2005), we estimate two “latent” factors to include in the model and, for full comparability with section 4.4, we directly include as “observed” factors *gross domestic product*, *unemployment rate*, *current price index* and *effective federal funds rate*, so that the total number of variables is again six. As explained in section 2.3.1, we employ the two-step procedure proposed by Hae

<sup>13</sup>As argued in section 3, the NGIS algorithm is more general and leads to more precise estimates compared to the VAR-LiNGAM algorithm used by Guerini and Moneta (2017).

Hwang (2009) to estimate the latent factors, which we plot in Fig.5. The rest of the validation procedure follows as in section 4.4.

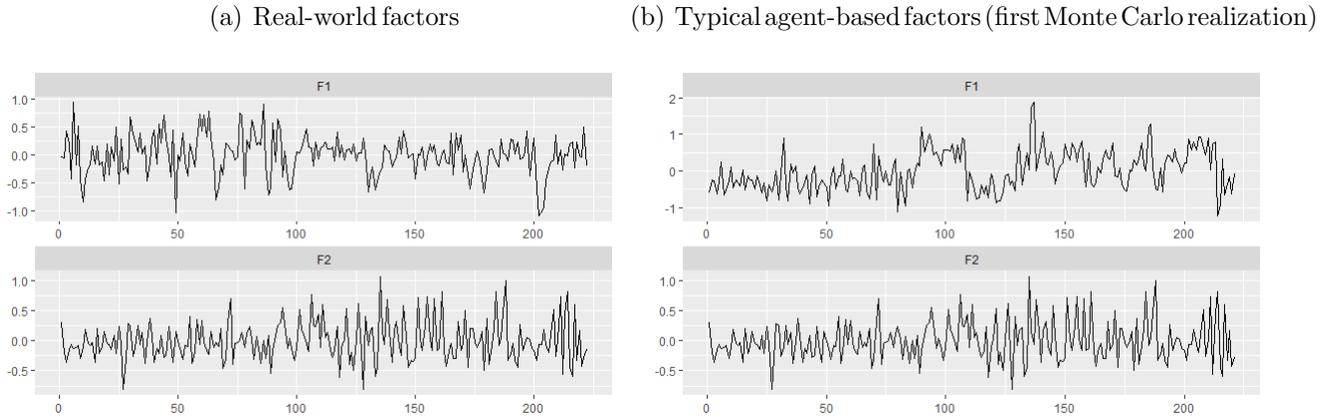


Figure 5: *Latent factors included in the FAVAR, estimated with the Hae Hwang (2009) procedure.*

We select the number of lags using the Akaike Information Criterion (AIC) and the number of cointegrating relationships following the Johansen procedure. We obtain  $p_{RW} = 3$  and  $p_{AB} = 3$  as a Monte Carlo mode (3.72 on average with a standard deviation of 1.02) and 4 and 5 cointegrating relationships respectively for the real-world and the agent-based data (as a Monte Carlo mode). We then proceed to the estimation of the reduced-form model by ordinary least squares and check the stability of the FAVAR(p) process. We have that the estimated real-world FAVAR has all eigenvalues of the companion matrix in modulus less than one, while 48% of the agent-based simulations display a unit root.

Tab.9 shows the results of the Shapiro-Wilk, Shapiro-Francia and the Jarque-Bera tests for normality on the FAVAR residuals, estimated from real-world data and from a typical Monte Carlo realization of the agent-based data. We have that for all the real-world residuals but the first factor, the tests reject the null hypothesis of normality, while for the agent-based residuals, at the 0.1 level of significance, we have that Y shows normality and U and r are close to normality. We report the

(a) Real-world data			
Variable	Shapiro-Wilk p-value	Shapiro-Francia p-value	Jarque-Bera p-value
Y	0.03	0.01	0.00
U	0.00	0.00	0.00
p	0.00	0.00	0.00
r	0.00	0.00	0.00
F1	0.90	0.51	0.41
F2	0.00	0.00	0.00
(b) Typical agent-based data (first Monte Carlo realization)			
Variable	Shapiro-Wilk p-value	Shapiro-Francia p-value	Jarque-Bera p-value
Y	0.87	0.62	0.86
U	0.08	0.09	0.08
p	0.00	0.00	0.00
r	0.10	0.06	0.03
F1	0.00	0.00	0.00
F2	0.02	0.02	0.02

Table 9: *Normality tests on the FAVAR residuals.*

empirical distributions of the same residuals in Fig.6. With these results in mind, we proceed to use the NGSi algorithm for the identification of the FAVAR.

We check the convergence of NGSi by running the algorithm 1000 times with different seeds. We have that in both the real-world and the agent-based data the convergence is exact. In both datasets, NGSi detects a recursive structure and thus performs the ‘‘Cholesky + order’’ identification scheme. We have that the estimated ordering of the real-world variables is  $[p \rightarrow Y \rightarrow F1 \rightarrow U \rightarrow F2 \rightarrow r]$  (which is coherent with the ordering found in the real-world VAR), while in the agent-based simulations considered, we find 11 different (but similar) orderings. The three most common, which account for 89% of the total orderings found, are  $[p \rightarrow Y \rightarrow F2 \rightarrow F1 \rightarrow r \rightarrow U]$ ,  $[p \rightarrow Y \rightarrow F2 \rightarrow F1 \rightarrow U \rightarrow r]$ ,  $[p \rightarrow Y \rightarrow F2 \rightarrow F1 \rightarrow U \rightarrow r]$ . We therefore have that, as in the VAR case, the estimated  $\Gamma_0$ s can be seen as block recursive matrices, where p and Y are ‘‘slow moving’’ variables and U and r are ‘‘fast moving.’’ We then look at how often each variable is set in the same position as in

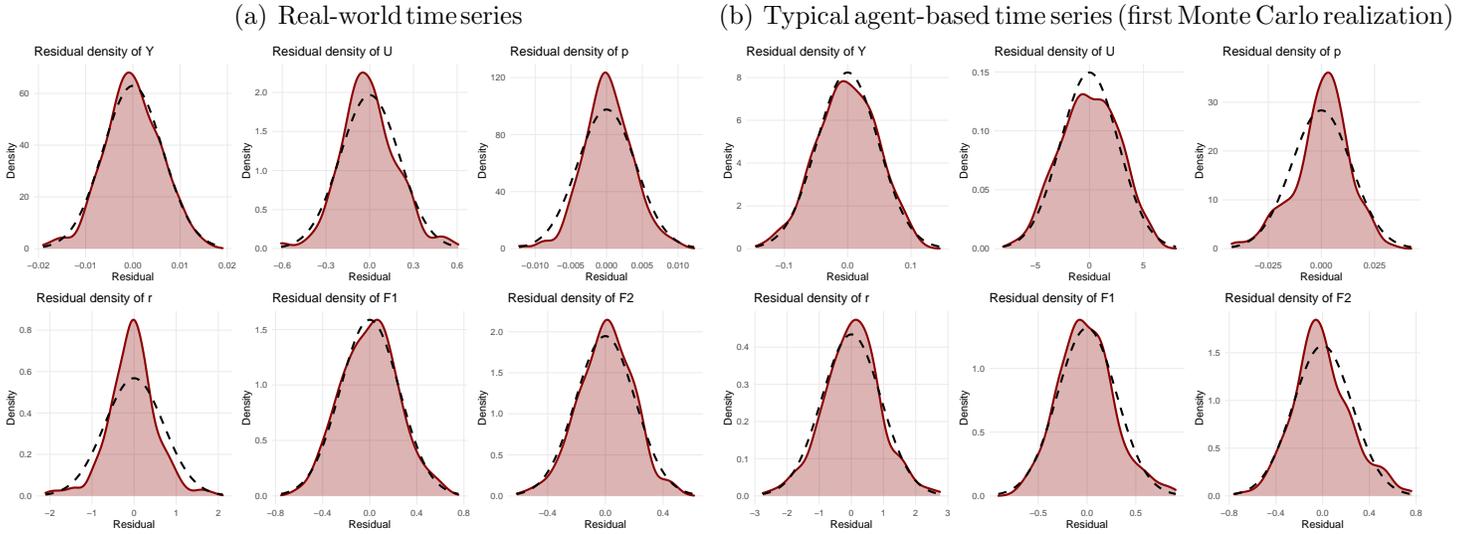


Figure 6: *FAVAR residuals distribution (solid red line) and normal distribution (dashed black line).*

the real-world case: 96% for p, 93% for Y, 30% for F1, 0% for U, 2% for F2 and 8% for r.

We then proceed to the computation of the impulse response functions and, as for the VAR, we corroborate the identification of the structural FAVAR impulses by inspecting the forecast error variance decompositions (FEVD), shown in Fig.7. The panels show the percentage of forecast-error variance explained by the different impulses in the periods following an impulse of each type. We have that the forecast error variance of each variable can be largely explained by the impulse that is identified as that variable’s impulse, in particular for the agent-based dataset.

After having completed the identification of the real-world and the agent-based FAVARs, we compute the four similarity measures as defined in section 2.5 across the Monte Carlo realizations of the “K+S” model. We report the mean and standard deviation of the similarity measures in Tab.10. As in the VAR case, the measures computed on the reduced-form coefficients do not differ substantially from the “structural” similarity measures. The results related to the first similarity measure (sign-based, that is, concerning the direction of the causal relationship) suggest that the

