A hitchhiker's guide to empirical macro models[*]

Filippo Ferroni
Chicago Fed

Fabio Canova
Norwegian Business School, CAMP and CEPR

January 10, 2022
(First draft: May 2019)

Abstract

This paper describes a package which uses MATLAB functions and routines to estimate VARs, factor models and local projections with classical or Bayesian methods. The toolbox allows a researcher to conduct inference under various prior assumptions on the parameters, to produce point and density forecasts, to measure spillovers and to trace out the causal effect of shocks using a number of identification schemes. The toolbox is equipped to handle missing observations, mixed frequencies and time series with large cross-section information (e.g. panels of VAR, FAVAR and factor models). It also contains a number of routines to extract cyclical information and to date business cycles. We describe the methodology employed and implementation of the functions with a number of practical examples.

Keywords: VARs, Local Projections, Bayesian Inference, Identification, Forecasts, Missing Values, Filters and Cycles, MATLAB

JEL classification: E52, E32, C10

Contents

1   INTRODUCTION

The toolbox we described in this paper is written in MATLAB. It also works with OCTAVE, with the provisos explained below. The package is available as free software, under the GNU General Public License version 3. The toolbox and its documentation are continually updated and the lastest version can be downloaded from

https://github.com/naffe15/BVAR_

with all the supplementary material (data and source codes) to replicate the examples we present. The toolbox collects into a set of routines programs that the authors have used for Ph.D. courses or classes in central banks and international institutions. The toolbox comes as it is. We assume no responsibility whatsoever for its use by third parties and make no guarantees, implicit or explicit, about its quality, reliability, or any other characteristic. We would appreciate acknowledgment by citation of this working paper whenever the software is used. We extensively tested the toolbox prior to its public release. Users may inform the authors via email[1] or by opening an issue in GitHub of potential problems with the routines when they emerge.

The toolbox deals specifically with fixed coefficient Bayesian VAR models and local projections and has a number of identification options. It is equipped to handle missing observations, mixed frequencies and time series with large cross-section information (e.g. panels of VAR, FAVAR and factor models). It also contains a number of routines to extract cyclical information and to date business cycles. It does not directly cover topics such as time varying coefficients VAR models, models with stochastic volatility, non-gaussian or non-causal VARs, even though it provides tricks to deal with the first two with the available routines. Furthermore, it 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). When building a toolbox, one needs to decide when to stop adding features. We chose the current one: as it stands the toolbox has options that cover, more or less, everything that is currently interesting in modern applied macroeconomic analysis.

The toolbox is intended to users which are dissatisfied by the inflexibility of packages like BEAR (see Dieppe, van Roye and Legrand (2016)) or STATA, think that existing routines 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 details 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 (Adjemian, Bastani, Juillard, Karamé, Maih, Mihoubi, Perendia, Pfeifer, Ratto and Villemot 2011), 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-

---

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

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 VARs, FAVARs, factor models and panels of VAR models.

2 HITTING THE ROAD

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_/bvartools`

This has to be done every time you begin a MATLAB session. If you want to add permanently 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_/bvartools`, and save. For MATLAB 2020 and onward releases, the researcher should refer to the latest release version, see remark 1. 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`

If instead of MATLAB you use OCTAVE you will be doing the same. With OCTAVE, however the graphic is not as good and some of the pictures with present will not come out as nicely. Also, while you can use Chris Sims' routines, the other optimization routines featured in MATLAB, can not be employed. Finally, because the functions the toolkit uses are optimized for MATLAB, their use in OCTAVE slows down the computations quite a bit.

2.1 LOADING THE DATA

The user needs to prepare the data she wants 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 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_/examples/BVAR tutorial` and thus has dimension 188×5.

In principle, there is no need to specify the frequency: the time unit can be months, quarters, years, or mixed monthly and quarterly. Section 3.7 describes how to use the toolkit

---

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

when some data are missing. For plotting purposes, we define an array column vector, `T`, which is of the same length of the estimation sample. We use the convention that the first quarter (the first month) of the year corresponds to the integer of that year; for example, 2000Q1 (2000m1) corresponds to 2000.00. 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','sheetname');
```

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

3   VAR MODELS

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

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

$$= \Phi(L) y_{t-1} + \Phi_0 + u_t \tag{2}$$

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

$VAR(p)$ models can also be written in other ways. The one we will use quite often in this toolkit in the companion $VAR(1)$ form. A companion form is useful to generate forecasts and to compute impulse response functions. The companion form is

$$\underset{(np \times 1)}{\mathbf{x}_t} = \underset{(np \times np)}{F} \underset{}{\mathbf{x}_{t-1}} + \underset{(np \times 1)}{F_0} + \underset{(np \times n)}{G} u_t \tag{3}$$

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

where I is a $n \times n$ identity matrix and 0 is a $n \times n$ matrix of zeros.

For computing forecast error variance and historical decompositions, it is useful to write the VAR model in its Vector Moving Average (VMA) form:

$$y_t = u_t + \Psi_1 u_{t-1} + ... + \Psi_t u_1 + \overline{\Psi}_t \tag{4}$$

$$= \Psi(L) u_t + \overline{\Psi}_t \tag{5}$$

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

$(y_0, y_{-1}, ..., y_{-p})$ and represents the pure deterministic or non-stochastic component.[3]

At times, especially when dealing with small open economies, it may be useful to have in VAR variables which are taken as given from the point of view of the local economy. In this case, we consider a $VARX(p)$ model which we write as

$$y_t = \Phi(L)y_{t-1} + \Phi_0 + \Gamma z_t + u_t \tag{6}$$

where, as before, we assume that there are p lags of y_t, and z_t is the vector of exogenous variables. Here y_t is $n \times 1$ vector of endogenous variables, Φ_j with $j = 0, ..., p$ and Σ are defined as in equation (2), z_t is a $q \times 1$ vector, and Γ is a $n \times q$ matrix. For this model, the vector of parameters to be estimated is $\vartheta = vec(\Phi_0, \Phi_1, ..., \Phi_p, \Gamma, \Sigma)$. As we will see later on, the user supplies z_t to the toolkit and thus may include in the z_t not only variables which are contemporaneous to y_t but also lags, i.e. $z_t = [w_t, w_{t-1}]$.

3.1 INFERENCE

The toolkit allows to estimate ϑ and conducts inference conditional on the estimates of ϑ. Estimation can be performed using both classical (flat-prior) and Bayesian techniques. By default, Jeffrey (uninformative) prior is assumed; other options for the prior are available and discussed below. 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, an array of size T × n, 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.

Remark 1 *In previous version of the toolbox, the estimation of the VAR was implemented by the function* bvar.m. *The MATLAB 2020 release has a built-in object called 'bvar' which causes a crash with the* bvar.m *function. We recommend to use* bvar_.m *instead. Routines are backward compatible so that* bvar.m *still works with MATLAB releases earlier than 2020.*

The autoregressive model is specified with a constant. One can remove it using the noconstant option, before the estimation command is launched, e.g.

```
options.noncostant = 1;
BVAR = bvar_(y,lags,options)
```

---

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

When a flat or Multivariate Normal Inverse Wishart Conjugate prior is used, one can allow also a time trend in the model, if that is of interest. To do so, type `options.timetrend= 1` before the `bvar_` command. Time trends are not allowed in the case of Bayesian estimation with Minnesota prior as they are taken care of by means of prior restrictions. Unless otherwise specified, we will assume throughout the document that the user assumes a constant in the VAR specification and no time trend.

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

- `BVAR.Phi_draws` is a $(n \times \text{lags}+1) \times n \times K$ matrix containing K draws from the posterior distribution of the autoregressive parameters, Φ_j, assuming a Jeffrey prior. For each draw `k`, the first $n \times n$ 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 to be present.

- `BVAR.Sigma_draws` is a $n \times n \times K$ matrix containing K draws from the posterior distribution of the innovation covariance matrix, Σ.

- `BVAR.e_draws` is a $(T - \text{lags}) \times n \times K$ matrix containing K draws of the innovations.

- `BVAR.ir_draws` is a four dimensional object (i.e. $n \times \text{hor} \times n \times K$ 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 3.2 for the implementation of different identification schemes.

- `BVAR.forecasts.no_shocks` is a three dimensional object (i.e. $\text{fhor} \times n \times K$ 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 3.8 has the details on forecasting options.

- `BVAR.logmlike` contains the log marginal likelihood − a measure of fit.

- `BVAR.InfoCrit` contains four information criteria which can be used to select the lag length of the model: the Akaike information criterion, `AIC`, Hannan-Quinn information criterion `HQIC` and the Bayes information criterion, `BIC`, see A.2 for details.

By default, `K` is set to 5,000. It can be changed by setting `options.K` to any positive desired number prior to calling the `bvar_` routine.

Example 2 *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 log of HICP excluding food and energy (`CORE`), the one-year Euribor (`Euribor1Y`), the log of M3*

(`M3`) *and the exchange rate (*`EXRATE`*) from 2000m1 to 2015m8. We optimize the lag length*
of the VAR.

```
load Data
% collect the endogenous variables to be used in the VAR
y     = [IPI HICP CORE Euribor1Y M3 EXRATE];
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, opt);
  disp(['Number of lags' num2str(nlags)])
  disp(BVAR.InfoCrit)
end
```

3.1.1   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* regulating: (i) the prior tightness for the autoregressive coefficients of order one; (ii) the prior tightness for the autoregressive coefficients of higher lags; (iii) the weight on its own-persistence; (iv) the weight on the co-persistence of the data; and (v) the weight for the priors of the 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`).

---

[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.

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 sequence of commands:

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

As with Jeffrey prior, 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 a 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 a scalar controlling the Sum-of-Coefficient prior (default value 5)

4. `options.minn_prior_mu` is a scalar controlling the co-persistence prior (default value 2)

5. `options.minn_prior_omega` is a scalar controlling the prior on the covariance matrix (default value 2)

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 row vector that selects the parameters to be optimized. By default, `options.index_est=1:5`, that is, 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`.

---

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

3. `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 (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 3 (Mixed Options Maximization) *Using the same data in Example 2, we estimate the VAR using a Minnesota prior and 6 lags. 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. We use the MATLAB's Symplex routine to perform the optimization. The following set of instructions perform the task:*

```
options.max_minn_hyper = 1;
options.minn_prior_tau = 10;        % set tau
options.index_est       = [3 4];    % define the hyper-parameters to maximize
options.max_compute     = 7;        % optimization by Matlab Simplex
lags = 6;
BVAR = bvar_(y,lags,options);
```

The output printed in the command window is reported below

Iteration	Func-count	min f(x)	Procedure
0	1	-3791.37	
1	3	-3791.47	initial simplex
2	5	-3792.76	expand
...			
37	70	-3800.44	contract inside

```
Optimization terminated:
 the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-03
 and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-03


====================================================================


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

```
tau = 10
decay = 0.5
lambda = 5
mu = 2
omega = 2
log density = 3791.4


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


tau = 10
decay = 0.5
lambda = 1.0533
mu = 1.4001
omega = 2
log density = 3800.4


Maximization Successful: I will use the the mode values.
```

In this case, the optimum is found and the marginal likelihood increases by nine log points. Once the maximum is found, the posterior distribution is computed using the optimal hyper-parameter values. If the maximization is unsuccessful (convergence criteria are not satisfied or Hessian not positive definite are the typical reasons), the posterior distribution is computed with default values. In such case a warning is issued 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 advices 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 in the previous maximization step.

Since draws from the posterior distribution are not needed, rather than using the function `bvar_` it is possible to speed up the calculating using the function `bvar_max_hyper`, which is much faster.

Example 4 (Maximize the Marginal Likelihood Only) *We use the same setting of Example 3. We maximize the marginal likelihood only over the overall tightness (`tau`), given default values of the other hyper-parameters. We add an upper and lower bound for `tau` and use the MATLAB constraint minimization routine, `fmincon.m`. The following commands instruct the toolbox to optimize over the prior tightness hyper-parameter:*

```
clear options
% 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 to maximize
options.max_compute = 2;   % maximize using Matlab fmincon function
options.lb          = 0.8; % Lower bound
options.ub          = 10; % Upper 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 = 3792.6


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


tau = 5.1539
decay = 0.5
lambda = 5
mu = 2
omega = 2
log density = 3802
```

Thus, the log marginal likelihood increases by nine log points, when `tau` moves from 3 to 5.15.

Example 5 (continuation) *Using the mode value of the maximization in Example 4, 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:

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


tau = 5.1539
decay = 0.5
lambda = 5
mu = 2
omega = 2
log density = 3802


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


tau = 1.2553
decay = 1.8578
lambda = 1.1325
mu = 2
omega = 2
log density = 3813.9
```

The log marginal likelihood increases by almost twelve log 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.

3.1.2 Conjugate priors

Posterior distributions can also be constructed using a Multivariate Normal-Inverse Wishart conjugate prior. The command to activate 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 with a unitary diagonal matrix as scale and `n+1` degrees of freedom (see appendix A.12 for details). The way the prior is set is consistent with all the variables of the VAR having the same units. If this is not the case, one should use the option (`options.priors.Phi.cov` and/or `options.priors.Sigma.scale`) which changes the prior scale of the covariance matrix. In general, the hyper-parameters can be customized as follows:

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

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

- `options.priors.Sigma.scale` is a $(\mathtt{n} \times \mathtt{n})$ 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 distributions using a Gibbs sampler algorithm; see section A.4 for more details. Draws from the posterior distribution are collected in `BVAR.Phi_draws` and `BVAR.Sigma_draws`, which have the same dimensions as before.

3.1.3 Estimating a VAR with exogenous variables

A $VARX$ model can be estimated in the toolkit assuming either Jeffrey priors or conjugate priors; the Minnesota prior with dummy observations and with exogenous variables is not currently supported.

In order to activate the $VARX$ specification the exogenous variables need to be specified with the command:

```
options.controls = z;
```

where `z` is a $T \times q$ matrix. The first dimension of `z` is time and must coincide with the time dimension of `y` or with the sum of the time dimension of `y` and the out-of-sample forecast horizon, see section 3.8 for more details. As in a basic VAR, draws from the posterior distribution are collected in `BVAR.Phi_draws` and `BVAR.Sigma_draws`. 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 higher order lags. The $(\mathtt{lags} \times \mathtt{n})\mathtt{+1}$ row of `BVAR.Phi_draws(:,:,k)` contains the constant, Φ_0, if it assumed to be present. The following rows contain the estimate of Γ. Lags of exogenous controls can be used and specified as additional columns in `z`.

3.1.4 Regularization and Shrinkage

When dimension of the VAR is not trivial, estimation of an unrestricted VAR is problematic because the numbers of parameters to be estimated is large, even when one uses just one lag for each variable. To reduce the curse of dimensionality, the toolbox uses both regularization approaches and Bayesian shrinkage. With the term "regularization" we refer to techniques that employ a penalty function in the estimation of the coefficients, therefore producing a more parsimonious estimation problem and estimates with smaller variance. The toolbox considers Ridge (where parameters are penalized with a quadratic function), Lasso (where

parameters are penalized with an absolute value function), and Elastic-Net (which combines Ridge and Lasso approaches) techniques. With Bayesian shrinkage we refer to Minnesota-type priors, favoring a unit root behavior for the dynamics of single units so that *a-priori* no interdependences among variables are allowed.[6]

In both type of approaches, the penalization and the shrinkage functions are controlled by hyper-parameters. For penalization, BIC or other information criteria can be employed to select the values optimally. For shrinkage, the parameters are chosen to maximize the in-sample marginal likelihood, see section 3.1.1.

To activate the estimation of VAR systems with regularization, the user needs to specify the appropriate options. In the MATLAB command window (or in your script) the following sequence of commands can be used:

```
% activate the Ridge estimator
options.Ridge.est       = 1;
% penalty parameter
options.Ridge.lambda    = 0.25; % default value = 0.02
% activate the Lasso estimator
options.Lasso.est       = 1;
% penalty parameter
options.Lasso.lambda    = 0.25; % default value = 0.05
% activate the Elastic-Net estimator
options.ElasticNet.est  = 1;
% penalty parameter (alpha relative weight Lasso-Ridge)
options.ElasticNet.lambda = 0.25; % default value = 0.05
options.ElasticNet.alpha  = 0.5;  % default value = 0.5
```

Then, the default estimation command can be employed:

$$\texttt{BVAR = bvar\_(y, lags, options)}$$

With these options the output of the function, `BVAR`, contains additional fields summarizing the statistics obtained with the penalized estimator. For example, if the Ridge estimator is activated, we have

- `BVAR.Ridge.Phi` is a $(\texttt{n} \times \texttt{lags} + 1) \times \texttt{n}$ matrix containing the penalized estimate of the autoregressive parameters with the usual order.

- `BVAR.Ridge.Sigma` is a $\texttt{n} \times \texttt{n}$ matrix containing the penalized estimate of the innovation covariance matrix.

- `BVAR.Ridge.e` is a $(\texttt{T} - \texttt{lags}) \times \texttt{n}$ matrix containing the resulting realized innovations.

- `BVAR.Ridge.ir` is $\texttt{n} \times \texttt{hor} \times \texttt{n}$ matrix that collects the impulse response functions obtained with recursive identification with the usual order.

---

[6]Discussion and the implementation of the Minnesota-type prior can be found in section 3.1.1.

- `BVAR.Ridge.InfoCrit` contains three information criteria: the Akaike information criterion, `AIC`, Hannan-Quinn information criterion `HQIC` and the Bayes information criterion, `BIC`, see A.2 for details.

Similar sub-field structure are obtained with Lasso or Elastic-Net. In particular, the same estimates lised above can be found in `BVAR.Lasso` and `BVAR.ElasticNet`.

3.2 COMPUTING IMPULSE RESPONSES

Vector Autoregressive (VAR) models have become very popular to study how certain 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 the theoretical equilibrium conditions. The cost of using a limited number of restrictions is that structural disturbances may be confused, see Canova and Paustian (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 model. Under the conditions described in Canova and Ferroni (2021), the dynamic transmission of disturbances in the structural model can be mapped into structural VAR impulse responses.

Assuming normally distributed innovations[7], 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
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

---

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

using

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

for $h = 0, ..., H$, where 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 described in the previous section; the baseline function to generate the impulse response function is

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

where `options` can be used to select 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; by default, responses are computed using one standard deviation impulse in the shock.

In section 3.4 we present examples illustrating the use of several identification schemes.

3.2.1 Recursive or Cholesky identification

The Cholesky decomposition assumes a recursive ordering 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 of the variables in y will alter the responses 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 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 $n \times n$ matrix containing the OLS 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 $n \times$ hor $\times n \times$ K matrix), where the first dimension corresponds to the variables, the second to the horizon, 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.

3.2.2 Long run identification

The implementation of long run restrictions 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.[8] 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 $n \times$ hor $\times n \times$ K matrix), where the first dimension corresponds to the variables, the second to the horizon, the third to the disturbances, and the fourth to the responses obtained with particular draw from the posterior distribution of the parameters.

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

---

[8]See Appendix A.9 for details.

introducing more variables in first difference. I.e. VAR may include the growth rate of the relative price of consumption over investment and the growth rate of output and other variables. The first shock identified via long run restrictions corresponds to the investment specific technological change, i.e. the only shock affecting the long run level of the relative price; the second innovation corresponds to the permanent neutral technology shock so that the long run level of output is determined by investment specific and permanent neutral shocks.

3.2.3 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 one can impose. Note 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 `n × hor × n × K` matrix), where the first dimension corresponds to the variables, the second to the horizon, 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 `n × n × 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 also 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`. Alternatively, one could impose the constraint that the impact elasticity between the variables `a_1` and `a_2` (i.e. the ratio of the responses of two variables in log-levels) is larger than a threshold, i.e.

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

In general, the toolkit allows one to build generic 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 (see Kilian and Murphy (2012)). In others, cross sectional differences can be used (see De Graeve and Karas (2014)). 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.

3.2.4 Narrative restrictions identification

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 Ben Zeev (2018) or Antolin-Diaz and Rubio-Ramírez (2018)). 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, we use the structural VMA format:

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

where $\varphi_0 = \Omega$, $\varphi_j = \Psi_j \Omega$ for $j = 1, \dots, t$ and $\overline{\Psi}_t$ has deterministic variables.

Sign restrictions are imposed on the elements of φ_j for some j, $j = 0, \dots, t$ for a candidate Ω to be accepted. Narrative restriction are instead imposed on $\nu_t, t = [\tau_0, \tau_1]$, where τ_0 and τ_1 are chosen by the user for a candidate Ω to be accepted. In other words, we now have

21

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:

$$\texttt{options.narrative\{1\} = 'v(tau,n)>0'}$$

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,

$$\texttt{BVAR.irnarrsign\_draws} \qquad \texttt{BVAR.Omegan};$$

`BVAR.irnarrsign_draws` is a four dimensional object (i.e. a $\texttt{n} \times \texttt{hor} \times \texttt{n} \times \texttt{K}$ matrix), where the first dimension corresponds to the variables, the second to the horizon, 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 $\texttt{n} \times \texttt{n} \times \texttt{K}$ 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

$$\texttt{options.narrative\{1\} = 'sum(v(tau\_0:tau\_1),n)>0'}$$

where `tau_0` and `tau_1` are integers and `n` is an integer. The syntax means that the sum of the shock `n` between periods `tau_0` and `tau_1` is positive.

3.2.5 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:

$$\texttt{options.zero\_signs\{1\} = 'y(j,k)=+1'}$$

   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`.

22

2. Zero restrictions on impact (short run):

$$\texttt{options.zero\_signs\{2\} = 'ys(j,k\_1)=0'}$$

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

3. Zero long run restrictions:

$$\texttt{options.zero\_signs\{3\} = 'yr(j,k\_2)=0'}$$

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

$$\texttt{BVAR.irzerosign\_draws} \qquad \texttt{BVAR.Omegaz};$$

`BVAR.irzerosign_draws` is a four dimensional object (i.e. a `n` × `hor` × `n` × `K` matrix), where the first dimension corresponds to the variables, the second to the horizon, 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 `n` × `n` × `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.

3.2.6 External, instrumental or proxy variable 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.[9]

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

$$\texttt{options.proxy = instrument};$$

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

---

[9]See appendix A.8 for details.

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.

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 horizon, 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,:);
```

First-stage F-statistics with robust standard errors and adjusted R^2 of the instruments are reported respectively as

```
BVAR.proxy.F_m, BVAR.proxy.F_m_rob and BVAR.proxy.R2adj_m
```

3.2.7   Volatility-change or heteroskedasticy identification

It is well known that if the relative variances of structural shocks change across 'regimes', then the coefficients of a structural VAR may be identified withotu additional restrictions. More generally, lack of static identification disappears in a non-Gaussian time series model when the structural shocks are assumed to be instantaneously and serially independent, see Gourieroux et al. (Forthcoming). Intuitively, when empirical disturbances are non-Gaussian, moments higher than the second ones can be used to provide additional restrictions to select the unique rotation matrix that identifies the structural shocks.

As discussed in Sims (2020), when data can be split into two 'regimes', in which the volatility is different across regimes but the autoregressive parameters are constant, the identification of the structural shocks can be achieved by taking the ratio of the covariance matrix estimated within each regime.

To identify the structural shocks using volatility changes or heteroskedasticy restrictions, the user needs to specify a $T \times 1$ vector of the same size as the data array `y`, containing

zero and ones, where zero (one) indicate the first (second) regime. More than two regimes are currently not allowed for in the toolkit. Volatility changes or heteroskedasticy based responses are obtained when the following options is activated:

$$\text{options.heterosked\_regimes = Regimes;}$$

where `Regimes` is T×1 vector with zeros and ones. The estimated responses and the rotations are stored in, respectively

$$\text{BVAR.irheterosked\_draws} \qquad \text{BVAR.Omegah;}$$

`BVAR.irheterosked_draws` is a four dimensional object (i.e. a n × hor × n × K matrix), where the first dimension corresponds to the variables, the second to the horizon, the third to the disturbances, and the fourth to the responses produced by a particular draw from the posterior distribution. `BVAR.Omegah` is instead a three dimensional object (i.e. a n × n × 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.

3.2.8 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 obtained 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 function generates orthonormal matrices of size `n`

$$\text{Q = generateQ(n)}$$

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

$$\text{y = iresponse(Phi, Sigma, hor, Q)}$$

where `y` is a n × hor × 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 (Φ), the variance covariance matrix (Σ), the maximum horizon of the responses (hor), and the orthonormal (rotation) matrix (Q). If the latter is omitted, the identity matrix is used.

To illustrate how these commands could be employed, consider the situation where one wants to identify the shock maximizing the forecast error variance decomposition of a particular variable at a specific horizon, see section 3.3. 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,

$$\text{Qbar = max\_fevd(i, h, j, Phi, Sigma, Kappa)}$$

where `i` stands for the variable, `h` stands for the horizon and `j` corresponds to the shock. The function `max_fevd` generates `Kappa` draws for orthonormal matrices calling the function `generateQ.m`; if it is 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 3.3 and appendix A.10); FEVD is a $n \times n$ 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.

