A Bayesian VAR toolbox for MATLAB*

Inference, Prediction and Causality

Fabio Canova
Norwegian Business School, CAMP and CEPR

Filippo Ferroni
Chicago Fed

April 30, 2020
(First draft: August 2019)

Abstract

The Bayesian VAR toolbox is a new package that includes MATLAB functions to estimate VAR models with Bayesian methods. The toolbox allows to conduct inference under various prior assumptions, to produce point and density forecasts and to trace the causal effect of shocks using a number of identification schemes. The toolbox is also equipped to handle missing observations. This paper describes the methodology and implementation of the functions and illustrates their use with examples.

Keywords: VAR, Bayesian Inference, Identification, Forecasts, Missing Values, MATLAB

JEL classification: E52, E32, C10

*The views in this paper are solely the responsibility of the authors and should not be interpreted as reflecting the views of the Federal Reserve Bank of Chicago or any other person associated with the Federal Reserve System.

Contents

1 Introduction

The Bayesian VAR toolbox is written in MATLAB. It is available as free software, under the GNU General Public License version 3, and can be downloaded from

https://github.com/naffe15/BVAR_

with all the supplementary material (data and source code) to replicate the examples presented in this paper. The toolbox collects into a set of routines programs that the authors have used for Ph.D. courses or for classes in central banks and international institutions. The toolbox comes as it is. We assume no responsibility whatsoever for its use by other parties and make no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgment by citation of this working paper when the software is used. Users may inform the authors via email[1] of potential problems with the routines if they emerge.

The toolbox deals specifically with fixed coefficient Bayesian VAR models and has a number of identification options. It also covers mixed frequency VARs and FAVARs models. It does not cover topics like time varying coefficients VAR models, models with stochastic volatility, panel VARs, non-gaussian VARs, non-causal VARs, and does not have an option for identification via volatility changes. There are also a number of other issues that are not explicitly dealt with, e.g. the critique of sign restrictions identification discussed in Baumeister and Hamilton (2015). Whenever possible we describe tricks that can help to deal with models not specifically covered in the paper.

The toolbox is intended to users which are dissatisfied by the inflexibility of packages like BEAR or that STATA, think that existing routine are too mechanical, but do not want to get directly involved in programming the empirical tools they need; and are more concerned with the economics rather than the econometrics of the exercises.

Other toolboxes covering similar topics exist, e.g. the VAR toolbox of Ambrogio Cesa-Bianchi, the Econometric toolbox of James LaSage, the dynare project, the Global VAR toolbox of Vanessa Smith, the mixed frequency VAR toolbox of Brave, Butters and Kelley (2020), the Bayesian VAR and Bayesian Local Projection of Miranda-Agrippino and Ricco (2017), as well as series of codes from Haroon Mumtaz or Dimitris Korobilis. To the best of our knowledge, we are the first to mechanize new identification procedures and to provide ready available tools to estimate mixed frequency and FAVAR models.

2 Getting Started

Getting started is easy. Open MATLAB and add the toolbox to the MATLAB path. Assuming that it is stored in `C:\USER_LOCATION`, the toolbox is added to the MATLAB path by typing in the command window

```
addpath C:\USER_LOCATION\BVAR_\v4.1
```

[1]Our emails are: fabio.canova@bi.no; fferroni@frbchi.org

This has to be done anytime you begin a Matlab session. If you want to permanently add the toolbox to the MATLAB PATH, go to the set-path icon in the MATLAB HOME window, click on it, add with the subfolders `BVAR_\v4.1` and Save. A similar procedure should be used to install Chris Sims' routines[2]; which are used to optimally select the hyper-parameters of the Minnesota prior. In this case, type in the command window

```
addpath C:\USER_LOCATION\BVAR_\cmintools
```

2.1 Loading the Data

The user needs to prepare the data she want to use beforehand. Data can be formatted in any way the users wants but needs to be loaded in the MATLAB workspace and collected in a matrix format (of dimension $T \times n$), where time occupies the first dimension (so T is the sample size) and different variables the second (so n is the number of variables). For instance, in the forecasting example we use monthly Euro Area data for the one-year Euribor (`Euribor1Y`), the log of HICP (`HICP`), the log of HICP excluding food and energy (`CORE`), the log of industrial production index (`IPI`) and the log of M3 (`M3`) from 2000m1 2015m8. The data is stored in the MATLAB file `Data.mat`, which is located in `C:\USER_LOCATION\BVAR_\v4.1\BVAR tutorial`.

In principle, there is no need to specify the frequency: the time unit can be months, quarters, years, or mixed monthly and quarterly. Section 6 describe how to use the toolkit when some data are missing.

Data already converted to MATLAB format is loaded with the standard command

```
load Data
```

Data available in spreadsheets is loaded in the workspace with the command

```
y = xlsread('filename');
```

For details about reading in data formatted in other ways, see the MATLAB manual.

2.2 Some notation

A vector of autoregressive model with p lags, $VAR(p)$, is written as

$$y_t = \Phi_1 y_{t-1} + ... + \Phi_p y_{t-p} + \Phi_0 + u_t = \Phi(L) y_{t-1} + \Phi_0 + u_t \tag{1}$$

where y_t is $n \times 1$ vector of endogenous variables, Φ_0 is $n \times 1$ vector of constants and $\Phi_j, j = 1, ..., p$ are $n \times n$ matrices; u_t is an i.i.d. normally distributed, zero mean, random vector with covariance matrix Σ. The vector of parameters to be estimated is $\vartheta = vec(\Phi_0, \Phi_1, ..., \Phi_p, \Sigma)$.

[2]The toolbox can be downloaded from http://sims.princeton.edu/yftp/optimize/ or from https://github.com/naffe15/BVAR_

Any $VAR(p)$ model can be expressed in a companion $VAR(1)$ form. A companion form is useful to generate forecasts and to compute impulse response functions. The companion form is

$$\underset{(np\times 1)}{\mathbf{x}_t} = \underset{(np\times np)}{F}\mathbf{x}_{t-1} + \underset{(np\times 1)}{F_0} + \underset{(np\times n)}{G}u_t \tag{2}$$

$$\mathbf{x}_t = \begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+1} \end{pmatrix} \quad F = \begin{pmatrix} \Phi_1 & \Phi_2 & \dots & \Phi_{p-1} & \Phi_p \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \\ 0 & 0 & \dots & I & 0 \end{pmatrix} \quad G = \begin{pmatrix} I \\ 0 \\ \vdots \\ 0 \end{pmatrix} F_0 = \begin{pmatrix} \Phi_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where I is a $n \times n$ identity matrix and 0 is a $n \times n$ matrices of zeros. For computing forecast error variance and historical decompositions, it is useful to write the the VAR model in its VMA form:

$$y_t = u_t + \Psi_1 u_{t-1} + ... + \Psi_t u_1 + \overline{\Psi}_t \tag{3}$$

where Ψ_j for $j = 1, ..., t$ are functions of $(\Phi_1, ..., \Phi_p)$ and $\overline{\Psi}_t$ is a deterministic function of the constant matrix Φ_0, of the autoregressive matrices $(\Phi_1, ..., \Phi_p)$ and of the initial conditions $(y_0, y_{-1}, ..., y_{-p})$.[3]

3 Inference

The toolkit allows to estimate ϑ and conducts inference conditional on the estimates of ϑ one obtains. Estimation can be performed using both classical (flat-prior) and Bayesian techniques. By default, a Jeffrey (uninformative) prior is assumed; other options for the prior are available. Given the prior, draws are generated from the posterior distribution of the parameters using a Gibbs sampler algorithm; see section A.3 for more details. The baseline estimation function is

```
[BVAR] = bvar(y, lags, options)
```

The first input is the data, `y`, where for now, we assume that there are no missing values. The second input is the number of lags, `lags`, which has to be an integer greater than zero. The third input, `options`, specifies the options; it can be omitted if default options are used. The autoregressive model is specified with a constant. One can remove the constant using the `noconstant` option, before the estimation command is launched, e.g.

```
options.nocostant = 1;
[BVAR]            = bvar(y,lags,options)
```

In the case of flat prior, one can allow also a time trend in the model, if that is of interest. To do so, the first column of the data set should be the vector `[1:T]'`. Time trends are not

[3]For more details on the mapping between the MA coefficients (Ψ's) and VAR parameters (Φ's) see Appendix section A.1.1.

allowed in the case of Bayesian estimation with Minnesota prior as they are taken care of by means of prior restrictions. We discourage users to use time trends also with conjugate priors as they may interfere with the prior restrictions imposed.

The output of the function, `BVAR`, is a structure with several fields and sub-fields containing the estimation results:

- `BVAR.Phi_draws` is a $(\mathtt{n}\times\mathtt{lags}+1)\times\mathtt{n}\times\mathtt{K}$ matrix containing `K` draws from the posterior distribution of the autoregressive parameters, Φ_j, assuming a Jeffrey prior. For each draw `k`, the first $\mathtt{n}\times\mathtt{n}$ submatrix of `BVAR.Phi_draws(:,:,k)` contains the autoregressive matrix of order one and subsequent submatrices contain higher order lags. The last row of `BVAR.Phi_draws(:,:,k)` contains the constant, Φ_0, if it assumed. By default, `K` is set to 5,000. It can be changed by setting `options.K` to any positive desired number.
- `BVAR.Sigma_draws` is a $\mathtt{n}\times\mathtt{n}\times\mathtt{K}$ matrix containing `K` draws from the posterior distribution of the innovation covariance matrix, Σ.
- `BVAR.e_draws` is a $(\mathtt{T}-\mathtt{lags})\times\mathtt{n}\times\mathtt{K}$ matrix containing `K` draws of the innovations.
- `BVAR.logmlike` contains the log marginal likelihood $-$ a measure of fit.
- `BVAR.InfoCrit` contains four information criteria, `AIC`, `SIC`, `HQIC` and `BIC`, where

$$
\begin{aligned}
AIC(p) &= \ln\left(\frac{T-(np+1)}{T}\det\Sigma_e\right)+\frac{2}{T}pn^2\\
SIC(p) &= \ln\left(\frac{T-(np+1)}{T}\det\Sigma_e\right)+\frac{2}{\ln T}pn^2\\
HQ(p) &= \ln\left(\frac{T-(np+1)}{T}\det\Sigma_e\right)+\frac{2\ln\ln T}{T}pn^2\\
BIC(p) &= \ln\left(\frac{T-(np+1)}{T}\det\Sigma_e\right)+\frac{\ln T}{T}pn^2
\end{aligned}
$$

- `BVAR.ir_draws` is a four dimensional object (i.e. $\mathtt{n}\times\mathtt{hor}\times\mathtt{n}\times\mathtt{K}$ matrix) that collects the impulse response functions with recursive identification. The first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses obtained with particular draw from the posterior distribution of the VAR parameter. See Section 4 for more details on the implementation of different identification schemes.
- `BVAR.forecasts.no_shocks` is a three dimensional object (i.e. $\mathtt{fhor}\times\mathtt{n}\times\mathtt{K}$ matrix) that collects the forecasts assuming zero shocks in the future. The first dimension corresponds the horizon, the second to the variable, and the third to the draw from the posterior distribution of the parameters. See Section 7 for more details on forecasting options.

Example 1 *We use monthly Euro Area data (*`Data`*) and consider a VAR with the following variables: the log of industrial production index (*`IPI`*), the log of HICP (*`HICP`*), the one-year Euribor (*`Euribor1Y`*) and the log of M3 (*`M3`*) from 2000m1 2015m8. We optimize the lag length of this VAR*

```
load Data
y       = [IPI HICP Euribor1Y M3]; % collect the variables used in the VAR
maxlag  = 10; % maximum  number  of  lags allowed
opt.K   = 1;  % generate only 1 draw from the posterior
for nlags=1:maxlag
    BVAR = bvar(y, nlags, options);
    disp(['Number of lags ' num2str(nlags)])
    disp(BVAR.InfoCrit)
end
```

3.1 Minnesota Prior

The Minnesota prior is a special conjugate prior specification. In the original formulation, the prior for the VAR parameters is normally distributed while the covariance of the innovations is treated as fixed. More details on the features of the Minnesota prior are in Canova (2007, Chapter 5). The Minnesota prior is convenient, since under the assumption that the prior hyper-parameters are fixed (or estimable), it is possible to derive analytically not only the format of the posterior distribution but also its moments.

The toolbox considers the version of the Minnesota prior discussed in Sims and Zha (1998), Del Negro and Schorfheide (2011) or Villemot and Pfeifer (2017), where the covariance matrix of the innovations is random, and there are various layers of shrinkage implemented via dummy observations. The direction of the shrinkage is controlled by a handful of *hyper-parameters* that regulate: (i) the prior tightness for the autoregressive coefficients of order one; (ii) the prior tightness for the autoregressive coefficients of lags higher than one; (iii) the weight on its own-persistence; (iv) the weight on the co-persistence of the data; and (v) the weight for the priors for the variance and covariance matrix of innovations. (iii) is typically labeled *sum-of-coefficients* prior and reflects the belief that if a variable has been stable at its initial level, it will tend to stay at that level, regardless of the value of other variables. (iv) is typically labeled *co-persistence* prior dummy observations and reflects the belief that when the data is stable at its initial levels, it will tend to stay at that level.[4]

The hyper-parameters controlling the restrictions on the prior second moments are computed using pre-sample information. By default, the size of pre-sample coincides with the number of lags. This feature can be changed by setting `options.presample` to an integer (default `options.presample = 0`).

To use the Minnesota prior, the user needs to activate it with the `option.priors.name` command. To do so, type in the MATLAB command window (or put in your script) the following:

[4]Section A.5 describes how to draw from the posterior distribution with Minnesota priors. Some of the functions used in the Minnesota prior are adapted from Chris Sims's VAR codes, see http://sims.princeton.edu/yftp/VARtools. The default values for the hyperparameters also follow Sims.

```
options.priors.name = 'Minnesota';
BVAR                = bvar(y,lags,options);
```

As in the Jeffrey prior case, draws from the posterior distribution are collected in `BVAR.Phi_draws` and `BVAR.Sigma_draws`, which have the same dimensions as before. The hyper-parameters controlling the Minnesota prior are:

1. `options.minn_prior_tau` is a scalar controlling the overall tightness (default value 3). The larger is this value, the tighter is prior.
2. `options.minn_prior_decay` is scalar controlling the prior tightness on the lags greater than one (default value 0.5). The larger is this value, the faster is the lag decay.
3. `options.minn_prior_lambda` is scalar controlling the Sum-of-Coefficient prior (default value 5)
4. `options.minn_prior_mu` is scalar controlling the co-persistence prior (default value 2)
5. `options.minn_prior_omega` is scalar controlling the prior on the covariance matrix (default value 2)

The larger the value for `options.minn_prior_tau` the tighter is the overall prior on the reduced form parameters, inducing more shrinkage on the prior first moment. The hyper-parameter values can be cherry-picked by the user. This can be done by setting their values in the `options` command. When one of the hyper-parameter is set to a specific value - for example, `options.minn_prior_tau=10` - the Minnesota prior is automatically activated. Prior hyper-parameters can also be chosen to maximize the marginal data density as described in Canova (2007) or Giannone, Lenza and Primiceri (2015). If you prefer this option, use the command:

```
options.max_minn_hyper = 1;
BVAR                   = bvar(y,lags,options);
```

Note that hyper-parameter maximization does not work with missing data. By default, optimization is performed unconstrained and Chris Sims optimizer `cminwel.m` is used (this is `options.max_compute=3`).[5] The following options can be set in the maximization step:

1. `options.index_est` is a raw vector that selects the parameters to be optimized. Default, `options.index_est=1:5` (i.e. all five hyper-parameters are optimized).
2. `options.lb` and `options.ub` set the lower and upper bounds for the optimization. Both are row array vectors of the same size of `options.index_est`.
3. `options.max_compute` is a scalar selecting the maximization routine to be employed:
    - `options.max_compute = 1` uses the MATLAB `fminunc.m`
    - `options.max_compute = 2` uses the MATLAB `fmincon.m`

[5] The optimizer can be downloaded at https://github.com/naffe15/BVAR_.

- `options.max_compute = 3` uses the Chris Sims's `cminwel.m`
- `options.max_compute = 7` uses the MATLAB `fminsearch.m`

The first three are Newton, derivative-based algorithms; the latter is a direct search (symplex) method based on function comparisons. While typically slower, the latter method is useful in situations where derivatives are not well behaved.

The ordering of the hyper-parameters must follow the order of the list, i.e. 1 is `tau`; 5 is `omega`. Mixed options can be used as described in the following example.

Example 2 (Mixed Options Maximization) *Using the same data in Example 1, we estimate the VAR parameters using a Minnesota prior. We set the overall prior tightness to 10 (instead of the default value of 3) and optimally estimate the sum-of-coefficient and the co-persistence prior hyper-parameters. Set the bounds for the optimized values of the two hyper-parameters to be zero and 20, and use the MATLAB's Symplex routine to perform the optimization. The following set of instructions can be used:*

```
options.max_minn_hyper  = 1;
options.minn_prior_tau  = 10;      % set tau
options.index_est       = [3 4];   % define the hyper-parameters over which to maximize
options.lb              = [0 0];   % sets the lower bounds
options.ub              = [20 20]; % sets the upper bounds
options.max_compute     = 7;       % optimization  by Matlab Simplex
BVAR                    = bvar(y,lags,options);
```

Once the maximum is found, the posterior distribution is computed using the optimal hyper-parameter values. If the maximization is unsuccessful (converge criteria are not satisfied or Hessian not positive definite are the typical reasons), the posterior distribution is computed using default values. In such case a warning is printed in the command window.

Often, the maximization is not successful in the first few runs. Issues making routines fail could be diverse, ranging from 'poor' initial conditions to the mode too close to the boundaries of parameter space. It is difficult to give general advice on how to perform the optimization, as the solution tends to be case specific and to depend on the data used. It is advisable to start maximizing the log-marginal likelihood using few hyper-parameters, say, the overall prior tightness, and add one or more parameters at time starting the optimization from the values obtained from the previous maximization.

Since draws from the posterior distribution are not needed, rather than using the function `bvar` it is possible to use the function `bvar_max_hyper`, which is faster.

Example 3 (Maximize the Marginal Likelihood Only) *We use the same data in Example 1. Given default values of the other hyper-parameters, we find the optimal overall tightness hyper-parameter. The following commands instruct the toolbox to optimize over the prior tightness hyper-parameter:*

```
% setting the default values for the hyperparameters
hyperpara(1)     = 3;    % tau
hyperpara(2)     = 0.5;    % decay
hyperpara(3)     = 5;    % lambda
hyperpara(4)     = 2;    % mu
hyperpara(5)     = 2;    % omega
% setting the options
options.index_est     = 1:1;     % hyper-parameter over which maximize
options.max_compute     = 2;       % maximize  using Matlab fmincon function
options.lb             = [0.05]; % Lower bound
[postmode,logmlike,HH] = bvar_max_hyper(hyperpara,y,lags,options);
```

In the output of the `bvar_max_hyper` *routine* `postmode` *has the mode,* `logmlike` *the log of the marginal likelihood at the mode, and* `HH` *the Hessian evaluated at the mode. The output of the optimization process is reported below.*

```
** Initial Hyperpara Values and Log Density **

tau = 3
decay = 0.5
lambda = 5
mu = 2
omega = 2
log density = 3670.4

** Posterior Mode: (Minimization of -Log Density)  **

tau = 4.5256
decay = 0.5
lambda = 5
mu = 2
omega = 2
log density = 3675.5
```

Thus, the log marginal data density increases by 5 point when `tau` *moves from 3 to 4.52.*

Example 4 (continuation) *Using the mode value of the maximization in Example 3, jointly maximize over the overall tightness, the decay and the sum-of-coefficient hyper-parameters. The bounds for the parameters to be maximized are [0.05, 50]. The commands are as follows*

```
hyperpara(1)              = postmode(1); % use as starting value previous mode
options.index_est         = 1:3; % set hyper-parameters over which maximize
```

```
options.lb              = [0.05 0.05 0.05]; % Lower bounds
options.ub              = [50 50 50];       % Upper bounds
[postmode,log_dnsty,HH] = bvar_max_hyper(hyperpara,y,lags,options);
```

The output of the optimization process is as follows:

```
** Initial Hyperpara Values and Log Density **

tau = 4.5256
decay = 0.5
lambda = 5
mu = 2
omega = 2
log density = 3675.5

** Posterior Mode: (Minimization of -Log Density)  **

tau = 1.1632
decay = 1.8373
lambda = 1.1963
mu = 2
omega = 2
log density = 3688.7
```

The log marginal data density increases by 13 points with the new optimal values of `tau`, `decay` *and* `lambda`. *If needed the other two parameters can now be added in a third stage of the maximization process.*

3.2 Conjugate Priors

Posterior distributions can also be constructed using a Multivariate Normal-Inverse Wishart conjugate prior. The command to active this prior is

```
options.priors.name = 'Conjugate';
```

With a conjugate setting, the prior for the autoregressive parameters is centered at zero with a diagonal covariance matrix of 10 and the prior for the covariance matrix of the residual is inverse Wishart (see appendix A.11 for details) with a unitary diagonal matrix as scale and `n+1` degrees of freedom. If the user does not like these value, she can customize the hyper-parameters as follows:

- `options.priors.Phi.mean` is a $(\texttt{n} \times \texttt{lags} + 1) \times \texttt{n}$ matrix containing the prior means for the autoregressive parameters.

- `options.priors.Phi.cov` is a $(\texttt{n} \times \texttt{lags} + 1) \times (\texttt{n} \times \texttt{lags} + 1)$ matrix containing the prior covariance for the autoregressive parameters.
- `options.priors.Sigma.scale` is a $(\texttt{n} \times \texttt{n})$ matrix containing the prior scale of the covariance of the residuals.
- `options.priors.Sigma.df` is a scalar defining the prior degrees of freedom.

One can set these options one at the time or jointly. Given this prior, draws are generated from the posterior distribution of the parameters using a Gibbs sampler algorithm; see section A.4 for more details. As before, draws from the posterior distribution are collected in `BVAR.Phi_draws` and `BVAR.Sigma_draws`, which have the same dimensions as before.

4 Computing Impulse Response Functions

Vector Autoregressive (VAR) models have become very popular to study how certain policy and non-policy disturbances are dynamically transmitted to macroeconomic aggregates. To study the transmission of interesting disturbances, we need to impose ‘some structure‘ on the VAR. Once this is done, one can trace out the causal effects of structural disturbances.

Compared to theoretical business cycle models, VARs are less restricted. For this reason VAR-based analyses are less likely to be distorted by incorrect specification of theoretical equilibrium conditions. The cost of using a limited number of restrictions is that structural disturbances may be confused, see Canova and Pappa (2011) or Wolf (forthcoming). When the VAR includes all the theoretical relevant variables, it can be regarded as an unrestricted representation of the linear solution of a structural macroeconomic models. Under the conditions described in Canova and Ferroni (2019), the dynamic transmission of disturbances in the structural model can be mapped into structural VAR impulse responses.

Assuming normally distributed innovations[6], if the solution of the structural model has linear (state space) representation, the reduced form VAR innovations u_t can be mapped into structural disturbances ν_t by means of

$$u_t = \Omega\ \nu_t = \Omega_0\ Q\ \nu_t$$

where $E(\nu_t \nu_t') = I$, $\Omega\Omega' = \Sigma$, Ω_0 is the Cholesky decompostion of Σ and Q is an orthonormal rotation such that $Q'Q = QQ' = I_n$. To recover ν_t, we need to impose restrictions on Ω. This is because Σ only contains $n(n+1)/2$ estimated elements, while Ω has n^2 elements. Several identification schemes have been proposed in the literature, see Ramey (2016) for a survey. Once Ω is measured, the dynamics of the endogenous variables in response to the disturbances can be obtained by means of impulse response functions.

To characterize the posterior distribution of the responses we use the following algorithm

Algorithm 1

Given the posterior distribution $p(\vartheta|Y)$, do the following steps $m = 1, ..., M$ times

[6] For identification non Gaussian VARMA models see Gourieroux, Monfort and Renne (Forthcoming)

1. Draw $\vartheta^{(m)} = vec(\Phi_0^{(m)}, ..., \Phi_p^{(m)}, \Sigma^{(m)})$ from $p(\vartheta|Y)$.

2. Construct the impact matrix $\Omega^{(m)}$ and generate a candidate impulse response function using

$$x_{t+h}^{(m)} = \left(F^{(m)}\right)^h \; G \; \Omega^{(m)} \; \nu_t$$

for $h = 0, ..., H$ and F and G are the matrices of the VAR companion form.

The algorithm will generate M trajectories, $\left\{y_{t+h:T+H}^{(m)}\right\}_{m=1}^{M}$; these trajectories can then be used to obtain numerical approximations to moments, quantiles, or to the empirical distribution of the impulse response functions. Note that the algorithm works for just-identified, sign identified or instrument identified models. For over-identified models, see Canova and Perez-Forero (2015).

The BVAR toolkit allows one to construct responses when disturbances are identified with a number of schemes and assuming any of the prior descried in the previous section; the baseline function to generate the impulse response function is

```
[BVAR] = bvar(y, lags, options)
```

where `options` allows to introduce the desired identification scheme. By default, responses are computed for 24 periods and constructed assuming a one standard deviation impulse; both can be changed using the option command as follows

- `options.hor` controls the horizon for the IRF; default value is 24.
- `options.irf_1STD` activates a 1% increase in the shock; recall that, by default, responses are computed using one standard deviation impulse in the shock.

At the end of this section we present examples illustrating the use of several identification schemes.

4.1 Recursive or Cholesky identification

The Cholesky decomposition assumes a recursive ordering among the disturbances so that the disturbance to the first variable is predetermined relative to the disturbances to the variables following in the order. For example, in a trivariate VAR with GDP, prices and interest rate in that order, the Cholesky decomposition implies that disturbances to GDP have an immediate impact on all variables; disturbances to prices have an immediate impact on prices and the interest rate, but not on GDP; and disturbances to the interest rate do not have a contemporaneous impact on GDP or prices. These restrictions imply that the matrix Ω is lower triangular:

$$\begin{pmatrix} u_t^y \\ u_t^p \\ u_t^r \end{pmatrix} = \begin{pmatrix} \omega_{1,1} & 0 & 0 \\ \omega_{2,1} & \omega_{2,2} & 0 \\ \omega_{3,1} & \omega_{3,2} & \omega_{3,3} \end{pmatrix} \begin{pmatrix} \nu_t^y \\ \nu_t^p \\ \nu_t^r \end{pmatrix}$$

The Cholesky decomposition is unique, meaning that there is only one matrix lower triangular such that $\Omega\Omega' = \Sigma$, up to a permutation of the order of the variables. It is easy to check the order condition for identification (there are $n(n+1)/2$ restrictions) is satisfied and thus the scheme produces a point-identified system. Following Rubio-Ramírez, Waggoner and Zha (2010), one can easily check that a Cholesky system is both locally and globally identified.

By default, the function `[BVAR] = bvar(y,lags)` computes impulse response functions based on a recursive identification scheme, in the order the variable are listed in the vector y. Changing the order or the variables in y will alter the response of the endogenous variables only if the innovations in different variables are highly correlated. When the covariance of u_t is almost diagonal the ordering of the variables is irrelevant. If there are doubts about the ordering to be used, one can use the estimated covariance matrix with is produced by `bvar` and check which innovations are highly correlated (values above 0.3-0.4 should deserve some thinking.) The estimation routine produces `BVAR.Sigma_ols`, a $\mathtt{n} \times \mathtt{n}$ matrix containing estimate of the variance covariance matrix of the shocks, Σ. The correlation matrix can be obtained taking that matrix and scaling the off-diagonal elements by the $\sqrt{\sigma_{ii}\sigma_{jj}}$.

Cholesky responses are stored in

```
BVAR.ir_draws;
```

`BVAR.ir_draws` is a four dimensional object (i.e. a $\mathtt{n} \times \mathtt{hor} \times \mathtt{n} \times \mathtt{K}$ matrix), where the first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses obtained with particular draw from the posterior distribution of the VAR parameter. Note that in a Cholesky decomposition the elements of Ω are treated as fixed when drawing from the posterior of VAR parameters. Such an approach is valid as long as there is a one-to-one mapping between the u's and the ν's.

4.2 Long Run identification

The implementation of long run restrictions in the toolkit follows Gali (1999). We assume that the first variable in the `y` lists is specified in log difference and that the first disturbance has long run effects on the first variable.[7] To activate identification via the long run restrictions, the user needs to set the following option:

```
options.long_run_irf = 1;
```

Responses obtained with this identification scheme are stored in

```
BVAR.irlr_draws;
```

where `BVAR.irlr_draws` is a four dimensional object (i.e. a $\mathtt{n} \times \mathtt{hor} \times \mathtt{n} \times \mathtt{K}$ matrix), where the first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses obtained with particular draw from the posterior distribution of the parameters.

[7]See Appendix A.7 for details.

Typically, the variables other than the first one are assumed to be stationary and specified in (log) level. More than one long run shock, as in Fisher (2006), can be estimated by introducing more variables in first difference.

4.3 Sign and Magnitude Restrictions identification

A popular way to impose loose restrictions and identify a set rather than a point is to employ sign constraints, see Canova and de Nicolo (2003) and Uhlig (2005). Sign restrictions are flexible: they can be imposed on one or more variables, at one or multiple horizons, to identify one or more shocks. We generate candidate draws using the algorithm discussed in Rubio-Ramírez et al. (2010).

