UNIVERSITA' DEGLI STUDI DI PADOVA

DIPARTIMENTO DI SCIENZE ECONOMICHE ED AZIENDALI "M. FANNO"

CORSO DI LAUREA MAGISTRALE IN
ECONOMICS AND FINANCE

TESI DI LAUREA

"IF THE FED IS EXCITED, DOES THE EURO GET HOT?"

RELATORE:

CH.MO PROF. EFREM CASTELNUOVO

LAUREANDO: Francescomaria SIMEONE

MATRICOLA N. 2058910

ANNO ACCADEMICO

2022 – 2023

Dichiaro di aver preso visione del "Regolamento antiplagio" approvato dal Consiglio del Dipartimento di Scienze Economiche e Aziendali e, consapevole delle conseguenze derivanti da dichiarazioni mendaci, dichiaro che il presente lavoro non è già stato sottoposto, in tutto o in parte, per il conseguimento di un titolo accademico in altre Università italiane o straniere. Dichiaro inoltre che tutte le fonti utilizzate per la realizzazione del presente lavoro, inclusi i materiali digitali, sono state correttamente citate nel corpo del testo e nella sezione 'Riferimenti bibliografici'.

I hereby declare that I have read and understood the "Anti-plagiarism rules and regulations" approved by the Council of the Department of Economics and Management and I am aware of the consequences of making false statements. I declare that this piece of work has not been previously submitted – either fully or partially – for fulfilling the requirements of an academic degree, whether in Italy or abroad. Furthermore, I declare that the references used for this work – including the digital materials – have been appropriately cited and acknowledged in the text and in the section 'References'.

Firma (signature) ……………………

Table of Contents

Abstract

The primary objective of this thesis is to explore the distinct effects of monetary policy shocks and information shocks and to determine whether information shocks, followed by United States Federal Open Market Committee (FOMC) communications, can have significant impacts on key macroeconomic variables within European Union markets.

This research entails a comprehensive analysis of the impact of these shocks on critical economic indicators. To achieve this, a VAR model is utilized, employing a recursive VAR framework used as a proxy for a structural VAR model. The analysis of results and the exploration of key research questions are conducted through the use of impulse response functions and forecast error variance decompositions.

The dataset covers the period from 1991 to 2015, containing monthly data, and includes a range of economic indicators, such as inflation, unemployment rates, short and long-term interest rates, as well as variables used as a proxy for the global financial cycle and financial distress indicators.

The VAR model includes seven variables and incorporates six lags to better understand the different impacts of monetary and information shocks and their implications for European macroeconomic variables.

Introduction

In an increasingly interconnected global economy, the transmission of financial and economic information has become more rapid and influential than ever before. Financial markets around the world are intrinsically linked, and any event or piece of information originating in one part of the world can have a profound impact on markets elsewhere. This dynamic is particularly evident when considering the European markets and their relationship with the United States, a financial powerhouse that serves as a primary driver of global economic sentiment and information flow.

This study aims to explore the connection between the United States and European financial markets, with a special focus on the effects of "surprising information" issued by the United States Federal Open Market Committee (FOMC).

To get started, the work introduces and explain two distinct types of effects that occur when FOMC releases such information. The first is the standard "monetary policy" (MPI) impact, and the second is the "information shock" (INFO) impact. It's important to understand that they have completely different effects on financial markets.

Starting with the research of Miranda - Agrippino and Ricco (2021) as well as Jarocinski and Karadi (2020), the aim is to deliver a comprehensive explanation of the development of these two impacts.

Subsequently, building upon this initial description, my analysis will delve into the effects of these shocks on key macroeconomic variables in the European markets. In order to conduct this analysis, I will examine impulse response functions (IRFs) and forecast error variance decomposition (FEVD) calculated using a VAR model, specifically, a recursive VAR model, which serves as a proxy for the Structural VAR. To validate the results, a variety of robustness tests will be employed, accounting for different variables and proxies, to assess the presence or absence of significant impacts (Møller and Wolf, 2021).

The dataset used includes both EU and US macroeconomic variables such as short and long-term interest rates, inflation, unemployment, industrial production, and exchange rates. Additionally, an indicator of the global financial cycle (GFC) and a proxy for a financial distress indicator (Baa10Y) will also be incorporated.

Ultimately, once the significance of the data sets has been rigorously tested and the different variables have been utilized to demonstrate the model's capacity to capture the impulse response of these shocks, a concluding analysis will be provided. This final commentary will endeavor to compare the results with conventional economic theory and offer thoughtful insights.

Chapter 1

1.1 Introduction to information shocks.

First of all, an “Information shock” (INFO) occurs when central banks release announcements with new and unexpected insights into the economic outlook. These announcements can affect financial markets by altering how investors see and expect future economic conditions. Unlike the more common “monetary policy shocks” (MPI), which are recognized when interest rates and stock prices move in opposite directions, “INFO shocks” are characterized by interest rates and stock prices moving together.

During the late 1990s, these INFO shocks were not well-documented, puzzling macroeconomic researchers. For example, on March 20, 2001, the FOMC surprised the market by making a bigger than anticipated 50-basis-point cut in the federal funds rate. Conventional economic theory would predict that the S&P 500 stock market index should have gone up, but instead, it experienced a significant drop within just 30 minutes of the announcement. This kind of event is not unique; around one-third of FOMC announcements since 1990 have shown this unusual pattern of interest rates and stock market changes moving together. The main explanation for this deviation from standard economic theory is that the FOMC, in its communication, highlighted "substantial risks that demand and production could remain soft" in the foreseeable future. This pessimistic communication led to a decrease in stock values, regardless of the surprise rate cut (Jarocinski and Karadi, 2020).

Miranda, Agrippino and Ricco (2021) have created a method to measure these information shocks and convert them into numerical data based on FOMC announcements over time. This allows them to distinguish between changes caused by policy decisions and those brought about by central bank information, helping them assess how these shocks impact asset prices and the broader economy. This method enables to use market prices to understand the hidden messages in central bank announcements, which are otherwise challenging for econometricians to uncover.

More specifically, the approach used by Miranda, Agrippino, and Ricco (2021) aims to determine how policy changes and central bank information influence the economy over time using a Bayesian structural vector autoregression (VAR).

In their basic VAR model, using U.S. data, they add information from high-frequency financial market surprises that occur during monetary policy announcements to variables such as interest rates, the price level, economic activity, and financial indicators.

This approach is closely connected to proxy VARs (Stock and Watson ,2012) and (Mertens and Ravn ,2013), which use high-frequency interest rate surprises to detect monetary policy changes (Gertler and Karadi 2015).

The novelty in their proposal is their use of sign restrictions on multiple high-frequency surprises, allowing them to identify multiple simultaneous shocks. In particular, they use the three-month fed funds futures to measure changes in expectations about short-term interest rates and the S&P 500 index to assess changes in stock values within a half-hour window around FOMC announcements.

They assume that during this brief period, only two structural shocks - a monetary policy shock and a central bank information shock - systematically influence financial market surprises (Jarocinski and Karadi, 2020).

1.2 Modeling imperfect information in economic models.

In this exploration, I will venture into the methods and approaches employed by economists to dissect and quantify information shocks.

The considerations highlighted in models of imperfect information are two:

First, as observed in Coibion and Gorodnichenko (2015), for models of imperfect information, a commonly observed phenomenon is the slower response of average expectations to changes in economic fundamentals when compared to the variables being forecasted. This suggests that revisions in expectations, and notably, shifts in market prices, exhibit correlations over time. These correlations offer insights into both current and past structural shocks.

Second, Melosi (2017) - Romer and Romer (2000), recognize the inherent information asymmetry between policymakers and market participants. This insight underscores that the actions taken by central banks, which are observable, can provide valuable information about underlying economic fundamentals.

Miranda, Agrippino, and Ricco (2021) presented a model that incorporates noisy and asymmetric information dynamics into their research. Let's consider an economy where a *k*-dimensional vector representing macroeconomic fundamentals follows an autoregressive process.

Equation (1):

$$X_t = \rho X_{t-1} + \xi_t \qquad \xi_t \sim N(0, \Sigma_\xi)$$

Where ξt represents the vector of structural shocks. Each time period, denoted as t, is divided into two distinct stages: an opening stage $\underline{t}$ and a closing stage $\bar{t}$. During the opening stage $\underline{t}$, structural shocks become realized. Notably, economic agents and central banks lack direct observation of the variable X_t. Instead, they rely on a Kalman filter to form their expectations about X_t, drawing from private, yet noisy, signals. Specifically, private agents receive $S_{I-\bar{t}}$, while the central bank receives $S_{CB-\bar{t}}$. These signals inform their respective conditional forecasts, denoted as $F_{I-\bar{t}}$ and $F_{CB-\bar{t}}$.

Economic agents have the capacity to engage in securities trading, such as futures contracts, based on the policy rate realization at time $t + h$, denoted as P (i_{t+h}). The pricing of a futures contract for i_{t+h} is a reflection of their collective expectations concerning the variable X_t. This relationship can be described as follows. **Equation (2):**

$$P_{\bar{t}}\,(i_{t+h}) = F_{\bar{t}} X_{t+h} + \mu_t$$

The term μ_t in this context represents a stochastic element, which could encompass factors such as the risk premium (Gürkaynak, Sack, and Swanson, 2005) or a stochastic process related to asset supply (Hellwig, 1980) and (Admati, 1985).

At the closing stage $\bar{t}$, and contingent on its internal forecast $F_{CB-\bar{t}}\, X_t$, the central bank formulates the interest rate i_t for the ongoing period through the application of a Taylor rule. **Equation (3):**

$$i_t = \phi_0 + \phi'_x F_{CB,\underline{t}} X_t + u_t + W_{t|t-1}$$

The variable u_t represents the monetary policy shock. Additionally, at this point, the central bank has the option to either publicly announce or indirectly leak a deviation from the Taylor rule, marked $W_{t|t-1}$ .This deviation becomes effective at time t.

Following the observation of the current policy rate, economic agents adjust their forecasts and engage in trading at $\bar{t}$.

When conditioned on the interest rate at time $(t-1)$, the act of observing the current interest rate is analogous to receiving a public signal, encompassing common noise, $S_{CB-\bar{t}}$ disseminated by the central bank.

Due to this forecast update stemming from the policy announcement, the pricing of futures contracts experiences a revision that is proportionate to the average adjustment of expectations across the population. This revision can be expressed as the **Equation (4):**

$$P_{\bar{t}}\,(i_{t+1}) - P_{\underline{t}}\,(i_{t+1}) \propto (F_{\bar{t}}X_{t+1} - F_{\underline{t}}X_{t+1})$$

The terms $F_{\underline{t}}\,x_{t+1}$ and $F_{\bar{t}}\,x_{t+1}$ represent the average forecast updates that result from the signals $S_{I-\underline{t}}$ and $S_{CB-\bar{t}}$ respectively.

After a central bank policy announcement, aggregate revisions in expectations progress according to the equation (4). **Equation (5):**

$$\begin{aligned}(F_{\bar{t}}X_t - F_{\underline{t}}X_t) &= (1-K_2)(1-K_1)[F_{\overline{(t-1)}}X_t - F_{\underline{(t-1)}}X_t] + K_2(1-K_1)\xi_t + K_2[v_{CB\,\underline{t}} - (1 \\ &\quad - K_1)\rho v_{CB,\underline{(t-1)}}] + K_2\left(K_{cb}\phi'_x\right)^{-1}[u_t - \rho(2-K_{cb}-K_1)u_{t-1} + (1-K_1)(1 \\ &\quad - K_{cb})\rho^2 u_{t-2}]\end{aligned}$$

In this equation, K_1 and K_2 are the Kalman gains used by economic agents in forming their expectations through $F_{\underline{t}}$ and $F_{\bar{t}}$ respectively. K_{CB} , on the other hand, signifies the Kalman gain employed by the central bank. The variable u_t corresponds to the monetary policy shock, while v_{CB}, pertains to the observational noise originating from the central bank. These elements collectively contribute to the evolution of aggregate expectation revisions in response to monetary policy announcements.

Equation (5) reveals two crucial features of models operating under the assumption of imperfect information that have significant implications for identifying monetary policy shocks.

The first feature is that average revisions in expectations (and consequently high-frequency surprises), which serve as a direct measure of shocks under conditions of complete information, are not orthogonal to their past values or past information. This is due to the gradual assimilation of new information over time.

The second notable feature is that observable policy actions have the capacity to convey information about economic fundamentals from the policymaker to market participants. As seen in equation (5), economic agents update their expectations by extracting information about the structural shocks ξt from the policy announcement. This phenomenon is often referred to as the "Fed information effect," observed in the works of Romer and Romer (2000) and Nakamura and Steinsson (2018), or the "signaling channel," as discussed by Melosi (2017) and others like Tang (2013) and Hubert and Maule (2016).

This implicit disclosure of information has a significant impact on the transmission of monetary impulses and the central bank's ability to stabilize the economy. Failure to account for this effect can result in both price and output puzzles, as illustrated by the fact that a policy rate increase can be interpreted differently by information-constrained agents. It can be viewed as either a deviation from the central bank's monetary policy rule, signifying a contractionary monetary shock, or as an endogenous response to anticipated inflationary pressures on the horizon. Despite both scenarios leading to a policy rate increase, they imply markedly different outcomes for macroeconomic aggregates and agents' expectations.

Moreover, equation (5) offers testable predictions about market-based monetary surprises. In the presence of imperfect information, these surprises exhibit three key characteristics: they are (i) serially correlated, (ii) predictable using other macroeconomic variables, and (iii) correlated with the central bank's projections of relevant economic indicators. These features offer valuable insights into the impact of imperfect information on the identification of monetary policy shocks and their effects on the economy (Miranda, Agrippino, and Ricco 2021).

1.3 Testing serial correlation, predictive patterns, and the Fed's projections in monetary policy analysis.

In this section, the empirical analysis aims to verify the three testable implications discussed earlier. Test for (i) serial correlation, (ii) predictability with lagged state variables for leading instruments for monetary policy shocks, and (iii) correlation with the Fed's internal forecasts.

Table 1 examines the correlation between high-frequency market surprises in the fourth federal funds futures (FF4) and various economic forecasts. This analysis is based on movements in the fourth federal funds futures contracts surrounding (FOMC) announcements, as proposed by Gürkaynak, Sack, and Swanson (2005). Dataset from 1990 to 2009.

The first column of Table 1 corresponds to a regression model similar to Romer and Romer (2004) and includes forecasts for output, inflation, and other economic indicators over various time horizons. The results show a statistically significant relationship between high-frequency market surprises and the Federal Reserve's Greenbook forecasts, in line with the intuition presented in the authors' model. However, interpreting individual coefficients is complicated due to multicollinearity among forecasts for the same variables at different horizons.

Subsequent columns (2 to 5) evaluate the predictive content of forecasts and forecast revisions grouped by horizon. The null hypothesis of joint non significance of coefficients is rejected for all horizons up to one quarter ahead. This suggests that the information transfer primarily pertains to the central bank's short-term macroeconomic outlook. Furthermore, output forecasts have significant and positive coefficients, indicating their role in capturing aggregate demand shocks and their effects on prices through the Phillips curve. On the other hand, negative coefficients on inflation forecasts may reflect the impact of supply shocks to which the central bank might respond differently.

The authors also address concerns about the influence of unscheduled FOMC meetings, which might attract market attention during times of economic distress. To address this concern, they re-run the regression using data only from scheduled FOMC meetings and confirm the robustness of the results (Miranda, Agrippino, and Ricco 2021).

Table 1—Central Bank Information Channel

	(1)	(3)	(4)	(5)	(6)
Output forecasts					
$h = -1$	0.001	0.005			
	(0.004)	(0.002)			
$h = 0$	0.006		0.009		
	(0.006)		(0.003)		
$h = 1$	0.001			0.003	
	(0.009)			(0.004)	
$h = 2$	−0.001				0.001
	(0.008)				(0.005)
$h = 3$	−0.004				
	(0.008)				
Inflation forecasts					
$h = -1$	−0.009	−0.008			
	(0.006)	(0.004)			
$h = 0$	0.012		−0.000		
	(0.008)		(0.005)		
$h = 1$	−0.036			−0.014	
	(0.016)			(0.006)	
$h = 2$	0.040				−0.008
	(0.020)				(0.006)
$h = 3$	−0.018				
	(0.019)				
Unemployment forecasts					
$h = -1$		0.002			
		(0.005)			
$h = 0$	0.003		0.001		
	(0.005)		(0.004)		
$h = 1$				−0.000	
				(0.003)	
$h = 2$					−0.003
					(0.003)
$h = 3$					
Output forecasts revisions					
$h = -1$	−0.007	−0.010			
	(0.006)	(0.005)			
$h = 0$	0.000		0.005		
	(0.008)		(0.005)		
$h = 1$	0.008			0.006	
	(0.011)			(0.007)	
$h = 2$	0.010				−0.005
	(0.010)				(0.010)
Inflation forecasts revisions					
$h = -1$	−0.004	−0.002			
	(0.010)	(0.009)			
$h = 0$	−0.002		0.004		
	(0.010)		(0.009)		
$h = 1$	0.043			0.018	
	(0.021)			(0.012)	
$h = 2$	−0.021				0.023
	(0.025)				(0.018)

Table 1—Central Bank Information Channel (*continued*)

Unemployment forecasts revisions					
$h = -1$	0.068	0.053			
	(0.067)	(0.065)			
$h = 0$	−0.015		0.009		
	(0.047)		(0.028)		
$h = 1$	−0.098			−0.028	
	(0.076)			(0.028)	
$h = 2$	0.104				−0.023
	(0.064)				(0.029)
Constant	−0.011	−0.022	−0.040	0.005	0.016
	(0.045)	(0.036)	(0.037)	(0.034)	(0.036)
R^2	0.044	0.027	0.045	0.040	−0.004
F-statistic	1.651	2.024	2.636	2.436	1.045
p-value	0.039	0.065	0.018	0.027	0.398
Observations	186	186	186	186	186

Miranda – Agrippino – Ricco (2021)

Table 2 presents the results of tests conducted to assess autocorrelation in both high-frequency and narrative instruments used to measure monetary policy shocks. High-frequency surprises are aggregated at a monthly frequency using various methods, and the findings are as follows:

The first column reports the results for an instrument created by summing all $FF4$ surprises registered between 1990 and 2009 within each month. These surprises are derived from high-frequency movements in financial markets following Federal Open Market Committee (FOMC) announcements.

The second column, only scheduled FOMC meetings are considered. It excludes unscheduled meetings.

The third column the results are reported for an instrument introduced by Gertler and Karadi in 2015. Their method involves monthly aggregation that takes into account the date of the FOMC meeting within each month and assigns weights based on the assumption of a one-month duration for each event.

The fourth column presents findings related to the narrative instrument introduced by Romer and Romer in 2004 (referred to as MPNt). This narrative instrument is constructed by running a regression that links the change in the policy rate to central bank forecasts, using an empirical Taylor rule. The residuals from this regression are used as a measure of monetary policy shocks.

The results reveal that serial correlation is present in the series of high-frequency surprises that are registered around scheduled FOMC meetings only (denoted as $FF4_t{}^{\dagger}$). This observation aligns with the theoretical framework presented above. It is important to note that this analysis is somewhat limited due to the relatively small number of scheduled FOMC meetings each year, which results in missing data points in the regression.

The serial correlation structure is weaker for the series that includes both scheduled and unscheduled FOMC meetings ($FF4_t$). This is likely due to the more unsystematic nature of unscheduled events.

In contrast, the high-frequency instrument of Gertler and Karadi (2015) exhibits strong autocorrelation. This is partially attributed to the weighting scheme used in their monthly aggregation, as also observed in studies by Stock and Watson (2012) and Ramey (2016).

While these findings are not entirely conclusive, they provide evidence that supports the presence of information frictions in the economy. Additionally, the null hypothesis is strongly rejected for the narrative series of Romer and Romer (2004), indicating the importance of narrative instruments in capturing monetary policy shocks, as demonstrated by Stock and Watson (2012).

TABLE 2—SERIAL CORRELATION IN INSTRUMENTS FOR MONETARY POLICY

	$FF4_t$	$FF4_t^{\dagger}$	$FF4_t^{GK}$	MPN_t
instrument_{t-1}	0.065	−0.164	0.380	0.014
	(0.090)	(0.057)	(0.137)	(0.091)
instrument_{t-2}	−0.025	−0.048	−0.164	0.227
	(0.119)	(0.066)	(0.073)	(0.087)
instrument_{t-3}	0.145	−0.066	0.308	0.381
	(0.130)	(0.073)	(0.150)	(0.102)
instrument_{t-4}	0.179	−0.007	−0.035	0.075
	(0.105)	(0.068)	(0.094)	(0.102)
Constant	−0.016	−0.011	−0.011	0.011
	(0.005)	(0.004)	(0.003)	(0.015)
R^2	0.026	0.001	0.168	0.172
F-statistic	1.459	2.279	2.965	7.590
p-value	0.217	0.063	0.021	0.000
Observations	167	167	166	152

Notes: AR(4) for instruments in each column. From left to right, the monthly surprise in the fourth federal funds future ($FF4_t$), the monthly surprise in the fourth federal funds future in scheduled meetings only ($FF4_t^{\dagger}$), the instrument in Gertler and Karadi (2015) ($FF4_t^{GK}$), and the narrative series of Romer and Romer (2004) (MPN_t). 1990:2009. Robust standard errors are in parentheses.

Miranda – Agrippino – Ricco (2021)

In Table 3, the authors introduce a test to evaluate the predictability of monetary policy shocks using historical information. Specifically, they project various measures of monetary policy shocks onto a set of lagged macro-financial factors, which are derived from a comprehensive dataset of over 130 monthly economic and financial variables compiled by

McCracken and Ng (2015). These variables cover key macroeconomic indicators and financial metrics. The regressions take into account data with a one-month lag.

TABLE 3—PREDICTABILITY OF MONETARY POLICY INSTRUMENTS

	$FF4_t$	$FF4_t^{\dagger}$	$FF4_t^{GK}$	MPN_t
$f_{1,t-1}$	−0.012	−0.007	−0.011	−0.087
	(0.006)	(0.003)	(0.004)	(0.021)
$f_{2,t-1}$	0.001	0.000	0.004	−0.009
	(0.003)	(0.002)	(0.002)	(0.010)
$f_{3,t-1}$	0.002	0.003	−0.001	0.000
	(0.005)	(0.004)	(0.004)	(0.012)
$f_{4,t-1}$	0.015	0.008	0.008	0.060
	(0.007)	(0.004)	(0.004)	(0.023)
$f_{5,t-1}$	0.002	−0.005	−0.000	0.002
	(0.007)	(0.004)	(0.004)	(0.026)
$f_{6,t-1}$	−0.011	−0.009	−0.006	−0.003
	(0.005)	(0.003)	(0.003)	(0.011)
$f_{7,t-1}$	−0.010	−0.009	−0.005	−0.041
	(0.006)	(0.004)	(0.004)	(0.016)
$f_{8,t-1}$	−0.001	−0.002	0.000	−0.028
	(0.003)	(0.002)	(0.003)	(0.012)
$f_{9,t-1}$	−0.002	−0.001	−0.004	−0.036
	(0.004)	(0.003)	(0.003)	(0.021)
$f_{10,t-1}$	−0.004	−0.001	0.000	0.030
	(0.005)	(0.003)	(0.003)	(0.012)
Constant	−0.014	−0.006	−0.011	0.010
	(0.004)	(0.003)	(0.003)	(0.011)
R^2	0.075	0.097	0.145	0.182
F-statistic	2.297	2.363	3.511	3.446
p-value	0.011	0.009	0.000	0.000
Observations	239	239	268	216

Notes: Regressions include a constant and one lag of the dependent variable. 1990:2009. From left to right, the monthly surprise in the fourth federal funds future ($FF4_t$), the monthly surprise in the fourth federal funds future in scheduled meetings only ($FF4_t^{\dagger}$), the instrument in Gertler and Karadi (2015) ($FF4_t^{GK}$), and the narrative series of Romer and Romer (2004) (MPN_t). The ten dynamic factors are extracted from the set of monthly variables in McCracken and Ng (2015). Robust standard errors are in parentheses.

Miranda – Agrippino – Ricco (2021)

The results presented in Table 3 confirm that market-based monetary surprises can be predicted using past information. This finding indicates that past data is valuable in understanding and forecasting monetary policy shocks. Additionally, the analysis demonstrates that narrative measures of "unanticipated" changes in interest rates are also predictable using state variables. These state variables are influenced by past structural shocks.

This outcome is consistent with the earlier observation of serial correlation in Table 2, which suggested the presence of information frictions in the economy. Furthermore, the fact that factors are estimated using the latest available data, which may include revisions to initial releases, underscores the idea that imperfect information plays a role in economic dynamics.

In a world of perfect information, markets would efficiently aggregate data, and there would be no need for data revisions or the involvement of national accounting offices.

In conclusion, these findings highlight the significance of considering the Fed's internal forecasts, the serial correlation of high-frequency surprises, and the predictability of market-based monetary surprises using historical data and state variables. Taking into account this evidence, the following section of the study proposes an instrument to account for the presence of information frictions in the economy.

1.4 Monetary policy shocks in focus: an instrument accounting for information dynamics.

To address the presence of information frictions and define monetary policy shocks, the research proposes an instrument that considers both the gradual absorption of information within the economy and the signaling channel of monetary policy, arising from the differing information sets of the central bank and market participants.

The instrument for monetary policy shocks is constructed by isolating the component of high-frequency market surprises triggered by policy announcements that is orthogonal to both the central bank's economic projections and past market surprises. This process unfolds in three steps.

First, they project high-frequency market-based surprises in the fourth federal funds futures around FOMC announcements onto Greenbook forecasts and forecast revisions for real output growth, inflation (measured as the GDP deflator), and the unemployment rate, following a methodology similar to Romer and Romer (2004). This approach allows to control for the influence of the central bank's private information, thereby accounting for the central bank information channel. The following regression is conducted at the frequency of FOMC meetings. **Equation (6):**

$$FF4_m = \alpha_0 + \sum_{j=-1}^{3} \theta_j \, F_m{}^{cb} x_{q+j} + \sum_{j=-1}^{2} \vartheta_j \, [F_m{}^{cb} x_{q+j} - F_{m-1}{}^{cb} x_{q+j}] + MPI_m$$

In this process, $FF4_m$, represents the high-frequency market-based monetary surprise computed around the FOMC announcement indexed by m, ${F_m}^{cb} x_{q+j}$ represents Greenbook forecasts for the vector of variables x at the horizon $q + j$, which are compiled before each meeting, and $({F_m}^{cb} x_{q+j}) - ({F_{m-1}}^{cb} x_{q+j})$ represents revisions to forecasts between consecutive meetings.

These forecasts are typically released approximately a week before each scheduled FOMC meeting and serve as a proxy for the information available to the FOMC when making policy decisions.

The latest available forecast is used for each surprise. This first step yields an instrument for monetary policy shocks ($MPIm$) at the frequency of FOMC meetings, which accounts for the implicit transfer of information that occurs at the time of FOMC announcements.

Next, they create a monthly instrument by aggregating the daily ($MPIm$) within each month. In most cases, there is only one surprise per month, in which case the monthly surprise is simply the sum of the daily values. For months without FOMC meetings, a value of zero is assigned. Similar aggregation methods are employed in studies such as Stock and Watson (2012) and Caldara and Herbst (2019).