Once this is obtained, 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 four periods ahead:

```
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 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
```

3.2.9   Plotting impulse responses

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

$$\texttt{plot\_irfs\_(irfs\_to\_plot)}$$

`irfs_to_plot` is a fourth dimensional array where the first dimension corresponds to the variables, the second to the horizon, 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 $\sqrt{n} \times \sqrt{n}$ 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:

$$\texttt{plot\_irfs\_(irfs\_to\_plot, options)}$$

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 be plotted when one wants to compare responses.

3.3 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 we 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 mentioned, $\varphi_0 = \Omega$ and $\varphi_j = \Psi_j \Omega$ for $j = 1, ..., t$ are functions of $(\Phi_1, ..., \Phi_p)$[10] 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 observables $(y_t - \overline{\Psi}_t)$ as the sum of the contribution of each shock, i.e.

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

In the toolkit, an historical decomposition is performed with the function

$$\text{[yDecomp,v] = histdecomp(BVAR,opts);}$$

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 the posterior distribution.

Remark 6 (Historical Decomposition with non stationary data) *When data is non-stationary (e.g. it contains 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

$$\text{plot\_shcks\_dcmp\_(yDecomp, BVAR, optnsplt)}$$

where `optnsplt` is optional and controls a number of options:

- `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

---

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

`BVAR.varname`. Thus, 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.

- `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 contributions 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 employed, 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:

  `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` which are described in .

Forecast error variance decomposition (FEVD)

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:

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

We define the forecast error at horizon h as

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

The forecast error variance and its decomposition is given by[11]

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

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(hor,Phi,Sigma,Omega);
```

where `FEVD` is a `n` × `n` array containing the fraction explained by each shock; the first dimension corresponds to the variable, and the second to the shock; hence, the `(i,j)` element corresponds to the share of variance of variable `i` explained by shock `j` at horizon `h`. The inputs are respectively the horizon *hor*, the autoregressive parameters Φ, the variance covariance matrix Σ, and the rotation matrix Ω. If the latter is omitted, the function considers a recursive identification.

3.4 THE TRANSMISSION OF STRUCTURAL DISTURBANCES: SOME EXAMPLES

In this section we use the monthly US data of Gertler and Karadi (2015) (henceforth GK) 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 7 (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, i.e. twelve. We proceed in steps. First, we prepare the data. Since flat priors are used 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. 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. The relevant disturbance 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 percentiles of interest. The commands are as follows:*

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

---
[11]See appendix A.10 for details

```matlab
lags         = 12;                  % number of lags
options.hor = 48;                   % response horizon
% run the BVAR
bvar1    = bvar_(y,lags,options);
% sort the responses
irf_sort     = sort(bvar1.ir_draws,4);
% 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 8 (Plot responses with build-in function) *We 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. Figure 1 displays the results.*

```matlab
% 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];
% specify the responses
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'};
```
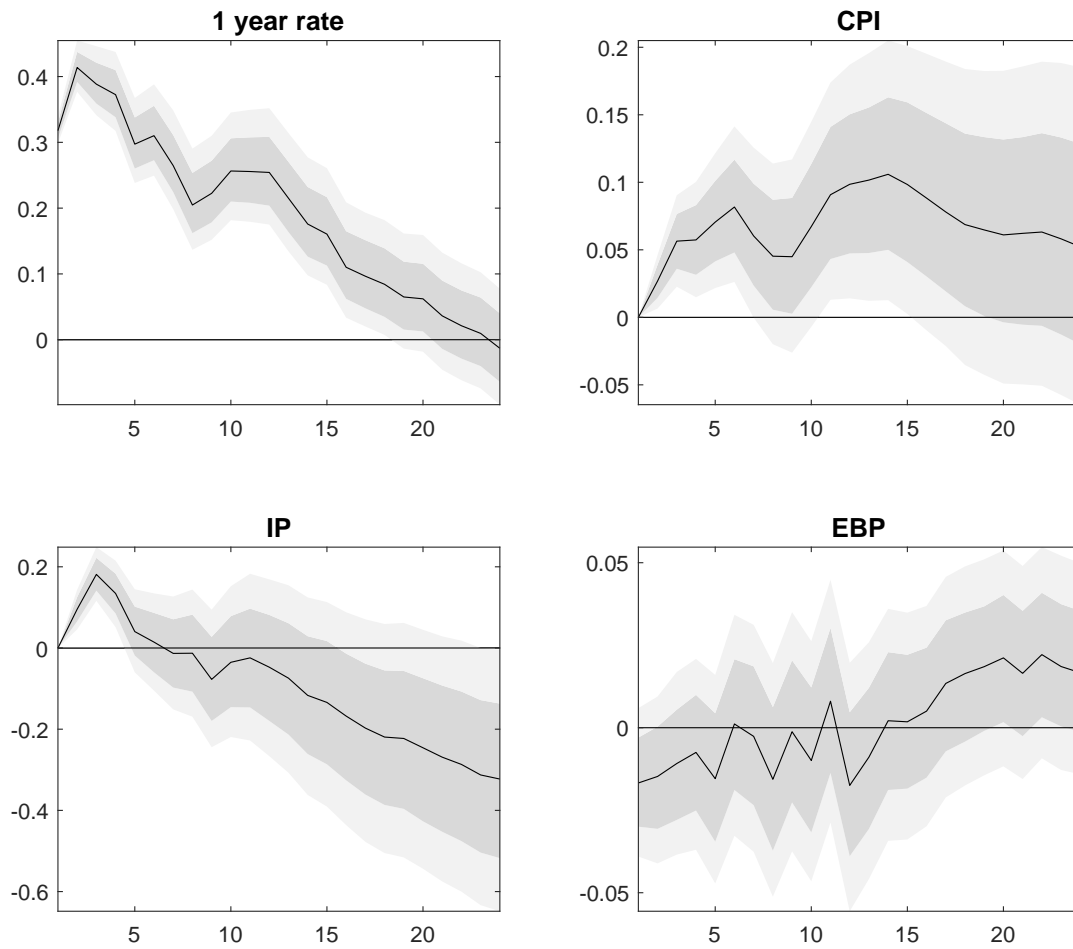
```
% additional 90% HPD set
options.conf_sig_2  = 0.9;
% finally, the plotting command
plot_irfs_(irfs_to_plot,options)
```

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

Example 9 (Responses to monetary Policy shocks, sign identification) *Assume the same data and the same setting of Exercise 7. 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 below. We plot the responses with the same settings as in Example 8 and figure 3 reports the results.*

```
% 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);
% Define the IRFs
% index of the shocks of interest (shock to gs1)
indx_sho     = [1];
% IRFs (indx_var is defined in Example 7: 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 2: Responses to monetary policy tightening, sign restrictions. Light (dark) gray bands report 68 (90)% credible sets.
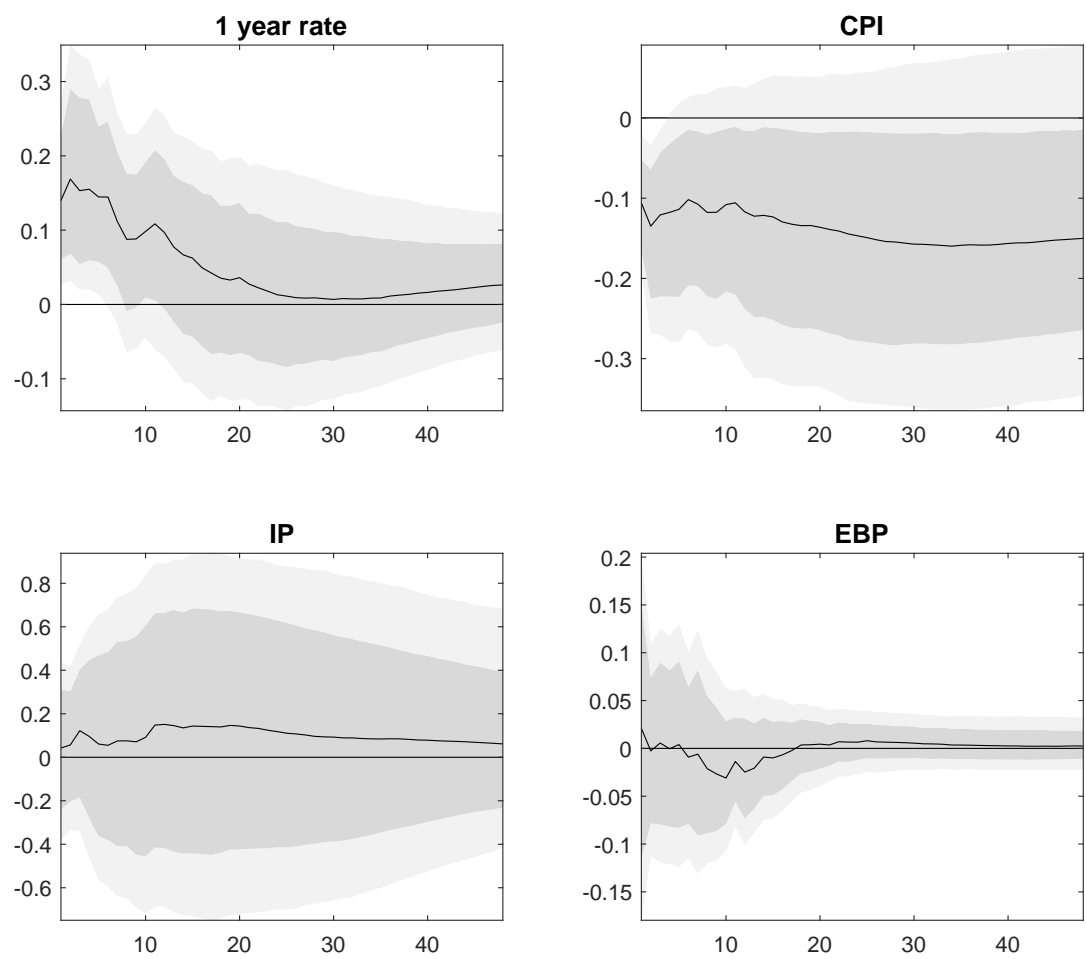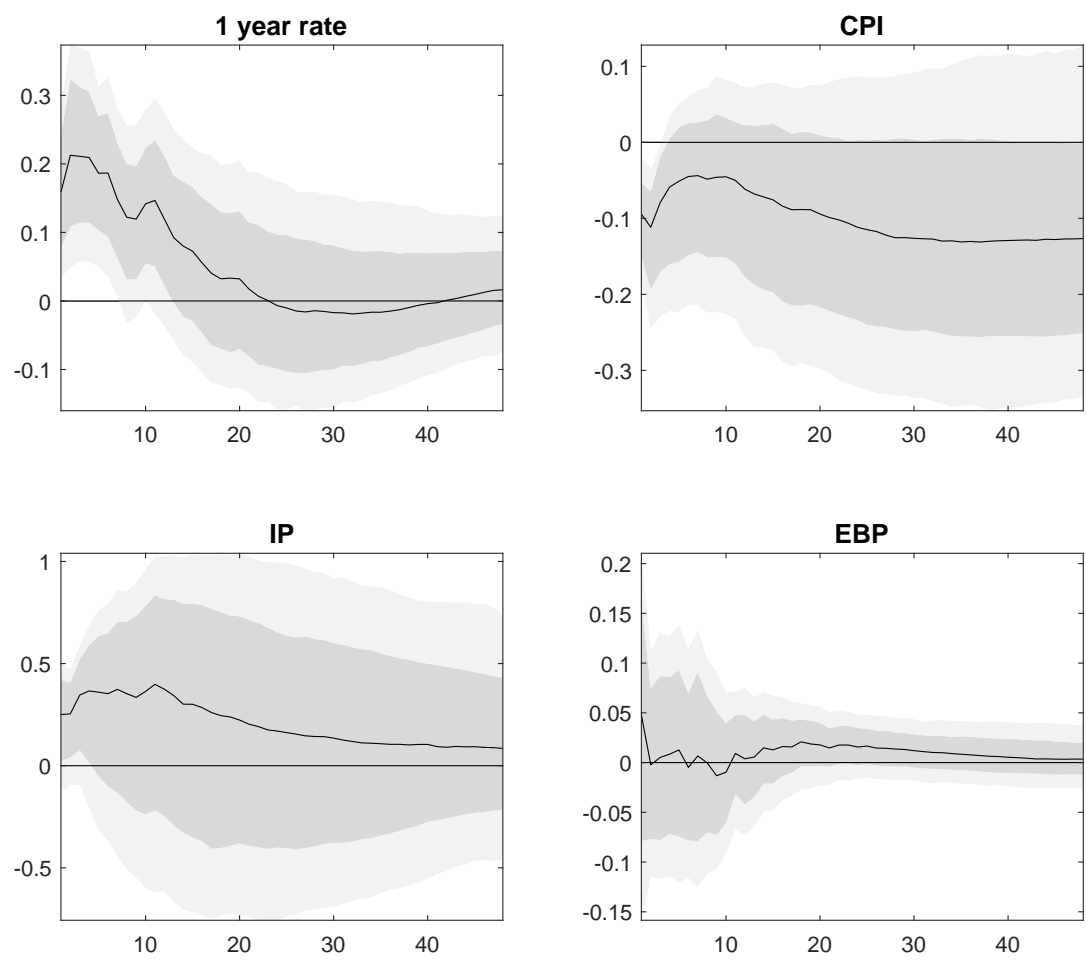
Example 10 (Responses to Monetary Policy shocks, sign and narrative restrictions)

Assume the same data and the same setting of Exercise 9. We 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 is the third, for 1980m10 the forth, for 1980m11 the fifth, and the innovation for 1981m05 the eleventh. We specify the restrictions below. Figure 3 reports the results. Bands are a bit narrower relative to the case of signs restriction. For sharper results, one could add more narrative restrictions, at the cost of increasing the computation burden. .

```
% 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'; % MP tightening in 1980m9-m11
options.narrative{2} = 'v([11],1)>0';  % MP tightening in 1981m5
% run the BVAR
bvar3       = bvar_(y,lags,options);
% Define the IRFs of interest
% index of the shocks of interest
indx_sho    = [1];
% IRFs (indx_var is defined in Example 7: 1. gs1; 2. logcpi; 3. logip; 4. ebp )
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: Responses to monetary policy tightening, sign and narrative restrictions. Light (dark) gray bands report 68 (90)% credible sets.
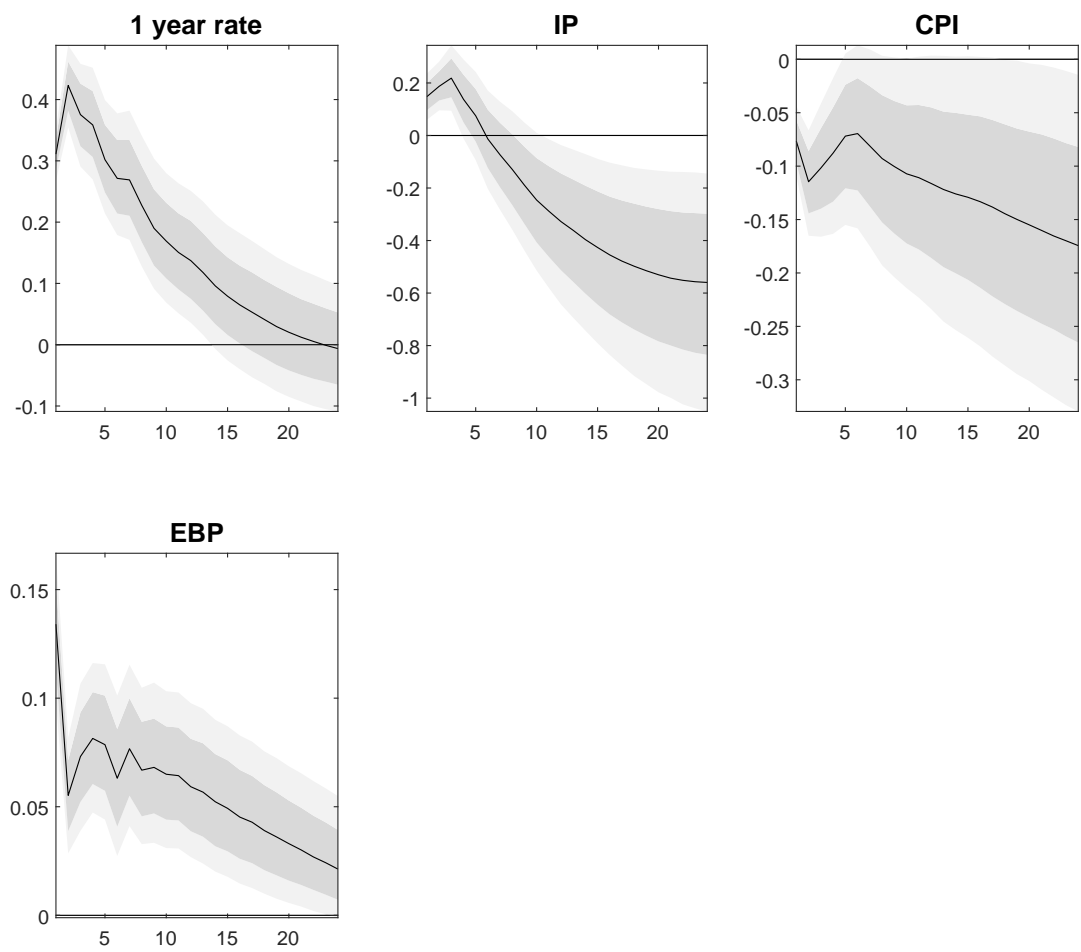
Example 11 (Responses to monetary policy shocks, identification with instruments)
We instrument the one-year government bond innovation with the GK preferred monetary policy surprise proxy. The ordering of the variables in y *is the following: one-year government bond rate(*gs1*), log of industrial production index (*logip*), log of CPI (*logcpi*) and the excess bond premium (*ebp*). The commands reported below allow to construct the responses with external instruments. The responses are plotted in figure 4.*

```
% define the dataset for the identification 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 are needed.
% 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 a monetary policy tightening, instrumental variables identification. Gray bands report 68 (90)% credible sets.

Example 12 (Responses to AD, AS, and MP shocks identified with a mixture of restrictions)

Assume the same data and the same setting of Exercise 7. We jointly identify an aggregate supply (AS), an aggregate demand (AD), and a monetary policy (MP) 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 instructions listed below. We plot the responses to the three shocks with the same settings as in Example 8. Figures 5, 6 and 7 present the responses to a one standard deviation orthogonalized innovation in AD, AS and MP.

```
% specify the restrictions
% 1) ad = aggregate demand disturbance [sign restrictions]
options.zeros_signs{1} = 'y(1,1)=1;'; % positive sign on IPI
options.zeros_signs{end+1} = 'y(2,1)=1;'; % 'end+1' adds a new cell - positive sign on CPI
options.zeros_signs{end+1} = 'y(3,1)=1;'; % positive sign on gs1
% 2) as = aggregate supply shock [sign restrictions]
options.zeros_signs{end+1} = 'y(1,2)=1;'; % positive sign on IPI
options.zeros_signs{end+1} = 'y(2,2)=-1;'; % negative sign on CPI
% 3.1) mp = monetary policy shock [zero restrictions]
options.zeros_signs{end+1} = 'ys(1,3)= 0;'; % no effect on impact on IPI
options.zeros_signs{end+1} = 'ys(2,3)= 0;'; % no effect on impact on CPI
% 3.2) mp = rate and bond premium go up [sign restrictions]
options.zeros_signs{end+1} = 'y(3,3)=1;'; % positive sign on gs1
options.zeros_signs{end+1} = 'y(4,3)=1;'; % positive sign on EBP
% run the BVAR
bvar4       = bvar_(y,lags,options);
% Define the shocks of interest
indx_sho          = [1:3];
% IRFs (indx_var is defined in Example 7: 1. gs1; 2. logcpi; 3. logip; 4. ebp )
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)
```

Figure 5: Responses to a one standard deviation innovation in AD identified with zero and sign restrictions. Light (dark) gray bands report 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 report 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 report 68 (90)% credible sets.
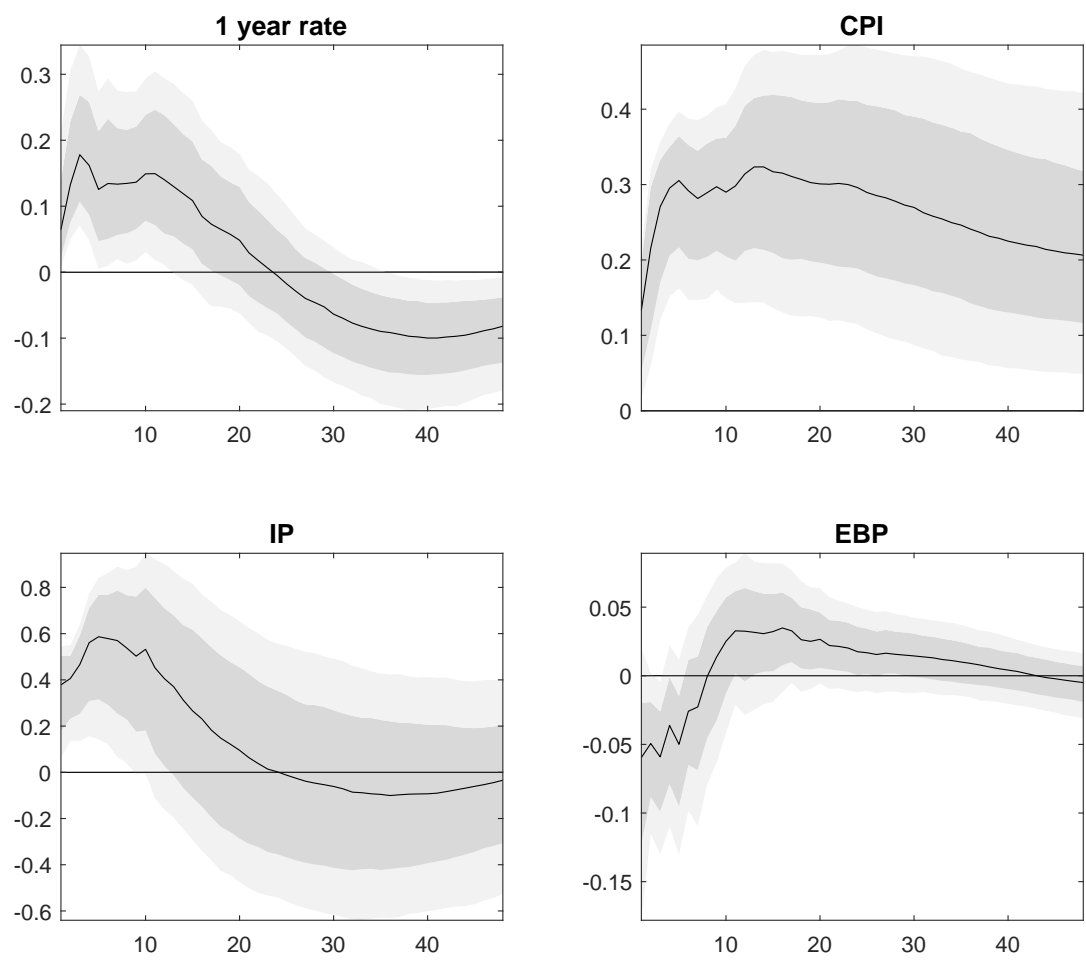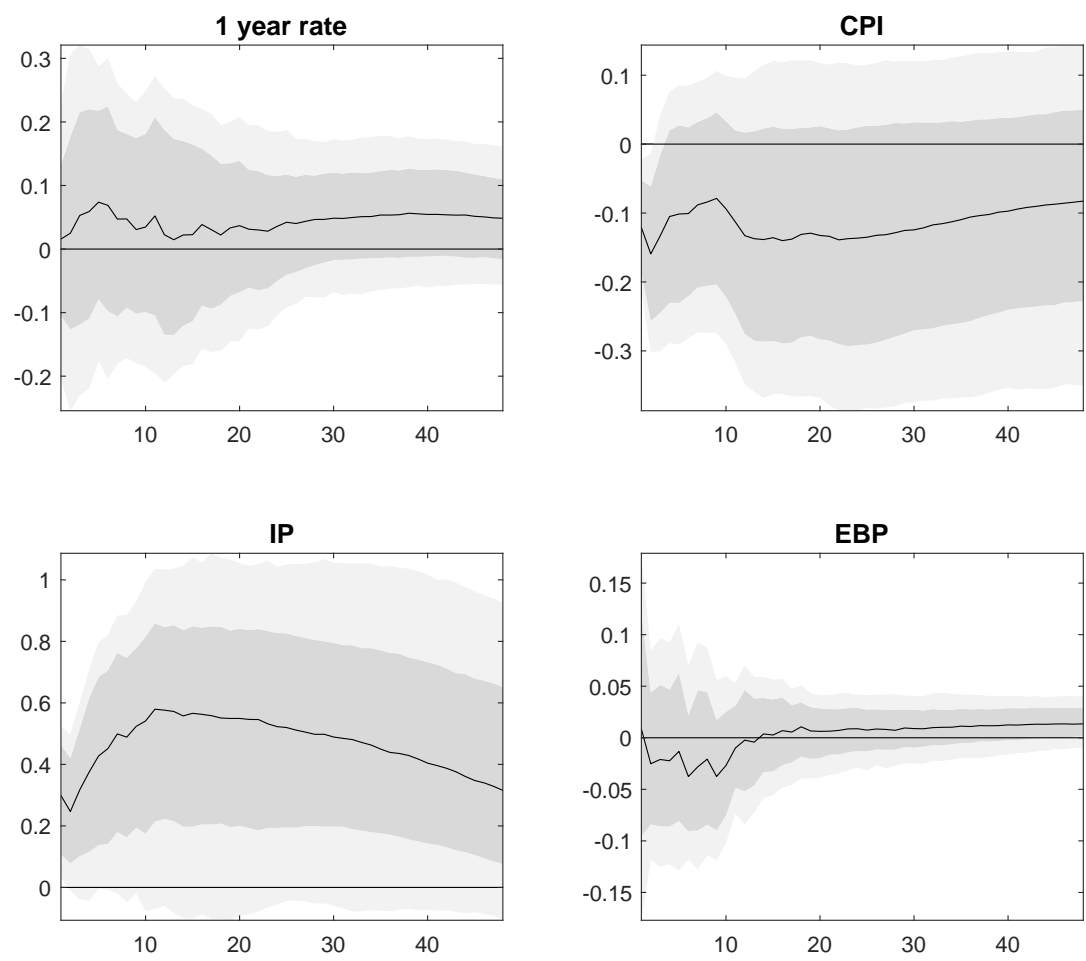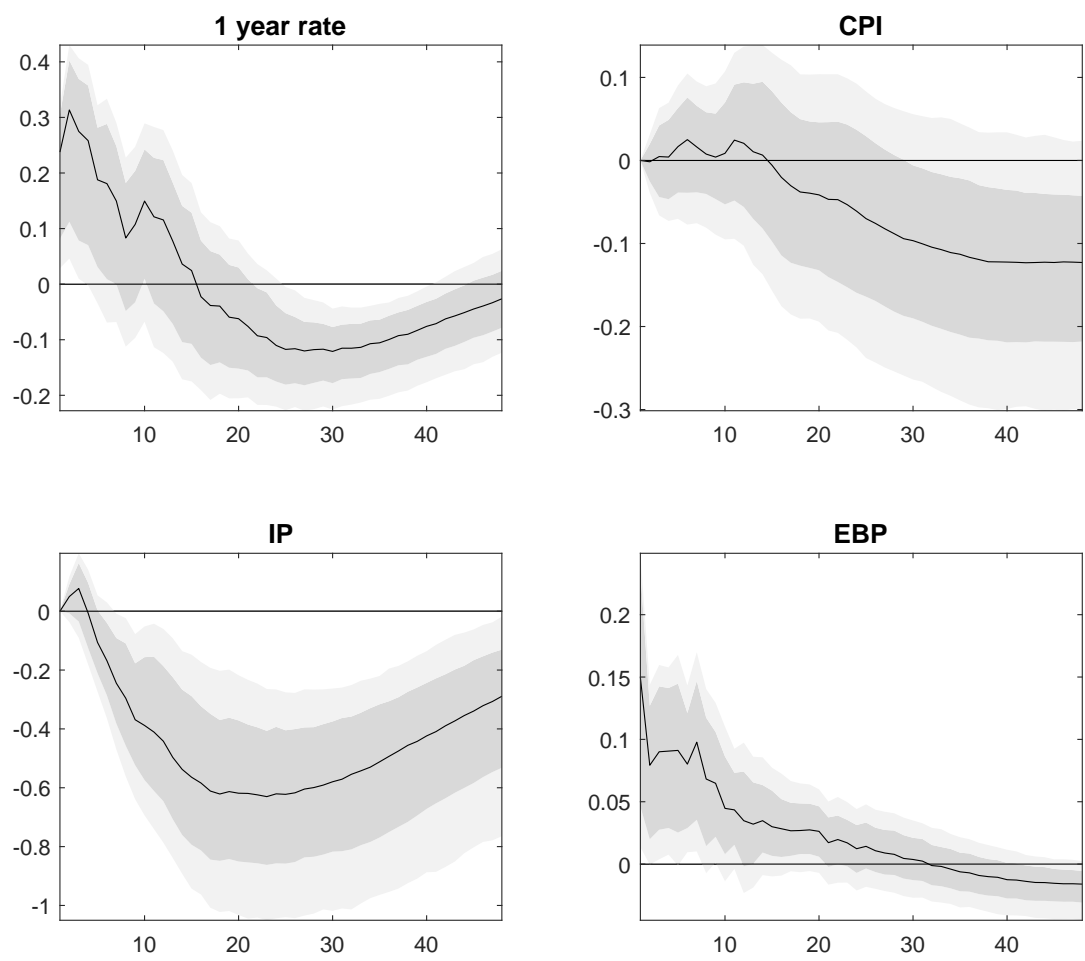
Example 13 (Responses to technology shocks, identified via long run restrictions)
Assume the same data and the same setting of Exercise 7. We identify a technology shock assuming that it is the only shocks having a long run impact on industrial production. We specify log of industrial production in first difference. For housekeeping we also remove previous identification settings. These commands are detailed below. We plot the responses with the same settings as in Example 8. Figure 8 reports the responses.