Sign restriction are activated by setting the following option:

```
options.signs{1} = 'y(a,b,c)>0';
```

where `a`, `b`, and `c`, are integer. The syntax means that shock `c` has a positive impact on variable `a` at horizon `b`. There is no limit to the number of restrictions on can impose. Note however that the larger the set of restrictions, the harder it is to find a rotation satisfying them, and thus the longer the time it takes to construct impulse response functions. When a large number of restrictions are imposed, one or no rotation may be found.

Accepted responses and accepted rotations are stored in, respectively

```
BVAR.irsign_draws          BVAR.Omegas;
```

`BVAR.irsign_draws` is a four dimensional object (a $\mathtt{n} \times \mathtt{hor} \times \mathtt{n} \times \mathtt{K}$ matrix), where the first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses obtained with a particular draw from the posterior distribution of the parameters and one accepted rotation. `BVAR.Omegas` is a three dimensional object (a $\mathtt{n} \times \mathtt{n} \times \mathtt{K}$ matrix) where the first dimension corresponds to the variables, the second to the disturbances, and the third to a particular draw from the posterior distribution.

The syntax above is quite general and allows to impose a variety of restrictions. For example, one could impose magnitude restrictions, where a certain shock has some particular sign on a variable k at horizon h and it is bounded (above, below, or both). For example, one could use:

```
options.signs{1} = 'y(a,b,c)>m';
options.signs{2} = 'y(a,b,c)<M';
```

where `m` and `M` are positive numbers with `M > m` and as before `a`, `b`, and `c`, are integer corresponding to the variable, the horizon and the shock of interest.

One could also impose magnitude restrictions on the maximum of the cumulative response of a specific variable as follows

```
options.signs{1} = 'max(cumsum(y(a,:,c),2))<m';
```

where `cumsum(X,DIM)` computes the cumulative sum of `X` along the dimension specified by `DIM`. Or impose the constraint that the product of the responses of two variables is larger than a threshold, i.e.

```
options.signs{1} = 'y(a_1,b,c)*y(a_2,b,c)>m';
```

In general, the toolkit allows one can build restrictions using any mathematical operation on the elements of the three dimensional object `y`.

Needless to say, that one must justify where these bounds come from. In some cases, elasticity restrictions or capacity constraints can be used to impose these bounds. In others, cross sectional differences can be used. In many situations, however, one has to think hard why such bounds may be reasonably imposed. Again, the larger the set of restrictions, the harder it is to find a rotation satisfying them, and the longer becomes the time it takes to construct impulse response functions.

4.4 Narrative Restrictions

The sets of responses identified with sign restrictions tend to be large, making inference sometimes difficult. One could reduce the length of the intervals by imposing additional sign restrictions or restricting other shocks, even if they are not direct interest, see Canova and Paustian (2011). Alternatively, one may also have a-priori information on the sign or the relative magnitude of shocks (see Antolin-Diaz and Rubio-Ramírez (2016)). For example, using the Volker period (say 1979-1982), a contractionary monetary policy disturbance need to satisfy certain sign restrictions and, in addition, be positive during these years. To see how these restrictions can be imposed, write the the VAR model in its structural VMA format:

$$
\begin{aligned}
y_t &= u_t + \Psi_1 u_{t-1} + ... + \Psi_t u_1 + \overline{\Psi}_t \\
&= \varphi_0 \nu_t + \varphi_1 \nu_{t-1} + ... + \varphi_t \nu_1 + \overline{\Psi}_t
\end{aligned}
$$

where $\varphi_0 = \Omega$, $\varphi_j = \Psi_j \Omega$ for $j = 1, ..., t$ and $\overline{\Psi}_t$ has deterministic variables. Sign restrictions are imposed on the elements of φ_j for some j, $j = 0, ..., t$ for a candidate Ω to be accepted. Narrative restriction are instead imposed on $\nu_t, t = [\bar{T}, \bar{T}+K]$, where K is chosen by the user for a candidate Ω to be accepted. In other words, we now have two types of sign restrictions: on φ_j and on ν_t. If the second type of restrictions are binding, the set of accepted impulse responses should be smaller at each j.

Narrative restrictions can be jointly activated with sign restrictions by setting, in addition to the sign option, the following option:

```
options.narrative{1} = 'v(tau,n)>0';
```

where `tau` is a vector of integers and `n` is an integer. The syntax means that shock `n` is positive on the time periods `tau`. Note that if no signs restrictions are specified, a warning message is printed. There is no limit to the number of narrative restrictions on can impose. However, the larger the set of restriction, the harder is to find a rotation jointly satisfying sign and narrative constraints.

Accepted responses and accepted rotations are stored in, respectively,

`BVAR.irnarrsign_draws` `BVAR.Omegan`;

`BVAR.irnarrsign_draws` is a four dimensional object (i.e. a `n` $\times$ `hor` $\times$ `n` $\times$ `K` matrix), where the first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses obtained with a particular draw from the posterior distribution of the parameters and one accepted rotation. `BVAR.Omegan` is a three dimensional object (i.e. a `n` $\times$ `n` $\times$ `K` matrix) where the first dimension corresponds to the variables, the second to the disturbances, and the third to a particular draw from the posterior distribution.

Narrative restriction can be imposed also on the cumulative value of the structural shock of interest using the following option

```
options.narrative{1} = 'sum(v(tau0:tau1),n)>0';
```

where `tau0` and `tau1` are integers and `n` is an integer. The syntax means that the sum of the shock `n` between periods `tau0` and `tau1` is positive.

4.5 Mixed identification strategy

One can also identify shocks using a mix of the primitive restrictions we have discussed so far. For example, one can identify shocks using zeros and sign restrictions, see Arias, Rubio-Ramírez and Waggoner (2018). We generate candidates using the algorithm discussed in Binning (2013). Let `j`, `k`, `k_1` and `k_2` be integers. Below is an example of a mix of restrictions to identify different shocks.

1. Sign restrictions on impact:

   ```
   options.zero_signs{1} = 'y(j,k)=+1';
   ```

   This restriction implies that shock `k` has a positive effect on the `j`-th variable **on impact**. `'y(j,k)=-1'` imposes a negative sign on impact on variable `j`.

2. Zero restrictions on impact (short run):

   ```
   options.zero_signs{2} = 'ys(j,k_1)=0';
   ```

   This restriction implies that shock `k_1` has a zero impact effect on the `j`-th variable.

3. Zero long run restrictions:

   ```
   options.zero_signs{3} = 'yr(j,k_2)=0';
   ```

This restriction implies that shock `k_2` has a zero long run effect on the `j`-th variable. Notice that short and long run restrictions are distinguished by the addition of `r` or `s` to the y vector.

Accepted responses and accepted rotations are stored in, respectively

```
BVAR.irzerosign_draws          BVAR.Omegaz;
```

`BVAR.irzerosign_draws` is a four dimensional object (i.e. a $\mathtt{n} \times \mathtt{hor} \times \mathtt{n} \times \mathtt{K}$ matrix), where the first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses produced by a particular draw from the posterior distribution. `BVAR.Omegaz` is a three dimensional object (i.e. a $\mathtt{n} \times \mathtt{n} \times \mathtt{K}$ matrix), where the first dimension corresponds to the variables, the second to the shocks of interest, and the third to a particular draw from the posterior distribution.

4.6 External Instrumental Variable or Proxy identification

The proxy, external, or instrumental variable approach, was pioneered by Stock and Watson (2012) and Mertens and Ravn (2013). The basic idea is that the disturbance of interest is identified by the predicted value in the regression of a reduced form VAR innovation on the instrument. For the approach to provide valid inference we need the instrument to be relevant (i.e. correlated with the disturbance of interest) and exogenous (uncorrelated with the other disturbances). The regression allows one to identify a column of the rotation matrix, and thus recover transmission mechanism of one disturbance.[8]

The toolkit considers the case where one wishes to identify **one** structural disturbance and has at least one instrument, stored in the array column vector called `instrument`, is available. The toolkit computes proxy/IV responses when the following options is activated

```
options.proxy = instrument;
```

The length of the times series for the instrument cannot be longer than the length of the time series of the innovation, i.e. `T - lags`. By default it is assumed that the last observation for the instrument coincides with the last observation of the VAR (this need not be the case for the first observation). When this is not the case, the user needs to inform the toolkit about the number periods that separate the last observation of the instrument and the last observation of the innovation as follows

```
options.proxy_end = Periods
```

where `Periods` corresponds to the number time periods between the last observation of the instrument and the last observation in the innovation.

[8]See appendix A.6 for details.

Impulse responses identified via external instruments are stored in

```
BVAR.irproxy_draws;
```

`BVAR.irproxy_draws` is a four dimensional object (i.e. a `n` $\times$ `hor` $\times$ `n` $\times$ `K` matrix), where the first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses produced by a particular draw from the posterior distribution.

Multiple proxy variables are allowed to identify one structural shocks. However, the codes do not allow to identify multiple structural shocks with multiple proxy variables. Clearly, if one is interested in identifying two disturbances, she can repeat the IV exercise twice separately for each disturbance.

By convention, the structural shock of interest is ordered **first**. Responses to the IV identified disturbance can be retrieved using

```
BVAR.irproxy_draws(:,:,1,:);
```

4.7 Additional Identification Schemes

While the toolbox covers a number of standard identification schemes, it does not exhaust all of them, e.g. Angeletos, Collard and Dellas (2018). Yet, it can be used to generate draws for the posterior distribution for the impulse responses with additional methods as long as the desired identification scheme makes use of an orthonormal matrix `Q`, i.e. `QQ'=Q'Q=I`. Two functions are useful for this purpose; the first generates orthonormal matrices of size `n`

```
Q = generateQ(n)
```

where `n` is an integer. The second function computes responses, given Q:

```
y = iresponse(Phi, Sigma, hor, Q)
```

where `y` is a `n` $\times$ `h` $\times$ `n` array containing the responses; and the first dimension corresponds to the variable, the second to the horizon and the third to the shock. The inputs are, respectively, the matrix of the autoregressive parameters, the variance covariance matrix, the maximum horizon of the responses and the orthonormal (rotation) matrix. If the latter is omitted, the identity matrix is used.

To illustrate how these commands could be employed, consider the case where one wants to identify the shock maximizing the forecast error variance decomposition of a particular variable at a specific horizon, see section 4.9. Given a draw (`Phi,Sigma`) of the reduced form VAR parameters, one can use the following function, which finds the orthonormal rotation maximizing the forecast error variance decomposition,

```
Qbar  = max_fevd(i, h, j, Phi, Sigma, Kappa)
```

where i stands for the variable, h stands for the horizon and j corresponds to the shock. The function max_fevd generates Kappa draws of orthonormal matrices calling the function generateQ.m; if omitted its value is set to 1000. For each of these draws, the forecast error variance decomposition (FEVD) at horizon h is computed (see section 4.9 and appendix A.8); FEVD is a $\mathtt{n} \times \mathtt{n}$ matrix where the (i,j) element corresponds to the share of variance of variable i explained by shock j. Then, Qbar corresponds to the rotation matrix where the latter is maximized.

Then one can use the following loop to construct the distribution of the responses of interest. Assume that the shock of interest is the first and we want it to explain the largest portion of the variance of the first variable in four periods

```
j = 1; % shock
i = 1; % variable
h = 4; % periods
for k = 1 : BVAR.ndraws % iterate on posterior draws
      Phi      = BVAR.Phi_draws(:,:,k);
      Sigma    = BVAR.Sigma_draws(:,:,k);
      Qbar     = max_fevd(i, h, j, Phi, Sigma);
      % compute impulse responses with a particular posterior draw
      % and the rotation matrix Qbar.
      [ir]    = iresponse(Phi, Sigma, hor, Qbar);
      % store the IRF of interest
      BVAR.irQ_draws(:,:,:,k) = ir;
end
```

4.8 Plotting impulse responses

One can obtain numerical approximations to the distribution of the impulse response functions. Typically, numerical approximations are constructed sorting the IRF and plotting the percentiles of interests. The function that performs that task is

```
plot_irfs_(irfs_to_plot)
```

irfs_to_plot is a fourth dimensional array where the first dimension corresponds to the variables, the second to the horizons, the third to the shocks, and the fourth to the responses produced by a particular draw from the posterior distribution. For example, when considering a recursive identification, type irfs_to_plot = BVAR.ir_draws. The output is a sequence of figures reporting the response functions, one figure per shock; by default, each figure consists of the upper bound integers of $\sqrt{\mathtt{n}} \times \sqrt{\mathtt{n}}$ subplots reporting the IRF of all variables included in the VAR.

In each panel, a black solid line represents the median response and the gray area corresponds to the 68% high probability density (HPD) set. Different settings are possible. If one wants to change the default options type:

`plot_irfs_(irfs_to_plot, options)`

where the available options are:

- `options.varnames` is a cell string containing the variable names for the subplot titles. The cell must be of the same length as the first dimension of `irfs_to_plot`.
- `options.shocksnames` is a cell string containing the shock names for the figure title. The cell must be of the same length as the third dimension of `irfs_to_plot`.
- `options.normz` is a scalar that normalizes the responses to a 100% increase when the impulse set to 1 (as opposed to a one standard deviation).
- `options.conf_sig` is a number between 0 and 1 indicating the size of HPD set to be plotted; the default is 0.68.
- `options.conf_sig_2` is a number between 0 and 1 indicating the size of the **second** HPD set to be plotted (if more than one set are required).
- `options.nplots` is a 1×2 array indicating the structure of the subplots.
- `options.saveas_strng` a string array indicating the name of the plot.
- `options.saveas_dir` a string array indicating the directory where to save the plot; the figures are not saved if `options.saveas_dir` or `options.saveas_strng` are not specified.
- `options.add_irfs` allows to add additional IRF to compare responses.

4.9 Forecast Error Variance and Historical Decomposition

Often researchers are interested in whether certain shocks explain the second moment of a VAR variable (forecast error variance decomposition, FEVD) or its level (historical decomposition). Since structural shocks are orthogonal, we can rewrite the vector of structural shocks as follows

$$\nu_t = \nu_t^1 + ... + \nu_t^n \quad \nu_t^k \sim N(0, J_k) \quad \forall k = 1, .., n$$

where each ν_t^k is a $n \times 1$ vector of zeros except for k-th position Similarly, J_k is a $n \times n$ a matrix with zeros everywhere and with one in the (k, k) position. With this setup it is easy to explain what the two decompositions do.

Historical decomposition

The historical decomposition is typically employed to measure the contribution of certain structural shocks to the deviation of the level variables from a forecasted path. That is, we want to answer the question: how much of the deviation of inflation from the predicted path is due to oil price shocks? To perform an historical decomposition use the structural VMA form:

$$y_t = u_t + \Psi_1 u_{t-1} + ... + \Psi_t u_1 + \overline{\Psi}_t$$
$$= \varphi_0 v_t + \varphi_1 v_{t-1} + ... + \varphi_t v_1 + \overline{\Psi}_t$$

where, as previously described, $\varphi_0 = \Omega$ and $\varphi_j = \Psi_j \Omega$ for $j = 1, ..., t$ are functions of $(\Phi_1, ..., \Phi_p)$[9] and Ω, and $\overline{\Psi}_t$ is the pure deterministic component. Given the linearity of the model and orthogonal structure of v_t we can decompose the mean adjusted level of the VAR observable $(y_t - \overline{\Psi}_t)$ as the sum of the contribution of each shock, i.e.

$$y_t - \overline{\Psi}_t = \sum_{j=0}^{t} \varphi_j v_{t-j}^1 + ... + \sum_{j=0}^{t} \varphi_j v_{t-j}^n$$

An historical decomposition in the toolkit is performed with function

```
[yDecomp,v] = histdecomp(BVAR,opts);
```

where `yDecomp` is a `T` $\times$ `n` $\times$ `n+1` array where the first dimension corresponds to the time, the second to the variable, and the third to shock with the `n+1` element indicating the initial condition $\overline{\Psi}_t$. `v` is the `T` $\times$ `n` array of structural innovations.

The first input of the `histdecomp.m` function is `BVAR`, the output of the `bvar.m` function. By default the function uses the mean value of the parameters over posterior draws and a recursive identification scheme. Both settings can be changed using `opts`. `opts.Omega` is a `n` $\times$ `n` array that defines the rotation matrix to be used; `opts.median=1` uses the median of the posterior draws and `opts.draw = integer` selects instead a specific draw from posterior distribution.

Remark 5 (Historical Decomposition with non stationary data) *When data is non-stationary (e.g. real quantity and price indices), the deterministic component is overwhelmingly dominant. Thus, a historical decomposition of growth rates or of the variables in deviation from the deterministic component tends to be more informative.*

The function that plots the historical decomposition is

```
plot_shcks_dcmp_(yDecomp, BVAR, optnsplt)
```

where `optnsplt` is optional controls a number of options which are:

- `optnsplt.plotvar_` a cell array with the variables that one wishes to consider for the decomposition. Without this option, the decomposition of all variables in the VAR is plotted. By default, variables' names are `Var1`, `Var2`, ..., `VarN` and are stored in `BVAR.varname`. So if one is only interested in the historical shock decomposition of the variable ordered second in the VAR, she should type `optnsplt.plotvar_= {'Var2'}`. Notice: names in `BVAR.varnames` can be changed to any string array while preserving the order of the variable entering the VAR specification.

[9] See section A.1.1 in the appendix.

- `optnsplt.snames_` is a cell array with the shocks that one wishes to consider for the decomposition. Shocks are called `Shck1`, `Shck2`, ..., `ShckN`. Dy default, the decomposition consider each shock individually. One could also group some of them. For example, when $N = 4$ and one is interested in the contribution of the first shocks and the remaining three, one can type

```
optnsplt.snames_ = { {'Shck1'} ; {'Shck2', 'Shck3', 'Shck4'} };
```

  In such case, the plot consists of two bars, one that identifies the contribution of `Shck1` only and one that groups together the contribution of `Shck2, Shck3, Shck4`.

- `optnsplt.stag_` is a cell array with the shocks names or shock aggregation names. By default, shocks are called `Shck1`, `Shck2`, ..., `ShckN`. When `optnsplt.snames_` is declared, one must declare also the names for the shock groups. As in the previous example, when $N = 4$ and one is interested in the contribution of the first shocks and the remaining three, one can type for example

```
optnsplt.stag_  = { {'Shck1'} ; {'Shcks: 2+3+4'} };
```

- `optnsplt.time` is a `T-lags` $\times$ 1 vector that defines the time vector. By default, `optnsplt.time = [1 : 1 : T-lags]`.
- Other options available are `options.saveas_dir` and `options.save_strng` and are described in 8.2.

Forecast error variance decomposition

The contribution of the structural shocks to the volatility of the variables is usually measured by means of the forecast error variance decomposition. The variance decomposition can be used to answer the question: how much of the fluctuations in inflation are due to supply shocks? Let $\widehat{y}_{t+h} = E_t(y_{y+h})$ be the h-step head forecast of the VAR model, which is the conditional expectation of y_t given time t information; we have that

$$\begin{aligned}\widehat{y}_{t+h} &= E_t(\varphi_0 v_{t+h} + \varphi_1 v_{t+h-1} + ... + \varphi_h v_t + ... + \varphi_{t+h} v_1 + \overline{\Psi}_{t+h}) \\ &= \varphi_h v_t + ... + \varphi_{t+h} v_1 + \overline{\Psi}_{t+h}\end{aligned}$$

We define the forecast error at horizon h as

$$\begin{aligned}e_{t+h} = y_{y+h} - \widehat{y}_{t+h} \\ = \varphi_0 v_{t+h} + \varphi_1 v_{t+h-1} + ... + \varphi_{h-1} v_{t+1}\end{aligned}$$

The forecast error variance and its decomposition is given by[10]

$$E(e_{t+h}e'_{t+h}) = \left[\sum_{\ell=0}^{h-1} \varphi_\ell J_1 \varphi'_\ell\right] + ... + \left[\sum_{\ell=0}^{h-1} \varphi_\ell J_n \varphi'_\ell\right]$$

[10]See appendix A.8 for details

where $\left[\sum_{\ell=0}^{h-1} \varphi_\ell J_1 \varphi_\ell'\right]$ represents the contribution of shock 1 to the h-step ahead forecast error variance decomposition.

The horizon `h` FEVD is called with the function

```
FEVD    = fevd(h,Phi,Sigma,Omega);
```

where `FEVD` is a $\mathtt{n} \times \mathtt{n}$ array containing the fraction of FEV explained by each shock; the first dimension corresponds to the variable, and the second to the shock. The inputs are respectively the horizon, the autoregressive parameters, the variance covariance matrix, and the rotation matrix. If the latter is omitted, the function considers the recursive identification.

4.10 The transmission of structural disturbances: some examples

In this section we use the monthly US data of Gertler and Karadi (2015) (henceforth GK) and to study the transmission of a number of structural shocks using various identification schemes. The `DataGK.mat` contains the log of industrial production index (`logip`), the log of CPI (`logcpi`), the one-year government bond rate (`gs1`) and the excess bond premium (`ebp`) which are the four series used by the Gertler and Karadi (2015) paper. The sample runs from 1979m7 to 2012m6.

Example 6 (Responses to Monetary Policy shocks, recursive identification, default commands) *We estimate and plot the responses to a monetary policy impulse (a shock to the one-year government bond rate,* `gs1`*) identified using a recursive identification scheme and uninformative priors. We use the same lag length as in GK. We proceed in steps. First, we prepare the data*

```
load DataGK                  % load the data in the workspace
y = [logip logcpi gs1 ebp]; % combine the data in a Tx4 matrix y
```

Since flat priors are set by default, no specific instruction is needed. By default, the horizon of the responses is set to 24. We modify it to have the same horizon as in GK and then run the model.

```
lags        = 12;                     % number of lags
options.hor = 48;                     % response horizon
bvar1       = bvar(y,lags,options);  % run the BVAR
```

Recall that responses are stored in `bvar1.ir_draws`*. We sort them along the* ***fourth*** *dimension which corresponds to the draws and then select the response of interest.*

```
irf_sort        = sort(bvar1.ir_draws,4);                % sort the responses
```

In this case, the relevant shock is an orthogonalized shock to `gs1`*, which is ordered third. Suppose that the variable of interest is the interest rate, which is also ordered third. We plot the median response and some the percentiles of interest. The commands are as follows:*

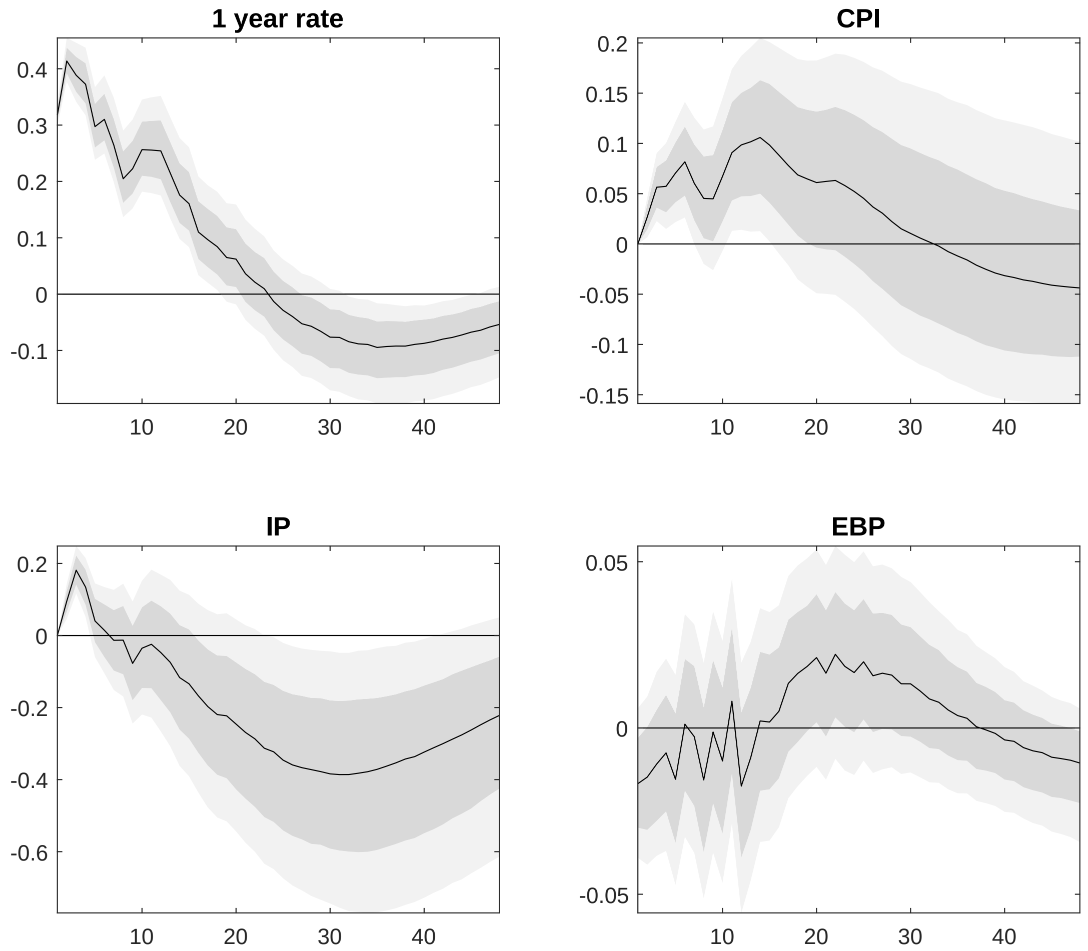
```
% select variable and shock of interest
irf_interest    = squeeze(irf_sort(3,:,3,:));
irf_m           = irf_interest(:,0.5*bvar1.ndraws);   % median response
irf_u           = irf_interest(:,0.05*bvar1.ndraws);  % 5th percentile
irf_l           = irf_interest(:,0.95*bvar1.ndraws);  % 95th percentile
% plots
plot(irf_m,'k'); hold on; plot(irf_u,'k-.'); plot(irf_l,'k-.'); axis tight
```

Example 7 (Plot responses with build-in function) *Use the toolbox function* `plot_irfs_` *to plot the responses. We display 68% and 90% credible sets. We use a different variable ordering for the plot, i.e. the one-year government bond rate (*`gs1`*) is ordered first followed by log of industrial production index, the log of CPI and the excess bond premium. We use titles and save the responses.*

```
% Define the IRFs
% index of the shocks of interest (shock to gs1)
indx_sho              = [3];
% Order of the variables in the VAR
% 1.  logip; 2. logcpi; 3.gs1; 4. ebp
% Change the order of the variables for plotting
% 1. gs1; 2. logcpi; 3. logip; 4. ebp
indx_var              = [3, 2, 1, 4];
% IRFs to PLOT
irfs_to_plot           = bvar1.ir_draws(indx_var,:,indx_sho,:);

% Customize the plot
% names of  the variables  to  be  plotted.
options.varnames      = {'1 year rate','CPI','IP','EBP'};
% name of the directory where the figure is saved
options.saveas_dir    = './irfs_plt';
% name of the figure to be saved
options.saveas_strng  = 'cholesky';
% name of the shock
options.shocksnames   = {'MP'};
% additional 90% HPD set
options.conf_sig_2    = 0.9;
% finally, the plotting command
plot_irfs_(irfs_to_plot,options)
```

Figure 1 displays the results.

Figure 1: Responses to a one standard deviation Cholesky orthogonalized innovation in MP. Light (dark) gray bands represent 68 (90)% credible sets.



Example 8 (Responses to monetary Policy shocks, sign identification) *Assume the same data and the same setting of Exercise 6. We identify monetary policy disturbances assuming, without loss of generality, that the first shock is the monetary policy disturbance and that a monetary tightening increases the one-year government bond rate and depresses the CPI price level. We assume that these restrictions hold on impact and for the next two months. These constraints can be coded as follows:*

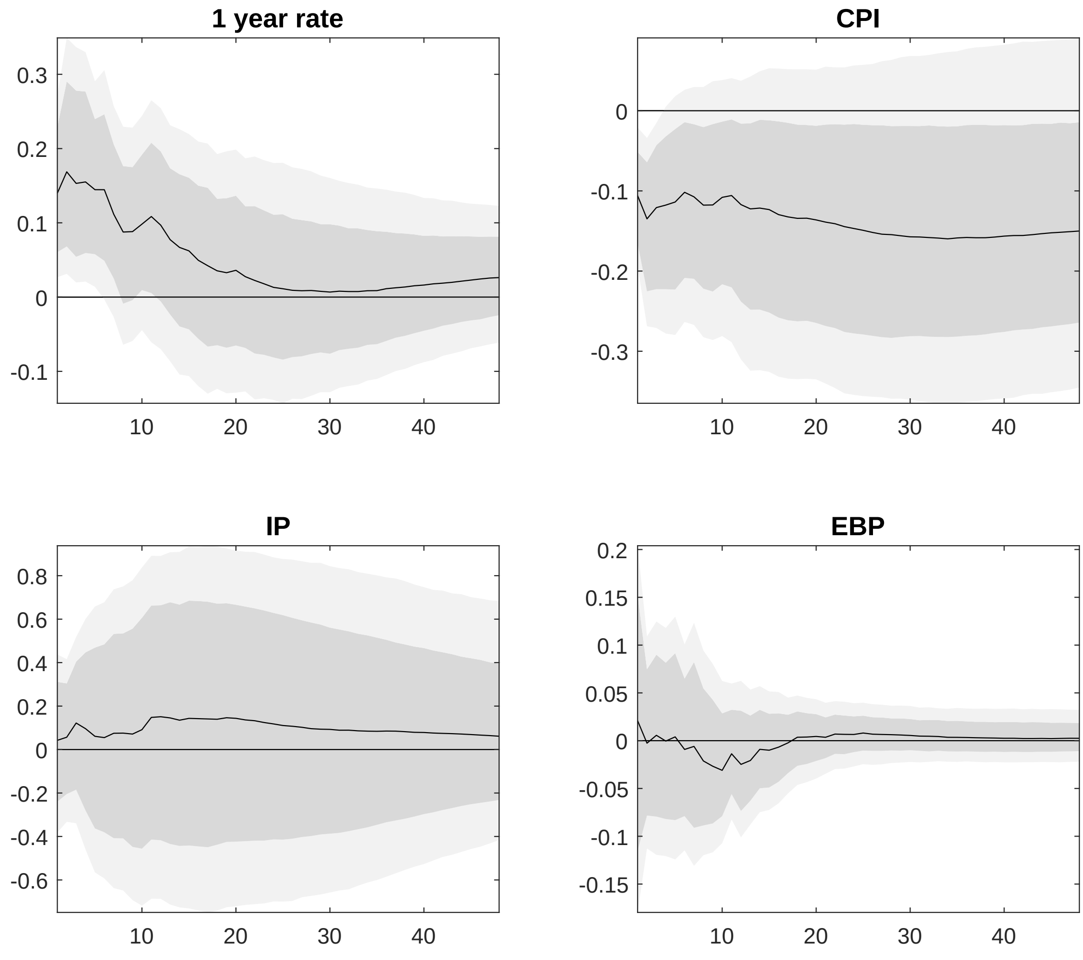
```
% specify the restrictions
options.signs{1} = 'y(3,1:3,1)>0'; % 1Y rate up in periods 1 to 3
options.signs{2} = 'y(2,1:3,1)<0'; % CPI down in periods 1 to 3
% run the BVAR
bvar2             = bvar(y,lags,options);
```

Plot the responses with the same settings as in Example 7.

```
% Define the IRFs
% index of the shocks of interest (shock to gs1)
indx_sho        = [1];
% IRFs to plot (indx_var: 1. gs1; 2. logcpi; 3. logip; 4. ebp )
irfs_to_plot     = bvar2.irsign_draws(indx_var,:,indx_sho,:);
% Customize the plot
% name of the figure to be saved
options.saveas_strng  = 'signs';
% name of the shock
options.shocksnames   = {'MP tightening'}; %
plot_irfs_(irfs_to_plot,options)
```

Figure 3 reports the results.

