Predictive and Structural Analysis for High-Dimensional Vector Autoregressive Models Estimated by Regularization

Erasmus School of Economics

MSc. Quantitative Finance

Author: O. Zanky

Supervisor: dr. A.M. Schnücker

Second Assessor: dr. J.W.N. Reuvers

January 6, 2020

Abstract

This paper tackles the problem of estimating a high-dimensional vector autoregression (VAR). The estimation of these high-dimensional systems is done via regularization procedures that select the model and estimates the parameters simultaneously and is particularly useful in vector autoregressive context. This paper builds on the penalized least square procedures proposed by Nicholson et al. (2018). All procedures consist of penalty functions that are made up of hierarchically nested Euclidean norms of the model-coefficients. I augment these regularization models with a function that increases with the variables' lag, incorporating the temporal dependence of the VAR more accurately. Moreover, I propose a new regularization model that is able to estimate high-dimensional VARs. The efficacy of the procedures, both in terms of forecasting and model discovery, is demonstrated in a simulation study as well as an empirical study. In addition, I show that high-dimensional VARs estimated by the proposed regularization methods produce credible impulse responses and are suitable for structural analysis.

Keywords: Vector Autoregression; Regularization; Forecasting; Structural Analysis

Contents

1 Introduction

VARs, introduced by Sims (1980), are linear multivariate time series models able to capture the joint dynamics of multiple time series. The VAR is a useful tool by which out-of-sample forecasts in macroeconomics can be constructed. VARs are especially useful for impulse response analysis, which is a structural analysis technique that forecasts how the variables change after a shock to one variable while holding all other shocks constant. As of now, the VAR is the workhorse model for macroeconomic data. In contrast to many macroeconomic models used prior to the seminal work of Sims (1980), VARs do not impose strict identifying restrictions on the parameters, resulting in a very general representation that is able to capture complex temporal and cross-sectional relationships among the time series. A drawback of this high level of flexibility is that it necessitates a large number of parameters, even for moderate-dimensional systems, which is an obstacle plaguing contemporary real-world practitioners in their application of VARs.

A $\text{VAR}_k(p)$ model is a stationary k-dimensional vector time series that is modeled in terms of its previous p values. A straightforward way of estimating this system is via least squares. It can quite handily be shown that applying the least squares on a VAR is equivalent to applying least squares separately to each equation of the VAR, which makes it a very convenient and computation-efficient method. In low-dimensional settings, in which pk, the amount of variables in each equation, is small relative to the sample length T, the parameters may be accurately estimated using least squares. However, in the case that pk is nearly equal but still smaller than T, there will be a lot of variability in the least squares estimates, resulting in an unreliable estimated model. Moreover, in the case that $pk \geq T$, least squares is not even feasible as the solutions are not unique anymore. Alternative estimation methods that are able to deal with the case $pk \geq T$ will often run into computation cost issues as pk^2, the total amount of parameters in a VAR, grows too large. Hence, building a high-dimensional VAR that features a large number of variables in any case poses a great challenge. This phenomenon of encountering difficulties as the dimensionality of a problem increases has been termed the 'curse of dimensionality' (Keogh and Mueen, 2017). This curse expeditiously occurs in VARs as an addition of a single variable to the system leads to a quadratic increase of the dimension of the parameter space. As a consequence of this dimensionality issue, a lot of research has conducted regarding estimation methods that identify a subset of predictors and their corresponding lags in a vector autoregressive context.

This thesis builds on that ongoing research of reducing dimensionality of a VAR in order to obtain a reliable estimated model and the ability to construct accurate forecasts. Recent developments in the literature have produced methodologies that incorporate regularized estimation techniques for variable selection in VARs. It is via this approach that I estimate high-dimensional VARs. Specifically, the regularization models developed by Nicholson et al. (2018) are used. These models incorporate relevant information (temporal dynamics and spatial dependence) that VARs inherently provide. To capture the temporal aspect of VARs better, I augment the models by including a lag-function that increases with the parameters' lag. Moreover, I propose a new regularization model, not found in Nicholson et al. (2018). The predictive strength of each regularization procedure is compared to one another in an extensive Monte Carlo analysis. In addition to studying the predictive power of each procedure, their capacity in uncovering the true model is also analyzed. The procedures are estimated for various time series length. The regularization procedures are also applied on the empirical macroeconomic dataset introduced by Stock and Watson (2005). In this empirical study the predictive performance of the various procedures are again compared, and via structural analysis I examine if economically sound interpretations can be derived from the estimated models.

Modeling VARs for macroeconomic data is particularly challenging as macroeconomic data is usually measured exclusively in low frequencies. An example of this is the dataset used in this paper, namely the dataset found in Stock and Watson (2005), which contains 168 macroeconomic variables measured over 193 quarters. In such contexts high-dimensional VARs are more susceptible to the curse of dimensionality. It is for this reason that researchers were forced to limit the size of their VARs for macroeconomic applications by excluding variables from the system. This typically led to an omitted variables bias. Omitting (relevant) variables in a VAR has adverse consequences for both forecasting and structural analysis. On top of that, jointly estimating high-dimensional time series is increasingly more important in an ever-globalizing world, where economic developments in different regions often propagate and affect one another, substantiating the necessity for viable estimation techniques for high-dimensional VARs. The importance of incorporating relevant variables in order to obtain accurate inference and out-of-sample forecasts in a statistical model, is quite straightforward. However, its importance in structural analysis may seem a bit less obvious.

By omitting variables from a VAR, impulse responses will often become distorted and as a consequence be worthless for structural analysis. A popular example of the importance of not excluding relevant variables in structural analysis has been the issue surrounding the 'price puzzle',

which is a term coined by Eichenbaum (1992) describing a phenomenon in which there is a rise in the aggregate price level in response to a contractionary monetary policy. The price-puzzle is often found in low dimensional VARs, as exhibited in Christiano et al. (1999). The literature has often argued that this is as a result of omitting forward looking variables, like the commodity price index, in a VAR. However, later literature, like Hanson (2004), have disputed on whether adding variables – including commodity prices – solves the prize-puzzle. Regardless, this phenomenon is briefly examined in this work.

Other than regularization, a number of methods have been proposed for reducing the dimensionality of the VAR models. These methods include canonical analysis (Box and Tiao, 1977), scalar component models or canonical correlation analysis (Tiao and Tsay, 1989), principal component analysis (Stock and Watson, 2002), dynamic orthogonal components analysis (Matteson and Tsay, 2011), Bayesian VARs (Banbura et al., 2010), and dynamic factor models (Forni et al., 2000, Barigozzi et al., 2016).

A particular challenge that has been scantly researched in VAR literature is to reduce dimensionality of VARs taking in consideration both forecast accuracy and intrepatability of the estimated model. Interpretability in the context of this thesis means that the estimated model itself should give information about the joint dynamics of the dependent variables, and that the sparsity in the parameter space is induced in a structured way – allowing for economic interpretation. This translates to applying a method that will both reduce the dimensionality of the VAR, while also performing automatic lag selection, which is the most natural way of inducing structured sparsity in a VAR.

Specifying an appropriate lag length is crucial in time series, as including too many lags of the dependent variables can easily lead to overfitting, which in turn results in a decrease in prediction power of the estimated model. In contrary, selecting a lower order lag length than the true lag length, may result in serial correlation in the error terms, and consequently result in an underestimation of the standard errors of the least square parameter estimates. Currently, most VARs are estimated using symmetric lag choice procedures (see, among others, Lütkepohl, 1985, Gonzalo and Pitarakis, 2002), i.e. the same lag length is used for all variables in all equations of the model. Implicitly, these approaches rely on the restrictive assumption of one lag order that applies across all components. However, there is no compelling economic nor statistical justification for such a strong assumption. In fact, there is evidence to the contrary. Asymmetric lags are often more suitable for macroeconomic analysis, in particular in a high dimensional setting. Hsiao (1981) was the first who examined

the possibility of asymmetric lag VAR models by proposing an extensive iterative procedure to appropriately specify a lag structure. Later on, Keating (1993) introduced asymmetries in the lag lengths of the variables in the VAR and termed this as the asymmetric VAR. More recently, Bayesian methods (see Ding and Karlsson, 2014, Albis and Mapa, 2014), and regularized regressions have been applied to identify the lag structure of a VAR, where the latter of the two approaches is the central theme of this paper.

The regularized regression that can achieve the goal of asymmetric lag length selection is the least absolute shrinkage and selection operator (lasso) method introduced by Tibshirani (1996). As the name of the method implies, the lasso is a regression analysis method that simultaneously performs both variable selection and parameter shrinkage. Hsu et al. (2008) derives theoretical properties of the lasso under a vector autoregressive process and shows in both a simulation study as well as an empirical study the lasso's superior forecast performance compared to using statistical information criteria for model selection and forecasting. Later work by Song and Bickel (2011) applies the lasso on VARs too, but they adjust the penalty function of the lasso as introduced by Tibshirani (1996), and postulate a regularization term that is VAR-specific, thereby making use of the information that the structure of a VAR provides. The VAR-specific lasso result in an improvement in forecasting and variable selection relative to the traditional lasso. As modeling with (structured) sparsity has become an increasingly important topic in many fields in recent years, a very useful book by Hastie et al. (2015) summarizes and explains much of the recent literature regarding the lasso and modeling with sparsity in general.

As stated earlier, this thesis' methodology builds on the paper by Nicholson et al. (2018), which in turn is primarily built on the works of Song and Bickel (2011) and Nicholson et al. (2017). They introduce the concept of hierarchical lag structures (HLag), which is a class of regularizes specifically designed for VARs. The HLag structures conveniently embed the notion of lag order selection in their penalty functions which produces estimated models with low lag orders. Nicholson et al. (2018) show that the traditional lasso estimator often selects too many higher order parameters and results in difficult to interpret estimated models. They thusly conclude that the HLag structures are advantageous in a (high-dimensional) time series context compared to other tried regularization procedures.

The results obtained in this paper corroborate much of the results obtained Nicholson et al. (2018). In the empirical application, the most robust – robust in the sense of widely applicable –

performing HLag structure, with mean squared forecast error (MSFE) as the performance measure, is a structure that consider a component's own lag separately from the lag of other variables. The lag-function that I add into the HLag structures, in both the simulation and empirical study, does little in improving the out-of-sample forecasts compared to incorporating that lag-function. However, the sparsity, i.e., the amount of parameters that are set to zero in the estimated model, increases if a lag-function is present. This implies that there is no strict relationship between how much sparsity is generated into an estimated regression model and how accurate an estimated model's out-of-sample forecasts is. In the simulation study the procedures are estimated using various time series length. In particular if T is small, there are some discrepancies across the procedures. As T increases, such that $T \gg kp$, these discrepancies fade away. The parameter estimates do not converge towards their true values, i.e., they are inconsistent. The parameter inconsistency of the lasso-based methods is not a surprise and is extensively discussed in Zou (2006). The results of empirical study show that the impulse response function of each of the HLag methods are observed to be roughly the same across the different HLag structures. Despite the parameter inconsistency of the lasso methods, the economic interpretation that can be derived from the impulse responses is mostly sound. However, the price-puzzle does exhibit for all procedures. Adding more variables to the system does decrease the prize-puzzle effect but it does not fade away completely.

The road map of this thesis is as follows: In Section 2 I explain my proposed methodology. The specification of weighting penalty parameter is detailed in Section 2.1. The explanation of the hierarchical lasso, the key modeling tool of the HLag structures, is detailed in Section 2.2. In Section 2.3 the motivation and theoretical suitability of each weighted HLag procedure is elaborated upon. The optimization algorithm, with which all regularization problems in thesis are solved with, is explained in Section 2.4, and in Section 2.5 the procedure that selects the optimal tuning parameters is explained. With an understanding of the weighted HLag procedures and how they are to be solved, a simulation study in Section 3 enables comparison of each HLag method in various simulation set-ups. In Section 4 an empirical analysis takes place, allowing for examination in the weighted HLags' performance in real-world applications. Finally, in Section 5 the thesis is concluded and suggestions for researcher in future work regarding this topic are proposed.

2 Methodology

Prior to describing the weighed HLag structures, it is necessary to formally define the VAR and some notation that is used in this paper. Consider a $\text{VAR}_k(p)$ model, with $k \in \mathbb{N}^+$ the amount of dependent variables and $p \in \mathbb{N}^+$ the lag length. This model may be expressed as

$$\mathbf{y}_t = \mathbf{c} + \sum_{\ell=1}^{p} \mathbf{\Phi}^{(\ell)} \mathbf{y}_{t-\ell} + \mathbf{u}_t, \quad \text{with} \quad \mathbf{u}_t \overset{\text{wn}}{\sim} (\mathbf{0}, \mathbf{\Sigma}_u) \quad \text{for} \quad t = 1, \dots, T, \tag{2.1}$$

where $\{\mathbf{y}_t \in \mathbb{R}^k\}_{t=1}^T$ are the dependent variables, $\mathbf{c} \in \mathbb{R}^k$ is a vector of intercepts, $\{\mathbf{\Phi}^{(\ell)} \in \mathbb{R}^{k \times k}\}_{\ell=1}^p$ are the coefficient matrices, and $\{\mathbf{u}_t \in \mathbb{R}^k\}_{t=1}^T$ are the error terms.

In describing the regularization structures later on, I will use the notational convention found in Nicholson et al. (2018): for $1 \leq \ell \leq p$, let

$$\begin{aligned}
\mathbf{\Phi} &= (\mathbf{\Phi}^{(1)}, \dots, \mathbf{\Phi}^{(p)}) \in \mathbb{R}^{k \times kp} \\
\mathbf{\Phi}^{(\ell:p)} &= (\mathbf{\Phi}^{(\ell)}, \dots, \mathbf{\Phi}^{(p)}) \in \mathbb{R}^{k \times k(p-\ell+1)} \\
\mathbf{\Phi}_i^{(\ell:p)} &= (\mathbf{\Phi}_i^{(\ell)}, \dots, \mathbf{\Phi}_i^{(p)}) \in \mathbb{R}^{1 \times k(p-\ell+1)} \\
\mathbf{\Phi}_{ij}^{(\ell:p)} &= (\mathbf{\Phi}_{ij}^{(\ell)}, \dots, \mathbf{\Phi}_{ij}^{(p)}) \in \mathbb{R}^{1 \times (p-\ell+1)} \\
\mathbf{\Phi}_{i,-i}^{(\ell:p)} &= (\mathbf{\Phi}_{i,-i}^{(\ell)}, \dots, \mathbf{\Phi}_{i,-i}^{(p)}) \in \mathbb{R}^{1 \times (k-1)(p-\ell+1)},
\end{aligned} \tag{2.2}$$

with $\mathbf{\Phi}_{i,-i}^{(\ell)} = (\mathbf{\Phi}_{i1}^{(\ell)}, \dots, \mathbf{\Phi}_{i,i-1}^{(\ell)}, \mathbf{\Phi}_{i,i+1}^{(\ell)}, \dots, \mathbf{\Phi}_{ik}^{(\ell)}) \in \mathbb{R}^{1 \times (k-1)}$.

In describing the HLag methods, it is useful to introduce the lag matrix

$$\mathbf{L}_{ij} = \max\{\ell : \mathbf{\Phi}_{ij}^{(\ell)} \neq 0\},$$

in which $\mathbf{L}_{ij} = 0$ if $\mathbf{\Phi}_{ij}^{(\ell)} = 0$ for all $\ell = 1, \dots, p$. The integer $\mathbf{L}_{ij}$ denotes the maximal parameter lag for independent variable j in the equation of dependent variable i.

Finally, the L_q-norm for matrices is defined as

$$\|\mathbf{\Phi}\|_q = \left(\sum_{ij} |\mathbf{\Phi}_{ij}|^q \right)^{\frac{1}{q}} \quad \text{for} \quad q = 1, 2, \dots.$$

The L_q-norm is defined analogously for vectors, except the summation is taking over all elements of the vector.

2.1 Lag-Weighted Lasso

A general representation of regularized least squares of a VAR is given by

$$\hat{\mathbf{\Phi}}, \hat{\mathbf{c}} = \underset{\mathbf{\Phi}, \mathbf{c}}{\arg\min} \left\{ \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{c} - \sum_{\ell=1}^{p} \mathbf{\Phi}^{(\ell)} \mathbf{y}_{t-\ell}\|_2^2 + \mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) \right\}, \tag{2.3}$$

where $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)$ is a penalty function for the coefficients, and λ and γ are tuning parameters that control the strength of the penalty term. In regularization problems, the intercept $\hat{\mathbf{c}}$ is typically not regularized, as it can be derived separately. I omit this derivation, however using basic vector calculus it can be acquired that

$$\hat{\mathbf{c}} = \bar{\mathbf{y}} - \sum_{\ell=1}^{p} \hat{\mathbf{\Phi}}^{(\ell)} \bar{\mathbf{y}}_\ell, \tag{2.4}$$

where $\bar{\mathbf{y}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{y}_t$ and $\bar{\mathbf{y}}_l = \frac{1}{T} \sum_{t=1}^{T-\ell} \mathbf{y}_t$. The penalty function $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)$ determines the solution space of $\hat{\mathbf{\Phi}}$, and depending on its postulation, may produce a vastly different estimated model.

A basic lasso penalty function takes the form

$$\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{\ell=1}^{p} f(\ell; \gamma) \|\mathbf{\Phi}^{(\ell)}\|_1, \tag{2.5}$$

where $f(\ell; \gamma) > 0$ is function determining the lag-effects. This specification is a lag-weighted variation of the traditional L_1-norm penalty function. The lasso penalty function expressed in (2.5) will generate sparsity in $\mathbf{\Phi}$ by setting individual parameters to zero. The obvious downside to this penalty function is the fact that the sparsity induced will be completely unstructured. On the other hand, the lasso term is augmented with a weighting specification, therefore incorporating the time dependence that is embedded in VARs. Under assumption of stationarity, the magnitude of previous variables will reduce to zero with increasing lag length. Hence, by incorporating a function of ℓ into the regularizer should improve forecast accuracy and model selection. The concept of a lag-weighted lasso method predates this paper. For example, Song and Bickel (2011) incorporated weights that geometrically increase with lag order, and Park and Sakaori (2013) propose a lag weighted lasso routine with weights that relate both to the magnitude of the parameters and the lag effect of univariate time series model.

A desirable property of the lag-function for time series that do not exhibit seasonality is that $f(\ell; \gamma)$ increases as ℓ increases. This means that recent lags are penalized less relative to distant lag, which corresponds to intuitive belief that recent information is more important than distant information. In the weighted lasso setting, the weights are used to control the strength of the penalty

of each variable by simply postulating a functional form. The two downsides to this approach are the addition of another tuning parameter that needs to be determined (therefore resulting in an increase in computation time), and that one assumes a particular functional form. However, the latter downside is quite benign, as imposing a functional form, instead of estimating $f(\ell)$ for each ℓ from the data, may prevent overfitting.

In this paper a functional form will be assumed. This functional form is the same one that was postulated by Song and Bickel (2011) and is it specified as

$$f(\ell;\gamma) = \ell^{\gamma} \quad \text{with} \quad \gamma \in [0,1]. \tag{2.6}$$

The tuning parameter γ governs the relative importance of distant lags with respect to the more recent ones. As stated, other (more complex) approaches may be interesting to explore, however, as the lag-function in (2.6) has been tried and tested in an earlier work – and good results were achieved with it – I decide on this lag-function.

2.2 Hierarchical Lasso

The HLag framework is based on the hierarchical lasso as introduced by Zhao et al. (2009). The hierarchical lasso is an extension of the standard group lasso, which was introduced some time earlier by Yuan and Lin (2006). To fully appreciate the hierarchical lasso, it is useful to be familiar with the group lasso.

The group lasso assumes the existence of a group structure of the independent variables and that it is desirable to shrink all parameters of that group simultaneously. All parameters within a group will either be set to zero or it will be 'active' and all parameters will be shrunk but not equal to zero. A general representation of a lag-weighted group lasso is specified by

$$\mathcal{P}(\mathbf{\Phi};\lambda,\gamma) = \lambda \sum_{\ell=1}^{p} \ell^{\gamma} \|\mathbf{\Phi}^{g_\ell}\|_2, \tag{2.7}$$

where the superscript g_ℓ specifies the group structure of $\mathbf{\Phi}$ at each ℓ. The reason why the group lasso sets groups of parameters to zero (or have them remain non-zero) can be derived from the formulation above, namely the L_2-norm of a matrix (or vector) is zero if and only if all of its element are zero. Specifying the regularizer in this manner may not only help with obtaining more accurate parameter estimates, but it may also improve the interpretability of the model. The group lasso is closely related to the famous ridge regression introduced by Hoerl and Kennard (1970). Both

apply the L_2-norm in their regularizers. The difference lies in that group lasso is a sum over several L_2-norms, with each norm containing a subset of the parameters in the model. Ridge regression, which was never intended for parameter selection, takes the L_2-norm over every parameter, resulting in an estimated model with shrunk parameters but without sparsity.