```
% Housekeeping: remove previous identification settings
options = rmfield(options,'signs');
options = rmfield(options,'zeros_signs');
options = rmfield(options,'saveas_strng');
options = rmfield(options,'shocksnames');
options = rmfield(options,'proxy');
% 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);
% 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: Responses to a technology shock identified with long run restrictions. Light (dark) gray bands report 68 (90)% credible sets.
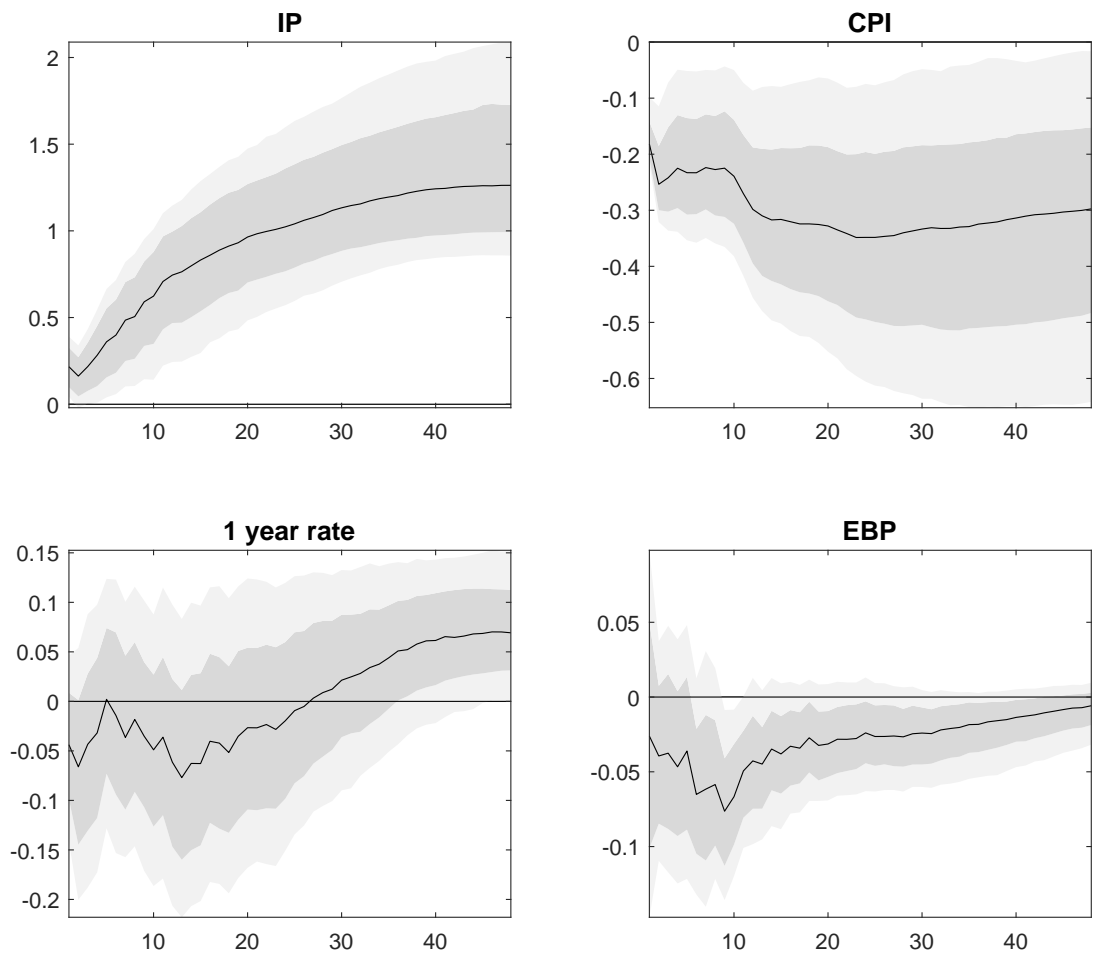
Example 14 (Transmission of Monetary Policy shocks with flat and Minnesota priors)
Assume the same settings in Example 7. We now consider a different prior; in particular, assume a Minnesota prior with a relatively loose overall shrinkage, e.g. $\tau = 0.5$, and use the first two years of the sample to calibrate the first and second moment of the prior. We compute the responses and compare them with the median response obtained with uninformative priors. The following commands allow to compare the responses under the two settings. The responses are reported in figure 9. Responses computed with the Minnesota prior tend to be more persistent than those obtained with uninformative (flat) priors.

```
lags            = 12;
options.presample   = 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];
% 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)
```
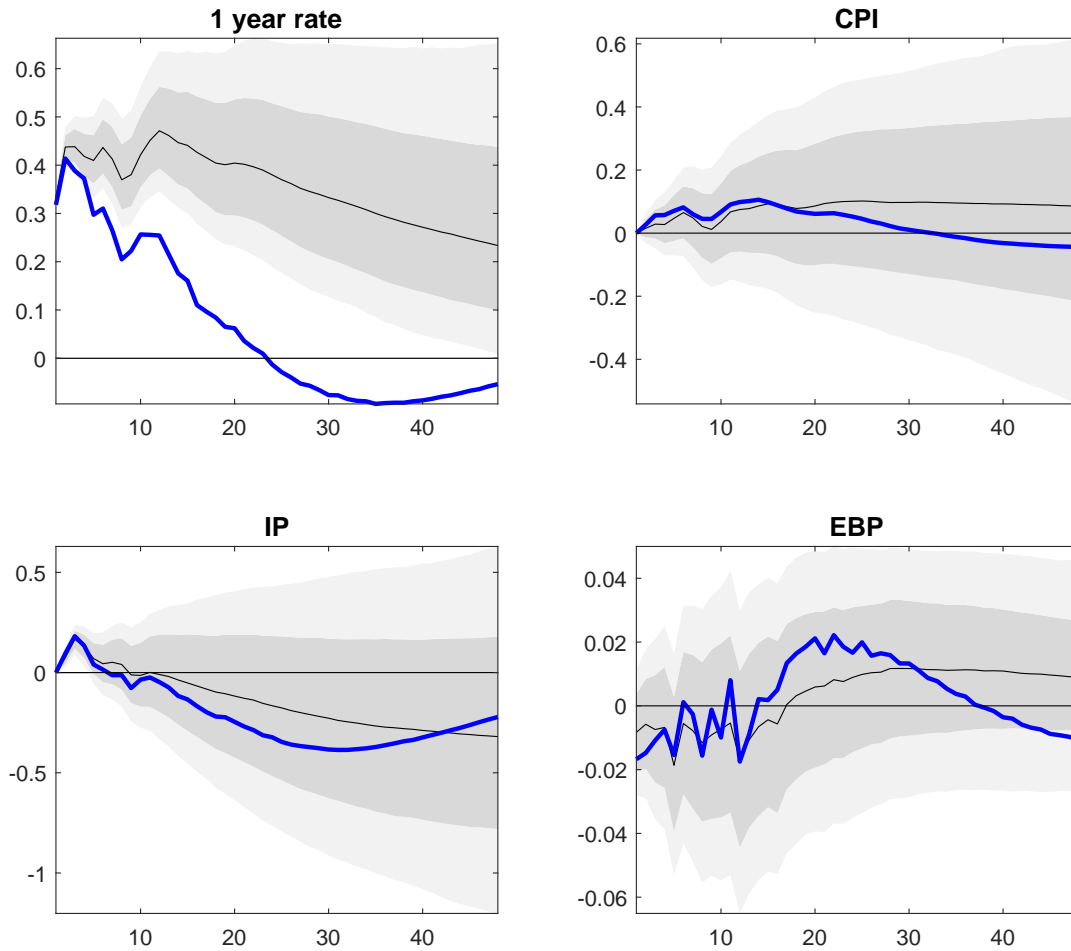
Example 15 (FEVD: The role of monetary policy shocks) *We assume the same settings of exercise 7. We compute the share of the forecast error variance at the two years*

45

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

horizon explained by monetary policy shocks. The commands are described below.

```
% use 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 forecast error
hh    = 24;
FEVD  = fevd(hh,Phi,Sigma);
```

Recall that `FEVD` is a 4×4 matrix where rows correspond to variables (in the same order as in the VAR) and columns to shocks. Thus, the monetary policy shock is ordered third and the contribution 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 volatility explained by MP shock    %
  logip   logcpi  gs1      ebp
  3.2456  3.6060  66.1847  1.5028
%======================================================%
```

Example 16 (Historical contribution of demand, supply and monetary policy shocks)
We consider the same settings of exercise 12. We plot the historical decomposition of the one year government bond and the excess bond premium in terms of AD, AS, MP and initial conditions. We combine together demand and supply shocks. The commands described below generate the required historical decomposition. Figures 10 and 11 report the historical decomposition of the one year government bond and excess bond premium from 2006 until 2012.

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

```matlab
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];
% plot the decompostition
plot_sdcmp_(yDecomp,bvar4,optnsplt)
```

Figure 10: Historical decomposition of one year government bond in terms of aggregate demand, aggregate supply, monetary policy, and other shocks and the initial condition.
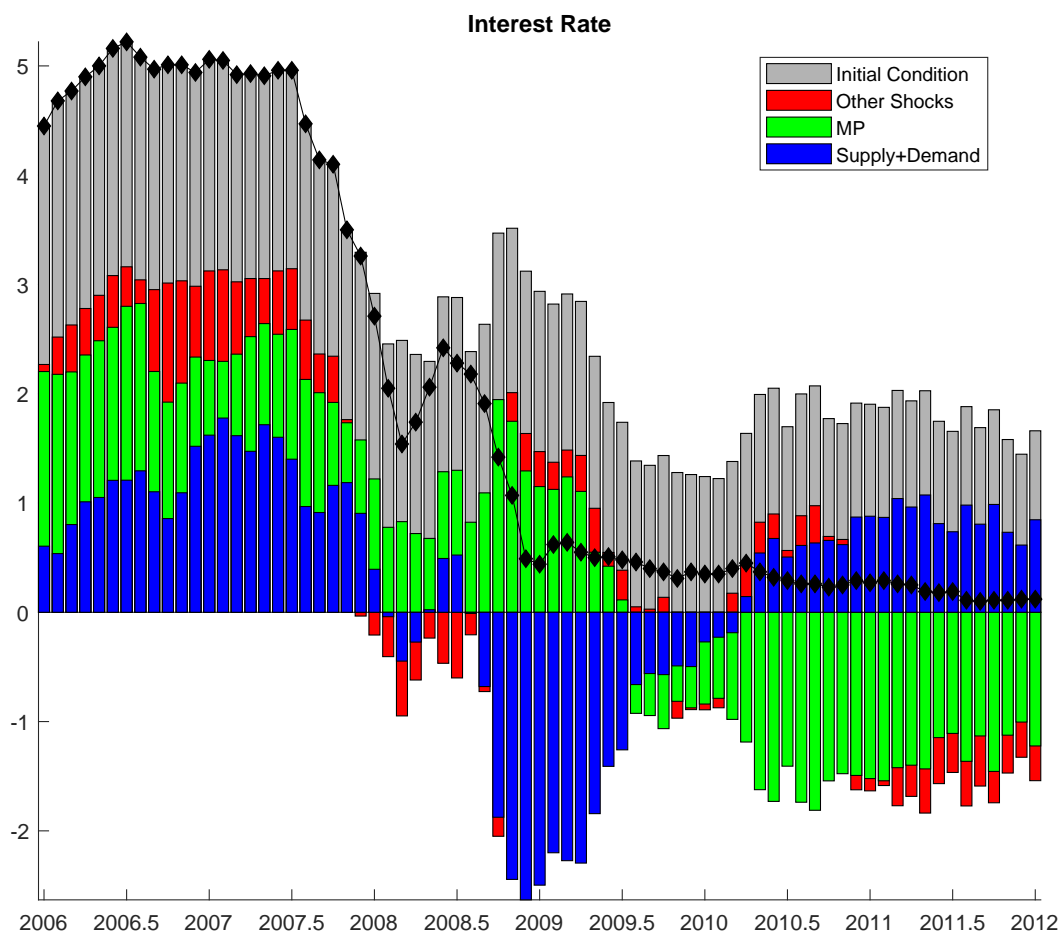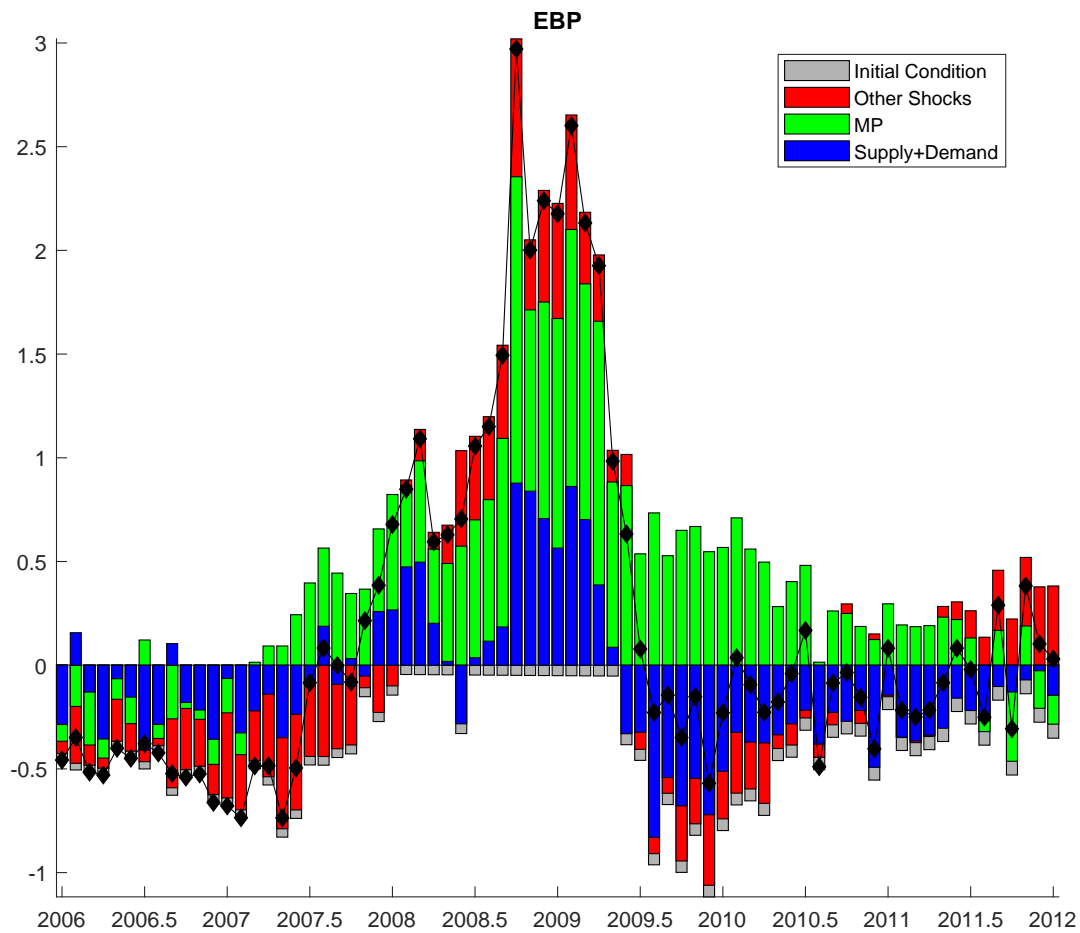
Figure 11: Historical decomposition of excess bond premium in terms of aggregate demand, aggregate supply, monetary policy, and other shocks and the initial condition.

Example 17 (Identification of Monetary Policy shocks with heteroskedasticity restrictions)
We consider the Euro Area industrial production index, core and headline HICP and the long term government bond; and the EA Monetary Policy (MP) events database constructed by Altavilla, Brugnolini, Gürkaynak, Motto and Ragusa (2019). This database collects variation of interest rates at different maturities in a narrow window around the ECB momentary policy communication and announcements. Altavilla et al. (2019) identify various monetary policy surprises from the high frequency changes in the long and short end of the term structure of the interest rates. In particular, they define a QE shock as the factor that has the smallest variance in the pre-crisis period (2 January 2002 - 7 Aug 2008). Essentially, they force the factor to affect the yield curve mostly in the post-crisis period, which turns out to contribute only to the movements in the long-end of the yield curve, and to be active only post-2014, naturally leading to the QE factor label.

We identify QE shocks using volatility-change restriction in a VAR model. We consider changes in the slope of the long end of the yield curve (e.g. ten year minus two year German government bond) in a narrow window around the ECB momentary policy communication and announcements, see Figure 12. As in Altavilla et al. (2019), we assume that the volatility break occurs in the post-crisis period. The commands are described below implement the identification and Figure 13 reports the responses obtained in response to the identified monetary shocks.