To address the characteristic slow absorption of information by economic agents, a common feature in models of imperfect information (Coibion and Gorodnichenko, 2015), they eliminate the autoregressive component from the monthly surprises. Let $MPIt$ represent the outcome of the monthly aggregation explained in the previous step. Our monthly monetary policy instrument $MPIt$ is formed by the residuals of the following regression. **Equation (7):**

$$MPI_t = \phi_0 + \sum_{j=1}^{12} \phi_j \, MPI_{t-j} + MPI_t$$

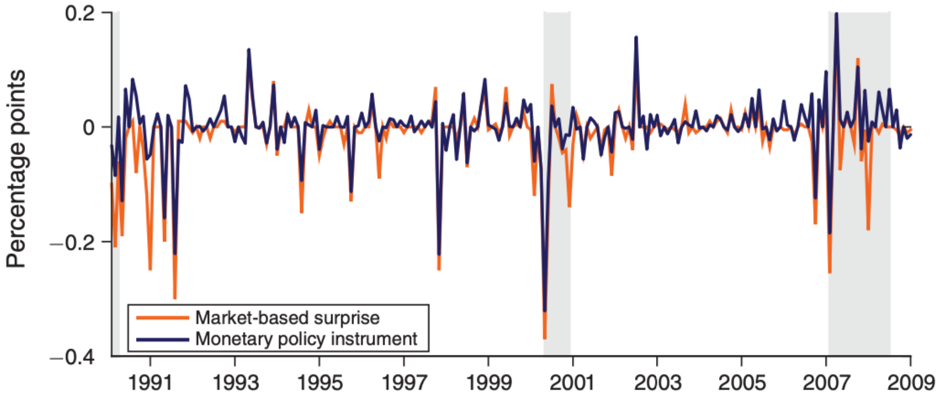
The regression defined in equation (7) use only data corresponding to non-zero $MPIt$ observations for the dependent variable. In months without FOMC meetings, MPIt is set to zero.

The conceptual model presented above provides the rationale for constructing the instrument as described in equation (6) to equation (7). Greenbook forecasts (and revisions) have a direct influence on the central bank's information set, and therefore, they account for the macroeconomic information transmitted to economic agents through policy announcements.

Even if there are potential misrepresentations in the empirical Taylor rule used in equation (6), the central bank's forecasts for output, inflation, and unemployment are expected to encompass the macroeconomic shocks that guide the monetary authority's policy decisions.

This, in conjunction with the lagged surprises, helps mitigate the reliance of high-frequency instruments on other concurrent and previous macroeconomic shocks.

In Figure 1, the monthly aggregation of the market monetary surprise obtained by summing daily surprises ($FF4_m$, orange line) alongside the instrument produced using our approach (MPIt, blue line). It's important to note that differences between the two series are especially conspicuous during periods of economic distress (Miranda, Agrippino, and Ricco 2021).



INFORMATIONALLY ROBUST INSTRUMENT FOR MONETARY POLICY SHOCKS

Notes: Market-based surprises conditional on private agents' information set $FF4_t$ (orange line), residual to equation (8) MPI_t (blue line). Shaded areas denote NBER recessions.

Figure 1 - Miranda – Agrippino – Ricco (2021)

In the upcoming section of this thesis, I will leverage the insights gained from the results of the market monetary surprise instrument. The goal is to employ a Vector Autoregression (VAR) model, which allows us to analyze the dynamic interactions among various economic variables. Specifically, I will create an Impulse Response Function (IRF) and perform Forecast Error Variance Decomposition (FEVD) to gain a deeper understanding of how these monetary policy shocks impact both European and US financial markets.

The VAR model will help us examine how these shocks affect key macroeconomic indicators such as inflation, unemployment rates, short-term and long-term interest rates. Additionally, I will explore their impact on the business cycle and indicators of financial distress.

Chapter 2

What is the impact of information shocks on short-term interest rates in the US and the EU? How does a monetary policy shock influence the unemployment rate? What effects do INFO shocks have on exchange rates and the global financial cycle? In the upcoming chapters, I will delve into these and related questions through a quantitative exploration employing a series of VAR models. It will be provided a detailed description of the VAR world, introduced the underlying theory and the approach used by VAR researchers. This foundation is essential for a comprehensive understanding of the subsequent analysis.

2.1 VAR models

Two decades ago, Christopher Sims (1980) introduced the concept of vector autoregressions (VARs), which offered a novel framework for macroeconometric analysis. Unlike univariate autoregressions that explain a single variable based on its own past values, VARs are multi-equation models. In a VAR, each variable is explained by its own past values and the current and past values of the other variables in the system.

This framework provides a structured way to capture complex dynamics across multiple time series, and the accompanying statistical tools made it user-friendly and interpretable.

As highlighted by Sims (1980) and other influential early researchers, VARs held the potential to offer a unified and credible approach for tasks like data description, forecasting, structural inference, and policy analysis (Stock and Watson, 2001).

VAR models come in three main varieties: *reduced form*, *recursive*, and *structural*.

In a *reduced form VAR*, each variable is expressed as a linear function of its own lagged values, as well as the lagged values of all other variables considered in the model. For instance, in a three-variable VAR there would be three equations. These equations, estimated using ordinary least squares (OLS) regression, capture the relationships between the variables over time. The error terms in these regressions represent the unexpected or "surprise" movements in each variable after considering its past values. These errors can be correlated across equations if the variables are correlated (Stock and Watson, 2001).

In a *recursive VAR* (Vector Autoregressive) model, the error terms in each regression equation are intentionally constructed to be uncorrelated with the error terms from previous equations. This is achieved by incorporating contemporaneous values of other variables as regressors in each equation. Recursive VARs are employed to perform regressions of a specific variable on all the other variables within the system. This process essentially quantifies how the dependent variable is influenced by not only its own historical values but also by the historical values of the other variables in the system. The key objective is to reveal the interdependencies among these variables while maintaining the model's structural integrity through the sequential imposition of causal orderings. Estimating each equation using OLS results in residuals that are uncorrelated across equations (Stock and Watson, 2001).

In a *structural VAR* (SVAR), economic theory guides the identification of causal links among the variables. This approach requires "identifying assumptions" that permit the interpretation of correlations as causal relationships. Identifying assumptions can apply to the entire VAR model or specific equations within it. These assumptions yield instrumental variables that enable the estimation of contemporaneous links through instrumental variables regression. Researchers can create numerous structural VARs by selecting different identifying assumptions based on their research objectives.

The choice of VAR type depends on the research goals, the relationships between variables, and the availability of economic theory to inform the model's structure and assumptions. Each VAR type has its own strengths and limitations in capturing the dynamics of multivariate time series data.

In this specific case, on the basis of the scenario that will be analyzed, I will employ a SVAR. Given the intricate nature of structural shocks and the complexities involved in their direct identification, as mentioned above, will be used a proxy, in the form of a Recursive VAR. This approach, inspired by the work of Plagborg - Møller and Wolf (2021), will enable to indirectly estimate and analyze the effects of structural shocks.

Leveraging the source of MATLAB codes provided by Ambrogio Cesa – Bianchi, will be demonstrated how these shocks manifest their presence and influence both the European and United States markets.

2.2 Vector Autoregression

This section is designed to provide a comprehensive understanding of the technical aspects of VAR analysis. We will explore the underlying methodologies, mathematical formulations, and the key steps involved in implementing VAR models.

A VAR model can be understood as a linear mathematical model consisting of a set of n-equations, n-variables, and p-lag terms. In this model, each variable is influenced by its own historical values *(lags)* and the current and past values of the remaining $n - 1$ variables. The maximum number of lags considered is commonly referred to as the *"lag-order"* or simply the *"order"* of the VAR model.

A VAR model of order p, often denoted as VAR*(p)*, takes into account the lagged observations of the data series up to the p^i lag. It's important to note that the data frequency typically used for VAR models is monthly or quarterly (Kilian and Lütkepohl, 2017). To better understand, let's visualize this function. **Equation (8):**

$$X_t = A_1 X_{t-1} + \cdots + A_p X_{t-p} + U_t$$

Denoting $X_t = (x_{1t}, \dots, x_{nt})'$ as an $(nx1)$ vector containing the values that n variables assume at date t and $U_t = (u_{1t}, \dots, u_{nt})'$ are the errors with zero mean. Additionally, is assumed that the series X_t exhibits covariance stationarity, which implies:

$$\text{constant mean: } E[X_{it}] = \mu_i,$$
$$\text{constant variance: } var[X_{it}] = \sigma_i^2,$$
$$\text{constant autocovariance: } \mathrm{cov}[X_{it}, X_{it+\tau}] = \gamma_i(\tau).$$

In this context, X_t represents the vector collecting the current values of model variables, X_{t-1} gathering the values at a lag of 1, up to X_{t-p} that encompass the values at a lag n. We also have matrices $(A_1 \dots A_p)$ representing the coefficients associated with the first lags up to the p^i lag of the variables, respectively. These matrices can be represented as follows.

Equation (9):

$$\begin{bmatrix} X_{1t} \\ \vdots \\ X_{nt} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_{1t-1} \\ \vdots \\ x_{nt-1} \end{bmatrix} + \cdots + \begin{bmatrix} a_{1p} & \cdots & a_{pn} \\ \vdots & \ddots & \vdots \\ a_{np} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_{1t-p} \\ \vdots \\ x_{nt-p} \end{bmatrix} + \begin{bmatrix} u_{1t} \\ \vdots \\ u_{nt} \end{bmatrix}$$

In our case, in the final analysis of this thesis, a system of 7 variables will be used, and each variable will consider its own past values up to 6 lags.

Next step transforms equation (8) in a structural VAR. This transformation allows for a clearer and more interpretable understanding of the data. Considering the SVAR*(p)* model presented above. **Equation (10):**

$$B_0 X_t \; = \; B_1 X_{t-1} + \cdots + B_p X_{t-p} + w_t$$

X_t represents the vector of the time series of interest, as described in equation (8), B_i collects the slope coefficients at the *i^th^* lag, similar to A_i in equation (8), w_t is the vector of the so-called structural residuals or shocks, and B_0 is the matrix whose elements capture the contemporaneous relationship between the variables in the model (Kilian and Lütkepohl, 2017).

Unlike reduced form VARs, SVARs allow for immediate relationships among model variables. Additionally, SVARs assume that the structural shocks are zero-mean white noise processes (WN), and these shocks are serially and mutually uncorrelated.

This assumption results in the covariance matrix of the structural shocks, denoted as $\sum w_t$, taking the form of a full-rank diagonal matrix. Notably, the number of variables aligns with the number of shocks. This mutual uncorrelation among structural shocks allows for distinctive economic interpretations to be associated with each shock, making them inherently "structural".

Furthermore, the absence of correlation among these structural shocks simplifies the computation of Impulse Response Functions (IRFs) in SVARs as straightforward functions of the structural shocks. As a consequence, it becomes possible to interpret every fluctuation observed in the data series of a stable VAR as being generated by these structural innovations (Kilian and Lütkepohl, 2017).

SVARs and reduced form VARs share a crucial relationship: reduced form VARs can be seen as the data generated by an underlying SVAR. You can easily transform a SVAR model into the corresponding VAR representation by multiplying both sides of equation (10) by B_0^{-1}.

Equation (11):

$$X_t = \underbrace{B_0^{-1}B_1}_{S_1} X_{t-1} + \cdots + \underbrace{B_0^{-1}\, B_p}_{S_i} X_{t-p} + \underbrace{B_0^{-1}w_t}_{u_t}$$

This operation results in $S_i = B_0^{-1}B_i$ and $u_t = B_0^{-1}w_t$. The innovations in the reduced form are, in fact, linear combinations of the structural innovations. Conversely, knowing the matrix B_0 or its inverse, is all that's necessary to recover the SVAR process underlying an estimated VAR model.

By normalizing $E[w_t w_t{}'] \equiv \sum w_t = I_k$, without any loss of generality, this equation can be employed to retrieve B_0^{-1}. **Equation (12):**

$$E[u_t u'] = \sum u = B_0^{-1}\;\; B_0^{-1\prime}$$

This represents a system of nonlinear equations, with the unknown parameters being the elements of B_0^{-1}. It can only be solved if the number of unknown parameters does not exceed the number of independent equations provided by $\sum u_t$. This condition is known as the order condition and is essential for the precise identification of structural shocks. However, it is not met in this equation. In fact, there are k^2 unknown parameters and only $k(k + 1) / 2$ independent equations available. Consequently, to satisfy the order condition, it is necessary to impose certain restrictions on the elements of B_0^{-1}.

Selecting appropriate economic constraints to uniquely identify B_0^{-1}, and consequently, the other structural parameters, is known as the “identification problem” in structural autoregressions. Over the years, various strategies have been developed to address this challenge (Kilian and Lütkepohl, 2017).

However, in this dissertation, I will focus on presenting only the strategies that will be employed in the empirical research of the upcoming chapters. These strategies include the widely used Cholesky decomposition, which leads to recursively identified models. Before delving into these approaches, the upcoming pages will provide explanations of key concepts such as Impulse Response Functions (IRF) and Forecast Error Variance Decomposition (FEVD).

2.3 Structural Impulse Responses

Economists often focus on a fundamental research question: how do economic aggregates react to unexpected changes in an economic variable? This question can be translated into assessing the effects of a one-time variation (or impulse) in the structural shocks w_t on the model variables X_t, following the identification of w_t as described in equation (11).

Once B_0 and u_t are known, we can readily derive $w_t = B_0 u_t$. However, our primary interest typically lies not in the shocks themselves but in understanding how each component of $X_t = (x_{1t}, \dots, x_{nt})'$ responds to a one-time impulse in $w_t = (w_{1t}, \dots, w_{nt})'$.

This is commonly known as the structural impulse response of X_{jt} to w_{qt} at horizon *"i"* and is denoted as $\theta_{jq,i}$ (Kilian and Lütkepohl, 2017). **Equation (13):**

$$\theta_{jk,i} = \frac{\partial X_{jt+i}}{\partial w_{kt}}, \qquad h = 0,1,\dots,H$$

The usual objective is to depict how each variable responds to each structural shock as time progresses. Given that there are *K* variables and *K* structural shocks, there are K^2 impulse response functions (IRFs), each spanning a length of $H + 1$, where *H* represents the maximum horizon over which the shocks propagate.

A valuable initial step in estimating the structural impulse responses $\theta_{jk,i}$ is to consider the responses of X_{t+i} to the reduced-form errors u_t. These responses can be determined by examining the VAR *(1)* representation of the VAR*(p)* process, which provides insight into the dynamic relationships among the variables. **Equation (14):**

$$X_t = AX_{t-1} + U_t$$

Where, **Equation (15):**

$$X_t \equiv \begin{bmatrix} x_t \\ \vdots \\ x_{t-p+1} \end{bmatrix} \quad A \equiv \begin{bmatrix} A_1 & \cdots & A_p \\ I_K & 0 & \vdots \\ 0 & I_K & 0 \end{bmatrix} \quad U_t \equiv \begin{bmatrix} u_t \\ 0 \\ 0 \end{bmatrix}$$

If X_t is covariance stationary, it can be represented as a weighted average of current and past shocks, with weights that decrease as the shocks move further into the past. This multivariate Moving Average (MA) representation is as follows. **Equation (16):**

$$X_t = \sum_{i=0}^{\infty} \Theta_i \, u_{t-i} = \sum_{i=0}^{\infty} B_0^{-1} B_0 \, u_{t-i}$$

If the VAR being analyzed is not stable, the same method for calculating Θ_i will still apply. However, in this case, the impulse responses may not approach zero as i → ∞, and they will no longer represent the coefficients of the structural Moving Average representation.

The *jk^th^* element Θ_i denoted by $\theta_{jk,i}$, where j and k represent variable indices and *(i = 0, 1, ..., H*), signifies the response of variable *j* to structural shock *k* at the given horizon.

It is also feasible to create linear combinations of structural impulse responses. For instance, in a VAR model, we might have responses $\theta_{lk,i}$ for the nominal interest rate and $\theta_{mk,i}$ for the inflation rate. In this case, we can calculate the implied response of the real interest rate as ($\theta_{lk,i} - \theta_{mk,i}$), for (*i = 0, 1, ..., H*) (Kilian and Lütkepohl, 2017).

Similarly, we can derive implied responses from the inflation rate, such as the implied response of the *log* price level, denoted as $\Delta\ pt$, based on its response over time.

In addition to linear transformations, it is also possible to compute non-linear transformations of impulse response functions, such as the "half-life" of responses. The half-life is defined as the time it takes for a response to decline to half of the initial impact response in absolute value (Kilian and Zha, 2002).

A consequence of the linearity of the VAR model is that responses to negative shocks are symmetric to responses to positive shocks. Furthermore, the magnitude of the structural shock does not affect the construction of IRFs.

Rescaling the shock only rescales the entire IRF. Typically, it is chosen to scale B_0^{-1} such that the structural shocks represent one standard deviation of the time series of structural shocks. This is because such a shock is considered of typical magnitude, but this choice is a convention and not a strict requirement.

It's also important to note that structural shocks, in general, are unit-free and do not have units of measurement associated with model variables. Only in specific cases will structural shocks be associated with a particular model variable (Kilian and Lütkepohl, 2017).

2.4 Forecast Error Variance Decomposition

A crucial question that a SVAR model can address is the allocation of forecast error variance decomposition (FEVD), or prediction mean squared error, of X_{t+h} at the horizon ($h = 0, 1, ..., H$) to each of the structural shocks, w_{kt}, where ($k = 1, ..., K$). In other words, it helps determine how each structural shock contributes to the deviations of individual variables from their predicted values at a given time horizon.

In a stationary model, as the forecast horizon extends to infinity ($h \rightarrow \infty$), the limit of the FEVD corresponds to the variance decomposition of X_t. This is because the forecast error covariance matrix converges to the unconditional covariance matrix of X_t. Therefore, for stationary systems, it is possible to construct an FEVD decomposition for an infinite forecast horizon.

In integrated systems, the FEVD diverges as the forecast horizon approaches infinity. However, the FEVD remains valid up to a finite maximum horizon of H. To compute this decomposition, we can leverage the *i* matrices, which we have already calculated for the structural impulse response analysis (Kilian and Lütkepohl, 2017). **Equation (17):**

FEVD at horizon *h is:*

$$FEVD(h) = \sum_{i=0}^{h-1} \Phi_i \Phi_i{}'$$

Where Φ_i refers to the structural impulse matrix at a specific horizon, denoted as *i*.

Employing the notation established in the previous section, the determination of how shock q contributes to the mean squared prediction error FEVD of the variable X_{jt} at horizon *h* can be expressed as follows.

Equation (18):

$$FEVD_j^k(h) = \theta_{jk,0} + \cdots + \theta_{jk,h-1}$$

As a result, the cumulative mean squared prediction error FEVD of the variable X_{jt} at horizon *h* can be expressed as follows. **Equation (19):**

$$FEVD^k(h) = \sum_{i=1}^{k} FEVD_j^k(h)$$

For a given h and k, the decomposition is as follows. **Equation (20):**

$$1 = \frac{FEVD_1^k(h)}{FEVD^k(h)} + \frac{FEVD_2^k(h)}{FEVD^k(h)} + \cdots + \frac{FEVD_K^k(h)}{FEVD^k(h)}$$

One key aspect of interest is the patterns across horizons. When examining FEVD, we aim to understand how different structural shocks impact the forecast error variance of a particular variable over time. For instance, at shorter horizons, certain shocks may have a relatively small impact on the forecast error variance, while their importance might grow over the long term. This evolving pattern provides information about the strength and persistence of the relationship between specific shocks and the variable under consideration.

Additionally, researchers may want to explore the relative contributions of different shocks at a given horizon. This analysis can help understand which shocks dominate in explaining the forecast error variance at a particular point in time (Kilian and Lütkepohl, 2017).

2.5 Cholesky identification

A widely used method for uniquely identifying the structural shocks is by employing the lower-triangular *Cholesky decomposition* of $\sum u^2$. We begin by defining a k x k lower-triangular matrix denoted as P, characterized by a positive main diagonal and satisfying the condition $\sum u = PP'$. This matrix P is commonly referred to as the lower-triangular Cholesky decomposition of $\sum u$. It is crucial to note that the relationship $\sum u = B_0^{-1} B_0^{-1\prime}$ immediately implies that one potential solution for recovering w_t is $B_0^{-1} = P$. Since P is lower triangular, it possesses $k(k-1)/2$ parameters that are zero.

Consequently, the order condition necessary for the precise identification of unknown parameters in $B_0^{-1}$is met. An important implication of this context is that if $B_0^{-1}$is lower triangular, then B_o is also lower triangular (Kilian and Lütkepohl, 2017).

In essence, this identification strategy enforces $k(k-1)/2$ zero restrictions on the elements of both B_0^{-1} and B_0. This approach serves two primary purposes: it ensures that the reduced-form errors are mutually uncorrelated and introduces a specific recursive order or causal chain between the structural shocks and the model variables (Kilian and Lütkepohl, 2017).

One widely employed method for disentangling the structural innovations, denoted as w_t, from the reduced-form innovations, is represented by u_t, and it involves the process of *'orthogonalization'.*

In this context, orthogonalization entails rendering the errors mutually uncorrelated. To achieve this, a mechanical procedure is employed, as outlined below.

It is crucial to bear in mind that the process of orthogonalizing reduced-form residuals, using Cholesky decomposition, is only suitable when the underlying recursive structure represented by matrix P can be economically justified.

The distinctive characteristic of orthogonalization through Cholesky decomposition is that it results in a structural model that is recursive, particularly conditional on lagged variables.

Essentially, researchers establish a specific causal chain by imposing this approach, rather than learning about the causal relationships from the data itself. In essence, they solve the problem of identifying which structural shock is responsible for the variation in u_t by imposing a particular solution.

However, it is important to note that this mechanical solution lacks economic significance without a plausible economic interpretation for the chosen recursive ordering (Kilian and Lütkepohl, 2017).

The seemingly neutral and scientific term *'orthogonalization'* might obscure the fact that they make strong assumptions about the error term in the VAR model.

In the early 1980s, many users of VAR models did not fully grasp this point and believed that the data alone could provide all the necessary insights. Such “atheoretical” VAR models quickly faced criticism (Cooley and LeRoy, 1985).

This critique led to a more rigorous examination of the economic foundations of recursive models. It was demonstrated that, in specific cases, a recursive model could be given a structural or semi-structural interpretation (Bernanke and Blinder 1992). This critique also spurred the development of structural VAR models that impose non-recursive identifying restrictions (Sims, 1986), (Bernanke, 1986), (Blanchard and Watson, 1986).

It became widely acknowledged that the structural VAR model is simply a specific instance of the Dynamic Stochastic General Equilibrium Model, with the primary distinguishing feature being the nature of its identifying restrictions.

In practical terms, for each arrangement of the K variables in the VAR model, there exists a different matrix P.

Some argue that sensitivity analysis should be conducted based on alternative orderings of these variables. However, this proposal encounters three key challenges:

1. The economic justification for different orderings may vary significantly, making it difficult to draw robust conclusions.
2. Changing the ordering might alter the interpretation of the structural shocks, potentially leading to different economic insights.
3. The range of possible orderings can become quite extensive in larger models, complicating the sensitivity analysis.

These considerations underscore the need for careful deliberation and economic reasoning when selecting an appropriate orthogonalization strategy in VAR modeling (Kilian and Lütkepohl, 2017).

Chapter 3

3.1 Variable selection

The analysis encompasses a set of nine key macroeconomic variables and two policy shocks. The dataset comprises:

1. Short-term and long-term interest rates US.
2. Consumer Price Index US.
3. Unemployment rate US.
4. U.S. Dollars to Euro Spot Exchange Rate.
5. Short-term and long-term interest rates EU.
6. Consumer Price Index EU.
7. Unemployment rate EU.
8. Global Financial Cycle Proxy (GFC).
9. Financial distress Proxy (Baa10Y).

In addition, the two instruments used to represent monetary policy shocks are:

1. Monetary policy shock (MPI).
2. Central bank information shock (INFO).

The inflation variables for both the US and UE are expressed as log differences. This choice of using log transformations in time series regressions and macroeconomic analyses is rooted in the conventional normality assumption of classical econometric methodologies, as outlined by Mayr and Ulbricht (2007).

3.2 Detailed description of the data

INFO and MPI Shocks:

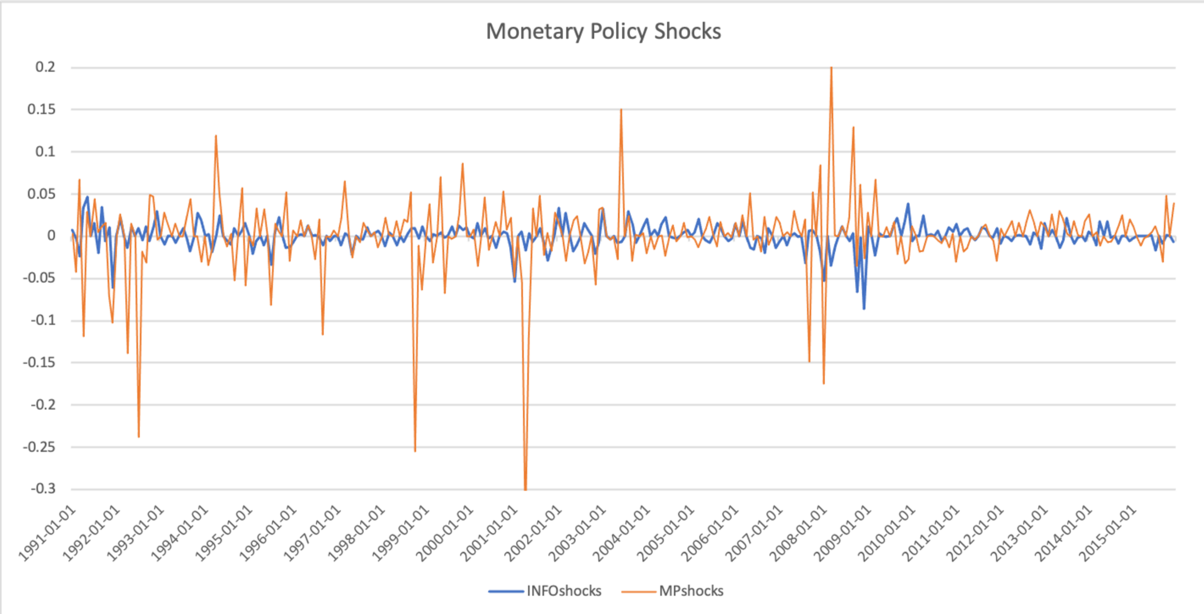


Figure 2

SOURCE:
https://github.com/GRicco/info-policy-surprises/tree/main

In this analysis we consider two alternative measures of identified U.S. monetary policy shocks, which are constructed by Miranda-Agrippino and Ricco (2021). The original INFO and MPI shocks are available at a monthly frequency. Consequently, all the data used in this analysis are provided on a monthly basis, which allows us to align well with the other variables without sacrificing information. Given that we are working with a VAR model consisting of 7 variables, resulting in a 7x7 matrix exposed to 6 lags, computations become significantly large. By maintaining the same monthly frequency, we ensure that we do not lose valuable information.

As mentioned at the beginning of this thesis, these variables lie at the core of the research. The objective spans two key aspects.

First, I aim to showcase the distinct impacts of INFO and MPI on macroeconomic international variables, demonstrating how the proposed theory aligns with the data results. Second, I endeavor to address the main question of this thesis: Can INFO shocks generate effects in the U.S. market, and are these effects strong enough to propagate into the European markets?

In the following paragraphs I will answer this question by providing a series of tests on the key economic variables mentioned earlier.