Figure 2: Responses to monetary policy tightening, sign restrictions. Light (dark) gray bands 68 (90)% credible sets.

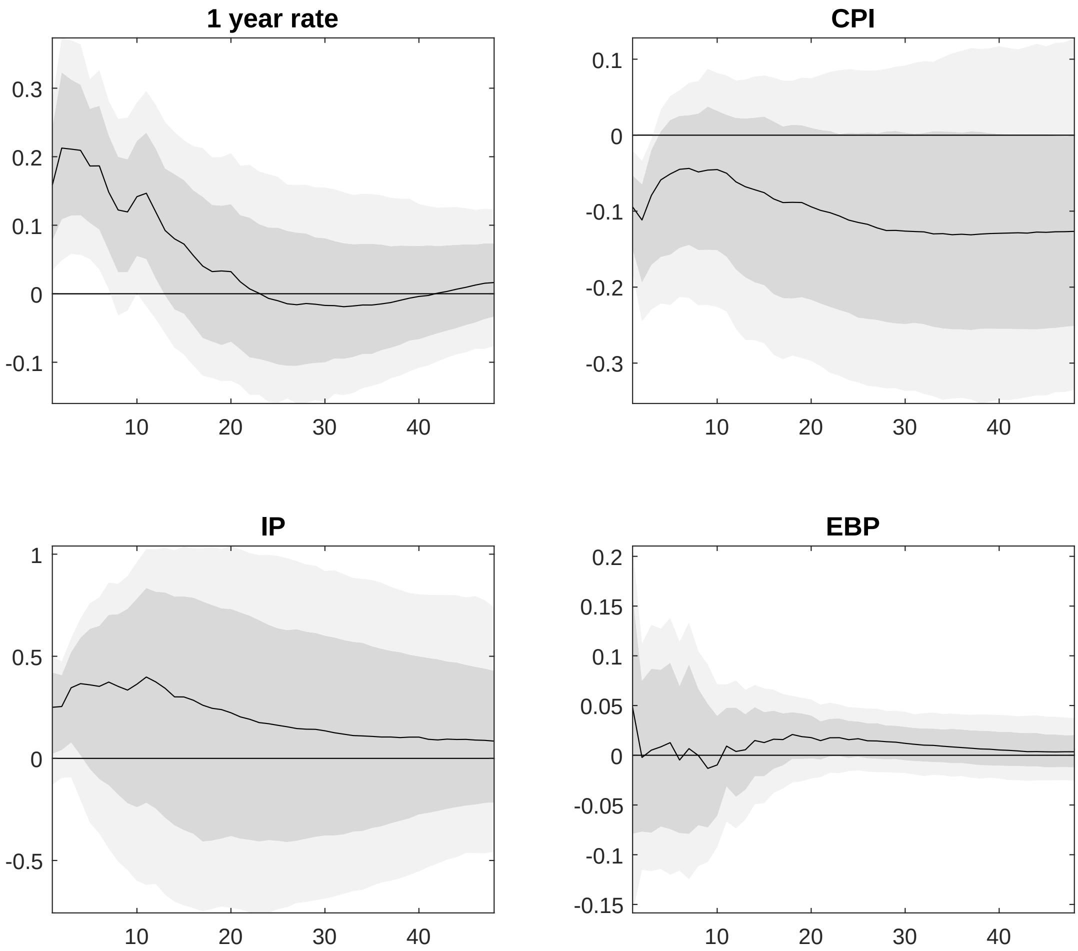


Example 9 (Responses to Monetary Policy shocks, sign and narrative restrictions) *Assume the same data and the same setting of Exercise 8. Add additional narrative restrictions on the MP shocks. We use the information during the Volker mandate and consider four large tightening episodes in the Romer and Romer database. In particular, we assume that in September, October and November 1980 and May 1981, monetary policy shocks were positive. Since the sample starts in 1979m7 and there are 12 lags, the first reduced form innovation is for 1980m7. Therefore, the innovation for 1980m09 = #3, 1980m10 = #4, 1980m11 = #5, and the innovation for 1981m05 = #11. We specify the restrictions as follows*

```
% sign restrictions
options.signs{1} = 'y(3,1:3,1)>0'; % 1Y rate up in period 1 to 3
options.signs{2} = 'y(2,1:3,1)<0'; % CPI down in period 1 to 3
% narrative restrictions
options.narrative{1} = 'v([3:5],1)>0';
options.narrative{2} = 'v([11],1)>0';

% run the BVAR
bvar3            = bvar(y,lags,options);

% Define the IRFs of Interest
% index of the shocks of interest (shock to gs1)
indx_sho         = [1];
% IRFs to plot (we change the order of the variables for the plot)
irfs_to_plot     = bvar3.irnarrsign_draws(indx_var,:,indx_sho,:);

% Customize the plot
% name of the directory where the figure is saved
options.saveas_dir    = './irfs_plt';
% name of the figure to be saved
options.saveas_strng  = 'signsnarrative';
% name of the shock
options.shocksnames   = {'MP tightening'}; %
plot_irfs_(irfs_to_plot,options)
```

Figure 3 reports the results. Bands narrow a bit relative to the case of signs restriction. Adding more narrative restrictions, increase the computation burden as it is more difficult to find valid rotations.

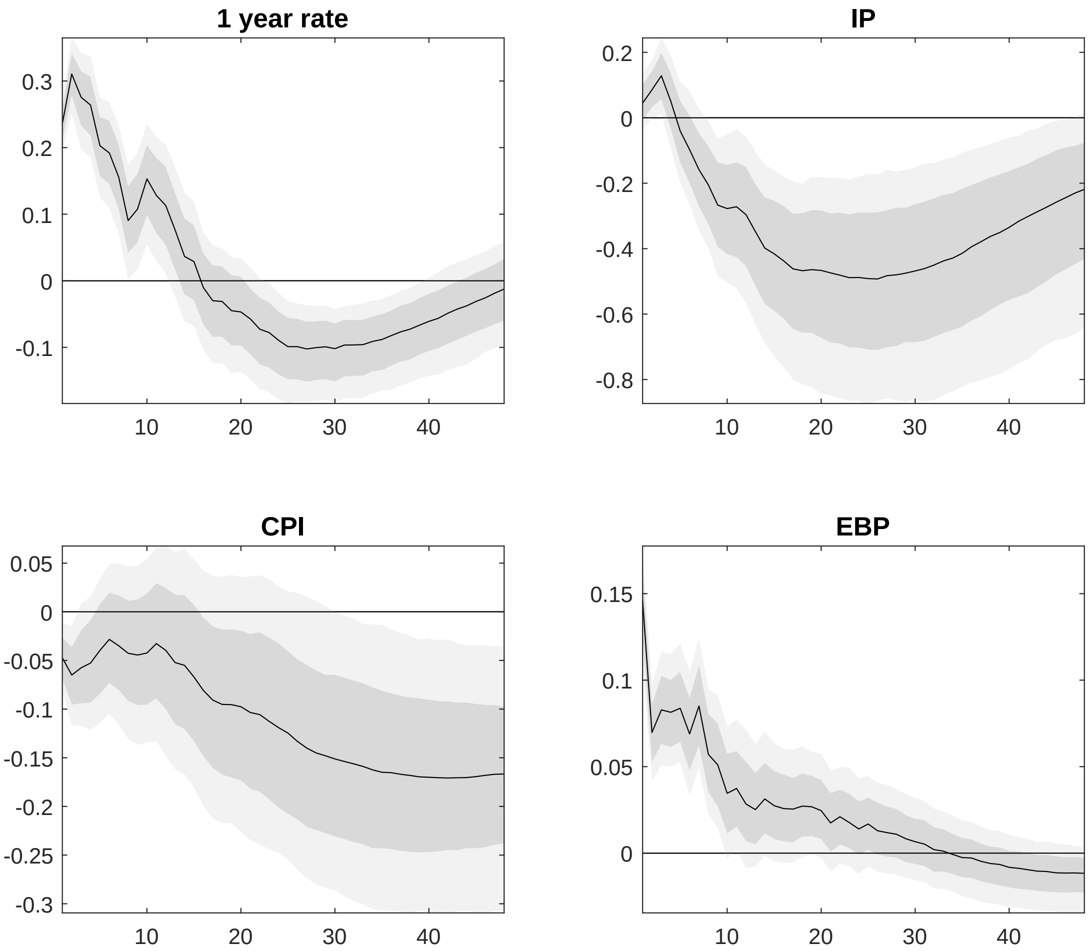
Figure 3: Responses to monetary policy tightening, sign and narrative restrictions. Light (dark) gray bands 68 (90)% credible sets.



Example 10 (Responses to monetary policy shocks, identified with instruments) *We instrument the one-year government bond innovation with the GK preferred monetary policy surprise proxy. The ordering of the variables in* `y` *is the following: one-year government bond rate(*`gs1`*), log of industrial production index (*`logip`*), log of CPI (*`logcpi`*) and the excess bond premium (*`ebp`*). The following commands allow to construct the responses with external instruments. The responses are plotted in figure 4.*

```
% define the dataset for the identification of MP shock with IV
y = [gs1 logip logcpi ebp];

% load the instruments
[numi,~,~] = xlsread('factor_data.csv','factor_data');
% use the same instrument as GK
options.proxy  = numi(:,4);

% Since both instruments and data end in 2012m6 - no more instructions on the IV
% run the BVAR
bvar6          = bvar(y,lags,options);

% Define  the  IRFs  of  interest
% shock index
indx_sho      =  [1];
% Keep the same order of variables as in the estimation
% 1. gs1; 2. logip; 2. logcpi; 4. ebp;
indx_var      = [1:4];
irfs_to_plot = bvar6.irproxy_draws(indx_var,:,indx_sho,:);

% Customize the plot
% names for the variables  in  the plots
options.varnames      = {'1 year rate','IP','CPI','EBP'};
% name of the figure to be saved
options.saveas_strng    = 'IV';
% name of the shock
options.shocksnames     = {'MP'};
plot_irfs_(irfs_to_plot,options)
```

Figure 4: Responses to monetary policy tightening, instrumental variables identification. Gray bands 68 (90)% credible sets.



Example 11 (Responses to AD, AS, and MP shocks identified with a mixture of restrictions) *Assume the same data and the same setting of Exercise 6. We jointly identify an aggregate supply, an aggregate demand, and a monetary policy disturbance using the following restrictions: (1) the aggregate demand disturbance increases the industrial production, prices (CPI) and the one year government rate; (2) the aggregate supply disturbance increases industrial production and decreases prices, (3) the monetary policy disturbance does not have contemporaneous effects on prices and quantities, but positively affect the excess bond premium. These sets of restrictions, can be implemented with the following instructions:*

```
% specify the restrictions
% 1) ad = aggregate demand disturbance [sign restrictions]
options.zeros_signs{1} = 'y(1,1)=1;';
options.zeros_signs{end+1} = 'y(2,1)=1;'; % 'end+1' adds a new cell
options.zeros_signs{end+1} = 'y(3,1)=1;';
% 2) as = aggregate supply shock [sign restrictions]
options.zeros_signs{end+1} = 'y(1,2)=1;';
options.zeros_signs{end+1} = 'y(2,2)=-1;';
% 3.1) mp = monetary policy shock  [zero restrictions]
options.zeros_signs{end+1} = 'ys(1,3)= 0;';
options.zeros_signs{end+1} = 'ys(2,3)= 0;';
% 3.2) mp = rate and bond premium go up [sign restrictions]
options.zeros_signs{end+1} = 'y(3,3)=1;';
options.zeros_signs{end+1} = 'y(4,3)=1;';
% run the BVAR
bvar4             = bvar(y,lags,options);
```

Plot the responses to the three shocks with the same settings as in Example 7.

```
% Define  the  IRFs of  interest
indx_sho     = [1:3];
irfs_to_plot = bvar4.irzerosign_draws(indx_var,:,indx_sho,:);
% Customize the plots
options.saveas_strng     = 'zerossigns';
options.shocksnames      = {'ADshock','ASshock','MPshock'}; %
plot_irfs_(irfs_to_plot,options)
```

Figures 5, 6 and 7 have the responses to a one standard deviation orthogonalized innovation in AD, AS and MP.

Figure 5: Responses to a one standard deviation innovation in AD identified with zero and sign restrictions. Light (dark) gray bands 68 (90)% credible sets.

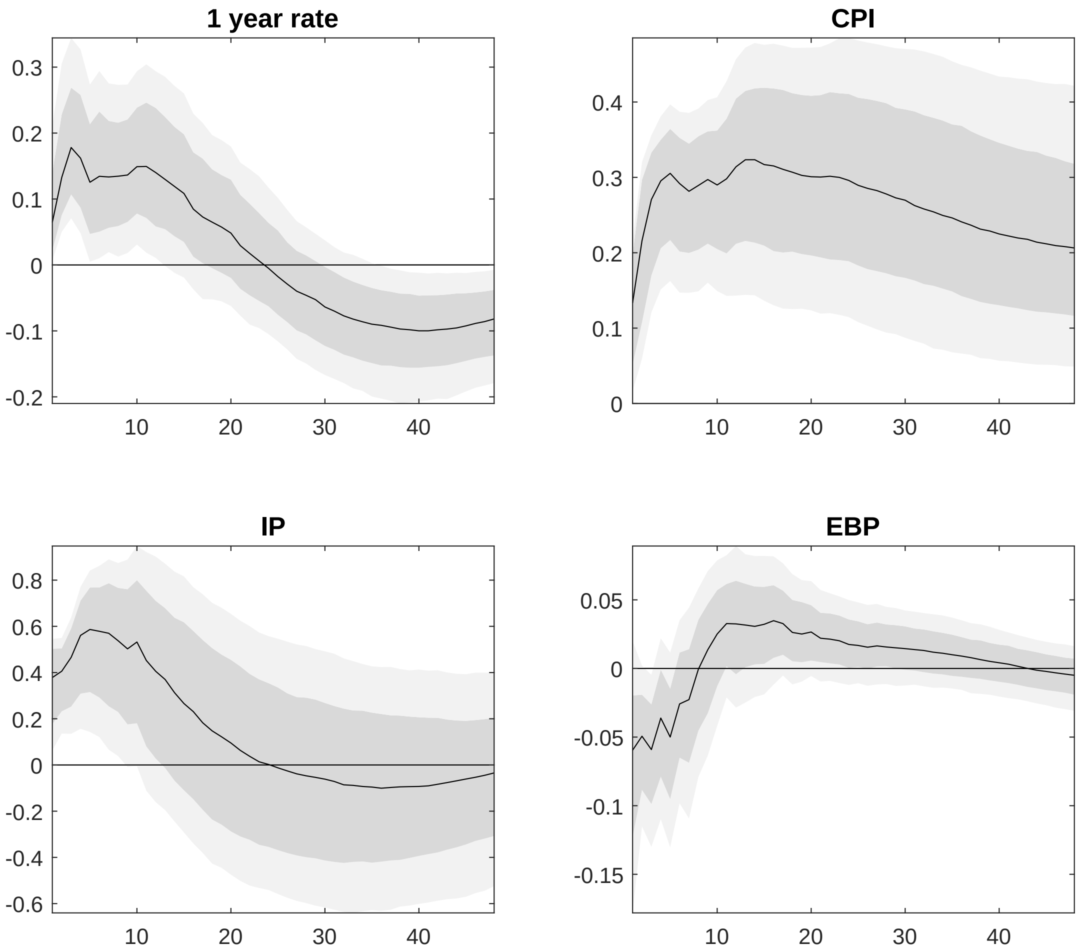


Figure 6: Responses to a one standard deviation innovation in AS identified with zero and sign restrictions. Light (dark) gray bands 68 (90)% credible sets.

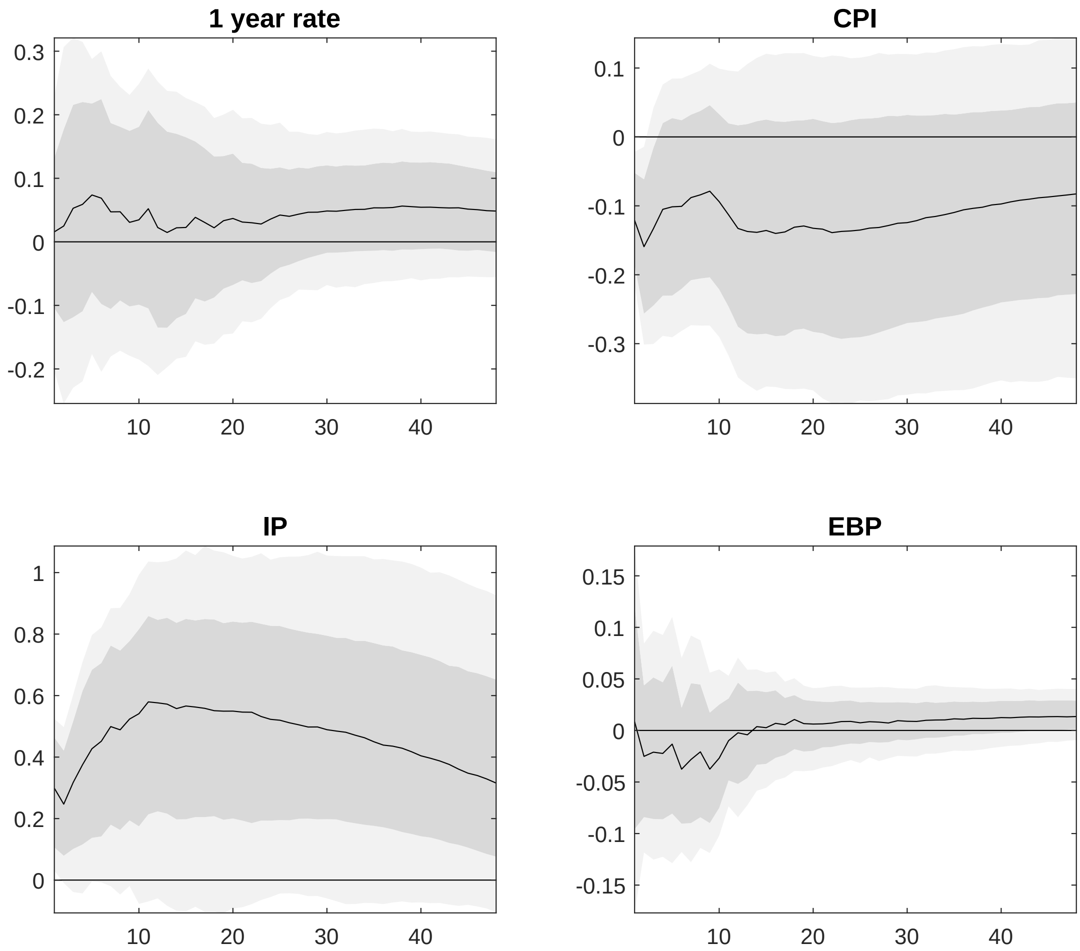
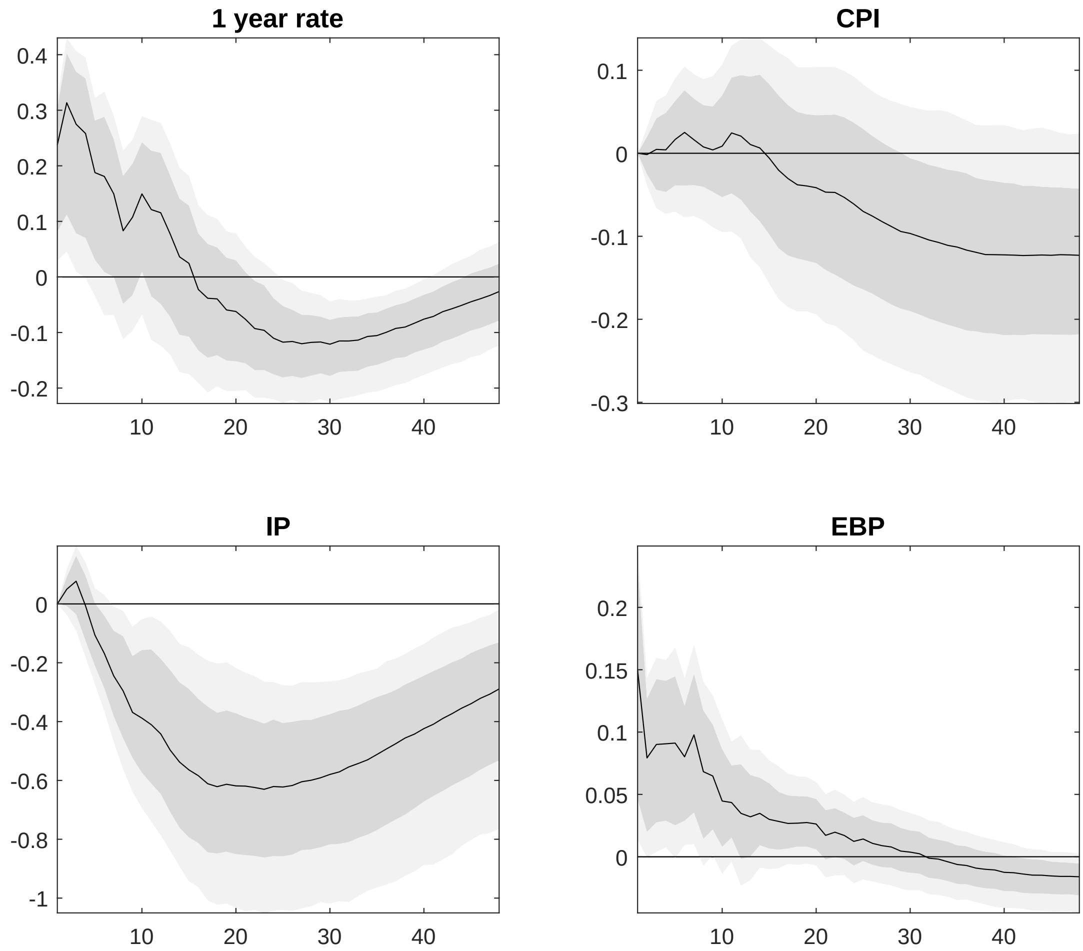


Figure 7: Responses to a one standard deviation innovation in MP identified with zero and sign restrictions. Light (dark) gray bands 68 (90)% credible sets.



Example 12 (Responses to technology shocks, identified via long run restrictions) *Assume the same data and the same setting of Exercise 6. Identify a technology shock assuming that it is the only shocks having a long run impact on industrial production. We specify log of industrial production in first difference. For housekeeping we also remove previous identification settings. These commands are detailed below:*

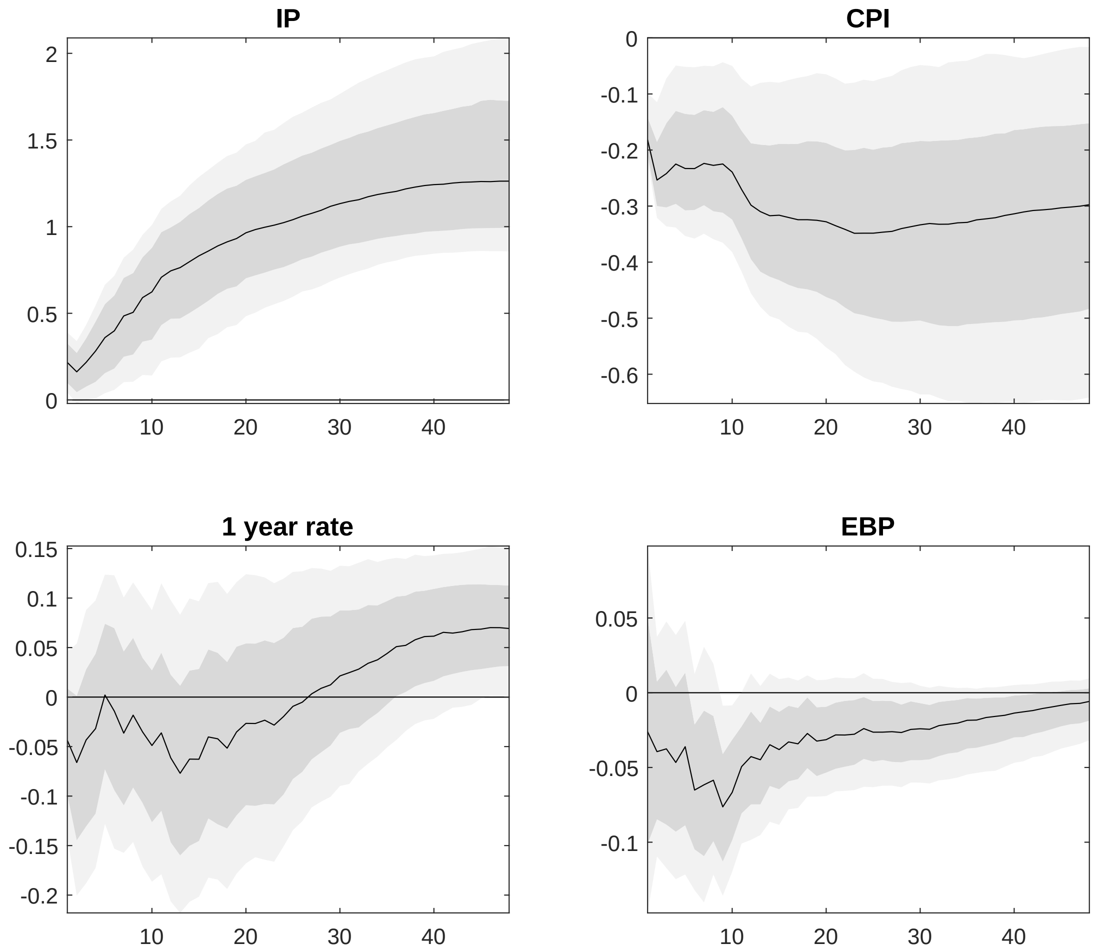
```
% Housekeeping: remove previous identification settings
options = rmfield(options,'signs');
options = rmfield(options,'zeros_signs');
options = rmfield(options,'saveas_strng');
options = rmfield(options,'shocksnames');
% define the data for the identification of LR shock (remove first obs)
y       = [diff(logip) logcpi(2:end) gs1(2:end) ebp(2:end)];

% Activate the LR identification and run the BVAR
options.long_run_irf    = 1;
bvar5                   = bvar(y,lags,options);
```

We plot the responses with the same settings as in Example 7.

```
% Define  the  IRFs  of  interest
% shock index
indx_sho     =  [1];
% Define the order of the variables for the plot
% 1. D(logip); 2. logcpi; 3.  gs1; 4. ebp;
indx_var     = [1:4];
irfs_to_plot  = bvar5.irlr_draws(indx_var,:,indx_sho,:);
% Transform D(logip) back to logip
cirfs_to_plot = cumsum(bvar5.irlr_draws(indx_var,:,indx_sho,:),2);
irfs_to_plot(1,:,:,:) = cirfs_to_plot(1,:,:,:);

% Customize the plot
% variable names for the plots
options.varnames      = {'IP','CPI','1 year rate','EBP'};
% name of the figure to be saved
options.saveas_strng    = 'LR';
% name of the shock
options.shocksnames     = {'Technology'}; %
plot_irfs_(irfs_to_plot,options)
```

Figure 8 reports the responses.

Figure 8: Responses to a technology shock identified with long run restrictions. Light (dark) gray bands 68 (90)% credible sets.