```
% EA MP events database + Macro data
load DataEAMP.mat
y               = [DE10Y-DE2Y IPI HICP CORE LTR10Y];
options.varnames = {'DE10Y-DE2Y' 'IPI' 'HICP' 'CORE' 'LRT10Y'};
% Options for Volatility-Changes identification
opts.heterosked_regimes = zeros(length(T),1);
% high volatility regime starts in 2008
opts.heterosked_regimes(find(T==2008) : end-20, 1) = 1;


opts.hor = 48;
lags     = 6;
bvar7    = bvar_(y,lags,opts);
% IRF to plot:
indx_var       = 1:size(y,2);
indx_sho       = 1;
irfs_to_plot   = bvar7.irheterosked_draws(indx_var,:,indx_sho,:);
% normalize to a 25 bpt increase in the 10-2Y spread
irfs_to_plot   = 0.25*irfs_to_plot/median(squeeze(irfs_to_plot(1,1,1,:)));
% names of the directory where the figure is saved
options.saveas_dir   = './irfs_plt';
```

```
% name of the figure to save
options.saveas_strng  = 'Heterosk';
% name of the shock
options.shocksnames   = 'MP';
% additional 90% HPD set
options.conf_sig_2    = 0.9;
options.nplots        = [2 3];
% add the Cholesky IRF with flat prior
XXX = squeeze(median(bvar7.ir_draws(indx_var,:,indx_sho,:),4));
% the plotting command
plot_irfs_(irfs_to_plot,options)
```

Figure 12: Slope of the long end of the yield curve (ten year minus two year German government bond) in a narrow window around the ECB momentary policy communication and announcements.
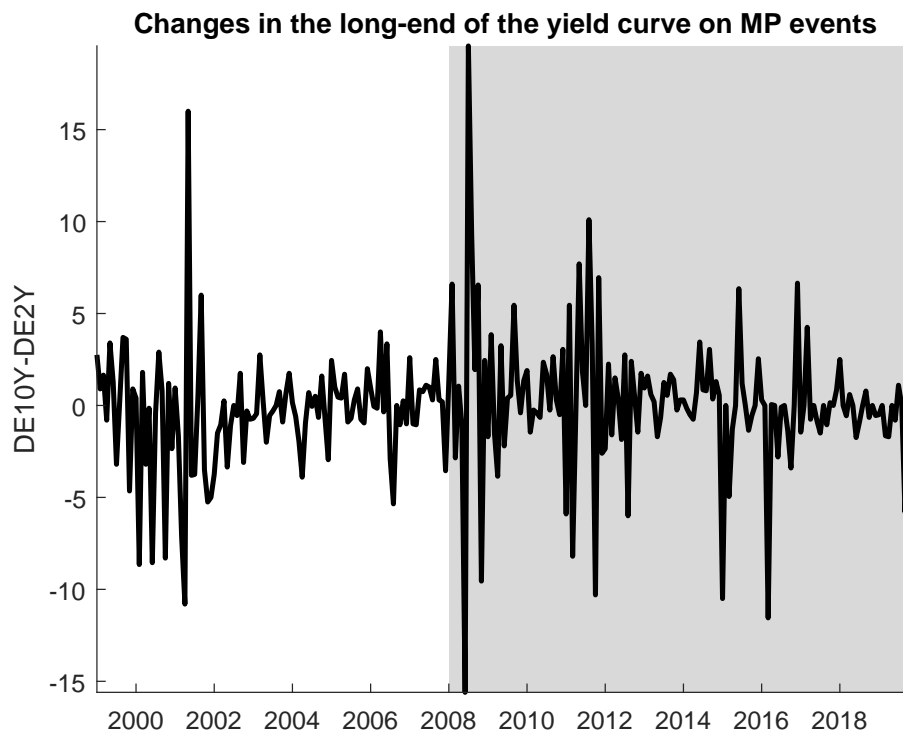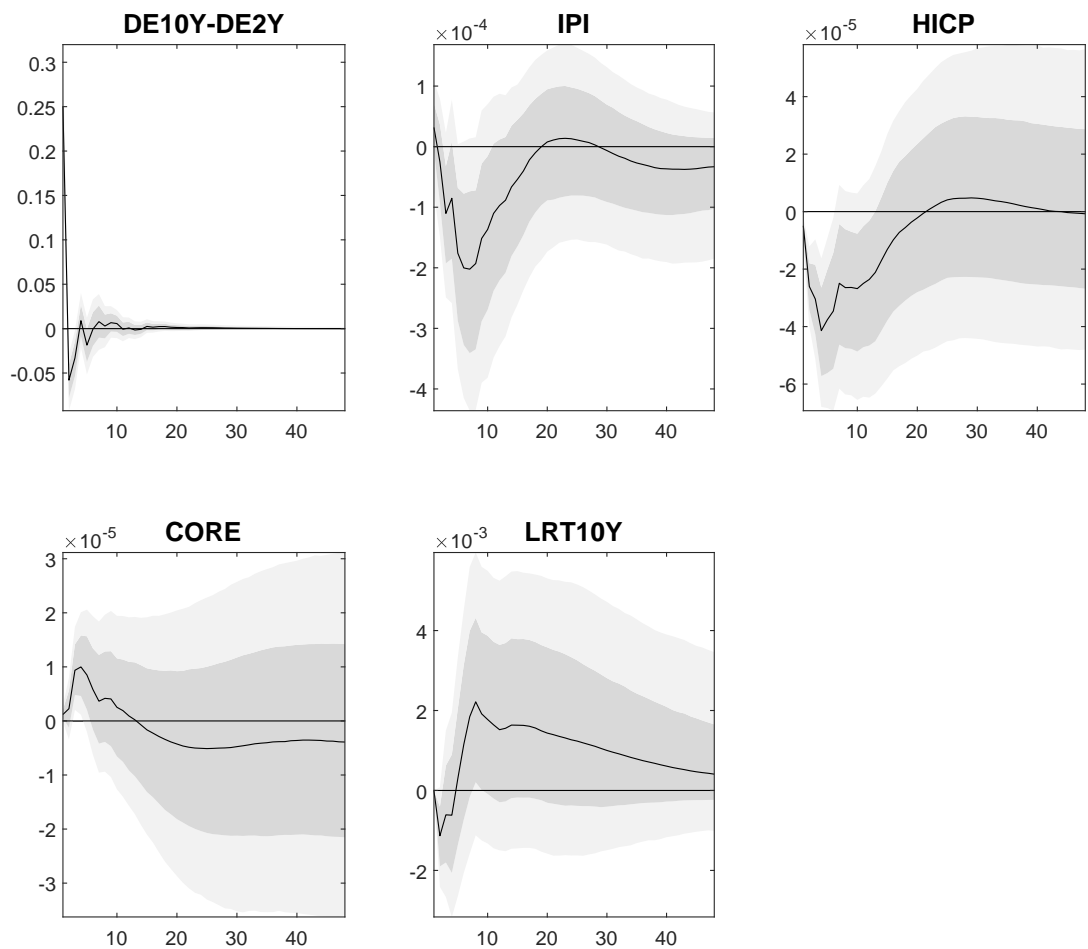
Figure 13: Responses to a QE shock identified via volatility changes (heteroskedasticity restrictions). Gray bands report 68 (90)% credible sets. The black line is the median Cholesky response with flat priors.

3.5   Network Connectedness and Spillovers

VAR models are often employed to construct measures of spillovers and connectedness of networks. In this case, one specifies a VAR with different units which are thought of parts of an interconnected network; typical examples includes countries GDP's, housing prices at the local level, bank's balance sheets indicators, stock market prices or volatilities. The flexibility of the VAR structure allows to match the covariance of the data with good precision and hence summarize fairly well the dependencies across units.

When dimension of the network is large, estimation of an unrestricted VAR is problematic because of the large number of parameters to be estimated. To reduce the curse of dimensionality, the toolbox uses both regularization approaches and Bayesian shrinkage, which are discussed in sections 3.1.4 and 3.1.1, respectively.

Once one of these approaches is used for estimation, one can measure dynamic interconnections among units, following Diebold and Yilmaz (2009, 2012). Connectedness is measured by the relative importance of the exogenous variations of unit i in explaining variations in unit j. In the VAR context this corresponds to the fraction of the forecast error variance of unit j explained by shocks originating from unit i at a given horizon.[12] The variance decomposition also allows one to distinguish how much of the forecast error variation in one unit or location (market, asset, etc.) is imported (vulnerability measure) or how important one unit or location (market, asset, etc.) is in spreading shocks elsewhere (systemicness measure).

In the literature, the identification of unit specific shocks is typically based on the generalized moving average representation of the VAR, as discussed in Pesaran and Shin (1998), which is thought to be convenient, as it produces variance decompositions which invariant to the ordering of the variables. However, as the shocks are not orthogonalized, the sum of the contributions to the variance of the forecast error (that is, the row sum of the elements of the variance decomposition matrix) is not necessarily equal to one. When a generalized MA decomposition is used, the toolbox normalizes each entry of the variance decomposition matrix by the row sum so that by construction rows add to one. The toolbox uses the generalized MA decomposition as a default option but allows users to specify alternative identification schemes, such as those discussed in section 3.2, if preferred.

The sum of all the elements outside the main diagonal of a variance decomposition matrix defines the *overall connectedness index*, which measures the contribution of the spillovers of shocks across classes of units to the total forecast error variance. In an hypothetical situation where units are independent, the forecast error variance matrix would be diagonal; spillovers negligible and the connectedness index zero. The relative magnitude of the off-diagonal elements thus offers an indication of the size of overall connectedness. Directional spillovers (how much unit i fluctuation spill to the system and how much of the fluctuations in unit i

---

[12]See also sections 3.3 and A.10.

are due to spillovers elsewhere) may also be used. The sum of the elements of a given column i (excluding the diagonal term) defines the *directional spillovers from* unit i to the rest of the system and measures the systemicness of unit i; the sum of the elements of a given row i (excluding the diagonal term) defines the *directional spillovers to* unit i from elsewhere in the market, which measures vulnerability.

To trigger the computation of connectedness measures at a specific horizon (say, 20 periods ahead), type in the MATLAB command window (or put in your script) the following sequence of commands:

```
options.nethor       = 20; % horizon of the FEVD - default 12
options.connectedness = 1;  % compute connectedness with Pesaran-Shin Generalized MA
```

`options.connectedness = 1` triggers the computation using the generalized MA representation of Pesaran and Shin (1998). If the user is interested in computing the connectedness using a recursive ordering, the appropriate option is:

```
% connectedness with recursive schemes
options.connectedness = 2;
```

For any identification schemes discussed in section 3.2, the option is

```
% connectedness with other identification schemes
options.connectedness = 2;
```

followed by standard identification commands, e.g.

```
% sign restrictions
options.signs1 = 'y(a,b,c)>0';
```

Then, the usual estimation command can be employed:

$$BVAR = bvar\_(y, lags, options)$$

The output of the function, `BVAR`, contains additional fields that summarize connectedness and spillover measures. For example, when combined with, say, the Lasso estimator, we have

- `BVAR.Lasso.Connectedness.index` is the *overall connectedness index*

- `BVAR.Lasso.Connectedness.FromAlltoUnit` is a `n × 1` matrix matrix containing *directional spillover to* unit.

- `BVAR.Lasso.Connectedness.FromUnitToAll` is a `n × 1` matrix containing *directional spillover from* unit.

- `BVAR.Lasso.Connectedness.theta` is `n × n` matrix collecting non-normalized forecast error variance decomposition (FEVD) at horizon `options.nethor`.

Identical sub-fields can be obtained when the Ridge or Elastic-Net estimators are activated; and the same measures can be found in `BVAR.Ridge.Connectedness` and `BVAR.ElasticNet.Connectedness`.

If a Minnesota prior (or any other prior discussed in this document) is used, the connectedness statistics are stored in `BVAR.Connectedness`. Unlike for penalized estimators, the overall index, the directional spillovers and the FEVD are computed for each draw of the posterior distribution. Hence, the user can then compute the posterior moments she prefers, e.g. median or mean across draws.

Example 18 (A Cryptocurrency network) *We consider the daily log price of about twenty crytpocurrencies*[13] *from January 1st of 2018 to May 11th 2021; the dimension of the cross-section is of medium size whereas the length of time series is quite large, i.e. 1227 data points (large T, medium N). We specify a VAR with three lags and consider an horizon of 10 days. The following commands allows to compute the statistics of interests*

```
options.nethor          = 10; % FEVD horizon - 10 days
options.connectedness  = 1; % Pesaran-Shin Identification


% activate the Ridge estimator
options.Ridge.est          = 1;
% penalty parameter
options.Ridge.lambda       = 0.25;


% activate the Lasso estimator
options.Lasso.est          = 1;
% penalty parameter
options.Lasso.lambda       = 0.25;


% activate the Elastic-Net estimator
options.ElasticNet.est     = 1;
% penalty parameter (alpha relative weight Lasso-Ridge)
options.ElasticNet.lambda = 0.25;
options.ElasticNet.alpha  = 0.5;


% Minnesota prior
options.prior.name = 'Minnesota';
options.max_minn_hyper  = 1;    % start the  optimization routine
options.index_est       = 1:2;  % hyper-parameters to maximize
options.minn_prior_tau     = 10;
options.minn_prior_lambda  = 0.1;
```

[13]Data have been downloaded from investing.com; the digital currency considered are Binance Coin, BitcoinCash, Bitcoin, Cardano, Dash, EOS, Ethereum, Ethereum Classic, IOTA, Litecoin, Monero, NEM, Neo, Qtum, Stellar, TRON, VeChain, XRP, Zcash.

```
lags           = 3;
options.K      = 5000;
bvar1          = bvar_(y, lags, options);
```

where y contains the log of the criptocurrency prices. The following commands allow to print the results in the command window

```
tmp0 = [mean(bvar1.Connectedness.Index) bvar1.Ridge.Connectedness.Index ...
    bvar1.Lasso.Connectedness.Index bvar1.ElasticNet.Connectedness.Index ];


tmp1 =[mean(bvar1.Connectedness.FromAlltoUnit,2) ...
    bvar1.Ridge.Connectedness.FromAllToUnit ...
    bvar1.Lasso.Connectedness.FromAllToUnit ...
    bvar1.ElasticNet.Connectedness.FromAllToUnit];


tmp2 =[mean(bvar1.Connectedness.FromUnitToAll,2) ...
    bvar1.Ridge.Connectedness.FromUnitToAll ...
    bvar1.Lasso.Connectedness.FromUnitToAll ...
    bvar1.ElasticNet.Connectedness.FromUnitToAll];


disp('=========================================')
disp('Overall Connectedness Index')
disp(['BVAR-Minnesota','Ridge','Lasso','ElasticNet'; num2cell(tmp0)])
disp('')
disp('=========================================')
disp('From Unit to All (Spillover)')
disp(['Crypto','BVAR-Minnesota','Ridge','Lasso','ElasticNet';
[cryptonames_' num2cell(tmp2)]])
disp('')
disp('=========================================')
disp('From All to Unit (Vulnerability)')
disp(['Crypto','BVAR-Minnesota','Ridge','Lasso','ElasticNet';
 [cryptonames_' num2cell(tmp1)]])
```

The output one obtain with these commands is as follows:

```
=========================================
Overall Connectedness Index
    'BVAR-Minnesota'    'Ridge'      'Lasso'       'ElasticNet'
    [       84.8309]    [87.5221]    [93.9536]    [   94.3299]
```

```
================================================

From Unit to All (Spillover)
    'Crypto'              'BVAR-Minnesota'     'Ridge'        'Lasso'        'ElasticNet'
    'Binance Coin'        [         3.5332]    [3.7637]      [ 3.3737]      [        0.4895]
    'Bitcoin Cash'        [         5.2934]    [5.5019]      [ 9.9311]      [        7.9953]
    'Bitcoin'             [         3.4505]    [3.4851]      [15.1561]      [       10.3404]
    'Cardano'             [         5.1485]    [5.3585]      [ 2.3836]      [       10.2717]
    'Dash'                [         4.6659]    [4.9631]      [ 7.7332]      [        7.6251]
    'EOS'                 [         5.4907]    [5.7174]      [ 2.9466]      [        1.5170]
    'Ethereum'            [         4.8594]    [5.1244]      [ 7.6583]      [        9.8134]
    'Ethereum Classic'    [         4.6885]    [4.7741]      [ 3.6264]      [        1.3985]
    'IOTA'                [         5.3183]    [5.5471]      [ 0.8907]      [        3.0406]
    'Litecoin'            [         4.6768]    [4.8538]      [ 8.2302]      [        5.0076]
    'Monero'              [         4.2444]    [4.4340]      [ 8.6582]      [        7.7734]
    'NEM'                 [         3.5711]    [3.0852]      [ 2.8140]      [        6.7822]
    'Neo'                 [         4.6892]    [4.8203]      [ 3.8007]      [        1.9244]
    'Qtum'                [         5.5555]    [5.7988]      [ 1.3109]      [        0.4691]
    'Stellar'             [         4.6006]    [4.8451]      [ 3.0314]      [        2.5121]
    'TRON'                [         4.2992]    [4.7131]      [ 7.4050]      [        5.0917]
    'VeChain'             [         6.4275]    [6.2478]      [ 1.4868]      [        8.8912]
    'XRP'                 [         4.2743]    [4.4660]      [ 2.0198]      [        1.3281]
    'Zcash'               [         4.7568]    [4.8850]      [ 6.7164]      [        7.2990]


================================================

From All to Unit (Vulnerability)
    'Crypto'              'BVAR-Minnesota'     'Ridge'        'Lasso'        'ElasticNet'
    'Binance Coin'        [         4.8543]    [4.9288]      [5.3281]       [        5.3993]
    'Bitcoin Cash'        [         4.8959]    [5.3056]      [4.9874]       [        5.1000]
    'Bitcoin'             [         5.2416]    [5.1993]      [4.7018]       [        4.9697]
    'Cardano'             [         5.1295]    [5.1615]      [5.3630]       [        4.9791]
    'Dash'                [         4.8888]    [5.1064]      [5.1046]       [        5.1189]
    'EOS'                 [         4.9668]    [5.1114]      [5.3467]       [        5.4337]
    'Ethereum'            [         5.0757]    [5.2256]      [5.1098]       [        4.9980]
    'Ethereum Classic'    [         4.9337]    [4.9393]      [5.3176]       [        5.4486]
    'IOTA'                [         5.1032]    [5.1351]      [5.3671]       [        5.3628]
    'Litecoin'            [         5.1024]    [5.1712]      [5.0803]       [        5.2611]
    'Monero'              [         5.0615]    [5.2171]      [5.0559]       [        5.1110]
    'NEM'                 [         2.3001]    [2.8206]      [5.3421]       [        5.1667]
    'Neo'                 [         4.7241]    [5.3412]      [5.3080]       [        5.4441]
```

58

'Qtum'	[	5.1237]	[5.1592]	[5.4002]	[	5.5080]
'Stellar'	[	4.8759]	[4.9859]	[5.3413]	[	5.3922]
'TRON'	[	4.7561]	[5.0166]	[5.1237]	[	5.2627]
'VeChain'	[	2.5415]	[2.7654]	[5.3620]	[	5.0372]
'XRP'	[	4.9285]	[4.7240]	[5.3753]	[	5.4401]
'Zcash'	[	5.0404]	[5.0702]	[5.1585]	[	5.1372]

Thus, the cryptocurrency market is extremely intertwined; e.g. the connedctedness indices computed using different estimators range between 86 to 94 percent. This means that most of the variations in cryptocurrency prices are the results of market's spillovers. From an investment strategy perspective, it seems very difficult to create a diversified portfolio with only cryptocurrencies. Perhaps surprisingly, Bitcoin the largest currency by far, in term of market capitalization has only limited influence on the price and volatility of the remaining cryptocurrencies; in fact, its price fluctuations only explain between 3-15 percent of the fluctuations of other currencies in the market.

One can also compute how these measures evolve over time. The next figure shows the connectedness index computed with a window of 150 days using the three regularization procedures.

Figure 14: Rolling estimates of Connectedness index.
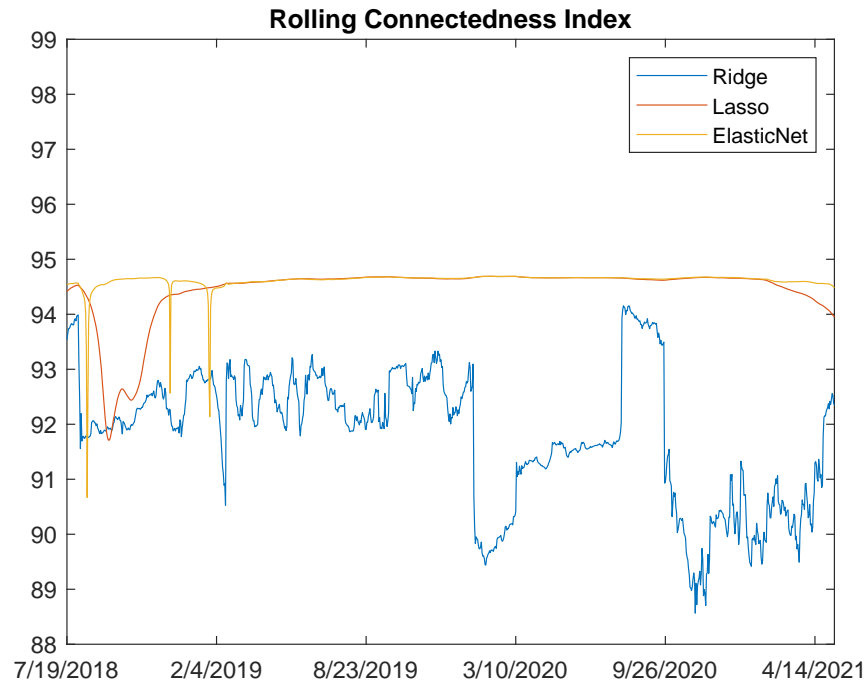
As we have mentioned, the toolbox has limitations. For example, it does not deal with parametric time varying (TVP) coefficient or with stochastic volatility (SV) models. However, it is easy to account for time variations in the coefficients and in the variance with the available functions. For example, one can run rolling windows VARs and examine whether the parameters, the responses, or other interesting statistics vary over time.[14]

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, see e.g. Boivin, Kiley and Mishkin (2010). The following example describes how one can use the toolbox to measure how much variations there are in the responses of unemployment to monetary policy disturbances over time.

Example 19 (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 a 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 of time. The sample runs from 1953Q1 to 2007Q4. Two lags are used for the estimation.

We employ 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 to the response functions obtained with OLS estimates of the autoregressive parameters and of the 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, the second to the sample, and the third to the response of unemployment rate. Figure 15 presents the dynamics of the unemployment rate to a standardized one percent increase in the 3-month 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
% select GDP deflator (year-on-year) Unemployment; 3m Tbill.
yQ     = [GDPDEF_PC1 UNRATE TB3MS];
lags   = 2;
% setting for parameters for the rolling windows
Wsize  = 120;   % quarterly lenght of the rolling window
```

---

[14]For Bayesian kernel estimation of VAR models see Petrova (2019).

```matlab
shift    = 4;        % time shift between adjacent windows
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 IRFs (one IRFs 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 normalized responses
  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,:,:)))   % three dimensional plot
axis tight;
```

Figure 15: Time evolution of the responses of the unemployment rate to a 1% increase in the 3-month Treasury bill.
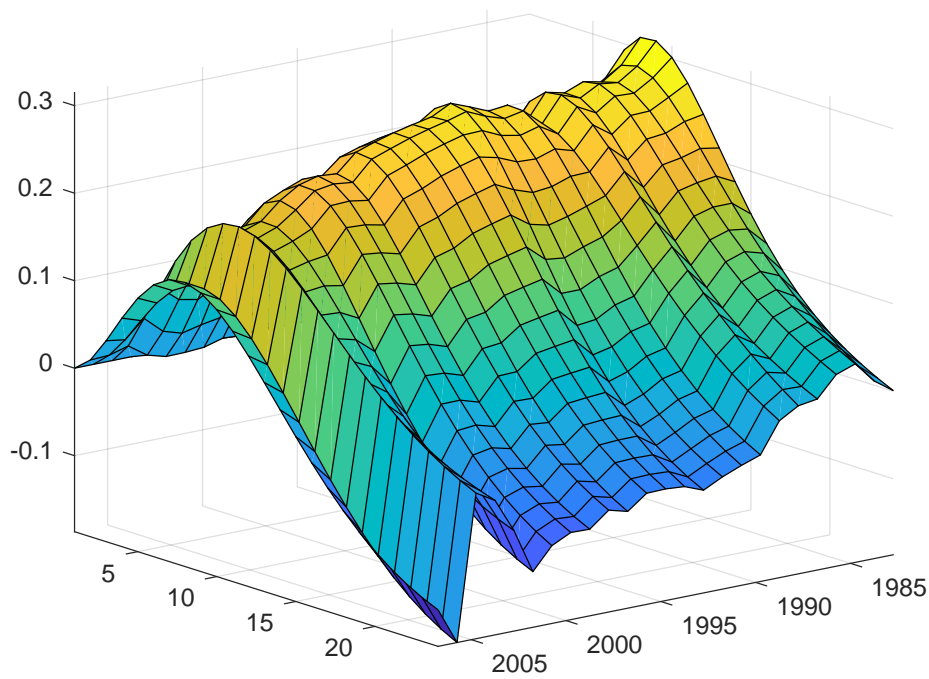
One way to take into account stochastic volatility is to scale the variables of the VAR, prior to estimation using a recursive (non-parametric) measure of volatility. For example, one can preliminary run univariate regression of y_{jt} on its own lags, compute $w_{1jt} = |e_{jt}| \equiv |y_{jt} - A_j(L)y_{jt-1}|$ or $w_{2jt} = e_t^2$ and use, as VAR variables, y_{jt}/w_{1jt} or $y_{jt}/\sqrt{w_{2jt}}$. The first variable transformation 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) estimation
  yhat(:,var) = y(lags+1:end,var) ./ abs(tmp_.e_ols);
end
```

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

Alternatively, the toolkit allows to estimate a VAR corrected for heteroskedasticity when high volatility cluster occurs at known dates, along the lines discussed in Lenza and Primiceri (2020). In particular, we allow the $VAR(p)$ to be specified as follows

$$y_t = \Phi(L)y_{t-1} + \Phi_0 + s_t u_t$$

where the factor s_t is typically equal to one and larger than one at specific known dates. The size of the increase in variance, s_t, can be calibrated or estimated. The two following examples describe how to setup the estimation under the two situations.

Example 20 (COVID-19 pandemic and Heteroskedastic VARs) *Lenza and Primiceri (2020) consider US macroeconomic monthly data on unemployment, industrial production, employment, real personal consumption expenditure (PCE), core and headline PCE prices from 1988m12 to 2020m7. They notice that at time of the outbreak of the COVID-19 pandemic (March 2020) an abnormal data variation is observed; thereby they assume that the change in volatility occurs there. Before then, there were no jumps in volatility. We postulate that the increase in volatility occurs in three months, March, April and June, and the volatility increases by a factor of 10. The following commands allow to correct the VAR parameters estimates for heteroskedasticity.*

```
tstar  = find(time==2020) + 2; % march 2020
% scale the observed variables by factor >1 in the periods that
% characterize the COVID-19 induced recession
st                  = ones(size(y,1),1);
st(tstar:tstar+2 ,:) = [10 10 10]; % March, April, May
st(1:lags)          = [];
options.heterosked_weights = st;
options.tstar = tstar;
bvar_het            = bvar_(y,lags,options);
```

Example 21 *Consider the same setting of exercise 24. Similar to Lenza and Primiceri (2020), we maximize marginal data density over the Minnesota hyper parameters and the size of the scaling. The following commands allow to compute the optimal level of scaling heteroskedasticity correction and optimal amount of Minnesota prior shrinkage.*