Global Financial Cycle (GFC) Proxy:

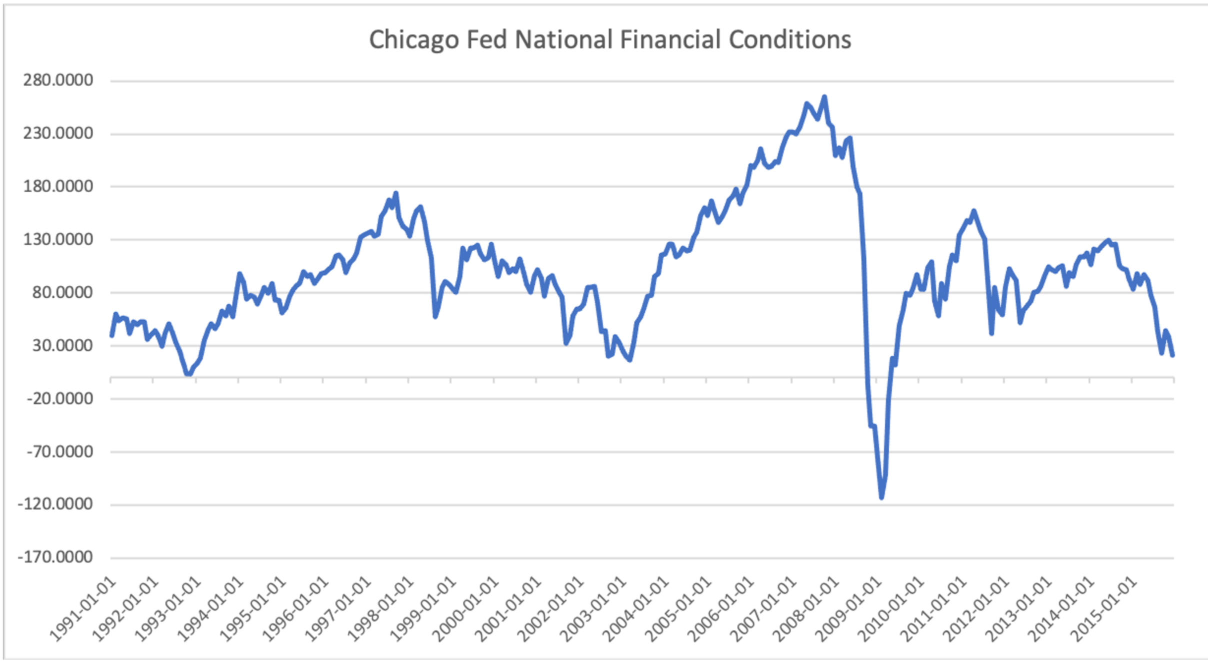


Figure 3

SOURCE:
https://fred.stlouisfed.org/series/NFCI#0

The Chicago Fed's National Financial Conditions Index (NFCI) offers a comprehensive weekly assessment of U.S. financial conditions, encompassing money markets, debt and equity markets, as well as both the traditional and "shadow" banking systems. Positive values of the NFCI indicate financial conditions that are tighter than the historical average, while negative values suggest looser-than-average financial conditions.

In this context, the NFCI will serve as a proxy for the global financial cycle, which represents the interconnected and cyclical movements in financial markets across the world. The global financial cycle often plays a significant role in influencing economic and financial developments on a global scale.

Financial distress Proxy:

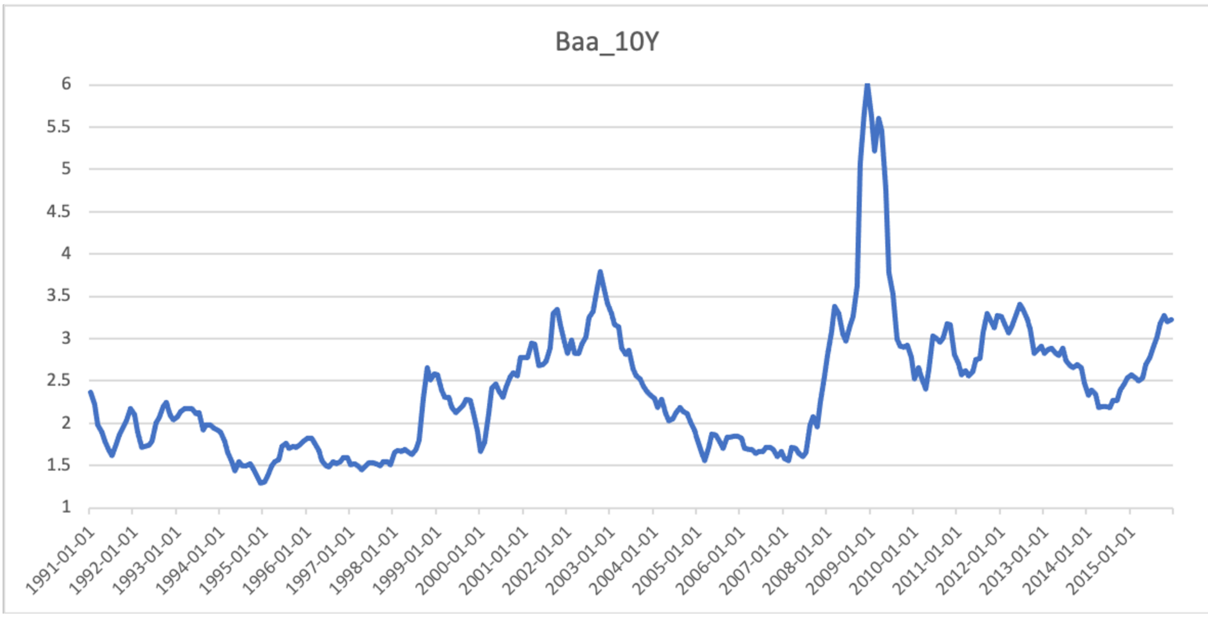


Figure 4

SOURCE:
https://fred.stlouisfed.org/series/BAA10Y#0

This series is derived by calculating the difference between Moody's Seasoned Baa Corporate Bond and the 10-Year Treasury Constant Maturity (BC_10YEAR). When this spread narrows, it serves as a synonym for financial distress, indicating a challenging economic situation. Conversely, a widening spread is considered an indicator of market stability. In this context, the series is employed as a proxy for financial distress. It will be used to understand if MPI or INFO shocks, followed by movements in the macroeconomic variables, are in line with the situation of financial stress or, conversely, financial expansion.

Short-term and long-term policy rates in the United States:

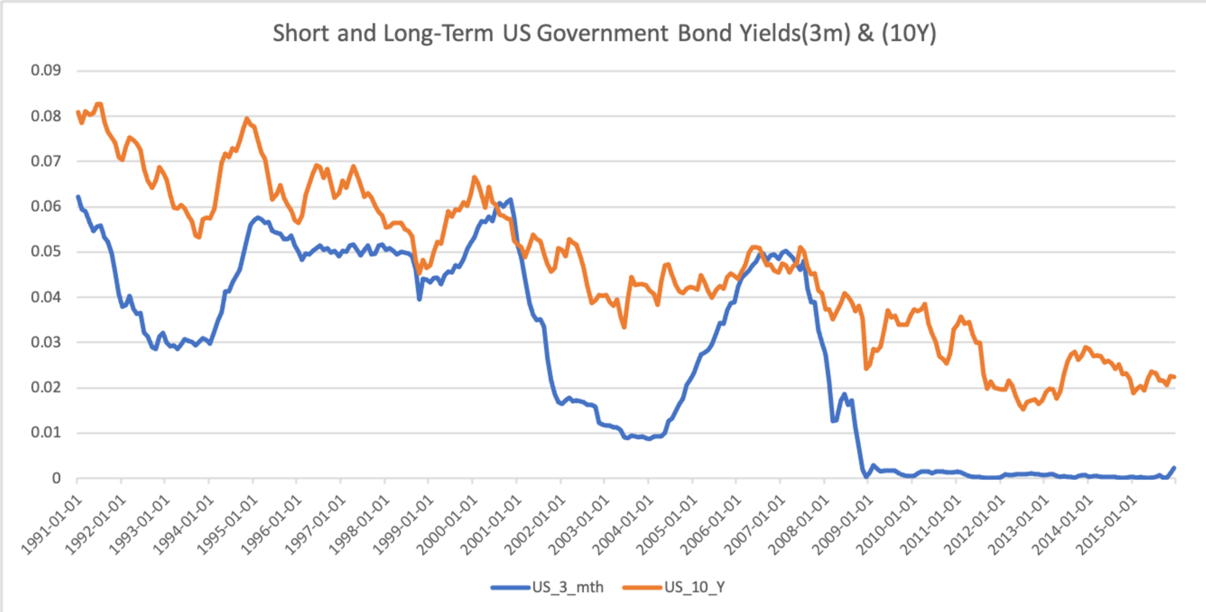


Figure5

SOURCE:
https://fred.stlouisfed.org/series/TB3MS
https://fred.stlouisfed.org/series/IRLTLT01USM156N

In this series are included data on the 3-Month Treasury Bill Secondary Market Rate and the 10-Year Long-Term Government Bond Yields. These datasets are instrumental in understanding how MPI and INFO influence Federal Reserve actions. These actions can either be restrictive, with the aim of restraining economic activity, or expansive, intended to stimulate the economy. Additionally, the long-term interest rate data are used to evaluate whether these effects have broader financial implications on the markets or not.

Short-term and long-term policy rates in the European Union:

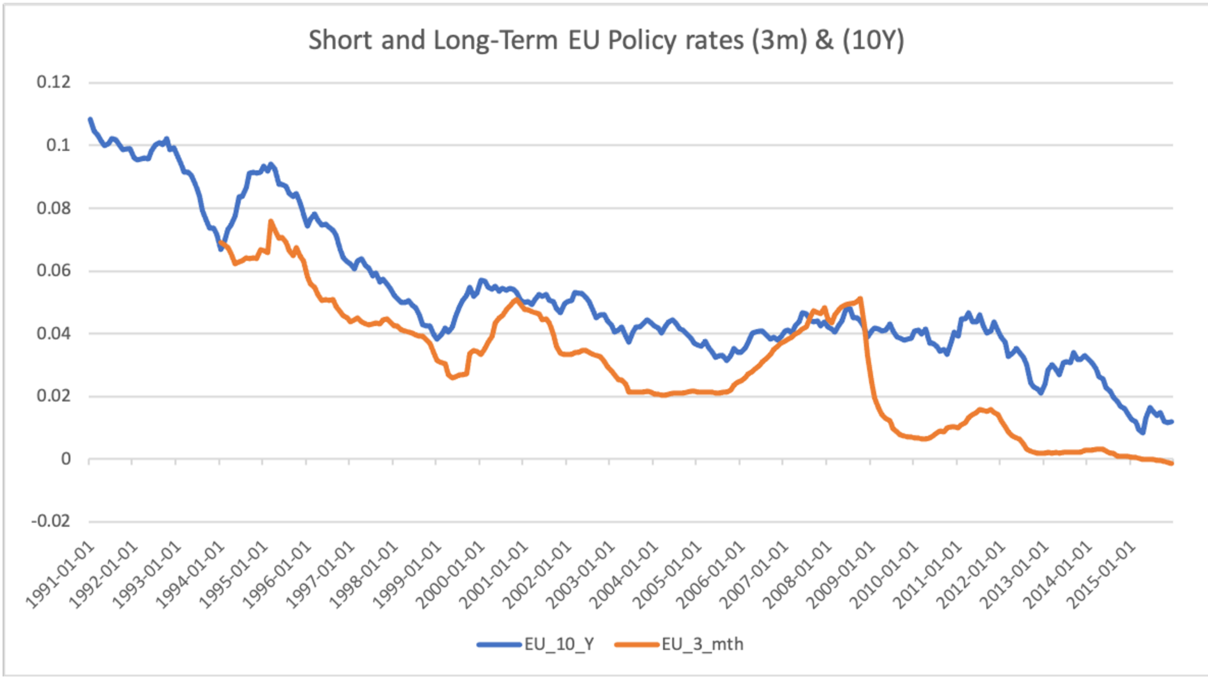


Figure 6

SOURCE:
https://fred.stlouisfed.org/series/IRLTLT01EZM156N
https://fred.stlouisfed.org/series/IR3TIB01EZM156N

In this section are included data on Interest Rates Long-Term Government Bond Yields 10-Year and 3-Month Interbank Rates of the Euro Area (19 Countries). These datasets play a help us to understand the potential impact of MPI and, more importantly, in answering the central thesis question: Can INFO shocks reach and influence the European market, and how significant is the resulting impact?

As in the previous section, the long-term rates enable us to assess whether the impact is also financial or if it affects just the short-term fluctuations of the economy.

Consumer Price Index US:

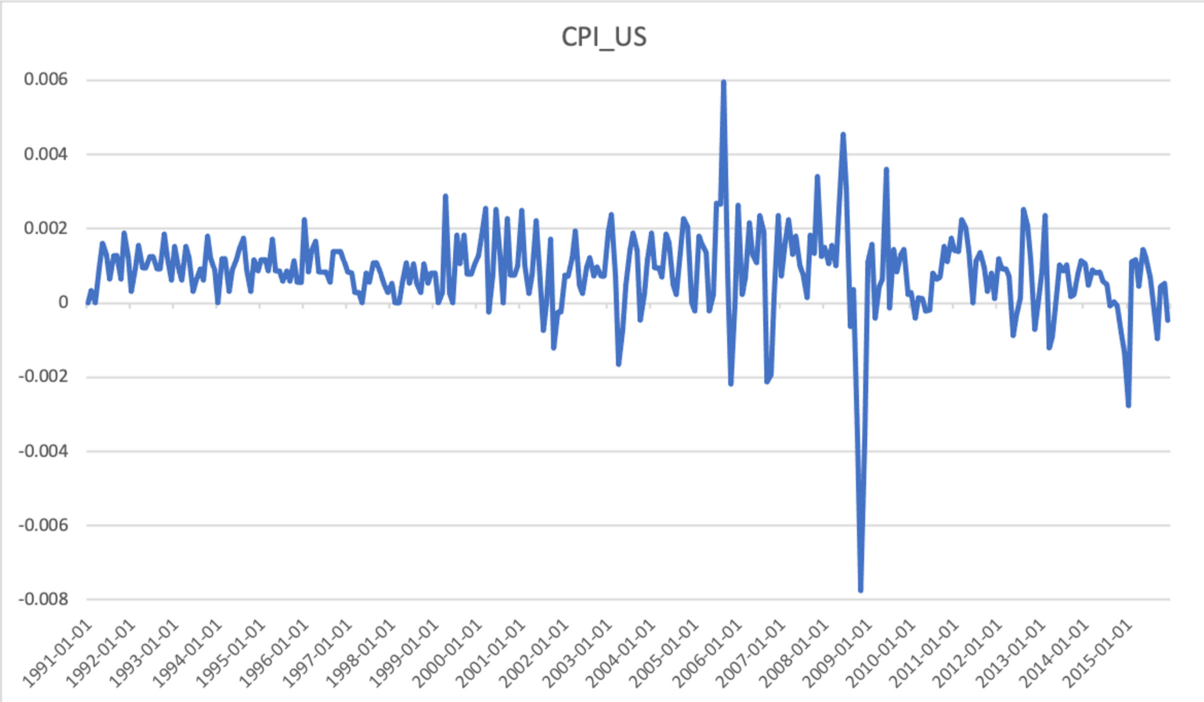


Figure 7

SOURCE:

https://fred.stlouisfed.org/series/CPIAUCSL

Unemployment rate US:

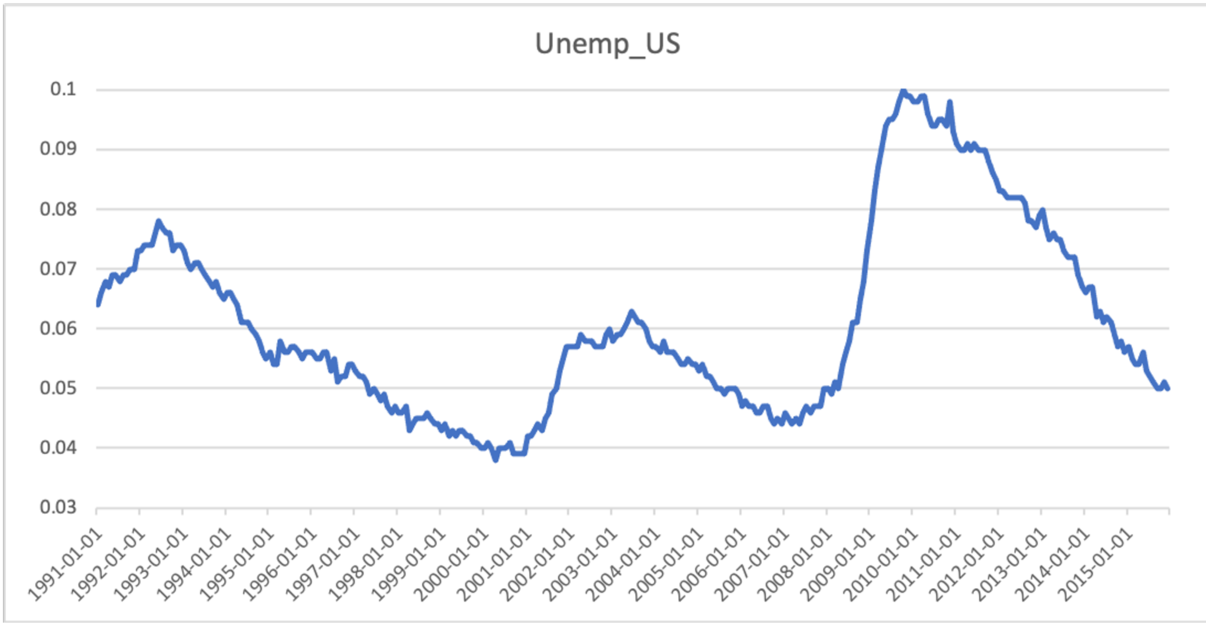


Figure 8

SOURCE:

https://fred.stlouisfed.org/series/UNRATE

U.S. Dollars to Euro, Spot Exchange Rate:

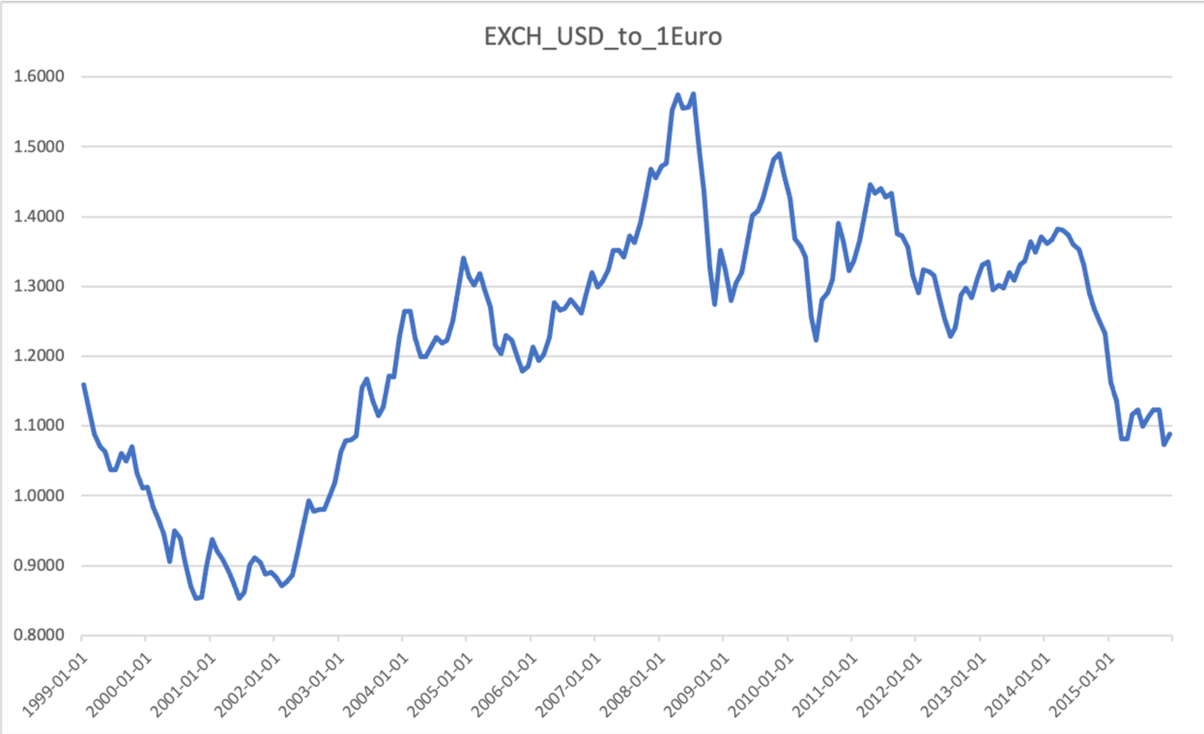


Figure 9

SOURCE:

https://fred.stlouisfed.org/series/DEXUSEU

The series is derived from the "U.S. Dollars to Euro Spot Exchange Rate". Its purpose is to assess the impact on the strength of the U.S. dollar in response to MPI or INFO shocks, particularly in relation to capital inflows or outflows following changes in monetary policy rates.

U.S. and UE Industrial productions indicator:

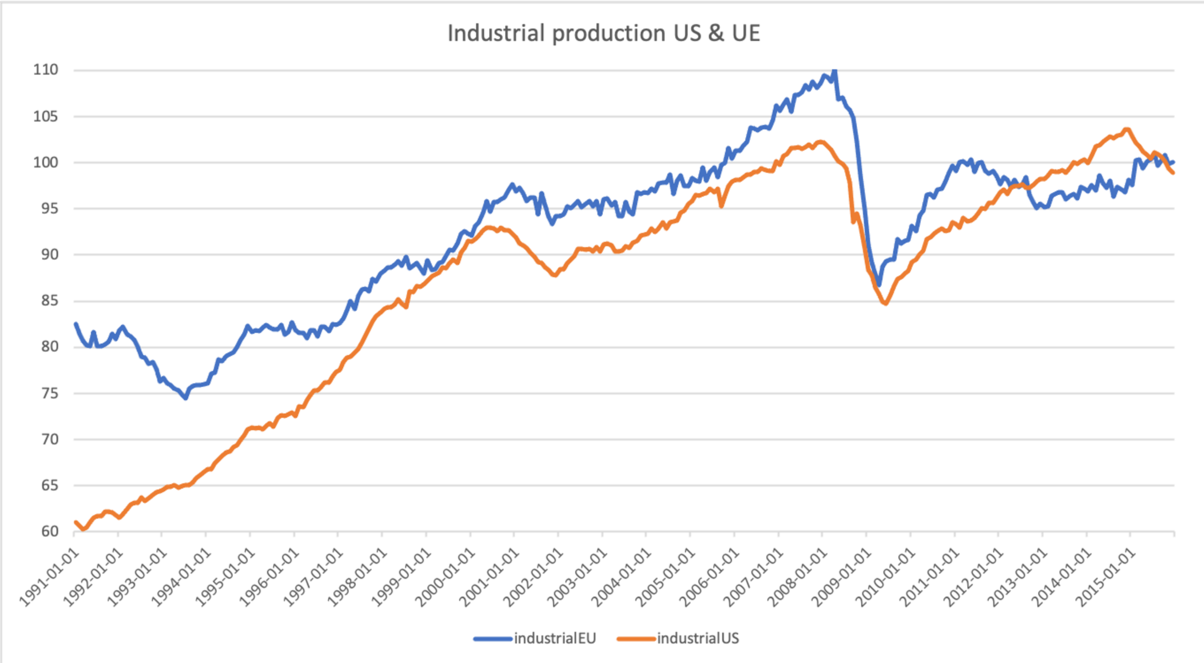


Figure 10

SOURCE:

https://fred.stlouisfed.org/series/EA19PRINTO01IXOBSAM

https://fred.stlouisfed.org/series/INDPRO

These series are utilized for conducting robustness checks. To validate the results and gain a different perspective, the research will investigate whether the impacts on unemployment, inflation, and interest rates are consistent with changes in industrial production. Initially, the effects on industrial production in the United States will be examined. Subsequently, an assessment will be made to determine whether these impulses also affect production levels in the European markets.

Inflation rate EU:

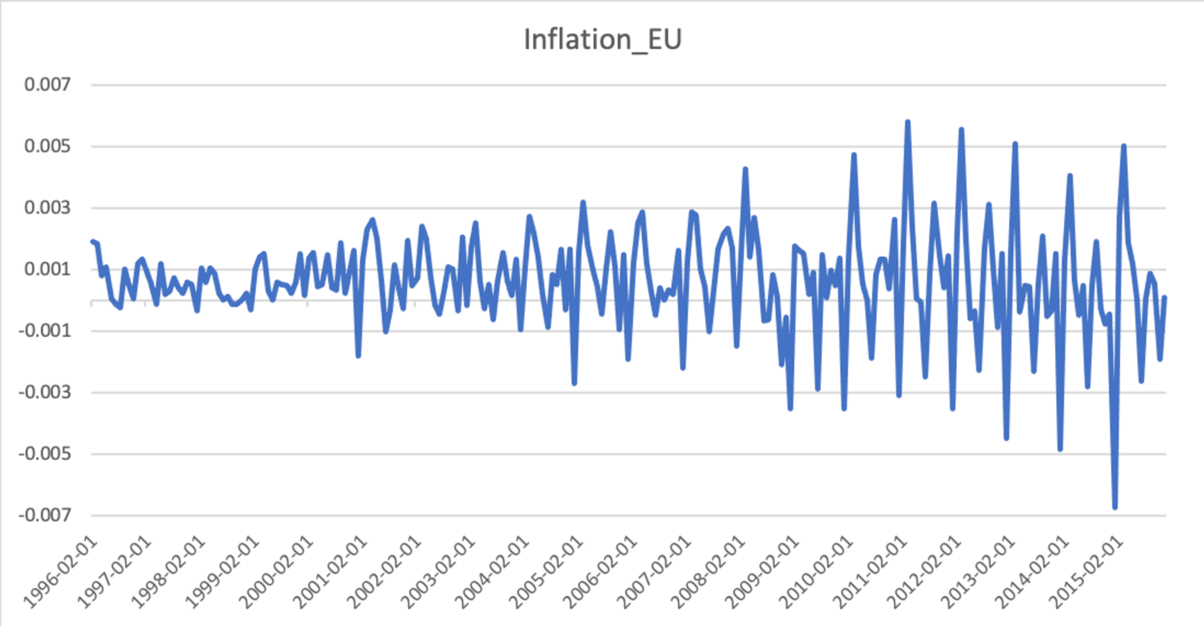


Figure 11

SOURCE:

https://fred.stlouisfed.org/series/CP0000EZ19M086NEST

Unemployment rate EU:

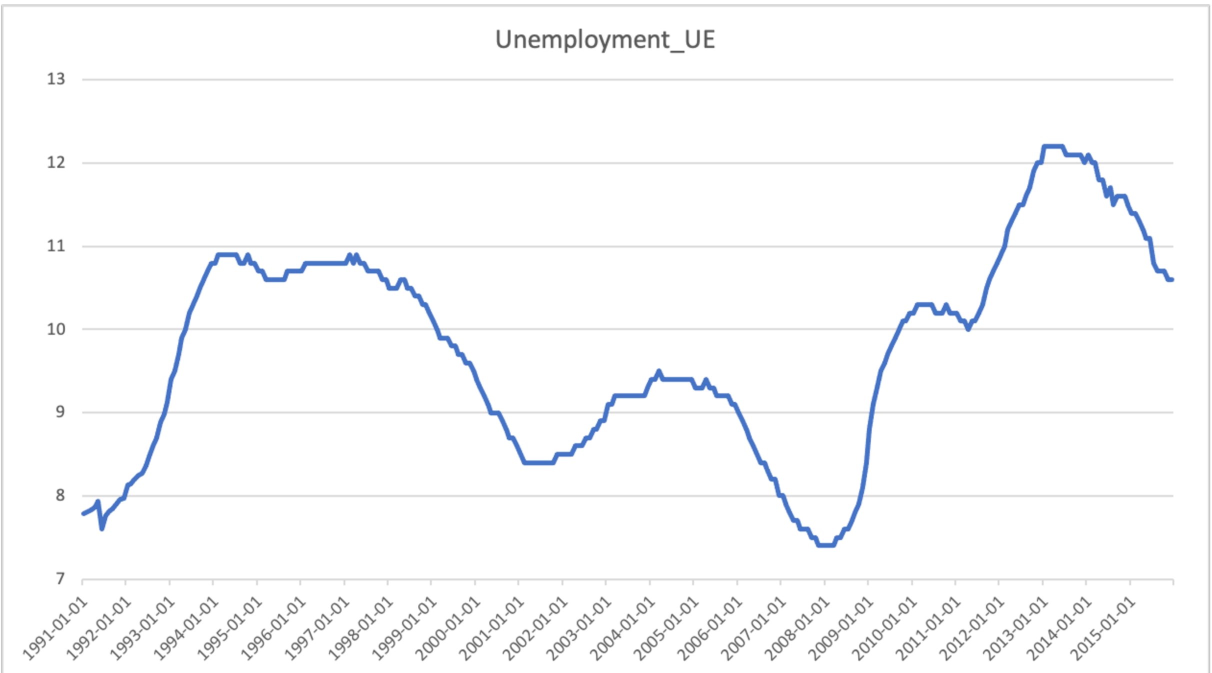


Figure 12

SOURCE:

https://fred.stlouisfed.org/series/LRHUTTTTEZM156S

Chapter 4

4.1 Empirical results

This chapter presents the empirical estimates obtained by implementing our VAR model. I analyze the results using impulse response functions (IRF) and forecast error variance decompositions (FEVD). IRFs are drawn from a VAR (7) with a time horizon of 24 months and 6 lags. In the figures that follow, the solid lines represent the mean impulse response functions, while the confidence intervals correspond to a 68% level of confidence.