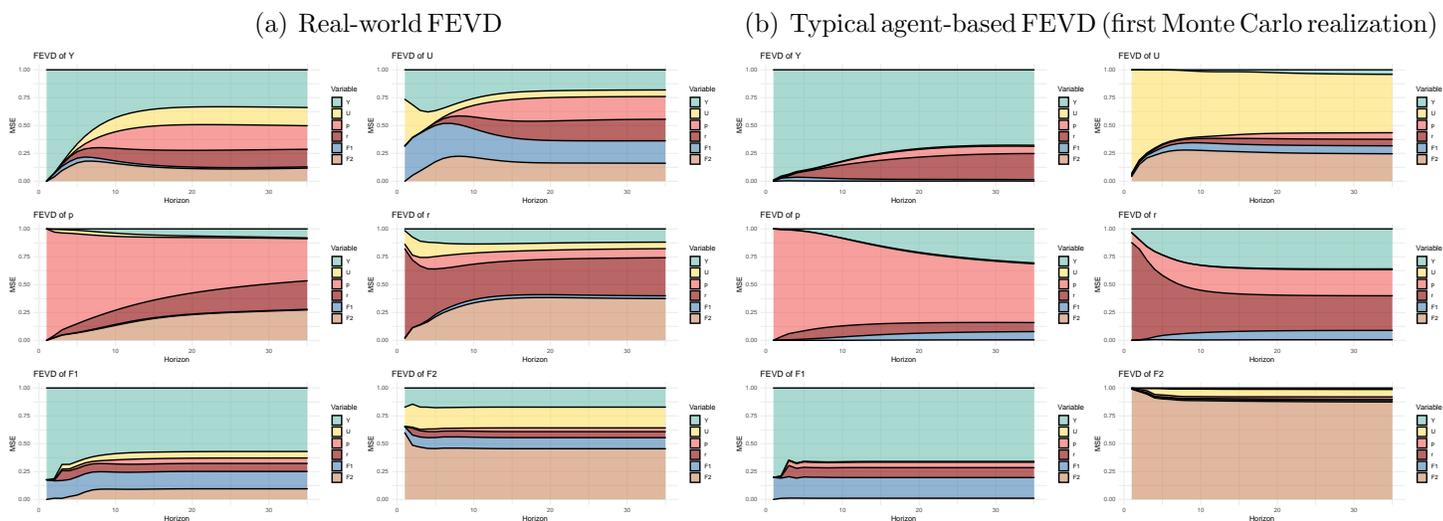


Figure 7: *FAVAR forecast error variance decompositions.*

*Schumpeter meeting Keynes* model is able to reproduce, on a Monte Carlo average, the 53% of the causal relations underlying the real-world data. The results related to the second similarity measure (size-based, that is, concerning the magnitude of the causal relations) show that a much higher proportion of causal relations are reproduced: 94% on a Monte Carlo average. The third similarity measure (conjunction) is mainly driven by the sign-based measure and, on a Monte Carlo average, we have that the agent-based model reproduces correctly 48% of the joint causal relations. Finally, we look at our proposed irf similarity measure. We have that the model generates, on a Monte Carlo average, impulse responses that lie for the 56% inside the confidence bands obtained for the real-world impulse responses. This result is shown in Fig.G.19. These results differ only slightly from the one obtained in section 4.4, although we here obtain values for the conjunction and the irf measures that are somewhat lower. This is coherent with the idea that the FAVAR approach is a “more severe test” than the simple VAR approach.

Similarity type	Reduced-form coefficients		Structural coefficients	
	$\mu$	$\sigma$	$\mu$	$\sigma$
sign-based	0.53	0.05	0.53	0.03
size-based	0.93	0.01	0.94	0.01
conjunction	0.48	0.05	0.48	0.03
irf			0.56	0.07

Table 10: *FAVAR validation results: mean and standard deviation of the four similarity measures across 97 Monte Carlo simulations.*

## 4.6 DFM validation results

As a final validation tool, we implement a dynamic factor model. To begin with, we select the optimal number of dynamic factors  $q$  following the procedure suggested by Amengual and Watson (2007), which applies the the ICp2 criterion by Bai and Ng (2002) on the residuals of a VAR model fitted on the estimated factors (see Eq.2.4.2).

In the real-world data we find  $q_{RW} = 6$  while in the agent-based data we have  $q_{AB} = 5$  on a Monte Carlo mode (5.16 as a Monte Carlo mean with a standard deviation of 2.8). Nonetheless, we estimate the DFM with six dynamic factors in the agent-based case also so as to have the same number of variables (and hence of impulse responses) as in sections 4.4 and 4.5. We then select only the impulse responses of the shocks in the dynamic factors on the six variables of interest (C, I, U, Y, p, r), listed in section 4.4. Finally, we compute the similarity measures, reported in Tab.11.

The results related to the sign-based similarity measure suggest that the model is able to reproduce, on a Monte Carlo average, the 44% of the causal relation underlying the real-world data. The results related to the size-based similarity measure show that a much higher proportion of causal relations are reproduced: 79% on a Monte Carlo average. The third similarity measure (conjunction) is again mainly driven by the sign-based measure and, on a Monte Carlo average, we have that the

model reproduces correctly 48% of the joint causal relations. Finally, we have that the model generates, on a Monte Carlo average, impulse responses that lie for the 78% inside the confidence bands obtained for the real-world impulse responses. This result is shown in Fig.G.20. Once again, these results are similar to the ones obtained in sections 4.4 and 4.5, with the first three similarity measures being slightly lower but the irf measure being considerably higher. With the exception of the latter, the overall results are therefore coherent with the idea that the DFM approach can be an “even more severe test” than the FAVAR approach.

However, in the case of a DFM, the irf similarity measure is perhaps not the best tool to assess the validation of the model, as it is likely to be influenced by the sign indeterminacy of the DFM impulse responses, which can lead to particularly wide confidence bands (especially in comparison to those obtained with the VAR and FAVAR approaches), as pointed out in section 2.4.

Similarity type	Reduced-form coefficients		Structural coefficients	
	$\mu$	$\sigma$	$\mu$	$\sigma$
sign-based	0.45	0.01	0.44	0.04
size-based	0.79	0.05	0.79	0.01
conjunction	0.41	0.03	0.40	0.03
irf			0.78	0.10

Table 11: *DFM validation results: mean and standard deviation of the four similarity measures across 97 Monte Carlo simulations.*

## 5 Conclusions

Recent developments in the VAR literature have demonstrated that it is possible to identify structural shocks by using only the distribution of reduced-form shocks and taking advantage of the information provided by its higher-order moments, making shock identification possible by relying solely on the assumptions of independence and non-Gaussianity of the structural shocks. However, the identification schemes proposed so far, which are rooted in independent component analysis, rely on additional assumption to solve the indeterminacy of the permutation and scaling of the columns of the mixing matrix. This work has proposed a strategy to solve these indeterminacies without relying on such auxiliary assumptions. After having reviewed some popular identification schemes and defined a way to ascertain when the estimate of the mixing matrix can be gauged correct, we have evaluated the performance of each of these schemes via an extensive simulation study. In doing so, we have identified which aspects of the structure of the mixing matrix (which convey specific assumptions) are relevant in determining the failure or success of the identification algorithms and we have studied the precision of each algorithm in different settings. This has served as a basis to develop NGSi (short for *Non Gaussian Shock Identification*), an innovative data-driven identification algorithm capable of inferring from the data which assumptions are likely to hold and accordingly applying the most appropriate (and precise) identification scheme, thereby only relying on the assumptions of shock independence and non-Gaussianity (the latter conveniently being testable).

To assess the performance of NGSi, we have performed a second simulation study, where we explored an exhaustive set of data generating processes by testing the algorithm in a wide variety of settings. We have shown that its performance is dependent on the structure of the mixing matrix, on the number of variables and, to

a marginal extent, also on the number of lags, with a “correctness” measure ranging from 55% to 100%. The only setting in which the algorithm is likely to fail is when the mixing matrix is such that its structure is non-recursive and for at least one variable there is a contemporaneous relation (with another variable) that is stronger than the contemporaneous relation with itself, as we do not dispose yet of an identification scheme able to correctly estimate the mixing matrix in this case. This is a key aspect on which we plan to develop our future investigations. Nonetheless, NGSi is able to signal correctly most of the time whether we are in the presence of such a mixing matrix and it outputs a warning when this is the case.

While describing how we built the NGSi algorithm, we have presented a series of techniques that can be put in practice to gain insight on the structure of the mixing matrix. We believe that similar ideas can be implemented to further enhance the performance of the algorithm or to devise new identification strategies. Furthermore, there are a series of hyperparameters of the NGSi algorithm that can be tuned. In the present work, we have performed only an elementary tuning of these parameters, based on theoretical considerations and trial-and-error. A proper extensive tuning, which is likely to further enhance the performance of the algorithm, is left for future research. We speculate that the best combination of hyperparameter values is likely to depend on the structure of the mixing matrix, the number of variables, and the number of lags. Moreover, we have shown that in real-world applications it is unlikely for the structural shocks to show Gaussianity, implying that the pivotal assumption upon which NGSi relies is likely to hold.

Several aspects of the algorithm are likely to be improvable, perhaps implementing new ideas similar to the ones we have presented. This being said, we are convinced that NGSi represents a significant step forward in the solution of the identification problem due to its general applicability and its outstanding performance (at least

in comparison to other identification schemes). Nonetheless, we believe the main contribution of this work to be the introduction of a new framework to approach the identification problem in VARs, more performance-oriented rather than aprioristically theory-driven, upon which future research can build.

Furthermore, in this work we have presented a new method to empirically validate simulation models that generate artificial time series data comparable with real-world data. The approach, which is based on the comparison of the causal structures estimated from the artificial and the real-world data, has extended the validation procedure proposed by Guerini and Moneta (2017) to the use of structural factor models, which, compared to standard SVARs, allow to consider a larger informative set, thereby leading to a more comprehensive validation assessment. Compared to the mere ex-post ability to reproduce a number of stylized facts, often used as main validation routine, the comparison of the causal relations constitutes a significant improvement in the assessment of a model, since a good matching between the causal structures incorporated in the model and the causal structures underlying the real-world data can provide better support to the policy statements drawn from the macroeconomic model under validation. Furthermore, this methodology is able to address both the problem of evaluating theoretical simulation models against the data and the problem of comparing different models in terms of their empirical reliability.

A critical aspect when implementing a VAR regards the choice of the variables to include and whether these can be considered a sufficient information set to recover the relevant causal structures. Indeed, a well-known problem in the traditional specification of VAR models is that only a small amount of variables can be directly included, as the number of parameters that need to be estimated rapidly increases with the number of variables. As a consequence, the choice of what variables to

consider is somewhat subjective and it constraints the researcher to exploit a thin informative set. By resorting to a factor-based approach, the problem of choosing which variables to consider is overshadowed since a big part of the information is included via the factors. In addition, the factor approach sometimes allows to get a more precise measure of given quantities which have a clear theoretical definition but cannot be distinctly observed in reality and, since the common components can be interpreted as a cleaner version of the variables that should be considered for structural analysis, hence free of measurement error, it allows for the recovery of structural shocks that are not contaminated by non-corresponding shocks (contamination which is instead possible in the case of simple VARs).