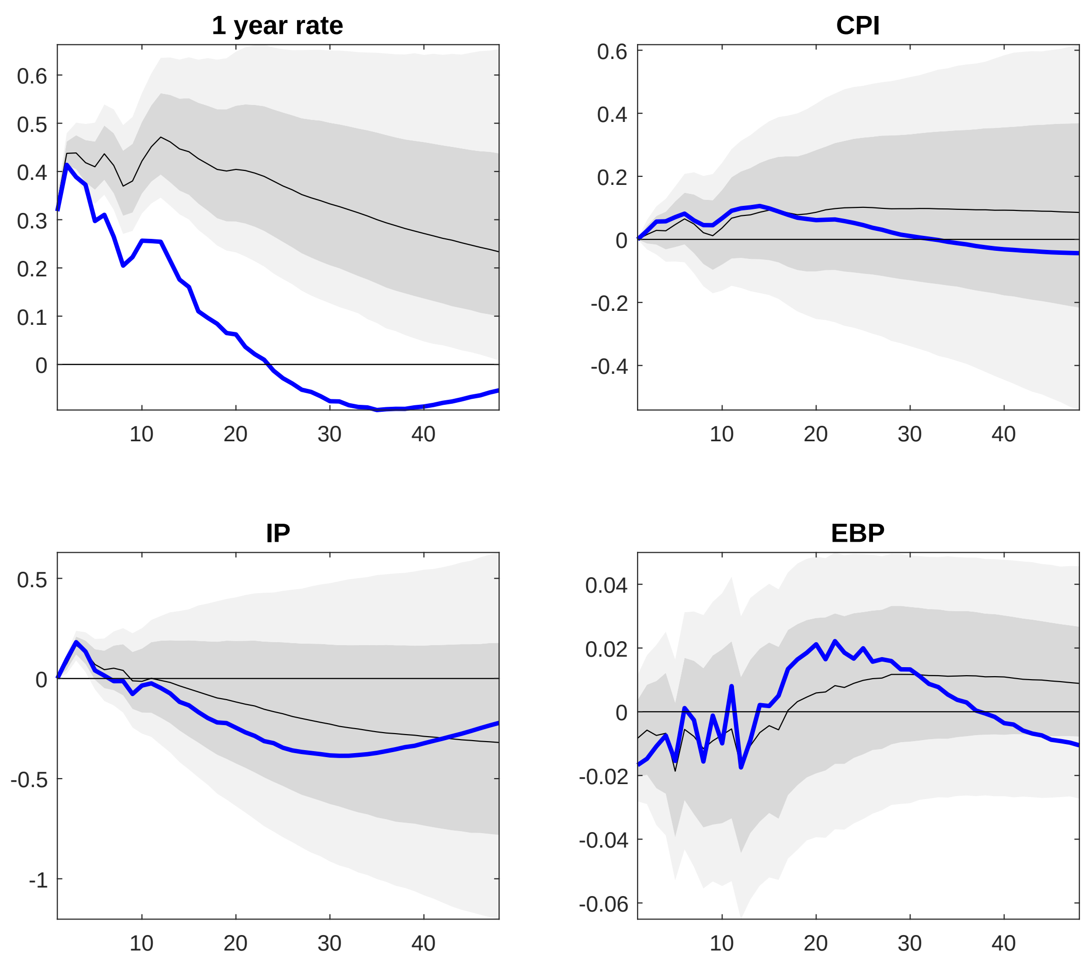


Example 13 (Comparing response to Monetary Policy shocks with flat and Minnesota priors) *Assume the same settings in Example 6. Consider now a different prior; in particular, assume a Minnesota prior with a relatively loose overall shrinkage, e.g. $\tau = 0.5$, and use the first two years of the sample to calibrate the first and second moment of the prior. Compute the IRF obtained and contrast them with the median IRF using uninformative priors. The following commands allow to compare the responses under the two settings*

```
lags                      = 12;
options.presmaple         = 12; % presample + lags to initialize the minnesota prior.
options.prior.name        = 'minnesota';
options.minn_prior_tau = 0.5;
bvar6                     = bvar(y,lags,options);

% Define the IRFs of interest
% index of the shocks of interest (shock to gs1)
indx_sho                 = [3];
% Change the order of the variables for the plot
% 1. gs1; 2. logcpi; 3. logip; 4. ebp
indx_var                 = [3, 2, 1, 4];
% plot IRFs
irfs_to_plot              = bvar6.ir_draws(indx_var,:,indx_sho,:);

% Customize the plot
% variables names for the plot
options.varnames         = {'1 year rate','CPI','IP','EBP'};
% name of the directory where the figure is saved
options.saveas_dir       = './irfs_plt';
% name of the figure to be saved
options.saveas_strng     = 'BayesianCholesky';
% name of the shock
options.shocksnames      = {'MP'};
% additional 90% HPD set
options.conf_sig_2       = 0.9;
% add the Cholesky IRF with flat prior
options.add_irfs         = squeeze(median(bvar1.ir_draws(indx_var,:,indx_sho,:),4));
% finally, the plotting command
plot_irfs_(irfs_to_plot,options)
```

The responses are reported in figure 9. Responses computed with the Minnesota prior tend the more persistent than those obtained with uninformative (flat) priors.

Figure 9: Responses to a monetary policy tightening, Minnesota prior, recursive identification. Gray bands 68 (90)% credible sets. The blue line reports the median response with flat priors.



Example 14 (FEVD: The role of Monetary Policy shocks) *Assume the same settings of exercise 6. We compute the share of the forecast error variance at the two years horizon explained by monetary policy shocks. The commands are*

```
% consider the mean of the posterior distribution
Phi   = mean(bvar1.Phi_draws,3);
Sigma = mean(bvar1.Sigma_draws,3);
% index of the shock of interest (shock to gs1)
indx_sho               = [3];
% 2 years ahead forecsat error
hh      = 8;
FEVD    = fevd(hh,Phi,Sigma);
```

Recall that `FEVD` *is a* 4×4 *matrix where rows correspond to variables (in the same or-*

der of the VAR specification) and columns to shocks. Thus, the monetary policy shock is ordered third and the contirbution of monetary policy at two year horizon is given by the FEVD(:,indx_sho) *column, which gives*

```
%=====================================================%
% Forecast Error Variance Decomposition of MP shock   %
% Percentage of variability explained by MP shock     %
    logip      logcpi    gs1        ebp
    1.2926     2.7191    83.7579    0.5216
%                                                     %
%=====================================================%
```

Example 15 (Historical Contribution of Demand, Supply and Monetary Policy shocks) *Consider the same settings of exercise 11. We plot the historical decomposition of the one year government bond and Excess Bond Premium in terms of AD, AS, MP and initial conditions. Combine together Demand and Supply shocks. The following commands generate the required historical decomposition:*

```
% uses the zero-sign restrictions average rotation
opts_.Omega        =  mean(bvar4.Omegaz,3);
% by default it uses mean over posterior draws
[yDecomp,ierror]  = histdecomp(bvar4,opts_);

% yDecomp = historical decomposition
% time, variable, shocks and initial condition
% ierror = structural innovation

% Declare the names of the variables in the order they appear in the VAR
bvar4.varnames       = {'IP','CPI','Interest Rate','EBP'};
% select the variables for the plot of the historical decomposition
optnsplt.plotvar_     = {'Interest Rate','EBP'};
% select the shocks combination to report
optnsplt.snames_ = { {'Shck1','Shck2'};...    Combine Supply and Demand
    {'Shck3'};...              MP
    {'Shck4'} ...              Other shock not identified in the VAR
    };
% declare the name of the shocks
optnsplt.stag_        = {'Supply+Demand';
            'MP';
            'Other Shocks';
            'Initial Condition'};
```

```
% name of the file to save
optnsplt.save_strng    = 'y0';
% define the time for the plot
optnsplt.time          = T(1+lags:end);
% define the directory where the plot is saved
optnsplt.saveas_dir    = '.\sdcmp_plt';
% limit the plot to a specific time window
optnsplt.Tlim          = [2006 2012];
% finally plot the decompostition
plot_sdcmp_(yDecomp,bvar4,optnsplt)
```

Figures 10 amd 11 report the historical decomposition of the one year government bond and EBP from 2006 until 2012.

Figure 10: Historical decomposition of one year government bond in terms of demand, supply, monetary policy, and other shocks and the initial condition.

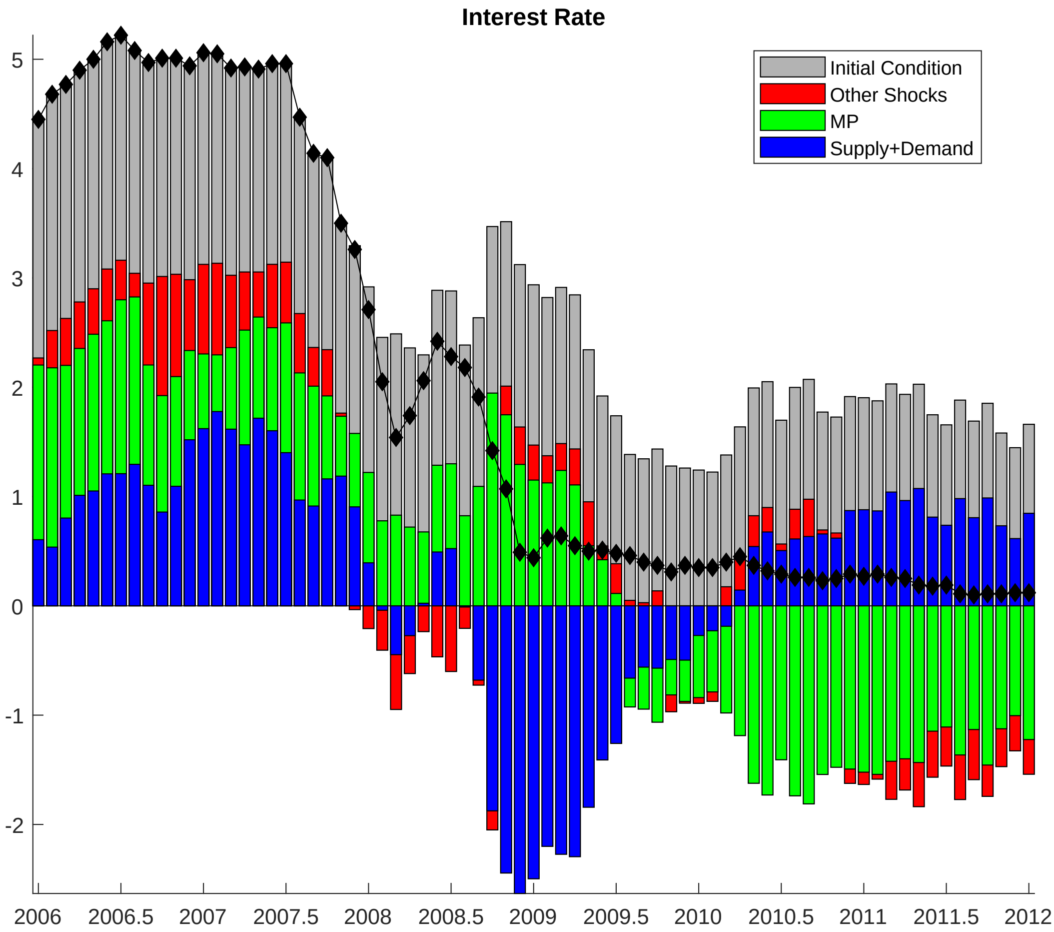
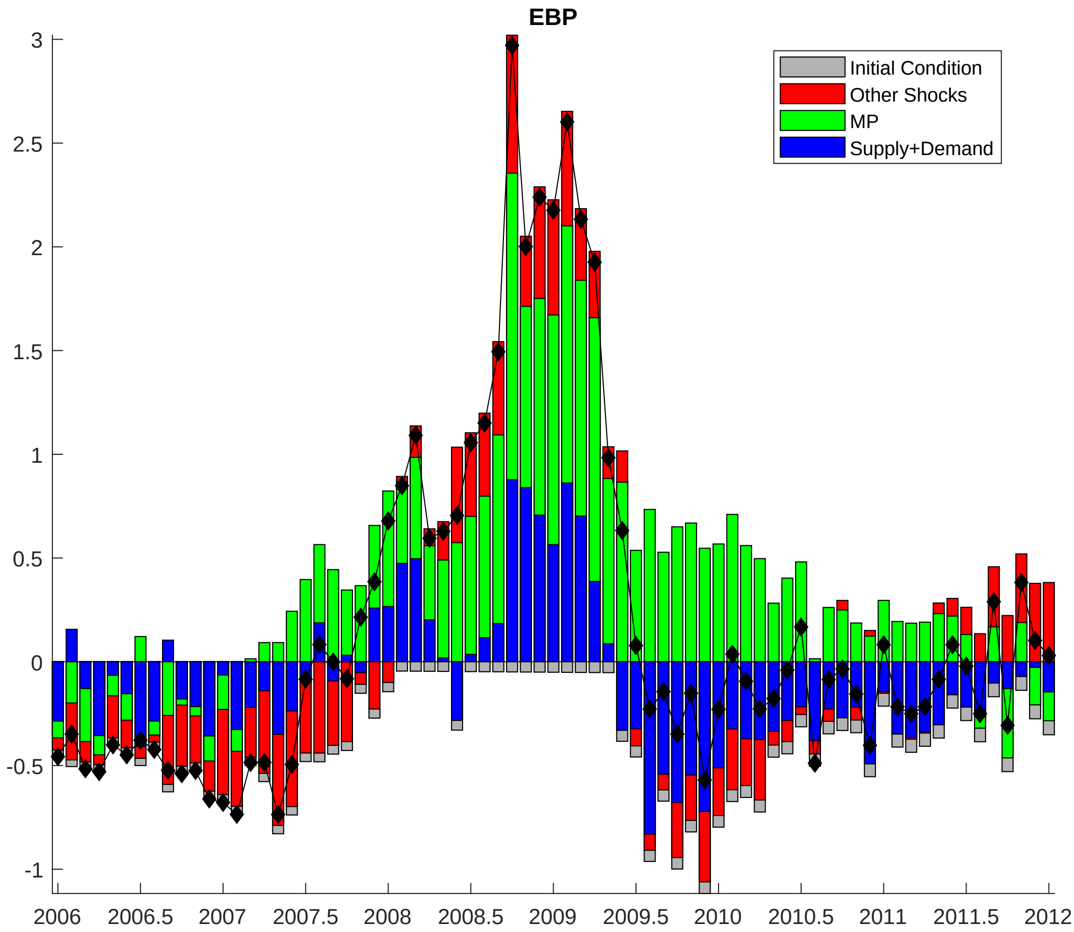


Figure 11: Historical decomposition of Excess Bond Premium in terms of demand, supply, monetary policy, and other shocks and the initial condition.



4.11 Tricks and applied tips

As we have mentioned at the beginning, the toolbox has limitation. For example, it does not deal with parametric time varying coefficient or stochastic volatility models. However, it is easy to take to account for time variations in the coefficients and in the variance within the toolkit. For example, one can run rolling windows VARs and measure whether the parameters, the responses, or other interesting statistics vary over time. [11]

A number of authors highlighted that the responses of real GDP to monetary policy shocks are larger in the pre-1979Q3 period than in the post-1984Q1 period.[12] The following example describes how one can measure how much variations there are in the response of unemployment to monetary policy disturbances over time.

Example 16 (The responses of unemployment to monetary policy shocks over time) *Consider a three variables VAR with inflation, a measure of economic activity, and the short term interest rate (as in Primiceri (2005)). Inflation is measured by the annual growth rate of a chain weighted GDP price index. Unemployment, which is used as measure of real activity, refers to the unemployment rate of all workers over 16. The nominal interest rate is the yield on 3-month Treasury bills, which is preferred to the more conventional federal fund rate, because it is available for a longer period. The sample runs from 1953Q1 to 2007Q4. Two lags are used for the estimation.*

We consider a 30-years window and a one year time shift between adjacent windows. Thus, the first estimation window is 1953Q1:1982Q4, the second is 1954Q1:1983Q4, etc. To speed up the calculation, we limit attention on the response functions obtained with OLS estimates of the autoregressive parameters and of covariance matrix which are stored in the matrix `BVAR.ir_ols` *with the usual ordering of the dimensions (i.e. variable, horizon, and shock). We report the dynamics of the unemployment rate, which is ordered second. Responses are plotted in a 3 dimensional space, where the first axis corresponds to the horizon of the response, the second to the window, and the third to the response of unemployment rate. Figure 12 presents the response of the unemployment rate to a standardized 1% increase in the 3m Treasury bill. As expected, the peak response has been weakening over time and the negative impact in the medium run is stronger.*

```
load DataQ   % load the data
yQ          = [GDPDEF_PC1 UNRATE TB3MS]; % select  GDP deflator (year-on-year) Unemploymen
lags        = 2;

% setting for the rolling windows
Wsize       = 120;     % quarterly lenght of the rolling window
shift       = 4;       % time shift between adjacent windows
```

[11]For Bayesian kernel estimation of VAR models see Petrova (2019).
[12]See e.g. Boivin, Kiley and Mishkin (2010).

```
indx_sho     = 3;       % shock of interest
options.K    = 1;       % no MC draws, use the OLS IRF (BVAR.ir_ols)
options.hor = 24;       % horizons of  the  IRFs
rollIRF      = ...      % initialize the rolling IRF (one IRF per window)
    nan(size(yQ,2),options.hor,1);
timespan     = nan(1,Wsize);
w            = 0;       % index for counting the windows

% start the rolling window estimation of the IRFs
while (w*shift + Wsize) <= size(yQ,1)
    w = w + 1;
    timespan(w,:) = shift*(w-1) + 1 :  Wsize + shift * (w-1);
    rollbvar    = bvar(yQ(timespan(w,:),:),lags,options);
    % normalize for the shock size  (1% increase in 3mTbill on impact)
    norm            = rollbvar.ir_ols(indx_sho,1,indx_sho);
    % Collect the responses to  monetary policy shocks in the window
    rollIRF(:,:,w) =  squeeze(rollbvar.ir_ols(:,:,indx_sho))/norm;
end

figure('name','UNR')
tt = T(timespan(:,end)');
surf(tt,[1:options.hor],squeeze(rollIRF(2,:,:)))
axis tight
```

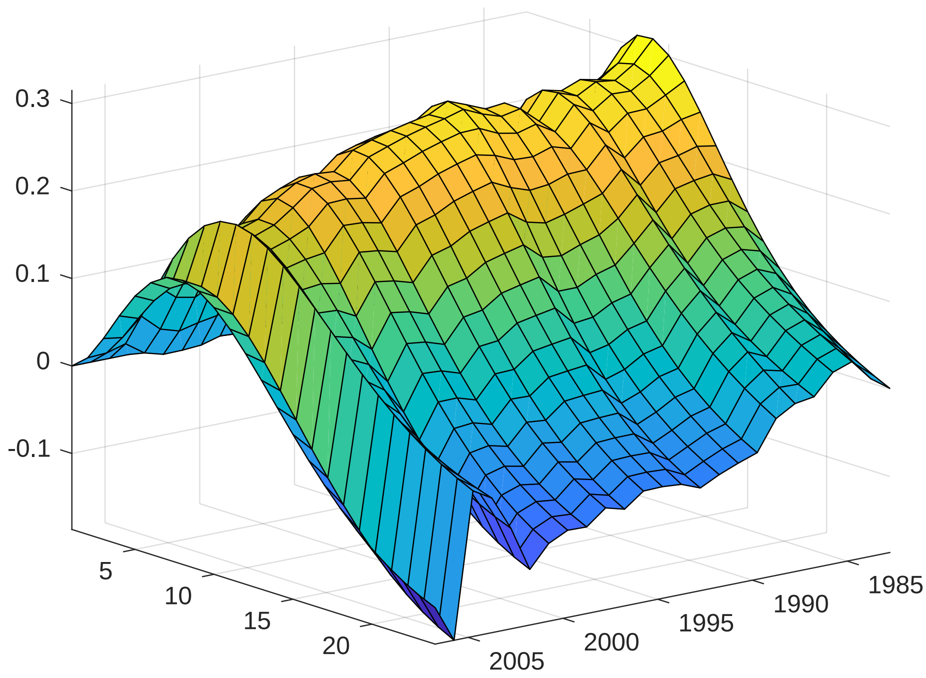
One way to take into account stochastic volatility is to scale the variables of the VAR prior to estimation using a recursive (non-parametric) measure of volatility. For example, one can preliminary run univariate regression of y_{jt} on its own lags compute $w_{1jt} = |e_{jt} = y_{jt} - A_j(L)y_{jt-1}|$ or $w_{2jt} = e_t^2$ and use, as VAR variables, y_{jt}/w_{1jt} or $y_{jt}/\sqrt{w_{2jt}}$. The first of these transformations can be performed using the following commands

```
opt.K =1;
yhat = nan(size(y,1)-lags,size(y,2));
for var =1 : size(y,2)
    tmp_      = bvar(y(:,var),lags,opt);        % univariate AR(lags) process
    yhat(:,var) = y(lags+1:end,var) ./ abs(tmp_.e_ols);
end
```

Then, one can run the VAR on the transformed variables, `yhat`.

VAR models are linear. Thus, interesting questions about the differential effects of positive or negative, or large or small shocks can not be answered. One can deal with asymmetric

Figure 12: Time evolution of the responses of the unemployment rate to a 1% increase in the 3m Treasury bill.



shocks is via identification restrictions (as in Canova and Pappa (2011)). Alternatively, can build VARs with interacted variables (as in Sa, Towbin and Wieladek (2014)), where the interaction term, for example, accounts for positive vs. negative states. Although none of these options is perfect, they can give researchers hints about what they care about without having to estimate a fully non-linear model, which is computationally costly.

These options can be managed within the toolkit, but they require some attention. For example, using sign and quantity restrictions one can measure the shocks that have large and small effects on the variables of interest, see 4.3. Similarly, by interacting one or more variables of the vector $\mathbf{y}$ with a set of dummies, one can see how shocks propagate in various states. We let the creative user to come up with her favorite trick to implement the non-linear model she wishes to analyze with the toolkit.

5 Mixed Frequency Vars

The Mixed Frequency VAR (MF-VAR) we considered is based on a standard constant parameter model and designed to deal primarily with data at the monthly and quarterly frequencies. The MF-VAR setup can be also employed when one or more of the VAR variables display missing observations. Thus, it can be used for backcasting, nowcasting, or for retrieving variables that have an arbitrary pattern of missing observations.

From a Bayesian perspective, we need to construct the joint distribution of the observables `y`, of the latent states, and of the parameters, conditional on the pre-sample used to initialize the lags. Using the Gibbs sampler, we generate a draw from the posterior distributions of the reduced form VAR parameters, conditional on observables and states. Based on this draw and with the Kalman smoother, we can estimate the unobserved states, and thus implicitly recover the values of the variables whose observations are missing.

A successful application of the MF-VAR function is to employ the most recent and more frequently available information (e.g. monthly) to construct timely estimates of variables of interest that are observed only at lower frequency (e.g. quarterly); this allows to generate nowcasts that incorporate available information efficiently. And the latter is crucial for forecasting as 'good' nowcasts are typically sound prerequisites for generating accurate out-of-sample forecasts (see section 7). For example, applied researchers and economic analysts are often interested in monthly version of the GDP. Since GDP is measured only at quarterly frequency and typically its release from national statistical offices comes with lags, researchers often use the information released monthly to estimate the monthly level of GDP. MF-VAR is well suited for this purpose. In particular, the MF-VAR treats the monthly GDP values as unobserved, and to cope with missing observations it uses a state space representation. The state vector includes the monthly observed variables and the monthly unobserved counterparts of the quarterly observed variables. The construction of the state space and its estimation closely follows Schorfheide and Song (2015). Another interesting application could be to use monthly and weekly times series and use the MF-VAR to estimate a weekly version of the monthly variable; e.g. one could consider the weekly Unemployment Insurance claims or/and weekly google trends to infer the weekly unemployment rate, which only is measured monthly. For the purpose of this document and for fixing ideas, we only discuss MF-VAR with monthly and quarterly variables but its extension to weekly-monthly or quarterly-annual frequency is relatively straightforward.

Estimation of a MF-VAR can be performed using any of the prior specifications discussed in section 3 with the caveat that maximization over the Minnesota prior hyper-parameters is not permitted. After MF-VAR estimation is performed, one can proceed with the identification and the computation of impulse response functions, as discussed in section 4, and of point or density forecasts, as discussed in section 7.

The MF-VAR estimation is triggered automatically whenever there are `NaN` in the array matrix containing the data, `y`. When this is the case, a warning message is printed on the command window. One can handle the unobserved variables in several ways; the toolbox's approach is to vary the dimension of the vector of observables as a function of time t (as in, e.g., Durbin and Koopman (2001) and Schorfheide and Song (2015)).

Assume that we have monthly and quarterly variables, $\{y_t^m, y_t^q\}$, and that t corresponds to the month. For quarterly variables, we have observations only on the last month of the quarter and `nan` for the other months. The measurement for monthly variables is trivial, i.e. $y_t^m = x_{m,t}$. By default, quarterly variables are treated as stocks; in other words, the monthly unobserved state, x_t^q, is mapped into quarterly observed variables, y_t^q, whenever the latter is observed. Thus we have:

$$y_t^q = x_{q,t}$$

Clearly this approach is appropriate for stock variables, such as debt or capital. It can also be used with mixed quarterly and annual frequency variables when the annual variable is a stock. For quarterly flow variables (such as GDP), we use the quarterly aggregator

$$y_t^q = \frac{1}{3}(x_{q,t} + x_{q,t-1} + x_{q,t-2})$$

where y_t^q is the quarterly observed variables and $x_{q,t}$ is the monthly unobserved counterpart. To activate this option, type

```
options.mf_varindex     = num;
```

where `num` is a scalar indicating the position (column on `y`) in the dataset of the flow variable we wish to aggregate. Once the mapping of monthly and quarterly variables is chosen, we assume that $[x'_{q,t} \;\; x'_{m,t}]'$ follows a VAR with p lags.

The unobserved or irregularly sampled variable is an additional output of the `bvar` function; in particular, the $\mathtt{T} \times \mathtt{n} \times \mathtt{K}$ array called

```
BVAR.yfill
```

contains the smoothed (two-sided) estimates of the unobserved states, where the first dimension is time, the second is the variable, and the third is a specific draw from the posterior distribution of the parameters. The $\mathtt{T} \times \mathtt{n} \times \mathtt{K}$ array called

```
BVAR.yfilt
```

contains the filtered (one-sided) estimates of the unobserved states.

5.1 Example: Construction of the monthly GDP for the Euro Area

We use monthly data on the log of industrial production index (`IPI`), the log of HICP (`HICP`), the log of HICP excluding food and energy (`CORE`), the one-year Euribor (`Euribor1Y`), the unemployment rate (`UNRATE`) from 2000m1 to 2015m8 to construct a monthly version of GDP and GDP growth. We assume a VAR with six lags. We first construct the database.

```
% load the mixed frequency data
load DataMF
% select the variables
y = [GDP IPI HICP CORE Euribor1Y UNRATE];
% specify the lag lenght
lags = 6;
```

We treat GDP as a flow variable, assume Minnesota prior with default hyper-parameters values; the following instruction are needed.

```
options.mf_varindex      = 1;
options.K                = 1000;    %   number of draws
options.priors.name      = 'Minnesota';
% estimate the bvar
bvarmf                   = bvar(y,lags,options);
```

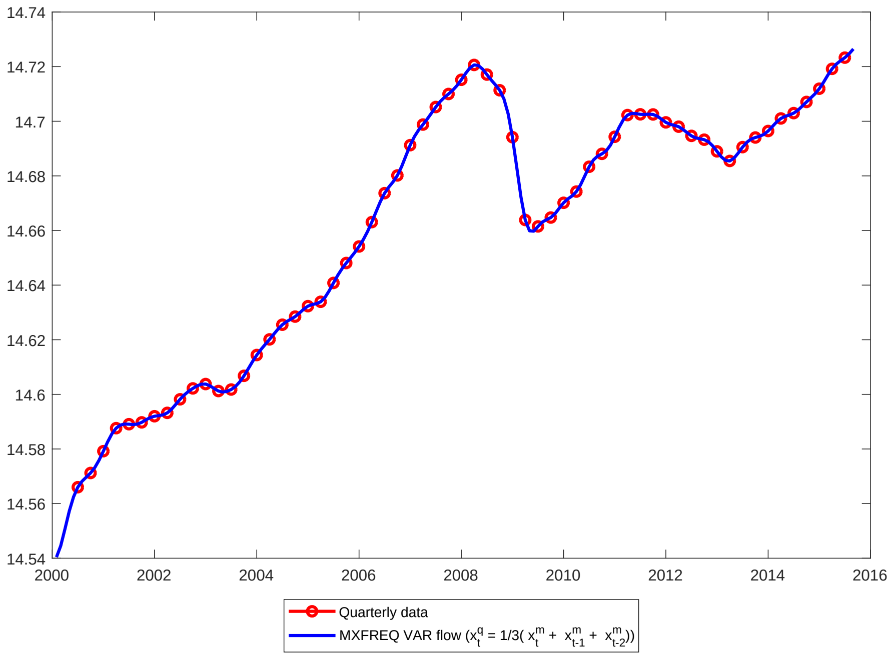
In the command window, the toolbox prints the maximum difference between the observed data and the corresponding filtered (one-sided) and smoothed (two-sided) states.