Song and Bickel (2011) apply the group lasso in a VAR context, and group the parameters by which lag they correspond to, i.e., they simply create groups of $\mathbf{\Phi}$ by setting

$$\mathbf{\Phi}^{g_1} = \mathbf{\Phi}^{(1)}, \ldots, \mathbf{\Phi}^{g_p} = \mathbf{\Phi}^{(p)}. \tag{2.8}$$

While this structure is advantageous for applications in which all component series tend to exhibit comparable dynamics, it fails to take the temporal structure of stationary time series in account, i.e., it neglects the fact that as the lag order increases the parameter magnitude (usually) decreases in a time series context.

The HLag structures, in contrary to the group lasso, do take the temporal structure of the VAR in account. They embed this structure into their penalty functions by creating nested groups out of $\mathbf{\Phi}$. Nested groups imply regularization constraints in which one group of coefficients being zero necessitates that another group of coefficients is zero. An application of this would be to let

$$\mathbf{\Phi}^{g_p} \subseteq \cdots \subseteq \mathbf{\Phi}^{g_1}. \tag{2.9}$$

This nested structure implies that if $\mathbf{\Phi}^{g_{\ell'}} = \mathbf{0}$ it would necessarily follow that $\mathbf{\Phi}^{g_\ell} = \mathbf{0}$ for any $\ell' > \ell$. In contrary, if $\mathbf{\Phi}^{g_\ell} = \mathbf{0}$ it would not imply that $\mathbf{\Phi}^{g_{\ell'}} = \mathbf{0}$. Such a structure guarantees that the sparsity pattern of lag parameters honors the VAR's ordered temporal structure. This is in contrast with grouping the variables as in (2.8). In that case it is entirely possible that, for $\ell' > \ell$, $\mathbf{\Phi}^{g_{\ell'}} \neq \mathbf{0}$ even if $\mathbf{\Phi}^{g_\ell} = \mathbf{0}$. The downside to a nested grouping structure is that seasonal patterns in time series are not able to be captured. Regardless, as many economic time series do not exhibit significant seasonal patterns, and seasonal patters can be adjusted for, the HLag structures will have many use cases in practice. Moreover, if one encounters seasonality in their time series and wishes not to correct for it, one can decide to not impose a hierarchical structure on the penalty functions, i.e, instead of a nested structure as in (2.9) apply a structure that is a sum of non-overlapping L_2-norms of the coefficients as in (2.8).

2.3 Weighted HLag Structures

Table 1: **Penalty Functions**

The different penalty functions, that are applied in the simulation and empirical analysis of this paper, are summarized in this table. All penalty functions are HLag structures, except for the basic lasso penalty function. To recall the meaning of the matrix notations, refer back to (2.2).

Name	$\mathcal{P}(\mathbf{\Phi};\lambda,\gamma)$
(2.5) Lasso	$\lambda\sum_{\ell=1}^{p}\ell^{\gamma}\|\mathbf{\Phi}^{(\ell)}\|_1$
(2.10) Lagwise	$\lambda\sum_{\ell=1}^{p}\ell^{\gamma}\|\mathbf{\Phi}^{(\ell:p)}\|_2$
(2.11) Componentwise	$\lambda\sum_{i=1}^{k}\sum_{\ell=1}^{p}\ell^{\gamma}\|\mathbf{\Phi}_{i}^{(\ell:p)}\|_2$
(2.12) Elementwise	$\lambda\sum_{j=1}^{k}\sum_{i=1}^{k}\sum_{\ell=1}^{p}\ell^{\gamma}\|\mathbf{\Phi}_{ij}^{(\ell:p)}\|_2$
(2.13) Own-Other	$\lambda\sum_{i=1}^{k}\sum_{\ell=1}^{p}\ell^{\gamma}\left(\ell\|\mathbf{\Phi}_{ii}^{(\ell:p)}\|_2+\ell(k-1)\|\mathbf{\Phi}_{i,-i}^{(\ell:p)}\|_2\right)$

For convenience of the reader, all penalty functions that are applied in this paper are summarized in Table 1. The paper by Nicholson et al. (2018) introduces the framework of the HLag penalty structures that incorporate automatic lag selection into their regularizes. This paper builds on their framework by postulating a new HLag structure and augmenting all the HLag structures with the lag-function postulated in (2.6).

The first weighted HLag structure is the lagwise HLag, which is a nested modification of (2.8) and is specified as

$$\mathcal{P}(\mathbf{\Phi};\lambda,\gamma)=\lambda\sum_{\ell=1}^{p}\ell^{\gamma}\|\mathbf{\Phi}^{(\ell:p)}\|_2. \tag{2.10}$$

This is the HLag only structure that is not found in the paper by Nicholson et al. (2018) and is an addition by me. It relates to (2.9) by setting

$$\mathbf{\Phi}^{g_p}=\mathbf{\Phi}^{(p:p)}\subset\mathbf{\Phi}^{g_{p-1}}=\mathbf{\Phi}^{(p-1:p)}\subset\cdots\subset\mathbf{\Phi}^{g_1}=\mathbf{\Phi}^{(1:p)}.$$

Because of the lag-function ℓ^{γ}, greater regularization is applied to higher order lags. The weighted lagwise HLag is a straightforward extension of the group lasso penalty function found in Song and Bickel (2011), who group their matrices as specified in (2.8), i.e., without a nested hierarchical structure. For the weighted lagwise HLag it holds that $\mathbf{\Phi}^{\ell'}=\mathbf{0}$ if $\mathbf{\Phi}^{\ell}=\mathbf{0}$ for any $\ell'>\ell$. The lag matrix produced by this penalty function will be of the form of $\mathbf{L}=L\mathbf{J}_k$, where $\mathbf{J}_k$ is $k\times k$ matrix consisting of only ones and L the selected lag of all the components. Such a lag matrix implies that all variables are chosen to have the same lag, corresponding to symmetric lag choice procedures,

such as lag selection via information criteria. Unlike information criteria procedures, this symmetric HLag will also shrink the active coefficients in addition to selecting a symmetric lag order across all components. This structure is advantageous for applications in which all components exhibit very similar temporal dynamics.

Estimating a VAR with the restriction of a symmetric lag order across all components may not be most appropriate for VARs. To presume that all series have the exact same lag for a high dimensional VAR is more than often too limiting of a presumption of the true temporal dynamics of set of macroeconomic and or financial series. Thus, to be able to estimate varying lags for different series, the weighted compenentwise HLag is postulated as

$$\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{i=1}^{k} \sum_{\ell=1}^{p} \ell^{\gamma} \|\mathbf{\Phi}_i^{(\ell:p)}\|_2. \tag{2.11}$$

In this case, it holds that $\mathbf{\Phi}_i^{\ell'} = \mathbf{0}$ if $\mathbf{\Phi}_i^{\ell} = \mathbf{0}$ for any $\ell' > \ell$ for $i = 1, \ldots, k$. This penalty function will produce lag matrices of the form of $\mathbf{L}_i = L_i \boldsymbol{\iota}_k$ for $i = 1, \ldots, k$, with $\boldsymbol{\iota}_k \in \mathbb{R}^{1 \times k}$ being a vector of ones, and L_i is the selected lag for all the variables of component i. The lag lengths differs across the components but is the same for a particular component in each equation. This structure identifies the various lag lengths on different variables, an idea proposed by Keating (1993), and therefore is expected to give more a parsimonious estimated model relative to the lagwise HLag.

The componentwise HLag structure is still quite inflexible, and may perform badly if, within one equation, some variables are informative and other variables are uninformative. To allow for more flexibility, and capacity to select lags of each individual variable within an equation, the weighted elementwise hierarchical lag structure is formulated

$$\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{j=1}^{k} \sum_{i=1}^{k} \sum_{\ell=1}^{p} \ell^{\gamma} \|\mathbf{\Phi}_{ij}^{(\ell:p)}\|_2 \tag{2.12}$$

In this case it holds that $\mathbf{\Phi}_{ij}^{\ell'} = 0$ if $\mathbf{\Phi}_{ij}^{\ell} = 0$ for any $\ell' > \ell$ and for every $i, j = 1, \ldots, k$. The lag matrix in this case takes the form of $\mathbf{L}_{ij} = L_{ij}$, with L_{ij} the selected lag for component i and variable j. The lag lengths not only differs across the variables but also differs for a particular variable in each equation. Therefore, as long as the assumption of equal lag length for each equation breaks, the elementwise HLag could provide a sparser model and likely result in more precise estimations.

In many settings, it may not be appropriate to give equal consideration to every entry in a coefficient matrix $\mathbf{\Phi}$. Diagonal entries are often in macroeconomic application more likely to be higher in magnitude than off-diagonal entries. This idea is substantiated by Litterman (1986), who

in a Bayesian setting presented the traditional Minnesota prior. The prior covariance is set-up such that a variables' own lags are more informative than other lags. A parameter own's lag is therefore shrunk by a smaller factor than the parameters corresponding to other variables. The weighted own-other HLag is specified as

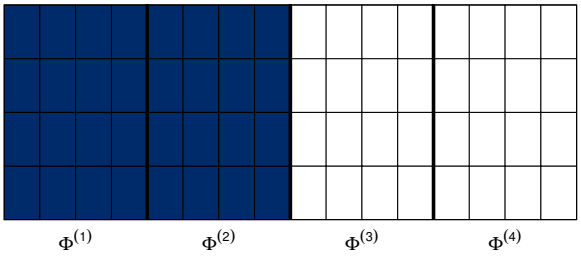
$$\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{i=1}^{k} \sum_{\ell=1}^{p} \ell^{\gamma} \left(\ell \|\mathbf{\Phi}_{ii}^{(\ell:p)}\|_2 + \ell(k-1) \|\mathbf{\Phi}_{i,-i}^{(\ell:p)}\|_2 \right) \tag{2.13}$$

This penalty function is a slightly modified version from the own-other HLag structure found in Nicholson et al. (2018). Unlike the previous penalty functions, the groups in this function differ in cardinality. Hence, weighting the the regularizer to avoid favoring larger groups is required. At each lag ℓ the terms are multiplied with their respective cardinalities, in case of the 'own' terms this equals to ℓ and in case of the 'other' terms this equals to $\ell(k-1)$. This structure implies that for $\ell' > \ell$, $\mathbf{\Phi}_{ii}^{\ell'} = 0$ implies $\mathbf{\Phi}_{ii}^{\ell} = 0$ and $\mathbf{\Phi}_{i,-i}^{\ell'} = \mathbf{0}$ implies $\mathbf{\Phi}_{i,-i}^{\ell} = \mathbf{0}$ for every $i = 1, \ldots, k$. The lag matrix that will be produced will be of the form $\mathbf{L}_{ii} = L_{ii}$ and $\mathbf{L}_{i,-i} = L_i^{\text{other}} \boldsymbol{\iota}_{k-1}$, with L_i^{other} a scalar corresponding to the selected lag of component i, excluding the lag of the variable corresponding to component i itself.

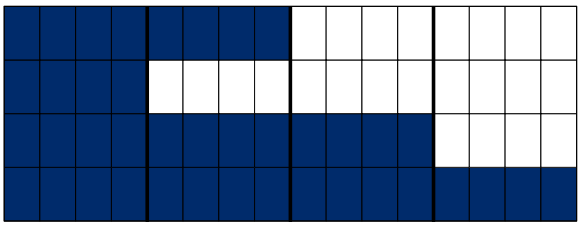
The different penalty functions will result in different sparsity patterns. To be able to graphically discern the different sparsity patterns, consider the example of a $\text{VAR}_4(4)$. In Figure 1 possible sparsity patterns induced by the different HLag structures are depicted.

Figure 1: **Sparsity Patterns of the HLag Structures**

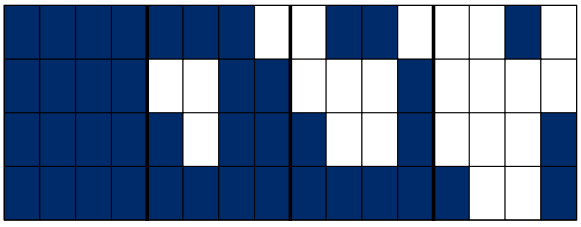
Illustrations of the sparsity patterns that may be induced by each of the four HLag penalty functions applied on a $\text{VAR}_4(4)$. For each illustration, the corresponding lag matrix is displayed next to it. From top to bottom, the HLag structures are: lagwise, componentwise, elementwise, own-other.



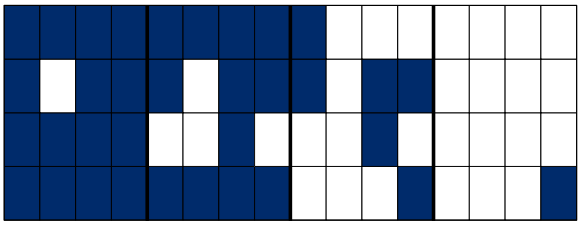
$$\mathbf{L} = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{pmatrix}$$



$$\mathbf{L} = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$



$$\mathbf{L} = \begin{pmatrix} 2 & 3 & 4 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 2 & 4 \\ 4 & 3 & 3 & 4 \end{pmatrix}$$



$$\mathbf{L} = \begin{pmatrix} 3 & 2 & 2 & 2 \\ 3 & 0 & 3 & 3 \\ 1 & 1 & 3 & 1 \\ 2 & 2 & 2 & 4 \end{pmatrix}$$

2.4 Optimization

Prior to describing the optimization algorithms for the regularization procedures, it is convenient to do away with the intercept term in (2.3). To this end, $\hat{\mathbf{c}}$, as specified in (2.4), is plugged in the least squares equation (2.3), obtaining the following:

$$\begin{aligned} \underset{\mathbf{\Phi}}{\arg\min} \left\{ \sum_{t=1}^{T} \|\mathbf{y}_t - \bar{\mathbf{y}} + \sum_{\ell=1}^{p} \hat{\mathbf{\Phi}}^{(\ell)} \bar{\mathbf{y}}_\ell - \sum_{\ell=1}^{p} \mathbf{\Phi}^{(\ell)} \mathbf{y}_{t-\ell} \|_2^2 \right\} = \\ \underset{\mathbf{\Phi}}{\arg\min} \left\{ \sum_{t=1}^{T} \|(\mathbf{y}_t - \bar{\mathbf{y}}) - \sum_{\ell=1}^{p} \hat{\mathbf{\Phi}}^{(\ell)} (\mathbf{y}_{t-\ell} - \bar{\mathbf{y}}_\ell) \|_2^2 \right\}. \end{aligned} \tag{2.14}$$

The result above shows that by temporally demeaning $\mathbf{y}_t$ the intercept term can be left out. For the analysis of this section the dependent variables are assumed to be demeaned and the

intercept term is consequently omitted. It is also convenient to express the least squares equation in 'compact' notation. Define $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T) \in \mathbb{R}^{k \times T}$, $\mathbf{Z}_t = (\mathbf{Y}'_{t-1}, \ldots, \mathbf{Y}'_{t-p})' \in \mathbb{R}^{kp \times 1}$, and let $\mathbf{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_T) \in \mathbb{R}^{kp \times T}$.

The regularized least squares in compact form solves the following convex problem

$$\arg\min_{\mathbf{\Phi}} \left\{ \|\mathbf{Y} - \mathbf{\Phi}\mathbf{Z}\|_2^2 + \mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) \right\}. \tag{2.15}$$

This minimization problem can be decomposed as a sum of two components. The first component is the loss function $\mathcal{L}(\mathbf{\Phi}) = \|\mathbf{Y} - \mathbf{\Phi}\mathbf{Z}\|_2^2$ and the second component is the penalty function $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)$. The loss function $\mathcal{L}(\mathbf{\Phi})$ is clearly convex and differentiable. The penalty term $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)$, in thesis, can take any of the forms as reported Table 1. The basic lasso penalty function is a sum of L_1-norms. The four HLag structures are all sums of nested L_2-norms. All these functions are convex but nondifferentiable. Therefore, (2.15) is sum of a convex and differentiable component and a convex but nondifferentiable component. This problem cannot be solved by a gradient-type algorithms because they are not able to deal with the nondifferentiable component. Fortunately, this class of optimization problems can be solved via proximal gradient descent. Define the proximal map of $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)$ as

$$\text{prox}_{\nu, \mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)}(\mathbf{x}) = \arg\min_{\tilde{\mathbf{x}}} \left\{ \frac{1}{2} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2 + \nu \mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) \right\}, \tag{2.16}$$

where ν is the step-size parameter calculated as $\nu = \frac{1}{L_c}$, and L_c is the Lipschitz constant of $\nabla \mathcal{L}(\mathbf{\Phi}) = -(\mathbf{Y} - \mathbf{\Phi}\mathbf{Z})\mathbf{Z}'$, which corresponds to the largest eigenvalue of $\mathbf{Z}\mathbf{Z}'$. The proximal operator does not depend on the loss function $\mathcal{L}(\mathbf{\Phi})$, rather it depends on the penalty function $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)$ is included. A convenient property of the proximal operator is that it can be evaluated efficiently for many popular nondifferentiable (penalty) functions. The proximal operator is evaluated at the gradient step that would have been taken if $\mathcal{L}(\mathbf{\Phi})$ alone were to be minimized. For $m = 1, 2, \ldots$, its updates are given by

$$\hat{\mathbf{\Phi}}[m] = \text{prox}_{\nu, \mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)} \left(\hat{\mathbf{\Phi}}[m-1] - \nu \nabla \mathcal{L}(\hat{\mathbf{\Phi}}[m-1]) \right).$$

This algorithm is called ISTA (Iterative Shrinkage-Thresholding Algorithm) and has a convergence rate of $O(\frac{1}{k})$ (for more information regarding ISTA, refer to Chambolle et al., 1998, Daubechies et al., 2004). Of course, the solutions to (2.16) depend very much on $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma)$. This algorithm is performing a proximal descent and can be accelerated by a Nesterov accelerated scheme.

The algorithm including this Nesterov step is called FISTA (Fast Iterative Shrinkage–Thresholding Algorithm) introduced by Beck and Teboulle (2009). The FISTA algorithm for the problem in (2.15)

is computed by

$$\hat{\mathbf{\Phi}}^* = \hat{\mathbf{\Phi}}[m-1] + \frac{m-2}{m+1}(\hat{\mathbf{\Phi}}[m-1] - \hat{\mathbf{\Phi}}[m-2])$$
$$\hat{\mathbf{\Phi}}[m] = \text{prox}_{\nu,\mathcal{P}(\mathbf{\Phi};\lambda,\gamma)}\left(\hat{\mathbf{\Phi}}^* - \nu\nabla\mathcal{L}(\hat{\mathbf{\Phi}}^*)\right). \tag{2.17}$$

This algorithm converges at rate $O(\frac{1}{k^2})$, which is a significant improvement compared to the unaccelerated proximal gradient method's $O(\frac{1}{k})$ rate. It is thusly this algorithm that is used to solve all the regularization problems encountered in this paper. Algorithm 1 depicts how FISTA works in pseudocode.

Algorithm 1: FISTA

Require: $\mathbf{Y}$, $\mathbf{Z}$, $\hat{\mathbf{\Phi}}[0]$, λ, γ, ε;
$\nu \leftarrow \max_\lambda(\mathbf{ZZ}')$;
for $m = 3, 4, \ldots$ **do**
  $\hat{\mathbf{\Phi}}^* \leftarrow \hat{\mathbf{\Phi}}[m-1] + \frac{m-2}{m+1}(\hat{\mathbf{\Phi}}[m-1] - \hat{\mathbf{\Phi}}[m-2])$;
  $\hat{\mathbf{\Phi}}[m] \leftarrow \text{prox}_{\nu,\mathcal{P}(\mathbf{\Phi};\lambda,\gamma)}\left(\hat{\mathbf{\Phi}}^* - \nu\nabla\mathcal{L}(\hat{\mathbf{\Phi}}^*)\right)$;
  if $\|\hat{\mathbf{\Phi}}[m] - \hat{\mathbf{\Phi}}^*\|_\infty < \varepsilon$ **then**
    break
  end
end
return $\hat{\mathbf{\Phi}}[m]$

Before demonstrating the solutions to the proximal operators, there is one important observation that needs to be made. All procedures, except for the lagwise HLag, can be broken down across rows of $\mathbf{\Phi}$. Therefore, in these cases one can concentrate on solving the one-row subproblem:

$$\arg\min_{\mathbf{\Phi}_i}\left\{\|\mathbf{Y}_i - \mathbf{\Phi}_i\mathbf{Z}\|_2^2 + \mathcal{P}(\mathbf{\Phi};\lambda,\gamma)\right\} \quad \text{for} \quad i = 1, \ldots, k, \tag{2.18}$$

and evaluate the proximal operater row-wise

$$\hat{\mathbf{\Phi}}_i[m] = \text{prox}_{\nu,\mathcal{P}(\mathbf{\Phi};\lambda,\gamma)}\left(\hat{\mathbf{\Phi}}_i^* - \nu\nabla\mathcal{L}_i(\hat{\mathbf{\Phi}}_i^*)\right) \quad \text{for} \quad i = 1, \ldots, k. \tag{2.19}$$

In this one-row subproblem case Algorithm 1 alters a bit. This altered form is presented in Algorithm 2. Solving the problems row-wise results in a significantly faster computation time.