The present work has focused mainly on the validation of agent-based models, for which we have presented a first application to the model proposed by Dosi et al. (2015). Nevertheless, our methodology can be easily generalized to any simulation model able to generate enough time series to justify the use of a factor-based approach. As a real-world dataset against which to validate the model, we have used FRED-QD (M. McCracken and Ng 2020). Prior to the validation step, we have performed an in-depth analysis of the factors that can be extracted from each dataset, and we have shown that the main forces driving the real-world economy are also present in the model, although their relative importance is not entirely matched. We then implemented our proposed validation procedure, based on the comparison of the causal structures underlying the real-world data and the causal structures incorporated in the model. We repeated the validation exercise three times, implementing three different structural models. We first performed a benchmark validation by means of a standard VAR model, replicating the exercise carried out by Guerini and Moneta (2017), with minor differences. This suggests that the model is able to resemble between 47% and 61% of the causal relations entailed by a SVAR esti-

mated on real-world data. We then extended the framework to factor models by first implementing a Factor Augmented VAR and then a Dynamic Factor Model, which constitute “more severe” validation procedures. We have calculated that, according to the proposed similarity measures, the model is able to reproduce between 48% and 56% of the causal relations entailed by a SFAVAR estimated on real-world data and between 40% and 78% of the causal relations entailed by a DFM estimated on real-world data. We believe this to be a somewhat positive result for the Schumpeter meeting Keynes model but, in order to reinforce this claim, it would be necessary to compare this result with the ones obtained by validating other competing models.

Rather than trying to settle the validation issue in an ultimate manner, this work has aimed at setting a new benchmark upon which future research might build. We believe that the use of a plurality of methods, capable of bringing together complementary evidence, can provide the necessary support to the policy statements drawn from macroeconomic simulation models, such as agent-based models.

## References

- Ahn, Seung C and Alex R Horenstein (2013). “Eigenvalue ratio test for the number of factors”. In: *Econometrica* 81.3, pp. 1203–1227.
- Amengual, Dante and Mark W Watson (2007). “Consistent estimation of the number of dynamic factors in a large N and T panel”. In: *Journal of Business & Economic Statistics* 25.1, pp. 91–96.
- Bai, Jushan and Serena Ng (2002). “Determining the number of factors in approximate factor models”. In: *Econometrica* 70.1, pp. 191–221.
- Bai, Jushan and Serena Ng (2007). “Determining the number of primitive shocks in factor models”. In: *Journal of Business & Economic Statistics* 25.1, pp. 52–60.
- Barigozzi, Matteo and Matteo Luciani (2019). “Quasi maximum likelihood estimation of non-stationary large approximate dynamic factor models”. In: *arXiv preprint arXiv:1910.09841*.
- Bernanke, Ben S., Jean Boivin, and Piotr Elias (Feb. 2005). “Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach”. In: *The Quarterly Journal of Economics* 120.1, pp. 387–422.
- Boivin, Jean and Marc Giannoni (Dec. 2006). *DSGE Models in a Data-Rich Environment*. NBER Technical Working Papers 0332. National Bureau of Economic Research, Inc.
- Breitung, Jörg and Sandra Eickmeier (2006). “Dynamic factor models”. In: *Allgemeines Statistisches Archiv* 90.1, pp. 27–42.
- Brock, William A (1999). “Scaling in economics: a reader’s guide”. In: *Industrial and Corporate change* 8.3, pp. 409–446.

- Caiani, Alessandro, Antoine Godin, Eugenio Caverzasi, Mauro Gallegati, Stephen Kinsella, and Joseph E Stiglitz (2016). “Agent based-stock flow consistent macroeconomics: Towards a benchmark model”. In: *Journal of Economic Dynamics and Control* 69, pp. 375–408.
- Cooley, Thomas F and Stephen F LeRoy (1985). “Atheoretical macroeconometrics: a critique”. In: *Journal of Monetary Economics* 16.3, pp. 283–308.
- Dosi, Giovanni, Giorgio Fagiolo, Mauro Napoletano, and Andrea Roventini (2013). “Income distribution, credit and fiscal policies in an agent-based Keynesian model”. In: *Journal of Economic Dynamics and Control* 37.8, pp. 1598–1625.
- Dosi, Giovanni, Giorgio Fagiolo, Mauro Napoletano, Andrea Roventini, and Tania Treibich (2015). “Fiscal and monetary policies in complex evolving economies”. In: *Journal of Economic Dynamics and Control* 52, pp. 166–189.
- Dosi, Giovanni, Giorgio Fagiolo, and Andrea Roventini (2010). “Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles”. In: *Journal of Economic Dynamics and Control* 34.9, pp. 1748–1767.
- Dosi, Giovanni, Andrea Roventini, and Emanuele Russo (2019). “Endogenous growth and global divergence in a multi-country agent-based model”. In: *Journal of Economic Dynamics and Control* 101, pp. 101–129.
- Doz, Catherine and Peter Fuleky (2020). “Dynamic factor models”. In: *Macroeconomic forecasting in the era of big data*, pp. 27–64.
- Doz, Catherine, Domenico Giannone, and Lucrezia Reichlin (2011). “A two-step estimator for large approximate dynamic factor models based on Kalman filtering”. In: *Journal of Econometrics* 164.1, pp. 188–205.

- Doz, Catherine, Domenico Giannone, and Lucrezia Reichlin (2012). “A quasi–maximum likelihood approach for large, approximate dynamic factor models”. In: *Review of economics and statistics* 94.4, pp. 1014–1024.
- Fagiolo, Giorgio, Daniele Giachini, and Andrea Roventini (2020). “Innovation, finance, and economic growth: an agent-based approach”. In: *Journal of Economic Interaction and Coordination* 15.3, pp. 703–736.
- Fagiolo, Giorgio and Andrea Roventini (2008). *On the scientific status of economic policy: a tale of alternative paradigms*. Tech. rep. LEM Working Paper Series.
- Fernández-Villaverde, Jesús, Juan Francisco Rubio-Ramírez, and Frank Schorfheide (2016). “Solution and estimation methods for DSGE models”. In: *Handbook of macroeconomics*. Vol. 2. Elsevier, pp. 527–724.
- Forni, Mario, Domenico Giannone, Marco Lippi, and Lucrezia Reichlin (2009). “Opening the black box: Structural factor models with large cross sections”. In: *Econometric Theory* 25.5, pp. 1319–1347.
- Giannone, Domenico, Lucrezia Reichlin, and Luca Sala (2006). “VARs, common factors and the empirical validation of equilibrium business cycle models”. In: *Journal of Econometrics* 132.1, pp. 257–279.
- Gourieroux, Christian, Alain Monfort, et al. (2014). *Revisiting identification and estimation in structural VARMA models*. Tech. rep.
- Grazzini, Jakob (2012). “Analysis of the emergent properties: Stationarity and ergodicity”. In: *Journal of Artificial Societies and Social Simulation* 15.2, p. 7.
- Guerini, Mattia and Alessio Moneta (2017). “A method for agent-based models validation”. In: *Journal of Economic Dynamics and Control* 82, pp. 125–141.

- Hae Hwang, Hae (2009). “Two-step estimation of a factor model in the presence of observable factors”. In: *Economics Letters* 105.3, pp. 247–249.
- Hyvärinen, Aapo (1999). “Fast and robust fixed-point algorithms for independent component analysis”. In: *IEEE transactions on Neural Networks* 10.3, pp. 626–634.
- Hyvärinen, Aapo and Erkki Oja (2000). “Independent component analysis: algorithms and applications”. In: *Neural networks* 13.4-5, pp. 411–430.
- Hyvärinen, Aapo, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer (2010). “Estimation of a structural vector autoregression model using non-gaussianity.” In: *Journal of Machine Learning Research* 11.5.
- Kapetanios, George (2010). “A testing procedure for determining the number of factors in approximate factor models with large datasets”. In: *Journal of Business & Economic Statistics* 28.3, pp. 397–409.
- Kilian, Lutz and Helmut Lütkepohl (2017). *Structural Vector Autoregressive Analysis*. Themes in Modern Econometrics. Cambridge University Press.
- Lamperti, Francesco, Giovanni Dosi, Mauro Napoletano, Andrea Roventini, and Alessandro Sapio (2018). “Faraway, so close: coupled climate and economic dynamics in an agent-based integrated assessment model”. In: *Ecological Economics* 150, pp. 315–339.
- Lanne, Markku and Helmut Lütkepohl (2010). “Structural vector autoregressions with nonnormal residuals”. In: *Journal of Business & Economic Statistics* 28.1, pp. 159–168.

- Lanne, Markku, Mika Meitz, and Pentti Saikkonen (2017). “Identification and estimation of non-Gaussian structural vector autoregressions”. In: *Journal of Econometrics* 196.2, pp. 288–304.
- Lanne, Markku and Pentti Saikkonen (2013). “Noncausal vector autoregression”. In: *Econometric Theory* 29.3, pp. 447–481.
- Lippi, Marco (2019). “Validating DSGE models with VARs and Large-Dimensional Dynamic Factor Models”. In: *unpublished manuscript, presented at Sant’Anna School of Advanced Studies*.
- McCracken, Michael and Serena Ng (2020). *FRED-QD: A quarterly database for macroeconomic research*. Tech. rep. National Bureau of Economic Research.
- McCracken, Michael W. and Serena Ng (2016). “FRED-MD: A monthly database for macroeconomic research”. In: *Journal of Business & Economic Statistics* 34.4, pp. 574–589.
- Moneta, Alessio, Doris Entner, Patrik O Hoyer, and Alex Coad (2013). “Causal inference by independent component analysis: Theory and applications”. In: *Oxford Bulletin of Economics and Statistics* 75.5, pp. 705–730.
- Onatski, Alexei (2009). “Testing hypotheses about the number of factors in large factor models”. In: *Econometrica* 77.5, pp. 1447–1479.
- Onatski, Alexei (2010). “Determining the number of factors from empirical distribution of eigenvalues”. In: *The Review of Economics and Statistics* 92.4, pp. 1004–1016.
- Reichlin, Lucrezia, Domenico Giannone, Luca Sala, et al. (2005). *Monetary policy in real time*. Tech. rep. ULB–Universite Libre de Bruxelles.

- Reijer, Ard HJ den, Jan PAM Jacobs, and Pieter W Otter (2020). “A criterion for the number of factors”. In: *Communications in Statistics-Theory and Methods*, pp. 1–7.
- Shimizu, Shohei, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen (2006). “A linear non-Gaussian acyclic model for causal discovery”. In: *Journal of Machine Learning Research* 7.Oct, pp. 2003–2030.
- Sims, Christopher A (1980). “Macroeconomics and reality”. In: *Econometrica: journal of the Econometric Society*, pp. 1–48.
- Stock, James H and Mark W Watson (2002). “Macroeconomic forecasting using diffusion indexes”. In: *Journal of Business & Economic Statistics* 20.2, pp. 147–162.
- Stock, James H and Mark W Watson (2005). *Implications of dynamic factor models for VAR analysis*. Tech. rep. National Bureau of Economic Research.
- Stock, James H and Mark W Watson (2012). *Disentangling the Channels of the 2007-2009 Recession*. Tech. rep. National Bureau of Economic Research.
- Windrum, Paul, Giorgio Fagiolo, and Alessio Moneta (2007). “Empirical validation of agent-based models: Alternatives and prospects”. In: *Journal of Artificial Societies and Social Simulation* 10.2, p. 8.