```
hyperpara(1)      = 3; % tau
hyperpara(2)      = 0.5;% decay
hyperpara(3)      = 1; % lambda
hyperpara(4)      = 1; % mu
hyperpara(5)      = 2; % omega
hyperpara(6)      = 2; % s0: scale march 2020
hyperpara(7)      = 2; % s1: scale april 2020
hyperpara(8)      = 2; % s2: scale may 2020
% setting the options
options.index_est         = [1 6:8];      % hyper-parameter over which maximize
options.max_compute       = 1;            % maximize  using Matlab fmincon function
options.objective_function = 'bvar_opt_heterosked';
options.tstar             = find(time==2020) + 2; %march 2020
% Compute the optimal values
[postmode,logmlike,HH] = bvar_max_hyper(hyperpara,y,lags,options);
```

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 shocks is via identification restrictions (as in Canova and Pappa (2011)). Alternatively, one 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 exercises can be conducted within the available functions, but they require some attention. For example, using sign and quantity restrictions one can measure when large and small shocks have similar effects on the variables of interest, see 3.2.3. Similarly, by interacting one or more variables of the vector y with 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.

3.7   Mixed Frequency VARs

The Mixed Frequency VAR (MF-VAR) model we consider has constant parameters and it is designed to deal primarily with monthly and quarterly data. 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, the problem is simple. 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. With the Gibbs sampler, we generate a draw from the posterior distribution of the reduced form VAR parameters, conditional on observables and states. With this draw and the Kalman smoother, we can estimate the unobserved states, and thus implicitly recover the values of the variables whose observations are missing.

A standard application of the MF-VAR function is when one has recent and more frequent available information (e.g. monthly data) and wants to construct timely estimates of variables of interest which are observed only at lower frequency (e.g. quarterly data). This generate nowcasts that incorporate available information efficiently. The efficiency property is crucial for forecasting as 'good' nowcasts are the necessary prerequisites for generating accurate out-of-sample predictions (see section 3.8). For example, applied researchers and economic analysts are often interested in the 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 contained in monthly data to estimate the monthly level of GDP. The MF-VAR function is well suited for this purpose. In particular, it treats the monthly GDP values as unobserved, and uses a state space representation to reconstruct missing observations. 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 in the toolbox closely follows Schorfheide and Song (2015).

Another application of interest is when one jointly use monthly and weekly times series 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 reported 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 straightforward.

The estimation of a MF-VAR can be performed using any of the prior specifications discussed in section 3.1 with the caveat that maximization over the Minnesota prior hyperparameters 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 3.2, and the production of point or density forecasts, as discussed in section 3.8.

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 $\{y_t^m, y_t^q\}$, 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 equation 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 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, the user needs to type

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

where `num` is a scalar indicating the position (column in `y`) in the dataset of the flow variable we wish to aggregate.[15] Once the mapping between monthly and quarterly variables is selected, we assume that $[x_{q,t}'\ \ x_{m,t}']'$ follows a VAR with p lags. The combination of the measurement equations and the transition equations of the VAR where some variables are latent (i.e. $x_{q,t}$) generate a state space system. Since system is linear and Gaussian, the Kalman filter allows to generate an estimate of the latent states and the smoother to generate a random draw from the distribution of the states conditional on a value of the VAR parameters.[16]

The unobserved or irregularly sampled variable becomes an additional output of the `bvar_` function; in particular, the `T × n × K` array called

<p style="text-align:center">BVAR.yfill</p>

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

<p style="text-align:center">BVAR.yfilt</p>

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

---

[15]For the case of flow variables with annual-quarterly or weekly-monthly data, the temporal aggregation is $y_t^h = \frac{1}{4}(x_{h,t} + x_{h,t-1} + x_{h,t-2} + x_{h,t-3})$ where $x_{h,t}$ is the quarterly (weekly) version of the annual (monthly) counterpart. Assuming that the low frequency variable occupies the second column, this option is activated by typing $options.mixed\_freq\_index = zeros(n,1);$ and $options.mixed\_freq\_index(2) = 4;$.

[16]For more details on estimating latent variables in linear state space system see also A.6.

3.7.1 Constructing 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;
```

This is how the data looks like

```
>> disp(y)
     NaN   4.6079   4.4897   4.5113   4.3645   9.0900
     NaN   4.6254   4.4909   4.5110   4.8485   9.0300
  14.5660   4.6103   4.4947   4.5127   4.9649   8.9900
     NaN   4.6210   4.4972   4.5148   5.1050   8.9400
     NaN   4.6252   4.4986   4.5159   5.2484   8.8800
  14.5712   4.6246   4.5031   4.5164   5.2192   8.8300
  ...
```

We treat `GDP` as a flow variable and 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 16 reports the median estimate of the unobserved monthly GDP in level and growth rate.

Figure 16: MF-VAR estimates of the monthly GDP level and of the monthly GDP growth
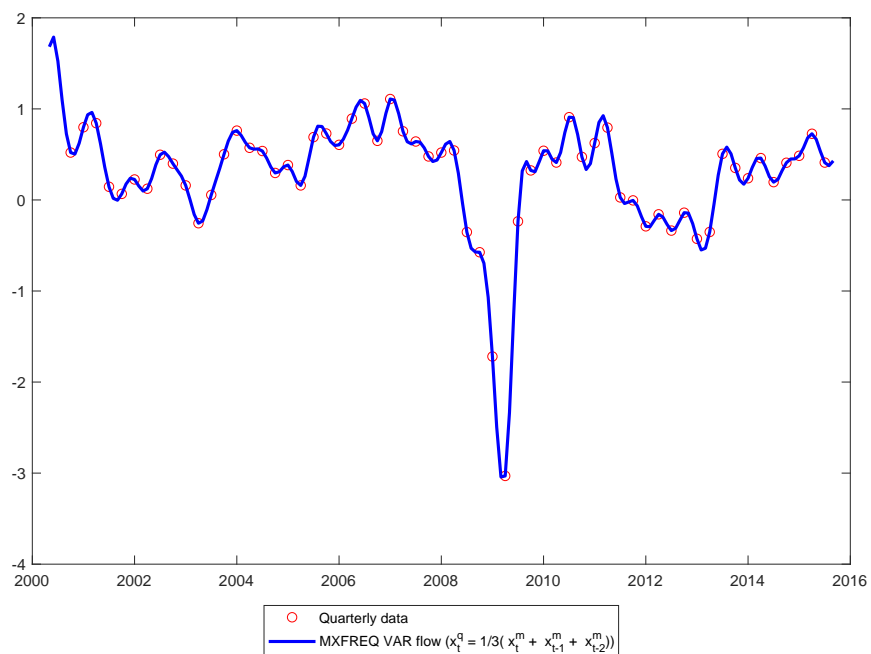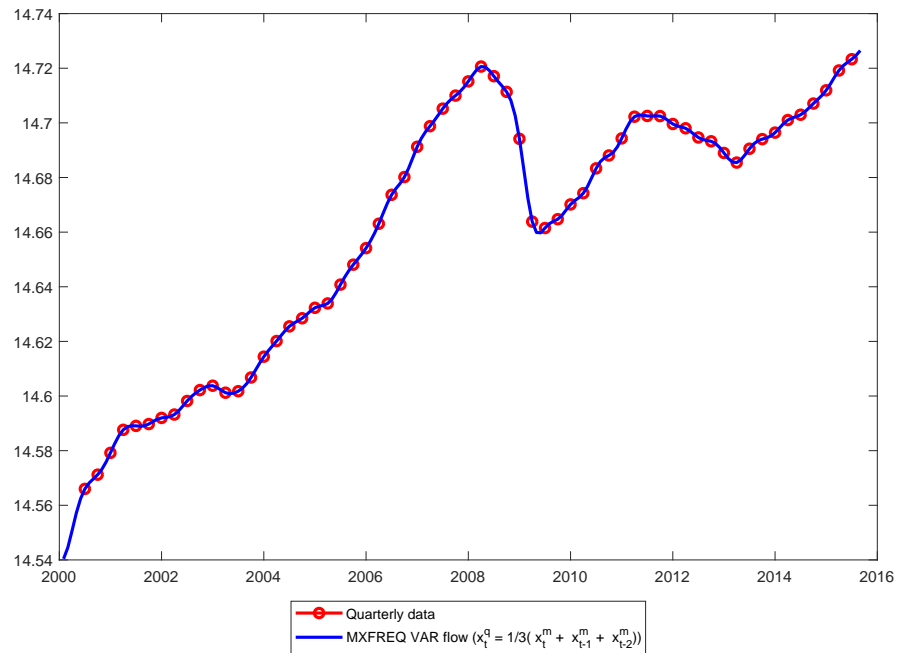
As mentioned, the maximization over the Minnesota prior hyper-parameters is not permitted in the toolbox. This is because with mixed frequency and unobserved states, the marginal likelihood cannot computed in closed form. There are two ways to discipline the choice of hyper-parameters in this context. First, one could run the prior hyper-parameters maximization using only the higher frequency variables and take the modal values of this first step to set they hyper parameters values for the MF-VAR. Alternatively, the toolkit computes the posterior kernel, $p(Y|\Sigma_{(m)}, \Phi_{(m)}, X_{(m)})$, evaluated at a specific draw (m); a $\texttt{K} \times 1$ array called $\texttt{BVAR.logL}$ 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 (1999) harmonic mean estimator, for a given value of the hyper-parameter. One could then maximize the estimated marginal likelihood over a grid of hyper parameter values. This alternative route is feasible, but obviously more time consuming.

3.8   PREDICTIONS

Regardless of the prior distribution used to estimate the parameters, one can easily generate forecasts from a VAR model. 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, that is:

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

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 to moments, quantiles, or the predictive density.

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

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

Forecasts are stored in $\texttt{BVAR.forecasts}$, which contains a number of subfields:

- $\texttt{BVAR.forecasts.no\_shocks}$ is a $\texttt{nfor} \times \texttt{n} \times \texttt{K}$ matrix containing unconditional forecasts. The first dimension corresponds to 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 the uncertainty in the parameter estimates.

- `BVAR.forecasts.with_shocks` is a $\texttt{nfor} \times \texttt{n} \times \texttt{K}$ matrix containing unconditional forecasts with shocks. The first dimension corresponds to 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 $\texttt{nfor} \times \texttt{n} \times \texttt{K}$ matrix containing the conditional forecasts, when the path for one or more endogenous variables of the VAR are specified. These conditional forecasts are generated as in 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 variables. `exo_index` could be one or more shocks. If no structural identification is performed, the program uses a Cholesky factorization by default.

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

Note that `bvar_` does allow the use of exogenous variables as long as Jeffrey or Conjugate priors are used. In this case, exogenous variables must be declared among the options

$$\texttt{options.controls = z;}$$

where `z` is a `(T+fhor)` $\times$ `n_z` array that contains the exogenous variables. A rough way to include exogenous variables in the VAR when Minnesota priors are assumed is to add them as endogenous variables and use the conditional forecast option to set their future path.

3.8.1   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 horizon, 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 `T` $\times$ `n` array of data and the `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 $\sqrt{n} \times \sqrt{n}$ subplots with the forecasts of all variables included in the VAR. Each panel displays 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:

$$\texttt{plot\_frcst\_(frcst\_to\_plot,y,T,options)}$$

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 with the name of the plot.

- `options.saveas_dir` is a string array with the directory where to save the plot. Note that figures are not saved if `options.saveas_dir` or `options.saveas_strng` are not specified.

- `options.add_frcsts` is a `(T+nfor) × 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 × 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 change multiplied by 100, i.e. $100(y_{t+12} - y_t)$. With monthly variables expressed in logs, this is the year-on-year percentage change.
  4. `4` = 4 period change multiplied by 100, i.e. $100(y_{t+4} - y_t)$. With quarterly variables expressed in logs, this is the year-on-year percentage change.
  5. `100` = period over period change multiplied by 100. With annual variables expressed in logs, this is the year-on-year percentage change.

3.8.2 A few examples

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

71

M3 (`M3`) and the nominal exchange rate (`EXRATE`). The sample runs from 2000m1 to 2015m8 (`time`). 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 the 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:in_sample_end);
```

Example 22 (Compare different point forecasts) *We compute a number of forecasts:*

1. *Unconditional forecasts, using a flat prior.*

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

2. *Unconditional forecasts, using a Minnesota prior, 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 inter-est 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 17 presents the mean forecasts with the various setups. The gray area identifies the forecasting period - September 2014 to August 2015; the top panel displays the year-on-year growth rate of industrial production; the central panel the year-on-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 an optimal Minnesota prior seem to perform better.

Example 23 (Fan Charts) *We plot the forecasts constructed using an optimal Minnesota priors (i.e. case 3 of Example 22) 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:*

- *We specify the directory where plots are saved (*`options.saveas_dir`*); the directory is created, if it does not exists.*

- *We control the appearance of the subplots, using* `options.nplots`.

- *We start the plots in 2013.*

- *We customize the names for the subplot titles.*

- *Variable transformations: for industrial production, HICP, CORE and the EXRATE we are interested in the year-on-year percentage change. Since the data is monthly and in logs,* `options.order_transform=12` *will take the 12 periods difference multiplied by 100.*

- *We plot 68% and 90% highest credible posterior bands.*

- *We add the plot the actual data realization (or of different point forecasts), using the option* `options.add_frcst`.

The appropriate commands are described below. Figure 18 presents the forecasts. The black line is the median; the light (dark) gray bands are the 68% (90%) credible sets. The blue line corresponds to the actual data.

```matlab
% 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-on-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 the actual data
options.add_frcst     = yactual;
% Plot the forecasts
plot_frcst_(frcsts,y,T,options)
```
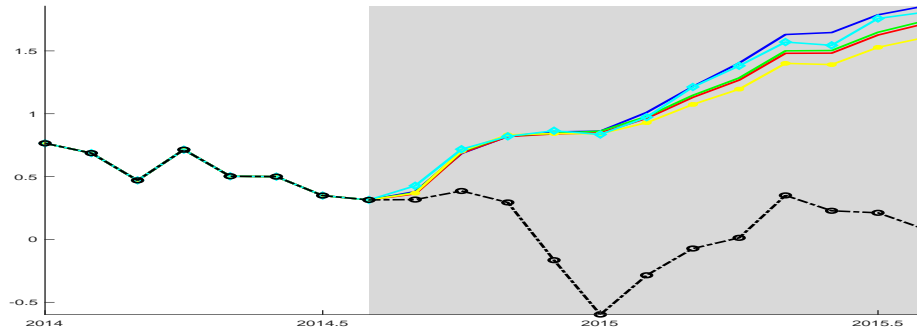
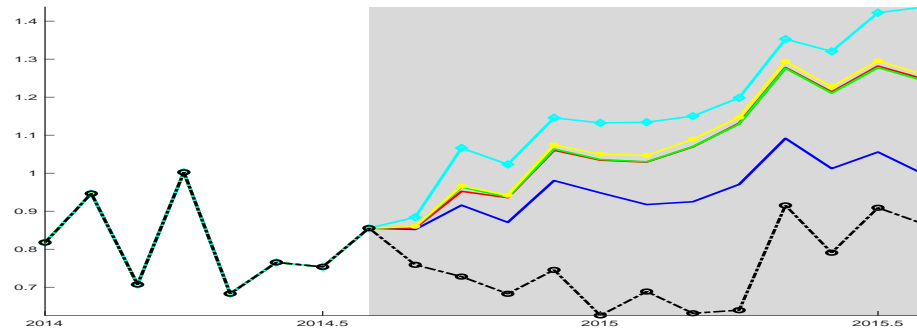# Figure 17: Forecasts with various specifications

(a) IPI
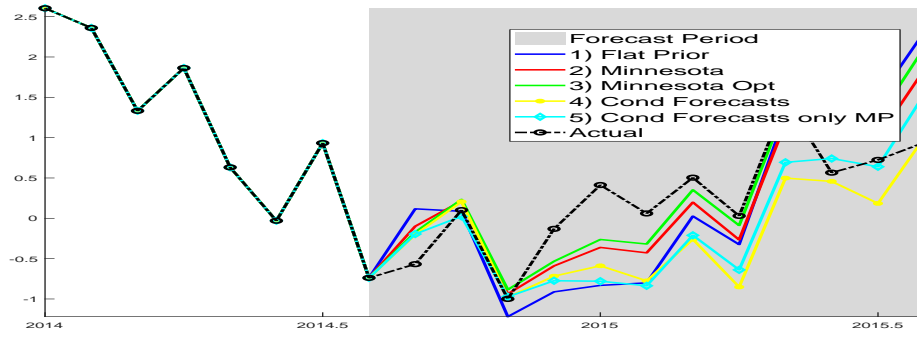
(b) HICP
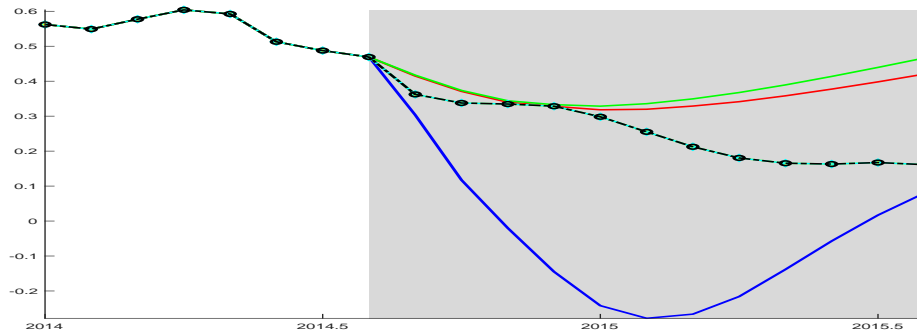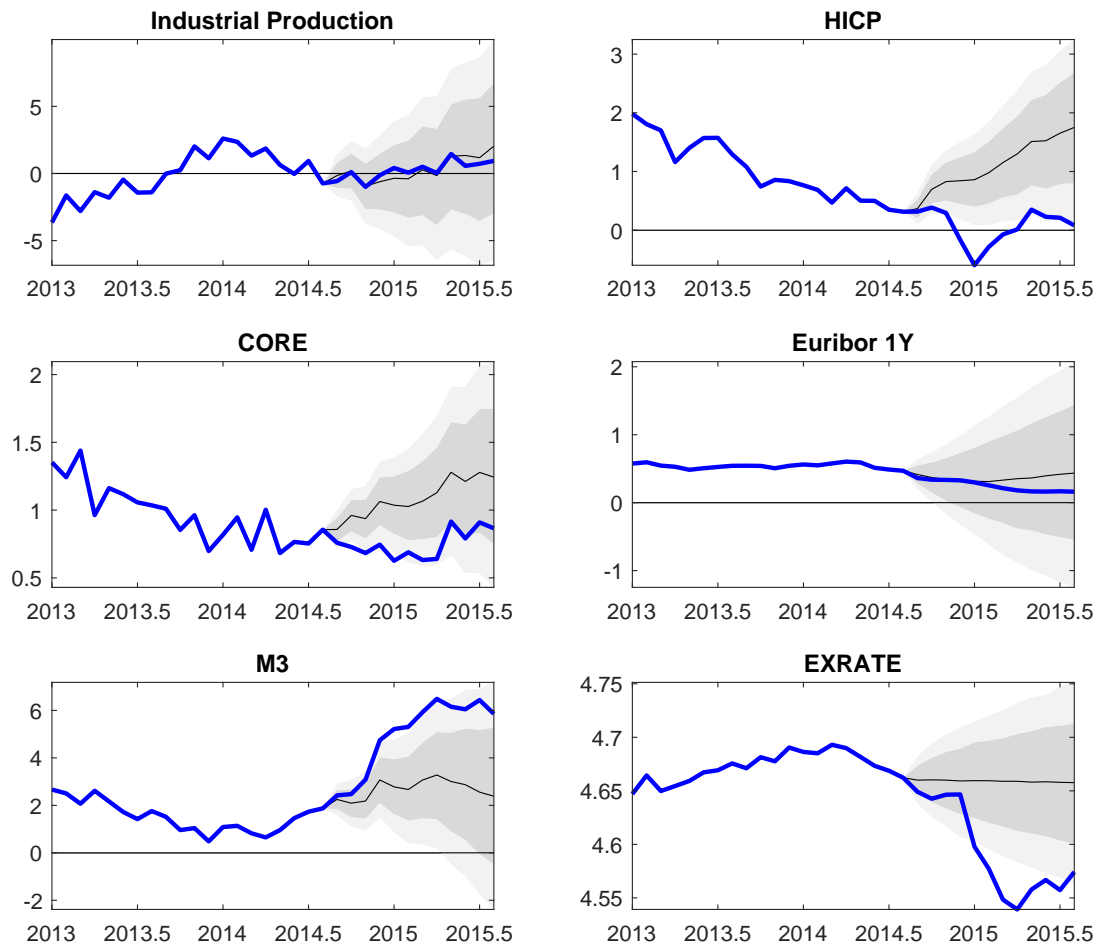

(c) CORE

(d) Euribor1Y

Figure 18: Fan charts with optimal Minnesota Prior. The black line is the median; the light (dark) gray bands 68% (90%) credible sets. The blue line corresponds to the actual data.

3.9 Panels of VAR models

The study of the macroeconomic propagation of shocks sometimes involves the use of different units of the economy, different sectors, different regions (see e.g. Canova and Pappa (2007)), different individuals, different banks (see e.g. Altavilla, Canova and Ciccarelli (2020)), or different countries. In these situations, the researcher has to take a stand on how to treat the cross-sectional dimensional of the panel. The modeling choice boils down to decide whether ones wants to (i) treat the units separately, (ii) pool the units together, or (iii) *partially* pool the units. An alternative approach would be to compress somehow the information contained in the cross-section using a principal component methods, see section 3.10 for details.

If T is large, the average of the responses estimated unit by unit (the so-called average time series estimator) is consistent for the mean response; this turns out to be the best strategy to use with long enough times series. When T is small, the cross section is large

and homogeneous, one could combine the cross section and time series dimensions and obtain estimates using the pooled data (the pooled estimator). A pooled estimator is consistent for the mean response only if the data features dynamics homogeneity; otherwise, the estimate is biased and instrumental variable approaches are unlikely to work because it is difficult to find instruments which are uncorrelated with the innovations and correlated with the regressors. When T is small and the homogeneity assumption is either difficult to verify or untenable, one could partially pool the data using cross-sectional prior specifications, for example an *exchangeable* prior, see Canova (2007).

Note also that while a pooled estimator only provides an estimate for the mean response, the average time series and the partial pool approach provide estimates for the mean response and for the responses of each separate unit and this allow researches to study in more details the reasons of potential heterogeneities in the cross section.

We discuss how to implement these alternative approaches with a series of examples. In all of them, we consider bank level data on lending and deposit rates and study the transmission of a lending shock to lending and deposit rates. `DataBanks` is a monthly panel with 100 data points for 50 banks, where `LendingRate` and `DepositRate` are 100×50 arrays of data, containing respectively lending and deposit rates.

Example 24 (Average time series estimator) *We run separate bivariate VAR models on bank lending and deposit rates, unit by unit. We assume two lags. First, we load the data (`DataBanks`); and define a bank specific dataset with lending rate ordered first followed by the deposit rate. To speed up the calculation, we limit attention on the response functions obtained with OLS estimates of the autoregressive parameters and of covariance matrix, stored in the matrix `BVAR.ir_ols` with the usual ordering of the dimensions (i.e. variable, horizon, and shock). We then collect the banks specific responses. Figure 19 reports the median (black), mean (blue) and the numerical dispersion produced by a normalized lending shock across the 50 banks. Identification of lending shocks is achieved with a Cholesky decomposition, with lending rate ordered first.*

```
% load the data
load DataBanks
[T,NBanks]  = size(LendingRate);
% Estimate a bivariate VAR for each unit
lags    = 2; % Number of lags
opt.K   = 1; % Draws from the posterior (=1 since they are not needed)
for i = 1 : NBanks
  % construct the banks i database
  yi  = [LendingRate(:,i) DepositRate(:,i)];
  % estimate a VAR(2)
  bvar0 = bvar_(yi,lags,opt);
```

```matlab
  % store the Cholesky IRFs
  irfs_to_plot(:,:,:,i) = bvar0.ir_ols;
end
% Customize the plot
% variables names
options.varnames   = {'Lending Rate' 'Deposit Rate'};
% name of the directory where the figure is saved
options.saveas_dir  = './panels_plt';
% names of the figure to save
options.saveas_strng = 'TSAverageIRF';
% name of the shock
options.shocksnames   = options.varnames;
% additional HPD set
options.conf_sig_2   = 0.95;
% plot appeareance
options.nplots = [2 2];
% add mean response
options.add_irfs = mean(irfs_to_plot,4);
% the plotting command
plot_all_irfs_(irfs_to_plot,options);
```

Figure 19: Average (blue), median (black) dynamic transmission of a normalized lending shock across 50 banks with dispersion bands.
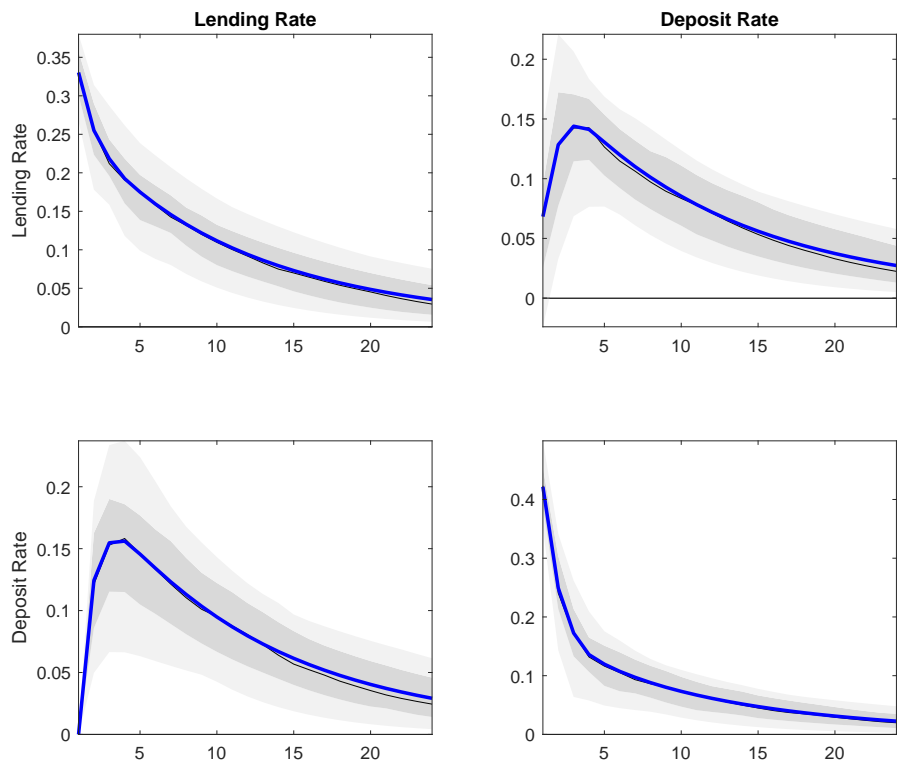
Example 25 (Pooling (with fixed effects)) *Using the same data as in example 24, we estimate the mean transmission of a lending shock, assuming homogeneous dynamics and fixed effects. Figure 20 reports the estimated dynamic transmission of a lending shock using the pooled estimator (black line). The blue line is the average time series estimate previously computed. The mean dynamics are similar, suggesting that dynamic heterogeneity is not very important. The pooled estimator with fixed effects is more precise, as expected under dynamic homogeneity, since it efficiently uses the NT observations.*