Evaluating the proximal operator for all procedures is remarkably efficient. As a matter of fact, they all have essentially a closed form solution. The solution to (2.16) for the L_1-norm lasso is

Algorithm 2: Row-Wise FISTA

Require: $\mathbf{Y}$, $\mathbf{Z}$, $\hat{\mathbf{\Phi}}[0]$, λ, γ, ε;

$\nu \leftarrow \max_{\lambda}(\mathbf{Z}\mathbf{Z}')$;

for $k = 1, \ldots, k$ **do**

  for $m = 3, 4, \ldots$ **do**

    $\hat{\mathbf{\Phi}}_i^* \leftarrow \hat{\mathbf{\Phi}}_i[m-1] + \frac{m-2}{m+1}(\hat{\mathbf{\Phi}}_i[m-1] - \hat{\mathbf{\Phi}}_i[m-2])$;

    $\hat{\mathbf{\Phi}}_i[m] \leftarrow \text{prox}_{\nu, \mathcal{P}(\mathbf{\Phi};\lambda,\gamma)}\left(\hat{\mathbf{\Phi}}_i^* - \nu \nabla \mathcal{L}_i(\hat{\mathbf{\Phi}}_i^*)\right)$;

    if $\|\hat{\mathbf{\Phi}}_i[m] - \hat{\mathbf{\Phi}}_i^*\|_\infty < \varepsilon$ **then**

      break

    end

  end

end

return $\hat{\mathbf{\Phi}}[m]$

just elementwise soft-thresholding. The HLag penalty functions have an equivalently simple form. Jenatton et al. (2011) show that if the regularization term is a sum of nested L_2-norms, the dual of its proximal operater can be solved exactly in a single pass of blockwise coordinate descent. By strong duality, this solution to the dual provides us with an analytical solution to the problem (2.16) for the HLag structures. For sake of brevity, the solutions are not displayed here, rather they are placed in Appendix A.

2.5 Tuning Parameter Choice

The performance of the regularization procedures discussed in Section 2 depend critically on the tuning parameters γ and λ. The tuning parameters are unknown in practice but can be determined via statistical model validation routines. The rolling cross-validation, as used by Song and Bickel (2011), is applied in this paper to determine the optimal γ and λ combination. Applying rolling cross-validation in vector autoregressive context is well-advised, as this method directly incorporates the inherent temporal ordering of time series data. The motivation behind rolling cross-validation is that ostensibly traditional n-fold cross validation is not suitable for a time-dependent model because time series data cannot adhere to the principle that training and validation datasets should be independent. However, n-fold cross-validation may still be appropriate in certain circumstances. Bergmeir et al. (2017) show that n-fold cross-validation remains valid in a purely autoregressive

model, given that it is assumed that the errors are uncorrelated. Regardless, as the performance of n-fold cross-validation is scantly researched in a vector autoregressive context, nor is tuning parameter selection the central theme in this thesis, I persist in using rolling cross-validation to determine the optimal tuning parameters[1].

The tuning parameter λ is selected from the grid $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_{G_\lambda})$, and γ is selected from $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_{G_\gamma})$, where G_λ and G_γ denote the amount of grid-points for λ and γ, respectively. Unless stated otherwise, all applications, for all regularization algorithms, have the γ-grid set as $\boldsymbol{\gamma} = (1, 0.5, 0)$. This is a rather restrictive grid, however it is set that way as increasing the amount of grid-points will significantly increase computation time. Moreover, the results marginally improve if the amount of grid-points is increased. The λ-grid is not fixed, rather it is procedure- and data-dependent. The first element of the grid, λ_1, is set to the smallest value such that for any $\lambda \geq \lambda_1$ it follows that $\hat{\boldsymbol{\Phi}}$ is always equal to $\mathbf{0}$. This value is found by a line-search algorithm. The values in $\boldsymbol{\lambda}$ decrease in log-linear increments to $\frac{\lambda_1}{d}$, where d is the depth of the grid. The depth of the grid determines how small the values in $\boldsymbol{\lambda}$ will be. A deep grid, i.e, if d is set to a large number, will result in a significant increase in computation cost. For most applications I let $G_\lambda = 10$ and $d = \frac{1}{25}\lambda_1$. Generally speaking, I have found good model selection and forecasting performance by setting up the λ-grid in this way.

Like in Song and Bickel (2011), the data is split three periods: The first period from 1 to $T_1 - 1$ is used for model estimation, based on the second period from T_1 to $T_2 - 1$ different penalty parameters are assessed, and the third period from T_2 to the end of the sample is used for forecast evaluation. Unless stated otherwise, the time indices in this thesis are set as $T_1 = \lfloor \frac{T}{3} \rfloor$ and $T_2 = \lfloor \frac{2T}{3} \rfloor$. The validation procedure is started by fitting a model using all data up to time T_1 and forecasting $\hat{\mathbf{y}}_{T_1+h|T_1}^{\lambda_i,\gamma_j}$ for $i = 1, \dots, G_\lambda$ and $j = 1, \dots, G_\gamma$. This process is repeated until $T_2 - h$, whereby at each new iteration one observation is added and another h-step point forecast is estimated. Figure 2 depicts the rolling cross-validation procedure for a fixed λ_i and γ_j.

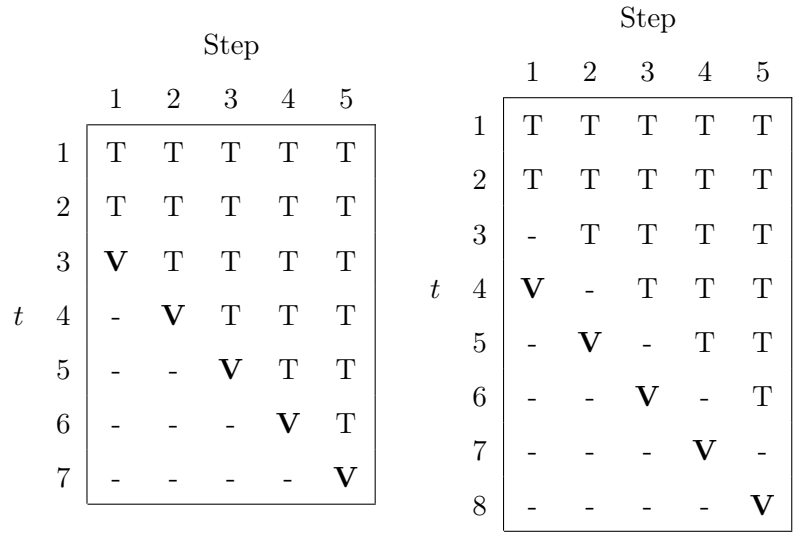
To quantify the regularization procedure's performance for each λ_i and γ_j, the h-step-ahead MSFE is used as a cross-validation score:

$$\hat{\mathcal{L}}_h^{\text{CV}}(\lambda_i, \gamma_j) = \frac{1}{T_2 - T_1 - h + 1} \sum_{t=T_1}^{T_2-h} \|\mathbf{y}_{t+h} - \hat{\mathbf{y}}_{t+h|t}^{\lambda_i,\gamma_j}\|_2^2. \tag{2.20}$$

[1]One may also opt for different methods other than rolling cross-validation and n-fold cross-validation. A recent paper by Na (2017) uses the generalized information criteria to determine the tuning parameters for autoregressive models and found good performance.

Figure 2: **Rolling Cross-validation**

Rolling h-step-ahead cross-validation with expanding training window for a fixed λ_i and γ_j. 'T' and '**V**' denote that the observation is included in the training and validation sample, respectively. A hyphen ('-') indicates that an observation is excluded from both training and validation sample. The figure on the left depicts rolling cross-validation for $h = 1$, and the figure on the right for $h = 2$.



A natural choice for selecting the optimal λ and γ from their respective grids, is selecting them such that they minimize the validation score in (2.20). However, the interest in this work lies not just in an estimated model that is able to forecast well, but also in a model that is parsimonious and interpretable. To realize this interest the 'one-standard error rule' is applied. As to limit the increase in bias, which will typically be a consequence of the one-standard error rule, it is only applied for γ whilst λ is selected such that it minimizes the validation-score and remains fixed throughout the process. Prior to formally explaining the rule, denote the standard deviation of cross-validation scores over gamma

$$\widehat{\mathrm{SD}}(\gamma) = \sqrt{\hat{\mathcal{L}}_h^{\mathrm{CV}}(\lambda_i, \gamma_1) + \cdots + \hat{\mathcal{L}}_h^{\mathrm{CV}}(\lambda_i, \gamma_{G_\gamma})},$$

and the corresponding standard error

$$\widehat{\mathrm{SE}}(\gamma) = \frac{\widehat{\mathrm{SD}}(\gamma)}{\sqrt{G_\gamma}}.$$

The procedure starts by selecting $\hat{\lambda}$ and $\hat{\gamma}$ such that they minimize the cross-validation score (2.20). The lag-parameter γ is then moved in the direction of increasing regularization until the following inequality is violated:

$$\hat{\mathcal{L}}_h^{\mathrm{CV}}(\hat{\lambda}, \gamma_j) \leq \hat{\mathcal{L}}_h^{\mathrm{CV}}(\hat{\lambda}, \hat{\gamma}) + \widehat{\mathrm{SE}}(\hat{\lambda}, \hat{\gamma}).$$

In words, the most regularized model whose error is within one standard error of the minimal cross-validation score is chosen rather than the model with a minimal validation-score. The choice of one standard error is entirely heuristic. One standard error typically is not large relative to the range of values.

Two dimensional rolling cross-validation can be computationally expensive, in particular when k is large. To reduce computation time, rolling cross-validation is applied such that it uses the result from the previous period as initialization for the current period. In practice, this substantially decreases computation time.

3 Monte Carlo Study

To evaluate the performance of all regularization procedures in a finite sample, I will focus on the problem of obtaining accurate point forecasts, the ability to uncover the correct sparsity pattern, and the consistency of the parameter estimates in various simulation set-ups. In the simulation study only one-step-ahead forecasts are constructed. The accuracy of the proposed procedures will be quantified by several evaluative metrics that are described in the following section.

3.1 Performance Measures

Several performance measures are considered. These measures aim to give an insight in the HLag structures' forecasting and parameter selection performance. A way to gauge the predictive power of each method is to compare the MSFE obtained by forecasts constructed with regularization procedures relative to the MSFE obtained by forecasts that are constructed by a benchmark method

$$\frac{1}{N}\sum_{n=1}^{N}\frac{\|\mathbf{y}_{t+1}-\hat{\mathbf{y}}^{[n]}\|_2^2}{\|\mathbf{y}_{t+1}-\bar{\mathbf{y}}^{[n]}\|_2^2},$$

whereby $\bar{\mathbf{y}}$ corresponds to the sample mean model – which in this case is the benchmark method – specified as

$$\bar{\mathbf{y}}_{T+h}=\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t, \tag{3.1}$$

and the superscript $[n]$ denotes the constructed forecasts for the n^{th} simulation.

To assess the accuracy of the estimated parameter values of each procedure, the mean squared

error of the parameter estimates is computed

$$\frac{1}{Npk^2}\sum_{n=1}^{N}\|\mathbf{\Phi}-\hat{\mathbf{\Phi}}^{[n]}\|_2^2.$$

As in this thesis there is a heavy focus on lag selection, a performance measure regarding it is also be designed. First, define the estimated lag matrix as $\hat{\mathbf{L}}_{ij}=\max\{\ell:\hat{\mathbf{\Phi}}_{ij}^{(\ell)}\neq 0\}$. A procedure's lag order selection accuracy is measured based on the L_1-norm of the difference between $\mathbf{L}$ and $\hat{\mathbf{L}}$, relative the difference between $\mathbf{L}$ and $\mathbf{0}$ (where $\mathbf{0}$ is the 'selected' lag order by the sample mean procedure)

$$\frac{1}{N}\sum_{n=1}^{N}\frac{\|\mathbf{L}-\hat{\mathbf{L}}^{[n]}\|_1}{\|\mathbf{L}\|_1}.$$

This performance measure coincides to comparing the the lag matrices produced by the regularization procedures to always selecting zero lags for all variables.

Sparsity recognition is assessed by looking at the true positive rate (TPR) and the true negative rate (TNR),

$$\frac{1}{N}\sum_{n=1}^{N}\frac{\#\{i,j:\mathbf{\Phi}_{ij}=0\wedge\hat{\mathbf{\Phi}}_{ij}^{[n]}=0\}}{\#\{i,j:\mathbf{\Phi}_{ij}=0\}}\quad\text{and}$$
$$\frac{1}{N}\sum_{n=1}^{N}\frac{\#\{i,j:\mathbf{\Phi}_{ij}\neq 0\wedge\hat{\mathbf{\Phi}}_{ij}^{[n]}\neq 0\}}{\#\{i,j:\mathbf{\Phi}_{ij}\neq 0\}},$$

respectively. The TPR gives the rate of correctly estimating an inactive parameter as inactive, whereas the TNR gives the rate of correctly excluding an inactive parameter. Both should be as large as possible for reliable parameter selection.

3.2 Simulation Scenarios

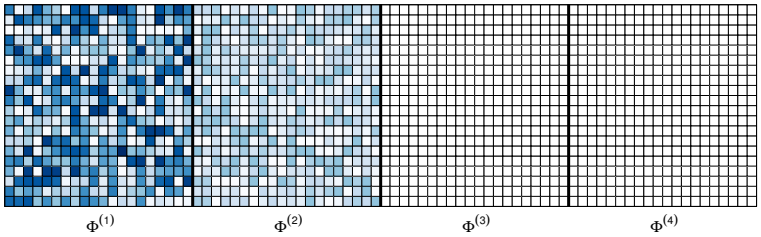
All the proposed procedures operate on a $\text{VAR}_{20}(4)$. Here $p=4$ entails that 4 is the maximal lag order. The maximal lag order is the largest order considered in the model fitting procedures. The choice of $p=4$ is selected because it represents one year of dependence for quarterly data, which is a common frequency of macroeconomic data. Formally, each method is performed on the following specification

$$\mathbf{y}_t=\sum_{\ell=1}^{4}\mathbf{\Phi}^{(\ell)}\mathbf{y}_{t-\ell}+\mathbf{u}_t,\quad\text{with}\quad\mathbf{u}_t\sim\mathcal{N}(\mathbf{0},0.01\mathbf{I}_{20})\quad\text{for}\quad t=1,\dots,T,\tag{3.2}$$

and 100 simulations are performed ($N=100$) for every method. I do not include a constant term in (3.2) because, as shown in (2.14), by temporally demeaning the variables one may omit the intercept in

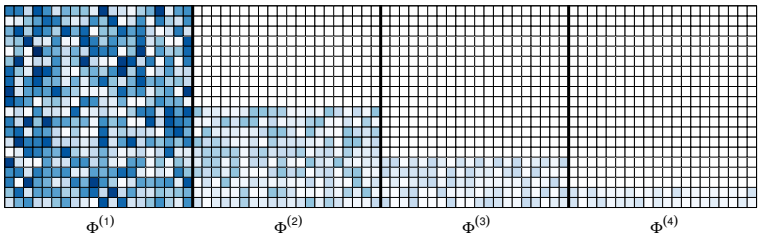
any scenario. To get an insight in the asymptotic performance of the proposed methods, all methods will be performed with varying amount of temporal observation, namely for $T \in \{50, 100, 200\}$. Finally, in order to test the applicability of every specification, different coefficient matrices (i.e. scenarios) are generated. An explanation and motivation of each of these scenarios follows.

Figure 3: **DGP1**



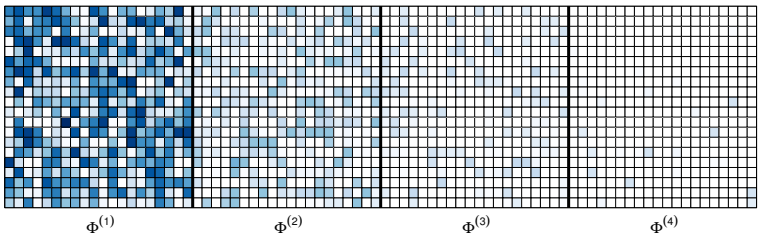
In the first scenario $\mathbf{\Phi}^{(1)}$ and $\mathbf{\Phi}^{(2)}$ are dense with coefficients. All other coefficents are set to zero. Such a design simulates a scenario in which all components have very similar temporal behavior. Under such a design, one should expect superior performance from the lagwise HLag structure. The coefficients are distributed as $\mathbf{\Phi}_{ij}^{(\ell)} \sim U(-\frac{0.4}{\ell}, \frac{0.4}{\ell})$ for $\ell = 1, 2$ and for $i, j = 1, \ldots, k$. The relative magnitude of the coefficients are depicted in Figure 3.

Figure 4: **DGP2**



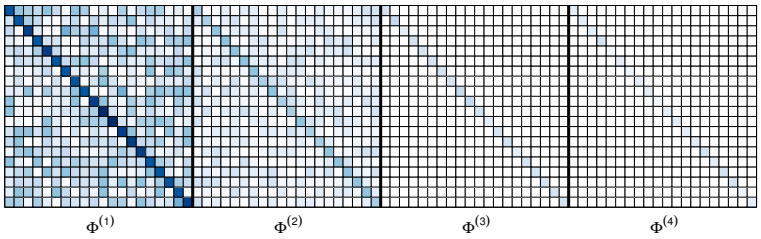
The second design is set-up such that it simulates a perfectly asymmetric (in equations) VAR. With each increment in ℓ, $\mathbf{\Phi}^{\ell}$ becomes (approximately) 50% more sparse. Under such a design, one should expect the componentwise HLag structure to perform the best. The active coefficients are distributed as $\mathbf{\Phi}_{ij}^{(\ell)} \sim U(\frac{-0.4}{\ell}, \frac{0.4}{\ell})$ for $\ell = 1, \ldots, p$. The relative magnitude of the coefficients are depicted in Figure 4.

Figure 5: **DGP3**



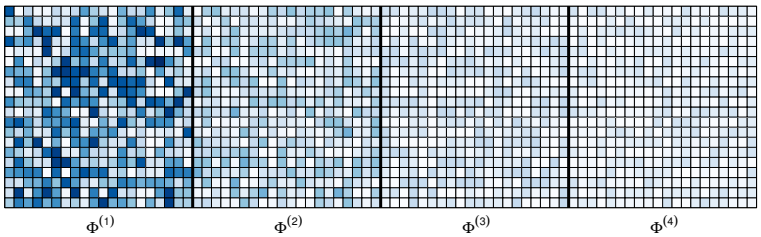
The third scenario simulates a coefficient matrix that has no spatial structure, but does have temporal structure, in the sense that if $\mathbf{\Phi}_{ij}^{\ell'} = 0$ then $\mathbf{\Phi}_{ij}^{\ell} = 0$ for $\ell' > \ell$. Each dependent variable has a 5% chance of having four lags, 25% chance of either having three lags, 30% chance of having two lags, 30% chance of having one lag, and a 10% chance of having no lag of for every variable. Under such a design, one should expect superior performance from the elementwise HLag structure. The active coefficients are drawn as $\mathbf{\Phi}_{ij}^{(\ell)} \sim U(-\frac{0.4}{\ell}, \frac{0.4}{\ell})$ for $\ell = 1, \ldots, p$. The relative magnitude of the coefficients are depicted in Figure 5.

Figure 6: **DGP4**



In this scenario a diagonal dominant structure is generated. For $\ell > 2$, all except the diagonal elements are set to zero. Under such a design, one should expect superior performance from the own-other structure. The active non-diagonal parameters are drawn as $\mathbf{\Phi}_{ij}^{\ell} \sim U(\frac{-0.35}{2\ell}, \frac{0.35}{2\ell})$ for $\ell = 1, 2$ for every $i \neq j$. The diagonal elements are drawn as $\mathbf{\Phi}_{ii}^{\ell} \sim U(\frac{-0.35}{\ell}, \frac{0.35}{\ell})$ for $\ell = 1, \ldots, p$ and $i = 1, \ldots, k$. To ensure the large magnitude of the diagonal elements relative to the non-diagonal elements, the draws of the diagonal elements are only 'accepted' if they are larger in magnitude than a certain threshold, whereby the threshold itself decreases as ℓ increases. If a draw does not meet the threshold, the diagonal parameter is redrawn until that threshold is met. The relative magnitude of the coefficients are depicted in Figure 6.

Figure 7: **DGP5**



This final scenario considers a dense matrix $\mathbf{\Phi}$, in which there is no sparsity whatsoever in the coefficient matrix. This scenario is primarily to assess if the penalty methods are able to deal which such a dense matrix and how accurate the parameter estimates are without wrongfully excluding any parameter. All coefficients are drawn as $\mathbf{\Phi}_{ij}^{(\ell)} \sim U(-\frac{0.35}{\ell}, \frac{0.35}{\ell})$ for $\ell = 1, \dots, p$ and for $i, j = 1, \dots, k$. The relative magnitude of the coefficients are depicted in Figure 7.