## Appendix A. Assessment of the identification schemes

TSE		Cholesky	Cholesky + order	MaxDiag	MaxDiag + order	VAR_LiNGAM
Big diagonal	Lower triangular	0.004	0.366	0.005	0.368	0.391
	Other recursive structure	5.501	0.361	5.620	0.362	0.387
	Not recursive	0.909	5.650	0.006	5.545	5.750
Small diagonal	Lower triangular	0.098	0.800	283275.300	283278.907	751.727
	Other recursive structure	1554.306	4.047	1333113.981	1330147.937	752.962
	Not recursive	9046.664	9349.754	28946.836	28807.151	9382.769

Table A.12: *Total squared error. Monte Carlo average over 500 simulations.*

SIGN		Cholesky	Cholesky + order	MaxDiag	MaxDiag + order	VAR_LiNGAM
Big diagonal	Lower triangular	0.992	0.956	0.658	0.635	0.946
	Other recursive structure	0.461	0.957	0.358	0.637	0.947
	Not recursive	0.574	0.425	0.984	0.648	0.431
Small diagonal	Lower triangular	0.995	0.987	0.573	0.569	0.961
	Other recursive structure	0.455	0.986	0.375	0.564	0.959
	Not recursive	0.554	0.406	0.734	0.569	0.398

Table A.13: *Percentage of correct sign. Monte Carlo average over 500 simulations.*

SIZE		Cholesky	Cholesky + order	MaxDiag	MaxDiag + order	VAR_LiNGAM
Big diagonal	Lower triangular	1.000	0.953	1.000	0.953	0.927
	Other recursive structure	0.331	0.954	0.339	0.954	0.928
	Not recursive	0.544	0.270	1.000	0.390	0.298
Small diagonal	Lower triangular	1.000	0.989	0.732	0.722	0.898
	Other recursive structure	0.382	0.986	0.324	0.720	0.897
	Not recursive	0.550	0.369	0.606	0.400	0.378

Table A.14: *Percentage of close-to-correct size. Monte Carlo average over 500 simulations.*

CONTEMPORANEOUS RELATIONS		Cholesky	Cholesky + order	MaxDiag	MaxDiag + order	VAR_LiNGAM
Big diagonal	Lower triangular	1.000	0.947	1.000	0.947	0.947
	Other recursive structure	0.205	0.947	0.205	0.947	0.947
	Not recursive	1.000	0.323	1.000	0.323	0.323
Small diagonal	Lower triangular	1.000	0.987	1.000	0.987	0.987
	Other recursive structure	0.211	0.984	0.211	0.984	0.984
	Not recursive	1.000	0.321	1.000	0.321	0.321

Table A.15: *Percentage of correct contemporaneous relations. Monte Carlo average over 500 simulations.*

OVERALL SIMILARITY		Cholesky	Cholesky + order	MaxDiag	MaxDiag + order	VAR_LiNGAM
Big diagonal	Lower triangular	1.000	0.920	1.000	0.920	0.920
	Other recursive structure	0.000	0.920	0.006	0.920	0.920
	Not recursive	0.038	0.022	1.000	0.220	0.016
Small diagonal	Lower triangular	1.000	0.980	0.196	0.186	0.980
	Other recursive structure	0.000	0.976	0.000	0.188	0.976
	Not recursive	0.006	0.002	0.188	0.074	0.002

Table A.16: *Overall similarity. Monte Carlo average over 500 simulations.*

OVERALL SIMILARITY		Cholesky	Cholesky + order	MaxDiag	MaxDiag + order	VAR_LiNGAM
Big diagonal	Lower triangular	TRUE	TRUE	TRUE	TRUE	TRUE
	Other recursive structure	FALSE	TRUE	FALSE	TRUE	TRUE
	Not recursive	FALSE	FALSE	TRUE	FALSE	FALSE
Small diagonal	Lower triangular	TRUE	TRUE	FALSE	FALSE	TRUE
	Other recursive structure	FALSE	TRUE	FALSE	FALSE	TRUE
	Not recursive	FALSE	FALSE	FALSE	FALSE	FALSE

Table A.17: *Overall similarity, a large margin classifier.*

VAR-LiNGAM		Correct order	Detected recursiveness
Big diagonal	Lower triangular	0.920	0.988
	Other recursive structure	0.920	0.988
	Not recursive	0.176	0.054
Small diagonal	Lower triangular	0.980	1.000
	Other recursive structure	0.976	1.000
	Not recursive	0.162	0.320

Table A.18: *Order of the variables and recursiveness of the mixing matrix as estimated by VAR-LiNGAM. Monte Carlo average over 500 simulations.*

# Appendix B. Performance of the NGSi and related algorithms

Number of variables	Number of lags	Big Diagonal			Small Diagonal		
		Lower triangular	Other recursive structure	Not recursive	Lower triangular	Other recursive structure	Not recursive
2	1	0.98	0.91	0.86	1.00	0.97	0.71
	2	0.98	0.89	0.86	1.00	0.97	0.71
	3	0.98	0.90	0.87	0.99	0.97	0.71
3	1	0.94	0.93	0.95	0.99	0.98	0.18
	2	0.95	0.94	0.96	0.96	0.97	0.20
	3	0.96	0.94	0.96	0.94	0.96	0.19
4	1	0.93	0.94	0.90	0.87	0.87	0.01
	2	0.92	0.90	0.89	0.88	0.90	0.02
	3	0.93	0.95	0.88	0.88	0.87	0.02
5	1	0.84	0.86	0.82	0.78	0.71	0.00
	2	0.91	0.89	0.81	0.77	0.73	0.00
	3	0.85	0.85	0.82	0.80	0.71	0.00
6	1	0.81	0.74	0.55	0.74	0.61	0.00
	2	0.80	0.74	0.54	0.71	0.58	0.00
	3	0.85	0.77	0.57	0.69	0.57	0.00

Table B.19: *Performance of the benchmark algorithm, as measured by the “correctness” distance measure. Monte Carlo average over 500 simulations.*

Number of variables	Number of lags	Big Diagonal			Small Diagonal		
		Lower triangular	Other recursive structure	Not recursive	Lower triangular	Other recursive structure	Not recursive
2	1	1.00	0.44	0.94	1.00	0.19	0.77
	2	1.00	0.45	0.93	1.00	0.17	0.77
	3	1.00	0.47	0.94	1.00	0.19	0.77
3	1	1.00	0.67	0.95	0.99	0.96	0.18
	2	1.00	0.70	0.96	0.96	0.95	0.20
	3	1.00	0.71	0.96	0.95	0.93	0.19
4	1	1.00	0.87	0.90	0.87	0.87	0.01
	2	1.00	0.84	0.89	0.88	0.90	0.02
	3	1.00	0.87	0.88	0.88	0.87	0.02
5	1	1.00	0.86	0.82	0.78	0.71	0.00
	2	1.00	0.88	0.81	0.77	0.73	0.00
	3	1.00	0.84	0.82	0.80	0.71	0.00
6	1	1.00	0.74	0.55	0.74	0.61	0.00
	2	1.00	0.74	0.55	0.71	0.58	0.00
	3	1.00	0.77	0.57	0.69	0.57	0.00

Table B.20: *Performance of the refined algorithm, as measured by the “correctness” distance measure. Monte Carlo average over 500 simulations.*

Number of variables	Number of lags	Big Diagonal			Small Diagonal		
		Lower triangular	Other recursive structure	Not recursive	Lower triangular	Other recursive structure	Not recursive
2	1	0.98	0.91	0.86	1.00	0.97	0.71
	2	0.98	0.89	0.86	1.00	0.97	0.71
	3	0.98	0.90	0.87	0.99	0.97	0.71
3	1	0.94	0.93	0.95	0.99	0.98	0.18
	2	0.95	0.94	0.96	0.96	0.97	0.20
	3	0.96	0.94	0.96	0.94	0.96	0.19
4	1	0.93	0.94	0.90	0.87	0.87	0.01
	2	0.92	0.90	0.89	0.88	0.90	0.02
	3	0.93	0.95	0.88	0.88	0.87	0.02
5	1	1.00	0.86	0.82	0.78	0.71	0.00
	2	1.00	0.88	0.81	0.77	0.73	0.00
	3	1.00	0.84	0.82	0.80	0.71	0.00
6	1	1.00	0.74	0.55	0.74	0.61	0.00
	2	1.00	0.74	0.55	0.71	0.58	0.00
	3	1.00	0.77	0.57	0.69	0.57	0.00

Table B.21: *Performance of the NGS algorithm, as measured by the “correctness” distance measure. Monte Carlo average over 500 simulations.*

Number of variables	Number of lags	Big Diagonal			Small Diagonal		
		Lower triangular	Other recursive structure	Not recursive	Lower triangular	Other recursive structure	Not recursive
2	1	0.00	0.00	0.00	0.00	0.00	0.29
	2	0.00	0.00	0.00	0.00	0.00	0.29
	3	0.00	0.00	0.00	0.00	0.00	0.30
3	1	0.00	0.00	0.02	0.00	0.00	0.62
	2	0.00	0.00	0.02	0.00	0.00	0.64
	3	0.00	0.00	0.01	0.00	0.00	0.65
4	1	0.00	0.00	0.05	0.00	0.00	0.67
	2	0.00	0.00	0.06	0.00	0.00	0.67
	3	0.00	0.00	0.06	0.00	0.00	0.64
5	1	0.00	0.00	0.02	0.00	0.00	0.61
	2	0.00	0.00	0.02	0.00	0.00	0.60
	3	0.00	0.00	0.03	0.00	0.00	0.56
6	1	0.00	0.00	0.08	0.00	0.00	0.49
	2	0.00	0.00	0.05	0.00	0.00	0.45
	3	0.00	0.00	0.05	0.00	0.00	0.44

Table B.22: *Performance of the refined and NGS algorithms, percentage of detections of structure  $f$ ). Monte Carlo average over 500 simulations.*