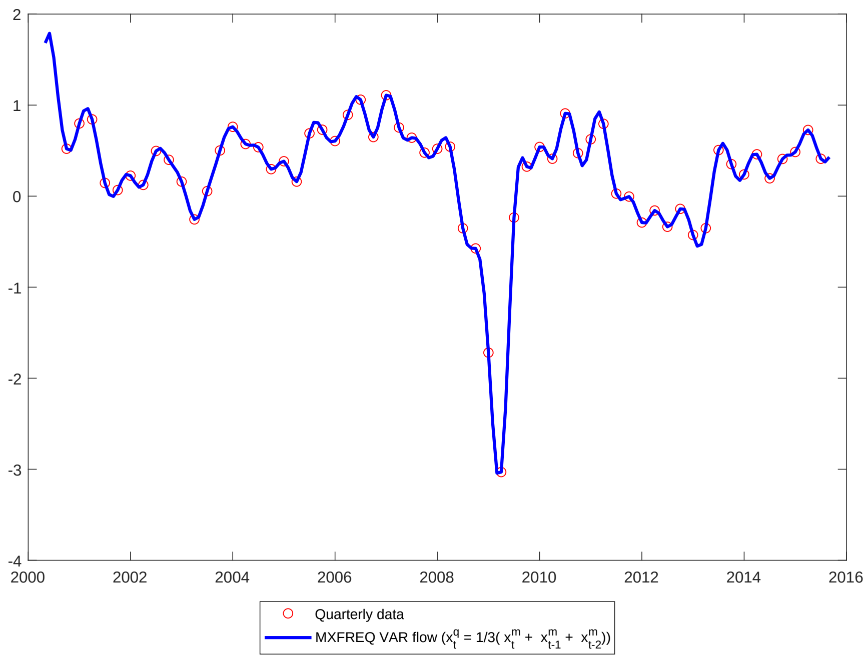
```
Warning: Activating the Mixed Frequency BVAR
Max Discrepancy Filtered vs. Actual Data: 1.9103e-12
Max Discrepancy Smooth vs. Actual Data: 8.1174e-11
Max Discrepancy Filtered vs. Actual Data: 2.1771e-11
Max Discrepancy Smooth vs. Actual Data: 7.0942e-10
...
```

To disactivate the warning, set `options.noprint =1`. Figure 13 reports the median estimate of the unobserved monthly GDP in level and growth rate.

Remark 17 *As mentioned, the maximization over the Minnesota prior hyper-parameters is not permitted in the toolbox. This occurs because with mixed frequency and unobserved states, the marginal likelihood cannot computed in closed form and we need estimators. There are two ways to disciple the Minnesota hyper-parameters in this context. First, one could run the prior hyper-parameters maximization using only the higher frequency variables and take the mode values of this first step to set they hyper parameters values of the MFVAR. Alternatively, the toolkit computes the posterior kernel,* $p(Y|\Sigma_{(m)}, \Phi_{(m)}, X_{(m)})$*, evaluated at a specific draw* (m)*; a* $\mathtt{K} \times 1$ *array called* `BVAR.logL` *collects the posterior kernel evaluation at each draw. Based on the kernel values one could construct an estimate of the marginal likelihood, using say Geweke (2000), for a given value of the hyper-parameter. One could then maximize the estimated marginal likelihood over a grid of value for the hyper parameter. This alternative route is obviously more time consuming.*

Figure 13: MF-VAR estimates of monthly GDP level and of monthly GDP growth





6 FAVARs

When n is large, and T is short, running a VAR may be problematic. In this case, let $y_t = [y_{1t}, y_{2t}]$, where y_{2t} contains data excluded from the VAR and to be used to construct factors w_t. Then, one can keep the VAR small by employing $\tilde{y}_t = [y_{1t}, w_t]$ as observables. This avoids overparameterization but spans the full information set a researcher has available. Once `BVAR` is run on $\tilde{y}_t$, structural inference on y_{2t} can be conducted either with standard FAVAR techniques; i.e. use the estimated loadings to transform the responses of the factors into responses of the variables in y_{2t} or, once the structural disturbances of interest are obtained, use a series of local projections of y_{2t} on the relevant identified disturbances.

Let y_2 be a $T \times n_2$ matrix containing the variables excluded from the VAR and suppose that y_2 is well described by a static factor model structure:

$$y_2 = W\Lambda' + E$$

where W is a $T \times n_w$ matrix of principal components, Λ is a $n_2 \times n_w$ matrix of factor loadings and E is a $T \times n_2$ matrix of idiosyncratic error terms. The function `{pc_T}` allows to extract the first principal components out of a time series database:

```
[E, W, Lambda, EigenVal, STD] = pc_T(y_2, n_w, transf);
```

`y2` contains the variables excluded from the VAR, `n_w` is a scalar indicating the number of factors and `transf` indicates the data transformation to be used, where 0 = no transformation, 1 = demean, 2 = demean and standardize. `E, W, Lambda` are the idiosyncratic errors, the principal components and the components loadings, respectively. Typically, `y_2` is assumed to be stationary and standardized to make the unit of the n_2 variables comparable; when `transf=2`, `STD` contains the $n_2 \times 1$ vector containing the standard deviation of `y_2`; otherwise it is a vector of ones.

The FAVAR model can be estimated using the `bvar` function on compressed and raw data as follows:

```
FAVAR = bvar([W y_1], lags)
```

Any of the prior specifications discussed in section 3 including the maximization over the Minnesota prior hyper-parameters can be used to estimate the FAVAR parameters. One can also construct impulse response functions as discussed in section 4, and point or density forecasts, as discussed in section 7.

Because the transformed (standardized) variables are typically not very interesting, we need to construct rescaling coefficients that map the dynamics of the principal components back to the uncompressed raw variables, `y_2`; these coefficients depend on the factor loadings, the standard deviation of the `y_2` variables, the relative dimension of `y_1`, `y_2` and `W`. Let `n_1` be the number of variable in `y_1`, the following function allows to construct the matrix of rescaling coefficients

```
ReScale = rescaleFAVAR(STD,Lambda,n_1)
```

where ReScale is a (n_2+n_1) $\times$ (n_w + n_1) matrix. By default factors loadings are ordered first. If factors are ordered after y_1 in the FAVAR, type

```
ReScale = rescaleFAVAR(STD,Lambda,n_1, 2)
```

The objects of interest (e.g. forecasts or impulse response functions) about y_2 can then be retrieved combining the FAVAR fields and ReScale. For example, assume that one is interested in the mean forecast of y_2, we can type

```
mean_forecast_y = mean(fabvar.forecasts.no_shocks,3) * ReScale';
```

which contains the forecast of both y_2 and y_1. Analogously, one can retrieve the response of y_2 to the structural shock identified using W and y_1, as it is shown in the following example

Example 18 (FAVAR: IRF of y_2 to monetary policy shocks) *Consider the database available at https://research.stlouisfed.org/pdl/788 which contains a number of slow moving quarterly variables, such as real quantities and price indices. We use the growth rate of the variables and the sample spans the period 1980Q1 to 2007Q4; we employ the same ID name available in the FRED database and variables names are collected in* varnames_y2. *We compress 28 slow moving variables, which are in y_{2t}, into three factors. The FAVAR has 4 variables: the three month Treasury Bill (y_{1t}) and three factors (w_{1t}, w_{2t}, w_{3t}). We identify a monetary policy disturbance using a recursive identification where slow moving variables are ordered first. We trace the responses of real GDP growth and core PCE inflation to the identified shock.*

```
clear all
% load favar data (Quarterly)
load DataFAVAR
% y1 is interest rate (TBILL3M)
% y2 are slow moving variables (YoY growth rates)
transf = 2;          % standardize y2
% extract the first 3 Principal Components (PC)
nfac   = 3;          % number of PC
[~,fhat,Lambda,~,STD] = pc_T(y2, nfac, transf);

% use the PC of the slow moving variables (PC) and the TBILL3M in the FAVAR
y = [fhat y1];

% estimate under recursive identification
lags   = 2;
fabvar = bvar(y,lags);
```

```
% Rescale the estimates from (3PC and y1) back to (y2 and y1)
% PC are ordered first
order_pc = 1;
C_       = rescaleFAVAR(STD,Lambda,size(y1,2),order_pc);

% construct the IRF for the shock of interest (using a  number  of  draws)
% shocks of interest: MP (4th innovation  in  the VAR)
indx_sho      = nfac + 1;
for k = 1: fabvar.ndraws   % iterate on draws
    fabvar.irX_draws(:,:,1,k) =  C_ * fabvar.ir_draws(:,:,indx_sho,k);
end

% Indentify the variables of interest for the plots (real GDP and CORE PCE)
[~,indx_var] = ismember ({'GDPC96','JCXFE'},varnames_y2);
% Real GDP and CORE PCE and TBILL3M
irfs_to_plot = fabvar.irX_draws( indx_var, :, 1, :);

% Customize the IRF plot
% variables names for the plots
options.varnames      = {'GDP','CORE PCE'};
plot_irfs_(irfs_to_plot,options)
```

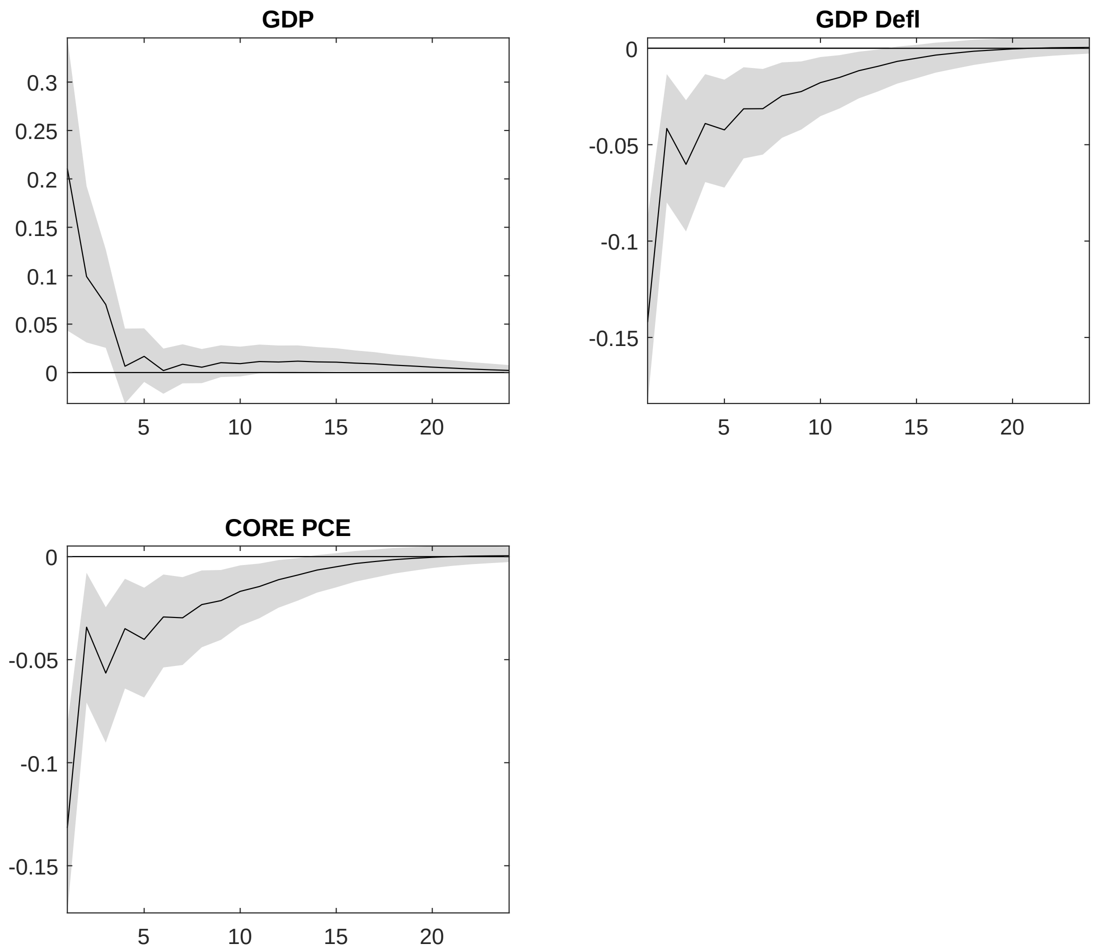
A bit more involved is the identification of shocks, when restrictions are imposed in part on y_2 variables. The following example shows how to conduct structural inference in this case.

Example 19 (FAVAR with sign restrictions on y_2) *Consider the same setting of exercise 18. We identify an aggregate supply shock assuming that the sign of real GDP (the GDP deflator) is positive (negative) for the first two quarters after the shock. We compute the responses of real GDP and deflator and of core personal consumption expenditure (PCE). The commands to perform this exercise are described below and Figure 14 reports the responses of the three variables to a sign-identified aggregate supply shock.*

```
% Use sign restrictions on the uncompressed variables.
% Idnetification: aggregate supply: GDP (+) GDP deflator (-).
% Assume that AD is the first shock
[~,indx_var] = ismember ({'GDPC96','GDPCTPI'},varnames_y2);

signrestriction{1} = ['y(' num2str(indx_var(1)) ',2:3,1)>0;'];
signrestriction{2} = ['y(' num2str(indx_var(2)) ',2:3,1)<0;'];
```

```
for k = 1 : fabvar.ndraws % iterate on draws
      Phi       = fabvar.Phi_draws(:,:,k);
      Sigma     = fabvar.Sigma_draws(:,:,k);
      % compute the impulse response function with a particular
      % draw from the posterior distribution.
      % Note: 'C_' is computed in the previous example
      [ir,Omeg] = iresponse_sign(Phi,Sigma,fabvar.hor,signrestriction,C_);
      fabvar.irXsign_draws(:,:,:,k) = ir;
end

[~,indx_var] = ismember ({'GDPC96','GDPCTPI','JCXFE'},varnames_y2);
indx_sho     = 1;       % shocks of interest
irfs_to_plot = fabvar.irXsign_draws(indx_var ,:,indx_sho,:);

% Customize the IRF plot
options.saveas_dir    = './irfs_plt';   % folder
options.saveas_strng  = 'FaVAR';        % names of the figure to save
options.shocksnames   = {'AS'};         % name of the shock
options.varnames      = {'GDP','GDP Defl','CORE PCE'};  % variables names for the plots
% the plotting command
plot_irfs_(irfs_to_plot,options)
```

Figure 14: Responses of real GDP, GDP deflator and Core PCE, following an aggregate supply shock.



7 Predictions

Regardless of the prior distribution used, one can easily generate forecasts from a BVAR. Following Del Negro and Schorfheide (2013), we employ the following algorithm

Algorithm 2

Given the posterior of the parameters, $p(\vartheta|Y)$, for $m = 1, ..., M$

1. Draw $\vartheta^{(m)} = vec(\Phi_0^{(m)}, ..., \Phi_p^{(m)})$ from $p(\vartheta|Y)$.

2. Draw a sequence of u_t's, i.e. $u_{T+1:T+h}^{(m)}$, from a $N(0, \Sigma^{(m)})$, and iterate on the VAR representation, i.e.

$$y_\tau^{(m)} = \Phi_0^{(m)} + \Phi_1^{(m)} y_{\tau-1} + ... + \Phi_p^{(m)} y_{\tau-p} + u_\tau^{(m)}$$

for $\tau = T + 1, ..., T + h$.

The algorithm generates M out-of-sample trajectories, $\left\{ y_{T+1:T+h}^{(m)} \right\}_{m=1}^M$; these trajectories can then be used to obtain numerical approximations of moments, quantiles, or the predictive density.

By default, the `bvar` function generates out-of-sample unconditional forecasts for 12 periods. Suppose we wish to generate forecasts for the next `nfor` periods. To do this, we need to set the appropriate options:

```
options.fhor = nfor;
BVAR         = bvar(y,lags,options);
```

Forecasts are stored in `BVAR.forecasts` which contains the following subfields:

- `BVAR.forecasts.no_shocks` is a `nfor`×`n`×`K` matrix containing unconditional forecasts. The first dimension corresponds the horizon, the second to the variable, and the third to the draw from the posterior distribution of the parameters. These trajectories are constructed assuming that all future shocks are zeros (i.e. $u_{T+1:T+h}^{(m)} = 0 \; \forall m$). Thus, these forecasts only account for parameter estimates uncertainty.

- `BVAR.forecasts.with_shocks` is a `nfor` × `n` × `K` matrix containing unconditional forecasts with shocks. The first dimension corresponds the horizon, the second to the variable, and the third to the draw from the posterior distribution of the parameters. These trajectories include uncertainty in the parameter estimates and in the shocks realization.

- `BVAR.forecasts.conditional` is a `nfor` × `n` × `K` matrix containing the conditional forecasts, where the path for one or more endogenous variables of the VAR are specified. These conditional forecasts are generated using Waggoner and Zha (1999) and Maih (2010). To activate conditional forecasts, more inputs are needed. In particular,

  1. `options.endo_index` is a row array containing the index of the variable constrained to a specified path.

2. `options.endo_path` is a matrix containing the path for each variable (rows horizon, column variables). Notice that the number of rows in `options.endo_path` must coincide with `options.fhor`.
3. `options.exo_index` specifies the shocks of the VAR used to generate the assumed paths of the endogenous variabes. `exo_index` could be one or more shocks. If no structural identification is performed, the program uses a Cholesky factorization by defualt.

- `BVAR.forecasts.EPS` contains the shocks used to generate the conditional forecasts.

7.1 Plotting the Forecasts

The function `plot_frcst_(frcst_to_plot,y,T)` plots fan charts. `frcst_to_plot` is a three dimensional array where the first dimension corresponds to the horizons, the second to the variables, and the third to the trajectory produced by a particular draw from the posterior distribution; `y` and `T` are respectively the $\mathtt{T} \times \mathtt{n}$ array of data and the $\mathtt{T} \times 1$ array that defines the in-sample time span. For example, when considering unconditional forecasts without shocks, type `frcst_to_plot = BVAR.forecasts.no_shocks`. The output is one figure reporting the forecasts; by default, the figure consists of the upper bound integers of $\sqrt{\mathtt{n}} \times \sqrt{\mathtt{n}}$ subplots reporting the forecasts of all variables included in the VAR. Each panel reports a black solid line, which corresponds to the median forecast, and a gray area, which corresponds to the 68% high probability density (HPD) set of the empirical distribution. Different settings are possible. The general plot command is:

```
plot_frcst_(frcst_to_plot,y,T,options)
```

where the options are as follows

- `options.varnames` is a cell string containing the variable names for the subplots title. The length of the cell must be of the same size as the first dimension of `irfs_to_plot`;
- `options.time_start` defines the initial period where plot starts. `options.time_start` must be included in `T`; the default value is first date of the in-sample data.
- `options.conf_sig` is a number between 0 and 1 indicating the size of HPD set to be plotted; the default is 0.68.
- `options.conf_sig_2` is a number between 0 and 1 indicating the size of the **second** HPD set to be plotted.
- `options.nplots` is a 1×2 array indicating the structure of the subplots.
- `options.saveas_strng` is a string array indicating the name of the plot.
- `options.saveas_dir` is a string array indicating the directory where to save the plot. Figures are not saved if `options.saveas_dir` or `options.saveas_strng` are not specified.

- `options.add_frcsts` is a (`T+nfor`) $\times$ `n` array containing additional data and forecasts. The first dimension must coincide with the sum of the in-sample and out-of-sample sizes; the second with the size of the second dimension of `frcst_to_plot`.
- `options.order_transform` is a `1` $\times$ `n` array indicating the transformation to apply to each variable. Allowed transformation are
  1. `0` = no transformation
  2. `1` = period-by-period change
  3. `12` = 12 period over 12 period change multiplied by 100. With monthly variables expressed in logs, this represents the year over year change %.
  4. `4` = 4 period over 4 period change multiplied by 100. With quarterly variables expressed in logs, it represents the year over year change %.
  5. `100` = period over period change multiplied by 100. With annual variables expressed in logs, it represents the year over year change %.

Note that `bvar` does not allow the use of exogenous variables. A simple way to include exgovenous variables in the VAR is to add them as endogenous variables and use the conditional forecast option to set their future path.

7.2 A few examples: Euro Area forecasts

We use monthly data on the log of industrial production index (`IPI`), the log of HICP (`HICP`), the log of HICP excluding food and energy (`CORE`), the one-year Euribor (`Euribor1Y`), log of M3 (`M3`) and the nominal exchange rate (`EXRATE`). The sample runs from 2000m1 to 2015m8. We use the convention that January 2000 corresponds to 2000 and December 2000 to 2000+11/12. Data up to 2014m8 is used to estimate a VAR with six lags and the remaining 12 months are used to compare the forecasts with actual data. We first construct the database.

```
% load the data
load Data
% select the variables
y_actual = [IPI HICP CORE Euribor1Y M3 EXRATE];
% stop estimation at August 2014
in_sample_end = find(time==2014 + 7/12);
y             = yactual(1:in_sample_end,:);
T             = time(1:end-in_sample_end);
```

Example 20 (Comparing different point forecasts) *Compute the following conditional and unconditional forecasts:*

1. *Unconditional forecasts, using flat priors.*

```
lags         = 6;
% one year forecasts
options.fhor = 12;
b.var(1)   = bvar(y,lags,options);
```

2. *Unconditional forecasts, using Minnesota priors and default values.*

```
options.priors.name = 'Minnesota';
b.var(2)     = bvar(y,lags,options);
```

3. *Unconditional forecasts, using an optimal Minnesota prior.*

```
options.max_minn_hyper  = 1;
options.index_est       = 1:4;
options.max_compute     = 3;         % sims optimization routine
options.lb              = [0 0 0 0]; % set the lower bound
b.var(3)                = bvar(y,lags,options);
```

4. *Forecasts conditional on a path of the short run interest rate. We assume that interest rate path coincides with the actual realization of the Euribor rate from 2014m9 to 2015m8. We use an optimal Minnesota prior.*

```
options.max_minn_hyper      = 0;
options.minn_prior_tau      = b.var(3).prior.minn_prior_tau;
options.minn_prior_decay    = b.var(3).prior.minn_prior_decay;
options.minn_prior_lambda   = b.var(3).prior.minn_prior_lambda;
options.minn_prior_mu       = b.var(3).prior.minn_prior_mu;
% select the conditioning variables (Interest rate =4)
options.endo_index          = 4;
% impose a trajectory for the short term interest rate
% which coincides with the actual trajectory observed
options.endo_path          = Euribor1Y(in_sample_end+1:end);
c.var(1) = bvar(y,lags,options);
```

5. *Forecasts conditional on the path of the short run interest rate as before, using only monetary policy shocks identified via Cholesky decomposition to generate that path. We use an optimal Minnesota prior.*

```
options.exo_index    = options.endo_index;
c.var(2) = bvar(y,lags,options);
```

Figure 15 presents the mean forecasts of the exercises in 1) to 5). The gray area identifies the forecasting period - September 2014 to August 2015; the top panel displays the year over year growth rate of industrial production, the central panel the year over year growth rate of HICP and HICP excluding food and energy, and the bottom panel the one year Euribor.

While all forecasts miss the HICP disinflation of 2015, the forecasts with optimal Minnesota priors seem to perform better.

Example 21 (Fan Charts) *Plot the forecasts constructed using an optimal Minnesota priors (i.e. case 3 of Example 20) with credible sets. We use the* `plot_frcst_.m` *function to plot. The inputs are the forecasts (with shocks) e.g* `b.var(3).forecasts.with_shocks`*, the data used for in-sample estimation (*`y`*) and the in-sample time span (*`T`*). We also add the following options:*

- *Specify the directory where plots are saved (*`options.saveas_dir`*); the directory is created if it does not exists.*
- *Control the appearance of the subplots by using* `options.nplots`*.*
- *Start the plots in 2013.*
- *Customize the names for the subplot titles.*
- *Variable transformations: for industrial production, HICP, CORE and the EXRATE we are interested in the Year over Year percentage change. Since these data are monthly and in logs,* `options.order_transform=12` *will take the 12 periods difference and multiply by 100. We plot 68% and 90% coverage probability bands.*
- *Add the plot the actual data realization (or of different point forecasts) using the option* `options.add_frcst`*.*

The appropriate commands are reported below,

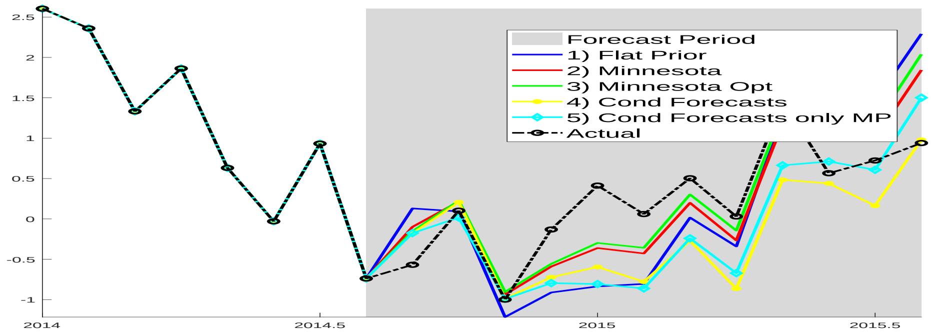
```
% select the forecast to plot (Option Minnesota)
frcsts                     = b.var(3).forecasts.with_shocks;

% Customize the plot
% declare the directory where the plots are saved
options.saveas_dir         = 'frcsts_plt';
% control the appearence of the subplots
options.nplots             = [3 2];
% start of the forecast plot - default first date of the in-sample data
options.time_start         = 2013;
% Transformations
% 12 = Year over Year percentage change with monthly data for IPI and HICP
options.order_transform  = [12 12 12 0 12 0];
% Titles for subplot
options.varnames          = ...
              {'Industrial Production','HICP','CORE','Euribor 1Y','M3','EXRATE'};
% second credible set to  be  plotted
```

```
options.conf_sig_2        = 0.9;
% add actual data
options.add_frcst         = yactual;
% Plot the forecasts
plot_frcst_(frcsts,y,T,options)
```

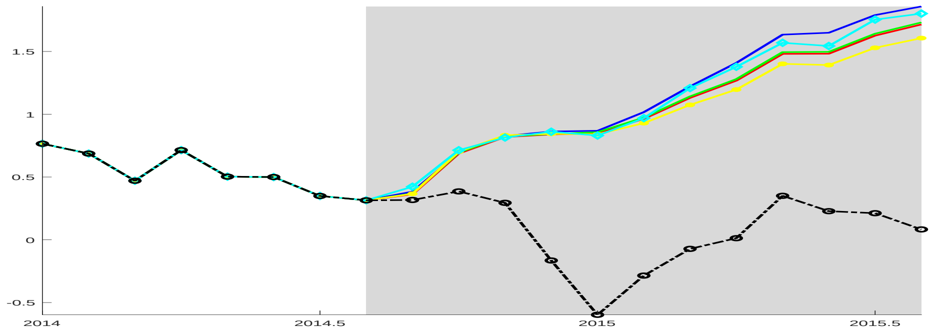
The figure 16 reports the forecasts. The black line is the median; light (dark) gray bands are the 68% (90%) credible sets. The blue line corresponds to the actual data.

Figure 15: Forecasts with various specifications

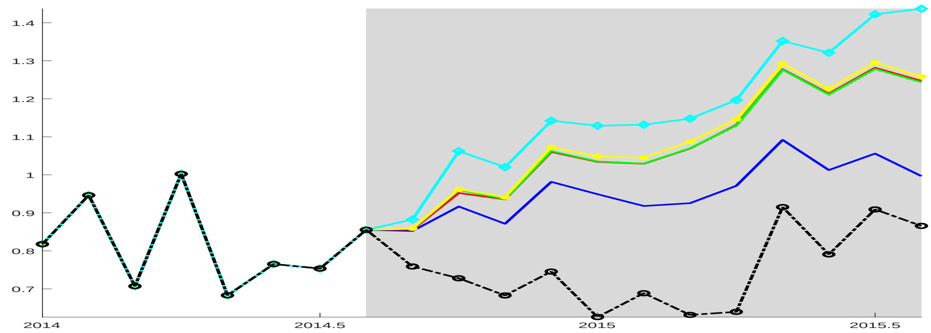
(a) IPI



(b) HICP



(c) CORE



(d) Euribor1Y

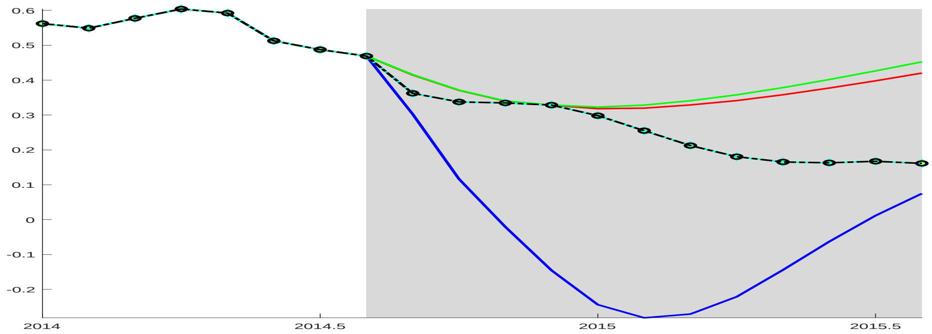
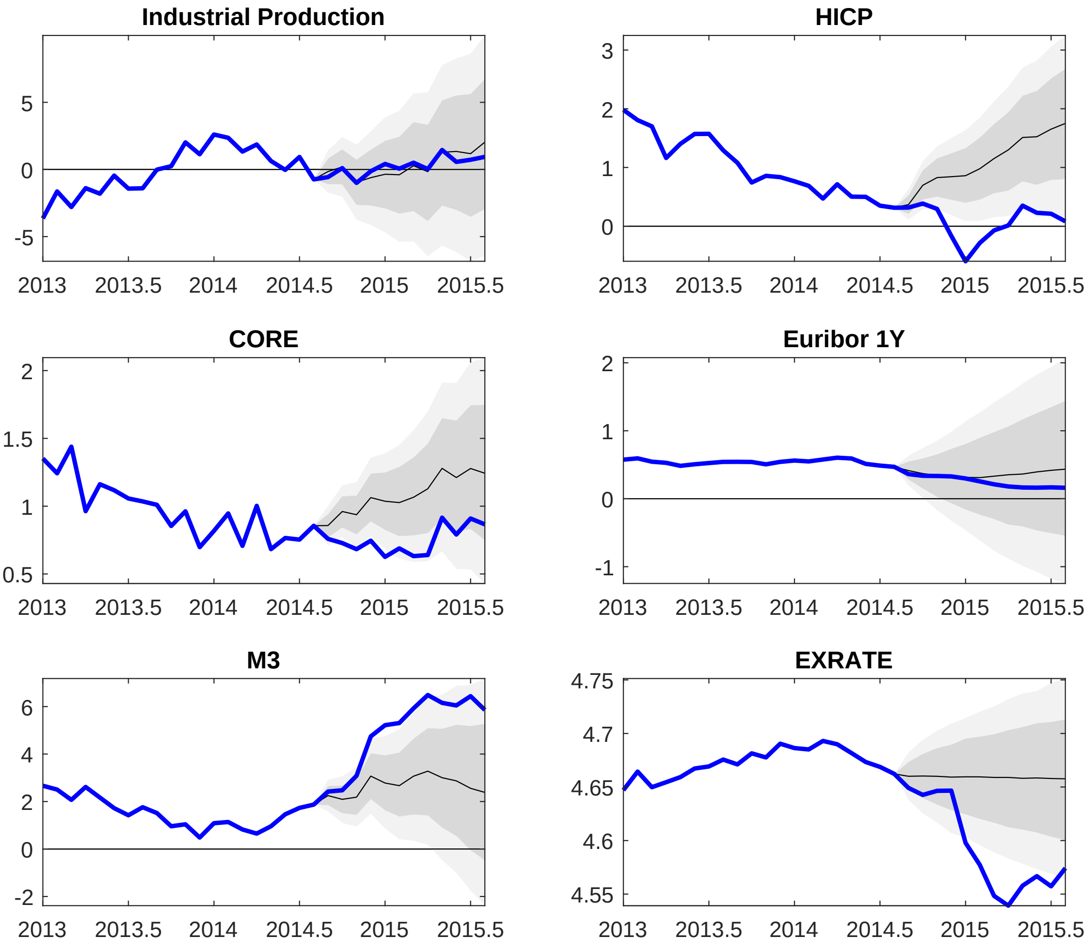


Figure 16: Fan charts with optimal Minnesota Prior. The black line is the median; light (dark) gray bands 68% (90%) credible sets. The blue line corresponds to the actual data.