Figures 13 and 14 first illustrate the distinct impacts of MPI and INFO shocks. As stated in the initial part of this thesis, MPI represents a contractionary shock. Following its occurrence, there is a response from the FED, resulting in a gradual increase in the short-term interest rate by approximately 25-50 basis points within 3-8 months. Subsequently, there is a negative impulse on inflation and an increase in the unemployment rate. A key variable observed in these figures is the EU 3-month interest rate. As we can observe, according to macroeconomic theory, the impact of the U.S. market is significant enough to prompt the ECB to respond to the increase in interest rates in the U.S. This aligns with the theory of capital outflows, which pose a risk of devaluing the euro if interest rates do not ensure parity returns.

Moving on to Figure 14, there is a crucial result for our research. As suggested by Miranda-Agrippino and Ricco (2021), INFO shocks appear to be positive for the economy. They stimulate economic growth, and in response, the Fed increases the short-term interest rate by approximately 50 basis points within 5 months to slow down the economy. Additionally, the global financial cycle experiences a positive impulse following the manifestation of these shocks, and the results confirm the positive impact on U.S. CPI and a reduction on U.S. unemployment rates. Finally, as our thesis proposes, INFO shocks have the capability to influence EU markets. The ECB responds with a substantial increment in the short-term interest rate, around 75 basis points within 3 months.

In the appendix, Figure 27 will illustrate the FEVD, percentage of the variance in forecasting errors attributed to a specific shock at a given horizon. At a forecast horizon of 3 months, approximately 5 percent of the variation in EU 3-month rate forecast errors can be attributed to the INFO shock. This implies that economic shocks will have a moderate direct impact on 3-month EU rate forecasts in the short term.

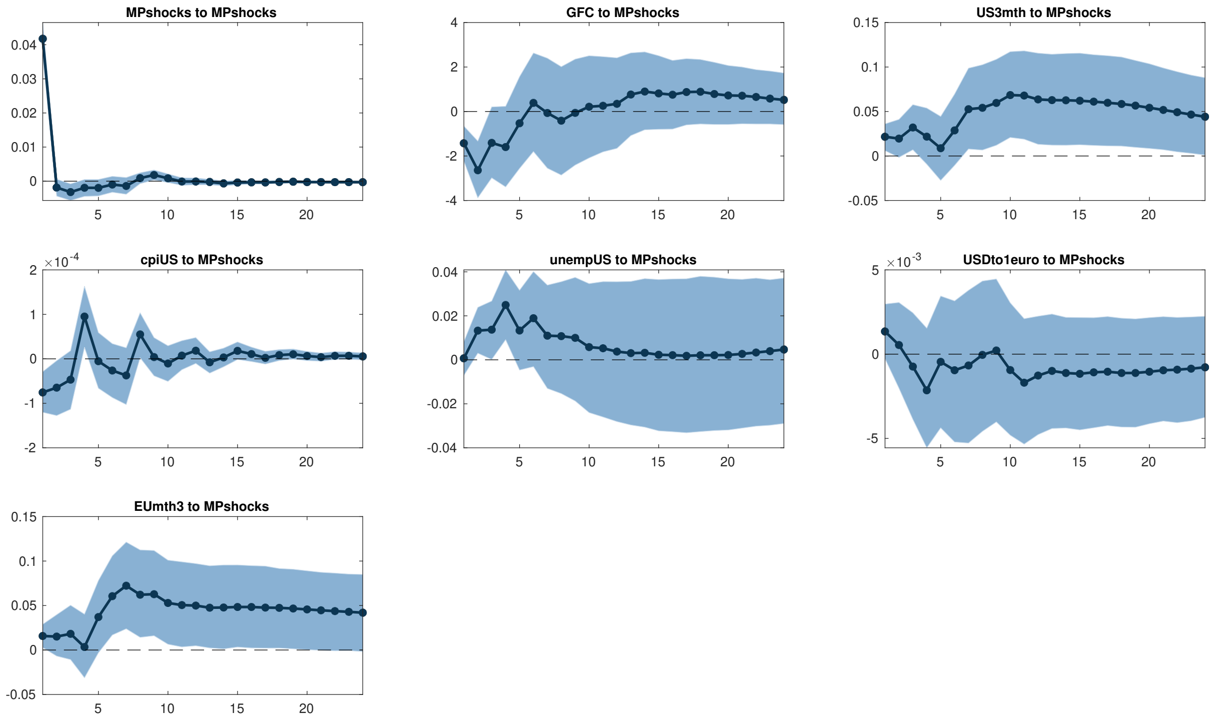


Figure 13: IRF of MPI shock on EU - short term policy rate.

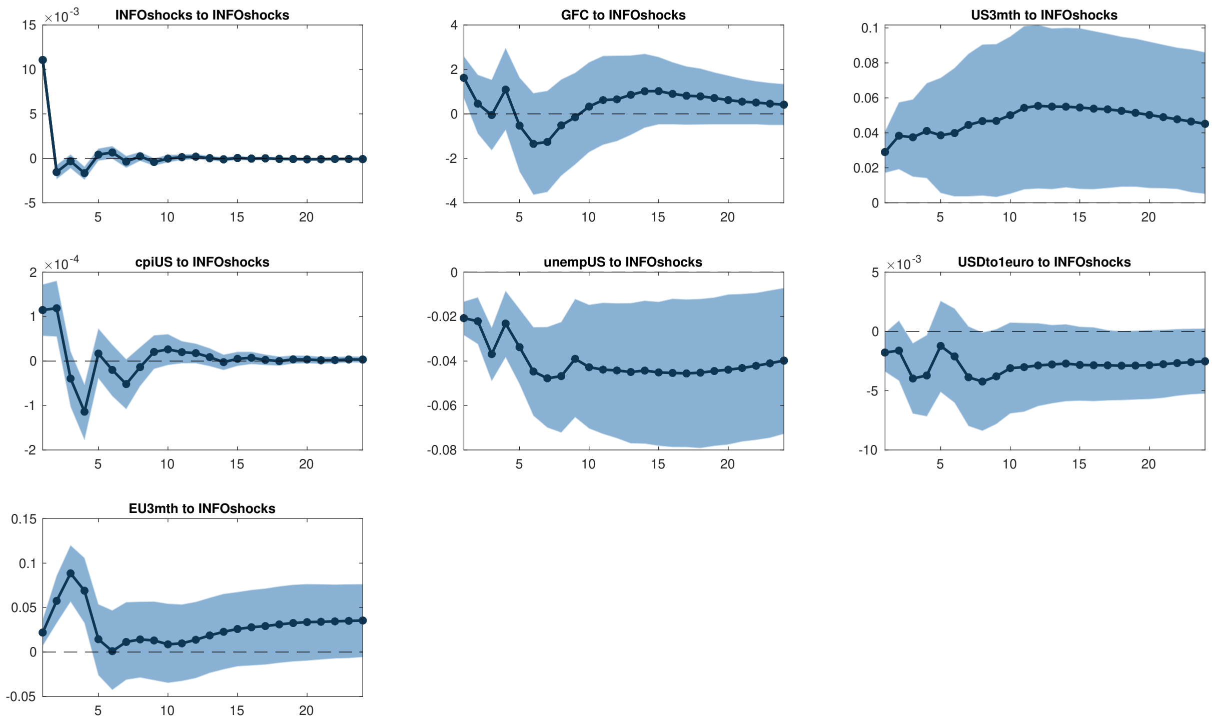


Figure 14: IRF of INFO shock on EU - short term policy rate.

Figures 15 and 16 present the same results as previously discussed but with a different time horizon for interest rates. Specifically, the analysis includes a 10-year interest rate for the EU. This addition allows us to assess whether the impact of IRFs is significant and if it has financial implications. As observed, the results reaffirm the INFO shocks' capacity to influence European markets, which remains evident.

However, in the long run, there is only a 68% probability of observing a response in interest rates on a longer horizon in Europe. It's important to note that the significance of these results falls within the 68% confidence bands.

On the other hand, for MPI shocks, it is evident that their impact is substantial, leading to an increase of 50 basis points in long-term interest rates within 2-3 months of lags.

Regarding the FEVD, the appendix figure 29 demonstrates that INFO shocks account for approximately 8% of the variation in long-term interest rates in Europe.

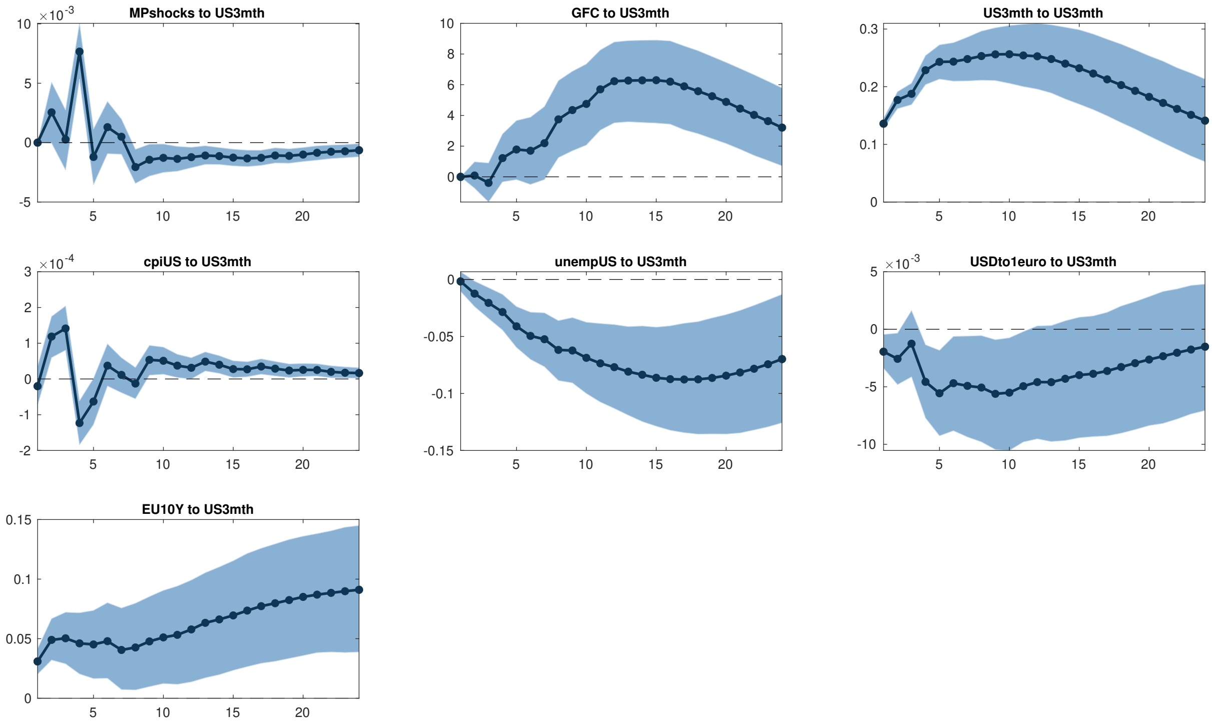


Figure 15: IRF of MPI shock on EU - Long term policy rate.

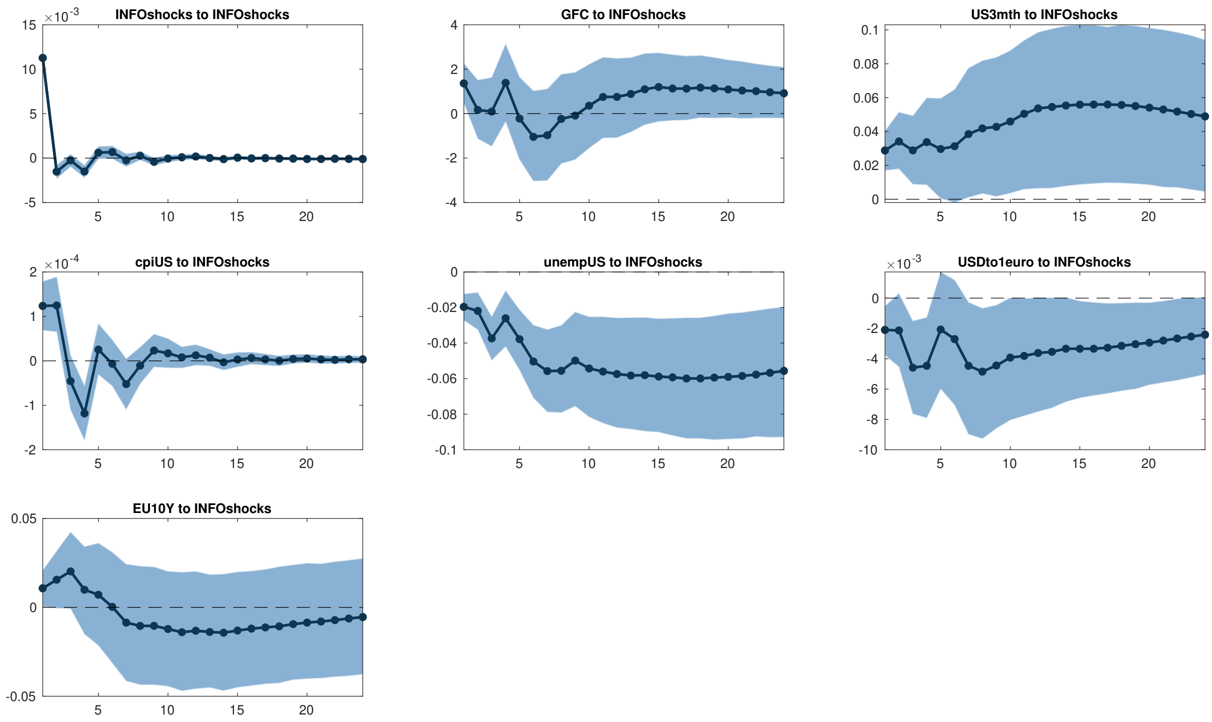


Figure 16: IRF of INFO shock on EU - Long term policy rate.

Lastly, Figures 17 and 18 depict the IRFs of INFO shocks on unemployment and inflation in the EU. Once again, INFO shocks, as suggested by Miranda Agrippino and Ricco (2021), reinforce the idea that FOMC communications occasionally provide a positive impulse to the markets.

This further aids in addressing the primary question of this thesis. It appears to be validated that INFO shocks initially impact US macroeconomic variables and that these impacts are robust enough to be transmitted to the EU. The results support the significance of the data, with confidence bands reaching up to 68%.

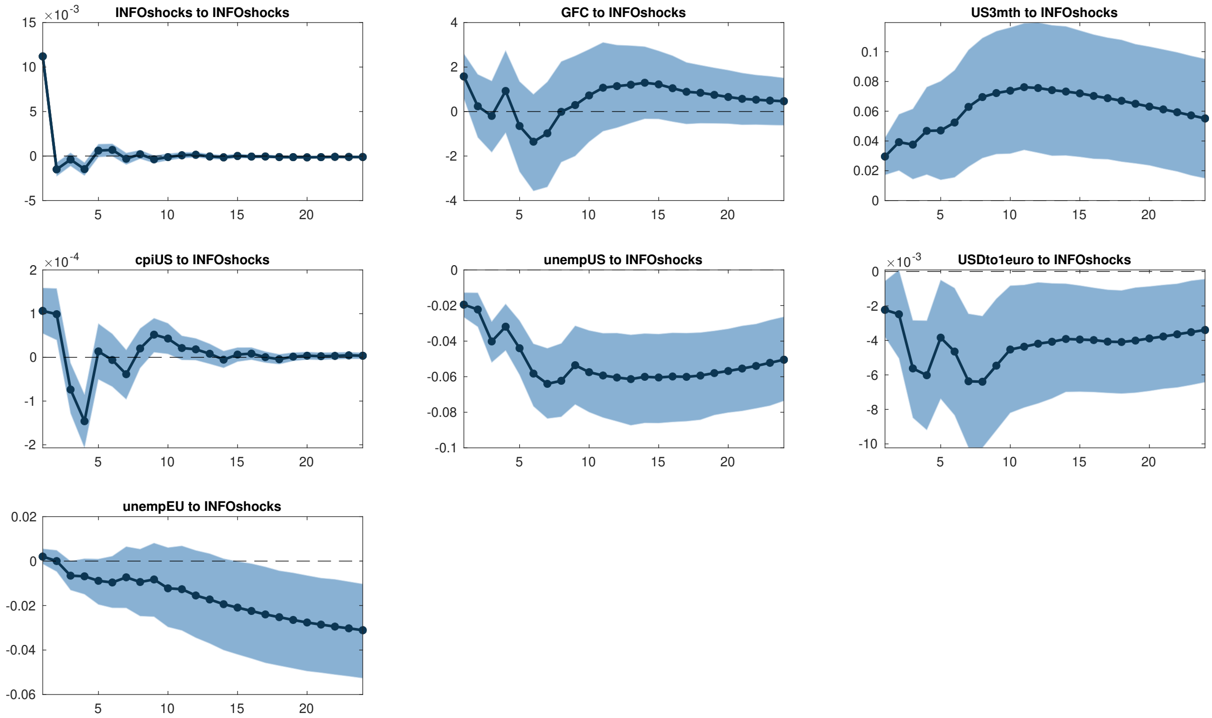


Figure 17: IRF of INFO shock on EU – Unemployment.

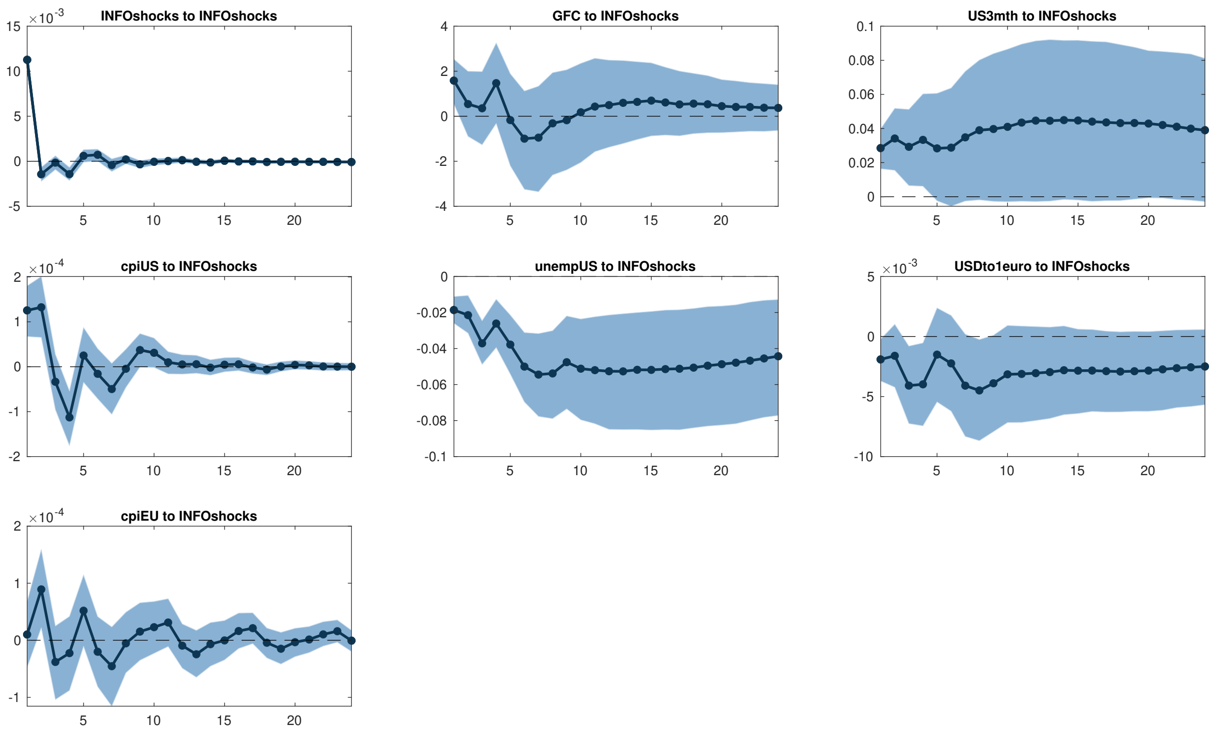


Figure 18: IRF of INFO shock on EU – Consumer price index.

4.2 Robustness Checks

There are various methods to evaluate the robustness of a VAR model. In this context, I will employ one of the most prevalent approaches, which involves assessing the model's performance using alternative variables. Specifically, I will incorporate a financial distress indicator (baa10Y) and the industrial production index for both the US and EU as alternatives. Additionally, I will examine the impact of other key economic variables such as unemployment, short and long-term interest rates, and inflation on the financial distress indicator (baa10Y).

Figures 19 and 20 display the IRFs of both MPI and INFO shocks, using the Baa10Y as a proxy for the financial distress indicator. The results confirm the restrictive impacts associated with MPI shocks. Within a short timeframe of 1 month, the Baa10Y indicator experiences a 2% increase. Given that this indicator reflects the spread between corporate and government bonds, such an increase indicates a rise in financial distress.

Similar confirmation is observed for INFO shocks, with their IRFs demonstrating a reduction in the spread between corporate and government bonds, signaling a decrease in financial distress. Once again, these robustness checks reaffirm the ability of INFO shocks in generating a notably significant impact on the 3-month interest rate. In terms of the FEVD, at 3-month horizon around 8 percent of the variability in EU short term interest rate can be attributed to the INFO shock. This suggests that economic shocks will exert a moderate and direct influence on EU policy rate.

Furthermore, I introduce additional variables, including EU industrial production and US industrial production, to assess the impacts of the shocks. As shown in Figure 21 and 22, INFO shocks once again exhibit their typical positive effects on the markets, resulting in a significant increase in industrial production. This reaffirms the ability of INFO shocks to influence the European market, as the significant impacts extend to the European context.

Figures 23, 24, and 25 continue this analysis, demonstrating the impact on EU variables, such as the growth of long-term interest rates, the reduction in unemployment, and the increase in inflation

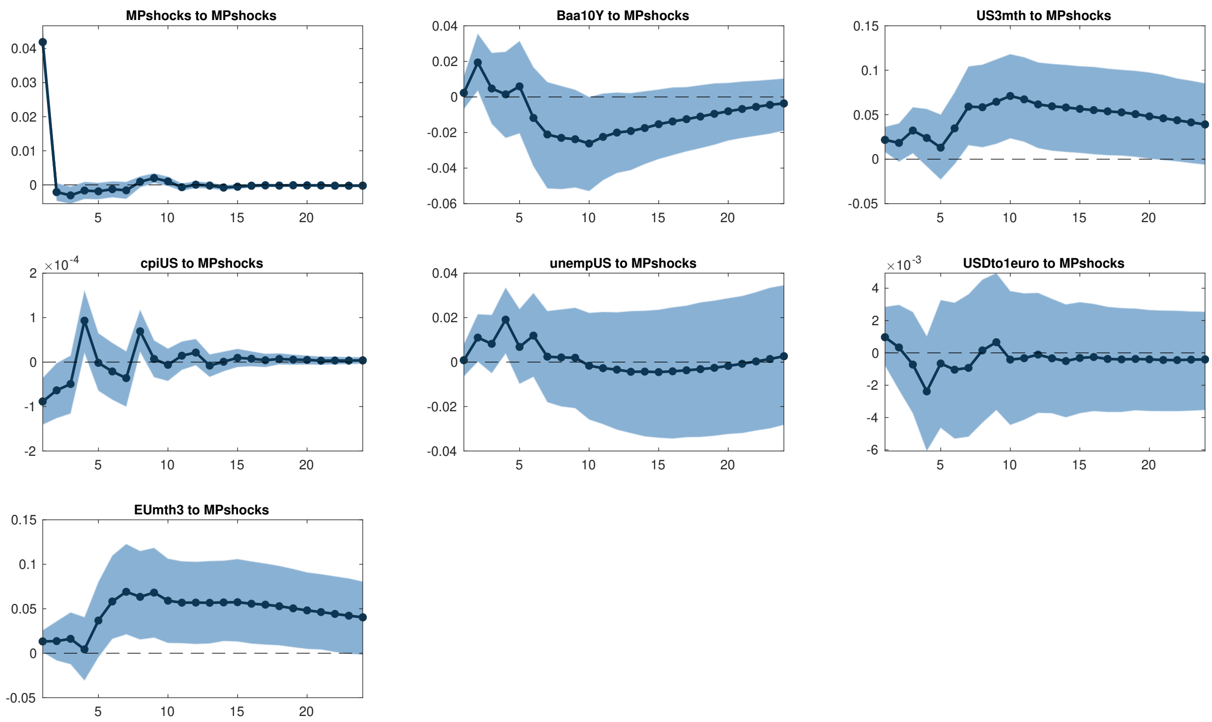


Figure 19: IRF of MPI shock on EU – - short term policy rate. Robustness check include Baa10Y

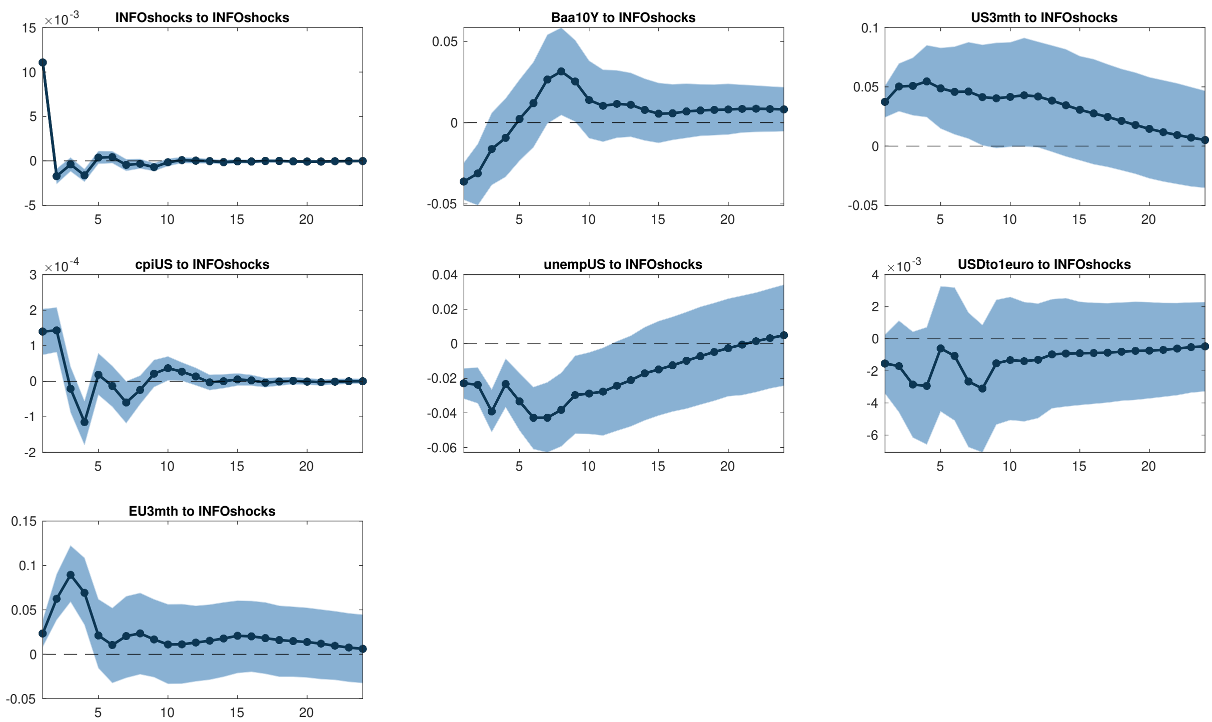


Figure 20: IRF of INFO shock on EU – - short term policy rate. Robustness check include Baa10Y

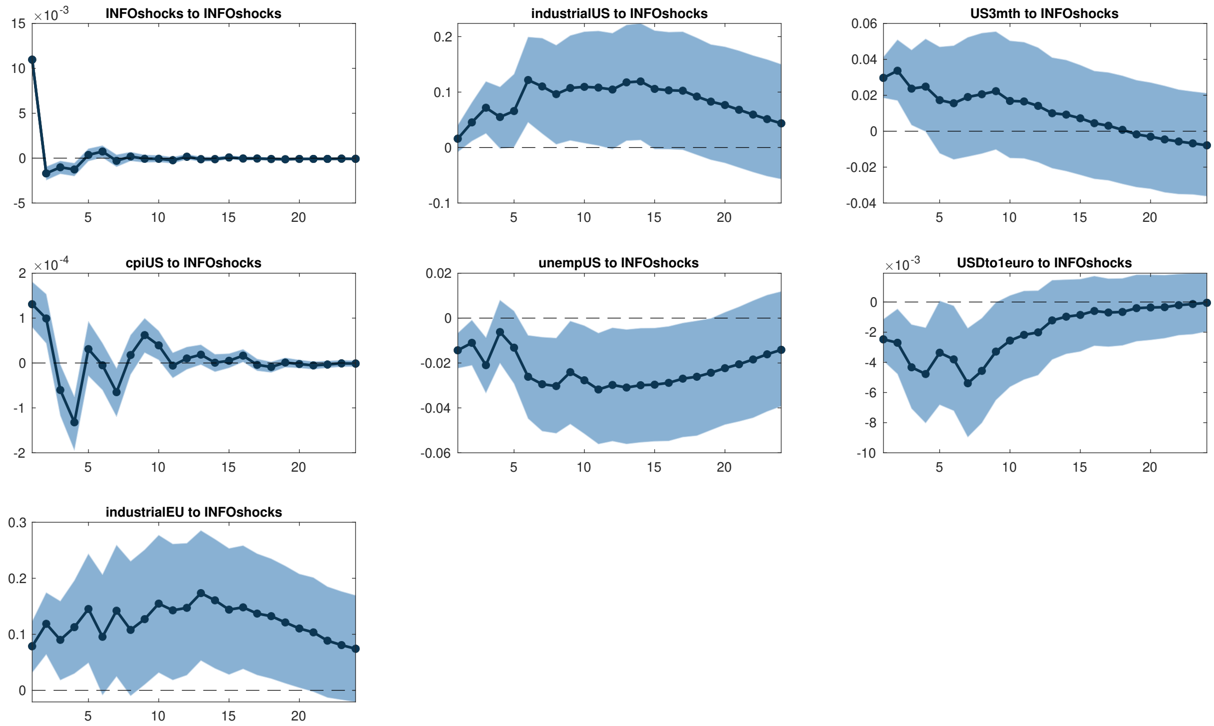


Figure 21: IRF of INFO shock on EU – - Industrial production. Robustness check include US industrial production.