```
%   remove fixed effects and reorder the data
LendingRate_ = reshape(demean(LendingRate),T*NBanks,1);
DepositRate_ = reshape(demean(DepositRate),T*NBanks,1);
% pool the data
ypooled = [LendingRate_ , DepositRate_];
% Estimate one bivariate VAR with the data from all units
lags    = 2;
bvar1   = bvar_(ypooled,lags);
% IRFs
irfs_to_plot     = bvar1.ir_draws;
% name of the figure to save
options.saveas_strng = 'Pooling';
% the plotting command
plot_all_irfs_(irfs_to_plot,options);
```

Figure 20: The dynamic transmission of a lending shock using a pooled estimator with fixed effects. The blue line is the average time series estimators. Confidence bands are in grey
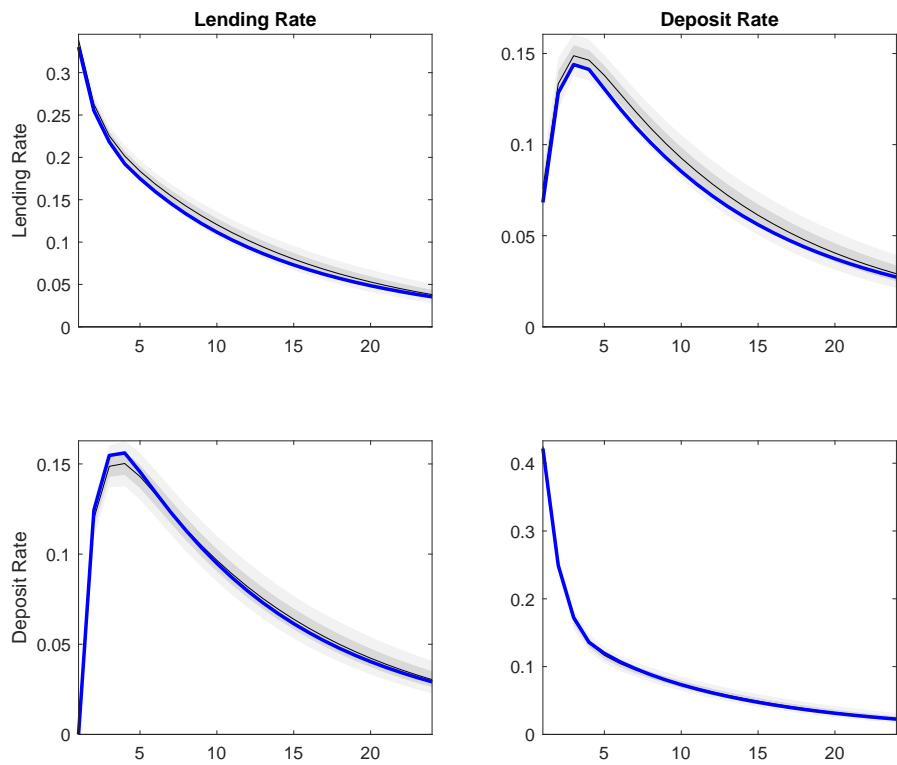
Example 26 (Partial pooling with a data-based prior) *Using the same data of example 24, estimate the transmission of a lending shock in a VAR with dynamic heterogeneity, using a prior that favors homogeneity. We employ a subset of units to estimate the prior first two moments and use the remaining units to estimate the propagation of a lending shock. For illustration, we focus on the propagation for bank number 50. Rather than using the identify matrix in* `options.priors.Phi.cov` *one could also use the dispersion matrix of* (`bvar2.Phi_draws` *to scale the prior.*

Figure 21 reports the median (black line) dynamic transmission of a lending shock for the 50th bank using the constructed prior.

```
% Construct the prior, pooling the first 20 units
N1       = 20;
LendingRate_ = reshape(demean(LendingRate(:,1:N1)),T*N1,1);
DepositRate_ = reshape(demean(DepositRate(:,1:N1)),T*N1,1);
yp       = [LendingRate_ , DepositRate_];
bvar2    = bvar_(yp,lags);
% Activate a conjugate prior
options.priors.name   = 'Conjugate';
% Center the prior at the pooled estimator
options.priors.Phi.mean = mean(bvar2.Phi_draws,3);
% covariance matrix
% if gam large you get OLS; if it is small, you get close to perfect pooling.
gam = 0.1;
options.priors.Phi.cov = gam * eye(size(options.priors.Phi.mean,1));
options.K = 1000;
% recover banks 50 (=NBanks) in database
i = NBanks;
yi = [LendingRate(:,i) DepositRate(:,i)];
% estimate a bivariate VAR for bank i with the databased prior
bvar_e = bvar_(yi,lags,options);
options.saveas_strng = ['BayesianPartialPoolingBank#' num2str(i)];
plot_all_irfs_(bvar_e.ir_draws,options)
```

There are many other alternatives one can use to tune up the prior for a panel of VAR models. For example, when aggregate data is available, one could use it to form a prior and then shrink the unit specific estimates as we have done above. Note that the toolbox only allows priors with fixed parameters (which are estimable from a training sample). Fully hierarchical priors are not currently supported. From experience, we can say that fully hierarchical estimates nd standard posterior estimates, constructed conditional on a carefully tuned prior, are very similar.

Figure 21: Median and credible sets for the dynamic transmission of a lending shock for the 50th bank when Bayesian partial pooling is used. The blue line is the average time series estimator.
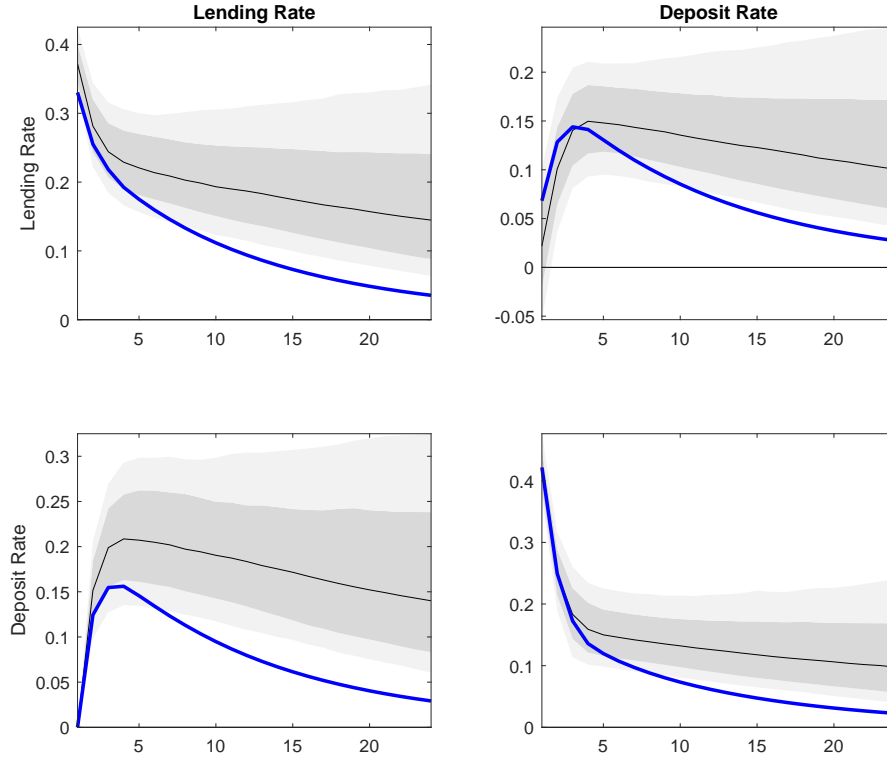
The important underlying assumption we have used in this section is that units can be treated independently of each other. Thus, current and lagged cross unit dynamics are unimportant. Clearly, if the focus of investigation are regions of a country or nations in a monetary union, such an assumption is untenable. One could use aggregate control variables to try to absorb cross unit interdependences. But this is just a short cut. A better way to approach the transmission problem is to use a panel VAR model with full interdependences. This is just a large scale VAR model, where the dimension of the vector of endogenous variables is N*G where G is the number of variables per unit. Canova and Ciccarelli (2009) provide a convenient reparameterization of the such a large scale VAR model which makes it possible to estimate it by standard Bayesian methods when T is moderate or small. A full panel VAR is not currently supported in the toolbox.

3.10   FAVARs

When n, the number of variables one would like to include in the VAR, is large, and T is short, running a VAR may be problematic, both because of low sample information and of identification difficulties. In this case, let $y_t = [y_{1t}, y_{2t}]$. y_{2t} contains data excluded from the VAR which is going to be used to construct factors w_t. Then, one can keep the VAR sufficiently small by employing $\tilde{y}_t = [y_{1t}, w_t]$ as observables. This choice avoids overparameterization but allows the VAR to span the full information set available to researchers. 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}, see section 5.2, 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 the data generating process for y_2 is well described by a static factor model:

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

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 one to extract the first principal components from a time series database[17]:

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

The syntax of this function is the following: `y_2` contains the variables excluded from the VAR, `n_w` is a scalar, indicating the number of factors to be extracted, and `transf` indicates the data transformation to be used, where 0 means no transformation, 1 means demean, and 2 means demean and standardize. `E, W, Lambda` are, respectively, the idiosyncratic errors, the principal components, and the loadings. Typically, `y_2` is assumed to be stationary and standardized to make the units of the n_2 variables comparable; when `transf=2`, `STD` is 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 3.1, including the maximization over the Minnesota prior hyper-parameters, can be used to estimate FAVAR parameters. One can also construct impulse response functions as discussed in section 3.2, and point or density forecasts, as discussed in section 3.8.

Because the transformed (and standardized) variables W 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

---

[17]See section 4 for more details on principal components and factor models.

factor loadings, the standard deviations of the `y_2` variables, the relative dimension of `y_1`, `y_2` and `W`. Let `n_1` be the number of variables in `y_1`, the following function allows to construct the matrix of rescaling coefficients:

$$\texttt{ReScale = rescaleFAVAR(STD,Lambda,n\_1)}$$

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

$$\texttt{ReScale = rescaleFAVAR(STD,Lambda,n\_1, 2)}$$

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, assuming that one is interested in the mean forecast of `y_2`, we can type

$$\texttt{mean\_forecast\_y = mean(fabvar.forecasts.no\_shocks,3) * ReScale';}$$

which contains the forecast of both `y_2` and `y_1`. Analogously, one could 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 27 (FAVAR: Responses of y_2 variables to monetary policy shocks) *We 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 these 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 with a recursive identification scheme, where slow moving variables are ordered first. We trace the responses of real GDP growth and core PCE inflation to the identified shock.*

```
% 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
```

```matlab
lags  = 2;
fabvar = bvar_(y,lags);
% Rescale the estimates from (3 PC 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;  % index for the shock
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
irfs_to_plot = fabvar.irX_draws( indx_var, :, 1, :);
% Customize the IRFs 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 28 (FAVAR with sign restrictions on y_2) *Consider the same setting of exercise 27. 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, of the deflator, and of core personal consumption expenditure (PCE). The commands to perform this exercise are described below and Figure 22 reports the responses to a sign-identified aggregate supply shock.*

```matlab
% Use sign restrictions on the uncompressed variables.
% Identification: 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);
```

```matlab
    % 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 IRFs 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
% the plotting command
plot_irfs_(irfs_to_plot,options)
```

Figure 22: Dynamics of real GDP, of GDP deflator, and of Core PCE in response to a sign-identified aggregate supply shock.
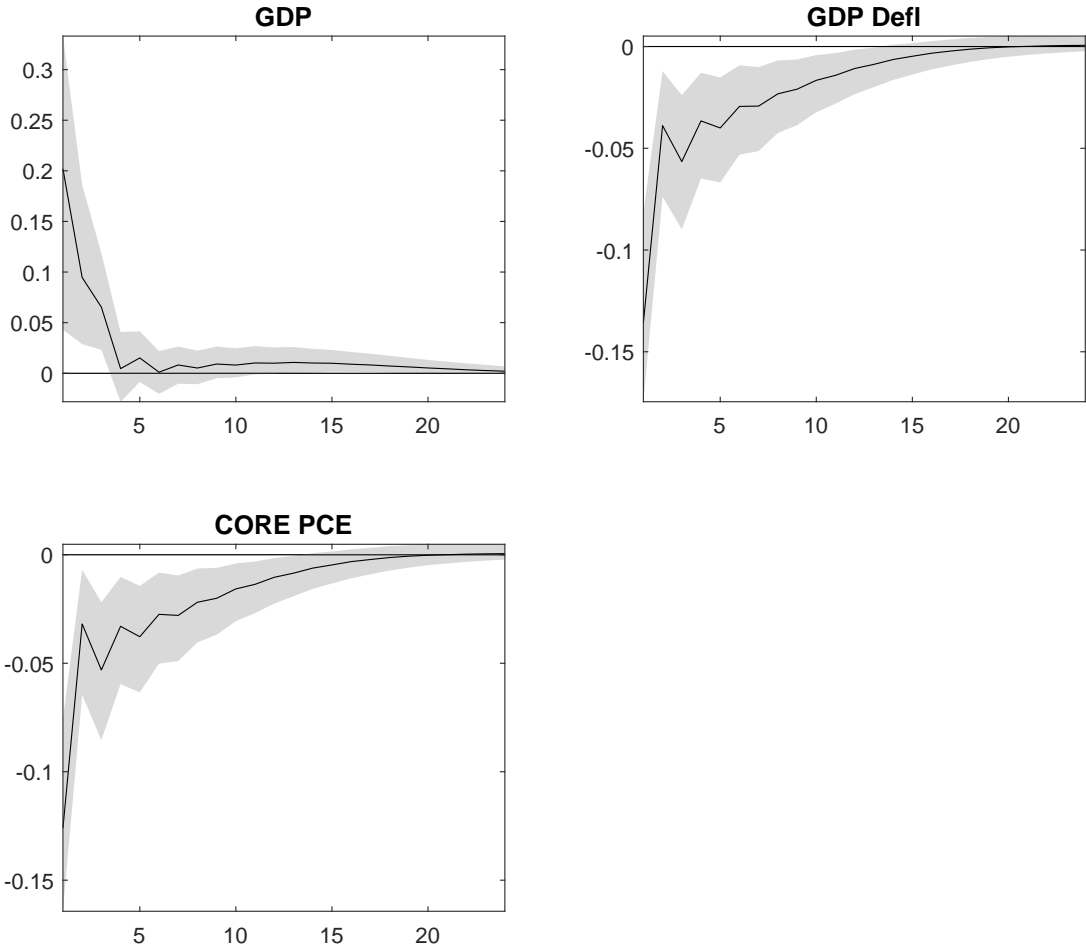
4   Factor models

VAR models typically contain less variables than one would like to consider, and researchers often face the standard trade-off between dimensionality and estimability. Factor models reduce the dimensionality problem but maintain the information content of the data. The idea is that a few factors explain the common portion of the variability of macro and financial variables is well established. The scope of factor models is to find a few indicators which are responsible for the comovements of a set of cross sectionally correlated time series. Factor models are therefore alternatives to the (panel) VAR models analysed in previous sections. In the following subsections, we discuss first static principal components; and then static and dynamic factor models.

4.1   Principal Components

Principal Components (PC) are linear combinations of observable variables with the property that they maximize the explained portion of the variance of the observables. Formally, let Y a $T \times n$ matrix of random variables independently distributed across time with covariance Σ_y. We want to find the linear combination $Y\delta$, where δ is a $n \times m$ vector, such that the variance of $Y\delta$ (i.e. $E(\delta'Y'Y\delta) = E(\delta'\Sigma_y\delta)$) is maximized subject to $\delta'\delta = I$. Note that without normalization, the solution would be $\delta = \infty$. PC analysis produce a reduced rank representation of the data.

PCs are easy to compute: one just needs to find the eigenvectors and eigenvalues of $Y'Y$, take as PC the first m eigenvectors multiplied by Y. PCs depend on the units in which variables are measured; therefore, it is preferrable to standardize Y if variables are measured in different units. If the elements of Y are nearly orthogonal, each PC explains about $1/m$ of the total variance.

The function `pc_T.m` allows one to extract the first principal components from a database:

```
[e, f, Lambda, EigenVal, STD] = pc_T(y, m, transf);
```

The syntax of the function is the following: `y` contains the variables we wish to compress, `m` is a scalar, indicating the number of factors to be extracted, and `transf` indicates the data transformation to be used, where 0 means no transformation, 1 means demean, and 2 means demean and standardize. `e, f, EigenVal, Lambda` are, respectively, the idiosyncratic errors, the PCs, the eigenvalues associated to the PCs, and the loadings. Typically, `y` is assumed to be stationary and standardized; when `transf=2`, `STD` is the $n \times 1$ vector containing the standard deviation of `y`; otherwise, it is a vector of ones.

There are a number of procedure to select the number `m` of principal components. An informal procedure typically used is to study the structure of the eigenvalues. Since eigenvalues are typically reported in decreasing order, one can select a desired fraction of total variance to be explained and include as many factors as it is necessary to reach that level (recall that

the fraction of variance explained by first m factors is $\frac{\sum_{i=1}^{m} \lambda_i}{trace(\Sigma)}$). There is no agreement in the profession on how much variance should be explained (usually 80, 90%, but at times 50% is used). Sometimes it is recommended to include those factors with $\lambda_i > 1$, since these factors may explain more than the "average factor" or to look at scree plots. More formal procedures include a score test and the Bai and Ng (2002) information criteria.

4.2   STATIC FACTOR MODELS (SFM)

A model with m static factors can be represented as:

$$y_t = \Lambda f_t + e_t$$

where y_t is a $n \times 1$ vector of observable variables, f_t is a $m \times 1$ random vector of common factors with zero mean and covariance matrix $\Sigma_f = E(f_t f_t')$; e_t is a $n \times 1$ random vector of idiosyncratic errors with zero mean and diagonal covariance matrix $\Sigma_e = E(e_t e_t')$; Λ is the $n \times m$ matrix of factor loadings. We assume that $n > m$. Since e's are cross-sectionally independent, all covariations in y are due to the common components. Factors could be observable or latent. When they are unobservable, they can be estimated with principal component techniques or Bayesian methods as we will describe next.

Latent factors and loadings are not separately identified since $\Lambda f_t = \Lambda C C^{-1} f_t = \Lambda^\star f_t^\star$. As in VAR models, one needs an order condition so that the number of free parameters in the covariance of y_t is greater or equal to the number of parameters to be estimated. The number of free parameters in the data is $n(n+1)/2$. The number of parameters to be estimated is $nm + m(m+1)/2 + n$. One scheme that generates (over-)identification is to assume that factors are orthogonal and standardized, i.e. $\Sigma_f = I$. Otherwise, one can assume that the $m \times m$ sub-matrix of Λ is normalized to have ones on the main diagonal and zeros on the upper triangular part. The toolbox uses both schemes for the static factor model and the latter for the dynamic factor model.

If the latent factors and the idiosyncratic error terms are normally distributed, the static factor model can be easily estimated with Bayesian methods. In particular, the toolkit generates draws from the marginal distribution of the parameters using the Gibbs sampler (see section A.6 in the Appendix for details). By default, the toolkit uses a Multivariate Normal prior for Λ with zero mean and with a diagonal covariance matrix of 10; and an inverse Gamma distribution is assumed for the standard deviations of the error terms with unitary scale and four degrees of freedom. These default settings can be changed, as discussed in the next section. The draws obtained after the convergence threshold is reached are from the desired marginal posteriors.

Since the static factor model is a special case of the dynamic factor model, we describe the estimation function valid for both models in the next subsection.

4.3 Dynamic factor models (DFM)

A model with m dynamic factors, each featuring p lags, can be represented as:

$$y_t = \Lambda f_t + e_t$$
$$f_t = \Phi_1 f_{t-1} + ... + \Phi_p f_{t-p} + u_t$$

where the measurement equation parameters are the same as in section 4.2; Φ_j with $j = 1, ..., p$ are $m \times m$ matrices; u_t is an i.i.d. normally distributed, zero mean, random vector with covariance matrix Σ_u.

The toolkit allows to estimate $\vartheta = [\Lambda, \Sigma_e, \Phi_1, ..., \Phi_p, \Sigma_u, \{f_t\}_{t=1}^T]$ and to conduct inference conditional on the estimates of ϑ. The baseline estimation function is

```
BDFM = bdfm_(y, lags, m, options)
```

The first input, `y`, is the data, an array of size `T × n`. The second input, `lags`, is the number of lags of the dynamic factors, a non-negative integer; when `lags = 0` the toolkit computes estimates for the static factor model (see section 4.2). The third input, `m`, is the number of factors, a number larger than zero. The fourth input, `options`, specifies the options; it can be omitted if default options are employed. Data are demeaned by default, unless otherwise specified (use `options.do_the_demean=0`).

When the factors and the idiosyncratic error terms are normally distributed, the dynamic factor model can also be estimated with Bayesian methods. In particular, the toolkit generates draws from the marginal distribution of the parameters using the Gibbs sampler (see section A.6 in the Appendix for details). By default, the toolkit assumes a multivariate Normal prior for Λ with zero mean and with a diagonal covariance matrix of 10; and an inverse Gamma prior for the standard deviations of the error term in the measurement equation with unitary scale and four degrees of freedom. For the parameters governing the dynamics of the factors, the default is a multivariate Normal prior for $\Phi_1, ..., \Phi_p$ with zero mean (except for the first own lag, which is set at 0.2) and a diagonal covariance matrix of 10. The factor residual covariance matrix is assumed to be a-priori distributed as an inverse Wishart with unitary diagonal scale and `m+1` degrees of freedom. These default values can be changed, as discussed next.

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

- `BDFM.Phi_draws` is a $(m \times$ `lags`$) \times m \times$ `K` matrix containing `K` draws from the posterior distribution of the factors autoregressive parameters, Φ_j, assuming a MN-IW prior. For each draw `k`, the first $m \times m$ submatrix of `BDFM.Phi_draws(:,:,k)` contains the autoregressive matrix of order one; subsequent submatrices higher order lags.

- `BDFM.Sigma_draws` is a $m \times m \times$ `K` matrix containing `K` draws from the posterior distribution of the covariance matrix of the factor innovations, Σ_u.

91

- `BDFM.lambda_draws` is a n × m × K matrix containing K draws from the posterior distribution of the factor loading, Λ.

- `BDFM.sigma_draws` is a n × K array containing K draws from the posterior distribution of the variance of the error terms, Σ_e.

- `BDFM.f_draws` is a T × m × K matrix containing K draws of the factors, obtained with the Carter and Kohn backward smoothing algorithm.

- `BDFM.f_filt_draws` is a T × m × K matrix containing K draws of the factors, obtained with the Kalman filter forward recursions.

- `BDFM.ef_draws` is a T × m × K matrix containing K draws of the factor innovations.

- `BDFM.eg_draws` is a T × n × K matrix containing K draws of the measurement error terms.

- `BDFM.ir_draws` is a four dimensional object (i.e. n × hor × m × K matrix) that collects the impulse responses obtained with a recursive identification on the innovations of the factors. 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 factor parameters. Alternative identification schemes are possible, see Section 4.4 for the implementation.

- `BDFM.forecasts.no_shocks` (`BDFM.forecasts.with_shocks`) is a three dimensional object (i.e. fhor × n × K matrix) that collects the forecasts assuming zero (non-zero) shocks in the factors 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.

The default value for K is 20,000. It can be changed by setting `options.K` to any positive desired number, prior to calling the `bdfm_` routine.

As mentioned, the priors parameters can be modified using appropriate options:

Φ: `options.priors.F.Phi.mean` is a (m × lags) × m matrix containing the prior means for the autoregressive parameters of the factors.