Each simulation scenario was generated such that they produced a stationary coefficient matrix. The conditions of stationary VARs and how they were generated is explained in Appendix B.1. On a last note, the performance of the regularization procedures are not only compared against one another, rather they are also compared against OLS. The OLS is of course only possible when $kp \leq T$. If $k = 20$, $p = 4$ and $T = 50$ this inequality is clearly violated. In that case the $\text{VAR}_{20}(1)$ is estimated by least squares:

$$\hat{\mathbf{\Phi}}, \hat{\mathbf{c}} = \underset{\mathbf{\Phi}, \mathbf{c}}{\arg\min} \left\{ \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{c} - \mathbf{\Phi}\mathbf{y}_{t-1}\|_2^2 \right\}.$$

3.3 Simulation Results

In Table 2 the numerical results for $T \in \{50, 100, 200\}$ are reported. Figure 8 depicts densities of the estimates over the 100 Monte Carlo replications of the first parameter in the first equation. To examine the consistency of a coefficient's estimate the true value of the first parameter is manually set 0.3 for each generated coefficient matrix. For convenience that value is depicted with a vertical black line in each density plot. Table 2 report some quite surprising and counter-intuitive results. One of them is that lag selection accuracy does not necessarily increase as sample size increases. This hold in particular for the first three DGPs. Moreover, the TPR mostly worsens (i.e., decreases) as sample size increases. This entails that, surprisingly, correct sparsity in $\mathbf{\Phi}$ is less likely to be identified as T increases, across all DGPs and across all procedures. In contrary, TNR increases with

sample size, across all DGPs and across all procedures. Thus, from these two dynamics it seems that can derive that with an increase in sample size, more parameters are included in the estimated model regardless whether they should actually be included. Most importantly though, the MSFE for all methods, across all scenarios, always decreases as T increases. Thus, in terms of minimizing MSFE, not correctly identifying sparsity seems to be less detrimental than incorrectly excluding parameters. Finally, the results of the MSE of the parameter estimates behaves as expected: As T increases the parameter MSE, generally speaking, decreases. There is one exception to this and it is in the case of DGP4, the diagonally dominant scenario, where an increase in T causes the MSE to increase slightly or remain roughly the same. Table 6 in Appendix B.4 report the standard errors (expressed in percentages) of the results. The standard errors are well-behaved in the sense that as T increases the standard errors decrease for every performance measure. A more intriguing observation, however, is that the lagwise HLag has a standard error of 0 for the TPR across all DGPs. This means that for all 100 simulations, for every DGP, it fails to set a group of parameters to $\mathbf{0}$ even once. This includes for DGP1, which is a particularly favorable set-up for the lagwise HLag.

Several things can be derived from the density plots displayed in Figure 8. Firstly, an increase in sample size centers the densities more towards the true parameter. Also, as T increases the density of the various procedures converge to each other. In small sample size, in particular if $T = 50$, the compenentwise and lagwise HLag have narrower density plots relative. The density plots, across all procedures, are (usually) biased to the left of the true parameter value, which is expected from a lasso based method. Methods based on the lasso are always biased as it shrinks the parameters to zero. This phenomenon is extensively discussed in Zou (2006). Interestingly though, DGP5, the dense scenario, is biased to the right of the true parameter value, i.e, it estimates the parameter higher than its actual magnitude. Of course, as there is no sparsity in this scenario, the shrinkage terms λ and γ are selected to be very small such that the regularization procedures barely shrink the parameters. Three figures depicting the relative magnitude of the estimated parameters for DGP1 are placed in Appendix B.3. For each plot, the $N = 100$ estimated coefficient matrices are averaged and then plotted. As $T = 200$ it starts resembling Figure 3, however, here too the estimated parameters' magnitude values tend to be smaller than the true parameter values.

Table 7 in Appendix B.5 reports the results with γ fixed to zero. The TNR that results from procedures without a lag-function parameter is always better than with a lag-function. In complete opposite, the TPR that results from procedures with is always better than without a lag-function.

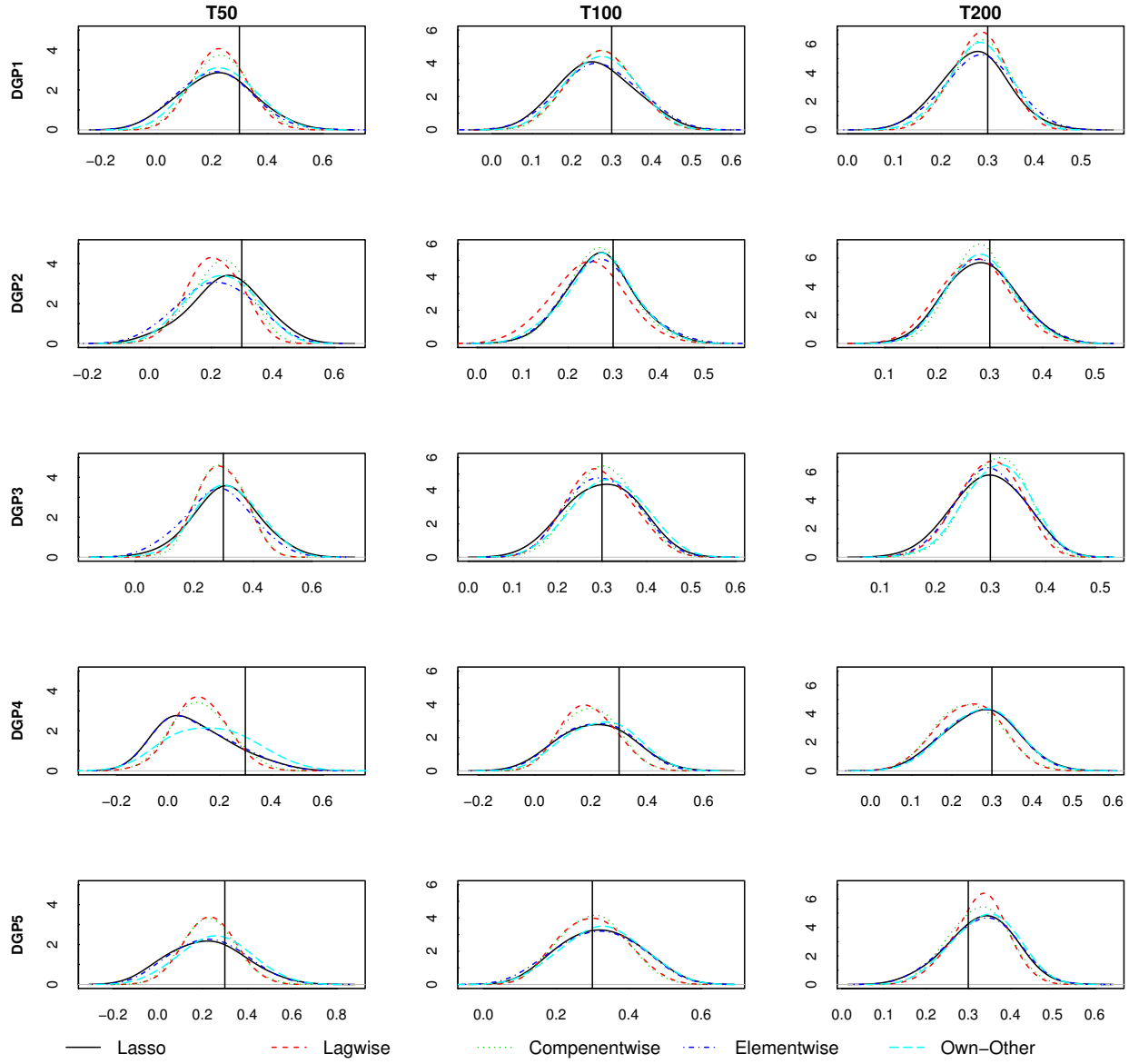
Table 2: **Simulation Results**

This table depicts numerical results of several performance measure for each simulation setting, and for various time series lengths. All procedures in this table included a weighting parameter.

		Scenario														
		DGP1			DGP2			DGP3			DGP4			DGP5		
	T	50	100	200	50	100	200	50	100	200	50	100	200	50	100	200
		MSFE Relative to Sample Mean														
Method	Lasso	0.222	0.110	0.079	0.188	0.101	0.068	0.263	0.133	0.097	0.909	0.795	0.712	0.369	0.209	0.147
	Lagwise	0.188	0.100	0.073	0.171	0.097	0.066	0.228	0.130	0.092	0.857	0.771	0.686	0.298	0.188	0.139
	Componentwise	0.192	0.101	0.075	0.173	0.094	0.066	0.232	0.125	0.094	0.861	0.770	0.697	0.301	0.189	0.139
	Elementwise	0.216	0.105	0.077	0.191	0.099	0.067	0.259	0.129	0.095	0.896	0.787	0.705	0.345	0.202	0.143
	Own-Other	0.193	0.102	0.076	0.170	0.095	0.066	0.232	0.126	0.094	0.846	0.759	0.688	0.303	0.190	0.139
	OLS	2.003	1.919	0.124	1.167	1.073	0.111	1.796	1.704	0.154	1.692	1.528	1.252	2.285	2.354	0.210
		MSE of the Parameter Estimates														
	Lasso	0.008	0.005	0.003	0.007	0.004	0.002	0.007	0.004	0.003	0.006	0.008	0.009	0.010	0.006	0.004
	Lagwise	0.006	0.003	0.002	0.005	0.003	0.002	0.006	0.003	0.002	0.007	0.008	0.008	0.007	0.005	0.003
	Componentwise	0.006	0.004	0.002	0.006	0.003	0.002	0.006	0.003	0.002	0.008	0.008	0.009	0.007	0.005	0.003
	Elementwise	0.008	0.004	0.003	0.007	0.004	0.002	0.007	0.004	0.002	0.006	0.008	0.008	0.009	0.006	0.004
	Own-Other	0.006	0.004	0.002	0.005	0.003	0.002	0.006	0.004	0.002	0.007	0.008	0.008	0.007	0.005	0.003
		Lag Selection Accuracy Relative to Sample Mean														
	Lasso	0.634	0.620	0.711	0.754	0.790	0.874	0.646	0.636	0.712	0.937	0.729	0.565	0.774	0.510	0.291
	Lagwise	1.000	1.000	1.000	1.162	1.162	1.162	1.060	1.060	1.060	0.905	0.905	0.905	0.000	0.000	0.000
	Compenentwise	0.925	0.966	0.974	1.088	1.091	1.111	1.027	1.047	1.046	0.900	0.908	0.905	0.000	0.000	0.000
	Elementwise	0.713	0.607	0.721	0.929	0.807	0.856	0.709	0.626	0.691	0.922	0.764	0.648	0.691	0.354	0.201
	Own-Other	0.971	0.999	1.000	1.190	1.179	1.175	1.052	1.052	1.048	0.989	0.970	0.955	0.049	0.039	0.028
		True Positive Rate														
	Lasso	0.913	0.756	0.607	0.879	0.771	0.637	0.866	0.737	0.593	0.984	0.973	0.912	-	-	-
	Lagwise	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	-
	Componentwise	0.076	0.034	0.026	0.066	0.062	0.044	0.042	0.016	0.016	0.055	0.006	0.014	-	-	-
	Elementwise	0.757	0.686	0.493	0.611	0.671	0.542	0.687	0.663	0.505	0.949	0.791	0.707	-	-	-
	Own-Other	0.065	0.023	0.016	0.041	0.034	0.025	0.022	0.018	0.015	0.005	0.001	0.001	-	-	-
		True Negative Rate														
	Lasso	0.552	0.779	0.893	0.560	0.709	0.818	0.507	0.696	0.823	0.180	0.393	0.585	0.293	0.516	0.714
	Lagwise	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Componentwise	1.000	1.000	1.000	0.998	0.999	1.000	0.991	0.998	0.999	0.991	1.000	0.999	1.000	1.000	1.000
	Elementwise	0.689	0.881	0.963	0.733	0.808	0.896	0.659	0.797	0.908	0.264	0.577	0.759	0.506	0.784	0.885
	Own-Other	0.991	0.997	0.999	0.992	0.996	0.997	0.986	0.993	0.996	0.971	0.986	0.991	0.985	0.993	0.996

Figure 8: **Density Plots**

Density plots of the first parameter in the first equation for $T \in \{50, 100, 200\}$, and for various simulation set-ups that were explained in Section 3.2. The true value of the first parameter of the first equation is illustrated by a vertical black line.



All methods exhibit an improvement in terms of MSFE and MSE (of the parameter estimates) if a lag-function is incorporated, however only for the basic lasso is this improvement really substantial. Including a lag-function results, in most cases, in sparser solution, and the forecasting performance (slightly) improves. Finally, lag selection accuracy is, by and large, better if a lag-function is present. In Figure 11 density plots are again depicted, except now γ is fixed to zero. There is no substantial difference between the asymptotic behaviors of the estimates with a weighting specification compared to without. This lack of difference become increasingly visible as T increases. However, when $T = 50$

one does see more 'erratic' behavior of the densities that were estimated without a lag-function.

In terms of MSFE, the own-other, lagwise, and componentwise HLags perform the best. However, this difference is only substantial in case $T = 50$. As T increases the results of each method seems to converge to one another. OLS always under performs compared to the regularization procedures, even for $T = 200$. Of course, as T increase one should expect a superior performance of OLS. After all, it is proven to be asymptotically efficient and consistent given certain assumptions, all of which hold for the simulation set-up in (3.2). In terms of lag selection accuracy, the elementwise HLag and the lasso perform the best. These two methods are the least constrained in terms of variable selection, which is why they are superior in uncovering the true sparsity for different DGPs. Finally, while each of the five different simulations scenarios were set-up such that one method theoretically should be preferred above all others, in practice it did not really matter in terms of performance of each procedure. Rather, as stated, the best performing procedures in terms of MSFE are the own-other, lagwise, and componentwise HLags, regardless of the simulation scenario.

4 Empirical Study

The procedures are evaluated on the dataset compiled by Stock and Watson (2005) and augmented by Koop (2013). This dataset is a popular choice for researchers in the VAR literature. The full dataset is publicly available at The Journal of Applied Econometrics Data Archive[2]. The dataset contains 168 quarterly macroeconomic variables in the time span of Quarter 4, 1959 to Quarter 4, 2007, amounting to 193 temporal observations. The macroeconomic indicators represent various informative aspects about the economy of the United States. Among them being income, industrial production, capacity, stock prices, interest rates for different maturities, exchange rates, and so forth. All variables, expect for the financial ones, are seasonally adjusted. Koop (2013) partitions the series into four nested groups. The motivation and exact composition of each partition is given in Koop (2013). However, for convenience of the reader, the mnemonics, descriptions, and to which partition each variable belongs, are all displayed in Table 8 in Appendix C.1.

In this paper three of the four partitions are considered, namely a partition that incorporates 20 macroeconomic variables (medium-small, $k = 20$), a partition that incorporates the medium-small group and twenty other macroeconomic variables (medium-large, $k = 40$), and finally partition

[2]link: http://qed.econ.queensu.ca/jae/2013-v28.2/koop/

that incorporates the medium-large group and 128 other macroeconomic variables (large, $k = 168$). Before estimating the model-parameters, monthly series are transformed to quarterly series by taking the sample average over every three months. Each series is then transformed to (approximately) stationarity following the transformation codes reported in Stock and Watson (2005). Finally, all time series are then standardized by subtracting their respective sample means and dividing by their respective sample standard deviations. In the following subsection, the forecast performance of the weighted HLag structures are compared for each data-partition. Following that, the structural analysis of the effect of a monetary policy shock is examined.

4.1 Forecasting

A comparative analysis of the forecasting performance is set-up for the medium-small, medium-large, and large partitions. The period from Quarter 3, 1975 to Quarter 3, 1991 is used for penalty parameter selection, while Quarter 4, 1992 to Quarter 4, 2007 is used for expanding-window forecast comparisons. The penalty parameters are selected using rolling cross-validation as elaborated in Section 2.5.

The regularization procedures will not only be compared to one another, but also to several popular methods often applied in literature. A standard method is to select a lag order using Akaike's information criterion (AIC) or Bayesian information criterion (BIC), introduced by Akaike (1974) and Schwarz (1978), respectively. The AIC and BIC of a $\text{VAR}_k(p)$ are defined as

$$\begin{aligned}\text{AIC}(\ell) &= \log\det(\hat{\mathbf{\Sigma}}_u^\ell) + \frac{2k^2\ell}{T} \\ \text{BIC}(\ell) &= \log\det(\hat{\mathbf{\Sigma}}_u^\ell) + \frac{\log(T)k^2\ell}{T},\end{aligned}$$

where $\hat{\mathbf{\Sigma}}_u^\ell$ is the estimated residual sample covariance matrix resulting from using least squares to fit the $\text{VAR}_k(p)$. The lag order ℓ that minimizes $\text{AIC}(\ell)$ or $\text{BIC}(\ell)$ is selected. Since $\log T > 2$ for any $T > 7$, the BIC statistic virtually always places a heavier penalty on models with many variables, resulting in the selection of smaller models than AIC. This method of lag order selection is only possible when $k\ell \leq T$ since otherwise least squares is not well-defined. For the large partition, AIC and BIC are overparameterized and therefore not included in the analysis. For more information on the use of model selection criteria in VARs consult Lütkepohl (2007). In addition to information criteria, two naive benchmarks are included that can be used for any high-dimensional system. One of them being the unconditional sample mean model as specified in (3.1). The second of them is the

random walk model, which makes h-step-ahead forecast based upon the most recent realization, i.e, $\hat{\mathbf{y}}_{t+h} = \mathbf{y}_t$.

The maximal lag order for all procedures is set to $p = 4$. The out-of-sample h-step-ahead mean MSFE with cross-validated selected forecast over the forecast evaluation period from T_1 to T_2 equals to $\hat{\mathcal{L}}_h^{\text{CV}}(\hat{\lambda}, \hat{\gamma})$, where $\hat{\mathcal{L}}_h^{\text{CV}}$ is defined in equation (2.20). The MSFEs for each group is displayed will not only be computed for forecast horizon $h = 1$, but for the horizons $h \in \{1, 2, 4\}$. As inducing sparsity is among the central themes of the paper, the amount of sparsity resulting from each method is also compared. To measure the amount of sparsity generated by each method, the following metric is used

$$\frac{\#\{i, j : \hat{\mathbf{\Phi}}_{ij} = 0\}}{pk^2}.$$

4.1.1 Forecast Results

The forecasting results with a lag-function are reported in Table 3. For comparison's sake, the results without a lag-function (i.e., where γ is fixed to 0) are reported in Table 9. The greatest improvements in forecast performance occurred with the medium-large VAR. This was also the outcome in the paper by Nicholson et al. (2018). It is therefore this partition that captures the most useful information. The regularization procedures outperform the benchmark methods across all datasets and across all forecast horizons. AIC in particular performs poorly. AIC imposes weaker penalty for higher lags and has a tendency to overfit, whereas BIC has a tendency to underfit. AIC, in most cases, selects the maximum lag order of four. BIC, all in except one case, simply degenerates to the unconditional sample mean model. It is also these two models, the BIC and sample mean model, that are the closest competitors to the regularization procedures.

The lasso and the elementwise HLags result in the most parsimonious models. This follows from the fact that both procedures are very flexible in variable selection. On the other hand, the lagwise HLag produces the least sparse estimated models. As a matter of fact, for $h = 1$ and $h = 2$, with a maximum lag order of four, it does not produce any sparsity whatsoever. It is after all the most restrictive HLag structure possible, and the maximum lag order of four is quite low. The own-other and componentwise HLags generate a moderate amount of sparsity. The own-other HLag is however more suitable for VARs than the compenentwise HLag since it makes use of the fact that in most economic applications a variable's own lags are more informative than the lags of other variables.

The predictive power of the procedures that include a lag-function is roughly the same as the

Table 3: **Forecast Results**

(Absolute) MSFE and sparsity percentage for each partition and for the horizons $h \in \{1, 2, 4\}$.

Group		$h = 1$		$h = 2$		$h = 4$	
$k = 20$	**Method**	MSFE	Sp. (%)	MSFE	Sp. (%)	MSFE	Sp. (%)
	Lasso	0.736	79.75	0.759	89.12	0.806	92.75
	Lagwise	0.721	0	0.785	0	0.828	0
	Componentwise	0.759	45.00	0.770	15.00	0.816	68.75
	Elementwise	0.726	72.69	0.760	87.62	0.806	92.88
	Own-Other	0.694	21.62	0.756	47.00	0.807	77.75
	AIC	1.446	0	1.709	75.00	1.776	75.00
	BIC	0.893	75.00	0.843	100	0.847	100
	Sample Mean	0.843	-	0.847	-	0.847	-
	Random Walk	1.798	-	1.435	-	1.640	-
$k = 40$	**Method**	MSFE	Sp. (%)	MSFE	Sp. (%)	MSFE	Sp. (%)
	Lasso	0.563	90.02	0.610	95.84	0.676	95.88
	Lagwise	0.573	0	0.633	0	0.691	25.00
	Componentwise	0.569	5.620	0.622	23.75	0.682	74.38
	Elementwise	0.558	81.75	0.610	95.88	0.676	95.92
	Own-Other	0.537	20.97	0.606	17.25	0.675	74.56
	AIC	3.074	0	3.161	0	3.430	0
	BIC	0.703	100	0.704	100	0.708	100
	Sample Mean	0.703	-	0.704	-	0.708	-
	Random Walk	1.266	-	1.103	-	1.326	-
$k = 168$	**Method**	MSFE	Sp. (%)	MSFE	Sp. (%)	MSFE	Sp. (%)
	Lasso	0.559	96.23	0.643	97.90	0.697	97.90
	Lagwise	0.594	0	0.660	0	0.710	50.00
	Componentwise	0.598	0.300	0.657	8.630	0.709	75.00
	Elementwise	0.552	94.56	0.642	97.33	0.698	97.90
	Own-Other	0.543	15.45	0.632	74.90	0.686	74.90
	Sample Mean	0.729	-	0.730	-	0.733	-
	Random Walk	1.302	-	1.207	-	1.283	-

procedures without a lag-function. To statistically examine this two-sided Diebold-Mariano (Diebold and Mariano, 1995) tests are conducted. The distance between the forecast errors of the model with and without a lag-function are measured via squared loss. The p-values of these tests are reported in Table 10 in Appendix C.3. If a significance level of $\alpha = 0.05$ is upheld then the null hypothesis of equal predictive ability cannot be rejected in most cases. An exception to this is when $k = 168$ and $h = 4$. In this case, all HLag methods seems to give statistically better predictions without a lag-function included, even though numerically in this case the MSFEs are also very similar. The penalty functions that include a lag-function tend to produce sparser models. If the forecast horizon increases the generated sparsity becomes much larger (approximately by 60% for some procedures). When h increases the belief that recent information is more important than distant information becomes more relevant. It is in that case γ is more often selected to be 1 (maximum lag penalization) and rarely 0. This implies that not including a lag-function in the penalty functions results in variables being over-selected, and doing away with those variables does not result in a deterioration of the accuracy of the out-of-sample forecasts.