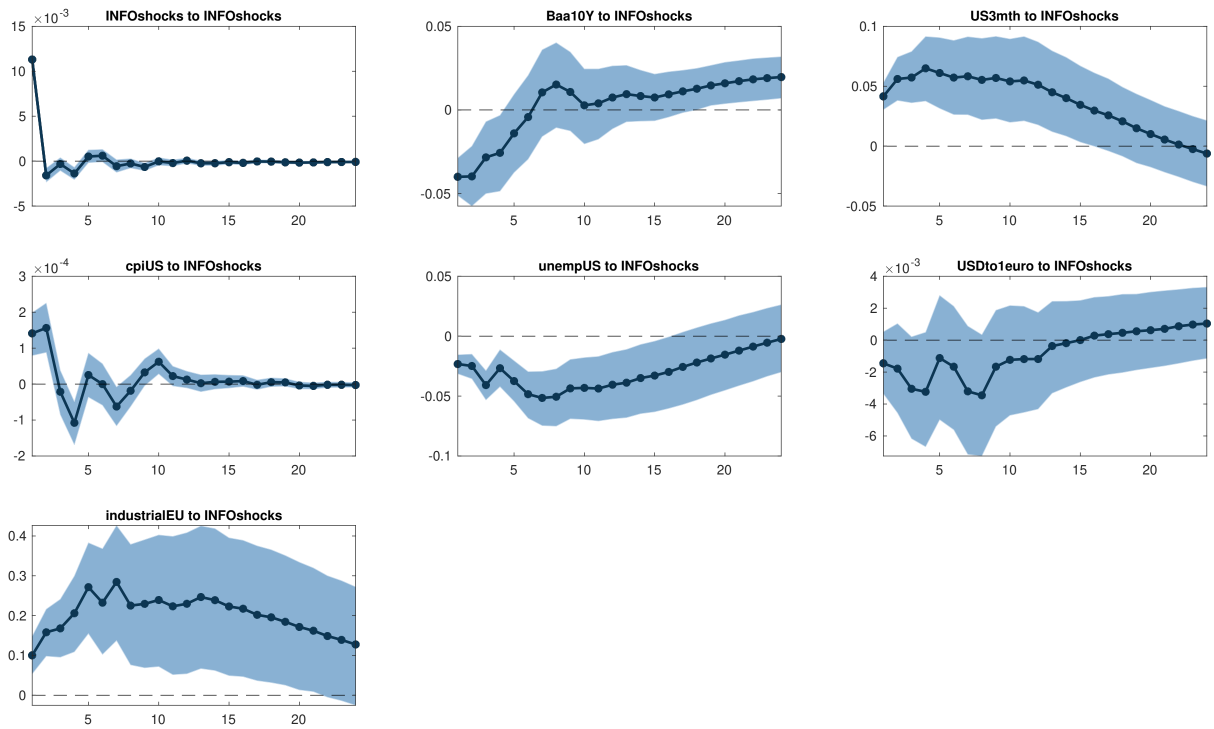


Figure 22: IRF of INFO shock on EU – - Industrial production. Robustness check include Baa10Y.

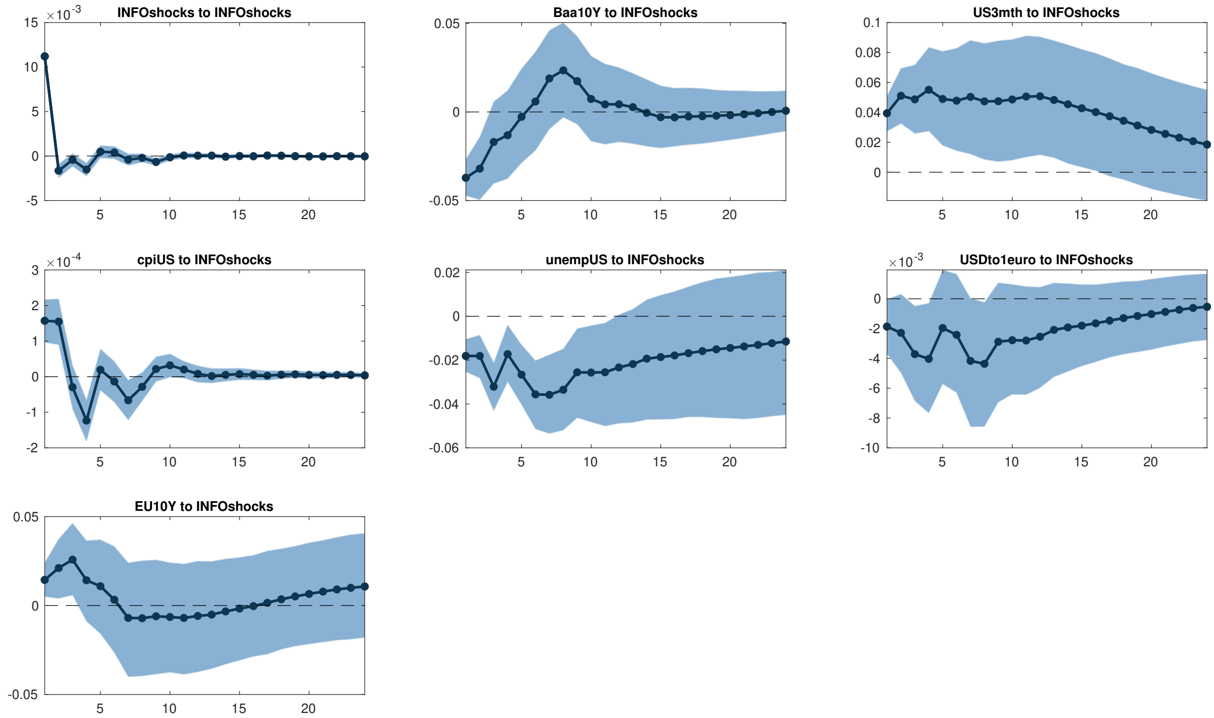


Figure 23: IRF of INFO shock on EU – - Long term policy rate. Robustness check include Baa10Y.

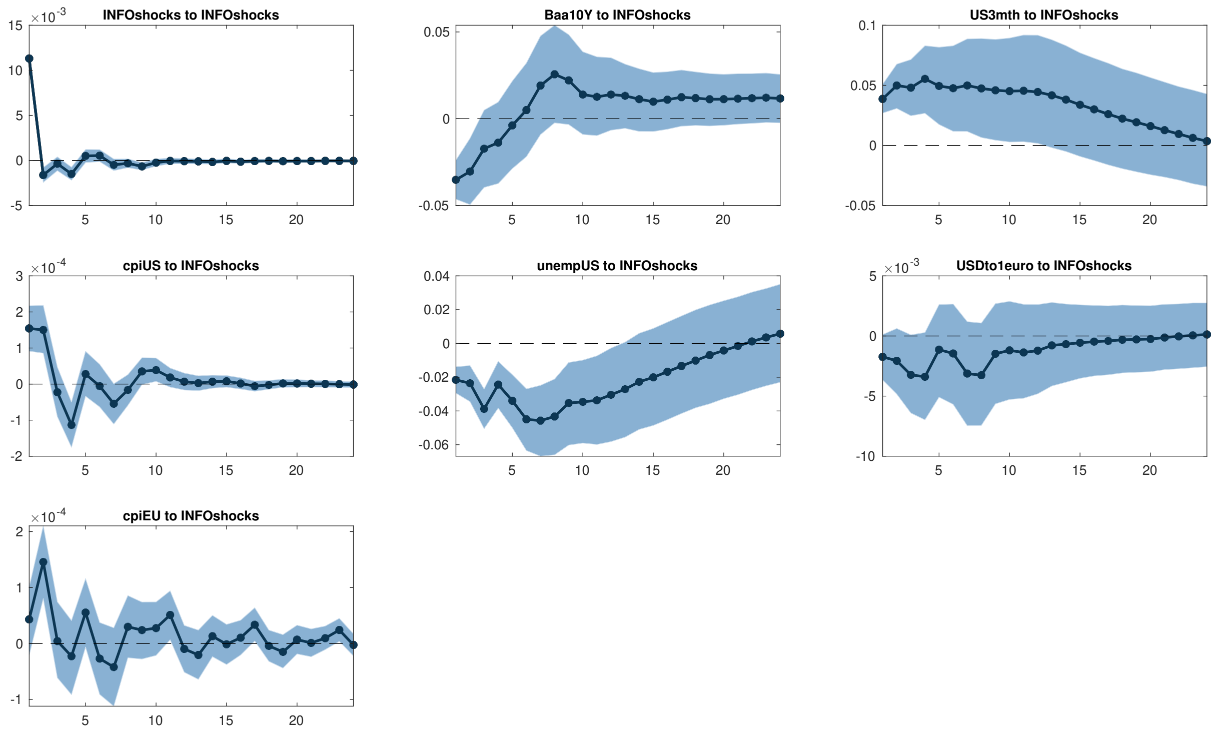


Figure 24: IRF of INFO shock on EU – - Consumer price index. Robustness check include Baa10Y.

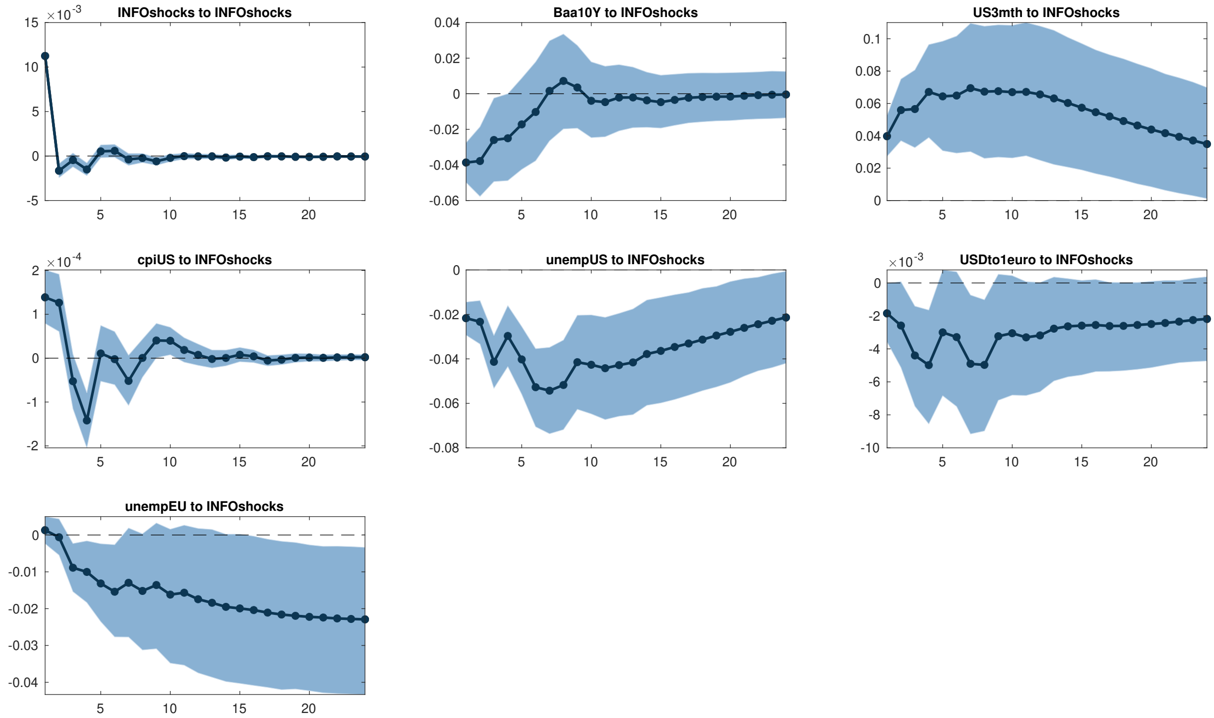


Figure 25: IRF of INFO shock on EU – - unemployment rate. Robustness check include Baa10Y.

4.3 Conclusions

In this chapter, the presentation focuses on the results of the 7-variable VAR model, the forecast error variance decomposition analyses, and robustness checks aimed at better interpreting the consistency of the findings. Two key conclusions emerge from the analysis.

First, INFO shocks have a substantial impact on several essential macroeconomic variables in the EU, particularly inflation, unemployment, short-term, and long-term interest rates. These impulses, as denoted in the FEVD present in the appendix, are derived from the INFO shocks themselves and the GFC indicator, underscoring the strength of the US market in influencing international financial conditions.

Second, both MPI and INFO shocks, originating from FOMC meetings and translated into data by Miranda-Agrippino and Ricco (2021), exhibit distinct and tangible impacts on the market. The former represents a positive shock, bringing favorable economic information, while the latter signifies a negative shock associated with market downturns.

Furthermore, the results remain robust even when re-estimating our VAR model with different variables. In this case, the Baa10Y financial distress indicator, as well as US and EU industrial production, align with our observations and validate the existence of INFO shocks and their capacity to influence economic agents' expectations. I acknowledge that using a 68% confidence level is justified due to the complexity of the model, which includes 7 coefficients, 6 lags, and a linearized quadratic trend. This complexity can result in some loss of information during the estimation of IRFs, making it challenging to achieve high levels of significance. Nevertheless, our approach is consistent with Stock and Watson (2001), who employ a 66% confidence interval for validating the Taylor rule.

Conclusion

The main questions addressed in this thesis were as follows: What are INFO shocks, and how do they differ from classical MPI shocks? Do FOMC communications have the capacity to generate real effects on the markets, particularly in Europe?

To investigate these questions, I leveraged the results of MPI and INFO shocks as presented in the work of Miranda - Agrippino and Ricco (2021) and Jarocinski - Karadi (2020). They employed analytical tools, as detailed in the first chapter, to transform FOMC communications into a dataset, which was subsequently used to analyze the IRFs and FEVD within the VAR models applied in this research.

In addressing the main questions, I employed a recursive VAR as a proxy for a SVAR, as discussed in Chapter 2. This approach allowed me to bypass the complexities associated with directly estimating an SVAR, following the work of Plagborg-Møller and Wolf (2021). The VAR model consistently included 7 variables and was lagged for 6 periods, with a time horizon extending to 24 months.

The key variables included INFO and MPI shocks, in addition to standard U.S. macroeconomic indicators such as inflation, unemployment, and short and long-term interest rates. These were used to measure the impacts of the shocks within the U.S. market. Additionally, I incorporated standard EU macroeconomic indicators, including inflation, unemployment, and short and long-term interest rates, to assess the intensity of the shocks' influence on foreign markets. I also utilized two proxies for financial conditions: a global financial cycle (GFC) indicator to gain insights into broader financial effects and a spread indicator that takes into account Moody's Seasoned Baa Corporate Bond and the 10-Year Treasury Constant Maturity (BC10Y). The latter serves as a proxy for assessing financial distress within the markets.

Furthermore, I introduced U.S. and EU industrial production indicators to assess whether the data continued to support the observed impacts on the standard key macroeconomic variables.

The results presented in Chapter 4 affirmed the presence of distinct types of shocks between INFO and MPI and demonstrated that INFO shocks are sufficiently robust to transmit their influence to European markets. So, in conclusion, I can provide a positive answer to the main question of this thesis, thereby confirming the existence and differentiation in the impacts of these shocks.

Bibliography

Admati, A. R. (1985). A noisy rational expectations equilibrium for multi-asset securities markets. *Econometrica: Journal of the Econometric Society*, 629-657.

Bernanke, B. S. (1986). Alternative explanations of the money-income correlation.

Bernanke, B. S., & Blinder, A. S. (1992). VThe Federal Funds Rate and the Channels of Monetary TransmissionV The American Economic Review.

Blanchard, O. J., & Watson, M. W. (1986). Are business cycles all alike?. In *The American business cycle: Continuity and change* (pp. 123-180). University of Chicago Press.

Caldara, D., & Herbst, E. (2019). Monetary policy, real activity, and credit spreads: Evidence from Bayesian proxy SVARs. *American Economic Journal: Macroeconomics*, *11*(1), 157-192.

Coibion, O., & Gorodnichenko, Y. (2015). Information rigidity and the expectations formation process: A simple framework and new facts. *American Economic Review*, *105*(8), 2644-2678.

Cooley, T. F., & LeRoy, S. F. (1985). Atheoretical macroeconometrics: a critique. *Journal of Monetary Economics*, *16*(3), 283-308.

Gertler, M., & Karadi, P. (2015). Monetary policy surprises, credit costs, and economic activity. *American Economic Journal: Macroeconomics*, *7*(1), 44-76.

Gürkaynak, R. S., Sack, B., & Swanson, E. (2005). The sensitivity of long-term interest rates to economic news: Evidence and implications for macroeconomic models. *American economic review*, *95*(1), 425-436.

Hellwig, M. F. (1980). On the aggregation of information in competitive markets. *Journal of economic theory*, *22*(3), 477-498.

Hubert, P., & Maule, B. (2021). Policy and Macro Signals from Central Bank Announcements. *International Journal of Central Banking*, *17*(2), 255-296.

Jarociński, M., & Karadi, P. (2020). Deconstructing monetary policy surprises—the role of information shocks. *American Economic Journal: Macroeconomics*, *12*(2), 1-43.

Kilian, L., & Lütkepohl, H. (2017). *Structural vector autoregressive analysis*. Cambridge University Press.

Kilian, L., & Zha, T. (2002). Quantifying the uncertainty about the half-life of deviations from PPP. *Journal of Applied Econometrics*, *17*(2), 107-125.

Mayr, J., & Ulbricht, D. (2007). *VAR model averaging for multi-step forecasting* (No. 48). Ifo Working Paper.

McCracken, M. W., & Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, *34*(4), 574-589.

Melosi, L. (2017). Signalling effects of monetary policy. *The Review of Economic Studies*, *84*(2), 853-884.

Mertens, K., & Ravn, M. O. (2013). The dynamic effects of personal and corporate income tax changes in the United States. *American economic review*, *103*(4), 1212-1247.

Nakamura, E., & Steinsson, J. (2018). High-frequency identification of monetary non-neutrality: the information effect. *The Quarterly Journal of Economics*, *133*(3), 1283-1330.

Plagborg-Møller, M., & Wolf, C. K. (2021). Local projections and VARs estimate the same impulse responses. *Econometrica*, *89*(2), 955-980.

Ramey, V. A. (2016). Macroeconomic shocks and their propagation. *Handbook of macroeconomics*, *2*, 71-162.

Romer, C. D., & Romer, D. H. (2000). Federal Reserve information and the behavior of interest rates. *American economic review*, *90*(3), 429-457.

Romer, C. D., & Romer, D. H. (2004). A new measure of monetary shocks: Derivation and implications. *American economic review*, *94*(4), 1055-1084.

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, 1-48.

Sims, C. A. (1986). Are forecasting models usable for policy analysis?. *Quarterly Review*, *10*(Win), 2-16.

Stock, J. H., & Watson, M. W. (2001). Vector autoregressions. *Journal of Economic perspectives*, *15*(4), 101-115.

Stock, J. H., & Watson, M. W. (2012). *Disentangling the Channels of the 2007-2009 Recession* (No. w18094). National Bureau of Economic Research.

Tang, Jenny. 2013. "Uncertainty and the Signaling Channel of Monetary Policy." Federal Reserve Bank of Boston Research Department Working Paper 15-8.

APPENDIX

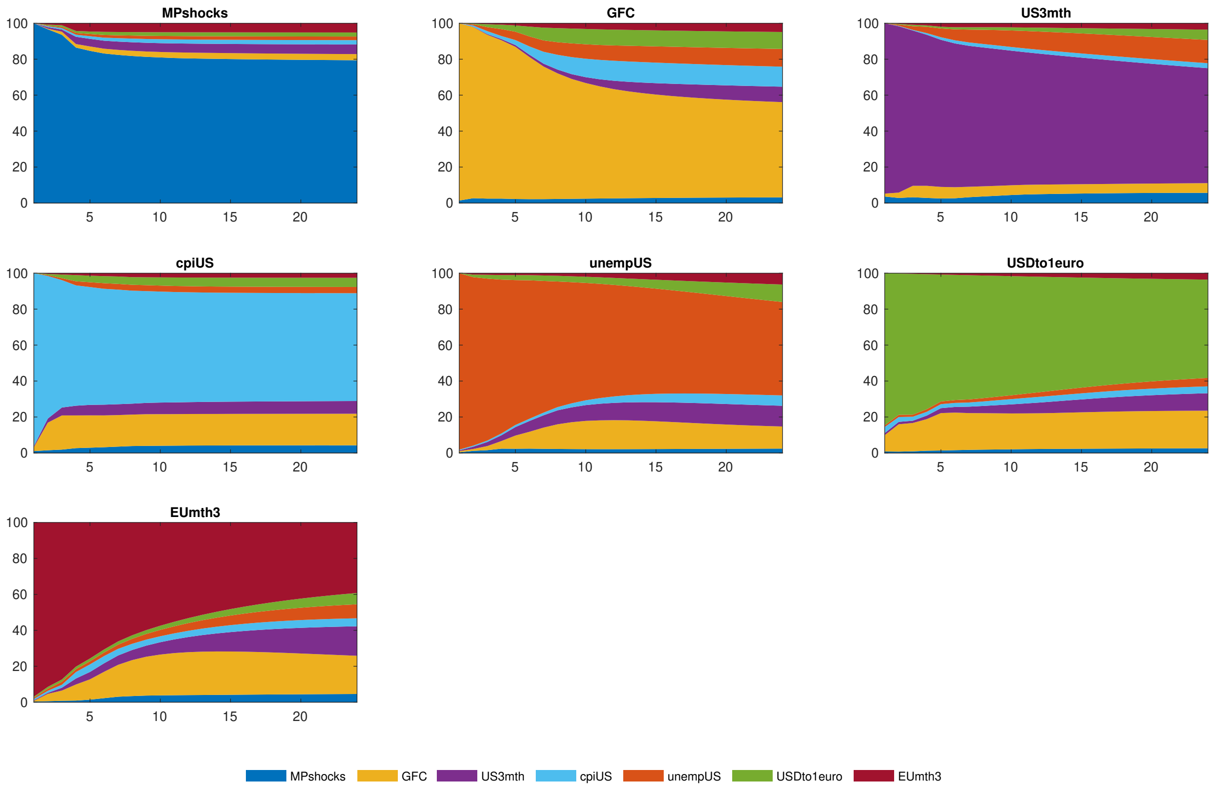


Figure 26: FEVD of MPI shock on EU – - short term policy rate

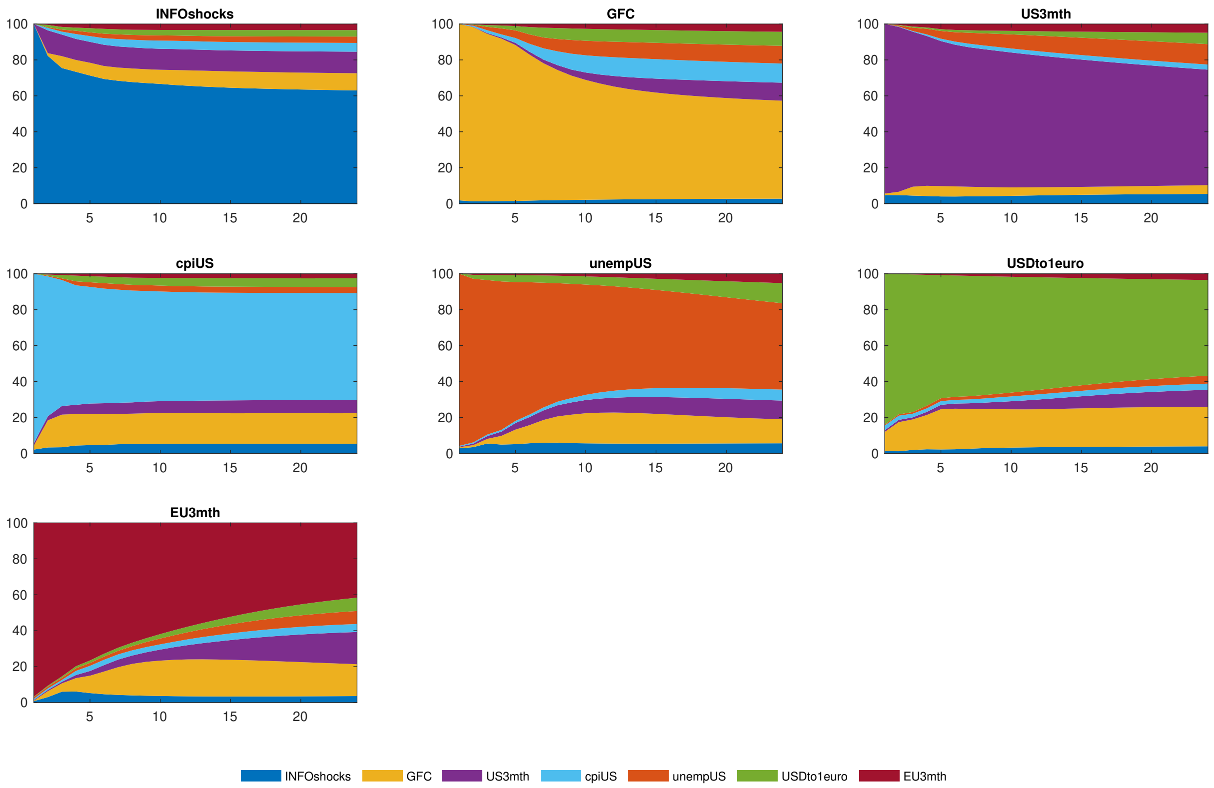


Figure 27: FEVD of INFO shock on EU – - short term policy rate.