Number of variables	Number of lags	Big Diagonal			Small Diagonal		
		Lower triangular	Other recursive structure	Not recursive	Lower triangular	Other recursive structure	Not recursive
2	1	0.53	0.18	0.66	0.66	0.52	0.68
	2	0.57	0.22	0.67	0.70	0.53	0.70
	3	0.48	0.23	0.71	0.69	0.56	0.72
3	1	0.02	0.01	0.26	0.29	0.32	0.14
	2	0.01	0.00	0.22	0.40	0.39	0.15
	3	0.02	0.01	0.23	0.34	0.34	0.20
4	1	0.00	0.00	0.01	0.23	0.24	0.02
	2	0.00	0.00	0.01	0.21	0.21	0.00
	3	0.00	0.00	0.03	0.20	0.23	0.01
5	1	0.00	0.00	0.00	0.16	0.17	0.00
	2	0.00	0.00	0.01	0.15	0.16	0.00
	3	0.00	0.00	0.00	0.15	0.17	0.00
6	1	0.00	0.00	0.00	0.05	0.05	0.00
	2	0.00	0.00	0.00	0.04	0.07	0.00
	3	0.00	0.00	0.00	0.04	0.06	0.00

Table B.23: *Performance of the NGS algorithm, Gaussian residuals. Monte Carlo average over 500 simulations.*

## Appendix C. NGSi algorithm: an R implementation

```
## EXAMPLE 1 - TYPE A)
## === True mixing matrix ===
##           [,1]      [,2]      [,3]      [,4] [,5]
## [1,]  1.00000000  0.00000000  0.00000000  0.00000000  0
## [2,]  0.27186684  1.00000000  0.00000000  0.00000000  0
## [3,]  0.09850571  0.58029102  1.00000000  0.00000000  0
## [4,] -0.44593505 -0.05565642 -0.4703982  1.00000000  0
## [5,]  0.59624841  0.08698521 -0.1395924 -0.4331102  1
##
## === Estimation with NGSi ===
## [1] "Testing Gaussianity of residuals:"
## [1] "0 Gaussian residuals found, no problems for ICA..."
## [1] "Mixing matrix likely recursive --> checking number of variables:"
## [1] "There are more than 4 variables --> checking comparisons:"
## [1] "Chol == MaxDiag found, case a) is likely --> using unordered Cholesky."
##           y1          y2          y3          y4 y5
## y1  1.00000000  0.00000000  0.00000000  0.00000000  0
## y2  0.24901617  1.00000000  0.00000000  0.00000000  0
## y3  0.09279908  0.60063423  1.00000000  0.00000000  0
## y4 -0.42199045 -0.04765287 -0.4877141  1.00000000  0
## y5  0.58009640  0.01004399 -0.1765716 -0.3711138  1
##
```

Figure C.8: R implementation of the NGSi algorithm: lower triangular mixing matrix with “big diagonal.”

```
## EXAMPLE 2 - TYPE B)
## === True mixing matrix ===
##           [,1]      [,2]      [,3]      [,4] [,5]
## [1,]  0.09850571  0.58029102  1.00000000  0.00000000  0
## [2,]  0.27186684  1.00000000  0.00000000  0.00000000  0
## [3,]  1.00000000  0.00000000  0.00000000  0.00000000  0
## [4,] -0.44593505 -0.05565642 -0.4703982  1.00000000  0
## [5,]  0.59624841  0.08698521 -0.1395924 -0.4331102  1
##
## === Estimation with NGSi ===
## [1] "Testing Gaussianity of residuals:"
## [1] "0 Gaussian residuals found, no problems for ICA..."
## [1] "Mixing matrix likely recursive --> checking number of variables:"
## [1] "There are more than 4 variables --> checking comparisons:"
## [1] "No useful comparisons found but big diagonal likely --> using refined ordered Cholesky."
##           y3          y2          y1          y4 y5
## [1,]  0.09068958  0.60283860  1.00000000  0.0000000  0
## [2,]  0.25119419  1.00000000  0.00000000  0.0000000  0
## [3,]  1.00000000  0.00000000  0.00000000  0.0000000  0
## [4,] -0.41646290 -0.05040124 -0.4895949  1.0000000  0
## [5,]  0.57622755  0.01164346 -0.1767446 -0.368009  1
##
```

Figure C.9: R implementation of the NGSi algorithm: recursive mixing matrix with “big diagonal.”

```

## EXAMPLE 3 - TYPE C)
## === True mixing matrix ===
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,]  1.00000000  0.77176623  0.07762953  0.3602777  0.2859462
## [2,]  0.27186684  1.00000000 -0.38480943  0.4854660 -0.1986105
## [3,]  0.09850571  0.58029102  1.00000000 -0.2793930  0.5950150
## [4,] -0.44593505 -0.05565642 -0.47039820  1.0000000 -0.6172858
## [5,]  0.59624841  0.08698521 -0.13959241 -0.4331102  1.0000000
##
## === Estimation with NGSI ===
## [1] "Testing Gaussianity of residuals:"
## [1] "1 Gaussian residuals found, no problems for ICA..."
## [1] "Mixing matrix likely non recursive --> using MaxDiag algorithm"
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,]  1.00000000  0.75118577  0.1093223  0.3588811  0.3176370
## [2,]  0.25908890  1.00000000 -0.3665201  0.4975361 -0.1431254
## [3,]  0.07378982  0.54700177  1.0000000 -0.3162895  0.5314783
## [4,] -0.43634479 -0.01130275 -0.4617429  1.0000000 -0.5187679
## [5,]  0.60254341  0.01660923 -0.1256544 -0.4359643  1.0000000
##

```

Figure C.10: *R* implementation of the NGSI algorithm: non-recursive mixing matrix with “big diagonal.”

```

## EXAMPLE 4 - TYPE D)
## === True mixing matrix ===
##           [,1]      [,2]      [,3]      [,4] [,5]
## [1,]  1.0000000  0.0000000  0.0000000  0.000000  0
## [2,] -0.6422172  1.0000000  0.0000000  0.000000  0
## [3,] -0.2326950 -0.89916917  1.0000000  0.000000  0
## [4,]  1.0534097  0.08624042 -1.2251961  1.000000  0
## [5,] -1.4084873 -0.13478482 -0.3635815  7.700936  1
##
## === Estimation with NGSI ===
## [1] "Testing Gaussianity of residuals:"
## [1] "1 Gaussian residuals found, no problems for ICA..."
## [1] "Mixing matrix likely recursive --> checking number of variables:"
## [1] "There are more than 4 variables --> checking comparisons:"
## [1] "No useful comparisons found but big diagonal not likely --> using ordered Cholesky."
##           y1      y2      y3      y4 y5
## [1,]  1.0000000  0.0000000  0.0000000  0.000000  0
## [2,] -0.6625046  1.0000000  0.0000000  0.000000  0
## [3,] -0.2049223 -0.88611746  1.0000000  0.000000  0
## [4,]  1.0687459  0.08258909 -1.2460191  1.000000  0
## [5,] -1.1877423 -0.10845868 -0.5639603  7.765881  1
##

```

Figure C.11: *R* implementation of the NGSI algorithm: lower triangular mixing matrix with “small diagonal.”

```

## EXAMPLE 5 - TYPE E)
## === True mixing matrix ===
##           [,1]      [,2]      [,3]      [,4] [,5]
## [1,]  1.0000000  0.0000000  0.0000000  0.000000  0
## [2,] -0.6422172  1.0000000  0.0000000  0.000000  0
## [3,] -1.4084873 -0.13478482 -0.3635815  7.700936  1
## [4,]  1.0534097  0.08624042 -1.2251961  1.000000  0
## [5,] -0.2326950 -0.89916917  1.0000000  0.000000  0
##
## === Estimation with NGSi ===
## [1] "Testing Gaussianity of residuals:"
## [1] "1 Gaussian residuals found, no problems for ICA..."
## [1] "Mixing matrix likely recursive --> checking number of variables:"
## [1] "There are more than 4 variables --> checking comparisons:"
## [1] "No useful comparisons found but big diagonal not likely --> using ordered Cholesky."
##           y1          y2          y5          y4 y3
## [1,]  1.0000000  0.0000000  0.0000000  0.000000  0
## [2,] -0.6642176  1.0000000  0.0000000  0.000000  0
## [3,] -1.1953500 -0.11308453 -0.5503931  7.761787  1
## [4,]  1.0654984  0.08215316 -1.2441912  1.000000  0
## [5,] -0.2016365 -0.88618840  1.0000000  0.000000  0
##

```

Figure C.12: *R* implementation of the NGSi algorithm: recursive mixing matrix with “small diagonal.”

```

## EXAMPLE 6 - TYPE F)
## === True mixing matrix ===
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,]  1.0000000 -1.19586273  0.2021934 -6.405934 -1.0600481
## [2,] -0.6422172  1.0000000 -1.0022722 -8.631851  0.7362805
## [3,] -0.2326950 -0.89916917  1.0000000  4.967761 -2.2058151
## [4,]  1.0534097  0.08624042 -1.2251961  1.000000  2.2883763
## [5,] -1.4084873 -0.13478482 -0.3635815  7.700936  1.0000000
##
## === Estimation with NGSi ===
## [1] "Testing Gaussianity of residuals:"
## [1] "0 Gaussian residuals found, no problems for ICA..."
## [1] "Mixing matrix likely non recursive --> using MaxDiag algorithm"
## [1] "Warning: likely to be in case f, for which can't correctly identify mixing matrix!"
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,]  1.0000000  0.7371499  0.09889269 -0.65639420 -0.7212154
## [2,] -0.60187584  1.0000000 -1.09691989  0.05034639  0.5793139
## [3,]  0.50523648 -0.5772221  1.0000000 -0.77322038  0.1491046
## [4,]  0.04950718 -0.1164692 -1.09462795  1.00000000 -0.8573872
## [5,]  0.03892688 -0.8878531 -0.23504708  0.67797082  1.0000000
##

```

Figure C.13: *R* implementation of the NGSi algorithm: non-recursive mixing matrix with “small diagonal.”