4.1.2 Evaluating Model Performance with Model Confidence Sets

In addition to simply evaluating a model's forecast performance based on their MSFEs, the model confidence set (MCS) proposed by Hansen et al. (2011) is also used. The reason why the Diebold-Mariano test – the equivalence-test that was used in Section 4.1.1 – is not conducted, is because it is not designed to deal with a lot of different competing models simultaneously. If one wants to rank the models without a particular interest in choosing a specific benchmark then the MCS framework is a more appropriate method. In this subsection a concise description is given about the MCS procedure. For a comprehensive explanation regarding the MCS approach consult Hansen et al. (2011) and Bernardi and Catania (2014).

The objective of the MCS procedure is to determine the set of models that consists of the best model(s), $\mathcal{M}^*$, from a set of models, $\mathcal{M}$, with a given probability. To determine the set of superior models several significance tests are sequentially conducted. Models that are found to be significantly inferior to to other models are deleted from $\mathcal{M}$, resulting in set of superior models $\mathcal{M}^*$, within which the null hypothesis of equal predictive ability cannot be rejected. The MCS procedure starts by computing the sum of squared forecast error (SSFE) for each model over the period $T_2 + 1$ to T.

The SSFE for model i at time t is defined as $\text{SSFE}_{i,t} = \|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(i)}\|_2^2$. Formally, let

$$d_{ij,t} = \text{SSFE}_{i,t} - \text{SSFE}_{j,t}, \quad \text{for} \quad i, j \in \mathcal{M},$$

denote the loss differential between models i and j, and let

$$\bar{d}_{ij} = \frac{1}{T - T_2} \sum_{T_2+1}^{T} d_{ij,t} \quad \text{for} \quad i, j \in \mathcal{M},$$

denote the relative sample loss between models i and j model. Finally, define the test static

$$v_{ij} = \frac{\bar{d}_{ij}}{\sqrt{\widehat{\text{var}}(\bar{d}_{ij})}} \quad \text{for} \quad i, j \in \mathcal{M},$$

where $\widehat{\text{var}}(\bar{d}_{ij})$ is obtained via block bootstrap. The asymptotic distributions of the test statistic is non-standard. The distribution under the null hypothesis is therefore also estimated using a bootstrap procedure.

The test statistic

$$V_{R,M} = \max_{i,j \in \mathcal{M}} |v_{ij}|$$

is used to sequentially test for equal predictive ability. The model with largest pairwise differential is deleted from $\mathcal{M}$ if equal predictive ability is rejected at a confidence level of $1 - \alpha$. Thereafter, the procedure is restarted on the subset of models. The procedure stops only when equal predictive ability for a set of models cannot be rejected. For the experiment I create 5000 bootstrap samples and set $\alpha = 0.15$. The results are displayed in Table 4.

None of the regularization procedures outperform each other if $k = 20$. It is only as k increases to 40 or 168 that the cardinality of $\mathcal{M}^*$ decreases. The weighted own-other HLag is an element of $\mathcal{M}^*$ in all except one case, namely for the medium-large dataset and forecast horizon $h = 4$. In case of the large dataset, $\mathcal{M}^*$ constitutes (for every forecast horizon) only of the own-other HLag. This gives statistical significance to the result of the robust and superior forecast performance of the own-other HLag.

Table 4: **Model Confidence Set Results**

Model sets $\mathcal{M}^*$ of equal predictive ability ($\alpha = 0.15$) for each dataset and across forecast horizons $h \in \{1, 2, 4\}$. Within each set of models, the null hypothesis of equal predictive ability can not be rejected, though they achieve superior forecasting performances relative to all excluded models.

Data	MCS for 1-step forecasts	MCS for 2-step forecasts	MCS for 4-step forecasts
Medium-Small ($k = 20$)	Lasso	Lasso	Lasso
	Lagwise	Lagwise	Lagwise
	Componentwise	Componentwise	Componentwise
	Elementwise	Elementwise	Elementwise
	Own-Other	Own-Other	Own-Other
	BIC	BIC	BIC
	Sample Mean		Sample Mean
Medium-Large ($k = 40$)	Own-Other	Lasso	Lasso
		Elementwise	
		Own-Other	
Large ($k = 168$)	Own-Other	Own-Other	Own-Other

4.2 Lag Order Selection and Structural Analysis

In the previous subsection it was shown that the own-other HLag generally performs the best across all datasets. Recall however, the goal of this thesis was to consider an estimator that is both strong in forecasting power and interpretable. In the following sections the latter is examined in order to determine which HLag structure satisfies both accurate predictability and interpretability.

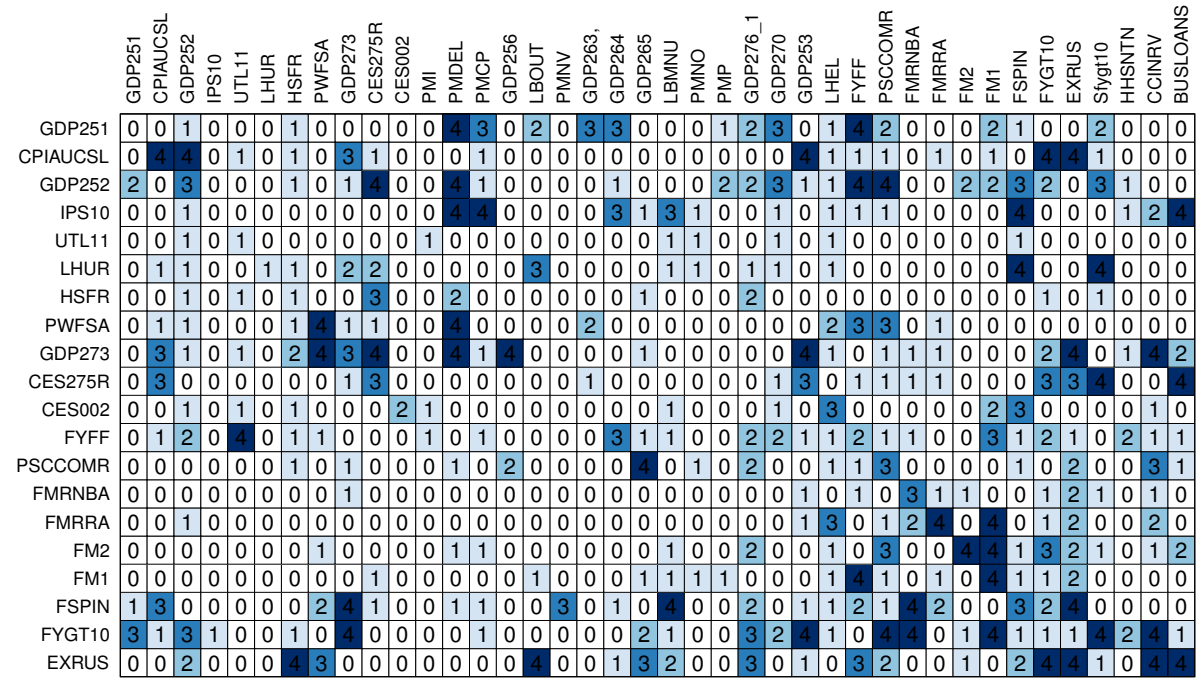
4.2.1 Lag Order Selection

As reported in Table 3, in case of $h = 1$, out of all the four HLag structures only the elementwise HLag structure generates a lot of sparsity. Moreover, it is the most natural HLag structure in terms of extracting information from the estimated lag matrix $\hat{\mathbf{L}}$. Hence, following Nicholson et al. (2018), it is this HLag that will be used to derive economic interpretations from the estimated lag matrix. The partition that results in the lowest MSFE is the medium-small partition, which means that this partition captures the most useful information of the variables. For sake of brevity, only the first

twenty rows of the estimated lag matrix are depicted in Figure 9.

Figure 9: **Estimated Lag Matrix**

Estimated lag matrix $\hat{\mathbf{L}}$, where, for sake of brevity, only the medium-small macroeconomic variables' estimated lags are depicted. The model was estimated on the medium-small dataset.



Prior to examining the lag matrix, it should be noted that having a coefficient to set to zero does not mean that a variable is completely useless in explaining its respective dependent variable. Rather, the HLag procedure simply found that in a specific system setting excluding those variables minimized the empirical loss function. This is substantiated by observing Figure 15 and Figure 16 in Appendix C.4 where two estimated lag matrices resulted from applying the elementwise HLag on the medium-small and large partition, respectively. Each figure gives a different picture of which variables are explanatory and which are not. Regardless, there is still useful economic interpretation that one can derive from the estimated lag matrices. As this is not an economics paper, the theory behind each and every variable will not be discussed. Rather, only three variables will be analyzed: real economic activity (`GDP251`), consumer price index (`CPIUACSL`), and the federal funds rate (`FYFF`).

The first thing to note is that `GDP251` does not depend on itself. Real economic activity is driven by many variables, many of which are included in the regression model, making the variable `GDP251` redundant. The relationship between `GDP251` and `FYFF` is a well-studied one. When a country goes

into recession the government attempts to reduce unemployment by boosting economic growth. They primarily do this by an expansionary monetary policy. Reducing the federal funds rate incentives businesses and consumers to borrow money, increasing the economic activity of a country. This relationship is detailed in the paper by Bernanke and Blinder (1992). One of the most reliable leading indicators for assessing the state of the U.S. economy is the purchasing managers' index (`PMI`). Despite that, it has an estimated lag of 0. This is caused by the fact that `PMI` is a composite index of various sub-indices, several of which are included in the equation of `GDP251`. supplier deliveries (`PMDEL`) and commodity index (`PMCP`) seems to be the primary drives of `PMI`, as they have an estimated lag of 4 and 3, respectively.

In the equation of `CPIUACSL` maximum lag order order is estimated for real personal consumption expenditure (`GDP252`). Naturally, consumer expenditure itself is directly affected by the the federal funds rate. This may also explain the estimated lag order of 1 for `FYFF` as most of the effect is already captured by `GDP252`. Maximum lag order was also selected for the US exchange rate (`EXRUS`). Exchange rate is one of the foremost macroeconomic variables that affect inflation (Edwards, 2006). Moreover, for many emerging economies inflation-targeting is typically done by intervening in the foreign exchange market.

The only maximum lag order selection for the `FYFF` equation is the unemployment rate (`UTL11`). Prag (1994) suggest there is a response of the interest rate to announcements of unexpected changes in the unemployment rate. Specifically, in response to an unexpectedly low unemployment rate announcement, interest rates rise are expected to rise. This indicates that the real rate is responding to these announcements in general. By this theory it is sensible that the past four quarters of unemployment is able to explain changes in the interest rate.

4.2.2 Impulse Response Analysis and Innovation Accounting

A very popular method that enables macroeconomists to conduct policy analysis is impulse response analysis. The HLag methods were initially designed to construct accurate forecasts in high-dimensional VARs. However, they are also suitable for structural analysis. The responses to a shock in the system are heavily dependent on which variables are included in the estimated model, and as the different procedures result in different models, one may expect discrepancy in the impulse responses across procedures.

Before explaining how the impulses responses are constructed it is important to recall the VAR

specification in (2.1) and to note that impulse response analysis relies on the assumption that a shock occurs only in one variable at a time. Such an assumption may be reasonable if the structural shocks in different variables are independent. However, in many applications, in particular economic ones, the shocks in different variables are correlated with one another. This is the reason why impulse response analysis is often performed in terms of the moving-average (MA) representation

$$\mathbf{y}_t = \sum_{i=1}^{p} \mathbf{\Psi}^{(i)} \mathbf{w}_{t-i}, \tag{4.1}$$

where the components of $\mathbf{w}_t$ are uncorrelated and have unit variance, $\mathbf{\Sigma}_w = \mathbf{I}_k$. The constant term is dropped in the present analysis as all the datasets are standardized. The MA representation in (4.1) is obtained by the Choleski decomposition of $\mathbf{\Sigma}_u$ as $\mathbf{\Sigma}_u = \mathbf{P}\mathbf{P}'$, where $\mathbf{P}$ is a lower triangular matrix, and defining $\mathbf{\Psi}^{(i)} = \mathbf{\Phi}^{(i)}\mathbf{P}$ and $\mathbf{w}_t = \mathbf{P}^{-1}\mathbf{u}_t$. A change in one component of $\mathbf{w}_t$ has no effect on the other components because the components in this specification are uncorrelated. The assumption that impulse response analysis rests on, namely uncorrelated shocks, holds true for (4.1).

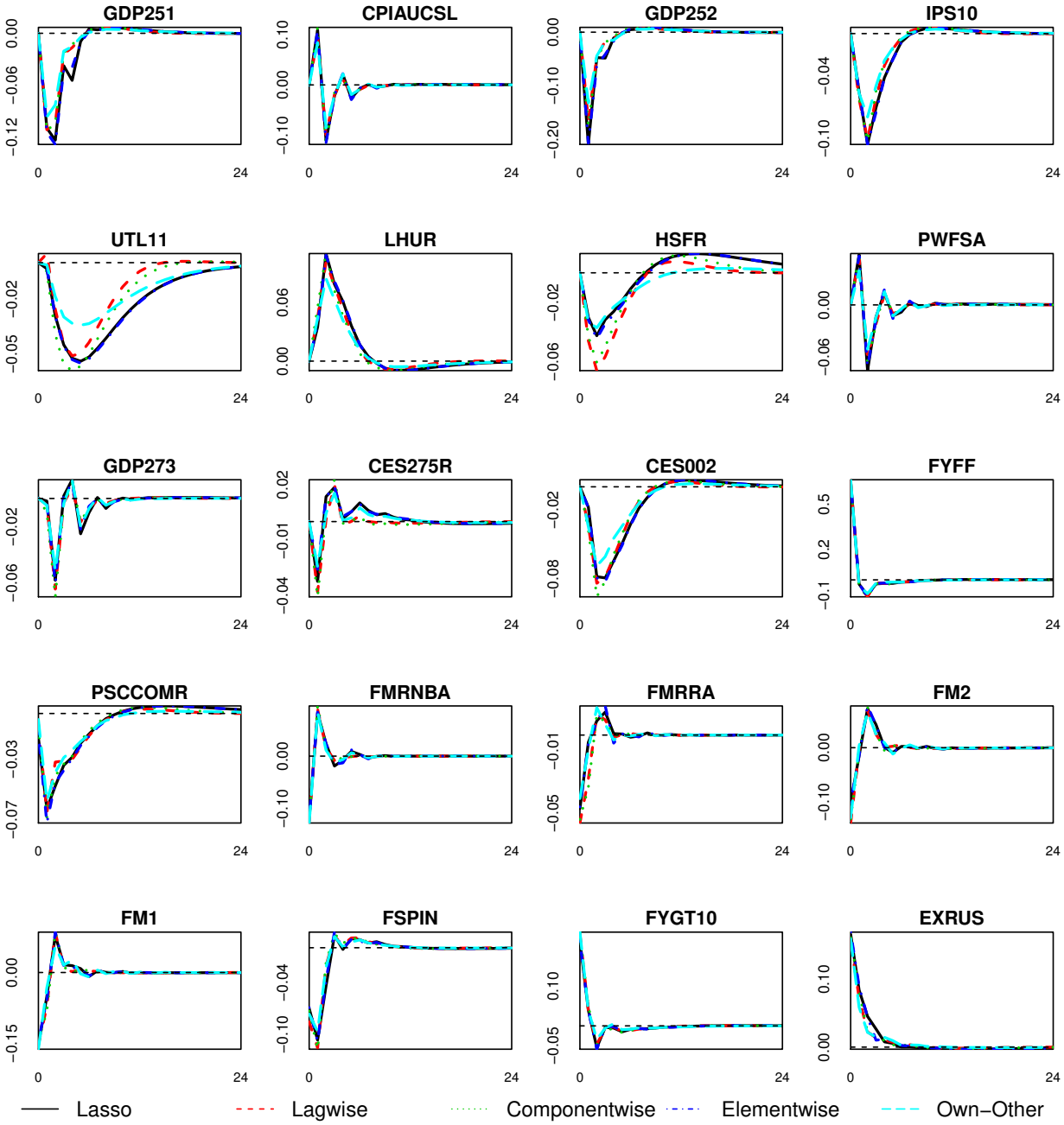
The goal of the current analysis is to trace out the response to a monetary policy shock. With that end in view, there remains one important point to note, namely the fact that because $\mathbf{P}$ is lower triangular, it implies that the ordering of the variables is of importance. To this purpose, the macroeconomic variables will be placed into two categories: slow- and fast-moving variables. Such an ordering is often used in VAR literature to trace out the effect of monetary policy innovations on the economy (see, among others, Bernanke and Eliasz, 2005, Christiano et al., 1999, Banbura et al., 2010). To concisely describe this identifying assumption, consider $\mathbf{k}_t = (\mathbf{s}_t, r_t, \mathbf{f}_t)$, where $\mathbf{s}_t$ contains the k_1 slowly moving variables, r_t is the monetary policy instrument, and $\mathbf{f}_t$ contains the k_2 fast moving variables. A slow-moving variable – think of a real variable – is assumed to not react contemporaneously to a monetary policy shock, while a fast-moving variable – think of a financial variable – is assumed to react contemporaneously to monetary policy shocks. The classification of which variables are categorized as slow and which are categorized as fast is provided in Table 8 in Appendix C.1.

Following Banbura et al. (2010), The experiment consists of increasing the federal funds rate by one hundred basis points. In Figure 10 the impulse responses that are generated as the result of a 100 basis point increase to the federal funds rate are displayed. The IRF plots in Figure 10 were estimated on the medium-small dataset. In Appendix C.5 IRF plots estimated on the medium-large dataset and large dataset are depicted in Figure 17 and Figure 18, respectively.

The responses are generally speaking well-behaved. First of all, a monetary contraction has a

Figure 10: **Impulse Responses to a Monetary Policy Shock**

Impulse response functions for to monetary policy shock of the medium-small variables. The model is estimated on the medium-small dataset. The impulse responses are generated as the result of a 100 basis point increase to the federal funds rate (FYFF).



negative effect on real economic activity (GDP251), consumption (GDP252), industrial production (IPS10), and capacity utilization (UTL11). In contrary, unemployment (LHUR) increases. All these phenomena can be intuitively explained: When the interest rate is increased, demand for goods and services tend to decrease, which in turn decreases wages and other costs, resulting in the lower

demand for workers and materials that are necessary for production.

As the model contains more than the standard nominal and real variables, the effect of monetary shocks on housing starts, stock prices and exchange rated can also be studied. The impact on housing starts (`HSFR`), and the effect on stock prices (`FSPIN`) are significantly negative. These two responses result from the general decrease in investment, whether that is investing in housing or stocks. Following a contractionary monetary policy shock, real activity measures decline, prices eventually go down and (`FM1`) and (`FM2`) have an initial negative shock, whereafter they increase steeply. The dividend yields (`FYGT10`) initially jump above the steady state, but go down quite quickly. The US exchange rate (`EXRUS`) appreciates which substantiates the main finding of Eichenbaum and Evans (1995). Overall these results seem to provide theoretical-consistent and sensible interpretations of the effect of monetary policy.

The two inflationary measures in our economy, the consumer price index (`CPIUACSL`) and the producer price index (`PWFSA`), do exhibit the price puzzle, in particular if the model is estimated on the medium-small dataset. As more variables are added (see Figure 17 and Figure 18 in Appendix C.5) the price-puzzle does decrease, but does not fade away completely. In case the model is estimated on the large dataset, the prize-puzzle exhibits the least. However, in that case all responses seem to decrease greatly in their magnitude.

Another popular tool for gaining insight in an estimated VAR is variance decomposition. Variance decomposition is a way to quantify how important each shock is in explaining the variation in each of the variables in the system. The variance decomposition indicates what portion of the variance of the forecast error in predicting $\mathbf{y}_{i,T+h}$ is due to the structural shock $\mathbf{w}_j$ for $j = 1, \ldots, k$. In Table 5 the forecast variance decomposition results of the monetary policy shock are reported.