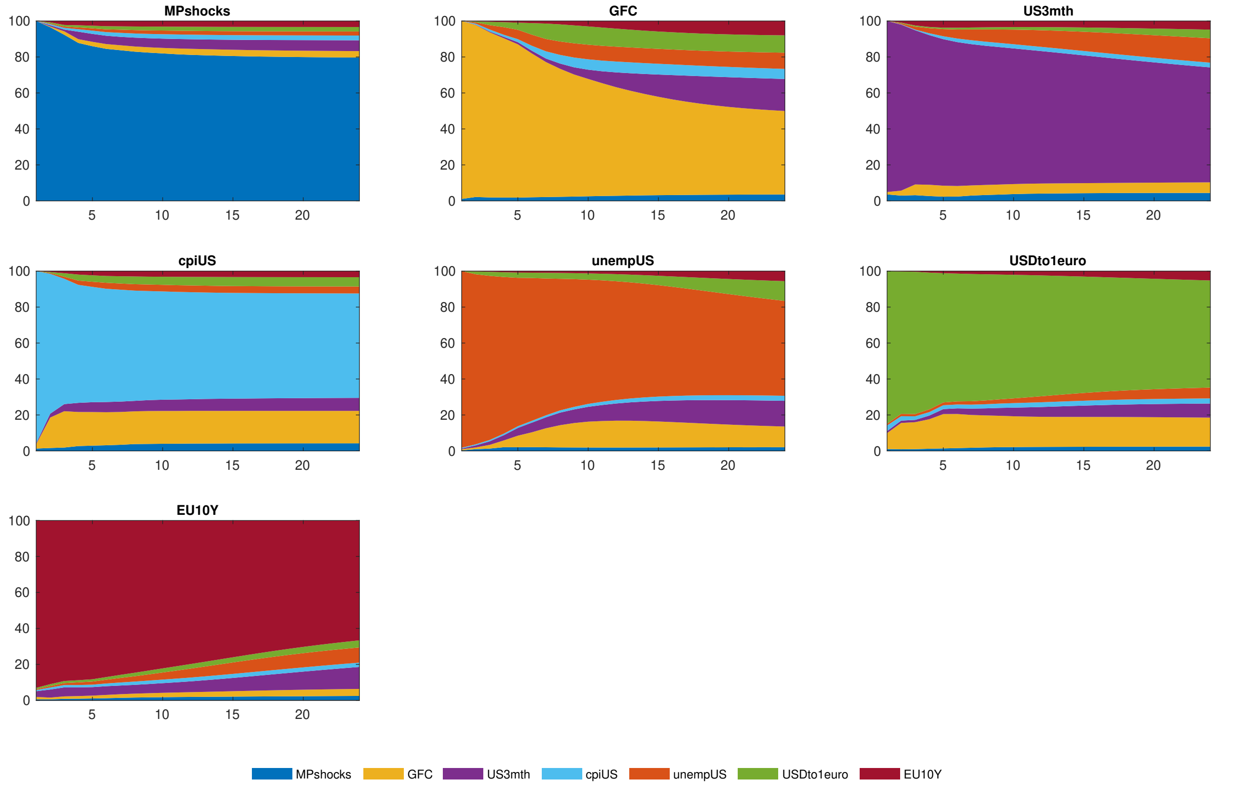


Figure 28: FEVD of MPI shock on EU – - Long term policy rate.

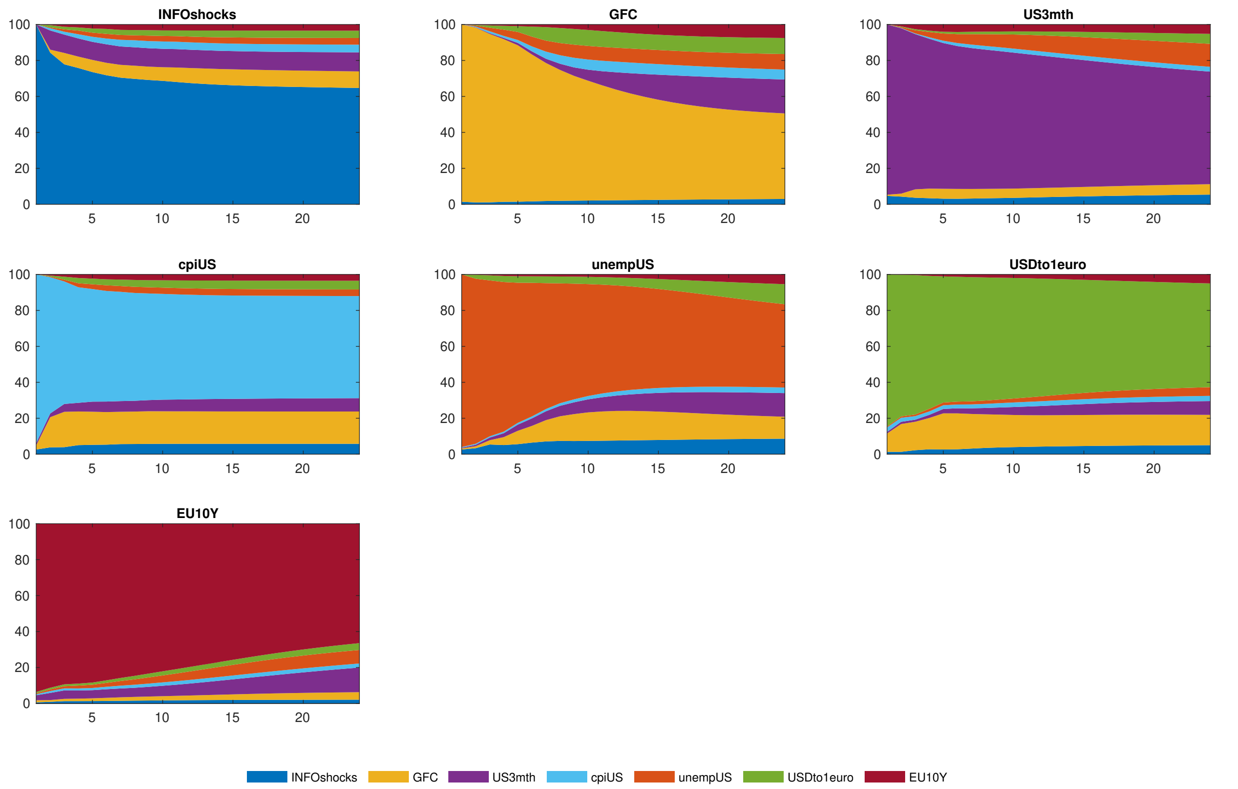


Figure 29: FEVD of INFO shock on EU – - Long term policy rate.

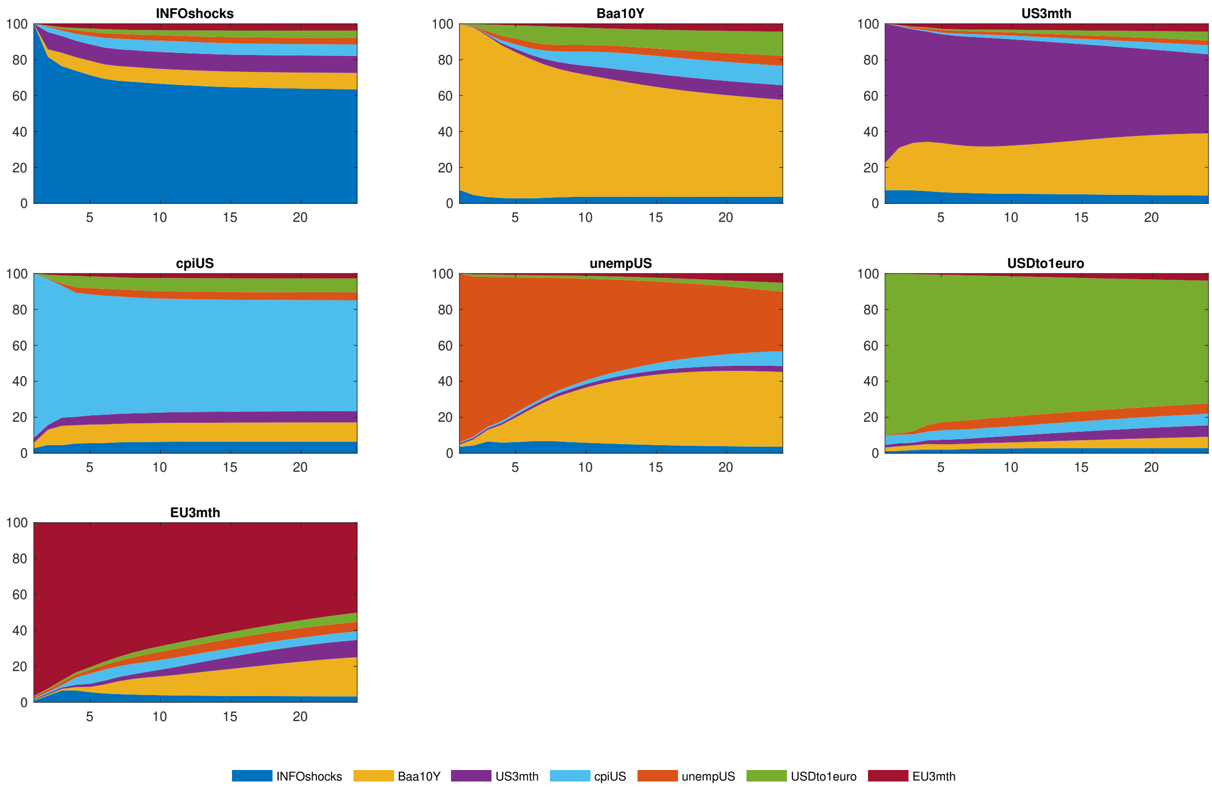


Figure 30: FEVD of MPI shock on EU – - Short term policy rate. - Robustness check include Baa10Y

MATLAB codes

In this section, I will present the MATLAB codes employed for obtaining the results of this empirical research. Specifically, the following MATLAB files will be provided:

1. The first file contains comprehensive codes used to plot Impulse Response Functions (IRFs), generate tables for Forecast Error Variance Decomposition (FEVD), and create the figures described in Chapter 4.
2. The second file includes the codes necessary to perform AIC (Akaike Information Criterion) and Schwarz Bayesian Information Criterion (BIC) tests.
3. The third file is employed to set up trend models, such as a constant and quadratic trend.
4. The forth file is used for the VAR estimations as described in Chapter 2.

Please note that I utilized the third version of the VAR toolbox developed by Ambrogio Cesa-Bianchi (2020). The VAR toolbox can be freely downloaded at:

https://github.com/ambropo/VAR-Toolbox/tree/main/v3dot0/Primer

Modifications have been applied to these code sets in order to generate the results mentioned.

```
% Replication of the trivariate VAR in Stock and Watson (2001, JEP).
% Figure 1 and Table 1.B.
%===========================================================================

clear all;
clear session;
close all;
clc
warning off all

% Load data

[xlsdata, xlstext] = xlsread('SW2001_Data.xlsx','Sheet1');
X = xlsdata;
dates = xlstext(3:end,1);
vnames_long = xlstext(1,2:end);
vnames = xlstext(2,2:end);
nvar = length(vnames);
data   = Num2NaN(xlsdata);
% Store variables in the structure DATA
for ii=1:length(vnames)
    DATA.(vnames{ii}) = data(:,ii);

end

% Convert the first date to numeric
year = str2double(xlstext{3,1}(1:4));
quarter = str2double(xlstext{3,1}(6));
% Observations
nobs = size(data,1);
%% VAR ESTIMATION
%===========================================================================
% Set deterministics for the VAR
det = 1;
% Set number of nlags
nlags = 4;
% Estimate VAR
[VAR, VARopt] = VARmodel(X,nlags,det);
% Print estimation on screen
VARopt.vnames = vnames;
[TABLE, beta] = VARprint(VAR,VARopt,2);
%% COMPUTE IR AND VD
%===========================================================================
% Set options some options for IRF calculation
VARopt.nsteps = 24;
VARopt.ident = 'short';
VARopt.vnames = vnames_long;
VARopt.FigSize = [26,12];
% Compute IRF
[IRF, VAR] = VARir(VAR,VARopt);
% Compute error bands
[IRinf,IRsup,IRmed,IRbar] = VARirband(VAR,VARopt);
% Plot
VARirplot(IRbar,VARopt,IRinf,IRsup);

% Compute VD

[VD, VAR] = VARvd(VAR,VARopt);
% Compute VD error bands
[VDinf,VDsup,VDmed,VDbar] = VARvdband(VAR,VARopt);
```

```
% Plot VD

VARvdplot(VDbar,VARopt);

%% Print Table 1.B on screen
%=========================================================================
% Retrieve Forecast Error Variance Decomposition
FEVD_Table(1, :) = VD(1,:,1);
FEVD_Table(2, :) = VD(4,:,1);
FEVD_Table(3, :) = VD(8,:,1);
FEVD_Table(4, :) = VD(12,:,1);
FEVD_Table(5, :) = VD(1,:,2);
FEVD_Table(6, :) = VD(4,:,2);
FEVD_Table(7, :) = VD(8,:,2);
FEVD_Table(8, :) = VD(12,:,2);
FEVD_Table(9, :) = VD(1,:,3);
FEVD_Table(10,:) = VD(4,:,3);
FEVD_Table(11,:) = VD(8,:,3);
FEVD_Table(12,:) = VD(12,:,3);
% Print on screen
disp(' ')
disp('Variance Decomposition of Inflation (t=1,4,8,12)')
disp('---------------------------------------------------')
mprint(FEVD_Table(1:4,:));
disp('Variance Decomposition of Unemployment (t=1,4,8,12)')
disp('---------------------------------------------------')
mprint(FEVD_Table(5:8,:));
disp('Variance Decomposition of Fed Funds (t=1,4,8,12)')
disp('---------------------------------------------------')
mprint(FEVD_Table(9:12,:));
```

```
function [AIC, SBC, logL] = VARlag(ENDO,maxlag,const,EXOG,lag_ex)
% =======================================================================
% Determine VAR lag length with Akaike (AIC) and Schwarz Bayesian
% Criterion (SBC)criterion.
% =======================================================================
% [AIC, SBC, logL] = VARlag(ENDO,maxlag,const,EXOG,lag_ex)
% -----------------------------------------------------------------------
% INPUT
%
%
%
% -----------------------------------------------------------------------
% OPTIONAL INPUT
- ENDO: an (nobs x nvar) matrix of endogenous variables.
- maxlag: the maximum lag length over which Akaike information
criterion is computed
```

- `%    - const: 0 no constant; 1 constant ; 2 constant and trend;`
-
- `%            3 constant and trend^2; [dflt = 1]`
-
- `%    - EXOG: optional matrix of variables (nobs x nvar_ex)`
-
- `%    - nlag_ex: number of lags for exogeonus variables (dflt = 0)`
- `% -----------------------------------------------------------------------`
- `% OUTPUT`
-

- `%    - AIC: preferred lag lenghth according to AIC`
-
- `%    - SBC: preferred lag lenghth according to SBC`
-
- `%    - logL: vector [maxlag x 1] of loglikelihood`
- `% -----------------------------------------------------------------------`
- `% EXAMPLE`
-

- `%  x=[12;34;56;78;910];`
- `%    OUT = VARmakelags(x,2)`
- `% =======================================================================`
- `% VAR Toolbox 3.0`
- `% Ambrogio Cesa-Bianchi`
- `% ambrogiocesabianchi@gmail.com`
- `% March 2012. Updated November 2020`
- `% -----------------------------------------------------------------------`
- `%% Check inputs`
- `%===========================================================`
- `[nobs, ~] = size(ENDO);`
- `% Check if ther are constant, trend, both, or none`
- `if ~exist('const','var')`
- `    const = 1;`
- `end`
- `% Check if there are exogenous variables`
- `if exist('EXOG','var')`
- `    [nobs2, num_ex] = size(EXOG);`
- `    % Check that ENDO and EXOG are conformable`
- `    if (nobs2 ~= nobs)`
- `        error('var: nobs in EXOG-matrix not the same as y-matrix');`
- `    end`

- clear nobs2
- else

```
num_ex = 0;

end

% Check if there is lag order of EXOG, otherwise set it to 0
if ~exist('lag_ex','var')
    lag_ex = 0;

end

% number of exogenous variables per equation
nvar_ex = num_ex*(lag_ex+1);
%% Compute log likelihood and Akaike criterion
%=========================================================
logL = zeros(maxlag,1);
AIC  = zeros(maxlag,1);

SBC  = zeros(maxlag,1);
for i=1:maxlag
    X = ENDO(maxlag+1-i:end,:);
    aux = VARmodel(X,i,const);
    if nvar_ex>0
        Y = EXOG(maxlag+1-i:end,:);
        aux = VARmodel(X,i,const,Y,lag_ex);

end

    NOBSadj = aux.nobs;
    NOBS = aux.nobs + i;
    NVAR = aux.nvar;
    NTOTCOEFF = aux.ntotcoeff;
    RES = aux.resid;
    % VCV of the residuals (use dof adjusted denominator)
    SIGMA = (1/(NOBSadj)).*(RES)'*(RES);
    % Log-likelihood
    logL(i) = -(NOBS/2)* (NVAR*(1+log(2*pi)) + log(det(SIGMA)));
    % AIC: �2*LogL/T + 2*n/T, where n is total number of parameters (ie,
NVAR*NTOTCOEFF)
    AIC(i) = -2*(logL(i)/NOBS) + 2*(NVAR*NTOTCOEFF)/NOBS;
    % SBC: �2*LogL/T + n*log(T)/T
    SBC(i) = -2*(logL(i)/NOBS) + (NVAR*NTOTCOEFF)*log(NOBS)/NOBS;

end

% Find the min of the info criteria
AIC = find(AIC==min(AIC));
SBC = find(SBC==min(SBC));
```

```
function ARDL = ARDLmodel(ENDO,nlag,const,EXOG,nlag_ex)
% =======================================================================
% Estimate ARDL models with OLS
% =======================================================================
% ARDL = ARDLmodel(ENDO,nlag,const,EXOG,nlag_ex)
% -----------------------------------------------------------------------
% INPUT
```

- %   - ENDO: an (nobs x 1) vector of endogenous
-
- %   - nlag: lag length
- % ---
- % OPTIONAL INPUT
-
- %   - const: 0 no constant; 1 constant; 2 constant and trend; 3 constant,
-
- %   trend, and trend^2 [dflt = 0]
-
- %   - EXOG: optional vector of exogenous variable (nobs x 1)
-
- %   - nlag_ex: number of lags for exogeonus variable [dflt = 0]
- % ---
- % OUTPUT
-
- %   - VAR: structure including VAR estimation results
-
- %   - VARopt: structure including VAR options (see VARoption)
- % ===
- % VAR Toolbox 3.0
- % Ambrogio Cesa-Bianchi
- % ambrogiocesabianchi@gmail.com
- % March 2012. Updated November 2020
- % ---
- %% Check inputs
- %===
- [nobs, ~] = size(ENDO);
- ARDL.ENDO = ENDO;
- ARDL.nlag = nlag;
- % Check if ther are constant, trend, both, or none
- if ~exist('const','var')
-     const = 1;
- end
- ARDL.const = const;
- % Check if there is exogenous variable
- if exist('EXOG','var')
-     [nobs_ex, nvar_ex] = size(EXOG);
-     % Check that ENDO and EXOG are conformable
-     if (nobs_ex ~= nobs)
-         error('var: nobs in EXOG-matrix not the same as y-matrix');
-     end
-     clear nobs_ex
-     % Check if there is lag order of EXOG, otherwise set it to 0
-     if ~exist('nlag_ex','var')

```
        nlag_ex = 0;

        end

            ARDL.EXOG = EXOG;

        else

        end

        %% Save some parameters and create data matrices
        %================================================
nvar_ex = 0;
nlag_ex = 0;
ARDL.EXOG = [];
nobse
ARDL.nobs
ARDL.nlag
ARDL.nlag_ex = nlag_ex;
ncoeff       = nlag;
ARDL.ncoeff  = ncoeff;
= nobs - max(nlag,nlag_ex);
= nobse;
= nlag;
ncoeff_ex
nvar
ARDL.nvar
ARDL.nvar_ex = nvar_ex;
ARDL.const   = const;
= nvar_ex + nvar_ex*nlag_ex;
= ncoeff + ncoeff_ex + const;
= nvar;
% Create independent vector and lagged dependent matrix
[Y, X] = VARmakexy(ENDO,nlag,const);
% Create (lagged) exogenous matrix
if exist('EXOG','var')
    X_EX  = VARmakelags(EXOG,nlag_ex);
    if nlag == nlag_ex
        X = [X X_EX];
    elseif nlag > nlag_ex
        diff = nlag - nlag_ex;
        X_EX = X_EX(diff+1:end,:);
        X = [X X_EX];
    elseif nlag < nlag_ex
        diff = nlag_ex - nlag;
        Y = Y(diff+1:end,:);
        X = [X(diff+1:end,:) X_EX];

end end

ARDL.meth = 'ols';
ARDL.y = Y;
ARDL.x = X;
% xpxi = (X'X)^(-1)
if nobse < 10000
  [~, r] = qr(X,0);
  xpxi = (r'*r)\eye(nvar);

else

  xpxi = (X'*X)\eye(nvar);
end;
% OLS estimator
beta = xpxi*(X'*Y);
ARDL.beta = beta;
% Predicted values & residuals
```

```
ARDL.yhat = X*ARDL.beta;
ARDL.resid = Y - ARDL.yhat;
% Covariance matrix of residuals
sigu = ARDL.resid'*ARDL.resid;
ARDL.sige = sigu/(nobse-nvar);
% Covariance matrix of beta
sigbeta = ARDL.sige*xpxi;
ARDL.sigbeta = sigbeta;
% Std errors of beta, t-stats, and intervals
tmp = (ARDL.sige)*(diag(xpxi));
bstd = sqrt(tmp);
ARDL.bstd = bstd;
tcrit=-tdis_inv(.025,nobse);
ARDL.bint=[ARDL.beta-tcrit.*bstd, ARDL.beta+tcrit.*bstd];
ARDL.tstat = ARDL.beta./(sqrt(tmp));
ARDL.tprob = tdis_prb(ARDL.tstat,nobs);

% R2

ym = Y - mean(Y);
rsqr1 = sigu;
rsqr2 = ym'*ym;
ARDL.rsqr = 1.0 - rsqr1/rsqr2; % r-squared
rsqr1 = rsqr1/(nobse-nvar);
rsqr2 = rsqr2/(nobse-1.0);
if rsqr2 ~= 0
    ARDL.rbar = 1 - (rsqr1/rsqr2); % rbar-squared
else
    ARDL.rbar = ARDL.rsqr;
end;
% Durbin-Watson
ediff = ARDL.resid(2:nobse) - ARDL.resid(1:nobse-1);
ARDL.dw = (ediff'*ediff)/sigu; % durbin-watson
ARDL.const = const;

% F-test

if const>0
    fx = X(:,1);
    fxpxi = (fx'*fx)\eye(1);
    fbeta = fxpxi*(fx'*Y);
    fyhat = fx*fbeta;
    fresid = Y - fyhat;
    fsigu = fresid'*fresid;
    fym = Y - mean(Y);
    frsqr1 = fsigu;
    frsqr2 = fym'*fym;
    frsqr = 1.0 - frsqr1/frsqr2; % r-squared
    ARDL.F = ((frsqr-ARDL.rsqr)/(1-nvar)) / ((1-ARDL.rsqr)/(nobse-nvar));

end

% % Long-run coefficients
% q = ncoeff_ex;
% p = ncoeff;
%
% sumendo = sum(beta(const+1:const+p)); % sum of lagged endo
% sumexog = sum(beta(const+p+1:end)); % sum of cont and lagged exog
%
% theta = sumexog/(1-sumendo);
% ARDL.theta = theta;
%
% aux1(1:p,1) = sumexog/((1-sumendo)^2);
% aux2(1:q,1) = 1/(1-sumendo);
% dtheta = [aux1; aux2];
% sigbeta_noconst = sigbeta(const+1:nvar,const+1:nvar);
% ARDL.sigtheta = dtheta'*sigbeta_noconst*dtheta;
```

```
%% 0. PRELIMINARIES
%-----------------------------------------------------------------
clear; close all; clc
warning off all
format short g
addpath(genpath('/Users/francescosimeo.../VARToolbox '))
%% 1. LOAD AND STORE DATA
%*****************************************************************
% Loads some US macro data to be used in the first example. The data is
% stored in the structure DATA
%-----------------------------------------------------------------
% Load data from US macro data set
[xlsdata, xlstext] = xlsread('data/Simple_Data.xlsx','Sheet1');
dates = xlstext(3:end,1);        % vector of dates in string format
datesnum = Date2Num(dates);      % vector of dates in numeric format
vnames_long = xlstext(1,2:end); % full variable names
vnames = xlstext(2,2:end);
nvar = length(vnames);
data   = Num2NaN(xlsdata);
% Store variables in the structure DATA
for ii=1:length(vnames)
DATA.(vnames{ii}) = data(:,ii);
% variable mnemonic
% number of variables in spreadsheet
% matrix of data in spreadsheet

end

% Observations
nobs = size(data,1);
%% 2. TREAT DATA
%*****************************************************************
% Computes the growth rate of real GDP and CPI index and the first
% difference of the 1-year Treasury Bill yield
%-----------------------------------------------------------------
% Select variables to treat
tempnames = {'gdp','cpi','i1yr'};          % variable mnemonics
temptreat = {'logdiff','logdiff','diff'}; % type of transformation
tempscale = [100 100 1];                   % rescaling (if needed)
% Treat and add to DATA structure
for ii=1:length(tempnames)
    aux = {['d' tempnames{ii}]};
    DATA.(aux{1}) = tempscale(ii)*...
        XoX(DATA.(tempnames{ii}),1,temptreat{ii});
delete temp*
%% 3. VAR ESTIMATION
%*****************************************************************
% VAR estimation is achieved in two steps. First, select list of
% endogenous variables (these will be pulled from the structure DATA,
% where all data is stored). Second, set desired number of lags and
% deterministic variables and run the VARmodel function.
%-----------------------------------------------------------------
% Select the list of endogenous variables...
Xvnames = {'dgdp','i1yr'};
% ... and corresponding labels to be used in plots
Xvnames_long = {'Real GDP Growth','1-year Int. Rate'};
% Number of endo variables
Xvnar = length(Xvnames);
% Create matrix X of variables to be used in the VAR
X = nan(nobs,Xvnar);
for ii=1:Xvnar
    X(:,ii) = DATA.(Xvnames{ii});
```

```
end

% Open a figure of the desired size and plot the selected variables
FigSize(26,8)
for ii=1:Xnvar
    subplot(1,2,ii)
    H(ii) = plot(X(:,ii),'LineWidth',3,'Color',cmap(1));

end

title(Xvnames_long(ii));
DatesPlot(datesnum(1),nobs,6,'q') % Set the x-axis labels
grid on;

end

% Save figure

SaveFigure('graphics/BIV_DATA',1)
clf('reset')
% Make a common sample by removing NaNs
[X, fo, lo] = CommonSample(X);
% Set the deterministic variable in the VAR (1=constant, 2=trend)
det = 1;
% Set number of lags
nlags = 1;
% Estimate VAR by OLS
[VAR, VARopt] = VARmodel(X,nlags,det);
% Print at screen the outputs of the VARmodel estimation
format short
disp(VAR)
disp(VAR.F)
disp(VAR.sigma)
disp(VARopt)
% Update the VARopt structure with additional details
VARopt.vnames = Xvnames_long;
% Print at screen VAR coefficients and create table
[TAB    LE, beta] = VARprint(VAR,VARopt,2);
%% 4. IDENTIFICATION WITH ZERO CONTEMPORANEOUS RESTRICTIONS
%************************************************************************
% Identification with zero contemporaneous restrictions is achieved in two
% steps: (1) set the identification scheme mnemonic in the structure
% VARopt to the desired one, in this case "short"; (2) run VARir, VARvd or
% VARhd functions.
%------------------------------------------------------------------------
% Update the VARopt structure to select zero short-run restrictions
VARopt.ident = 'short';
% Update the VARopt structure with additional details
VARopt.vnames = Xvnames_long;
VARopt.nsteps = 12;
VARopt.FigSize = [26,12];
VARopt.firstdate = datesnum(1);
VARopt.frequency = 'q';
VARopt.snames = {'\epsilon^{Demand}',... % shock names
    '\epsilon^{MonPol}'};
% Compute impulse response
[IR, VAR] = VARir(VAR,VARopt);
% Print at screen
format short
disp(VAR.B)
disp(IR(1:4,:,2))
% Compute structural shocks (Tx2)
eps_short = (VAR.B\VAR.resid')';
disp(corr(eps_short))
%% 5. IDENTIFICATION WITH ZERO LONG-RUN RESTRICTIONS
%*****************************************************************************************
% Identification with zero long-run restrictions is achieved in two
```

```
% steps: (1) set the identification scheme mnemonic in the structure
% VARopt to the desired one, in this case "long"; (2) run VARir, VARvd or
% VARhd functions.
%-----------------------------------------------------------------
% Update the VARopt structure to select zero long-run restrictions
VARopt.ident = 'long';
% Update the VARopt structure with additional details
VARopt.snames = {'\epsilon^{Supply}',... % shocks names
    '\epsilon^{Demand}'};
% Compute impulse responses
% variable names in plots
% max horizon of IRF
% size of window (figures)
% first date in plots
% frequency of the data
[IR, VAR] = VARir(VAR,VARopt);
% Print at screen
format short
disp(VAR.B)
disp((eye(2)-VAR.Fcomp)\VAR.B)
% Compute impulse responses in the very long-run and plot
VARopt.nsteps = 150;
[IR, VAR] = VARir(VAR,VARopt);
FigSize(26,8)
subplot(1,2,1)
plot(cumsum(IR(:,1,2)),'LineWidth',2,'Marker','*','Color',cmap(1)); hold on
plot(zeros(VARopt.nsteps),'--k','LineWidth',0.5); hold on
xlim([1 VARopt.nsteps]);
title('Cumulative response of GDP growth')
subplot(1,2,2)
plot(cumsum(IR(:,2,2)),'LineWidth',2,'Marker','*','Color',cmap(1)); hold on
plot(zeros(VARopt.nsteps),'--k','LineWidth',0.5); hold on
xlim([1 VARopt.nsteps]);
title('Cumulative response of the 1-year rate')
SaveFigure('graphics/longCum_',1);
clf('reset')
% Compute structural shocks (Tx2)
eps_long = (VAR.B\VAR.resid')';

%% 6. IDENTIFICATION WITH SIGN RESTRICTIONS
%************************************************************************ %
Identification with sign restrictions is achieved in two steps: (1)
% define a matrix with the sign restrictions that the impulse responses
% have to satisfy; (2) run the SR function. %-------------------------------------
----------------------------
% Define sign restrictions
% Positive 1, Negative -1, Unrestricted 0:
SIGN=[1,1; %RealGDP


         1,-1]; % 1-year rate
% Update the VARopt structure with inputs to the sign restriction routine
VARopt.ndraws = 500;
VARopt.sr_hor = 1;
VARopt.pctg = 68;
% Update the VARopt structure with additional details
VARopt.nsteps = 12;
VARopt.figname= 'graphics/sign_';
VARopt.FigSize = [26 8];
VARopt.snames = {'\epsilon^{Demand}',... % shocks names
    '\epsilon^{MonPol}'};
% Implement sign restrictions identification with SR routine
SRout = SR(VAR,SIGN,VARopt);
% Plot all Btilde
FigSize(26,8)
subplot(1,2,1)
plot(squeeze(SRout.IRall(:,1,2,:))); hold on
plot(zeros(VARopt.nsteps),'--k','LineWidth',0.5); hold on
xlim([1 VARopt.nsteps]);
```

```
title('Response of GDP growth to \epsilon^{MonPol}')
subplot(1,2,2)
plot(squeeze(SRout.IRall(:,2,2,:))); hold on
plot(zeros(VARopt.nsteps),'--k','LineWidth',0.5); hold on
xlim([1 VARopt.nsteps]);
title('Response of 1-year rate to \epsilon^{MonPol}')
SaveFigure('graphics/signAll_',1);
clf('reset')
% Plot credible intervals
VARirplot(SRout.IRmed,VARopt,SRout.IRinf,SRout.IRsup)
% Compute structural shocks (Tx2)
eps_sign = (SRout.B\VAR.resid')';
%% 7. IDENTIFICATION WITH EXTERNAL INSTRUMENTS
% horizon of impulse responses
% folder and file prefix
% size of window (figures)
%************************************************************************
% Identification with external instruments is achieved in three steps: (1)
% update the VAR structure with the external instrument to be used in the
% first stage; (2) set the identification scheme mnemonic in the structure
% VARopt to the desired one, in this case "iv"; (3) run the VARir function
%------------------------------------------------------------------------
% Create artificial instrument (demand shock from eps_short + noise)
rng(1);
noise = randn(nobs,1);              % random vector from N(0,1)
noise = noise(1+fo:end-lo);         % adjust to common sample
iv = [NaN; eps_short(:,1)] + noise; % add noise to demand shock (eps_short)
VAR.IV = iv;                        % update VAR structure
% Update the options in VARopt
VARopt.ident = 'iv';
VARopt.snames = {'\epsilon^{Demand}','\epsilon^{Other}'};
% Compute impulse responses
[IR, VAR] = VARir(VAR,VARopt);
% Plot impulse responses
FigSize(26,8)
for ii=1:Xnvar
    subplot(1,2,ii);
    plot(IR(:,ii),'LineWidth',2,'Marker','*','Color',cmap(1)); hold on;
    plot(zeros(1,VARopt.nsteps),'--k','LineWidth',0.5); hold on
    plot(1,IR(1,ii),'LineStyle','-','Color',cmap(5),'LineWidth',2,...
        'Marker','p','MarkerSize',20,'MarkerFaceColor',cmap(5)); hold on
    xlim([1 VARopt.nsteps]);
    title([Xvnames_long{ii} ' to '
VARopt.snames{1}],'FontWeight','bold','FontSize',10);
    set(gca, 'Layer', 'top');

end

SaveFigure('graphics/iv_',1)
clf('reset');
%% 8. IDENTIFICATION WITH EXTERNAL INSTRUMENTS AND SIGN RESTRICTIONS
%************************************************************************
% Identification with external instruments and sign restrictions is
% achieved in five steps: (1) update the VAR structure with the external
% instrument to be used in the first stage; (2) set the identification
% scheme mnemonic in the structure VARopt to the desired one, in this case
% "iv"; (3) run the VARir function to get an estimate of Biv; (4) define a
% matrix with the sign restrictions that the IRs have to satisfy, excluding
% the first shock identified with external instruments; (5) run the SR
% function.
%------------------------------------------------------------------------
% VAR structure has already been updated to include the first column of
% the B matrix that we identified with the instrument in the previous
% section:
disp(VAR.Biv)
```

```
% Define sign restrictions to identify monetary policy shock % Positive 1, Negative
-1, Unrestricted 0:
SIGN=[1; %RealGDP

        -1]; % 1-year rate
% Update the VARopt structure with additional details
VARopt.figname= 'graphics/iv_sign_';
VARopt.snames = {'\epsilon^{Demand}','\epsilon^{MonPol}'};
% Implement sign restrictions identification with SR routine
% conditional on VAR.Biv being already identified
SRIVout = SR(VAR,SIGN,VARopt);
% Plot impulse responses
VARirplot(SRIVout.IRmed,VARopt,SRIVout.IRinf,SRIVout.IRsup);
m2tex('VARToolbox_Primer_v1.m')
close all
```

```
function results =  adf(x,p,l)
% PURPOSE: carry out DF tests on a time-series vector
% Is there a positive probability that results.adf > results.crit? If so
% (e.g., prob >0.10) we fail to reject the null of I(1)
%---------------------------------------------------
% USAGE: results = adf(x,p,nlag)

% where: %
%
%

x = a time-series vector

p=

order of time polynomial in the null-hypothesis
p = -1, no deterministic part
p =  0, for constant term
p =  1, for constant plus time-trend
%
%
%
%---------------------------------------------------
% RETURNS: a results structure
        p >  1, for higher order polynomial
nlags = # of lagged changes of x included
```