8 Direct methods

Direct methods (DM) provide an alternative approach to study causal inference and for predicting in macroeconomics. In particular, one can forecast the future trajectory of interesting variables using direct forecasts (DF) rather than iterated forecasts, see Marcellino, Stock and Watson (2006), or study the dynamic transmission of monetary policy shocks using local projections (LP) rather than SVAR, as in Jordà (2005). While asymptotically VAR and LP procedures estimate the same impulse response function *up to a scaling factor* (see Plagborg-Møller and Wolf (2019)), in small samples researchers face a bias-variance trade off. The forecasting literature has highlighted that direct methods tend to have relatively lower bias, whereas VAR (iterated) methods tend to have relatively low variance but are more prone to be parametrically misspecified; and that the trade-off is most relevant at longer horizons. Furthermore, in the presence of near-unit roots and when considering long horizons, inferential conclusions may be different, see Plagborg-Møller and Montiel-Olea (2020).

To discuss how to implement DF and LP in toolbox, we need first some notation and the mapping between the two procedures.

8.1 Notation and Mapping

In a general formulation, direct methods can be represented as follows

$$y_{t+h} = \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + \Gamma_{(h)}(L)\mathbf{z}_t + e_t^{(h)} \qquad e_t^{(h)} \sim N(0, \Sigma_{(h)})$$

where y_{t+h} is a $n \times 1$ vector of endogenous variables, $\mathbf{x}_{t-1}$ is the vector containing p lags of the endogenous variables y_t and $\alpha_{(h)}$, $\beta_{(h)}$, and $\Sigma_{(h)}$ are matrices of coefficients of appropriate dimensions, $\mathbf{z}_t$ represents a set of exogenous controls, and $\Gamma_{(h)}(L)$ is a polynomial in the lag operator.

For the purpose of this section, we assume that a $VAR(p)$ correctly represents the data generating process; thus for now, $\mathbf{z}_t = 0 \ \ \forall t$. To ease the notation we consider the VAR companion form representation, i.e. equation (2), and pre-multiply by G' (notice that $G'G = I_n$). The following set of equations summarize the mapping between iterative and direct methods

$$
\begin{aligned}
y_t = \underset{(n\times np)}{G'} \mathbf{x}_t &= G' \underset{(np\times 1)}{F_0} + G' \underset{(np\times np)}{F} \mathbf{x}_{t-1} + \Omega\nu_t \\
&= \underset{(n\times 1)}{\alpha_{(0)}} + \underset{(n\times np)}{\beta_{(0)}} \mathbf{x}_{t-1} + \underset{(n\times 1)}{e_t^{(0)}}
\end{aligned}
$$

$$
\begin{aligned}
y_{t+1} = G'\mathbf{x}_{t+1} &= G'(I+F)F_0 + G'F^2\mathbf{x}_{t-1} + G'FG\Omega\nu_t + \Omega\nu_{t+1} \\
&= \alpha_{(1)} + \beta_{(1)}\mathbf{x}_{t-1} + e_t^{(1)} \\
&\vdots \\
y_{t+h} = G'\mathbf{x}_{t+h} &= G'\sum_{j=0}^{h} F^j F_0 + G'F^{h+1}\mathbf{x}_{t-1} + G'\sum_{j=0}^{h} F^{h-j}G\Omega\nu_{t+j} \\
&= \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + e_t^{(h)}
\end{aligned}
$$

where we assume that

$$
\alpha_{(h)} = G'\sum_{j=0}^{h} F^j F_0 \qquad \beta_{(h)} = G'F^{h+1} \qquad e_t^{(h)} = G'\sum_{j=0}^{h} F^{h-j}G\Omega\nu_{t+j}
$$

A few observations may highlight the feature of the mapping. First, if DM include $\mathbf{x}_{t-1}$ as regressors and $\Omega\nu_t \equiv u_t = e_t^{(0)}$, $e_t^{(0)}$ is the one-step ahead innovation in y_t. Second, under proper identification restrictions, the impulse response functions computed with the two procedures coincide:

$$
\frac{dy_{t+h}}{d\nu_t} \equiv G'F^hG\Omega = \beta_{(h-1)}G\Omega
$$

Third, assuming that T is the last observation data point, point forecasts computed with VAR and DM coincide if appropriate restrictions are imposed:

$$
\widehat{y}_{T+h} = G'\sum_{j=0}^{h-1} F^j F_0 + G'F^h\mathbf{x}_T = \alpha_{(h-1)} + \beta_{(h-1)}\mathbf{x}_T
$$

Fourth, the uncertainty around $\beta_{(h)}$ for $h > 0$ is typically larger than the uncertainty for $h = 0$, since $e_t^{(h)}$ compounds h shocks. Fifth, $\beta_{(h)}$ will be biased if some variable appearing in the DGP is excluded from $\mathbf{x}_{t-1}$ and omitted and included variables are correlated. Thus the use use of control variables may help to proxy for omitted lagged variables. Finally, the mapping between the two approaches could be use to elicit priors for Bayesian versions of direct methods, given the priors we have imposed on VAR coefficients and covariance matrix; see section 8.4 and appendix A.9 for details.

The restrictions to identify Ω presented in previous sections can be also employed with direct methods. For example, one could use rotation matrices or impose " hard" restrictions on the covariance matrix of $e_t^{(0)}$. When proxy or IV approaches are used, it is straightforward to adapt the SVAR we developed to direct methods. Assume that the shock of interest is ordered first in the VAR and there exists a parametric relationship between the shock of interest and the proxy variables as follows

$$\nu_{1,t} = \rho m_t + u_t^m$$

where u_t^m is a zero mean, measurement error. The DM specification corresponding to this situations is

$$\begin{aligned}
y_{t+h} &= \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + e_t^{(h)} \\
&= \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + F^h\Omega\nu_t + \sum_{j=1}^{h} F^{h-j}G\Omega\nu_{t+j} \\
&= \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + F^h\Omega\left(\begin{bmatrix}\nu_{1,t}\\ 0\end{bmatrix} + \begin{bmatrix}0\\ \nu_{-1,t}\end{bmatrix}\right) + \sum_{j=1}^{h} F^{h-j}G\Omega\nu_{t+j} \\
&= \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + F^h\Omega\left(\begin{bmatrix}\rho m_t\\ 0\end{bmatrix} + \begin{bmatrix}u_t^m\\ \nu_{-1,t}\end{bmatrix}\right) + \sum_{j=1}^{h} F^{h-j}G\Omega\nu_{t+j} \\
&= \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + \rho_{(h)}m_t + \overline{e}_t^{(h)}
\end{aligned}$$

where $\overline{e}_t^{(h)} = F^h\Omega\begin{bmatrix}u_t^m\\ \nu_{-1,t}\end{bmatrix} + \sum_{j=1}^{h} F^{h-j}G\Omega\nu_{t+j}$. The impulse responses of interest are contained in the vector $\rho_{(j)}$ $j = 0, ..., h$.

8.2 Local Projections

In the toolbox, local projection estimation can be performed using both classical and Bayesian techniques. By default, the toolbox computes the OLS (conditional Maximum Likelihood) estimator with robust HAC standard errors of the coefficients $\alpha_{(h)}$, $\beta_{(h)}$ and $\Sigma_{(h)}$ (and $\rho_{(h)}$ if a proxy is used). Once the parameters estimates are obtained, for a given rotation Ω, the impulse response function with confidence bands are computed as follows

$$\frac{dy_{t+h}}{d\nu_t} = \left(\widehat{\beta}_{(h-1)} \pm \tau_a SE(\beta_{(h-1)})\right) G\ \Omega \tag{4}$$

for $h > 0$, where $\widehat{\beta}_{(h-1)}$ is the OLS estimator, τ_a is the Student-t critical value probability a, $SE(.)$ is the standard error of the estimator. By default, it is assumed that

$$\Omega = chol(\Sigma_{(0)})$$

The impulse response function with confidence bands using IV-LP are instead given by

$$\frac{\partial y_{t+h}}{\partial m_t} = \widehat{\gamma}_{(h)} \pm \tau_a SE(\gamma_{(h)}) \tag{5}$$

The baseline estimation function for local projections is

```
[DM] = directmethods(y, lags, options)
```

The first input is the data, `y`, where we assume that there are no missing values. The second input is the number of lags, `lags`, which has to be an integer greater than zero, and which corresponds to the number of lags of the endogenous variables in y_t. The third input, `options`, specifies the options the user may want to employ; it can be omitted if default options are used. The options available in `DM` are:

- `options.hor` is a scalar indicating the horizon of the responses (default 24).
- `options.conf_sig` is a number between 0 and 1 indicating the size of the confidence interval; default is 0.9.
- `options.controls` is a (`T`×`n_c`) array of `n_c` exogenous controls and the first dimension of controls must coincide with the first dimension of the endogenous variables, `y`.
- `options.proxy` is a (`T` × `n_p`) array containing `n_p` variables that proxy the shock of interest. Notice that the first dimension of proxy variables must coincide with the first dimension of the endogenous variables, `y`.
- `options.robust_se_` a scalar indicating the type of standard errors for the OLS estimator; when it is set to 0 no adjustments are made; when it is set to 1 standard errors are HAC adjusted using Hamilton (2007), Ch 10, pag 282, eq (10.5.20) (default setting); and when it is set to 5, the MATLAB `hac.m` function is used (in which case the Matlab Econ Toolbox is required).
- `options.Q` is a (`n` × `n`) orthonormal matrix, i.e. $Q'Q = QQ' = I$, that can be used to rotate the Cholesky decomposition of the impact matrix. Recall that the toolbox assumes that $\Omega = chol(\Sigma_{(0)})Q$, and the default is $Q = I$.

The output of the function, `DM`, is a structure with several fields and sub-fields containing the estimation results:

- `DM.ir_lp` is a (`n` × `hor` × `n` × 3) array containing the impulse response function produced with a recursive identification scheme using equation (4). The first dimension corresponds the endogenous variable, the second to the horizon, the third to the shock. The first and third elements in the fourth dimension are to the upper and lower limits of the confidence interval, whereas the second dimension is the mean.
- `DM.irproxy_lp` a (`n` × `hor` × 1 × 3) array containing the impulse response function to a recursive identification scheme using LP using equation (5). The dimensions are the same as the ones discussed in the previous bullet point.

8.3 Direct Forecasts

The h-step ahead (point and confidence interval) direct forecasts, $\widehat{y}_{t+h}$, is computed as follows

$$\widehat{y}_{T+h} = \widehat{\alpha}_{(h)} + \widehat{\beta}_{(h)}\mathbf{x}_T \pm \tau_a \widehat{\Sigma}_{(h)}$$

Forecasts are also stored in `DM`; in particular,

```
DM.forecasts
```

is a ($\texttt{n} \times \texttt{hor} \times 3$) array containing the forecasts. The first dimension corresponds the endogenous variable, the second to the horizon. The first and third elements in the third dimension correspond to the upper and lower limits of the confidence interval, whereas the second element of the third dimension is the mean.

8.4 Bayesian Direct Methods

Bayesian inference is possible with direct methods. In this case, it is possible to generate the posterior distribution of the coefficients using a conjugate Multivariate Normal-Inverse Wishart prior, as discussed in the online Appendix of Miranda-Agrippino and Ricco (2017). We assume that $\alpha_{(h)}$, $\beta_{(h)}$ are asymptotically normally distributed and centered at the MLE estimator and $\Sigma_{(h)}$ is asymptotically inverted Wishart, centered on MLE estimator. Priors on $\alpha_{(h)}$, $\beta_{(h)}$ are assumed to be normally distributed and the prior $\Sigma_{(h)}$ is assumed to be inverse Wishart (see appendix A.11 for details). The first moments of the priors are centered on the iterated projections of F, F_0 and Σ. For example, the prior mean on $\beta_{(h)}$ is centered on F^{h-1}, where F is the estimated companion form matrix of the VAR parameters. Details on the prior and posterior construction of the Direct Methods parameters can be found in A.9. To construct the prior for the hyperparameters, one can use presample data or data from different country/sectors. The prior variance of the autoregressive coefficient is controlled by an hyperparameter, τ_h, which is horizon-specific. The command to active the prior is

```
options.priors.name = 'Conjugate';
```

With a conjugate setting, the prior for the autoregressive parameters is centered at zero with a diagonal covariance matrix of 10 and τ_h is set to 1; the prior for the covariance matrix of the residual is inverse Wishart with a unitary diagonal matrix as scale and `n+1` as degrees of freedom. If the user does not like these settings, she can customize the prior parameters as follows:

- `options.priors.Phi.mean` is a $(\texttt{n} \times \texttt{lags} + 1) \times \texttt{n}$ matrix containing the prior means for the autoregressive parameters.
- `options.priors.Phi.cov` is a $(\texttt{n} \times \texttt{lags} + 1) \times (\texttt{n} \times \texttt{lags} + 1)$ matrix containing the prior covariance for the autoregressive parameters.
- `options.priors.Sigma.scale` is a $(\texttt{n} \times \texttt{n})$ matrix containing the prior scale of the covariance of the residuals.
- `options.priors.Sigma.df` is a scalar defining the prior degrees of freedom.
- `options.priors.tau` a $(\texttt{hor} \times 1)$ vector controlling shrinkage at various horizons; the default value one.

One can set these options one at the time or jointly. The following approximate posterior distributions are constructed by the toolkit:

- `dm.ir_blp` is a (n×hor×n×K) matrix containing the Bayesian local projections impulse responses obtained with recursive identification. The first dimension corresponds to the variables, the second to the horizons, the third to the disturbances, and the fourth to the responses obtained with particular draw from the posterior distribution of the local projection parameters.
- `dm.irproxy_blp` is a (n × hor × 1 × K) matrix containing the Bayesian local projection impulse responses with IV identification. The first dimension corresponds to the variables, the second to the horizons, the third to the shock, and the fourth to the responses obtained with particular draw from the posterior distribution of the local projection parameters.
- `dm.bforecasts.no_shocks` is a (hor × n × K) matrix containing unconditional forecasts. The first dimension corresponds the horizon, the second to the variable, and the third to the draw from the posterior distribution of the parameters. These trajectories are constructed assuming that all future shocks are zeros. Thus, these forecasts only account for parameter estimates uncertainty.
- `dm.bforecasts.with_shocks` is a (hor × n × K) matrix containing unconditional forecasts with shocks. The first dimension corresponds the horizon, the second to the variable, and the third to the draw from the posterior distribution of the parameters. These trajectories include uncertainty in the parameter estimates and in the shocks realization.

The parameter vector that controls the overall prior shrinkage, `options.priors.tau`, can also be chosen to maximize the marginal data density. If the user prefers this option, she can use the command:

```
options.priors.max_tau = 1; % triggers maximization of the overall prior shrinkage
bdm_opt                = directmethods(y, lags, options);
```

By default, optimization is performed unconstrained and Chris Sims optimizer `cminwel.m` is used (this is `options.max_compute=3`). The following options can be set in the maximization step:

1. `options.lb` and `options.ub` set the lower and upper bounds for the optimization. Both are row array vectors of the same size of `options.priors.tau`.
2. `options.max_compute` is a scalar selecting the maximization routine to be employed:
   - `options.max_compute = 1` uses the MATLAB `fminunc.m` (unconstrained)
   - `options.max_compute = 2` uses the MATLAB `fmincon.m` (constrained)
   - `options.max_compute = 3` uses the Chris Sims's `cminwel.m`
   - `options.max_compute = 7` uses the MATLAB `fminsearch.m`

The first three are Newton, derivative-based algorithms; the latter is a direct search (simplex) method based on function comparisons. While typically slower, the latter method is useful in situations where derivatives are not well behaved.

8.5 A few examples

We use the same monthly data employed section 4.10. Recall that the `DataGK.mat` contains the log of industrial production index (`logip`), the log of CPI (`logcpi`), the one-year government bond rate (`gs1`) and the excess bond premium (`ebp`) which are the four series used in the Gertler and Karadi (2015) paper. The sample runs from 1979m7 to 2012m6.

Example 22 (LP: MP shock with recursive identification) *We estimate and plot the responses to a monetary policy impulse identified using a recursive identification scheme. We use the same lag length as in GK and estimate local projections impulse response function with default settings. First, we prepare the data*

```
load DataGK                    % load the data in the workspace
y = [logip logcpi gs1 ebp]; % combine the data in a Tx4 matrix y
```

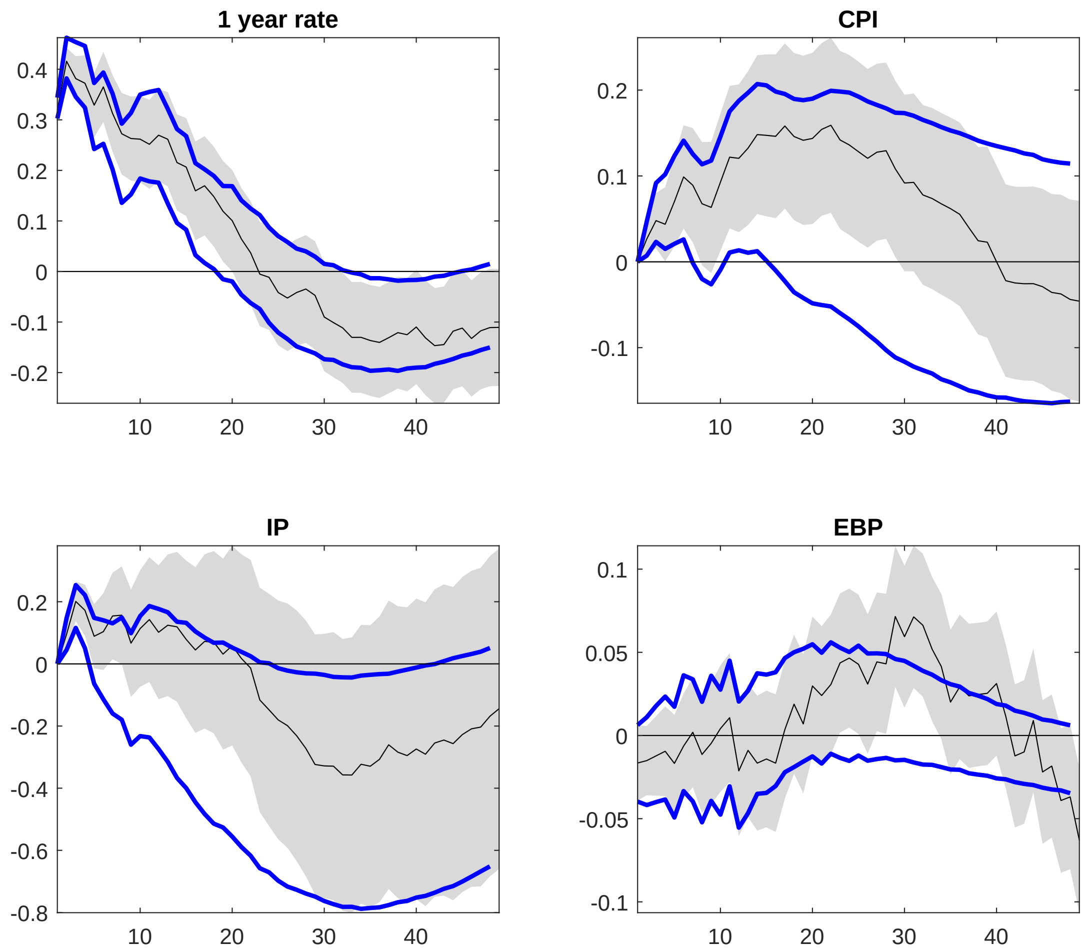
By default, the horizon of the responses is set to 24. We modify it to have the same horizon length as in GK and then run the model.

```
lags                = 12;
options.hor         = 48;
dm1 = directmethods(y,lags,options);

% Define the responses of interest
% index of the shocks of interest (shock to gs1)
indx_sho               = [3];
% Order of  the variables
% 1. logip; 2. logcpi; 3. gs1; 4. ebp
% Change the order for the plot
% 1. gs1; 2. logcpi; 3. logip; 4. ebp
indx_var               = [3, 2, 1, 4];
lprif2plot = dm1.ir_lp(indx_var,:,indx_sho,:);

% PLOT RESPONSES
% Customize the IRF plot
% variables names for the plots
options.varnames       = {'1 year rate','CPI','IP','EBP'};
% name of the directory where the figure is saved
options.saveas_dir     = './dm_plt';
```

```
% name of the figure to save
options.saveas_strng  = 'Cholesky';
% name of the shock
options.shocksnames   = {'MP'};
% Compare with BVAR estimates
bvar_ = bvar(y,lags,options);
var_irf_sort = sort(bvar_.ir_draws,4);
% add IRF plot
options.add_irfs(:,:,:,1) = var_irf_sort(indx_var,:,indx_sho,round(bvar_.ndraws*0.95));
options.add_irfs(:,:,:,2) = var_irf_sort(indx_var,:,indx_sho,round(bvar_.ndraws*0.05));
% plot LP
plot_irfs_(lpirf2plot,options)
```

Figure 17 reports the responses. Relative to the Cholesky VAR responses (figure 1), bands are wider for industrial production and for the excess bond premium. However, the dynamic profile of the median is comparable.

Figure 17: Local projection responses to a monetary policy shock identified with Cholesky restrictions. Dark gray bands report 90% confidence intervals. Blue tick lines report the VAR IRF 90% confidence set.



Example 23 (LP: MP shock with IV identification) *Consider the same setting of example 10. We now identify a monetary policy shock with external instruments and trace the dynamic transmission of the shock using LP. We consider the same set of endogenous variables as in the VAR setting, i.e.* $\mathbf{x}_{t-1}$ *includes lagged values of the log of industrial production index (*`logip`*), the log of CPI (*`logcpi`*), the one-year government bond rate (*`gs1`*) and the excess bond premium (*`ebp`*)*

```
%       load the instruments from GK data
[numi,txti,rawi] = xlsread('factor_data.csv','factor_data');
% NOTE: instrument must have the same length as the observed data
options.proxy  = nan(length(y),1);
% instruments and data end in 2012m6
options.proxy(length(T)- length(numi)+1:end) = numi(:,4);

dm2 = directmethods(y,lags,options);

% use previous saving settings
options0= options;
% figure tag
options0.saveas_strng  = 'IV';
% one plot per figure
options0.nplots = [1 1];
% variables names
options0.varnames = {'IP','CPI','1 year rate','EBP'};
% increase the title size
options0.fontsize =18;
% normalize to have a 25 bps increase in the one year govt bond
norm = dm2.irproxy_lp(3,1,1,2)*4;
% the plotting command
plot_irfs_(dm2.irproxy_lp(:,:,1,:)/norm,options0)
```

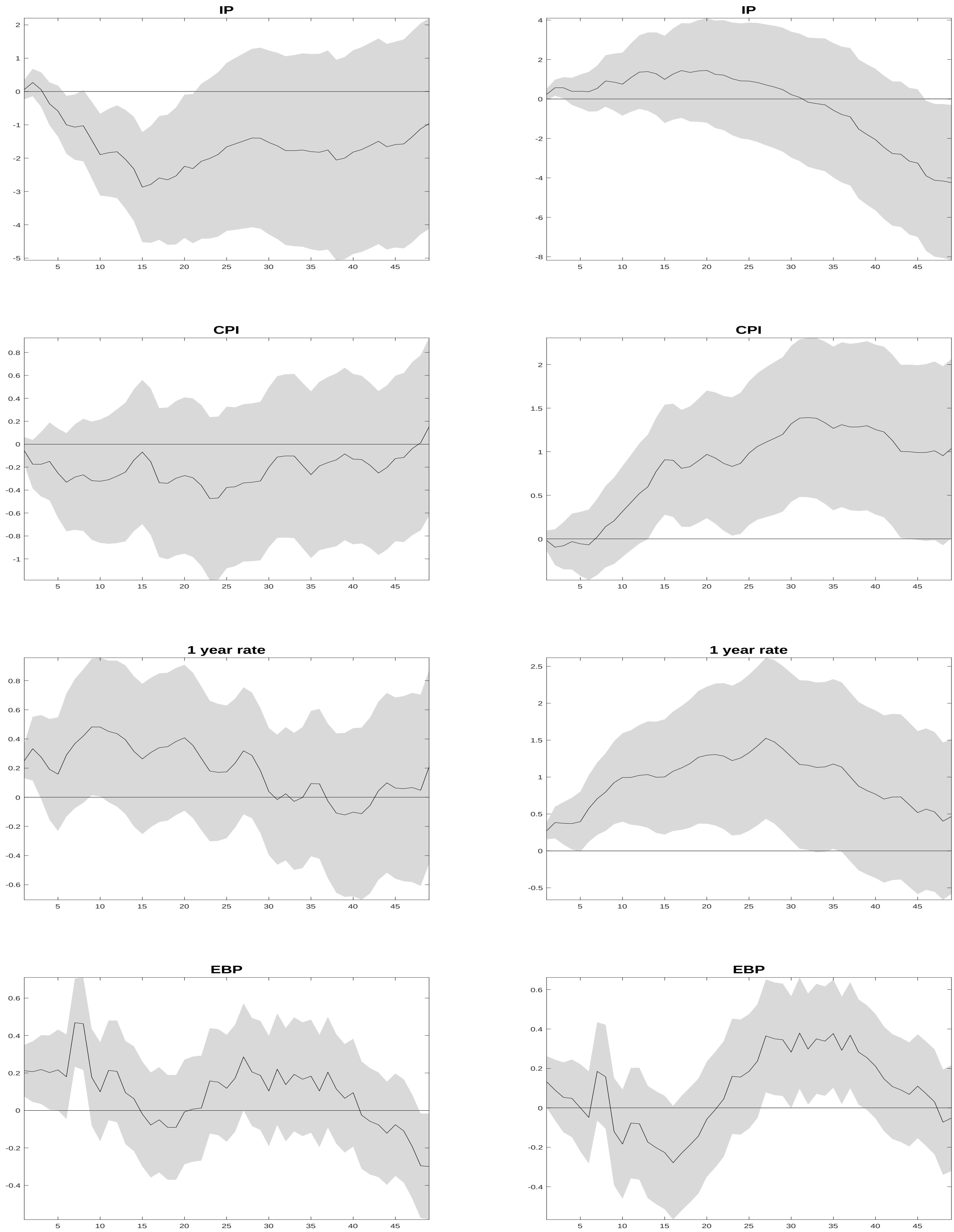
The left plots of figure 18 reports the responses. Relative to the IV VAR responses (figure 4), bands are much larger; especially for the interest rate and CPI. The uncertainty is so large that little can be said about the transmission properties of monetary policy disturbances in this setting.

Example 24 (LP: MP shock with IV identification with one endogenous variable at time) *Consider the same setting of example 23. We now construct the linear projection impulse response function using one endogenous variables at time, e.g. when computing the impulse response of the log of industrial production index (*`logip`*) we only include lagged values of* `logip`. *The right plots of figure 18 reports the responses and the commands to produce the plot are reported below.*

In this case not only the uncertainty is wide but also transmission dynamics are quite different. This shows that unless one has the information set correctly specified, the responses obtained are biased becasue the instrument ends up being correlated with the errors.

```
% use previous saving settings
options1 = options0;
% iterate over endogenous var
for vv = 1 :size(y,2)
    % consider only one endogenous variable at time
    dm3 = directmethods (y(:,vv),lags,options);
    % name of the file to save
    options1.saveas_strng  = ['IV_var' num2str(vv)];
    % variable name
    options1.varnames = options0.varnames(vv);
    % the plotting command
    % NOTE: we use the same normalization as in the previous example.
    plot_irfs_(dm3.irproxy_lp(:,:,1,:)/norm,options1)
end
```

Figure 18: Local projection responses to a monetary policy shock identified with IV restrictions. Dark gray bands report 90% confidence intervals. Left panels: specification of example 23. Right panels: specification of example 25.