The variance decompositions give similar results across procedures, which should not come as a surprise considering that the impulse responses across procedures resemble each other. An increase in the amount of variables in the model causes the size of the monetary shock to decrease. This is particularly visible in the dataset with 168 macroeconomic indicators. If variables are added (or deleted) to a VAR, the forecast error variance components will change as a consequence. After all, the forecast errors are conditional on the estimated model.

Table 5: **Variance Decomposition**

Table reports the percentage share of the monetary policy shock in the forecast error variance for $h \in \{1, 3, 6, 12, 24\}$

Group	Horizon	Lasso	Lagwise	Componentwise	Elementwise	Own-Other
Medium-Small ($k = 20$)	1	83.18	84.06	84.38	82.35	83.26
	3	57.06	57.95	55.70	56.46	59/75
	6	54.86	56.08	53.47	52.20	57.68
	12	54.08	55.79	53.08	53.42	56.94
	24	53.92	55.77	53.03	53.28	56.86
Medium-Large ($k = 40$)	1	76.68	74.37	75.78	76.53	77.08
	3	52.32	50.26	51.49	52.42	52.77
	6	51.06	48.95	50.30	51.01	51.24
	12	50.79	48.77	50.10	50.70	50.18
	24	50.67	48.69	50.05	50.58	50.74
Large ($k = 168$)	1	15.52	13.44	15.51	16.10	18.19
	3	10.41	9.16	9.53	10.85	11.43
	6	10.22	8.90	9.18	10.64	11.04
	12	10.19	8.85	9.10	10.61	10.95
	24	10.19	8.85	9.09	10.60	10.95

5 Conclusion and Future Work

This paper assesses the performance of the weighted HLag structures in both a simulation study and an empirical application. The weighted HLag procedures outperform popular existing methods in VAR literature. Throughout the simulation scenarios, the asymptotic behavior of the procedures is studied. As one should expect from lasso-based methods, the parameter estimates were biased, (usually) to a smaller value than the true parameter value. The parameter estimates of the various procedures move closer to each other as the sample size increases. Moreover, the simulation results show that there is marginal improvement in forecasting if the weighted HLag structures are used to estimate the model-parameters compared to using their unweighted counterparts. The weighted HLag structures do substantially increase the generated sparsity in an estimated model. The empirical study

highlighted again the marginal difference (in terms of forecasting performance) between the weighted and unweighted HLags. However, as observed in the simulation study, there is a significant difference in the amount of generated sparsity, in particular as the forecast horizon increases. Via impulse responses the effect of a monetary policy shock on the various variables in the system is studied. The impulse response result in (largely) economically valid interpretations. The prize-puzzle did exhibit in the impulse responses, especially if the system consisted of a small number of macroeconomic indicators. As the amount of indicators increases, the prize-puzzle decreases, but it does not fade away completely.

The work in this paper has considerable room for extensions. Perhaps the most interesting of them is both improving on the functional form of the lag-weighted function, and improving the HLag structures themselves. The weighting function that is used in this work is very simple, and postulating a more suitable weighting function for VARs may significantly improve forecasting and parameter selection. The same improvement may be witnessed if more suitable HLag structures are constructed. In particular, considering application specific HLag structures, in contrast to the 'fixed' HLag structures described in this thesis, would be an interesting study. A way to do this could be by grouping variables based on the correlation they have between them.

A big problem with high-dimensional VARs is that bootstrap inference for structural analysis is often not viable. This means that structural VARs cannot be validly estimated. The reason for this problem is that bootstrap requires re-estimating the the model B number of times, with B the amount of bootstrap samples. If the system is high-dimensional and or if there are multiple tuning parameters in the regularization model, re-estimating the VAR on bootstrapped values is computationally intractable. Researching if there are other ways to validly estimate structural VARs that do not require re-estimating a model B amount of times is definitely a topic worth researching into.

References

Akaike, H. (1974). New Look at Statistical-Model Identification. *IEEE Transactions on Automatic Control*, AC19: 716–723.

Albis, M. and Mapa, D. (2014). Bayesian Averaging of Classical Estimates in Asymmetric Vector Autoregressive (AVAR) Models. *MPRA Paper 55902*.

Banbura, M., Giannone, D., and Reichlin, L. (2010). Large Bayesian Vector Auto Regressions. *Journal of Applied Econometrics*, 25(1): 71–92.

Barigozzi, M., Lippi, M., and Luciani, M. (2016). Non-Stationary Dynamic Factor Models for Large Datasets. (2016-024).

Beck, A. and Teboulle, M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1): 183–202.

Bergmeir, C., Hyndman, R. J., and Koo, B. (2017). A Note on the Validity of Cross-validation for Evaluating Time Series Prediction. *Computational Statistics and Data Analysis*, 120: 70–83.

Bernanke, B., B. J. and Eliasz, P. (2005). Measuring Monetary Policy: A factor augmented Autoregressive (FAVAR) Approach. *Quarterly Journal of Economics*, page 387–422.

Bernanke, B. S. and Blinder, A. (1992). The Federal Funds Rate and the Channels of Monetary Transmission. *American Economic Review*, 82(4): 901–921.

Bernardi, M. and Catania, L. (2014). The Model Confidence Set Package for R. *arXiv:1410.8504*.

Box, G. E. P. and Tiao, G. C. (1977). A Canonical Analysis of Multiple Time Series. *Biometrika*, 64(2): 355–365.

Chambolle, A., De Vore, R. A., Lee, N.-Y., and Lucier, B. J. (1998). Nonlinear Wavelet Image Processing: Variational Problems, Compression, and Noise Removal Through Wavelet Shrinkage. *IEEE Transactions on Image Processing*, 7(3): 319–335.

Christiano, L. J., Eichenbaum, M., and Evans, C. L. (1999). Monetary Policy Shocks: What Have We Learned and to What End. *Handbook of Macroeconomics*, 1(1): 65–148.

Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11): 1413–1457.

Diebold, F. and Mariano, R. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, 13(3): 253–263.

Ding, S. and Karlsson, S. (2014). Bayesian VAR Models with Asymmetric Lags. *Technical report*.

Edwards, S. (2006). The Relationship Between Exchange Rates and Inflation Targeting Revisited. Technical Report 409.

Eichenbaum, M. (1992). Comment on 'Interpreting the Macroeconomic Time Series Facts: The effects of monetary policy'. *European Economic Review*, 36(5): 1001–1011.

Eichenbaum, M. and Evans, C. L. (1995). Some Empirical Evidence on the Effects of Shocks to Monetary Policy on Exchange Rates*. *The Quarterly Journal of Economics*, 110(4): 975–1009.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000). The Generalized Dynamic-Factor Model: Identification and Estimation. *Review of Economics and Statistics*, 82(4): 540–554.

Gonzalo, J. and Pitarakis, J.-Y. (2002). Lag Length Estimation in Large Dimensional Systems. *Journal of Time Series Analysis*, 23(4): 401–423.

Hansen, P. R., Lunda, A., and Nason, J. (2011). The Model Confidence Set for Vector. *Econometrica*, 79: 453–497.

Hanson, M. (2004). The 'Price Puzzle' Reconsidered. *Journal of Monetary Economics*, 51: 1385–1413.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12: 55–67.

Hsiao, C. (1981). Autoregressive Modeling and Money-income Causality Detection. *Journal of Monetary Economics*, 7: 85–106.

Hsu, N., Hung, H., and Chang, Y. (2008). Subset Selection for Vector Autoregressive Processes using Lasso. *Computational Statistics and Data Analysis*, 52: 3645–3657.

Jenatton, R., Mairal, J., Obozinski, G., and Bach (2011). Proximal Methods for Hierarchical Sparse Coding. *Journal of Machine Learning Research*, 12: 2297–2334.

Keating, J. (1993). Asymmetric Vector Autoregression. *Journal of the American Statistical Association*, pages 68–73.

Keogh, E. and Mueen, A. (2017). *Encyclopedia of Machine Learning and Data Mining*, pages 314–315. Springer US, Boston, MA.

Koop, G. (2013). Forecasting with Medium and Large Bayesian VARs. *Journal of Applied Econometrics*, 28(2): 177–203.

Litterman, R. (1986). Forecasting with Bayesian Vector Autoregressions: Five Years of Experience. *Journal of Business Economic Statistics*, 4(1): 25–38.

Lütkepohl, H. (1985). Comparison of Criteria for estimating the Order of a Vector Autoregressive Process. *Journal of Time Series Analysis*, 6(1): 35–52.

Lütkepohl, H. (2007). *New Iintroduction to Multiple Time Series Analysis*. Springer.

Matteson, D. S. and Tsay, R. S. (2011). Dynamic Orthogonal Components for Multivariate Time Series. *Journal of the American Statistical Association*, 106(496).

Na, O. (2017). Generalized Information Criterion for the AR Model. *Journal of the Korean Statistical Society*, 46: 146–160.

Nicholson, W., Matteson, D. S., and Bien, J. (2017). VARX-L: Structed Regurlization for Large Vector Autoregression with Exegenous Variables. *International Journal of Forecasting*, 33: 627–651.

Nicholson, W., Wilms, I., Bien, J., and Matteson, D. (2018). High Dimensional Forecasting via Interpretable Vector Autoregression. *arXiv:1412.5250v3*.

Park, H. and Sakaori, F. (2013). Lag Weighted Lasso for Time Series Model. *Computational Statistics*, 28(2): 493–504.

Prag, J. (1994). The Response of Interest Rates to Unemployment Rate Announcements: Is There a Natural Rate of Unemployment? *Journal of Macroeconomics*, 16(1): 171–184.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2): 461–464.

Sims, C. (1980). Macroeconomics and Reality. *Econometrica*, 48: 1–48.

Song, S. and Bickel, P. J. (2011). Large Vector Auto Regressions. *ArXiv e-prints arXiv:1106.3915*.

Stock, J. H. and Watson, M. W. (2002). Forecasting Using Principal Components from a Large Number of Predictors. *Journal of the American Statistical Association*, 97(460): 1167–1179.

Stock, J. H. and Watson, M. W. (2005). An Empirical Comparison of Methods for Forecasting Using Many Predictors. *Manuscript, Princeton University*, 46.

Tiao, G. C. and Tsay, R. S. (1989). Model Specification in Multivariate Time Series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2): 157–213.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B*, 58(1): 267–288.

Yuan, M. and Lin, Y. (2006). Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society, Series B*, 68(1): 49–67.

Zhao, P., Rocha, G., and Yu, B. (2009). The Composite Absolute Penalties Family for Grouped and Hierarchical Variable Selection. *The Annals of Statistics*, 37(6A): 3468–3497.

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101: 1418–1429.

A Algorithms

In case the penalty function is the lag-weighted lasso $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{\ell=1}^{p} \ell^{\gamma} \|\mathbf{\Phi}^{(\ell)}\|_1$, solving the row-wise proximal problem is done as follows

Algorithm 3: Solution to the lasso proximal operater in Algorithm 2

Require: $\mathbf{x}$, λ, γ, ν;

for $\ell = 1, \ldots, p$ **do**

  $\mathbf{x}^{(\ell)} \leftarrow \mathcal{ST}(\mathbf{x}^{(\ell)}, \nu\lambda\ell^{\gamma})$;

end

return x

Here $\mathcal{ST}$ represents the soft-threshold operator

$$\mathcal{ST}(x, \phi) = \text{sgn}(x)(|x| - \phi)_+,$$

sgn denotes the signum function and $(|x| - \phi)_+ = \max(|x| - \phi, 0)$.

In case the penalty function is the lagwise HLag $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{\ell=1}^{p} \ell^{\gamma} \|\mathbf{\Phi}^{(\ell:p)}\|_2$, solving the proximal problem is done as follows

Algorithm 4: Solution to the lagwise HLag proximal operater in Algorithm 1

Require: $\mathbf{x}$, λ, γ, ν;

for $\ell = p, \ldots, 1$ **do**

  $\mathbf{x}^{(\ell:p)} \leftarrow \left(1 - \frac{\nu\lambda\ell^{\gamma}}{\|\mathbf{x}^{(\ell:p)}\|_2}\right)_+ \mathbf{x}^{(\ell:p)}$;

end

return x

In case the penalty function is the componentwise HLag $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \sum_{i=1}^{k} \sum_{\ell=1}^{p} \ell^{\gamma} \|\mathbf{\Phi}_i^{(\ell:p)}\|_2$, solving the row-wise proximal problem is done as follows

Algorithm 5: Solution to the componentwise HLag proximal operator in Algorithm 2

Require: $\mathbf{x}$, λ, γ, ν;

for $\ell = p, \ldots, 1$ **do**

  $\mathbf{x}^{(\ell:p)} \leftarrow \left(1 - \frac{\nu\lambda\ell^{\gamma}}{\|\mathbf{x}^{(\ell:p)}\|_2}\right)_+ \mathbf{x}^{(\ell:p)}$;

end

return x

In case the penalty function is the elementwise HLag $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{j=1}^{k} \sum_{i=1}^{k} \sum_{\ell=1}^{p} \ell^{\gamma} \|\mathbf{\Phi}_{ij}^{(\ell:p)}\|_2$, solving

the row-wise proximal problem is done as follows

Algorithm 6: Solution to the elementwise HLag proximal operator in Algorithm 2

Require: $\mathbf{x}$, λ, γ, ν;

for $j = 1, \ldots, k$ **do**

  for $\ell = p, \ldots, 1$ **do**

    $\mathbf{x}_j^{(\ell:p)} \leftarrow \left(1 - \frac{\nu\lambda\ell^\gamma}{\|\mathbf{x}_j^{(\ell:p)}\|_2}\right)_+ \mathbf{x}_j^{(\ell:p)}$;

  end

end

return $\mathbf{x}$

In case the penalty function is the own-other HLag $\mathcal{P}(\mathbf{\Phi}; \lambda, \gamma) = \lambda \sum_{i=1}^{k} \sum_{\ell=1}^{p} \ell^\gamma \left(\ell \|\mathbf{\Phi}_{ii}^{(\ell:p)}\|_2 + \ell(k-1)\|\mathbf{\Phi}_{i,-i}^{(\ell:p)}\|_2\right)$, solving the row-wise proximal problem is done as follows

Algorithm 7: Solution to the own-other HLag proximal operator in Algorithm 2

Require: $\mathbf{x}$, λ, γ, ν, i;

for $\ell = p, \ldots, 1$ **do**

  $\mathbf{x}_i{}^{(\ell:p)} \leftarrow \left(1 - \frac{\ell\nu\lambda\ell^\gamma}{\|\mathbf{x}_i^{(\ell:p)}\|_2}\right)_+ \mathbf{x}_i^{(\ell:p)}$;

  $\mathbf{x}_{-i}^{(\ell:p)} \leftarrow \left(1 - \frac{\ell(k-1)\nu\lambda\ell^\gamma}{\|\mathbf{x}_{-i}^{(\ell:p)}\|_2}\right)_+ \mathbf{x}_{-i}^{(\ell:p)}$;

end

return $\mathbf{x}$

B Simulation

B.1 Stationary VARs

Any $\text{VAR}_k(p)$ with $p > 1$ can be written as a $\text{VAR}_k(1)$ model. The resulting $\text{VAR}_k(1)$ is often known as the companion form of $\text{VAR}_k(p)$. The exact specification of this companion form is unnecessary for the present analysis. Rather, only the coefficient matrix of this companion form is of interest and it is denoted by

$$\mathbf{\Theta} = \begin{pmatrix} \mathbf{\Phi}^{(1)} & \mathbf{\Phi}^{(2)} & \dots & \mathbf{\Phi}^{(p-1)} & \mathbf{\Phi}^{(p)} \\ \mathbf{I}_k & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k & & \mathbf{0} & \mathbf{0} \\ \vdots & & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_k & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{kp \times kp}.$$

Since VARs are dynamical models, it is imperative to establish conditions under which the VAR is stable. A condition for the stability of a $\text{VAR}_k(p)$ is that it requires that all the eigenvalues of $\mathbf{\Theta}$ are smaller than one in modulus or all the roots larger than one. Therefore it holds that a $\text{VAR}(p)$ is called stable if

$$\det(\mathbf{I}_{kp} - \mathbf{\Theta} z) = \det(\mathbf{I}_k - \mathbf{\Phi}^{(1)} z, \mathbf{\Phi}^{(2)} z^2, \dots, \mathbf{\Phi}^{(p)} z^p) \neq 0 \quad \text{for} \quad |z| \leq 1.$$

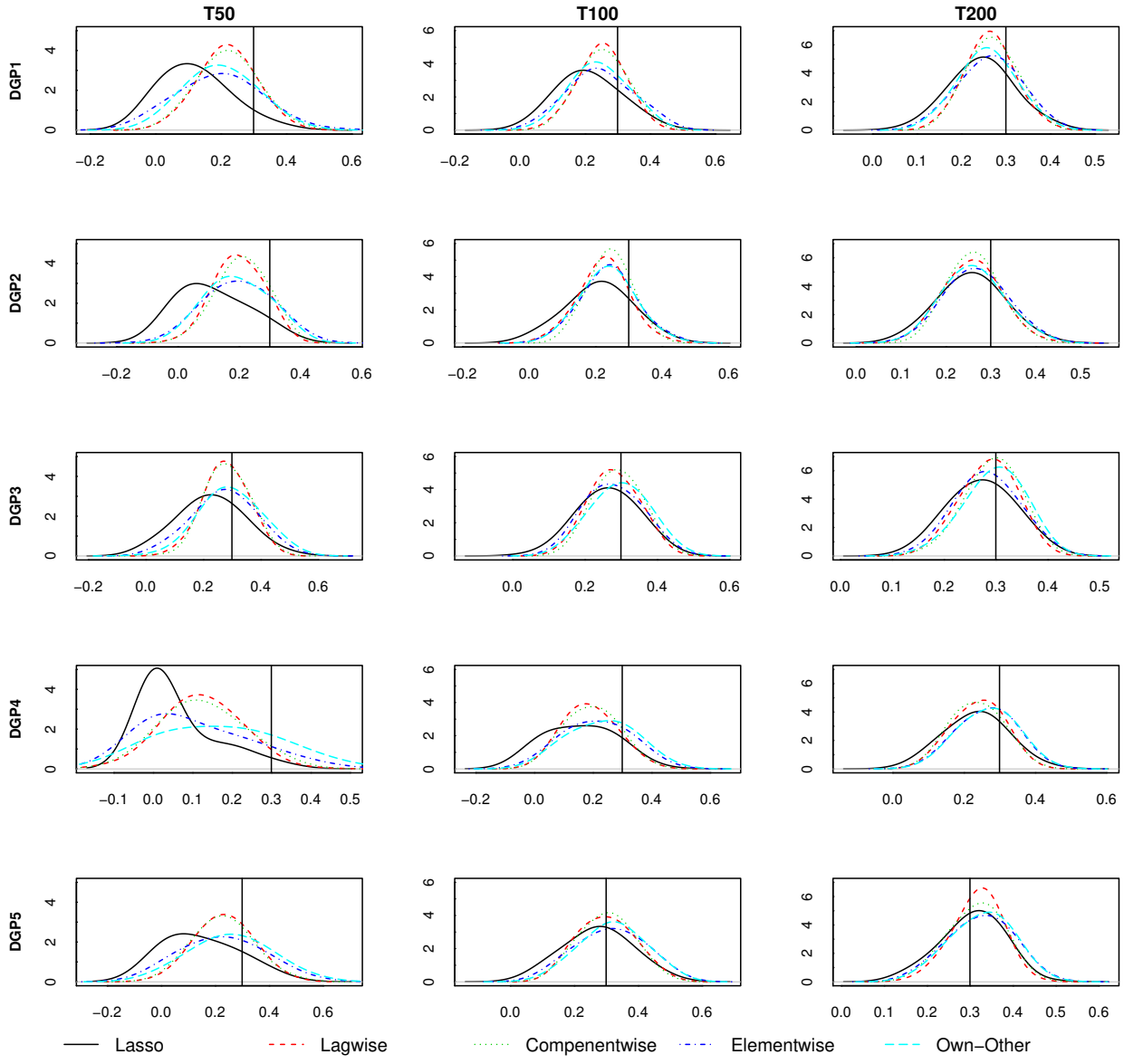
This polynomial is termed the reverse characteristic polynomial of the $\text{VAR}(p)$ process. Hence, the process (2.1) is stable if its reverse characteristic polynomial has no roots in and on the complex unit circle. An important fact is that stability implies stationarity – thus it is sufficient to test for stability to ensure that a VAR is stationary.

To generate stationary VARs, $\mathbf{\Phi}$ needs to be generated such that the eigenvalues $\mathbf{\Theta}$ are smaller than one in modulus. There is no algorithmic procedure that results in matrix coefficient matrices that are both stationary and structured. Instead, I generate structured random parameter matrices until a stationary matrix is acquired.

B.2 Simulation

Figure 11: **Density Plots**

Density plots of the first parameter in the first equation for $T \in \{50, 100, 200\}$, and for various simulation set-ups that were explained in Section 3.2. The true value of the first parameter of the first equation is illustrated by a vertical black line. The plots resulted from the procedures without a weighting parameter.