- `%          results.meth  = 'adf'`
-
- `%          results.alpha = estimate of the autoregressive parameter`
-
- `%          results.adf   = ADF t-statistic`
-
- `%          results.crit = (6 x 1) vector of critical values`
-
- `%                         [1% 5% 10% 90% 95% 99%] quintiles`
-
- `%          results.nlag = nlag`
- `%---------------------------------------------------`
- `% SEE ALSO: prt_coint()`
- `%---------------------------------------------------`
- `% References: Said and Dickey (1984) 'Testing for Unit Roots in`
- `% Autoregressive Moving Average Models of Unknown Order',`
- `% Biometrika, Volume 71, pp. 599-607.`
- `% written by:`
- `% James P. LeSage, Dept of Economics`
- `% University of Toledo`
- `% 2801 W. Bancroft St,`
- `% Toledo, OH 43606`
- `% jlesage@spatial-econometrics.com`
- `% Modeled after a similar Gauss routine by`
- `% Sam Ouliaris, in a package called COINT`
- `% error checking on inputs`
- `if (nargin ~= 3)`
- ` error('Wrong # of arguments to adf');`
- `end;`
- `if (p < -1)`
- ` error('p less than -1 in adf');`

```
•   elseif (cols(x) > 1)
•    error('adf cannot handle a matrix -- only vectors');

    end;

    nobs = rows(x);
    % if ((nobs - 2*l)+1 < 1)
    %  error('nlags too large in adf, negative dof');
    % end;
    dep = trimr(x,1,0);
    ch = tdiff(x,1);
    ch = trimr(ch,1,0);

    % Gerard van den Hout suggested the fix below % Erasmus University Rotterdam.
    % The Netherlands.
    k=0;

    z = [];
    while (k < l);
        k = k+1;
        z = [z lag(ch,k)];
end;
z = trimr(z,k,0);
dep = trimr(dep,k,0);
if (p > -1)
 z = [z ptrend(p,rows(z))];

end;

ylag=lag(dep);
ylag=trimr(ylag,1,0);
z2=trimr(z,1,0);
y2=trimr(dep,1,0);
regressor=[ylag,z2];
results=ols(y2,regressor);
•   %       b       = inv(z'*z)*(z'*dep);
•
•   %       % res     = dep - z*b ;
•
•   %       % BUG fix suggested by
•   % % Nick Firoozye
•   % % Sanford C. Bernstein, Inc
•

%      res = detrend(dep,0) - detrend(z,0)*b;
res=results.resid;
     %so      = (res'*res)/(rows(y2)-cols(regressor))
     so=results.sige;
     var_cov = so*inv(regressor'*regressor) ;
     results.nlag = l;
     results.alpha = results.beta(1,1);
     results.adf = (results.beta(1,1)-1)/sqrt(var_cov(1,1));
     results.crit = ztcrit(nobs,p);
     results.meth = 'adf';
```

```
function [TABLE, beta] = VARprint(VAR,VARopt,approx)
% =======================================================================
% Prints the output of a VAR estimation
% =======================================================================
% [TABLE, beta] = VARprint(VAR,VARopt,approx)
% -----------------------------------------------------------------------
% INPUT
```

- %   - VAR: structure output of VARmodel function
-
- %   - VARopt: options of the VAR (see VARopt from VARmodel)
- % ---
- % OPTIONAL INPUT
-

```
%   - approx: number of decimal digits. Default = 4
%-----------------------------------------------------------------------
% OUPUT
%
%
%
% -----------------------------------------------------------------------
% EXAMPLE
- TABLE: table of estimated coefficients, std errors, t-stats, and p
    values in in cell array
- beta: table of estimated coefficients only in cell array
%   - See VARToolbox_Code.m in "../Primer/"
% =======================================================================
% VAR Toolbox 3.0
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated November 2020
% -----------------------------------------------------------------------
%% Check inputs
%===============================================
if ~exist('VARopt','var')
    error('You need to provide VAR options (VARopt from VARmodel)');
end
% If there is VARopt get the vnames
vnames = VARopt.vnames;
vnames_ex = VARopt.vnames_ex;
% Check they are not empty
if isempty(vnames)
    error('You need to add label for endogenous variables in VARopt');
end
if VAR.nvar_ex>0
    if isempty(vnames_ex)
        error('You need to add label for exogenous variables in VARopt');
    end
end
if ~exist('approx','var')

approx = 4;

end

%% Retrieve and initialize variables
%==============================================================
nlag = VAR.nlag;
nlag_ex = VAR.nlag_ex;
const = VAR.const;
%% Additional checks
```

```matlab
%==============================================================
% Check size of vnames (and change it if necessary)
htext = vnames;
if size(htext,2)==1
    htext = htext';
    nvars = size(htext,2);

else

end

nvars = size(htext,2);
% Check size of vnames_ex (and change it if necessary)
if exist('vnames_ex','var')
    if size(vnames_ex,2)==1
        vnames_ex = vnames_ex';
        nvars_ex = size(vnames_ex,2);

else

end end

%% Labels of deterministic components
%================================================
switch const
    case 0
        aux = [];
    case 1
        aux = {'c'};
    case 2
        aux = {'c';'trend'};
    case 3
        aux = {'c';'trend';'trend2'};

end

vtext = {' '};
vtext = [vtext; aux];
clear aux
%% Labels of lagged variables
%================================================
for jj=1:nlag
    for ii=1:nvars
        aux(ii,1) = {[vnames{ii} '(-' num2str(jj) ')' ]};

end

    vtext = [vtext ; aux];

end

clear aux

%% Labels of exogenous variables
%================================================
if VAR.nvar_ex>0
    vtext = [vtext ; vnames_ex'];
    if nlag_ex > 0
        for jj=1:nlag_ex
            for ii=1:nvars_ex
                aux(ii,1) = {[vnames_ex{ii} '(-' num2str(jj) ')' ]};
            end
            vtext = [vtext ; aux];
```

```
end

clear aux end

end

%% Save
%================================================
% Save a beta table
beta = roundnum2cell(VAR.Ft,approx);
beta = [htext; beta];
beta = [vtext beta];
% Save a standard error table
bstd = [];
for ii=1:nvars
    eval( ['aux = VAR.eq' num2str(ii) '.bstd;'] );
    bstd = [bstd aux];

end

bstd = roundnum2cell(bstd,approx);
bstd = [htext; bstd];
bstd = [vtext bstd];
nvars_ex = size(vnames_ex,2);

clear aux

% Save a tstat table
tstat = [];
for ii=1:nvars
    eval( ['aux = VAR.eq' num2str(ii) '.tstat;'] );
    tstat = [tstat aux];

end

tstat = roundnum2cell(tstat,2);
tstat = [htext; tstat];
tstat = [vtext tstat];
clear aux
% Save a p-value table
tprob = [];
for ii=1:nvars
    eval( ['aux = VAR.eq' num2str(ii) '.tprob;'] );
    tprob = [tprob aux];

end

tprob = roundnum2cell(tprob,approx);
tprob = [htext; tprob];
tprob = [vtext tprob];
clear aux
% Save a beta & tstat table
nn = size(beta,1)-1;
TABLE = {''};
index = 1;
for ii=1:nn
    for jj=1:nvars
        TABLE(index,jj) = beta(1+ii,1+jj);
        TABLE(index+1,jj) = bstd(1+ii,1+jj);
        aux1 = cell2mat(tstat(1+ii,1+jj)); % get the numeric value from cell
        aux2 = [ '[' num2str(aux1) ']' ];  % add parenthesis to t-stat value
        TABLE{index+2,jj} = aux2;
        TABLE(index+3,jj) = tprob(1+ii,1+jj);

end
```

```
    index = index+4;
end
for jj=1:nvars
    eval( ['aux = VAR.eq' num2str(jj) '.rsqr;'] ); TABLE(index,jj) = num2cell(aux);
    eval( ['aux = VAR.eq' num2str(jj) '.rbar;'] ); TABLE(index+1,jj) =
num2cell(aux);
    TABLE(index+2,jj) = num2cell(VAR.nobs);

end

clear aux
TABLE = [htext; TABLE];
% Create vertical label
TAB_v = {''};
index = 2;
for ii=1:nn
    TAB_v(index,1) = vtext(1+ii);
    TAB_v(index+1,1) = {['std(' vtext{1+ii} ')']};
    TAB_v(index+2,1) = {['t(' vtext{1+ii} ')']};
    TAB_v(index+3,1) = {['p(' vtext{1+ii} ')']};
    index = index + 4;

end

TAB_v(index,1) = {'R2'};
TAB_v(index+1,1) = {'R2bar'};
TAB_v(index+2,1) = {'Obs'};
TABLE = [TAB_v TABLE];
%% Print the table on screen (only beta)
%================================================
format short g
info.cnames = char(htext);
info.rnames = char(vtext);
disp(' ')
%disp('---------------------------------------------------------------------')
disp(' ')
disp('Reduced form VAR estimation:')
disp(' ')
mprint(VAR.Ft,info)
%disp('---------------------------------------------------------------------')
disp(' ')
disp('VAR eigenvalues:')
disp(eig(VAR.Fcomp))
disp(' ')
disp('Reduced-form covariance matrix:')
disp(VAR.sigma)
```

```
function OLS = OLSmodel(y,x,const)
% =======================================================================
% OLS regression
% =======================================================================
% OLS = OLSmodel(y,x,const)
% -----------------------------------------------------------------------
% INPUT
```

- %   - y: dependent variable vector   (nobs x 1)
-
- %   - x: independent variables matrix (nobs x nvar)
- % ---
- % OPTIONAL INPUT
- %   - const: 0 no constant; 1 constant; 2 constant and trend; 3 constant
- %   and trend^2 [dflt = 0]
- % ---
- % OUPUT
-

```
%   - OLS: structure including VAR estimation results
% =======================================================================
% VAR Toolbox 3.0
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated November 2020
% -----------------------------------------------------------------------
% Check inputs
if isempty(x)
    [nobs, ~] = size(y);
    nvar = 0;

else


    [nobs, nvar] = size(x);
    [nobs2, ~] = size(y);
    if (nobs ~= nobs2); error('x and y must have same # obs'); end

end

% Check if ther are constant, trend, both, or none
if ~exist('const','var')
    const = 3;

end

% Add constant or trend if needed
if const==1 %constant
    x = [ones(nobs,1) x];
    nvar = nvar+1;
elseif const==2 % trend and constant
    trend = 1:nobs;
    x = [ones(nobs,1) trend' x];
    nvar = nvar+2;
elseif const==3 % trend^2, and constant
    trend = 1:nobs;
    x = [ones(nobs,1) trend'.^2 x];
    nvar = nvar+3;

end
```

```
OLS.meth = 'ols';
OLS.y = y;
OLS.x = x;
OLS.nobs = nobs;
OLS.nvar = nvar;
% xpxi = (X'X)^(-1)
if nobs < 10000
  [~, r] = qr(x,0);
  xpxi = (r'*r)\eye(nvar);

else

  xpxi = (x'*x)\eye(nvar);
end;

% OLS estimator
OLS.beta = xpxi*(x'*y);
% Predicted values & residuals
OLS.yhat = x*OLS.beta;
OLS.resid = y - OLS.yhat;
% Covariance matrix of residuals
sigu = OLS.resid'*OLS.resid;
OLS.sige = sigu/(nobs-nvar);
% Covariance matrix of beta
OLS.sigbeta = OLS.sige*xpxi;
% Std errors of beta, t-stats, intervals, and p-values
tmp = (OLS.sige)*(diag(xpxi));
sigb = sqrt(tmp);
OLS.bstd = sigb;
tcrit=-tdis_inv(.025,nobs);
OLS.bint=[OLS.beta-tcrit.*sigb, OLS.beta+tcrit.*sigb];
OLS.tstat = OLS.beta./(sqrt(tmp));
OLS.tprob = tdis_prb(OLS.tstat,nobs);

% R2

ym = y - mean(y);
rsqr1 = sigu;
rsqr2 = ym'*ym;
OLS.rsqr = 1.0 - rsqr1/rsqr2; % r-squared
rsqr1 = rsqr1/(nobs-nvar);
rsqr2 = rsqr2/(nobs-1.0);
if rsqr2 ~= 0
    OLS.rbar = 1 - (rsqr1/rsqr2); % rbar-squared
else
    OLS.rbar = OLS.rsqr;

end

% Durbin-Watson
ediff = OLS.resid(2:nobs) - OLS.resid(1:nobs-1);
OLS.dw = (ediff'*ediff)/sigu; % durbin-watson
OLS.const = const;

% F-test

if const>0
    fx = x(:,1);
    fxpxi = (fx'*fx)\eye(1);
    fbeta = fxpxi*(fx'*y);
    fyhat = fx*fbeta;
    fresid = y - fyhat;
    fsigu = fresid'*fresid;
    fym = y - mean(y);
    frsqr1 = fsigu;
    frsqr2 = fym'*fym;
```

```
    frsqr = 1.0 - frsqr1/frsqr2; % r-squared
    OLS.F = ((frsqr-OLS.rsqr)/(1-nvar)) / ((1-OLS.rsqr)/(nobs-nvar));

end
```

```
function [TABLE, beta] = OLSprint(OLS,vnames,ynames,approx)
% =======================================================================
% Prints the output of an OLS estimation
% =======================================================================
% [TABLE, beta] = OLSprint(OLS,vnames,snames,approx)
% -----------------------------------------------------------------------
% INPUT
```

- %   - OLS: structure output of OLSmodel function
-
- %   - vnames: name of the variables (no constant or trends)
- % ---
- % OPTIONAL INPUT
-
- %   - ynames: name of the dependent variable
-
- %   - approx: number of decimal digits. Default = 4
- %--
- % OUPUT
-

```
%
%
%
% =======================================================================
% VAR Toolbox 3.0
- TABLE: table of estimated coefficients, std errors, t-stats, and p
    values in in cell array
- beta: table of estimated coefficients only in cell array
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated November 2020
% -----------------------------------------------------------------------
%% Check inputs
%===============================================
if ~exist('vnames','var')
    error('You need to provide variable names');
end
if ~exist('snames','var')
    ynames = {''};
end
if ~exist('approx','var')

approx = 4;

end
if size(vnames,1)==1
    vnames = vnames';

end

%% Retrieve and initialize variables
%============================================================
const = OLS.const;
r = length(OLS.beta);
TAB = nan(2*r+4,1);
%% Table: deterministic components
%============================================================
switch const
```

```
    case 0
        aux = [];
    case 1
        aux = {'c'};
    case 2
        aux = {'c';'trend';};
    case 3
        aux = {'c';'trend';'trend2'};

end

vnames = [aux; vnames];
clear aux
%% Table: regressors
%==============================================================
indexb = 1; indexs = 2; indext = 3; indexp = 4;
for ii=1:r
TAB(indexb,1) = OLS.beta(ii);
TAB(indexs,1) = OLS.bstd(ii);
TAB(indext,1) = OLS.tstat(ii);
TAB(indexp,1) = OLS.tprob(ii);
vtext(indexb,1) = vnames(ii);
vtext(indexs,1) = {['std(' vnames{ii} ')']};
vtext(indext,1) = {['t(' vnames{ii} ')']};
vtext(indexp,1) = {['p(' vnames{ii} ')']};
indexb = indexb+4;
indexs = indexs+4;
indext = indext+4;
indexp = indexp+4;

end

index = indexb;
TAB(index) = OLS.rsqr; vtext(index) = {'R2'};
TAB(index) = OLS.rbar; vtext(index) = {'R2bar'}; index = index+1;
if const>0
    TAB(index) = OLS.F;    vtext(index) = {'F'};
    TAB(index) = [];    vtext(index) = {'F'};
TAB(index) = OLS.nobs; vtext(index) = {'Obs'};
%% Save
%================================================
beta = TabPrint(OLS.beta,ynames,vnames,approx);
TABLE = TabPrint(TAB,ynames,vtext,approx);
%% Print the TABLE on screen
%================================================
info.cnames = char(ynames);
info.rnames = char([{''}; vtext]);
disp(' ')
disp('OLS estimation:')
disp(' ')
mprint(TAB,info)
    index = index+1;
 index = index+1;
index = index+1;

else end

index = index+1;
```

```
function [HD, VAR] = VARhd(VAR,VARopt)
% =======================================================================
% Compute the historical decomposition of the time series in a VAR
% estimated with VARmodel and identified with VARir/VARfevd
% =======================================================================
% HD = VARhd(VAR)
% -----------------------------------------------------------------------
% INPUTS
```