## Appendix D. “K+S” model time series

The column tcode denotes the following data transformation for a series  $x$ : (1) no transformation; (2)  $\Delta x_t$ ; (3)  $\Delta^2 x_t$ ; (4)  $\log(x_t)$ ; (5)  $\Delta \log(x_t)$ ; (6)  $\Delta^2 \log(x_t)$ ; (7)  $\Delta(x_t/x_{t-1} - 1)$ .

tcode	variable name	description <sup>a</sup>	
1	5	GDP	Real gross domestic product
2	5	Creal	Real aggregate consumption
3	5	Ir	Real aggregate investments
4	2	dNtot	Changes in inventories
5	5	Eltot	Total expansionary investments, both sectors
6	5	Sltot	Total substitution investments
7	1	LD	Percentage of labour demand
8	1	U	Unemployment rate
9	5	w	Total wages
10	1	diff_w	Total wages (first differences)
11	5	cpi	Consumer price index
12	1	diff_cpi	Consumer price index (first differences)
13	5	Am	Average productivity over two periods
14	5	rw	Real wages
15	2	Mutot	Global mark-up
16	5	G	Public expenditures
17	5	Tax	Total tax revenues
18	2	Def	Public deficit
19	5	Deb	Public debt
20	1	DefonGDP	Public deficit to GDP ratio
21	2	DebonGDP	Public debt to GDP ratio
22	1	H1	Herfindal index, sector 1
23	1	H2	Herfindal index, sector 2
24	1	DF1	Variability of market shares index, sector 1
25	1	DF2	Variability of market shares index, sector 2
26	1	next1	Number of exits, sector 1
27	1	next2	Number of exits, sector 2
28	1	next2bc	Number of exits due to bankruptcy, sector 2
29	4	Debt_all	Private sector debt
30	5	BankEquity_all	Total bank equity
31	4	BankProfits_all	Bank profits
32	4	BadDebt_all	Non performing loans
33	5	CreditSupply_all	Total credit supply
34	4	CreditDemand_all	Total credit demand
35	1	HB	Herfindal index, banking sector
36	1	countbf_all2	Number of banking failures per time period
37	5	BankCash_all	Bank reserves
38	2	Nb_act	Total number of banking failures
39	1	mean_rdeb_all	Average interest rate on loans
40	4	Gbailout_all	Fiscal costs of bailouts
41	1	d_cpi	Inflation rate
42	1	r	Base interest rate
43	1	count_zerobound	1 if zero bound constraint
44	1	count_savings	1 if private savings < deficit
45	1	count_bonds	1 if demand for bonds does not cover deficit
46	1	count_inflation_target	1 if inflation $\neq$ inflation target
47	1	count_unemp_target	1 if $U(1) \leq$ target unemployment
48	1	Loan_profit_share	Share of loan interest over total interest revenues
49	1	r_bonds	Interest rate on bonds
50	2	share_FC_prod	Share of constrained firm production
51	2	share_FC_inv	Share of constrained firm investment
52	2	FC_Prod_tot	Ratio of production constraint
53	2	FC_Inv_tot	Ratio of investment constraint
54	1	bankr_LDtot	Share of employment of exiting firms
55	1	count_def_rec2	1 if fiscal rule is not binding due to recession
56	1	count_def2	1 if fiscal rule is binding
57	1	mean_rbank_all	Average prime rate of banks
58	1	std_rdeb_all	Standard loan rate
59	1	count_compact2	1 if fiscal compact is binding
60	5	GDPm	Nominal GDP

Table D.24: *Variables obtained as realization of the “K+S” model.*

<sup>a</sup>For further details refer to the original paper: Dosi et al. (2015).

## Appendix E. Parametrization of the “K+S” model

Description	Symbol	Value
Monte Carlo replications	$M$	100
Time sample	$T$	600
Number of firms in capital-good industry	$F_1$	50
Number of firms in consumption-good industry	$F_2$	200
Number of banks	$B$	10
Capital-good firms’ mark-up	$\mu_1$	0,04
Consumption-good firm initial mark-up	$\bar{\mu}_0$	0,25
Uniform distribution supports	$[\varphi_1, \varphi_2]$	[0,10, 0,90]
Wage setting $\Delta \bar{A}B$ weight	$\psi_1$	1
Wage setting $\Delta cpi$ weight	$\psi_2$	0,05
Wage setting $\Delta U$ weight	$\psi_3$	0,05
Bank deposits interest rate	$r^d$	0
Bond interest rate mark-up	$\mu^{bonds}$	-0,33
Loan interest rate mark-up	$\mu^{debt}$	0,3
Bank capital adequacy rate	$\tau^b$	0,08
Shape parameter of bank client distribution	$pareto_\alpha$	0,08
Scaling parameter for interest rate cost	$k_{const}$	0,1
Capital buffer adjustment parameter	$\beta$	1
RD investment propensity	$\nu$	0,04
RD allocation to innovative search	$\xi$	0,5
Firm search capabilities parameters	$\zeta_{1,2}$	0,3
Beta distribution parameters (innovation)	$(\alpha_1, \beta_1)$	(3, 3)
Beta distribution support (innovation)	$[\chi_1, \bar{\chi}_1]$	[-0,15, 0,15]
New customer sample parameter	$\bar{\omega}$	0,5
Desired inventories	$l$	0,1
“Physical” scrapping age	$\eta$	20
Payback period	$b$	3
Competitiveness weights	$\omega_{1,2}$	1
Coefficient in the consumption-good firm mark-up rule	$\upsilon$	0,04
Tax rate	$tr$	0,1
Unemployment subsidy rate	$\varphi$	0,6
Baseline interest rate	$r$	0,025

Table E.25: *Parametrization of the model under validation. The unique difference across the 100 Monte Carlo replications is the random seed.*

## Appendix F. Factor analysis: tables and figures

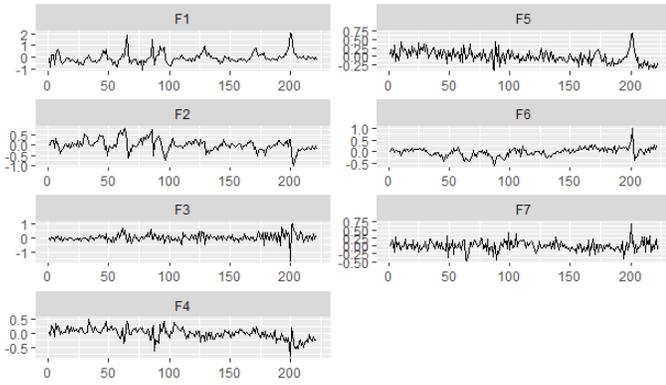
	Factor 1	Factor 2	Factor 3	Factor 4			
<b>mR2k1</b>	<b>0.198</b>	<b>mR2k2</b>	<b>0.076</b>	<b>mR2k3</b>	<b>0.067</b>	<b>mR2k4</b>	<b>0.041</b>
USPRIV	0.864	AAAFM	0.552	CUSR0000SA0L2	0.741	CESS9093000001	0.271
USGOOD	0.853	T5YFFM	0.492	PCECTPI	0.731	TWEXM/TH	0.264
PAYEMS	0.837	TCU	0.427	DGDSRG3Q086SBEA	0.722	CESS9092000001	0.250
MANEMP	0.797	BUSINVx	0.423	CUSR0000SAC	0.709	TLBSNNBBDIx	0.209
HOANBS	0.797	GS10TB3Mx	0.422	CUSR0000SA0L5	0.698	SLCEx	0.192
IPMANSICS	0.789	PERMIT	0.419	CP1AUCSL	0.694	NWP1x	0.187
UNRATE	0.788	HOUST	0.401	DNDGRG3Q086SBEA	0.670	GFDEBTNx	0.184
INDPRO	0.783	CUMFNS	0.390	CP1ULFSL	0.641	USGOVT	0.179
DMANEMP	0.782	PERMITS	0.366	DGOERG3Q086SBEA	0.570	USFIRE	0.177
LNS14000025	0.776	CPF3MTTB3Mx	0.362	WPSFD49207	0.543	PCEsvx	0.162
<b>Factor 5</b>	<b>Factor 6</b>	<b>Factor 7</b>					
<b>mR2k5</b>	<b>0.033</b>	<b>mR2k6</b>	<b>0.030</b>	<b>mR2k7</b>	<b>0.025</b>		
OPHPBS	0.244	CONSP1x	0.390	UNLPNBS	0.251		
OPHNFB	0.212	NONREVS1x	0.308	PCEPILFE	0.207		
TNWBSNNBx	0.204	TOTALSLx	0.254	MIREAL	0.186		
TNWBSHNOx	0.204	NWPIx	0.250	OPHPBS	0.173		
TABSHNOx	0.192	CONSUMERx	0.206	OPHNFB	0.147		
TFAABSHNOx	0.171	AHETPIx	0.196	ULCBS	0.145		
TARESAsx	0.169	AWHMAN	0.182	ULCMFG	0.140		
HOAMS	0.167	OUTMS	0.177	ULCNFB	0.135		
USGOVT	0.164	REALLNx	0.148	GFDEGDQ188S	0.133		
TABSNNBx	0.134	ULCNFB	0.144	COMPRMS	0.114		

Table F.26: Factors estimated from the real-world data: list of the 10 series that load most heavily on all each factor along with the  $R^2$  obtained from a regression of the series on the factor.

	Factor 1	Factor 2	Factor 3	Factor 4			
<b>mR2k1</b>	<b>0.165</b>	<b>mR2k2</b>	<b>0.156</b>	<b>mR2k3</b>	<b>0.118</b>	<b>mR2k4</b>	<b>0.072</b>
cpi	0.906	Deb	0.734	next2	0.703	BankEquity-all	0.442
diff_cpi	0.905	GDPm	0.578	LD	0.429	std_rdeb_all	0.373
d_cpi	0.904	GDP	0.488	U	0.429	mean_rdeb_all	0.344
r	0.784	DefonGDP	0.480	bankr_LDtot	0.385	mean_rbank-all	0.341
r_bonds	0.761	std_rdeb_all	0.403	DF2	0.384	countbf_all2	0.294
diff_w	0.672	U	0.358	next2bc	0.373	next2bc	0.265
w	0.672	LD	0.358	DefonGDP	0.328	Gbailout_all	0.192
count_zerobound	0.512	mean_rdeb_all	0.322	countbf_all2	0.307	DebonGDP	0.185
Tax	0.312	Creal	0.320	H2	0.288	BankProfits_all	0.164
Am	0.304	mean_rbank_all	0.319	Gbailout_all	0.271	BankCash_all	0.152
<b>Factor 5</b>		<b>Factor 6</b>		<b>Factor 7</b>			
<b>mR2k5</b>	<b>0.062</b>	<b>mR2k6</b>	<b>0.042</b>	<b>mR2k7</b>	<b>0.036</b>		
Ir	0.633	BadDebt_all	0.343	next1	0.233		
Slftot	0.520	Gbailout_all	0.210	Eltot	0.186		
G	0.430	next1	0.204	dNtot	0.168		
Creal	0.321	CreditDemand_all	0.144	FC_Inv_tot	0.140		
Am	0.314	H2	0.140	H1	0.114		
FC_Inv_tot	0.161	H1	0.139	DebonGDP	0.108		
dNtot	0.105	count_savings	0.128	FC_Prod_tot	0.088		
rw	0.105	Debt_all	0.111	Creal	0.076		
diff_w	0.065	countbf_all2	0.106	share_FC_inv	0.075		
w	0.065	BankProfits_all	0.102	GDPm	0.073		

Table F.27: Typical factors estimated from the agent-based data (first Monte Carlo realization): list of the 10 series that load most heavily on all each factor along with the  $R^2$  obtained from a regression of the series on the factor.