B.3 Relative Magnitude Plots

Figure 12: **Magnitude Plots**

Plots of the relative magnitudes of the averaged estimates, where the estimations were done on a simple size of $T = 50$. The average is taken of the $N = 100$ estimated $\hat{\mathbf{\Phi}}$s.

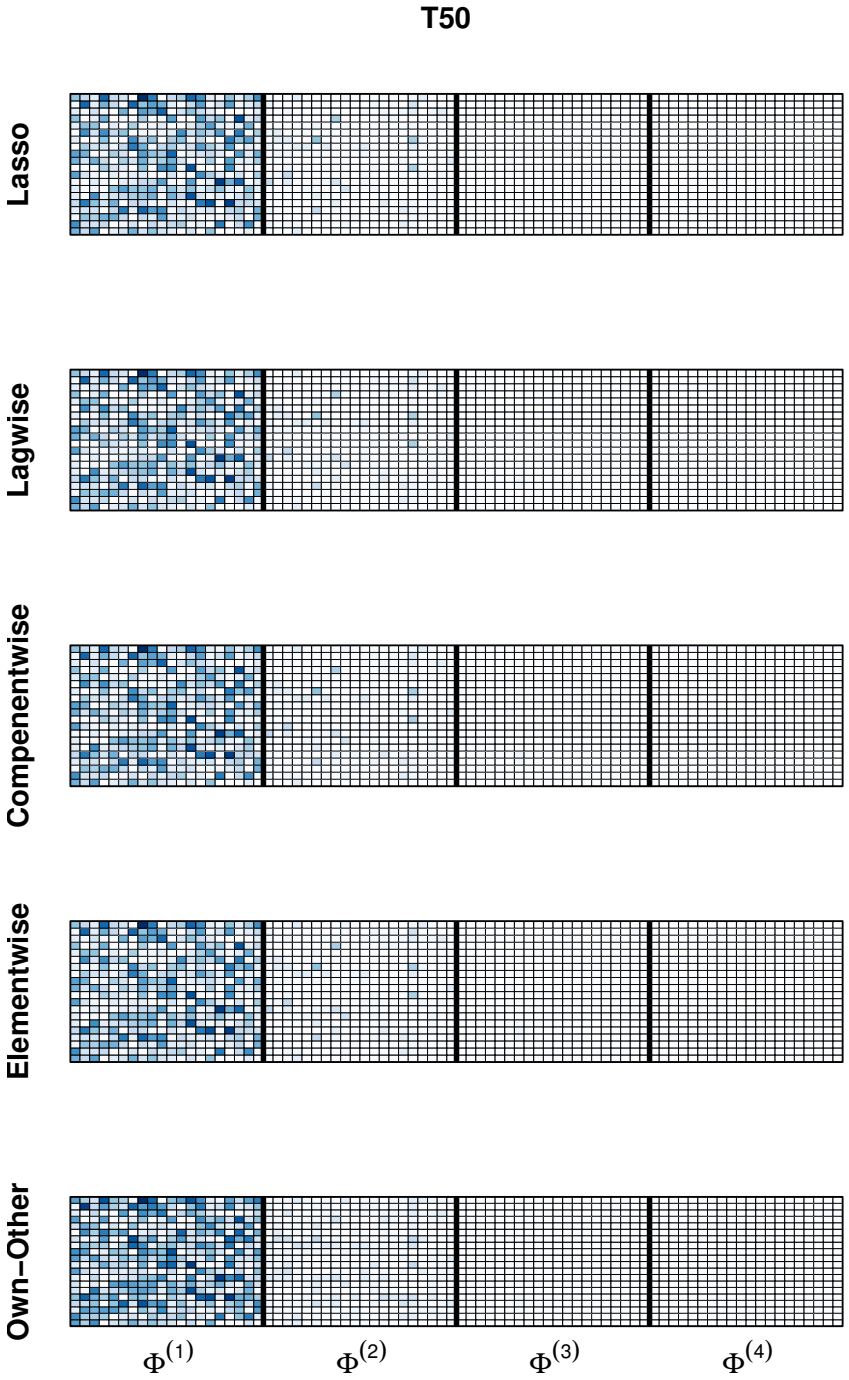


Figure 13: **Magnitude Plots**

Plots of the relative magnitudes of the averaged estimates, where the estimations were done on a simple size of $T = 100$. The average is taken of the $N = 100$ estimated $\hat{\mathbf{\Phi}}$s.

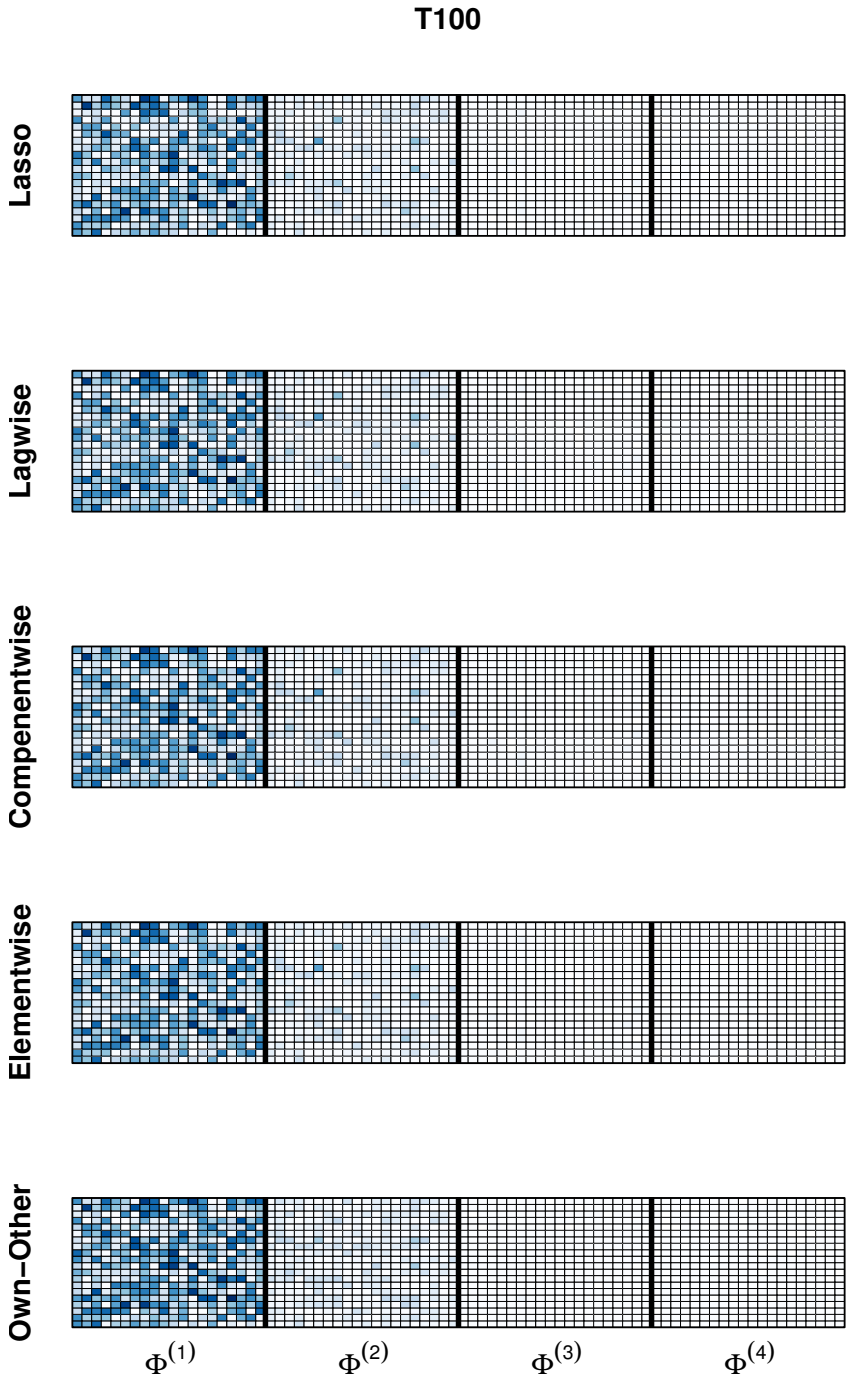
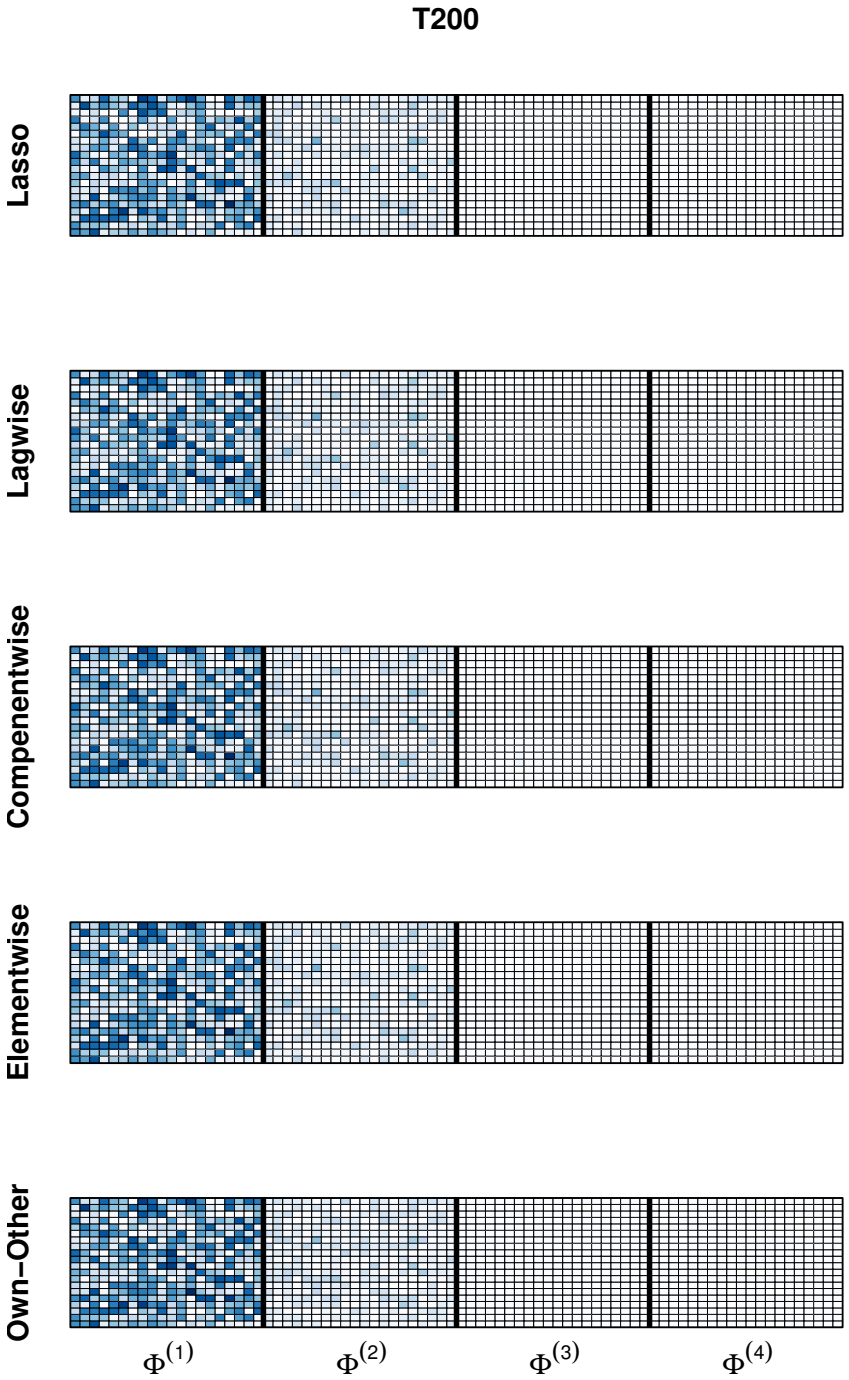


Figure 14: **Magnitude Plots**

Plots of the relative magnitudes of the averaged estimates, where the estimations were done on a simple size of $T = 200$. The average is taken of the $N = 100$ estimated $\hat{\mathbf{\Phi}}$s.



B.4 Standard Errors

Table 6: **Simulation Results**

This table depicts numerical results of several performance measure for each simulation setting, and for various time series lengths. All the values are percentages, and all procedures in this table included a weighting parameter.

		Scenario														
		DGP1			DGP2			DGP3			DGP4			DGP5		
	T	50	100	200	50	100	200	50	100	200	50	100	200	50	100	200
		MSFE Relative to Sample Mean														
Method	Lasso	6.930	2.843	1.972	6.049	3.244	2.007	8.869	4.007	2.933	7.876	9.539	5.338	10.08	4.873	2.696
	Lagwise	5.831	2.631	1.858	5.934	3.113	1.895	7.620	3.756	2.947	7.637	9.690	6.010	7.712	4.410	2.717
	Componentwise	6.280	2.618	1.883	6.701	3.003	1.927	7.894	3.797	2.852	7.608	9.407	5.170	7.867	4.429	2.559
	Elementwise	6.912	2.757	1.932	6.671	3.172	1.968	8.735	3.908	2.870	8.879	9.473	5.322	9.369	4.743	2.670
	Own-Other	5.950	2.663	1.906	6.353	3.032	1.934	7.642	3.830	2.874	7.814	9.326	5.235	7.989	4.479	2.616
	OLS	46.23	42.07	3.152	35.34	31.25	3.245	35.75	24.78	4.781	24.48	26.30	9.989	33.69	27.30	4.017
		MSE of the Parameter Estimates														
	Lasso	0.094	0.030	0.020	0.086	0.026	0.014	0.136	0.030	0.016	0.099	0.097	0.091	0.057	0.030	0.019
	Lagwise	0.040	0.022	0.014	0.105	0.018	0.011	0.035	0.018	0.010	0.101	0.114	0.097	0.030	0.022	0.016
	Componentwise	0.119	0.025	0.017	0.206	0.021	0.012	0.132	0.020	0.012	0.097	0.088	0.085	0.031	0.021	0.015
	Elementwise	0.106	0.028	0.019	0.157	0.022	0.015	0.130	0.020	0.014	0.100	0.100	0.082	0.040	0.027	0.017
	Own-Other	0.044	0.023	0.017	0.158	0.021	0.014	0.082	0.017	0.012	0.101	0.094	0.071	0.032	0.022	0.015
		Lag Selection Accuracy Relative to Sample Mean														
	Lasso	6.248	7.209	10.33	9.838	9.587	10.44	7.485	6.812	8.635	5.523	7.883	4.902	9.455	7.673	6.224
	Lagwise	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Compenentwise	15.23	8.534	3.724	15.70	10.46	9.175	7.656	3.036	2.417	5.842	0.919	1.674	0.176	0.000	0.000
	Elementwise	10.80	9.375	10.85	14.86	11.42	11.97	9.147	9.335	8.082	5.613	4.986	5.796	9.567	8.454	4.531
	Own-Other	6.736	0.492	0.000	4.964	4.443	2.617	1.369	1.395	1.504	1.454	1.314	1.611	0.360	1.297	1.037
		True Positive Rate														
	Lasso	8.970	8.656	11.95	10.70	8.805	10.76	12.41	7.554	8.701	3.329	2.880	5.147	-	-	-
	Lagwise	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	-
	Componentwise	15.35	8.534	3.724	14.12	9.066	7.895	9.969	3.732	2.770	9.541	1.583	3.470	-	-	-
	Elementwise	19.52	11.65	14.07	27.73	11.12	13.11	17.75	9.930	9.180	4.340	9.170	9.080	-	-	-
	Own-Other	8.450	1.221	0.759	4.101	3.758	2.255	1.053	0.526	0.725	2.190	1.000	0.704	-	-	-
		True Negative Rate														
	Lasso	9.907	6.068	4.431	7.888	5.420	4.420	9.257	5.088	3.998	7.546	7.221	6.055	6.216	6.246	5.544
	Lagwise	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Componentwise	0.250	0.000	0.000	1.065	0.380	0.000	2.364	0.569	0.180	2.158	0.075	0.165	0.176	0.000	0.000
	Elementwise	11.670	5.170	3.192	13.53	5.357	3.802	11.18	5.328	2.694	11.75	10.70	5.675	8.049	5.834	2.945
	Own-Other	0.723	0.221	0.135	0.481	0.268	0.210	0.694	0.365	0.292	1.373	0.812	0.443	0.492	0.367	0.247

B.5 Unweighted

Table 7: **Simulation Results**

This table depicts numerical results of several performance measure for each simulation setting, and for various time series lengths. All procedures in this table included a weighting parameter.

		Scenario														
		DGP1			DGP2			DGP3			DGP4			DGP5		
	T	50	100	200	50	100	200	50	100	200	50	100	200	50	100	200
		MSFE Relative to Sample Mean														
Method	Lasso	0.227	0.128	0.085	0.239	0.118	0.074	0.322	0.154	0.103	0.938	0.839	0.746	0.402	0.227	0.152
	Lagwise	0.189	0.103	0.076	0.173	0.099	0.067	0.230	0.130	0.093	0.857	0.772	0.693	0.298	0.188	0.140
	Componentwise	0.193	0.104	0.077	0.175	0.096	0.066	0.232	0.128	0.096	0.862	0.771	0.700	0.301	0.189	0.139
	Elementwise	0.219	0.110	0.079	0.194	0.102	0.069	0.261	0.134	0.097	0.896	0.789	0.708	0.344	0.202	0.144
	Own-Other	0.200	0.108	0.078	0.179	0.099	0.068	0.238	0.132	0.097	0.847	0.762	0.693	0.303	0.191	0.141
	OLS	2.003	1.919	0.124	1.167	1.073	0.111	1.796	1.704	0.154	1.692	1.528	1.252	2.285	2.354	0.210
		MSE of the Parameter Estimates														
	Lasso	0.012	0.007	0.004	0.010	0.006	0.003	0.011	0.006	0.003	0.006	0.007	0.008	0.011	0.007	0.005
	Lagwise	0.006	0.004	0.002	0.005	0.003	0.002	0.006	0.004	0.002	0.007	0.008	0.008	0.007	0.005	0.003
	Componentwise	0.007	0.004	0.003	0.006	0.003	0.002	0.006	0.004	0.002	0.008	0.008	0.008	0.007	0.005	0.003
	Elementwise	0.008	0.005	0.003	0.007	0.004	0.003	0.007	0.004	0.003	0.006	0.008	0.008	0.009	0.006	0.004
	Own-Other	0.007	0.004	0.003	0.006	0.004	0.002	0.006	0.004	0.003	0.007	0.008	0.008	0.007	0.005	0.002
		Lag Selection Accuracy Relative to Sample Mean														
	Lasso	0.821	0.846	0.934	1.093	1.070	0.123	0.788	0.816	0.906	0.963	0.805	0.744	0.685	0.344	0.176
	Lagwise	1.000	1.000	1.000	1.162	1.162	1.162	1.060	1.060	1.060	0.905	0.905	0.905	0.000	0.000	0.000
	Compenentwise	0.998	1.000	1.000	1.160	1.162	1.162	1.059	1.060	1.060	0.904	0.907	0.905	0.000	0.000	0.000
	Elementwise	0.799	0.861	0.929	1.011	1.054	1.099	0.751	0.812	0.892	0.924	0.777	0.735	0.688	0.332	0.159
	Own-Other	1.000	1.000	1.000	1.188	1.175	1.167	1.055	1.056	1.055	0.99	0.967	0.950	0.049	0.037	0.021
		True Positive Rate														
	Lasso	0.740	0.510	0.295	0.615	0.521	0.332	0.720	0.563	0.379	0.921	0.802	0.666	-	-	-
	Lagwise	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	-
	Componentwise	0.002	0.000	0.000	0.002	0.000	0.000	0.001	0.000	0.000	0.042	0.003	0.001	-	-	-
	Elementwise	0.646	0.365	0.212	0.523	0.390	0.227	0.622	0.408	0.245	0.943	0.744	0.571	-	-	-
	Own-Other	0.018	0.009	0.005	0.018	0.014	0.008	0.016	0.010	0.006	0.003	0.001	0.000	-	-	-
		True Negative Rate														
	Lasso	0.433	0.712	0.877	0.544	0.706	0.859	0.436	0.667	0.831	0.131	0.333	0.540	0.322	0.588	0.786
	Lagwise	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Componentwise	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.992	1.000	0.999	1.000	1.000	1.000
	Elementwise	0.719	0.927	0.980	0.759	0.884	0.959	0.684	0.876	0.957	0.268	0.579	0.792	0.510	0.803	0.920
	Own-Other	0.994	0.998	1.000	0.995	0.997	0.997	0.988	0.996	0.998	0.973	0.987	0.994	0.985	0.994	0.998

C Empirical

C.1 Data Description

Table 8: **Dataset Description**

An overview of the complete dataset used in this paper. The mnemonic, description, to which partition they belong, and whether the variable reacts fast to a monetary shock or not, are all described in this table.