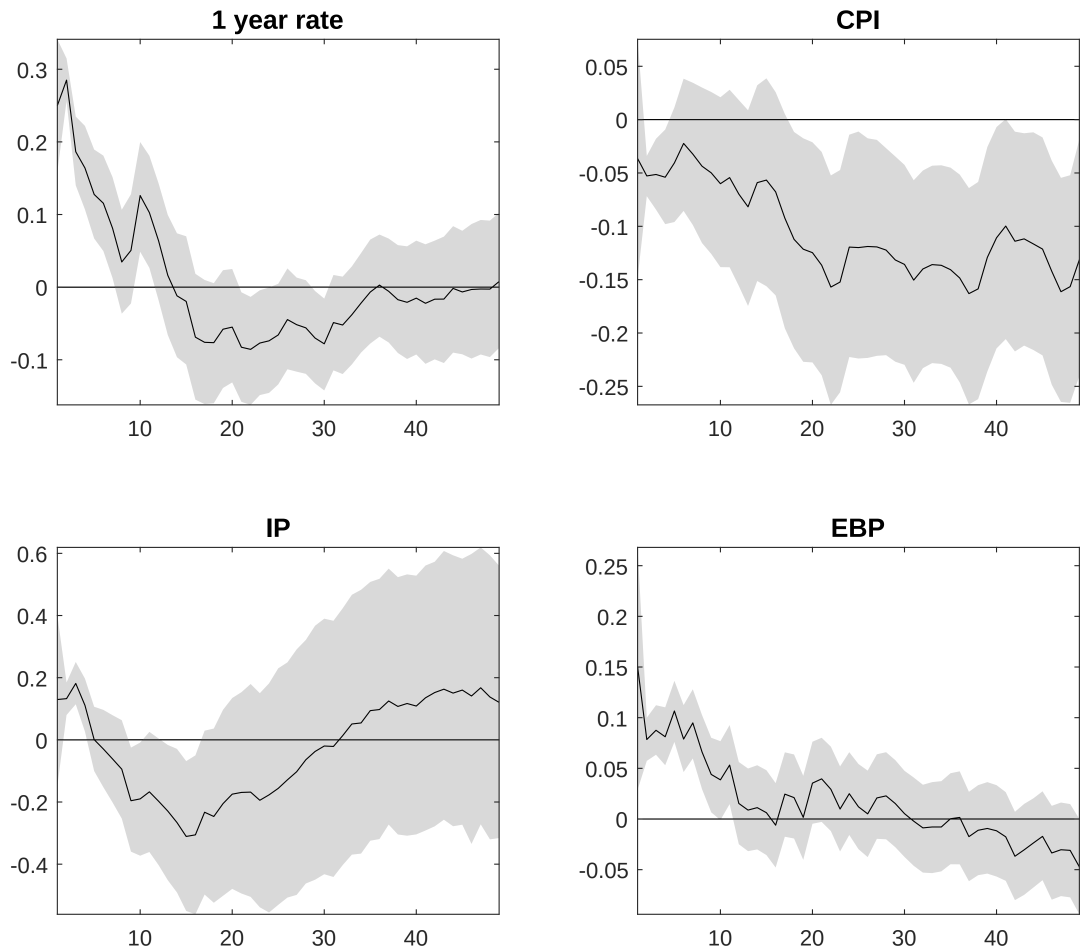
It may be interesting to see whether Bayesian methods allow a more precise characterization when IV are used. The next example considers this setup.

Example 25 (Bayesian LP with VAR priors) *Consider the same setting in 23. We use the first eight years of data to calculate the posterior distribution of the VAR reduced form parameters, and use the posterior means to center the priors for the LP parameters and to compute the appropriate response functions. The commands are as follows;*

```
%     run a VAR on a presample of data
presample = 96; % 8 years of presample
lags      = 12;
bvar_     = bvar(y(1:presample,:),lags);

% use the VAR estimates to set the priors for LP
options.priors.name        = 'Conjugate';
% use the posterior mean of the VAR coefficients
options.priors.Phi.mean    = mean(bvar_.Phi_draws,3);
% use the posterior variance of the VAR coefficients
options.priors.Phi.cov     = diag(mean(var(bvar_.Phi_draws,0,3),2));
% use the posterior mean of the covariance of the VAR residuals
options.priors.Sigma.scale = mean(bvar_.Sigma_draws,3);
options.priors.Sigma.df    = size(bvar_.Phi_draws,1)-2;
% overall shrinkage (<1 looser prior)
options.priors.tau         = 0.5*ones(options.hor); %

% adjust the length of the IV to match the data size
options.proxy(1:presample,:) =[];

% compute the Bayesian DM
bdm = directmethods(y(presample+1:end,:),lags,options);

% save figure with name
options.saveas_strng  = 'BLPIV';
% credible sets
options.conf_sig   = 0.90;
% normalize to have a 25 bps increase in the one year govt bond
norm = median(bdm.irproxy_blp(3,1,1,:),4)*4;
% plot command
plot_irfs_(bdm.irproxy_blp(indx_var,:,1,:)/norm,options)
```

Figure 19 reports the responses. Relative to figure 18, bands are narrower and inference more *precise.*

Figure 19: Bayesian local projection responses to a monetary policy shock identified with IV restrictions. Dark (light) gray bands 90% credible sets.



References

Adjemian, S., Bastani, H., Juillard, M., Karamé, F., Maih, J., Mihoubi, F., Perendia, G., Pfeifer, J., Ratto, M. and Villemot, S.: 2011, Dynare: Reference manual version 4, *Dynare Working Papers 1*, CEPREMAP.

Angeletos, G.-M., Collard, F. and Dellas, H.: 2018, Business Cycle Anatomy, *NBER Working Papers 24875*, National Bureau of Economic Research, Inc.

Antolin-Diaz, J. and Rubio-Ramírez, J. F.: 2016, Narrative Sign Restrictions for SVARs, *CEPR Discussion Papers 11517*, C.E.P.R. Discussion Papers.

Arias, J. E., Rubio-Ramírez, J. F. and Waggoner, D. F.: 2018, Inference Based on SVARs Identified with Sign and Zero Restrictions: Theory and Applications, *Econometrica* **86**, 685–720.

Baumeister, C. and Hamilton, J. D.: 2015, Sign Restrictions, Structural Vector Autoregressions, and Useful Prior Information, *Econometrica* **83**(5), 1963–1999.

Binning, A.: 2013, Underidentified SVAR models: A framework for combining short and long-run restrictions with sign-restrictions, *Working Paper 2013/14*, Norges Bank.

Blanchard, O. J. and Quah, D.: 1989, The Dynamic Effects of Aggregate Demand and Supply Disturbances, *American Economic Review* **79**(4), 655–673.

Boivin, J., Kiley, M. T. and Mishkin, F. S.: 2010, How Has the Monetary Transmission Mechanism Evolved Over Time?, *in* B. M. Friedman and M. Woodford (eds), *Handbook of Monetary Economics*, Vol. 3 of *Handbook of Monetary Economics*, Elsevier, chapter 8, pp. 369–422.

Brave, S. A., Butters, R. and Kelley, D.: 2020, A Practitioner's Guide and MATLAB Toolbox for Mixed Frequency State Space Models, *Mimeo*.

Canova, F.: 2007, Bayesian Time Series and DSGE Models, from Methods for Applied Macroeconomic Research, *Methods for Applied Macroeconomic Research*, Introductory Chapters, Princeton University Press.

Canova, F. and de Nicolo, G.: 2003, On the sources of business cycles in the G-7, *Journal of International Economics* **59**(1), 77–100.

Canova, F. and Ferroni, F.: 2019, Mind the gap! Stylized dynamic facts and structural models, *CEPR Discussion Papers 13948*, C.E.P.R. Discussion Papers.

Canova, F. and Pappa, E.: 2011, Fiscal policy, pricing frictions and monetary accommodation, *Economic Policy* **26**(68), 555–598.

Canova, F. and Paustian, M.: 2011, Business cycle measurement with some theory, *Journal of Monetary Economics* **58**(4), 345–361.

Canova, F. and Perez-Forero, F. J.: 2015, Estimating overidentified, nonrecursive, time varying coefficients structural vector autoregressions, *Quantitative Economics* **6**(2), 359–384.

Del Negro, M. and Schorfheide, F.: 2011, Bayesian Macroeconometrics, *in* J. Geweke, G. Koop and H. van Dijk (eds), *The Oxford Handbook of Bayesian Econometrics*, Oxford University Press.

Del Negro, M. and Schorfheide, F.: 2013, DSGE Model-Based Forecasting, *in* G. Elliott, C. Granger and A. Timmermann (eds), *Handbook of Economic Forecasting*, Vol. 2 of *Handbook of Economic Forecasting*, Elsevier, chapter 0, pp. 57–140.

Durbin, J. and Koopman, S. J.: 2001, *Time Series Analysis by State Space Methods*, number 9780198523543 in *OUP Catalogue*, Oxford University Press.

Fisher, J. D. M.: 2006, The Dynamic Effects of Neutral and Investment-Specific Technology Shocks, *Journal of Political Economy* **114**(3), 413–451.

Gali, J.: 1999, Technology, Employment, and the Business Cycle: Do Technology Shocks Explain Aggregate Fluctuations?, *American Economic Review* **89**(1), 249–271.

Gertler, M. and Karadi, P.: 2015, Monetary Policy Surprises, Credit Costs, and Economic Activity, *American Economic Journal: Macroeconomics* **7**(1), 44–76.

Giannone, D., Lenza, M. and Primiceri, G. E.: 2015, Prior Selection for Vector Autoregressions, *The Review of Economics and Statistics* **97**(2), 436–451.

Gourieroux, C., Monfort, A. and Renne, J.-P.: Forthcoming, Identification and Estimation in Non-Fundamental Structural VARMA Models, *Review Economic Studies* .

Hamilton, J. D.: 2007, *Time series analysis*, Princeton University Press.

Jordà, O.: 2005, Estimation and Inference of Impulse Responses by Local Projections, *American Economic Review* **95**(1), 161–182.

Maih, J.: 2010, Conditional forecasts in DSGE models, *Working Paper 2010/07*, Norges Bank.

Marcellino, M., Stock, J. H. and Watson, M. W.: 2006, A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series, *Journal of Econometrics* **135**(1), 499 – 526.

Mertens, K. and Ravn, M. O.: 2013, The Dynamic Effects of Personal and Corporate Income Tax Changes in the United States, *American Economic Review* **103**(4), 1212–47.

Miranda-Agrippino, S. and Ricco, G.: 2017, The Transmission of Monetary Policy Shocks, *Working Paper Series 657*, Bank of England.

Petrova, K.: 2019, A quasi-Bayesian local likelihood approach to time varying parameter VAR models, *Journal of Econometrics* **212**(1), 286–306.

Plagborg-Møller, M. and Montiel-Olea, J. L.: 2020, Local Projection Inference is Simpler and More Robust Than You Think, *Working paper*, Princeton University OpenScholar.

Plagborg-Møller, M. and Wolf, C. K.: 2019, Local Projections and VARs Estimate the Same Impulse Responses, *Working paper*, Princeton University OpenScholar.

Primiceri, G. E.: 2005, Time Varying Structural Vector Autoregressions and Monetary Policy, *Review of Economic Studies* **72**(3), 821–852.

Ramey, V. A.: 2016, Macroeconomic Shocks and Their Propagation, *in* J. B. Taylor and H. Uhlig (eds), *Handbook of Macroeconomics*, Vol. 2A, North Holland, chapter 3, pp. 71–162.

Rubio-Ramírez, J. F., Waggoner, D. F. and Zha, T.: 2010, Structural Vector Autoregressions: Theory of Identification and Algorithms for Inference, *Review of Economic Studies* **77**(2), 665–696.

Sa, F., Towbin, P. and Wieladek, T.: 2014, Capital Inflows, Financial Structure And Housing Booms, *Journal of the European Economic Association* **12**(2), 522–546.

Schorfheide, F. and Song, D.: 2015, Real-Time Forecasting With a Mixed-Frequency VAR, *Journal of Business & Economic Statistics* **33**(3), 366–380.

Sims, C. A. and Zha, T.: 1998, Bayesian Methods for Dynamic Multivariate Models, *International Economic Review* **39**(4), 949–968.

Stock, J. H. and Watson, M. W.: 2012, Disentangling the Channels of the 2007-09 Recession, *Brookings Papers on Economic Activity* (Spring), 81–156.

Uhlig, H.: 2005, What are the effects of monetary policy on output? Results from an agnostic identification procedure, *Journal of Monetary Economics* **52**(2), 381–419.

Villemot, S. and Pfeifer, J.: 2017, Bvar models "à la sims" in dynare, *Technical report*, Mimeo.

Waggoner, D. F. and Zha, T.: 1999, Conditional Forecasts In Dynamic Multivariate Models, *The Review of Economics and Statistics* **81**(4), 639–651.

Wolf, C.: forthcoming, SVAR (Mis-)Identification and the Real Effects of Monetary Policy Shocks, *American Economic Journal: Macroeconomics* .

Zellner, A.: 1971, *An introduction to Bayesian inference in econometrics*, Wiley-Interscience.

A Appendix

This appendix provides some technical details about the construction of posterior distributions and the algorithms used to implement different identification restrictions. More details on the construction of the posterior distributions can be found in Canova (2007) or Zellner (1971). The part on the Minnesota prior closely follows Del Negro and Schorfheide (2011).

A.1 Notation

Let a $VAR(p)$ be:

$$y_t = \Phi_1 y_{t-1} + ... + \Phi_p y_{t-p} + \Phi_0 + u_t$$

where y_t is $n \times 1$ vector and Φ_j are $n \times n$ matrices; u_t are i.i.d. normal distributed, zero mean, random vectors with covariance matrix Σ. We assume $y_0, \ldots, y_{-p+1}$ are fixed. Let $\vartheta = vec(\Phi_1, ..., \Phi_p, \Phi_0, \Sigma)$ be the vector of parameters to be estimated.

As we have seen a $VAR(p)$ can be transformed into a companion $VAR(1)$. For inferential purposes, however, it is useful to rewrite it in a seemingly unrelated regression (SUR) format. Let $k = np + 1$, we have

$$\underbrace{Y}_{T\times n} = \underbrace{X}_{T\times k}\underbrace{\Phi}_{k\times n} + \underbrace{E}_{T\times n}$$

$$Y = \begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_T' \end{pmatrix} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,n} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,n} \\ \vdots & & & \\ y_{T,1} & y_{T,2} & \cdots & y_{T,n} \end{pmatrix} \quad X = \begin{pmatrix} \mathbf{x}_0' & 1 \\ \mathbf{x}_1' & 1 \\ \vdots & \\ \mathbf{x}_{T-1}' & 1 \end{pmatrix} \quad \underset{(np\times 1)}{\mathbf{x}_t} = \begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+1} \end{pmatrix}$$

$$\Phi = \begin{pmatrix} \Phi_1' \\ \vdots \\ \Phi_p' \\ \Phi_0' \end{pmatrix} \quad E = \begin{pmatrix} u_1' \\ \vdots \\ u_T' \end{pmatrix}$$

A.1.1 From the VAR to the VMA

The $VAR(p)$ can be written in VMA form as follows

$$y_t = u_t + \Psi_1 u_{t-1} + ... + \Psi_t u_1 + \overline{\Psi}_t$$

where Ψ_j for $j = 1, ..., t$ are functions of $(\Phi_1, ..., \Phi_p)$ and $\overline{\Psi}_t$ is a deterministic function of the constant matrix Φ_0, of the autoregressive matrices $(\Phi_1, ..., \Phi_p)$ and of the initial conditions

$(y_0, y_{-1}, ..., y_{-p})$. The mapping between the VAR and MA coefficients is given by

$$\begin{aligned}
\Psi_0 &= I \\
\Psi_1 &= \Phi_1 \\
\Psi_2 &= \Psi_1\Phi_1 + \Psi_0\Phi_2 \\
\Psi_3 &= \Psi_2\Phi_1 + \Psi_1\Phi_2 + \Psi_0\Phi_3 \\
&\vdots \\
\Psi_j &= \sum_{i=1}^{j} \Psi_{j-i}\Phi_i \quad \text{ for } j < p \\
\Psi_j &= \Psi_{j-1}\Phi_1 + \Psi_{j-2}\Phi_2 + \cdots + \Psi_{j-p}\Phi_p \quad \text{ for } j \geq p
\end{aligned}$$

A.2 Classical Inference

By constructions the innovations $u_t = y_t - E(y_t|y_{t-1}, y_{t-2}, ...y_0)$, are orthogonal to the past values of the endogenous variables. Thus, $E(u_t|y_{t-1}, y_{t-2}..., y_0) = 0$ and the OLS estimator of the VAR parameters

$$\widehat{\Phi} = (X'X)^{-1}X'Y$$

is consistent. The estimator of the covariance matrix of the shocks is

$$S_{ols} = 1/(T - (np + 1))(Y - X\widehat{\Phi})'(Y - X\widehat{\Phi})$$

Under (asymptotic) normality of the shocks, the distribution for the OLS estimator is:

$$\widehat{\Phi} \sim N(\Phi, (X'X)^{-1} \otimes S_{ols})$$

where $\otimes$ is the Kronecker product.

A.3 Posterior Distributions with Jeffrey's prior

When E is a multivariate normal, i.e. $p(E|0, I_T, \Sigma)$ has the format given in A.10, $Y - X\Phi$ is also multivariate normal and the conditional likelihood has the following expression

$$p(Y|X, \Phi, \Sigma) = (2\pi)^{-Tn/2} |\Sigma|^{-T/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}(Y - X\Phi)'(Y - X\Phi)\right]\right\}$$

The conditional likelihood function can be conveniently expressed as (see Zellner (1971))

$$\begin{aligned}
p(Y|X, \Phi, \Sigma) = (2\pi)^{-Tn/2} |\Sigma|^{-T/2} & \qquad (6) \\
\exp\left(-1/2tr(\Sigma^{-1}\widehat{S})\right) \times \exp\left(-1/2tr(\Sigma^{-1}(\Phi - \widehat{\Phi})'X'X(\Phi - \widehat{\Phi}))\right) & \\
= p(\Sigma|Y, X)p(\Phi|\Sigma, Y, X) & \qquad (7)
\end{aligned}$$

where $\widehat{\Phi} = (X'X)^{-1}X'Y$ and $\widehat{S} = (Y - X\widehat{\Phi})'(Y - X\widehat{\Phi})$ and we used the following result:

$$\begin{aligned}
(Y - X\Phi)'(Y - X\Phi) &= (Y - X\widehat{\Phi} + X\widehat{\Phi} - X\Phi)'(Y - X\widehat{\Phi} + X\widehat{\Phi} - X\Phi) \\
&= \widehat{S} + (\widehat{\Phi} - \Phi)X'X(\widehat{\Phi} - \Phi)
\end{aligned}$$

The likelihood in (7) contains the kernel of two known distributions: the multivariate normal and the inverse Wishart. If we combine the conditional likelihood with Jeffrey's prior $p(\Phi, \Sigma) = |\Sigma|^{-(n+1)/2}$, we obtain a family posterior distribution for $(\Phi, \Sigma|Y)$[13]:

$$
\begin{aligned}
p(\Phi, \Sigma|Y) &= p(Y|X, \Phi, \Sigma) \times p(\Phi, \Sigma) \\
&= \underbrace{|\Sigma|^{-k/2}|X'X|^{n/2} \exp\left(-1/2tr(\Sigma^{-1}(\Phi - \widehat{\Phi})'X'X(\Phi - \widehat{\Phi}))\right)}_{\text{Kernel of } N(\widehat{\Phi}, \Sigma, (X'X)^{-1})} \\
&\times (2\pi)^{-Tn/2}\,|X'X|^{-n/2} \\
&\times \underbrace{|\Sigma|^{-(n+1)/2}|\Sigma|^{-T/2}|\Sigma|^{k/2} \exp\left(-1/2tr(\Sigma^{-1}\widehat{S})\right)}_{\text{Kernel of } IW(\widehat{S}, T-k+n+1)}
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\Phi|\Sigma, X, Y &\sim N(\widehat{\Phi}, \Sigma \otimes (X'X)^{-1}) \\
\Sigma|X, Y &\sim IW(\widehat{S}, T - k + n + 1)
\end{aligned}
$$

Thus, draws can be obtained as follows:
1. Draw Σ^j from $IW(\widehat{S}, T - k + n + 1)$
2. Conditional on Σ^j, draw Φ^j from $N(\widehat{\Phi}, \Sigma^j \otimes (X'X)^{-1})$

A.4 Posterior Distribution with Conjugate Priors

Assume a multivariate normal-inverse Wishart (MN-IW) prior:

$$
\begin{aligned}
\Phi \sim N(\Phi_0, \Sigma \otimes V) \quad &= (2\pi)^{-nk/2}|\Sigma|^{-k/2}|V|^{-n/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}(\Phi - \Phi_0)'V^{-1}(\Phi - \Phi_0)\right]\right\} \\
\Sigma \sim IW(\Sigma_0, d) \quad &= \frac{|\Sigma_0|^{d/2}}{2^{dn/2}\Gamma_n(d/2)}|\Sigma|^{-(n+d+1)/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma_0\Sigma^{-1}\right]\right\}
\end{aligned}
$$

It can be shown that the posterior distribution is also MN-IW:

$$
\begin{aligned}
\Phi|\Sigma, Y, X, &\sim N(\overline{\Phi}, \Sigma \otimes (X'X + V^{-1})^{-1}) \\
\Sigma|Y, X &\sim IW(\overline{S}, T + d)
\end{aligned}
$$

where

$$
\begin{aligned}
\overline{\Phi} &= (X'X + V^{-1})^{-1}(X'Y + V^{-1}\Phi_0) \\
\overline{S} &= \Sigma_0 + \widehat{S} + \Phi_0'V^{-1}\Phi_0 + \widehat{\Phi}'X'X\widehat{\Phi} - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi} \\
&= \Sigma_0 + (Y - X\overline{\Phi})'(Y - X\overline{\Phi}) + (\overline{\Phi} - \Phi_0)'V^{-1}(\overline{\Phi} - \Phi_0)
\end{aligned}
$$

It is also easy to draws from this joint posterior. The process is the same as with Jeffrey's prior, i.e.
1. Draw Σ^j from $IW(\overline{S}, T + d)$
2. Conditional on Σ^j, draw Φ^j from $N(\overline{\Phi}, \Sigma^j \otimes (X'X)^{-1})$

[13] See also Zellner (1971) equations (8.14) and (8.15) at page 227.

A.4.1 Derivation the MN-IW Conjugate result

The posterior distribution can be conveniently expressed as

$$
\begin{aligned}
&p(\Sigma, \Phi|Y) \propto p(Y|X,\Phi,\Sigma)\ p(\Phi|\Sigma)\ p(\Sigma) = \\
&\qquad (2\pi)^{-Tn/2}\,|\Sigma|^{-T/2} \exp\left(-1/2tr(\Sigma^{-1}\widehat{S})\right) \exp\left(-1/2tr(\Sigma^{-1}(\Phi-\widehat{\Phi})'X'X(\Phi-\widehat{\Phi}))\right) \qquad \text{[Likelihood]} \\
&\qquad (2\pi)^{-nk/2}|\Sigma|^{-k/2}|V|^{-n/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}(\Phi-\Phi_0)'V^{-1}(\Phi-\Phi_0)\right]\right\} \qquad \text{[Prior } \Phi\text{]} \\
&\qquad \frac{|\Sigma_0|^{d/2}}{2^{dn/2}\Gamma_n(d/2)}|\Sigma|^{-(n+d+1)/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma_0\Sigma^{-1}\right]\right\} \qquad \text{[Prior } \Sigma\text{]}
\end{aligned}
$$

Rearranging terms

$$
\begin{aligned}
p(\Sigma, \Phi|Y) \propto\ & (2\pi)^{-(T-k)n/2}\,\frac{|\Sigma_0|^{d/2}}{2^{dn/2}\Gamma_n(d/2)}|V|^{-n/2}\times \\
& |\Sigma|^{-T/2}|\Sigma|^{-k/2}|\Sigma|^{-(n+d+1)/2}\times \\
& \exp\left\{-\frac{1}{2}tr\left[(\widehat{S}+\Sigma_0)\Sigma^{-1}\right]\right\}\times \\
& \exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}\left(\underbrace{(\Phi-\widehat{\Phi})'X'X(\Phi-\widehat{\Phi})+(\Phi-\Phi_0)'V^{-1}(\Phi-\Phi_0)}_{A}\right)\right]\right\}
\end{aligned}
$$

Focus on the term A,and performing a number of algebraic manipulations:

$$
\begin{aligned}
&(\Phi-\widehat{\Phi})'X'X(\Phi-\widehat{\Phi})+(\Phi-\widehat{\Phi}+\widehat{\Phi}-\Phi_0)'V^{-1}(\Phi-\widehat{\Phi}+\widehat{\Phi}-\Phi_0) \\
&=(\Phi-\widehat{\Phi})'\left[X'X(\Phi-\widehat{\Phi})+V^{-1}(\Phi-\widehat{\Phi})+V^{-1}(\widehat{\Phi}-\Phi_0)\right]+\underbrace{(\widehat{\Phi}-\Phi_0)'V^{-1}(\Phi-\widehat{\Phi})}_{A_1}+\underbrace{(\widehat{\Phi}-\Phi_0)'V^{-1}(\widehat{\Phi}-\Phi_0)}_{A_2} \\
&=(\Phi-\widehat{\Phi})'\left[X'X\Phi-X'X\widehat{\Phi}+V^{-1}\Phi-V^{-1}\widehat{\Phi}+V^{-1}\widehat{\Phi}-V^{-1}\Phi_0)\right]+A_1+A_2 \\
&=(\Phi-\widehat{\Phi})'\left[(X'X+V^{-1})\Phi-X'X\widehat{\Phi}-V^{-1}\Phi_0\right]+A_1+A_2 \\
&=(\Phi-\widehat{\Phi})'(X'X+V^{-1})\left[\Phi-\underbrace{(X'X+V^{-1})^{-1}(X'X\widehat{\Phi}+V^{-1}\Phi_0)}_{\overline{\Phi}}\right]+A_1+A_2 \\
&=(\Phi-\widehat{\Phi})'(X'X+V^{-1})\left[\Phi-\overline{\Phi}\right]+A_1+A_2 \\
&=(\Phi-\overline{\Phi}+\overline{\Phi}-\widehat{\Phi})'(X'X+V^{-1})\left[\Phi-\overline{\Phi}\right]+A_1+A_2 \\
&=\underbrace{(\Phi-\overline{\Phi})'(X'X+V^{-1})\left[\Phi-\overline{\Phi}\right]}_{A_0}+(\overline{\Phi}-\widehat{\Phi})'(X'X+V^{-1})\left[\Phi-\overline{\Phi}\right]+A_1+A_2 \\
&=A_0+(\overline{\Phi}-\widehat{\Phi})'(X'X+V^{-1})(\Phi-\overline{\Phi})+(\widehat{\Phi}-\Phi_0)'V^{-1}(\Phi-\widehat{\Phi})+A_2 \\
&=A_0+(\overline{\Phi}-\widehat{\Phi})'(X'X+V^{-1})\Phi-(\overline{\Phi}-\widehat{\Phi})'(X'X+V^{-1})\overline{\Phi}+(\widehat{\Phi}-\Phi_0)'V^{-1}\Phi-(\widehat{\Phi}-\Phi_0)'V^{-1}\widehat{\Phi}+A_2 \\
&=A_0+\left[\underbrace{(\overline{\Phi}-\widehat{\Phi})'(X'X+V^{-1})+(\widehat{\Phi}-\Phi_0)'V^{-1}}_{A_3}\right]\Phi-\underbrace{\left[(\overline{\Phi}-\widehat{\Phi})'(X'X+V^{-1})\overline{\Phi}+(\widehat{\Phi}-\Phi_0)'V^{-1}\widehat{\Phi}\right]}_{A_4}+A_2
\end{aligned}
$$

It is easy to show that $A_3 = 0$

$$
\begin{aligned}
A_3 &= \overline{\Phi}'(X'X + V^{-1}) - \widehat{\Phi}'(X'X + V^{-1}) + \widehat{\Phi}'V^{-1} - \Phi_0'V^{-1} \\
&= (X'X\widehat{\Phi} + V^{-1}\Phi_0)'(X'X + V^{-1})^{-1}(X'X + V^{-1}) - \widehat{\Phi}'(X'X + V^{-1}) + \widehat{\Phi}'V^{-1} - \Phi_0'V^{-1} \\
&= \widehat{\Phi}'X'X + \Phi_0'V^{-1} - \widehat{\Phi}'X'X - \widehat{\Phi}'V^{-1} + \widehat{\Phi}'V^{-1} - \Phi_0'V^{-1} \\
&= 0
\end{aligned}
$$