- %   - VAR: structure, result of VARmodel -> VARir/VARfevd function
-
- %   - VARopt: options of the VAR (result of VARmodel.m)
- % ---
- % OUTPUT
-

```
%   - HD: structure including the historical decomposition
% -----------------------------------------------------------------------
% EXAMPLE
%   - See VARToolbox_Code.m in "../Primer/"
% =======================================================================
% VAR Toolbox 3.0
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated November 2020
% -----------------------------------------------------------------------
% I thank Andrey Zubarev for finding a bug in the contribution of the
% exogenous variables when nvar_ex~=0 and nlag_ex>0.
%% Check inputs
%========================================================================
if ~exist('VAR','var')
    error('You need to provide VAR structure, result of VARmodel');
end
IV = VAR.IV;
if strcmp(VARopt.ident,'iv')
    disp('---------------------------------------------')
    disp('Historical decomposition not available with')
    disp('external instruments identification (iv)');
    disp('---------------------------------------------')
    error('ERROR. See details above');

end

%% Retrieve and initialize variables
%========================================================================
                                          % VCV matrix of the VAR
                                          % Companion matrix
                                          % constant and/or trends
                                          % make comparable to notes
                                          % number of endogenous variables
                                          % number of exogenous (excluding constant
and
                                          % number of lagged endogenous per

equation
                                          % number of lags
                                          % number of lags of the exogenous
                                          % left-hand side
sigma
Fcomp
const
F
nvar
```

```
nvar_ex = VAR.nvar_ex;
trend)
= VAR.sigma;
= VAR.Fcomp;
= VAR.const;
= VAR.Ft';

= VAR.nvar;

nvarXeq = VAR.nvar * VAR.nlag;
nlag    = VAR.nlag;
nlag_ex = VAR.nlag_ex;
Y       = VAR.Y;
X       = VAR.X(:,1+const:nvarXeq+const); % right-hand side (no exogenous)
nobs    = size(Y,1);                      % number of observations
%% Identification: Recover B matrix
%==========================================================================
% B matrix is recovered with Cholesky decomposition
if strcmp(VARopt.ident,'short')
    [out, chol_flag] = chol(sigma);
    if chol_flag~=0; error('VCV is not positive definite'); end
    B = out';

% B matrix is recovered with Cholesky on cumulative IR to infinity
elseif strcmp(VARopt.ident,'long')
    Finf_big = inv(eye(length(Fcomp))-Fcomp); % from the companion
    Finf = Finf_big(1:nvar,1:nvar);
    D  = chol(Finf*sigma*Finf')'; % identification: u2 has no effect on y1 in the
long run
    B = Finf\D;
% B matrix is recovered with SR.m
elseif strcmp(VARopt.ident,'sign')
    if isempty(VAR.B)
        error('You need to provide the B matrix with SR.m and/or
SignRestrictions.m')
    else

B = VAR.B;

end

% B matrix is recovered with external instrument IV
elseif strcmp(VARopt.ident,'iv')
    disp('---------------------------------------------')
    disp('Forecast error variance decomposition not available with')
    disp('external instruments identification (iv)');
    disp('---------------------------------------------')
    error('ERROR. See details above');
% If none of the above, you've done something wrong :)

else

end

%% Compute historical decompositions
%==========================================================================
% Contribution of each shock
eps = B\transpose(VAR.resid); % structural errors
B_big = zeros(nvarXeq,nvar);
B_big(1:nvar,:) = B;
Icomp = [eye(nvar) zeros(nvar,(nlag-1)*nvar)];
HDshock_big = zeros(nlag*nvar,nobs+1,nvar);
HDshock = zeros(nvar,nobs+1,nvar);
for j=1:nvar % for each variable
    eps_big = zeros(nvar,nobs+1); % matrix of shocks conformable with companion
    eps_big(j,2:end) = eps(j,:);
```

```
    for i = 2:nobs+1
        HDshock_big(:,i,j) = B_big*eps_big(:,i) + Fcomp*HDshock_big(:,i-1,j);
        HDshock(:,i,j) =  Icomp*HDshock_big(:,i,j);

end end

% Initial value
HDinit_big = zeros(nlag*nvar,nobs+1);
HDinit = zeros(nvar, nobs+1);
HDinit_big(:,1) = X(1,:)';
HDinit(:,1) = Icomp*HDinit_big(:,1);
for i = 2:nobs+1
    HDinit_big(:,i) = Fcomp*HDinit_big(:,i-1);
    HDinit(:,i) = Icomp *HDinit_big(:,i);

end

% Constant

HDconst_big = zeros(nlag*nvar,nobs+1);
HDconst = zeros(nvar, nobs+1);
CC = zeros(nlag*nvar,1);
disp('---------------------------------------------')
disp('Identification incorrectly specified.')
disp('Choose one of the following options:');
disp('- short: zero contemporaneous restrictions');
disp('- long:  zero long-run restrictions');
disp('- sign:  sign restrictions');
disp('- iv:    external instrument');
disp('---------------------------------------------')
error('ERROR. See details above');

if const>0
    CC(1:nvar,:) = F(:,1);
    for i = 2:nobs+1
        HDconst_big(:,i) = CC + Fcomp*HDconst_big(:,i-1);
        HDconst(:,i) = Icomp * HDconst_big(:,i);

end end

% Linear trend
HDtrend_big = zeros(nlag*nvar,nobs+1);
HDtrend = zeros(nvar, nobs+1);
TT = zeros(nlag*nvar,1);
if const>1
    TT(1:nvar,:) = F(:,2);
    for i = 2:nobs+1
        HDtrend_big(:,i) = TT*(i-1) + Fcomp*HDtrend_big(:,i-1);
        HDtrend(:,i) = Icomp * HDtrend_big(:,i);

end end

% Quadratic trend
HDtrend2_big = zeros(nlag*nvar, nobs+1);
HDtrend2 = zeros(nvar, nobs+1);
TT2 = zeros(nlag*nvar,1);
if const>2
    TT2(1:nvar,:) = F(:,3);
    for i = 2:nobs+1
        HDtrend2_big(:,i) = TT2*((i-1)^2) + Fcomp*HDtrend2_big(:,i-1);
        HDtrend2(:,i) = Icomp * HDtrend2_big(:,i);

end end

% Exogenous
```

```
HDexo_big = zeros(nlag*nvar,nobs+1);
HDexo = zeros(nvar,nobs+1);
EXO = zeros(nlag*nvar,nvar_ex*(nlag_ex+1));
if nvar_ex>0
    for ii=1:nvar_ex
        VARexo = VAR.X_EX(:,ii);
        EXO(1:nvar,ii) = F(:,nvar*nlag+const+ii); % this is c in my notes
        for i = 2:nobs+1
            HDexo_big(:,i) = EXO(:,ii)*VARexo(i-1,:)' + Fcomp*HDexo_big(:,i-1);
            HDexo(:,i,ii) = Icomp * HDexo_big(:,i);

end end

end

% All decompositions must add up to the original data
HDendo = HDinit + HDconst + HDtrend + HDtrend2 + sum(HDexo,3) + sum(HDshock,3);
%% Save and reshape all HDs
%=========================================================================
HD.shock = zeros(nobs+nlag,nvar,nvar);  % [nobs x shock x var]
    for i=1:nvar
        for j=1:nvar
            HD.shock(:,j,i) = [nan(nlag,1); HDshock(i,2:end,j)'];

end end

HD.init   = [nan(nlag-1,nvar); HDinit(:,1:end)'];    % [nobs x var]
HD.const  = [nan(nlag,nvar);   HDconst(:,2:end)'];   % [nobs x var]
HD.trend  = [nan(nlag,nvar);   HDtrend(:,2:end)'];   % [nobs x var]
HD.trend2 = [nan(nlag,nvar);   HDtrend2(:,2:end)'];  % [nobs x var]
HD.exo    = zeros(nobs+nlag,nvar,nvar_ex);           % [nobs x var x var_ex]
    for i=1:nvar_ex
        HD.exo(:,:,i) = [nan(nlag,nvar);   HDexo(:,2:end,i)'];

end

HD.endo   = [nan(nlag,nvar);   HDendo(:,2:end)'];    % [nobs x var]
% Update VAR with structural impact matrix

VAR.B = B;
```

```
function VARhdplot(HD,VARopt)
% =======================================================================
% Plot the HD shocks computed with VARhd
% =======================================================================
% VARhdplot(HD,VARopt)
% -----------------------------------------------------------------------
% INPUT
```

- %   - HD: structure from VARhd
-
- %   - VARopt: options of the VAR (see VARopt from VARmodel)
- % ---
- % EXAMPLE
-

```
%   - See VARToolbox_Code.m in "../Primer/"
% =======================================================================
% VAR Toolbox 3.0
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated April 2021
% -----------------------------------------------------------------------
%% Check inputs
%========================================================================
if ~exist('VARopt','var')
    error('You need to provide VAR options (VARopt from VARmodel)');
end
% If there is VARopt check that vnames is not empty
vnames = VARopt.vnames;
if isempty(vnames)
    error('You need to add label for endogenous variables in VARopt');
end
% Define shock names
if isempty(VARopt.snames)
    snames = VARopt.vnames;

else

end

%% Check inputs and define some parameters
%========================================================================
filename = [VARopt.figname 'HD_'];
quality = VARopt.quality;
suptitle = VARopt.suptitle;
pick = VARopt.pick;
% Initialize HD matrix
[nsteps, nvars, nshocks] = size(HD.shock);
% If one shock is chosen, set the right value for nshocks
if pick<0 || pick>nvars
    error('The selected shock is non valid')
else
    if pick==0

pick=1;

else

end end

%% Plot
```

```
%==========================================================================
FigSize(VARopt.FigSize(1),VARopt.FigSize(2))
for ii=pick:nvars
    H = AreaPlot(squeeze(HD.shock(:,:,ii))); hold on;
    h = plot(sum(squeeze(HD.shock(:,:,ii)),2),'-k','LineWidth',2);
    if ~isempty(VARopt.firstdate);
DatesPlot(VARopt.firstdate,nsteps,8,VARopt.frequency); end
    xlim([1 nsteps]);
snames = VARopt.snames;
nshocks = pick;
    set(gca,'Layer','top');
    title([vnames{ii}], 'FontWeight','bold','FontSize',10);
    % Save
    FigName = [filename num2str(ii)];
    if quality
        if suptitle==1
            Alphabet = char('a'+(1:nvars)-1);
            SupTitle([Alphabet(jj) ') HD of '  vnames{ii}])

end

        opt = LegOption; opt.handle = [H(1,:) h];
        LegSubplot([snames {'Data'}],opt);
        set(gcf, 'Color', 'w');
        export_fig(FigName,'-pdf','-painters')

else

        print('-dpdf','-r100',FigName);
    end
    clf('reset');
end

close all

legend([H(1,:) h],[vsnames {'Data'}])
```

```
function [IR, VAR] = VARir(VAR,VARopt)
% =======================================================================
% Compute impulse responses (IRs) for a VAR model estimated with the
% VARmodel.m function. Four identification schemes can be specified:
% zero contemporaneous restrictions, zero long-run restrictions, sign
% restrictions, and external instrumenmts.
% =======================================================================
% [IRF, VAR] = VARir(VAR,VARopt)
% -----------------------------------------------------------------------
% INPUT
```

- %   - VAR: structure, result of VARmodel.m
-
- %   - VARopt: options of the VAR (result of VARmodel.m)
- % ---
- % OUTPUT
-

```
%
%
%
%
%
% =====================================================================
% VAR Toolbox 3.0
- IR(:,:,:) : matrix with IRF (H horizons, N variables, N shocks)
- VAR: structure including VAR estimation results. Note here that the
structure VAR is an output of VARmodel, too. This fucntion adds to
VAR some additional results, e.g. VAR.B is the structural impact
matrix
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated November 2020
% ---------------------------------------------------------------------
%% Check inputs
%=======================================================================
if ~exist('VAR','var')
    error('You need to provide VAR structure, result of VARmodel');
end
IV = VAR.IV;
if strcmp(VARopt.ident,'iv')
    if isempty(IV)
        error('You need to provide the data for the instrument in VAR (IV)');

end end

%% Retrieve and initialize variables
%=======================================================================
nsteps = VARopt.nsteps;
impact = VARopt.impact;
shut   = VARopt.shut;
recurs = VARopt.recurs;
Fcomp  = VAR.Fcomp;
nvar   = VAR.nvar;
nlag   = VAR.nlag;
sigma  = VAR.sigma;
IR     = nan(nsteps,nvar,nvar);
%% Compute Wold representation
%=======================================================================
% Initialize Wold multipliers
PSI = zeros(nvar,nvar,nsteps);
% Re-write F matrix to compute multipliers
```

```
VAR.Fp = zeros(nvar,nvar,nlag);
I = VAR.const+1;
for ii=1:nsteps
    if ii<=nlag
        VAR.Fp(:,:,ii) = VAR.F(:,I:I+nvar-1);

else

end

I = I + nvar;

VAR.Fp(:,:,ii) = zeros(nvar,nvar);

end
% Compute multipliers
PSI(:,:,1) = eye(nvar);
for ii=2:nsteps
    jj=1;
    aux = 0;
    while jj<ii
        aux = aux + PSI(:,:,ii-jj)*VAR.Fp(:,:,jj);

jj=jj+1;

end

    PSI(:,:,ii) = aux;

end

% Update VAR with Wold multipliers
VAR.PSI = PSI;
%% Identification: Recover B matrix
%==========================================================================
% B matrix is recovered with Cholesky decomposition
if strcmp(VARopt.ident,'short')
    [out, chol_flag] = chol(sigma);
    if chol_flag~=0; error('VCV is not positive definite'); end
    B = out';
% B matrix is recovered with Cholesky on cumulative IR to infinity
elseif strcmp(VARopt.ident,'long')
    Finf_big = inv(eye(length(Fcomp))-Fcomp);
    Finf = Finf_big(1:nvar,1:nvar);
    D  = chol(Finf*sigma*Finf')';
    B = Finf\D;
% B matrix is recovered with SR.m
elseif strcmp(VARopt.ident,'sign')
    if isempty(VAR.B)
        error('You need to provide the B matrix with SR.m and/or
SignRestrictions.m')
    else

B = VAR.B;

end

% B matrix is recovered with external instrument IV
elseif strcmp(VARopt.ident,'iv')
    % Recover residuals (first variable is the one to be instrumented - order
matters!)
    up = VAR.resid(:,1);     % residuals to be instrumented
    uq = VAR.resid(:,2:end); % residulas for second stage
    % Make sample of IV comparable with up and uq
    [aux, fo, lo] = CommonSample([up IV(VAR.nlag+1:end,:)]);
```

```
    p = aux(:,1);
    q = uq(end-length(p)+1:end,:); pq = [p q];
    Z = aux(:,2:end);
    % Run first stage regression and fitted
    FirstStage = OLSmodel(p,Z);
    p_hat = FirstStage.yhat;
    % Recover first column of B matrix with second stage regressions
    Biv(1,1) = 1;  % Start with impact IR normalized to 1
    sqsp = zeros(size(q,2),1);
    for ii=2:nvar
        SecondStage = OLSmodel(q(:,ii-1),p_hat);
        Biv(ii,1) = SecondStage.beta(2);
        sqsp(ii-1) = SecondStage.beta(2);

end

    % Update size of the shock (ftn 4 of Gertler and Karadi (2015))
    sigma_b = (1/(length(pq)-VAR.ntotcoeff))*...
        (pq-repmat(mean(pq),size(pq,1),1))'*...
        (pq-repmat(mean(pq),size(pq,1),1));
    s21s11 = sqsp;
    S11 = sigma_b(1,1);
    S21 = sigma_b(2:end,1);
    S22 = sigma_b(2:end,2:end);
    Q = s21s11*S11*s21s11'-(S21*s21s11'+s21s11*S21')+S22;
    sp = sqrt(S11-(S21-s21s11*S11)'*(Q\(S21-s21s11*S11)));
    % Rescale Biv vector
    Biv = Biv*sp;
    B = zeros(nvar,nvar);
    B(:,1) = Biv;
% If none of the above, you've done somerthing wrong :)

else

end

%% Compute the impulse response
%==========================================================================
for mm=1:nvar
    % Set to zero a row of the companion matrix if "shut" is selected
    if shut~=0
        Fcomp(shut,:) = 0;

end

    % Initialize the impulse response vector
    response = zeros(nvar, nsteps);
    % Create the impulse vector
    impulse = zeros(nvar,1);
    % Set the size of the shock
    if impact==0
        impulse(mm,1) = 1; % one stdev shock
    elseif impact==1
        impulse(mm,1) = 1/B(mm,mm); % unitary shock

else

        error('Impact must be either 0 or 1');
    end
    % First period impulse response (=impulse vector)
    response(:,1) = B*impulse;
    % Shut down the response if "shut" is selected
    if shut~=0
        response(shut,1) = 0;

end
```

```
    % Recursive computation of impulse response
    if strcmp(recurs,'wold')
        for kk = 2:nsteps
            response(:,kk) = PSI(:,:,kk)*B*impulse;
        end
    elseif strcmp(recurs,'comp')
        for kk = 2:nsteps
            FcompN = Fcomp^(kk-1);
            response(:,kk) = FcompN(1:nvar,1:nvar)*B*impulse;

end end

    IR(:,:,mm) = response';

end

% Update VAR with structural impact matrix
VAR.B = B;
if strcmp(VARopt.ident,'iv')
    VAR.FirstStage = FirstStage;
    VAR.sigma_b = sigma_b;
    VAR.Biv = Biv;

end

disp('---------------------------------------------')
disp('Identification incorrectly specified.')
disp('Choose one of the following options:');
disp('- short: zero contemporaneous restrictions');
disp('- long:  zero long-run restrictions');
disp('- sign:  sign restrictions');
disp('- iv:  external instrument');
disp('---------------------------------------------')
error('ERROR. See details above');
```

```
function [INF,SUP,MED,BAR] = VARirband(VAR,VARopt)
% =======================================================================
% Calculate confidence intervals for impulse response functions computed
% with VARir
% =======================================================================
% [INF,SUP,MED,BAR] = VARirband(VAR,VARopt)
% -----------------------------------------------------------------------
% INPUTS
```

- %   - VAR: structure, result of VARmodel.m
-
- %   - VARopt: options of the VAR (result of VARmodel.m)
- % ---
- % OUTPUT
-
- %   - INF(:,:,:): lower confidence band (H horizons, N variables, N shocks)
-
- %   - SUP(:,:,:): upper confidence band (H horizons, N variables, N shocks)
-
- %   - MED(:,:,:): median response (H horizons, N variables, N shocks)
-
- %   - BAR(:,:,:): mean response (H horizons, N variables, N shocks)
- % ---
- % EXAMPLE
-

```
%   - See VARToolbox_Code.m in "../Primer/"
% =======================================================================
% VAR Toolbox 3.0
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated November 2020
% -----------------------------------------------------------------------
%% Check inputs
%------------------------------------------------------------------------
if ~exist('VAR','var')
    error('You need to provide VAR structure, result of VARmodel');
end
if ~exist('VARopt','var')
    error('You need to provide VAR options (VARopt from VARmodel)');

end

%% Retrieve and initialize variables
%------------------------------------------------------------------------
nsteps = VARopt.nsteps;
ndraws = VARopt.ndraws;
pctg   = VARopt.pctg;
method = VARopt.method;
Ft      = VAR.Ft;  % this if \Phi' in the notes (rows are coeffs, columns are eqs)
nvar    = VAR.nvar;
nvar_ex = VAR.nvar_ex;
nlag    = VAR.nlag;
nlag_ex = VAR.nlag_ex;
const   = VAR.const;
nobs    = VAR.nobs;
resid   = VAR.resid;
ENDO    = VAR.ENDO;
EXOG    = VAR.EXOG;
```

```
IV = VAR.IV;

INF = zeros(nsteps,nvar,nvar);
SUP = zeros(nsteps,nvar,nvar);
MED = zeros(nsteps,nvar,nvar);
BAR = zeros(nsteps,nvar,nvar);
%% Create the matrices for the loop
%-------------------------------------------------------------------------
y_artificial = zeros(nobs+nlag,nvar);
%% Loop over the number of draws
%-------------------------------------------------------------------------
tt = 1; % numbers of accepted draws
ww = 1; % index for printing on screen
while tt<=ndraws
    % Display number of loops
    if tt==VARopt.mult*ww
        disp(['Loop ' num2str(tt) ' / ' num2str(ndraws) ' draws'])
        ww=ww+1;

end

%% STEP 1: choose the method and generate the residuals
    if strcmp(method,'bs')
        % Use the residuals to bootstrap: generate a random number bounded
        % between 0 and # of residuals, then use the ceil function to select
        % that row of the residuals (this is equivalent to sampling with
replacement)
        u = resid(ceil(size(resid,1)*rand(nobs,1)),:);
    elseif strcmp(method,'wild')
        % Wild bootstrap based on simple distribution (~Rademacher)
        if strcmp(VARopt.ident,'iv')
            rr = 1-2*(rand(nobs,size(IV,2))>0.5);
            u = resid.*(rr*ones(size(IV,2),nvar));
            Z = [IV(1:nlag,:); IV(nlag+1:end,:).*rr];

else

end else

        error(['The method ' method ' is not available'])
    end
%% STEP 2: generate the artificial data
    %% STEP 2.1: initial values for the artificial data
    % Intialize the first nlag observations with real data
    LAG=[];
    for jj = 1:nlag
        y_artificial(jj,:) = ENDO(jj,:);
        LAG = [y_artificial(jj,:) LAG];

end

    % Initialize the artificial series and the LAGplus vector
    T = [1:nobs]';
    if const==0
        LAGplus = LAG;
    elseif const==1
        LAGplus = [1 LAG];
    elseif const==2
        LAGplus = [1 T(1) LAG];
    elseif const==3
        T = [1:nobs]';
        LAGplus = [1 T(1) T(1).^2 LAG];
    end
    if nvar_ex~=0
        LAGplus = [LAGplus VAR.X_EX(jj-nlag+1,:)];
```

```
end

    %% STEP 2.2: generate artificial series
    % From observation nlag+1 to nobs, compute the artificial data
    for jj = nlag+1:nobs+nlag
        for mm = 1:nvar
            % Compute the value for time=jj
            y_artificial(jj,mm) = LAGplus * Ft(1:end,mm) + u(jj-nlag,mm);

end

        % now update the LAG matrix
        if jj<nobs+nlag
            LAG = [y_artificial(jj,:) LAG(1,1:(nlag-1)*nvar)];
            if const==0
rr = 1-2*(rand(nobs,1)>0.5);
u = resid.*(rr*ones(1,nvar));
                LAGplus = LAG;
            elseif const==1
                LAGplus = [1 LAG];
            elseif const==2
                LAGplus = [1 T(jj-nlag+1) LAG];
            elseif const==3
                LAGplus = [1 T(jj-nlag+1) T(jj-nlag+1).^2 LAG];
            end
            if nvar_ex~=0
                LAGplus = [LAGplus VAR.X_EX(jj-nlag+1,:)];

end end

end

%% STEP 3: estimate VAR on artificial data.
    if nvar_ex~=0
        [VAR_draw, ~] = VARmodel(y_artificial,nlag,const,EXOG,nlag_ex);

else

end

    % If "iv" identification is selected, update VAR_draw with bootstrapped
    %instrument
    if exist('Z','var')
        VAR_draw.IV = Z;

end

%% STEP 4: calculate "ndraws" impulse responses and store them
    % Uses options from VARopt and parameters from VAR_draw (from step 3)
    % to compute IRFs
    [IR_draw, VAR_draw] = VARir(VAR_draw,VARopt);
    if VAR_draw.maxEig<.9999
        IR(:,:,:,tt) = IR_draw;

tt=tt+1;

end end

disp('-- Done!');
disp(' ');
%% Compute the error bands
%------------------------------------------------------------------------
pctg_inf = (100-pctg)/2;
pctg_sup = 100 - (100-pctg)/2;
```

```
INF(:,:,:) = prctile(IR(:,:,:,:),pctg_inf,4);
SUP(:,:,:) = prctile(IR(:,:,:,:),pctg_sup,4);
MED(:,:,:) = prctile(IR(:,:,:,:),50,4);
BAR(:,:,:) = mean(IR(:,:,:,:),4);
[VAR_draw, ~] = VARmodel(y_artificial,nlag,const);
```

```
function VARirplot(IR,VARopt,INF,SUP)
% =======================================================================
% Plot the IRs computed with VARir
% =======================================================================
% VARirplot(IR,VARopt,vnames,INF,SUP)
% -----------------------------------------------------------------------
% INPUT
```

- %   - IR(:,:,:) : matrix with IRF (H horizons, N variables, N shocks)
-
- %   - VARopt: options of the VAR (see VARopt from VARmodel)
- % ---
- % OPTIONAL INPUT
-
- %   - INF: lower error band
-
- %   - SUP: upper error band
- % ---
- % EXAMPLE
-

```
%   - See VARToolbox_Code.m in "../Primer/"
% =======================================================================
% VAR Toolbox 3.0
% Ambrogio Cesa-Bianchi
% ambrogiocesabianchi@gmail.com
% March 2012. Updated April 2021
% -----------------------------------------------------------------------
%% Check inputs
%===============================================
if ~exist('VARopt','var')
    error('You need to provide VAR options (VARopt from VARmodel)');
end
% If there is VARopt get the vnames
vnames = VARopt.vnames;
% Check they are not empty
if isempty(vnames)
    error('You need to add label for endogenous variables in VARopt');

end

% Define shock names
if isempty(VARopt.snames)
    snames = VARopt.vnames;

else

end

%% Retrieve and initialize variables
%===============================================
filename = [VARopt.figname 'IR_'];
quality = VARopt.quality;
suptitle = VARopt.suptitle;
pick = VARopt.pick;
% Initialize IR matrix
[nsteps, nvars, nshocks] = size(IR);
% If one shock is chosen, set the right value for nshocks
if pick<0 || pick>nvars
```

```
    error('The selected shock is non valid')
else
    if pick==0

pick=1;

else

end end

% Define the rows and columns for the subplots
row = round(sqrt(nvars));
col = ceil(sqrt(nvars));
snames = VARopt.snames;
nshocks = pick;
% Define a timeline
steps = 1:1:nsteps;
x_axis = zeros(1,nsteps);
%% Plot
%=================================================
SwatheOpt = PlotSwatheOption;
SwatheOpt.marker = '*';
SwatheOpt.trans = 1;
FigSize(VARopt.FigSize(1),VARopt.FigSize(2))
for jj=pick:nshocks
    for ii=1:nvars
        subplot(row,col,ii);
        plot(steps,IR(:,ii,jj),'LineStyle','-
','Color','k','LineWidth',2,'Marker',SwatheOpt.
marker); hold on
        if exist('INF','var') && exist('SUP','var')
            PlotSwathe(IR(:,ii,jj),[INF(:,ii,jj) SUP(:,ii,jj)],SwatheOpt); hold on;
        end
        plot(x_axis,'--k','LineWidth',0.5); hold on
        xlim([1 nsteps]);
        title([vnames{ii} ' to ' snames{jj}], 'FontWeight','bold','FontSize',10);
        set(gca, 'Layer', 'top');

end

% Save

    FigName = [filename num2str(jj)];
    if quality
        if suptitle==1
            Alphabet = char('a'+(1:nshocks)-1);
            SupTitle([Alphabet(jj) ') IR to a shock to '  vnames{jj}])

end

        set(gcf, 'Color', 'w');
        export_fig(FigName,'-pdf','-painters')
    else
        print('-dpdf','-r100',FigName);
    end
    clf('reset');
end

close all
```

```
function [AIC, SBC, logL] = VARlag(ENDO,maxlag,const,EXOG,lag_ex)
% =======================================================================
% Determine VAR lag length with Akaike (AIC) and Schwarz Bayesian
% Criterion (SBC)criterion.
% =======================================================================
% [AIC, SBC, logL] = VARlag(ENDO,maxlag,const,EXOG,lag_ex)
% -----------------------------------------------------------------------
% INPUT
%
%
%
% -----------------------------------------------------------------------
% OPTIONAL INPUT
- ENDO: an (nobs x nvar) matrix of endogenous variables.
- maxlag: the maximum lag length over which Akaike information
criterion is computed
```

- `%    - const: 0 no constant; 1 constant ; 2 constant and trend;`
-
- `%            3 constant and trend^2; [dflt = 1]`
-
- `%    - EXOG: optional matrix of variables (nobs x nvar_ex)`
-
- `%    - nlag_ex: number of lags for exogeonus variables (dflt = 0)`
- `% -----------------------------------------------------------------------`
- `% OUTPUT`
-
- `%    - AIC: preferred lag lenghth according to AIC`
-
- `%    - SBC: preferred lag lenghth according to SBC`
-
- `%    - logL: vector [maxlag x 1] of loglikelihood`
- `% -----------------------------------------------------------------------`
- `% EXAMPLE`
-
- `%  x=[12;34;56;78;910];`
- `%    OUT = VARmakelags(x,2)`
- `% =======================================================================`
- `% VAR Toolbox 3.0`
- `% Ambrogio Cesa-Bianchi`
- `% ambrogiocesabianchi@gmail.com`
- `% March 2012. Updated November 2020`
- `% -----------------------------------------------------------------------`
- `%% Check inputs`
- `%==========================================================`
- `[nobs, ~] = size(ENDO);`
- `% Check if ther are constant, trend, both, or none`
- `if ~exist('const','var')`
- `    const = 1;`
- `end`
- `% Check if there are exogenous variables`
- `if exist('EXOG','var')`
- `    [nobs2, num_ex] = size(EXOG);`
- `    % Check that ENDO and EXOG are conformable`
- `    if (nobs2 ~= nobs)`

```
•           error('var: nobs in EXOG-matrix not the same as y-matrix');
•       end
•       clear nobs2
•   else

    num_ex = 0;

    end

    % Check if there is lag order of EXOG, otherwise set it to 0
    if ~exist('lag_ex','var')
        lag_ex = 0;

    end

    % number of exogenous variables per equation
    nvar_ex = num_ex*(lag_ex+1);
    %% Compute log likelihood and Akaike criterion
    %==========================================================
    logL = zeros(maxlag,1);
    AIC  = zeros(maxlag,1);
SBC  = zeros(maxlag,1);
for i=1:maxlag
    X = ENDO(maxlag+1-i:end,:);
    aux = VARmodel(X,i,const);
    if nvar_ex>0
        Y = EXOG(maxlag+1-i:end,:);
        aux = VARmodel(X,i,const,Y,lag_ex);

end

    NOBSadj = aux.nobs;
    NOBS = aux.nobs + i;
    NVAR = aux.nvar;
    NTOTCOEFF = aux.ntotcoeff;
    RES = aux.resid;
    % VCV of the residuals (use dof adjusted denominator)
    SIGMA = (1/(NOBSadj)).*(RES)'*(RES);
    % Log-likelihood
    logL(i) = -(NOBS/2)* (NVAR*(1+log(2*pi)) + log(det(SIGMA)));
    % AIC: �2*LogL/T + 2*n/T, where n is total number of parameters (ie,
NVAR*NTOTCOEFF)
    AIC(i) = -2*(logL(i)/NOBS) + 2*(NVAR*NTOTCOEFF)/NOBS;
    % SBC: �2*LogL/T + n*log(T)/T
    SBC(i) = -2*(logL(i)/NOBS) + (NVAR*NTOTCOEFF)*log(NOBS)/NOBS;

end

% Find the min of the info criteria
AIC = find(AIC==min(AIC));
SBC = find(SBC==min(SBC));
```