Mnemonic	Description	Group(s)	Fast or Slow?
GDP251	Real GDP, Quantity Index (2000=100)	MS/ML/L	S
CIPUACSL	CPI All Items	MS/ML/L	S
GDP252	Real Personal Cons. Exp., Quantity Index	MS/ML/L	S
IPS10	Industrial production index: total	MS/ML/L	S
UTL11	Capacity utilization: manufacturing (SIC)	MS/ML/L	S
LHUR	Unemp. rate: All workers, 16 and over (%)	MS/ML/L	S
HFSR	Housing starts: Total (thousands)	MS/ML/L	S
PWFSA	Producer price index: finished goods	MS/ML/L	S
GDP273	Personal Consumption Exp.: price index	MS/ML/L	S
CES275R	Real avg hrly earnings, non-farm prod. workers	MS/ML/L	S
CES002	Employees, nonfarm: total private	MS/ML/L	S
PMI	Purchasing managers' index	ML/L	S
PMDEL	NAPM vendor deliveries index (%)	ML/L	S
PMCP	NAPM commodity price index (%)	ML/L	S
GDP256	Real gross private domestic investment	ML/L	S
LBOUT	Output per hr: all persons, business sec	ML/L	S
PMNV	NAPM inventories index (%)	ML/L	S
GDP263	Real exports	ML/L	S
GDP264	Real imports	ML/L	S
GDP265	Real govt cons expenditures & gross investment	ML/L	S
LBMNU	Hrs of all persons: nonfarm business sector	ML/L	S
PMNO	NAPM new orders index (%)	ML/L	S
PMP	NAPM production index (%)	ML/L	S
GDP276_1	Housing price index	ML/L	S
GDP270	Real final sales to domestic purchasers	ML/L	S
GDP253	Real personal cons expenditures: Durable goods	ML/L	S
LHEL	Index of help-wanted ads in newspapers	ML/L	S
GDP254	Real personal consumption exp: nondur goods	L	S
GDP255	Real personal consumption exp: services	L	S
GDP257	Real gross priv domestic inv: fixed inv	L	S
GDP258	Real gross priv domestic inv: nonresidential	L	S
GDP259	Real gross priv domestic inv: nonres structures	L	S
GDP260	Real gross priv domestic inv: nonres equip	L	S
GDP261	Real gross priv domestic inv: residential	L	S

Table 8 – continued from previous page

Mnemonic	Description	Group(s)	Fast or Slow?
GDP266	Real gov cons exp & gross inv: federal	L	S
GDP267	Real gov cons exp & gross inv: state and local	L	S
GDP268	Real final sales of domestic product	L	S
GDP269	Real gross domestic purchases	L	S
GDP271	Real gross national product	L	S
GDP272	Gross domestic product, price index	L	S
GDP274	Personal cons exp: durable goods, price index	L	S
GDP275	Personal cons exp: nondur goods, price index	L	S
GDP276	Personal cons exp: services, price index	L	S
GDP277	Gross private domestic investment, price index	L	S
GDP278	Gross priv dom inv: fixed inv, price index	L	S
GDP279	Gross priv dom inv: nonresidential, price index	L	S
GDP280	Gross priv dom inv: nonres structures, price index	L	S
GDP281	Gross priv dom inv: nonres equipment, price index	L	S
GDP282	Gross priv dom inv: residential, price index	L	S
GDP284	Exports, price index	L	S
GDP285	Imports, price index	L	S
GDP286	Government cons exp & gross inv, price index	L	S
GDP287	Gov cons exp & gross inv: federal, price index	L	S
GDP288	Gov cons exp & gross inv: state & local, price index	L	S
GDP289	Final sales of domestic product, price index	L	S
GDP290	Gross domestic purchases, price index	L	S
GDP291	Final sales to domestic purchasers, price index	L	S
GDP292	Gross national product, price index	L	S
LBPUR7	Real comp per hour: employees, nonfarm business	L	S
LBLCPU	Unit labor cost: nonfarm business sector	L	S
GDP274_1	Motor vehicles and parts, price index	L	S
GDP274_2	Furniture and household equipment, price index	L	S
GDP274_3	Other durables, price index	L	S
GDP275_1	Food, price index	L	S
GDP275_2	Clothing and shoes, price index	L	S
GDP275_3	Gas, fuel oil, and other energy goods, price index	L	S
GDP275_4	Other nondurables, price index	L	S
GDP276_2	Household operation, price index	L	S
GDP276_3	Electricity and gas, price index	L	S
GDP276_4	Other household operation, price index	L	S
GDP276_5	Transportation, price index	L	S
GDP276_6	Medical care, price index	L	S
GDP276_7	Recreation, price index	L	S
GDP276_8	Other services, price index	L	S
GDP284_1	Exports of goods, price index	L	S
GDP284_2	Exports of services, price index	L	S

Table 8 – continued from previous page

Mnemonic	Description	Group(s)	Fast or Slow?
GDP285_1	Imports of goods, price index	L	S
GDP285_2	Imports of services, price index	L	S
IPS11	Industrial production index: products total	L	S
IPS299	Industrial production index: final products	L	S
IPS12	Industrial production index: consumer goods	L	S
IPS13	Industrial production index: consumer durable	L	S
IPS18	Industrial production index: consumer nondur	L	S
IPS25	Industrial production index: business equipment	L	S
IPS32	Industrial production index: materials	L	S
IPS34	Industrial production index: dur goods materials	L	S
IPS38	Industrial production index: nondur goods materials	L	S
IPS43	Industrial production index: manufacturing	L	S
IPS307	Industrial production index: residential utilities	L	S
IPS306	Industrial production index: fuels	L	S
CES275	Avg hrly earnings, prod wrkrs, nonfarm-goods prod	L	S
CES277	Avg hrly earnings, prod wrkrs, nonfarm-construction	L	S
CES278	Avg hrly earnings, prod wrkrs, nonfarm-manufacturing	L	S
CES277R	Real avg hrly earnings, prod wrkrs, nonfarm-const	L	S
CES278R	Real avg hrly earnings, prod wrkrs, nonfarm-manuf	L	S
CES003	Employees, nonfarm: goods-producing	L	S
CES006	Employees, nonfarm: mining	L	S
CES011	Employees, nonfarm: construction	L	S
CES015	Employees, nonfarm: manufacturing	L	S
CES017	Employees, nonfarm: durable goods	L	S
CES033	Employees, nonfarm: nondurable goods	L	S
CES046	Employees, nonfarm: service providing	L	S
CES048	Employees, nonfarm: trade, transport and utilities	L	S
CES049	Employees, nonfarm: wholesale trade	L	S
CES053	Employees, nonfarm: retail trade	L	S
CES088	Employees, nonfarm: financial activities	L	S
CES140	Employees, nonfarm: government	L	S
LHELX	Ratio: Help-wanted ads to number unemployed	L	S
LHEM	Civilian labor force employed, total	L	S
LHHAG	Civilian labor force employed, nonagric ind.	L	S
LHU680	Average unemployment duration (weeks)	L	S
LHU5	Unemp by duration, persons unemp less than 5 wks	L	S
LHU14	Unemp by duration, persons unemp btwn 5 and 14 wks	L	S
LHU15	Unemp by duration, persons unemp 15 wks or more	L	S
LHU26	Unemp by duration, persons unemp btwn 15 and 26 wks	L	S
LHU27	Unemp by duration, persons unemp 27 wks or more	L	S
CES151	Avg wkly hours, prod wrks, nonfarm goods-producing	L	S
CES155	Avg weekly overtime hrs, prod wrkrs, nonfarm, manuf	L	S

Table 8 – continued from previous page

Mnemonic	Description	Group(s)	Fast or Slow?
HSBR	Housing authorized: total new private housing units	L	S
HSNE	Housing starts: Northeast	L	S
HSMW	Housing starts: Midwest	L	S
HSSOU	Housing starts: South	L	S
HSWST	Housing starts: West	L	S
CPILFESL	CPI less food and energy	L	S
PCEPILFE	PCE price index less food and energy	L	S
PWFCSA	Producer price index: finished consumer goods	L	S
PWIMSA	Producer price index: interm mat supplies & components	L	S
PWCMSA	Producer price index: crude materials	L	S
PWCMSAR	Real prod price index: crude mat (PWCMSA/PCEPILFE)	L	S
PW561	Producer price index: crude petroleum	L	S
PW561R	PPI crude (relative to core PCE) (PW561/PCEPILFE)	L	S
MOCMQ	New orders (net): consumer goods and materials	L	S
MSONDQ	New orders: nondefense capital goods	L	S
FYFF	Interest rate: Federal funds (effective) (% per annum)	MS/ML/L	**R**
PSCCOMR	Real spot market price index: all commodities	MS/ML/L	F
FRMNBA	Depository inst reserves: nonborrowed (mil$)	MS/ML/L	F
FMRRA	Depository inst reserves: total (mil$)	MS/ML/L	F
FM2	Money stock: M2 (bil$)	MS/ML/L	F
FM1	Money stock: M1 (bil$)	MS/ML/L	F
FSPIN	S&P's common stock price index: industrials	MS/ML/L	F
FYGT10	Interest rate: US treasury const. mat., 10-yr	MS/ML/L	F
EXRUS	US effective exchange rate: index number	MS/ML/L	F
SFYGT10	Spread btwn 10 year and 3 month T-bill rates	ML/L	F
HHSNTN	Univ of Mich index of consumer expectations	ML/L	F
CCINRV	Consumer credit outstanding: nonrevolving	ML/L	F
BUSLOANS	Comm. and industrial loans at all comm. banks	ML/L	F
FYGM3	Interest rate: US T-bills, sec mkt, 3-month	L	F
FYGM6	Interest rate: US T-bills, sec mkt, 6-month	L	F
FYGT1	Interest rate: US T-bills const maturities 1-yr	L	F
FYGT5	Interest rate: US T-bills const maturities 5-yr	L	F
FYGT10	Interest rate: US T-bills const maturities 10-yr	L	F
FYAAAC	Bond yield: Moody's AAA corporate	L	F
FYBAAC	Bond yield: Moody's BAA corporate	L	F
SFYGM6	Spread: 6 month minus 3 month T-bill	L	F
SYGT1	Spread: 1 year minus 3 month T-bill	L	F
SFYAAAC	Spread: AAA corporate minus 10 yr T-bill	L	F
SFYBAAC	Spread: BAA corporate minus 10 yr T-bill	L	F
MZMSL	MZM FRB St. Louis	L	F
FMFBA	Monetary base, adj for res requirement changes	L	F
PSCCOM	Spot market price index: all commodities	L	F

Table 8 – continued from previous page

Mnemonic	Description	Group(s)	Fast or Slow?
EXRSW	Swiss francs per US$	L	F
EXRJAN	Japanese yen per US$	L	F
EXRUK	Cents per pound	L	F
EXRCAN	Canadian $ per US$	L	F
FSPCOM	S&P's common stock price index: composite	L	F
FSDXP	S&P's composite common stock: dividend yield	L	F
FSPXE	S&P's composite common stock: price-earnings ratio	L	F
FSDJ	Dow Jones industrial average common stock price	L	F

C.2 Unweighted

Table 9: **Forecast Results**

(Absolute) MSFE and sparsity percentage for each partition and for the horizons $h \in \{1, 2, 4\}$. The procedures in this table did not incorporate a weighting parameter.

Group		$h = 1$		$h = 2$		$h = 4$	
$k = 20$	**Method**	MSFE	Sp. (%)	MSFE	Sp. (%)	MSFE	Sp. (%)
	Lasso	0.706	80.44	0.760	89.00	0.806	89.00
	Lagwise	0.725	0	0.785	0	0.828	0
	Componentwise	0.730	0	0.770	15.00	0.815	15.00
	Elementwise	0.725	58.75	0.758	83.75	0.806	89.88
	Own-Other	0.702	4.69	0.758	15.96	0.807	37.44
$k = 40$	**Method**	MSFE	Sp. (%)	MSFE	Sp. (%)	MSFE	Sp. (%)
	Lasso	0.557	87.70	0.617	93.22	0.681	93.23
	Lagwise	0.573	0	0.638	0	0.690	0
	Componentwise	0.569	5.620	0.622	23.75	0.681	23.75
	Elementwise	0.558	81.75	0.609	94.52	0.675	94.56
	Own-Other	0.544	2.48	0.606	17.25	0.678	17.30
$k = 168$	**Method**	MSFE	Sp. (%)	MSFE	Sp. (%)	MSFE	Sp. (%)
	Lasso	0.560	96.10	0.649	98.34	0.695	98.34
	Lagwise	0.594	0	0.660	0	0.708	0
	Componentwise	0.598	0.30	0.657	8.63	0.706	10.86
	Elementwise	0.552	94.56	0.642	97.33	0.696	97.34
	Own-Other	0.543	1.04	0.628	7.86	0.683	7.87

C.3 Diebold-Mariano

Table 10: **Results Forecasts**

The p-values resulting from the Diebold-Mariano test applied to each procedure, for every h and k.

Group		$h = 1$	$h = 2$	$h = 4$
$k = 20$	**Method**			
	Lasso	0.316	0.837	0.892
	Lagwise	0.399	-	-
	Componentwise	0.099	-	0.831
	Own-Other	0.213	0.513	0.832
	Elementwise	0.832	0.347	0.695
$k = 40$	**Method**			
	Lasso	0.482	0.095	0.176
	Lagwise	-	-	0.141
	Componentwise	-	-	0.476
	Own-Other	0.002	-	0.236
	Elementwise	-	0.297	0.176
$k = 168$	**Method**			
	Lasso	0.738	0.016	0.546
	Lagwise	-	-	0.000
	Componentwise	-	-	0.000
	Own-Other	0.748	0.002	0.006
	Elementwise	-	-	0.050

C.4 Lag Matrices

Figure 15: **Estimated Lag Matrix**

Estimated lag matrix $\hat{\mathbf{L}}$. The complete matrix is depicted. The model is estimated on the medium-small dataset.

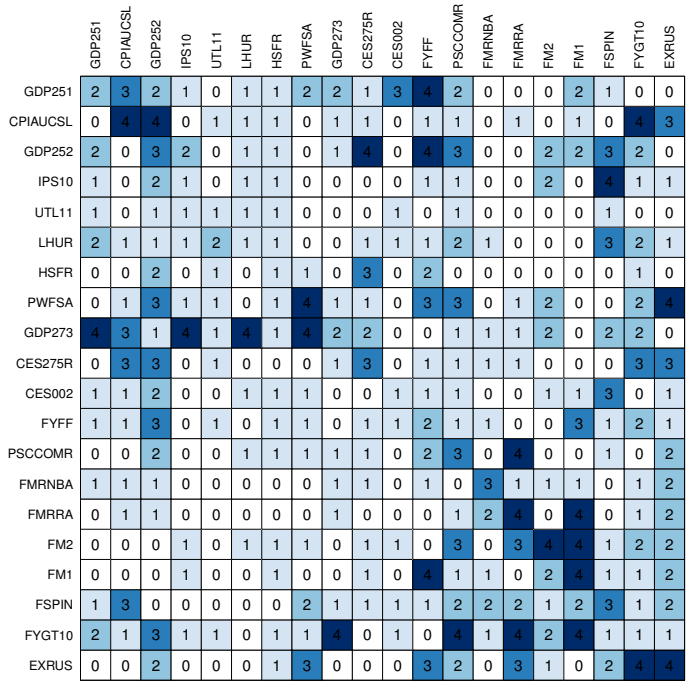
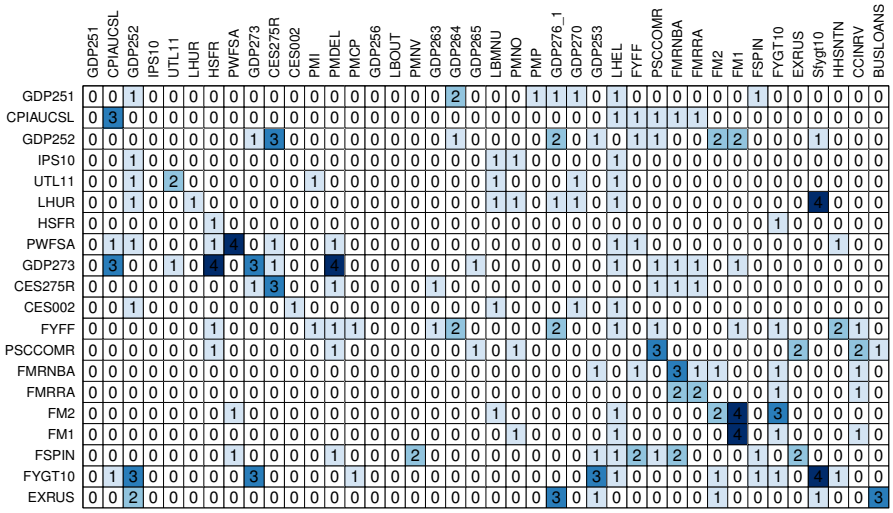


Figure 16: **Estimated Lag Matrix**

Estimated lag matrix $\hat{\mathbf{L}}$, where, for sake of brevity, only the medium-small macroeconomic variables' estimated lags of only the medium-small variables are depicted. The model is estimated on the large dataset.



C.5 Impulse Responses

Figure 17: **Impulse Responses to a Monetary Policy Shock**

Impulse response functions for to monetary policy shock of the medium-small variables. The model is estimated on the medium-large dataset. The Impulse responses are generated as the result of a 100 basis point increase to the federal funds rate (FYFF).

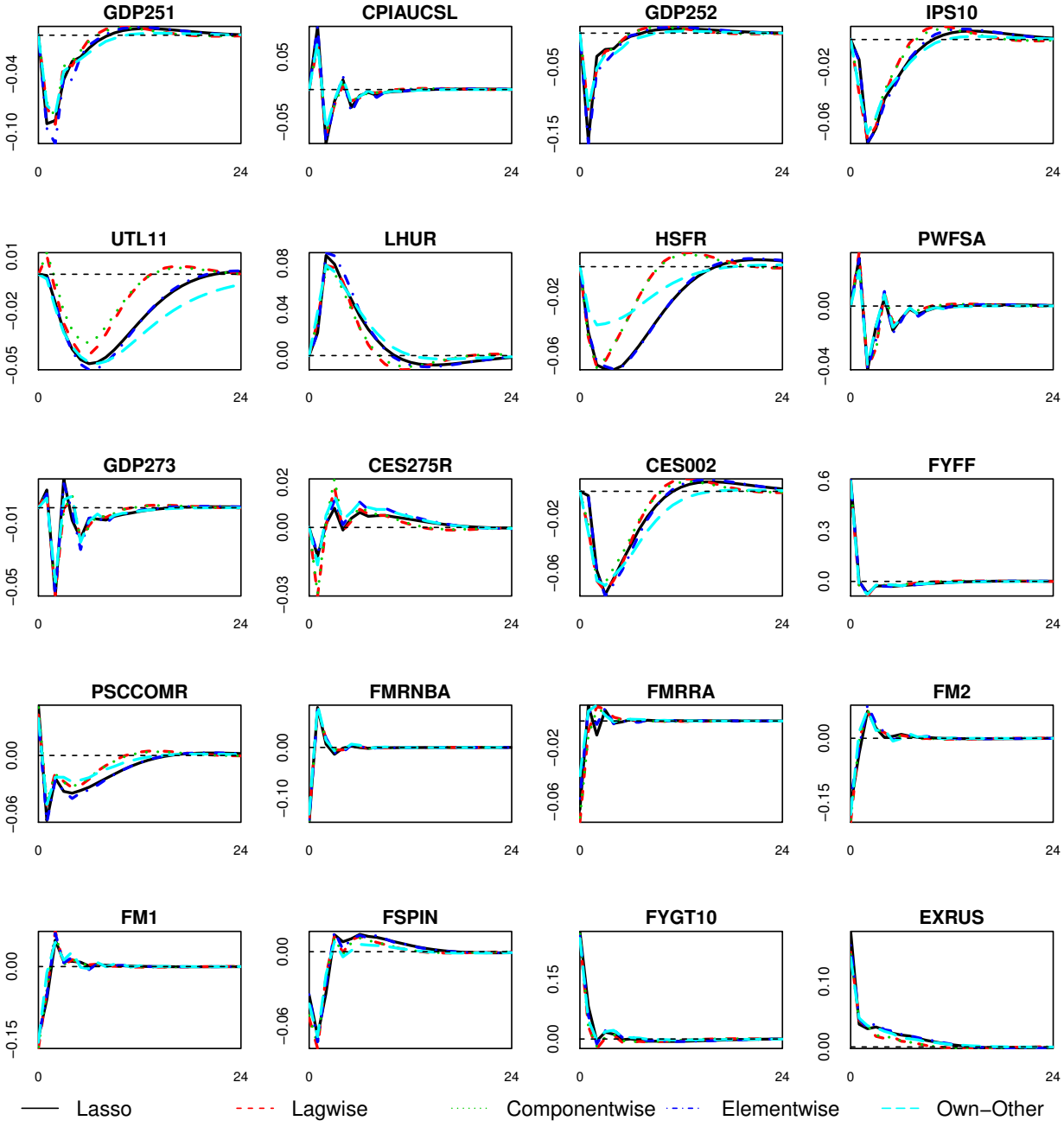


Figure 18: **Impulse Responses to a Monetary Policy Shock**

Impulse response functions for to monetary policy shock of the medium-small variables. The model is estimated on the large dataset. The Impulse responses are generated as the result of a 100 basis point increase to the federal funds rate (FYFF).