Moreover, $A_2 - A_4$ does not depend on Φ and can be rearranged as

$$
\begin{aligned}
A_2 - A_4 = & \\
&= (\widehat{\Phi} - \Phi_0)'V^{-1}(\widehat{\Phi} - \Phi_0) - (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1})\overline{\Phi} - (\widehat{\Phi} - \Phi_0)'V^{-1}\widehat{\Phi} \\
&= -(\widehat{\Phi} - \Phi_0)'V^{-1}\Phi_0 - (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1})\overline{\Phi} \\
&= -\widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi} + \widehat{\Phi}'(X'X + V^{-1})\overline{\Phi} \\
&= \widehat{\Phi}'(X'X + V^{-1})\overline{\Phi} - \widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi} \\
&= \widehat{\Phi}'(X'X + V^{-1})(X'X + V^{-1})^{-1}(X'X\widehat{\Phi} + V^{-1}\Phi_0) - \widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi} \\
&= \widehat{\Phi}'X'X\widehat{\Phi} + \widehat{\Phi}'V^{-1}\Phi_0 - \widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi} \\
&= \widehat{\Phi}'X'X\widehat{\Phi} + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi}
\end{aligned}
$$

The posterior distribution is then

$$
\begin{aligned}
& p(\Sigma, \Phi|Y) = \text{const} \times |\Sigma|^{-(T+k+n+d+1)/2} \times \\
& \exp\left\{-\frac{1}{2}tr\left[\left(\widehat{S} + \Sigma_0 + \widehat{\Phi}'X'X\widehat{\Phi} + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi}\right)\Sigma^{-1}\right]\right\} \qquad [p(\Sigma|Y) \text{ kernel of a IW}] \\
& \exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}(\Phi - \overline{\Phi})'(X'X + V^{-1})(\Phi - \overline{\Phi})\right]\right\} \qquad [p(\Phi|Y, \Sigma) \text{ kernel of a MN}]
\end{aligned}
$$

The marginal likelihood can be derived analytically by taking the integral of the latter with respect to the reduced form VAR parameters. Taking the integral with respect to Φ first, and to Σ second, and using $tr(A'B) = tr(AB') = tr(B'A) = tr(BA')$, we can compute the marginal likelihood

$$p(Y|X) = \int_\Sigma \int_\Phi p(\Sigma, \Phi|Y) d\Sigma d\Phi =$$

$$= \int_\Sigma \int_\Phi \text{const}|\Sigma|^{-(T+k+n+d+1)/2} \exp\left\{-\frac{1}{2} tr\left[\overline{S}\Sigma^{-1}\right]\right\} \exp\left\{-\frac{1}{2} tr\left[\Sigma^{-1}(\Phi-\overline{\Phi})'(X'X+V^{-1})(\Phi-\overline{\Phi})\right]\right\} d\Sigma d\Phi$$

$$= \int_\Sigma \underbrace{\left[\int_\Phi (2\pi)^{-kn/2}\,|\Sigma|^{-k/2}|X'X+V^{-1}|^{n/2} \exp\left\{-\frac{1}{2} tr\left[\Sigma^{-1}(\Phi-\overline{\Phi})'(X'X+V^{-1})(\Phi-\overline{\Phi})\right]\right\} d\Phi\right]}_{=1}$$

$$(2\pi)^{-Tn/2}\,\frac{|\Sigma_0|^{d/2}}{2^{dn/2}\Gamma_n(d/2)}|V|^{-n/2}|X'X+V^{-1}|^{-n/2}|\Sigma|^{-(T+n+d+1)/2} \exp\left\{-\frac{1}{2} tr\left[\overline{S}\Sigma^{-1}\right]\right\} d\Sigma$$

$$= (2\pi)^{-Tn/2}\,\frac{|\Sigma_0|^{d/2}}{2^{dn/2}\Gamma_n(d/2)}|V|^{-n/2}|X'X+V^{-1}|^{-n/2}\frac{2^{n(T+d)/2}\Gamma_n(\frac{T+d}{2})}{|\overline{S}|^{(T+d)/2}}$$

$$\underbrace{\int_\Sigma \frac{|\overline{S}|^{(T+d)/2}}{2^{n(T+d)/2}\Gamma_n(\frac{T+d}{2})}|\Sigma|^{-(T+n+d+1)/2} \exp\left\{-\frac{1}{2} tr\left[\overline{S}\Sigma^{-1}\right]\right\} d\Sigma}_{=1}$$

$$= \pi^{-Tn/2}\;\frac{\Gamma_n(\frac{T+d}{2})}{\Gamma_n(\frac{d}{2})}\;|\Sigma_0|^{d/2}\;|V|^{-n/2}\;|X'X+V^{-1}|^{-n/2}\;|\overline{S}|^{-(T+d)/2}$$

Taking logs,

$$\begin{aligned}\ln p(Y|X) = &-(Tn/2)\ln\pi + \ln\Gamma_n\left(\frac{T+d}{2}\right) - \ln\Gamma_n\left(\frac{d}{2}\right) + (d/2)\ln|\Sigma_0| \\ &- (n/2)\ln|V| - (n/2)\ln|X'X+V^{-1}| - \frac{T+d}{2}\ln|\overline{S}|\end{aligned}$$

To avoid numerical issues, it is useful to employ $F_V F_V' = V$ and $F_0 F_0' = \Sigma_0^{-1}$, so that $|X'X+V^{-1}| = |V^{-1}||I+F_V'X'XF_V|$ and $|\overline{S}| = |\Sigma_0||I+F_0'[(Y-X\overline{\Phi})'(Y-X\overline{\Phi})+(\overline{\Phi}-\Phi_0)'V^{-1}(\overline{\Phi}-\Phi_0)]F_0|$. With these transformation, we can rewrite the log marginal likelihood as

$$\begin{aligned}\ln p(Y|X) = &-(Tn/2)\ln\pi + \ln\Gamma_n\left(\frac{T+d}{2}\right) - \ln\Gamma_n\left(\frac{d}{2}\right) - (T/2)\ln|\Sigma_0| - (n/2)\ln|I+F_V'X'XF_V| \\ &- \frac{T+d}{2}\ln|I+F_0'[(Y-X\overline{\Phi})'(Y-X\overline{\Phi})+(\overline{\Phi}-\Phi_0)'V^{-1}(\overline{\Phi}-\Phi_0)]F_0|\end{aligned}$$

A.5 Posterior Distributions with the Minnesota Prior

The Minnesota prior for Φ and Σ we use belongs to the MN-IW family of conjugate priors. Let $y_{-\tau:0}$ be a presample, and let $\underline{y}$ and $\underline{s}$ be the $n \times 1$ vectors of means and standard deviations. The first moment of the prior for Φ is

$$E(\phi_{ij,k} \mid \Sigma) = \begin{cases} 1 & \text{if } j = i \text{ and } k = 1; \\ 0 & \text{else.} \end{cases}$$

The second moment of the prior for Φ_1 is

$$cov(\phi_{ij,1}, \phi_{hg,1} \mid \Sigma) = \begin{cases} \Sigma_{ih}/(\tau \underline{s}_j)^2 & \text{if } g = j; \\ 0 & \text{else.} \end{cases}$$

Second moments of the prior for $\Phi_l, l = 2, 3, \ldots, p$ is

$$cov(\phi_{ij,\ell}, \phi_{hg,\ell} \mid \Sigma) = \begin{cases} \Sigma_{ih}/(\tau \underline{s_j} 2^d)^2 & \text{if } g = j; \\ 0 & \text{else.} \end{cases}$$

A popular way to introduce Minnesota prior restrictions is via dummy observations; below we describe in detail how to construct artificial observations from the Minnesota prior to impose the sum of coefficient and co-persistence restrictions. First, we describe the logic of the dummy observations prior.

Suppose T^* additional observations are available and they are collected into the matrices Y^* and X^*. We use the likelihood function of the VAR to relate these additional observations to the parameters (Φ, Σ). Up to a constant, the product $p(Y^*|X^*, \Phi, \Sigma) * |\Sigma|^{-(n+1)/2}$ can be interpreted as

$$\Phi, \Sigma | Y^*, X^* \sim MNIW\left(\underline{\Phi}, (X^{*'}X^*)^{-1}, \underline{S}, T^* - k\right)$$

provided that $T^* > k+n$, $X^{*'}X^*$ is invertible and the distribution is proper. Let $\overline{T} = T + T^*$, $\overline{Y} = [Y^{*'}, Y']'$ and $\overline{X} = [X^{*'}, X']'$ and $\overline{\Phi}, \overline{\Sigma}$ be the analog of $\widehat{\Phi}, \widehat{\Sigma}$. Combining the two sets of observations we have that

$$\Phi, \Sigma | \overline{Y} \sim MNIW\left(\overline{\Phi}, (\overline{X}'\overline{X})^{-1}, \overline{S}, \overline{T} - k\right)$$

In other words, if the additional (dummy) observations are used to construct prior restrictions, they constitute a family of conjugate priors.

We next illustrate how to generate artificial data, $(Y^* X^*)$, that respect the Minnesota prior assumptions; to simplify the notation suppose that $n = 2$ and $p = 2$. For $j = 1, 2$ let $\underline{y_j}$ be the presample mean of y_j and $\underline{s_j}$ be the presample standard deviation of y_j and $\underline{s_{12}}$ the presample covariance between y_1 and y_2.

Prior for Φ_1

We use dummy observations to generate a prior distribution for Φ_1.

$$\begin{pmatrix} \tau \underline{s_1} & 0 \\ 0 & \tau \underline{s_2} \end{pmatrix} = \begin{pmatrix} \tau \underline{s_1} & 0 & 0 & 0 & 0 \\ 0 & \tau \underline{s_2} & 0 & 0 & 0 \end{pmatrix} \Phi + \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}$$

The first row implies $\tau \underline{s_1} = \tau \underline{s_1} \phi_{11,1} + u_{11}$ and $0 = \tau \underline{s_1} \phi_{21,1} + u_{12}$. When u is normally distributed, we have

$$\phi_{11,1} \sim N(1, \Sigma_{11}/(\tau^2 \underline{s_1}^2))$$
$$\phi_{21,1} \sim N(0, \Sigma_{22}/(\tau^2 \underline{s_1}^2))$$

with covariance

$$E(\phi_{11,1}\phi_{21,1}) = E((1 - u_{11}/(\tau \underline{s_1}))(-u_{12}/(\tau \underline{s_1})) = \Sigma_{12}/(\tau^2 \underline{s_1}^2)$$
$$E(\phi_{11,1}\phi_{22,1}) = E(\phi_{11,1}\phi_{12,1}) = 0$$

where Σ_{ij} is the (i,j) element of Σ. A similar argument holds for $\phi_{12,1}$ and $\phi_{22,1}$. Note that the prior distribution on $\phi_{ij,1}$ implies unit root behavior for the series.

Prior for Φ_2

The dummy observations that generate a prior distribution for Φ_2, where d scales the coefficients associated with the lags, are

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \tau \underline{s_1} 2^d & 0 & 0 \\ 0 & 0 & 0 & \tau \underline{s_2} 2^d & 0 \end{pmatrix} \Phi + \begin{pmatrix} u_{31} & u_{32} \\ u_{41} & u_{42} \end{pmatrix}$$

which imply

$$\begin{aligned} \phi_{11,2} &\sim N(0, \Sigma_{11}/(\tau \underline{s_1} 2^d)^2) \\ \phi_{21,2} &\sim N(0, \Sigma_{22}/(\tau \underline{s_1} 2^d)^2) \\ E(\phi_{11,2}\phi_{21,2}) &= E((-u_{31}/(\tau \underline{s_1} 2^d))(-u_{32}/(\tau \underline{s_1} 2^d))) = \Sigma_{12}/(\tau \underline{s_1} 2^d)^2 \end{aligned}$$

Sum-of-coefficient restrictions

The dummy observations are:

$$\begin{pmatrix} \lambda \underline{y_1} & 0 \\ 0 & \lambda \underline{y_2} \end{pmatrix} = \begin{pmatrix} \lambda \underline{y_1} & 0 & \lambda \underline{y_1} & 0 & 0 \\ 0 & \lambda \underline{y_2} & 0 & \lambda \underline{y_2} & 0 \end{pmatrix} \Phi + U$$

Which imply:

$$\begin{aligned} (\phi_{11,1} + \phi_{11,2}) &\sim N(1, \Sigma_{11}/(\lambda \underline{y_1})^2) \\ (\phi_{21,1} + \phi_{21,2}) &\sim N(0, \Sigma_{22}/(\lambda \underline{y_1})^2) \\ E((\phi_{11,1} + \phi_{11,2})(\phi_{21,1} + \phi_{21,2})) &= \Sigma_{12}/(\lambda \underline{y_1})^2 \end{aligned}$$

and similarly for the second variable. The prior implies that when the lagged values of $y_{1,t}$ equal $\underline{y_1}$, then $\underline{y_1}$ is a good forecaster for $y_{1,t}$.

Co-persistence restriction

The next set of dummy observations provide a prior for the intercept

$$\begin{pmatrix} \mu \underline{y_1} & \mu \underline{y_2} \end{pmatrix} = \begin{pmatrix} \mu \underline{y_1} & \mu \underline{y_2} & \mu \underline{y_1} & \mu \underline{y_2} & \mu \end{pmatrix} \Phi + U$$

These restrictions imply

$$\begin{aligned} \phi_{1,0} &= \underline{y_1} - \underline{y_1}(\phi_{11,1} + \phi_{11,2}) - \underline{y_2}(\phi_{12,1} + \phi_{12,2}) - (1/\mu)u_{71} \\ \phi_{1,0} &\sim N(0, \Sigma_{11}(\lambda^{-2} + \mu^{-2}) + \Sigma_{22}\lambda^{-2}) \end{aligned}$$

The sum-of-coefficients prior does not imply cointegration. The co-persistence prior states that a no-change forecast for all variables is a good forecast at the beginning of the sample.

Prior for Σ

The prior distribution for Σ is centered at the matrix with elements equals to the pre-sample variance of y_t

$$\begin{pmatrix} \underline{s_1} & \underline{s_{12}} \\ \underline{s_{12}} & \underline{s_2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Psi + U$$

Assume that these observations are considered only once ($\omega = 1$). In general, the prior for the covariance matrix of the shocks is controlled by the hyper-parameter ω, which defines the number of times we replicate these observations.

To summarize, we have generated the 9 artificial observations for Y^* and X^*:

$$Y^* = \begin{pmatrix} \tau\underline{s_1} & 0 \\ 0 & \tau\underline{s_2} \\ 0 & 0 \\ 0 & 0 \\ \lambda\underline{y_1} & 0 \\ 0 & \lambda\underline{y_2} \\ \mu\underline{y_1} & \mu\underline{y_2} \\ \underline{s_1} & \underline{s_{12}} \\ \underline{s_{12}} & \underline{s_2} \end{pmatrix} \qquad X^* = \begin{pmatrix} \tau\underline{s_1} & 0 & 0 & 0 & 0 \\ 0 & \tau\underline{s_2} & 0 & 0 & 0 \\ 0 & 0 & \tau\underline{s_1}2^d & 0 & 0 \\ 0 & 0 & 0 & \tau\underline{s_2}2^d & 0 \\ \lambda\underline{y_1} & 0 & \lambda\underline{y_1} & 0 & 0 \\ 0 & \lambda\underline{y_2} & 0 & \lambda\underline{y_2} & 0 \\ \mu\underline{y_1} & \mu\underline{y_2} & \mu\underline{y_1} & \mu\underline{y_2} & \mu \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

In a general setting, the total number of dummy observations for the Minnesota prior is given by

$$T^* = n \times p + n \times \omega + 1 + n = n(p + \omega + 1) + 1$$

A.6 IV Identification

Let m_t be a proxy for the unobserved shock; to simplify the notation assume that it is a univariate variable. Let ν_t be the structural shock vector and u_t the innovations. Assume that the proxy m_t is linked to the first shock in ν_t, $\nu_{1,t}$, and uncorrelated with the remaining structural shocks, $\nu_{2,t}$. This leads to the following mapping

$$\begin{aligned} E(m_t \ \nu_t') &= [\rho, \mathbf{0}] \\ E(m_t \ \nu_t')\Omega' &= [\rho, \mathbf{0}] \ \Omega' \\ E(m_t \ u_t') &= [\rho \ \Omega_{1,1}, \rho \ \Omega_{2,1}'] \\ E(m_t \ u_{2,t}')E(m_t \ u_{1,t})^{-1} &= \Omega_{2,1}'\Omega_{1,1}^{-1} \end{aligned}$$

where $u_{1,t}$ is the innovation in the first equation of the VAR and $u_{2,t}$ is the vector containing the remaining innovation; $\Omega_{1,1}$ is the $(1,1)$-element in Ω and $\Omega_{2,1}$ is the $(n-1) \times 1$ vector containing all the remaining elements in the first column of Ω. Under stationarity and ergodicity, we have

$$\frac{1}{T}\sum m_t u_{2,t}' \left(\frac{1}{T}\sum m_t u_{1,t}\right)^{-1} \to E(m_t u_{2,t}')E(m_t u_{1,t})^{-1} = \Omega_{2,1}'\Omega_{1,1}^{-1} \qquad (8)$$

Notice that the left hand side is observable and converges in population to the first column of Ω. Let $\widehat{\rho}$ be the regression coefficient of $u_{1,t}$ on m_t. Then the LHS of equation (8) can be rearranged as:

$$
\begin{aligned}
&\frac{1}{T}\sum m_t u'_{2,t}\left(\frac{1}{T}\sum m_t^2\right)^{-1}\underbrace{\left(\frac{1}{T}\sum m_t^2\right)\left(\frac{1}{T}\sum m_t u_{1,t}\right)^{-1}}_{1/\widehat{\rho}}\\
&=\frac{1}{T}\sum m_t u'_{2,t}\left(\frac{1}{T}\sum m_t^2\right)^{-1}\frac{\widehat{\rho}}{\widehat{\rho}^2}\\
&=\frac{1}{T}\sum(\widehat{\rho}m_t)u'_{2,t}\left(\frac{1}{T}\sum(\widehat{\rho}m_t)^2\right)^{-1}\\
&=\frac{1}{T}\sum \widehat{u}_{1,t}u'_{2,t}\left(\frac{1}{T}\sum \widehat{u}_{1,t}^2\right)^{-1}
\end{aligned}
$$

The last expression defines a two stage IV (2SLS) regression, where we first regress u_1 on the proxy m. The fitted values $\widehat{u}_1$ represent the portion of the reduced form shock explained by the proxy (which is correlated with the structural shock). By regressing, u_2 on $\widehat{u}_1$ we obtain the impact of the proxy on other reduced form shocks and as a byproduct the contemporaneous impact on the VAR variables. Given sequence of VAR innovation, u_t, as in Mertens and Ravn (2013), we run the following regressions:

$$u_{1,t} = \rho_0 + \rho_1 m_t + e_{1,t} \tag{9}$$

$$u_{2,t} = b_0 + b_1 \widehat{u}_{1,t} + e_{2,t} \tag{10}$$

and we expect $b_{1,OLS}$ to converge to $\Omega'_{2,1}\Omega_{1,1}^{-1}$. This coupled with the restriction $\Omega\Omega' = \Sigma$ implies the following system of equations

$$
\begin{aligned}
b_{1,OLS} &= \Omega'_{2,1}\Omega_{1,1}^{-1}\\
\Sigma_{1,1} &= \Omega_{1,1}^2 - \Omega_{1,2}\Omega'_{1,2}\\
\Sigma_{1,2} &= \Omega_{1,1}\Omega_{2,1} - \Omega_{1,2}\Omega'_{2,2}\\
\Sigma_{2,2} &= \Omega_{2,1}\Omega'_{2,1} - \Omega_{2,2}\Omega'_{2,2}
\end{aligned}
$$

where $\Omega_{j,i}$ are appropriate partitions of Ω. These equations allow us to recover $\Omega_{1,1}$ and $\Omega_{2,1}$ up to a sign normalization, see Mertens and Ravn (2013) for more details.

In the toolbox, impulse responses are constructed as follows; we take a draw from the posterior distribution of the reduced form parameters $\left(\Phi_{(d)}, \Sigma_{(d)}\right)$ and compute the innovations, $U_{(d)} = Y - X\Phi_{(d)}$. With the latter, we compute $b_{1,OLS}$ from (9)-(10). Then, using $\Sigma_{(d)}$ we retrieve $\Omega_{1,1}^{(d)}$ and $\Omega_{2,1}^{(d)}$ and compute IRF. This approach is akin to the methods used in Miranda-Agrippino and Ricco (2017).

A.7 Long Run identification

We assume that the first variable of the VAR is specified in first difference and that the model looks like:

$$z_t \equiv \begin{pmatrix} \Delta y_t \\ \bar{x}_t \end{pmatrix} = \Phi(L) \begin{pmatrix} \Delta y_{t-1} \\ \bar{x}_{t-1} \end{pmatrix} + \Omega \nu_t$$

The long run restriction we impose implies that "only the first shock has a long run impact on the level of the y_t variable". In Gali (1999), only technology shocks have a long run impact on labor productivity; in Blanchard and Quah (1989) only supply shocks have a long run impact on output.

The time $t+k$ companion form of the responses and of the cumulative impulse response function (CIRF) are

$$z_{t+k} = F^k G \Omega \nu_t$$

$$z_{t+k} + z_{t+k-1} + z_{t+k-2} + ... + z_t = \begin{pmatrix} y_{t+k} - y_{t-1} \\ * \end{pmatrix} = \left(\sum_{j=1}^{k} F^j \right) G \Omega \nu_t$$

where the '*' indicates the sum of the $\bar{x}$ variables. Taking the limit of the CIRF we have:

$$\lim_{k \to \infty} \sum_{j=0}^{k} z_t = (I-F)^{-1} G \Omega \nu_t = \begin{pmatrix} * & 0 \\ * & * \end{pmatrix} \begin{pmatrix} \nu_{1t} \\ \nu_{2t} \end{pmatrix}$$

where ν_{1t} is the long run shock, and

$$G\ \Omega = (I-F) chol((I-F)^{-1} G \Sigma G (I-F)^{-1})$$

where $chol(A)$ is the Cholesky decomposition of A and $\Sigma = \Omega \Omega'$.

A.8 Forecast Error Variance Decomposition

Let $\widehat{y}_{t+h}$ be the h-step head forecast of the VAR model:

$$\begin{aligned} \widehat{y}_{t+h} &= E_t(y_{y+h}) = E_t(\varphi_0 v_{t+h} + \varphi_1 v_{t+h-1} + ... + \varphi_h v_t + ... + \varphi_{t+h} v_1 + \overline{\Psi}_{t+h}) \\ &= \varphi_h v_t + ... + \varphi_{t+h} v_1 + \overline{\Psi}_{t+h} \end{aligned}$$

where, as previously described, $\varphi_0 = \Omega$ and $\varphi_j = \Psi_j \Omega$ for $j = 1, ..., t$ and $\overline{\Psi}_t$ is the pure deterministic component. The mapping between $(\Psi_1, ..., \Psi_p)$ and are functions of $(\Phi_1, ..., \Phi_p)$ is described in section A.1.1. The forecast error at horizon h is

$$e_{t+h} = y_{y+h} - \widehat{y}_{t+h} = \varphi_0 v_{t+h} + \varphi_1 v_{t+h-1} + ... + \varphi_{h-1} v_{t+1}$$

The forecast error variance is:

$$
\begin{aligned}
E(e_{t+h}e'_{t+h}) &= \varphi_0 E(v_{t+h}v'_{t+h})\varphi'_0 + \dots + \varphi_{h-1}E(v_{t+1}v'_{t+1})\varphi'_{h-1} \\
&= \varphi_0 E\left(\left(\sum_k \nu_t^k\right)\left(\sum_k \nu_t^k\right)'\right)\varphi'_0 + \dots + \varphi_{h-1}E((\sum_k \nu_t^k)(\sum_k \nu_t^k)')\varphi'_{h-1} \\
&= \sum_{\ell=0}^{h-1}\varphi_\ell E\left(\left(\sum_k \nu_t^k\right)\left(\sum_k \nu_t^k\right)'\right)\varphi'_\ell \\
&= \sum_{\ell=0}^{h-1}\varphi_\ell E\left(\nu_t^1\nu_t^{1'}\right)\varphi'_\ell + \dots + \sum_{\ell=0}^{h-1}\varphi_\ell E\left(\nu_t^n\nu_t^{n'}\right)\varphi'_\ell \\
&= \sum_{\ell=0}^{h-1}\varphi_\ell J_1\varphi'_\ell + \dots + \sum_{\ell=0}^{h-1}\varphi_\ell J_n\varphi'_\ell
\end{aligned}
$$

and $\sum_{\ell=0}^{h-1}\varphi_\ell J_i\varphi'_\ell$ represents the contribution of shock i at horizon h.

A.9 Priors and Posteriors for Direct Methods

Assume that the prior is given by[14]

$$
\begin{aligned}
\underset{n\times n}{\Sigma_{(h)}} &\sim IW(S_h, d) \\
S_h &= \sum_{j=0}^{h} F^{h-j}G\Sigma G'F^{h-j'} \\
\begin{bmatrix} \underset{n\times np}{\beta_{(h)}} & \underset{n\times 1}{\alpha_{(h)}} \end{bmatrix} &\sim N(B_h, \Sigma_{(h)} \otimes \frac{1}{\tau_h}V) \\
B_h &= \begin{bmatrix} \underbrace{G'F^h}_{n\times np} & \underbrace{G'(I_{np}-F)^{-1}(I_{np}-F^h)F_0}_{n\times 1} \end{bmatrix}
\end{aligned}
$$

where d are the prior degrees of freedom (usually number of regressors minus the 2), τ_h controls the tightness of the prior response at the horizon h; F and F_0 are the VAR reduced form parameters and Σ is the covariance of the VAR reduced form errors; V is the $(np+1)\times(np+1)$ prior variance matrix of the reduced form coefficients (usually the iden-

[14] In what follows we use the result that $\sum_{j=0}^{h-1} F^j = (I_{np}-F)^{-1}(I_{np}-F^h)$

tity matrix). Because the prior is conjugate, the posterior also has a N-IW format:

$$
\begin{aligned}
\Sigma_{(h)}|Y_{(h)}, \tau_h &\sim IW(\overline{S}, \overline{d}) \\
\overline{S} &= \widehat{E}'_{(h)}\widehat{E}'_{(h)} + S_h + B'_h V^{-1} B_h + \widehat{B}' X'_{(h)} X_{(h)} \widehat{B} - \overline{B}'(X'_{(h)}X_{(h)} + V^{-1})\overline{B} \\
\overline{d} &= (T-h) - np - 1 + d
\end{aligned}
$$

$$
\begin{aligned}
\left[\beta_{(h)}\ \alpha_{(h)}\right] |Y_{(h)}, \tau_h &\sim N(\overline{B}, \overline{V}) \\
\overline{V} &= \Sigma_{(h)} \otimes \left(X'_{(h)}X_{(h)} + \left(\frac{1}{\tau_h}V\right)^{-1}\right)^{-1} \\
\overline{B} &= \left(X'_{(h)}X_{(h)} + \left(\frac{1}{\tau_h}V\right)^{-1}\right)^{-1} \left(\left(X'_{(h)}X_{(h)}\right)\hat{B}_{(h)} + \left(\frac{1}{\tau_h}V\right)^{-1} B_h \right)
\end{aligned}
$$

where

$$
\begin{aligned}
\widehat{B}_{(h)} &= \left(X'_{(h)}X_{(h)}\right)^{-1} X'_{(h)}Y_{(h)} \\
\widehat{E}_{(h)} &= Y_{(h)} - X_{(h)}\widehat{B}_{(h)} \\
Y_{(h)} &= \begin{pmatrix} y'_{1+h} \\ y'_{2+h} \\ \vdots \\ y'_T \end{pmatrix} \quad X_{(h)} = \begin{pmatrix} \mathbf{x}'_0 & 1 \\ \mathbf{x}'_1 & 1 \\ \vdots & \vdots \\ \mathbf{x}'_{T-1-h} & 1 \end{pmatrix}
\end{aligned}
$$

Posterior draws can be generated using the Gibbs Sampler algorithm, described in A.4.

A.10 The Matrix Variate Normal Distribution

The multivariate matrix normal distribution $Z \sim N(M, U, V)$ is

$$
p(\underset{(T\times n)}{Z} | \underset{(T\times n)}{M}, \underset{(T\times T)}{U}, \underset{(n\times n)}{V}) = (2\pi)^{-Tn/2}\,|V|^{-T/2}|U|^{-n/2} \exp\left\{-\frac{1}{2}tr\left[V^{-1}(Z-M)'U^{-1}(Z-M)\right]\right\}
$$

A.11 The Inverse-Wishart Distribution

The inverse Wishart is a distribution for symmetric matrices, $S \sim IW(\Sigma, d)$, where Σ is of size $n \times n$ is

$$
p(\underset{(n\times n)}{\Sigma} | \underset{(n\times n)}{\Sigma_0}, d) = \frac{|\Sigma_0|^{-d/2}}{2^{-dn/2}\Gamma_n(d/2)}|\Sigma|^{-(n+d+1)/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma_0\Sigma^{-1}\right]\right\}
$$

Notice that

$$
\begin{aligned}
E(S) &= \frac{\Sigma_0}{d-n-1} \quad if \quad d > n+1 \\
mode(S) &= \frac{\Sigma_0}{d+n+1} \quad if \quad d > n+1
\end{aligned}
$$

To generate a draw from an $IW(\Sigma, d)$:

- Draw d times a random vector of size $n \times 1$ from a $N(0, \Sigma^{-1})$.
- Combine them horizontally $[\eta_1, ..., \eta_d] = \eta_{(d \times n)}$.
- Compute the inner product and take the inverse $\rightarrow (\eta'\eta)^{-1}$.

We generate random draws form the inverse Wishart using the Matlab function developed in Dynare (see Adjemian, Bastani, Juillard, Karamé, Maih, Mihoubi, Perendia, Pfeifer, Ratto and Villemot (2011))