(a) Real-world factors



(b) Typical agent-based factors (first Monte Carlo realization)

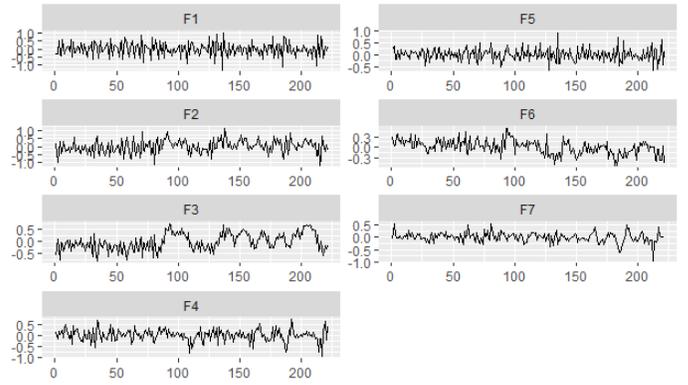
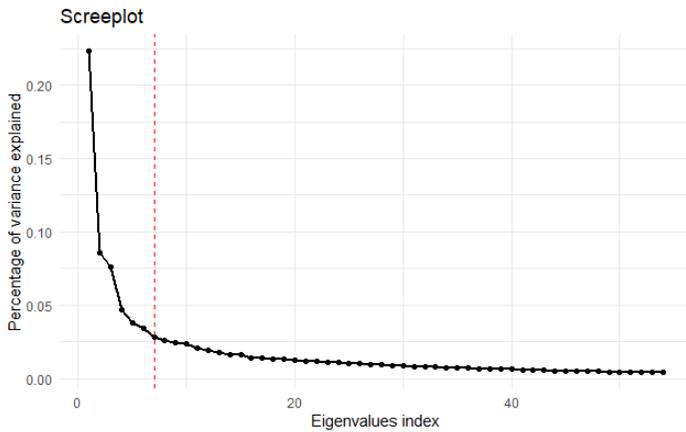


Figure F.14: Plot of the first 7 factors extracted from the datasets.

(a) Real-world factors



(b) Typical agent-based factors (first Monte Carlo realization)

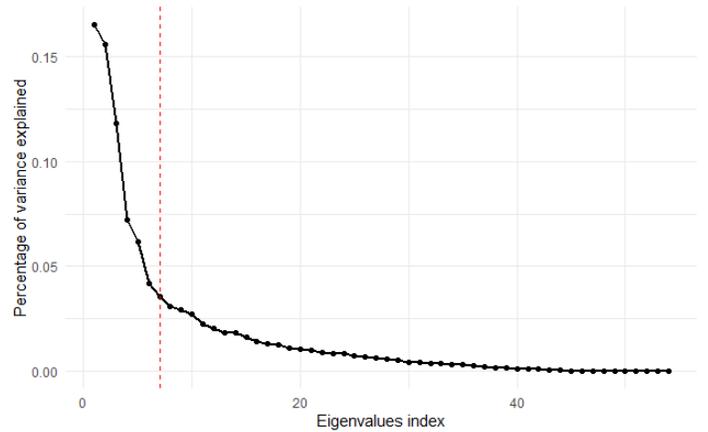


Figure F.15: Portion of data variability accounted by each factor. The dashed red line indicates the seventh factor.

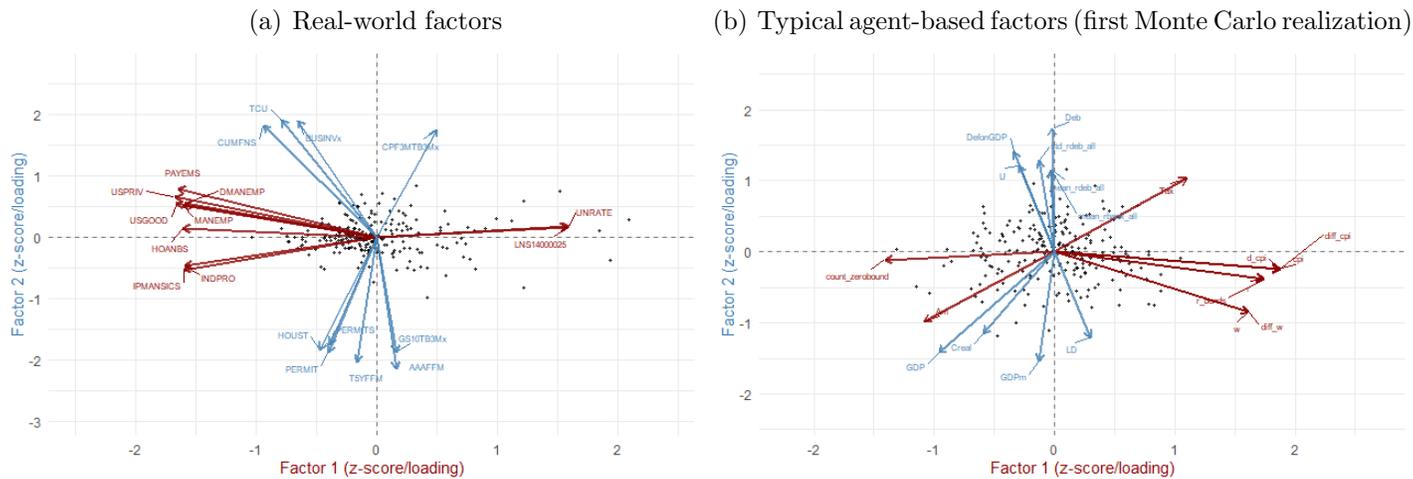


Figure F.16: *Biplot of the first two factors. For each factor the ten variables with the highest loadings are shown.*

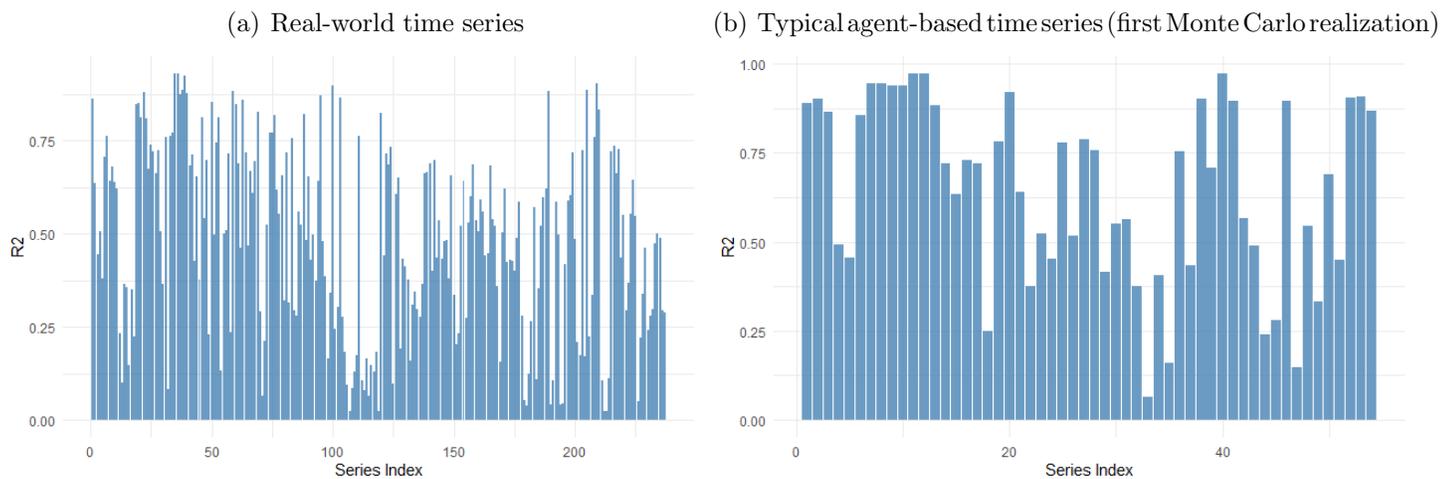


Figure F.17: *Explanatory power of the first seven factors in the time series, expressed as the  $R^2$  of the regression of each variable on the seven factors.*

# Appendix G. Impulse Response Functions

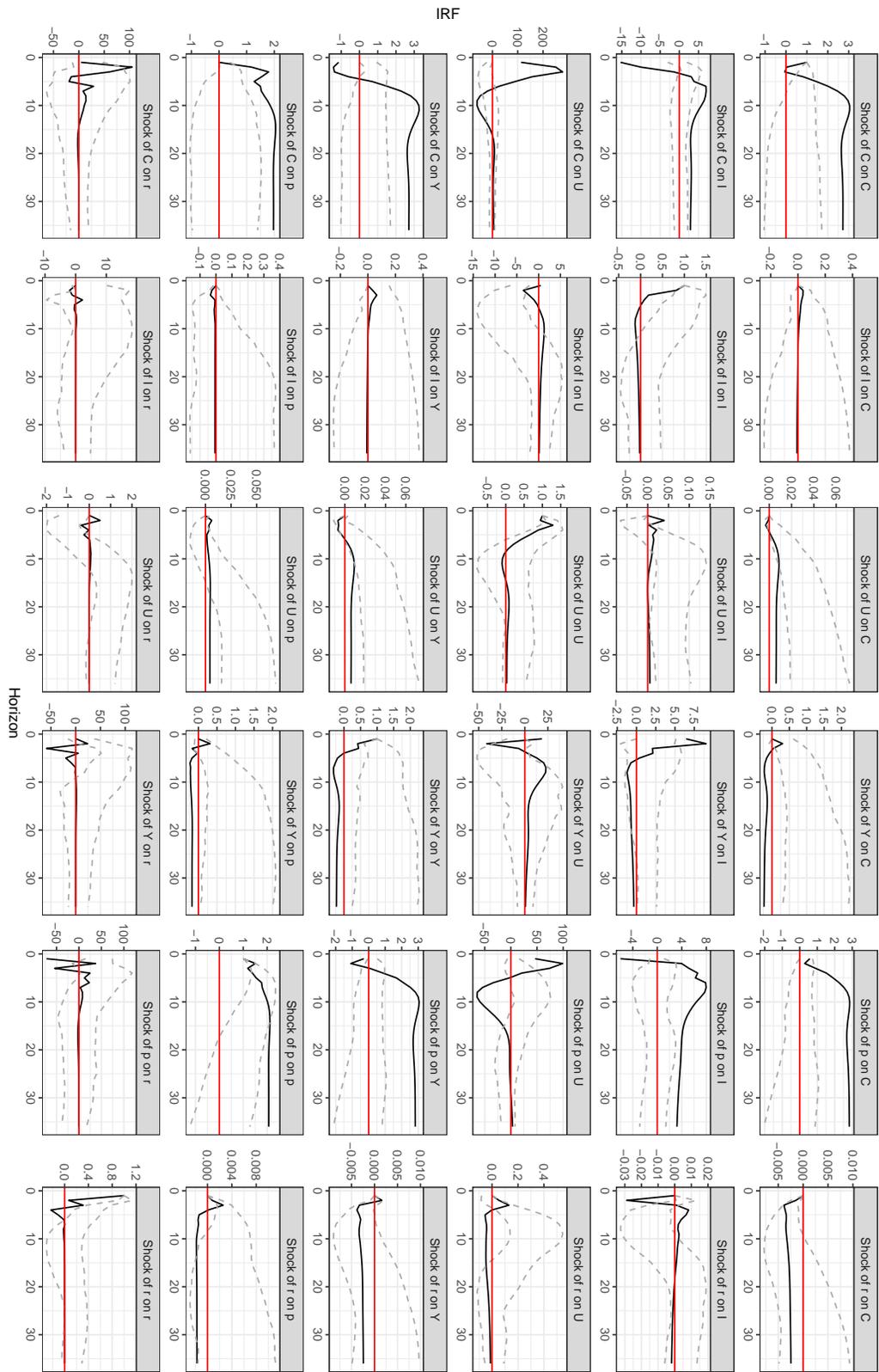


Figure G.18: Structural VAR validation: agent-based IRFs (solid line) and real-world IRF confidence intervals (dashed lines) based on 1000 bootstrap iterations.

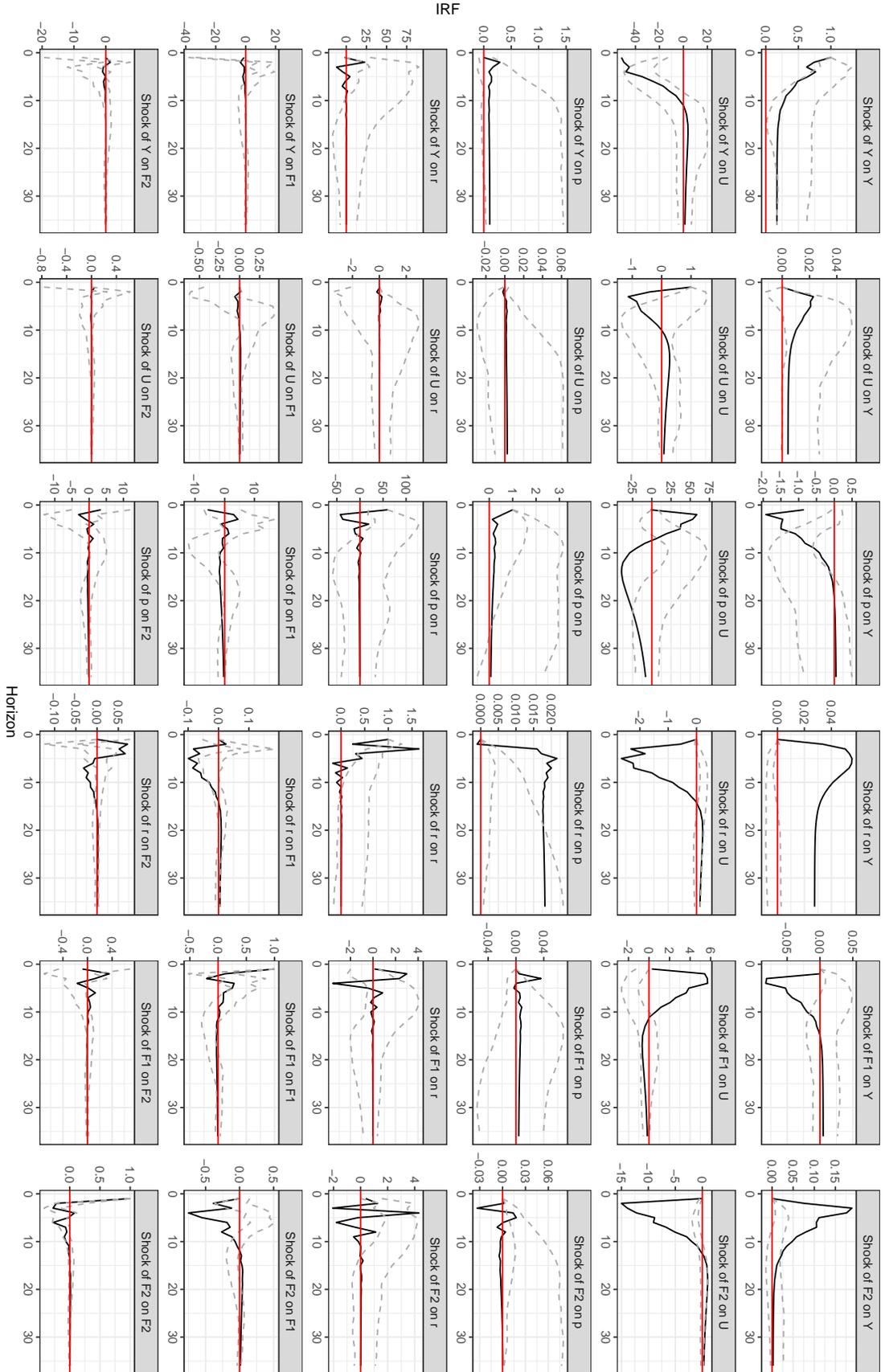
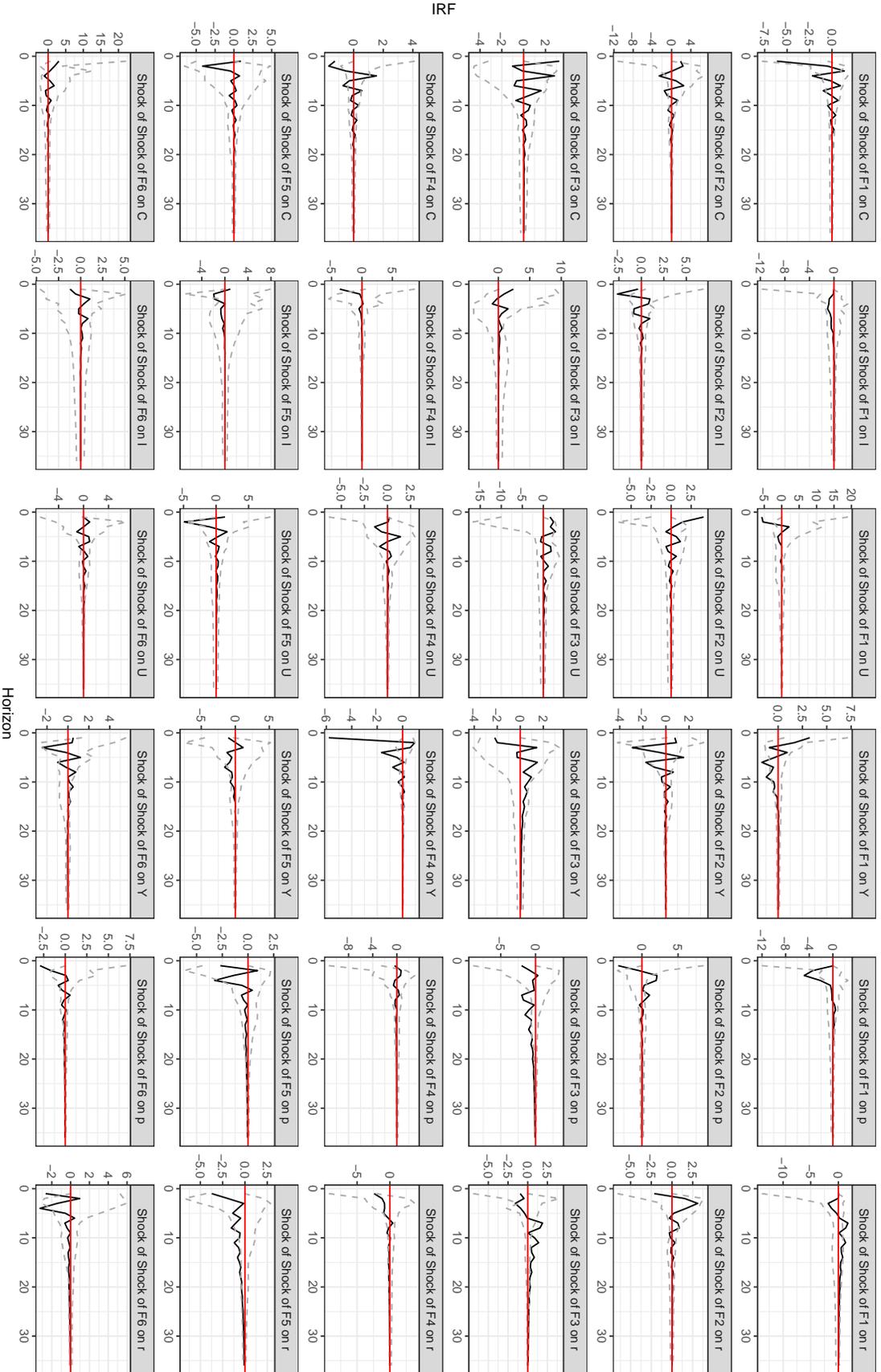


Figure G.19: Structural FAVAR validation: agent-based IRFs (solid line) and real-world IRF confidence intervals (dashed lines) based on 1000 bootstrap iterations.



IRF

Figure G.20: Structural DFM validation: agent-based IRFs (solid line) and real-world IRF confidence intervals (dashed lines) based on 1000 bootstrap iterations.