Φ: `options.priors.F.Phi.cov` is a (m × lags) × (m × lags) matrix containing the prior covariance for the autoregressive parameters of the factors.

Σ_u: `options.priors.F.Sigma.scale` is a (m × m) matrix containing the prior scale of the covariance of the factor innovations.

Σ_u: `options.priors.F.Sigma.df` is a scalar defining the prior degrees of freedom for the factor innovations.

Φ, Σ_u: `options.priors.name = 'Jeffrey';` activate a Jeffrey prior on Φ, Σ_u, see A.3

Λ: `options.priors.F.Lambda.mean` is a scalar defining the prior mean for the factor loadings.

Λ: `options.priors.F.Lambda.cov` is a scalar defining the prior variance for the factor loadings.

Σ_e: `options.priors.G.Sigma.scale` is a scalar defining the prior scale of the variance of the error terms.

Σ_e: `options.priors.G.Sigma.df` is a scalar defining the prior degrees of freedom for the variance of the measurement error terms.

One can set these options one at the time or jointly.

The number of dynamic factors can be selected using the approach proposed by Bai and Ng (2007). The approach consists of estimating first a VAR in r static factors, where the factors are obtained by applying the method of principal components to a large panel of data, then computing the eigenvalues of the residual covariance (or correlation) matrix and testing the rank of the matrix.

Sometimes the estimation of factor models can be time consuming; it is difficult to give general suggestions on how to improve the estimation speed, as the solution tends to be case specific and to depend on the data used. Below we offer few applied tips that might be useful in some circumstances.

- If the times series dimension is large and the cross section medium size, say $T = 200$ data and $N \approx 50$, the best approach is to use the default estimation options, keeping in mind that the larger the cross-section is, the slower the codes become.

- If the times series dimension is very large and the cross section is small, say $T > 1,000$ and $N < 10$, one could estimates the latent dynamic factors using the steady state Kalman Gain in the recursion of the Kalman filter (see section A.6). For example, such situation could arise when working with various years of daily data of 5/8 interest rate future contracts which are typically thought as driven by three factors, i.e. an intercept, a slope and a curvature factor. Using the steady state Kalman Gain avoids inverting the prediction error covariance matrix to construct the updated Kalman Gain (with five years of daily data this is $5 \times 365 \times 20,000$ times) as the former is computed outside of the time loop. One can activate the steady state Kalman Gain option by setting it before the estimation command

```
options.use_the_ss_kalman_gain =  1;
```

- If the cross section is very large, i.e. $N > 100$, the Law of Large Number applies and one can treat cross-section averages as data. One would then estimate a VAR on the cross section averages and use OLS regressions in the measurement equation to recover the dynamics of the original observables. For example, assume that data are a combination of regional and sectorial data- say 5 regions and 5 sectors - and we have sufficient data for each category. One then can construct 11 factors: one global factor obtained as the simple mean of all data; 5 sectorial factors constructed as the (simple

93

or weighted) mean of sectorial data; and 5 regional factors constructed as the (simple or weighted) mean of regional data. As long as we have a sufficiently large number of observations for each subgroup, the means are consistent estimates of the underlying factors.

4.4   Computing Impulse responses

Dynamic factor models can be used to study how certain disturbances to the factors are dynamically transmitted to macroeconomic aggregates. To study the transmission of interesting disturbances, we need, as usual, to impose some structure. Once this is done, one can trace out the causal effects of structural disturbances in the usual way.

Structural economic disturbances ν_t can be mapped into the innovations to the factors, u_t, by means of orthonormal rotation matrices, much in the same way as one does it for VAR models. Hence, the identification schemes discussed in section 3.2 can be used. In particular, the toolkit allows for:

- Choleski or recursive identification (default). The responses obtained with this identification scheme are stored in `BDFM.ir_draws`.

- Long-run identification. Activated with the option:

$$\texttt{options.long\_run\_irf = 1;}$$

  Responses obtained with this identification scheme are stored in `BDFM.irlr_draws`. For more details see section 3.2.2

- Sign and magnitude restrictions identification. Activated with the option:

$$\texttt{options.signs\{1\} = 'y(a,b,c)>0'}$$

  Responses obtained with this identification scheme are stored in `BDFM.irsign_draws`. For more details on the syntax see section 3.2.3

- Narrative restrictions identification. Activated with the option:

$$\texttt{options.narrative\{1\} = 'v(tau,n)>0'}$$

  Responses obtained with this identification scheme are stored in `BDFM.irnarrsign_draws`. For more details on the syntax, see section 3.2.4

- External instrumental or proxy variable identification. Activated with the option:

$$\texttt{options.proxy = instrument;}$$

  Responses obtained with this identification scheme are stored in `BDFM.irproxy_draws`. For more details on the syntax, see section 3.2.6

All the matrices storing the impulse responses are four dimensional objects (i.e. $n \times \text{hor} \times m \times K$ matrix). 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.

Numerical approximations to the distribution of the impulse response functions can be computed and visualized using the function `plot_irfs_.m`; see section 3.2.9 for more details.

4.5 A CONTROLLED EXPERIMENT

We create an artificial database consisting of $N = 50$ variables with $T = 300$ generated from a dynamic factor model with one lag. The following generates the desired database and computes the responses of the endogenous variables to shocks in the factors.

```
% Fix the seed
rng('default');        rng(999);
% Setting the parameters
% AR and Cov factors
Phi    = [0.75 0; 0.1 0.8]; Sigma  = [1 -0.2; -0.2 2]; Q =  chol(Sigma,'lower');
% factor loadings: (50 x 2) matrix with choleski structure
Lambda = [1 0; 0.3 1; 0.5 -1.5; -1.1 -0.5; 1.5 1.5; 5*randn(45,size(Phi,1))];
% persistence  and StD of idiosyncratic errors
rho    = 0;        sig    = 1;
% preallocating memory
% sample length
T      = 301;
f = zeros(T,size(Phi,1));    y = zeros(T,size(Lambda,1));
e = zeros(T,size(Lambda,1));   i = zeros(T,size(Phi,1));
 % Loop
for t = 1 : T-1
    % factor innovations
    i(t+1,:) = randn(size(f,2),1)';
    % factors
    f(t+1,:) = (Phi*f(t,:)' + Q*i(t+1,:)')';
    % idiosyncratic errors
    e(t+1,:) = (rho*e(t,:)' + sig*randn(size(Lambda,1),1))';
    % observed data
    y(t+1,:) = (Lambda*f(t+1,:)' + e(t+1,:)')';
end
f(1,:) = [];
y(1,:) = [];
% Compute true IRFs
```

```
hor = 24;
true    = iresponse(Phi,eye(size(f,2)),hor,Q);
Lam_    = repmat(Lambda,1,1,size(f,2));
for ff = 1 : size(f,2)
    ir_true(:,:,ff) = Lambda * true(:,:,ff);
end
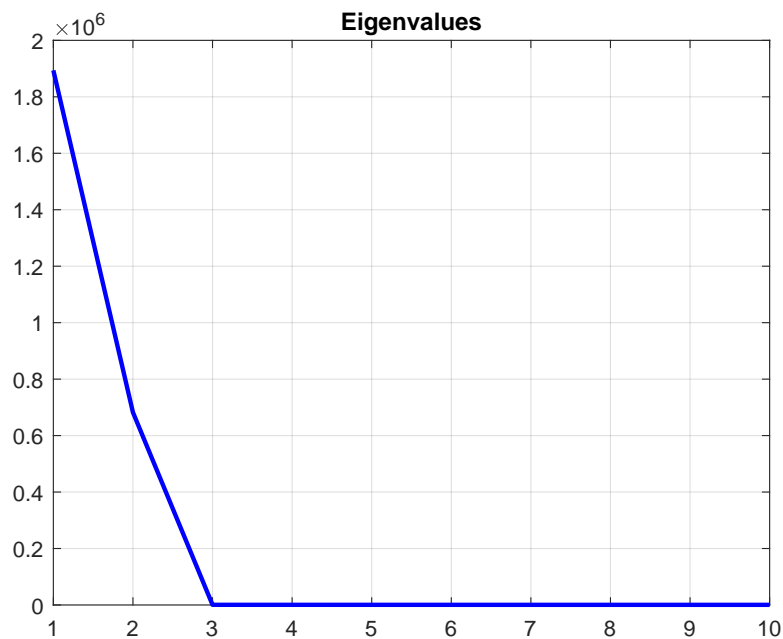```

**Example 29 (Principal Components and scree plot)** *We extract the first two principal components from the artificial data* y *and plot the first ten eigenvalues (scree plot). The commands are:*

```
nfac   = 2; transf = 0;
[~,pc,~,egv,~] = pc_T(y, nfac, transf);
% scree plot
figure, plot(egv(1:10),'b'); ggrid 'on');
```

Figure 23: Scree plot

**Example 30 (Static factor model)** *We estimate a static factor model on the artificial data* y, *assuming two factors. The prior for* $\Lambda$ *has mean is zero and the variance is 6. We plot the estimated factors against the 'true' ones (i.e.* f*). We burn the first 5,000 draws and plot one every 20 draws of the remaining draws. The commands are:*

```matlab
% 2 static factors
nfac = size(f,2);
lags = 0;
% priors
options.priors.F.Lambda.mean = 0;
options.priors.F.Lambda.cov  = 6;
% estimation command
[BSFM] = bdfm_(y,lags,nfac,options);
% consider a subset of the draws after the 5000th
index = 5000:20:size(BSFM.Phi_draws,3);
% plot estimates and true factors
figure,
for gg=1:nfac
    subplot(nfac,1,gg)
    % draws of factors, take all of them
    plot([squeeze(BSFM.f_draws(:,gg,index))],'Color',[0.7 0.7 0.7])
    hold on
    % true factors
    plot(f(:,gg),'b','Linewidth',2)
    hold on
    % mean across draws
    plot(mean(BSFM.f_draws(:,gg,index),3),'k','Linewidth',2)
end
```

Figure 24: Estimated static factors. The blue line represents the true factors. The black line the average estimate. The grey area represents the dispersion of the estimates
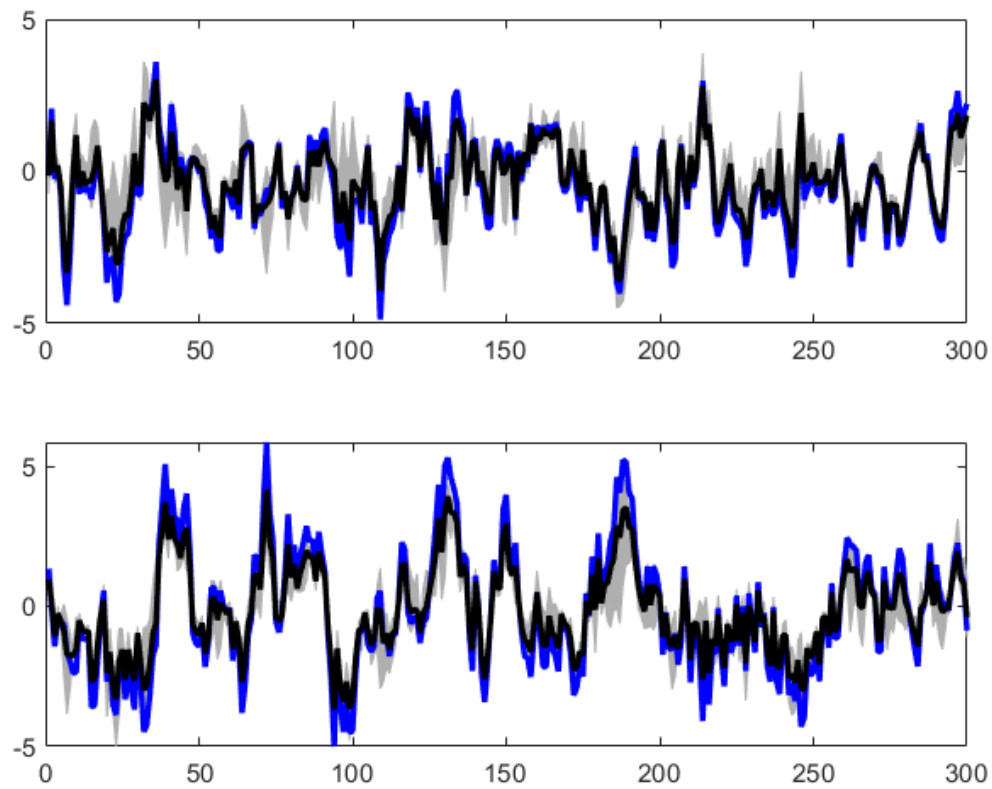
Example 31 (Dynamic factor model) *We estimate a dynamic factor model on the artificial data* y, *assuming two factors and one lag. We again plot the estimated factors against the 'true' ones (i.e.* f*). The estimated factors are computed burning the first 5,000 draws and considering one every 20 of the remaining draws. The commands are:*

```matlab
% 2 dynamic factors with one lag
nfac   = 2;
lags   = 1;
% estimation command
[BDFM] = bdfm_(y,lags,nfac);
% consider a subset of draws after the  5000th
index = 5000:20:size(BDFM.Phi_draws,3);
figure,
for gg=1:nfac
    subplot(nfac,1,gg)
    % draws of factors
    plot([squeeze(BDFM.f_draws(:,gg,index))],'Color',[0.7 0.7 0.7])
    hold on
    % true factors
    plot(f(:,gg),'b','Linewidth',2)
    hold on
    % mean across draws
    plot(mean(BDFM.f_draws(:,gg,index),3),'k','Linewidth',2)
end
```

Figure 25: Estimated dynamic factors. The blue line represents the true factors. The black line the average estimate. The grey area the dispersion of the estimates.
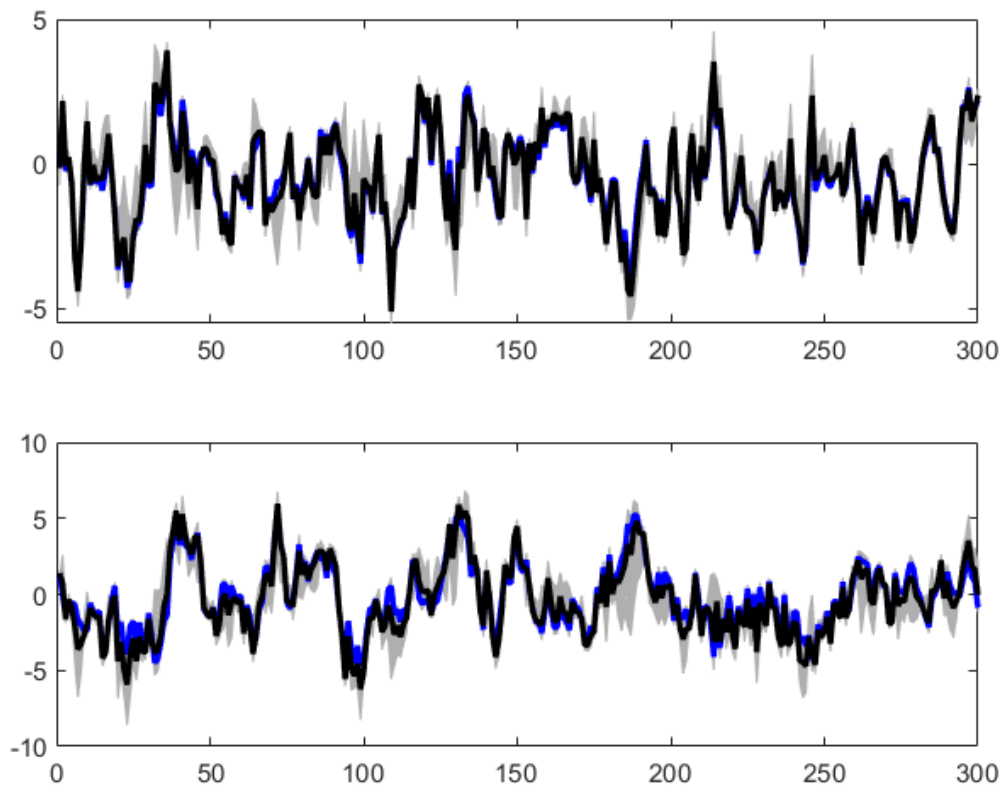
Example 32 (Structural factor model: Impulse responses) *As in example (31), we estimate a dynamic factor model on the artificial data* y *with two factors and one lag. We compute the dynamic responses induced by innovations to the first factor using two identification schemes: sign restrictions and IV or proxy identification. In the latter case we use the first of the true innovations* i(:,1) *as instrument; with sign restrictions we impose that the fourth (fifth) variable responds positively (negatively) on impact and on the following two periods. We plot the estimated responses of the variables [1 2 4 5 15 20] belonging to the data* y*, with the two identification schemes together with the true IRF. The commands are as follows:*

```
% 2 dynamic factors with one lag
nfac    = 2;
lags    = 1;
% IRF options
% IV identification
options.proxy    = i(lags+2:end,1);
% Sign identification
options.signs1 = 'y(4,1:3,1)<0';
options.signs2 = 'y(5,1:3,1)>0';
% estimation command
[BDFM] = bdfm_(y,lags,nfac,options);
% variables and shock of interest
index_var        = [1 2 4 5 15 20];
index_sho        = 1;
% some options:
% add the true IRF
options.add_irfs    = ir_true(index_var,:,index_sho);
% additional 90% HPD set
options.conf_sig_2 = 0.9;
% figure format
options.nplots = [2 3];
% name of the directory where the figure is saved
options.saveas_dir    = './factor_plt';
% name of the figure to save
options.saveas_strng  = 'sign';
% sign restricted IRF
irfs_to_plot        = BDFM.irsign_draws(index_var,:,index_sho,:);
plot_irfs_(irfs_to_plot,options)
% IV IRF
irfs_to_plot_iv     = BDFM.irproxy_draws(index_var,:,index_sho,:);
```

```
% name of the figure to save
options.saveas_strng  = 'iv';
plot_irfs_(irfs_to_plot_iv,options)
```

Figure 26: Structural dynamic factor model: sign restrictions identification of the innovation to the first factor.
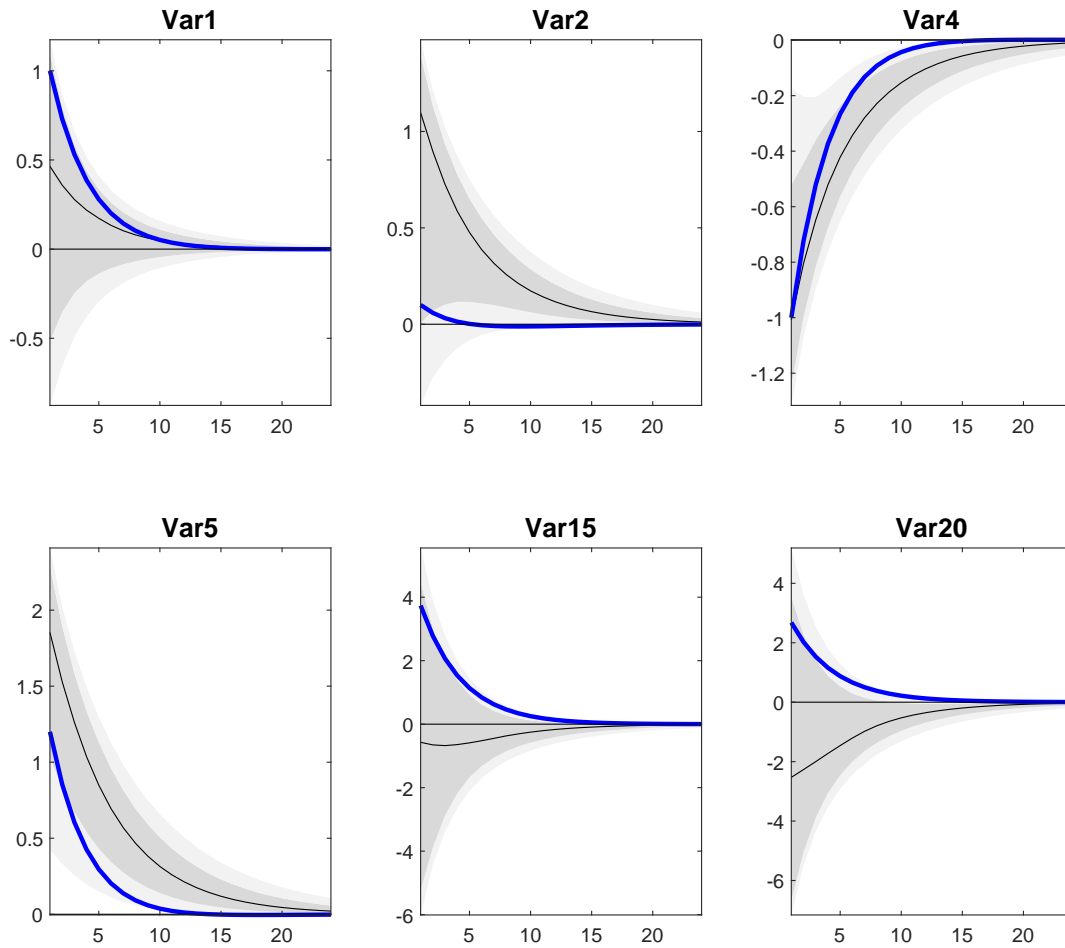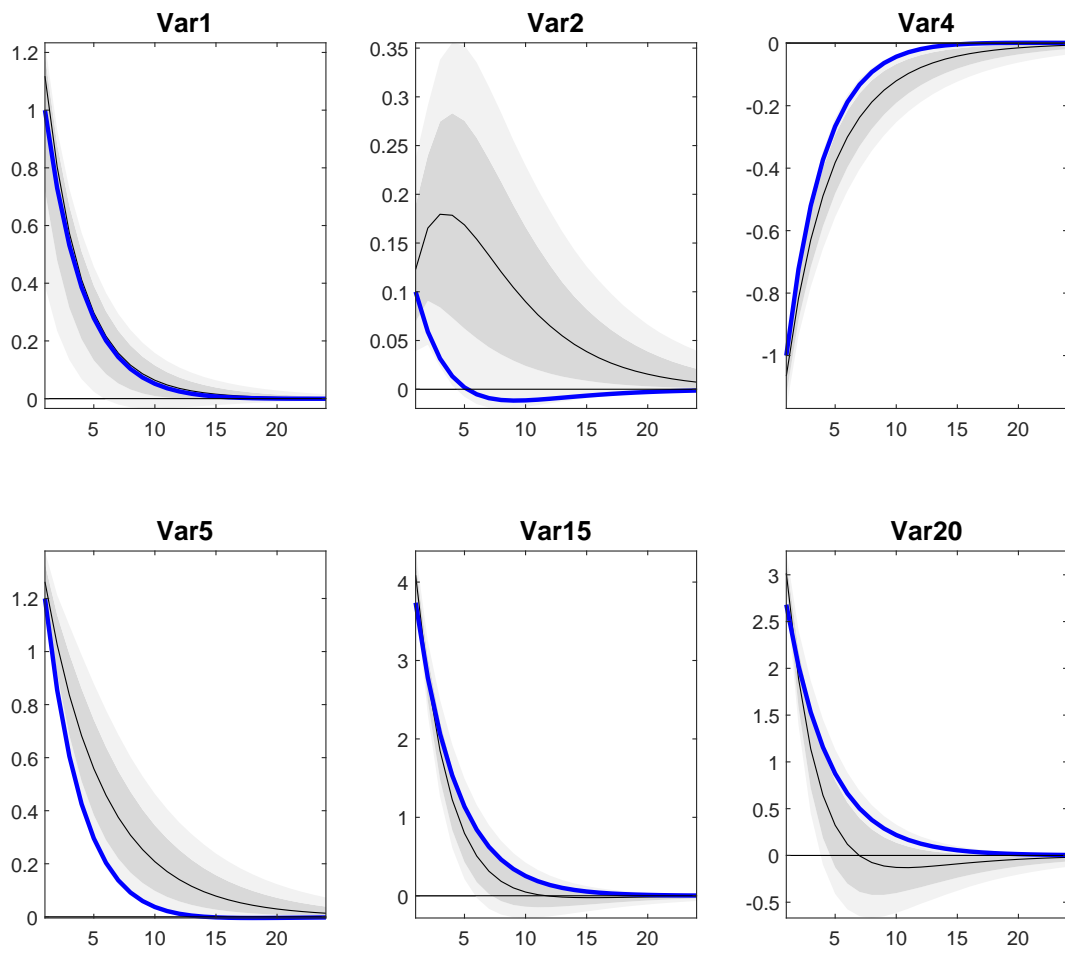
Figure 27: Structural dynamic factor model: proxy identification of innovation to the first factor.

5 Direct methods

Direct methods (DM) provide an alternative approach to study causal inference and to predict 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 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 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 at 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.

5.1 Notation and Mapping

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

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

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, $\alpha_{(h)}$, $\beta_{(h)}$, and $\Sigma_{(h)}$ are matrices of coefficients of appropriate dimensions, z_t represents a set of exogenous controls (and possibly, lags of the controls), and $\Gamma_{(h)}$ is an appropriate loading matrix.

For the purpose of this section, we assume that a $VAR(p)$ correctly represents the data generating process; thus, for now, $z_t = 0, \quad \forall t$. To ease the notation, we consider the VAR companion form representation pre-multiplied by the selection matrix G' which is defined in (3). Notice that $G'G = I_n$. The following set of equations summarize the mapping between

iterative and direct methods

$$y_t = \underset{(n\times np)}{G'} \mathbf{x}_t = G' \underset{(np\times 1)}{F_0} + G' \underset{(np\times np)}{F} \mathbf{x}_{t-1} + \Omega\nu_t$$

$$= \underset{(n\times 1)}{\alpha_{(0)}} + \underset{(n\times np)}{\beta_{(0)}} \mathbf{x}_{t-1} + \underset{(n\times 1)}{e_t^{(0)}}$$

$$y_{t+1} = G'\mathbf{x}_{t+1} = G'(I+F)F_0 + G'F^2\mathbf{x}_{t-1} + G'FG\Omega\nu_t + \Omega\nu_{t+1}$$

$$= \alpha_{(1)} + \beta_{(1)}\mathbf{x}_{t-1} + e_t^{(1)}$$

$$\vdots$$

$$y_{t+h} = G'\mathbf{x}_{t+h} = G' \sum_{j=0}^{h} F^j F_0 + G'F^{h+1}\mathbf{x}_{t-1} + G' \sum_{j=0}^{h} F^{h-j}G\Omega\nu_{t+j}$$

$$= \alpha_{(h)} + \beta_{(h)}\mathbf{x}_{t-1} + e_t^{(h)}$$

where we assume that

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

A few observations may highlight the features of the mapping. First, if DM includes $\mathbf{x}_{t-1}$ as regressors and $\Omega\nu_t \equiv u_t = e_t^{(0)}$, then $e_t^{(0)}$ is the one-step ahead innovation in y_t. Second, under proper identification restrictions, the impulse response function 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 data generating process is excluded from $\mathbf{x}_{t-1}$ and omitted and included variables are correlated. Thus, the use of control variables may help to proxy for omitted lagged variables. Finally, the mapping between the two approaches could be used to elicit priors for Bayesian versions of direct methods, see section 5.4 and appendix A.7 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 technology 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 of the form:

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

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

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

where $\bar{e}_t^{(h)} = F^h\Omega\begin{bmatrix}u_t^m \\ \nu_{-1,t}\end{bmatrix} + \sum_{j=1}^{h} F^{h-j}G\Omega\nu_{t+j}$, and $\rho_{(h)} = F^h\ \Omega\ [\rho\ \ 0]'$.

The impulse responses of interest are contained in the vector $\rho_{(j)}\ j = 0, ..., h$.

5.2   Local Projections

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

$$
\frac{dy_{t+h}}{d\nu_t} = \left(\widehat{\beta}_{(h-1)} \pm \tau_a SE(\beta_{(h-1)})\right)G\ \Omega \quad h > 0 \tag{7}
$$

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:

$$
\frac{\partial y_{t+h}}{\partial m_t} = \widehat{\gamma}_{(h)} \pm \tau_a SE(\gamma_{(h)}) \tag{8}
$$

The baseline estimation function for local projections is

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

The first input, `y`, is the data where we assume that there are no missing values. The second input , `lags`, is the number of lags, which is an integer greater than zero. Note that `lags` 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 `directmethods` 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 for the shock of interest. Notice that the first dimension of the proxy variables must coincide with the first dimension of the endogenous variables, `y`.

- `options.robust_se_` a scalar indicating the type of standard errors to be computed; when it is set to 0, no adjustments are made; when it is set to 1 standard errors are HAC adjusted as in 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 this option is used, 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 computed with a recursive identification scheme using equation (7). 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 computed with a recursive identification scheme using LP and equation (8). The dimensions are the same as the ones discussed in the previous bullet point.

5.3 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,

<div align="center">

`DM.forecasts`

</div>

is a (n × hor × 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.

5.4 Bayesian Direct Methods

Bayesian inference is possible with direct methods. In this case, one can 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 distributed, 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.12 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 for $\beta_{(h)}$ 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.7. To set the hyperparameters, one can use a pre-sample of data or data from different country/sectors. The prior variance of the autoregressive coefficient is controlled by a 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 for the covariance of the residuals.

- `options.priors.Sigma.df` is a scalar defining the prior degrees of freedom.

- `options.priors.tau` is a $(\texttt{hor} \times \texttt{1})$ vector controlling shrinkage at various horizons; the default value is one. The larger this value is, the stronger will be the shrinkage around the prior mean.

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 $(\texttt{n} \times \texttt{hor} \times \texttt{n} \times \texttt{K})$ matrix containing the Bayesian local projections impulse responses obtained with recursive identification. The first dimension corresponds to the variables, the second to the horizon, 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 ($\texttt{n} \times \texttt{hor} \times \texttt{1} \times \texttt{K}$) matrix containing the Bayesian local projection impulse responses with IV identification. The first dimension corresponds to the variables, the second to the horizon, 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 ($\texttt{hor} \times \texttt{n} \times \texttt{K}$) 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 uncertainty.

- `dm.bforecasts.with_shocks` is a ($\texttt{hor} \times \texttt{n} \times \texttt{K}$) 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 account for 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:

```
% triggers maximization of the overall prior shrinkage
options.priors.max_tau = 1;
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. The options are:

   - `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.

We use the same monthly data employed section 3.4. Recall that the `DataGK.mat` contains the log of industrial production index (`logip`), the log of CPI index (`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 33 (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 projection impulse response function with default settings. First, we prepare the data. By default, the horizon of the responses is 24. We modify it to have the same horizon length as in GK and then run the model. Figure 28 reports the responses. Relative to the Cholesky VAR responses (presented in figure 1), bands are wider for industrial production and for the excess bond premium. However, the dynamic profile of the median is comparable.*

```
load DataGK          % load the data
y = [logip logcpi gs1 ebp]; % combine the data in a Tx4 matrix
% Estimate LP IRF of 4 year horizon assuming 12 lags
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 in the dataset
% 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];
lpirf2plot = dm1.ir_lp(indx_var,:,indx_sho,:);
% Customize the IRFs 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 BVAR 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 the responses
plot_irfs_(lpirf2plot,options)
```

Figure 28: Local projection responses to a monetary policy shock identified with recursive restrictions. Dark gray bands report 90% confidence intervals. Blue lines report the 90% confidence set for VAR-based responses.
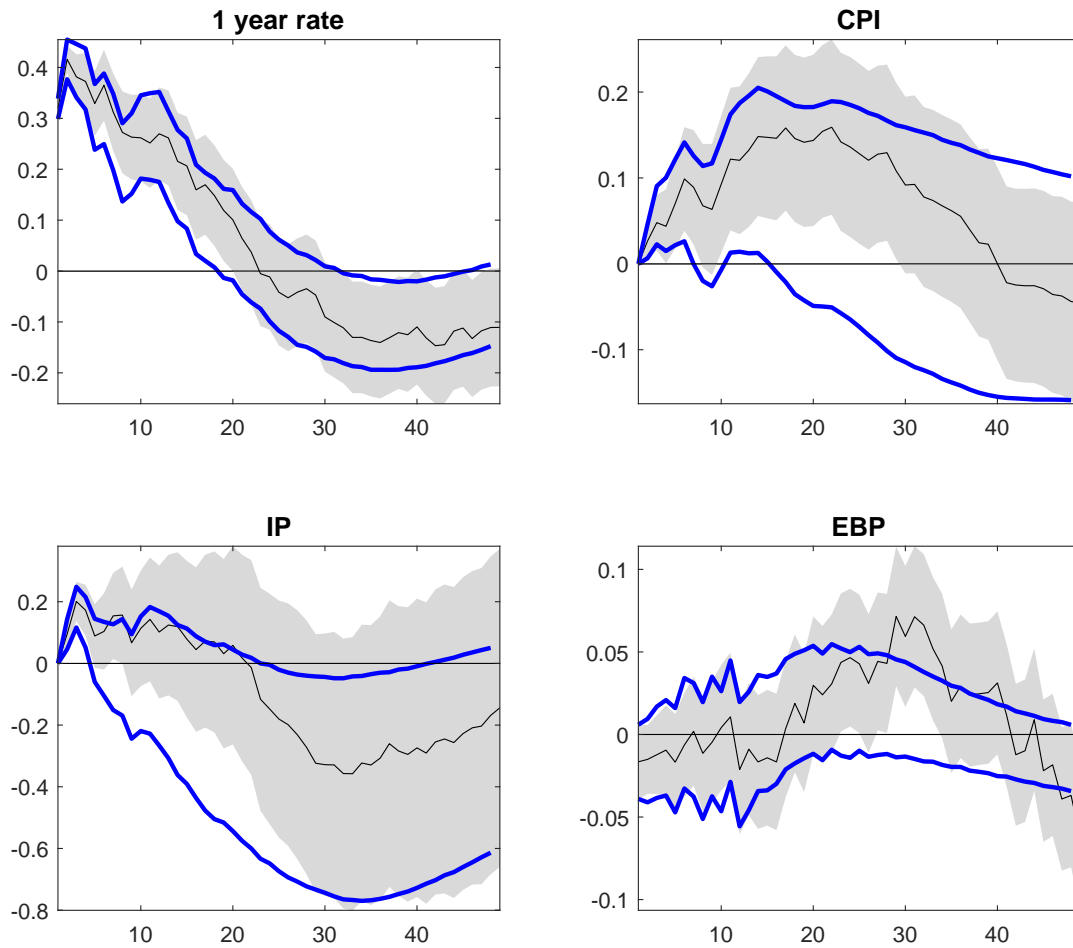
Example 34 (LP: MP shock with IV identification) *Consider the same setting of example 11. We now identify a monetary policy shock with external instruments and trace the dynamic transmission of the shock using LP. We consider the same set of endogenous variables as in the VAR setting, i.e.* $\mathbf{x}_{t-1}$ *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*). The left side of figure 29 reports the responses. Relative to the IV VAR responses (reported 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.*

```
%    load the instruments from GK dataset
[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(y)- length(numi)+1:end) = numi(:,4);
% Estimate LP IRF of 4 year horizon assuming 12 lags
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   = {  '1 year rate' 'CPI' 'IP' 'EBP'};
% increase the title size
options0.fontsize =18;
% normalize shock to be 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)
```

Example 35 (LP: MP shock with IV identification with one endogenous variable at time) *We consider the same setting of example 34. 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 side of figure 29 plots the responses (the commands used to produce the plot are reported below).*

112

In this case not only the uncertainty is significant but also the transmission dynamics are quite different. Thus, unless one has the information set correctly specified, the responses obtained are biased, because the instrument ends up being correlated with the error of the projection equation.

```matlab
% 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 29: Local projection responses to a monetary policy shock identified with IV restrictions. Dark gray bands report 90% confidence intervals. Left panels: specification of example 34. Right panels: specification of example 35.
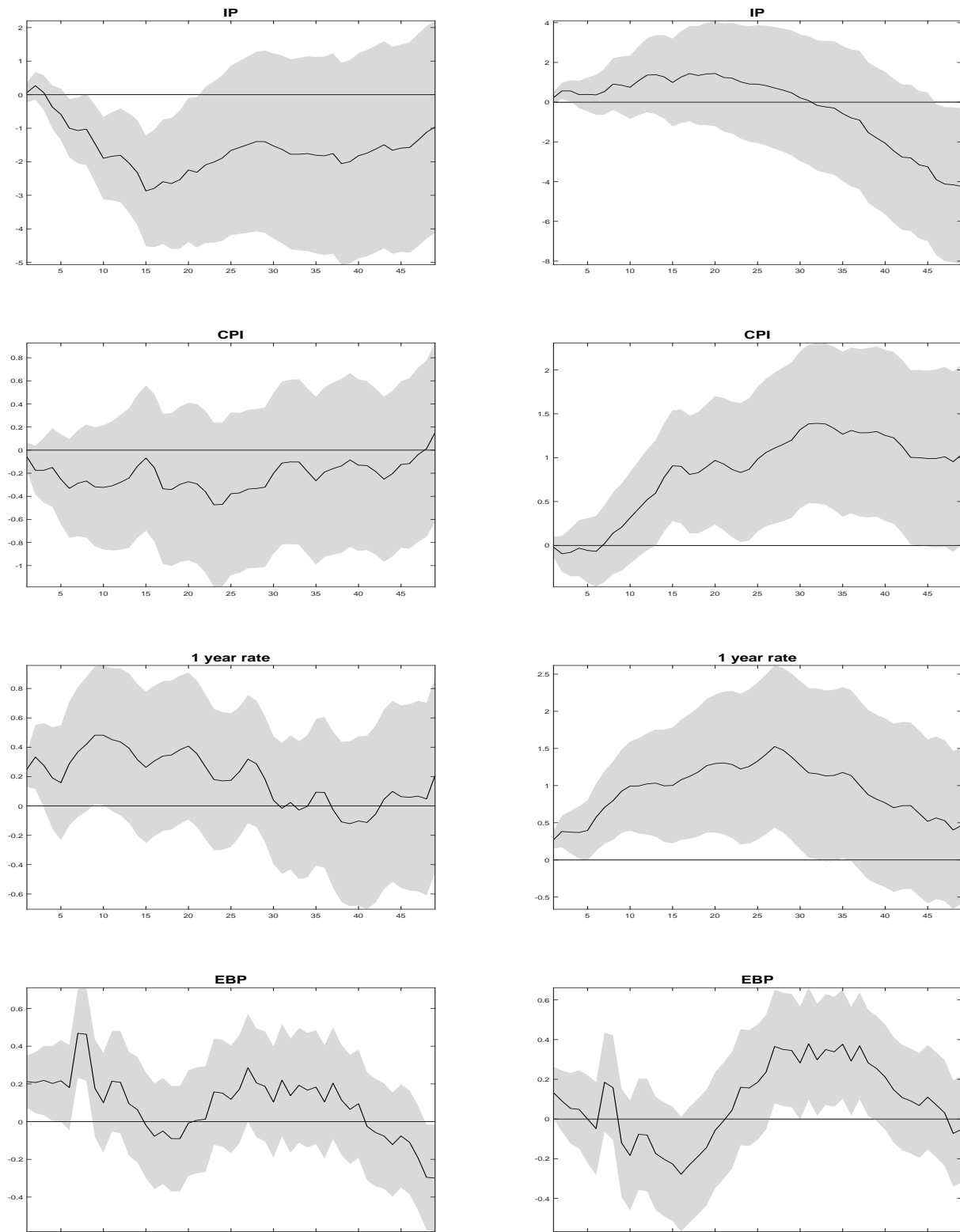
It may be interesting to see whether Bayesian methods allow a more precise characterization when IV are used. The next example considers this setup.

Example 36 (Bayesian LP with VAR priors) *We consider the same setting in 34. We use the first eight years of data to calculate the posterior distribution of the VAR reduced form parameters, use the posterior means to center the priors for the LP parameters, and to compute the response functions. The commands are reported below. Figure 30 reports the responses. Relative to the left panels of figure 29, bands are narrower and inference is, indeed, more precise.*

```
%   run a VAR on a presample of data
presample = 96; % 8 years of presample
lags    = 12;
bvar1   = bvar_(y(1:presample,:),lags);
% use 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(bvar1.Phi_draws,3);
% use the posterior variance of the VAR coefficients
options.priors.Phi.cov   = diag(mean(var(bvar1.Phi_draws,0,3),2));
% use the posterior mean of the covariance of the VAR residuals
options.priors.Sigma.scale = mean(bvar1.Sigma_draws,3);
options.priors.Sigma.df  = size(bvar1.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 the shock to 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 30: Bayesian local projection responses to a monetary policy shock identified with IV restrictions. Dark (light) gray bands 90% credible sets.
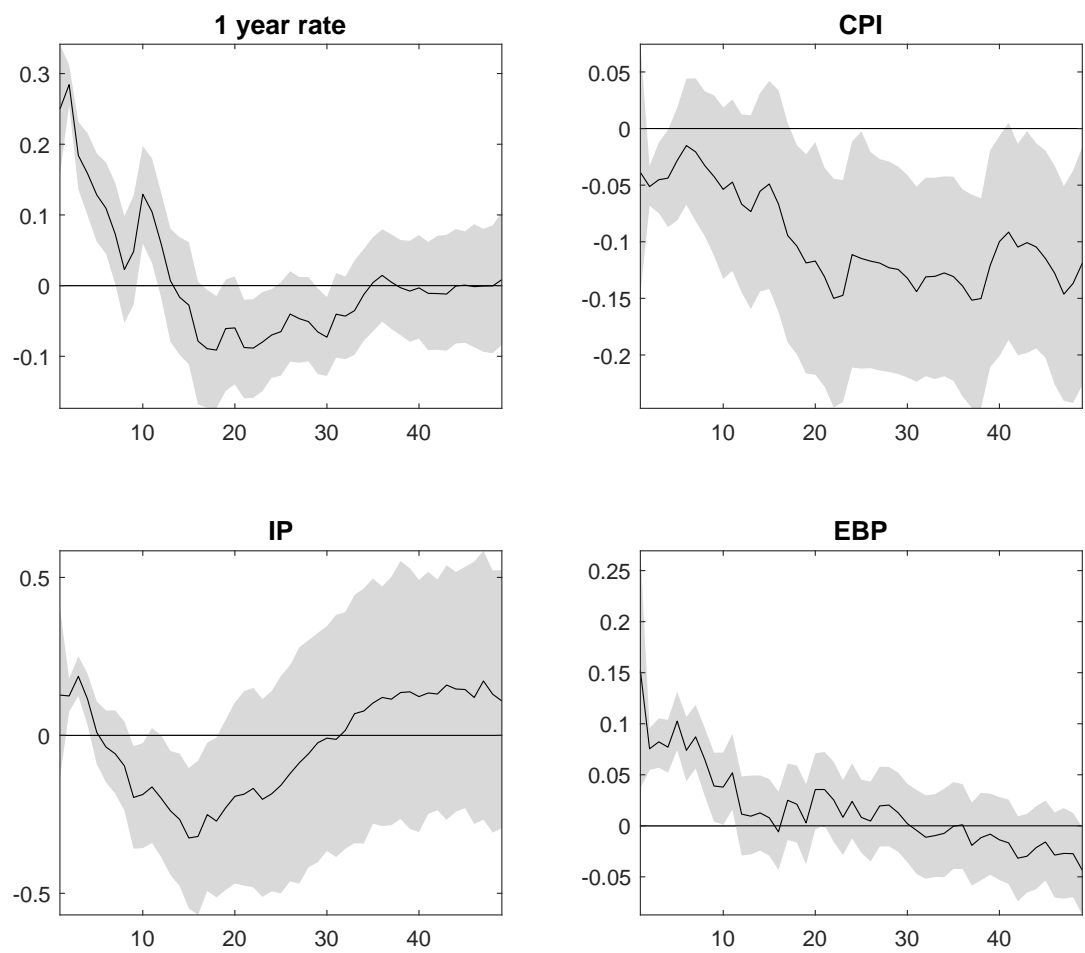
6 FILTERING AND MEASURING CYCLICAL INFORMATION

In this section we show how to use the toolkit to extract latent cyclical information from an observable time series and to date business cycles.

6.1 TREND AND CYCLE/PERMANENT AND TRANSITORY DECOMPOSITIONS

Trend and cycle (permanent and transitory) decompositions are very popular in academic and policy making as way to get estimates of latent variables. Canova (2020) has argued that decompositions of this sort may fail to recover objects that economists think important (output gap, cyclical fluctuations, relative importance of permanent and transitory shocks) whenever the data has been generated by the models that nowadays are used to explain the data. Thus, great care needs to be exercised in taking the output of the procedures we describe for inference.

Traditionally, time series are decomposed in trend, cycle, seasonal and irregulars. Since seasonals are not of general interest, we assume that data is seasonally adjusted. Furthermore, for most cases, we take the irregulars (high frequency variations) to be part of the cycle. Thus, one needs only to split a time series into two latent components which, for short, we call the trend and the cycle, even if the trend may have business cycle variations and may not be permanent and the cycle may feature low frequency variations and be highly persistent.

To split an observable (or a vector of) time series into two latent components one needs to make assumptions. The typical ones used concern the properties of the trend and the correlation between the two latent components. Thus, for each procedure, we spell out the assumptions used to perform the decomposition.

Historical decomposition

• Polynomial trend

This approach is the oldest and maintains that the trend is a deterministic process and it is uncorrelated with the cycle. Thus, the latter can obtained as the residual of a regression of the observables on a polynomial trend:

$$y_t = a + bt + ct^2 + \ldots + e_t$$

By construction, the cycle thus accounts for all the fluctuations in the observables. Thus, it will display business cycle as well as high and low frequency variations. The command to run the decomposition is:

$$[\texttt{dc,dt}] = \texttt{polydet(y,ord,gg)};$$

where, `y` is the data, `ord` is the order of the polynomial (the upper limit is 4) and `gg` an indicator; if it is equal to 1 it plots the data, the trend, and the cycle. The output of the

function are the estimated cycle `dc` and the estimated trend `dt`.

- Polynomial break trend

A deterministic trend has two unpleasant features: it can be perfectly predicted far out in the future; and no acceleration/deceleration of its growth rate is possible. In addition, the cycles which are extracted are highly persistent and have period of fluctuations which is larger than those typically relevant for business cycle analysis. Using a break in the trend, allows changes in its growth rate but leaves the other two problems untouched. The command to run the break trend decomposition is:

$$[\texttt{dc,dt}] = \texttt{polydet\_break(y,ord,tt,gg)};$$

where `y` is the data, `ord` is the order of the polynomial (upper limit 4), `tt` is the break date and `gg` an indicator; if it is equal to 1 it plots the data, the deterministic trend, and the cycle. The output of the function are the estimated cycle `dc` and the estimated trend `dt`. The break date must be selected in advance. Thus, introspection or preliminary statistical tests (e.g a Chow test on an interval of dates) needs to be used, prior to running the decomposition.

- Differencing

The third approach assumes that the trend is stochastic, displays at least one unit root, and it is uncorrelated with the cycle. Thus, trend estimates are equal to y_{t-k} and cycle estimates are equal to $y_t - y_{t-k}$. There is no specific function in the toolkit to perform this decomposition. Users may employ simple Matlab commands to do it. For example, to compute the decomposition for $k = 1, 4, 20$ the following can be used

```
% one quarter difference (quarterly data)
dy1(2:end,:) = y(2:end,:)-y(1:end-1,:);
dt1(2:end,:) = y(1:end-1,:);
% one year difference (quarterly data)
dy4(5:end,:) = y(5:end,:)-y(1:end-4,:);
dt4(5:end,:) = y(1:end-4,:);
% 5 years difference (quarterly data)
dy20(21:end,:) = y(21:end,:)-y(1:end-20,:);
dt20(21:end,:) = y(1:end-20,:);
```

Note that in the above `y` could be a $T \times n$ matrix, where T denotes the length of the time series and n the number of variables. Clearly, the longer is the differencing operator, the larger is the number of data points lost. The cycle that short differencing produces is typically very volatile and displays powerful high frequency variability. The cycle obtained with $k > 1$ is typically smoother, but displays potentially artificial serial correlation (moving average components of order $k - 1$).

118

Because the cycle computed with the above commands does not necessarily have zero mean, it is a good idea to demean y prior to computing the required difference.

• Hamilton (local projection) trend

The approach assumes that the trend is stochastic and uncorrelated with the cycle. The trend here is defined to be the medium term forecastable component of the series and may or may not display a unit root. To obtain the decomposition one runs the local projection:

$$y_{t+h} = b_0 y_t + b_1 y_{t-1} + ... + b_p y_{t-p} + e_{t+h} \tag{9}$$

and an estimate of the cycle is $y_{t+h} - \hat{b}_0 y_t - \hat{b}_1 y_{t-1} - ... - \hat{b}_p y_{t-p}$. The command to run the decomposition is:

```
[dc,dt] = hamfilter(y,h,lags,ff,gg);
```

where y is the data, h is the horizon of the projection (upper limit $h = T - p - 1$); lags is the number of lags included in the projection; ff an indicator; if it is equal to 1 a constant is included in the projection; gg an indicator; if it is equal to 1 it plots the data, the trend, and the cycle. The output of the function are the estimated cycle dc and the estimated trend dt. The typical value for h is 2 for yearly data, 8 for quarterly data, and 24 for monthly data. For lags one would like to use at least one year of data; thus lags = 1 for annual data, lags=4 for quarterly data; lags=12 for monthly data.

The function allows y to be a $T \times n$ matrix, where T is the sample size and n the number of series. Conditioning variables, other than the lags of the endogenous variable y are not allowed in the specification. Contrary to previous approaches, cycle estimates obtained with this approach are model free and thus are robust to misspecification of the time series properties of the trend process. Furthermore, they will be stationary if y_t has up to d unit roots and properties similar to h-differencing of the data (see Hamilton (2018)).

• Hodrick and Prescott trend.

Here the trend is assumed to be stochastic but smooth (its acceleration should be small) and uncorrelated with the cycle. The Hodrick and Prescott (HP) trend is obtained via the ridge estimator:

$$\tilde{y} = (H'H + \lambda Q'Q)^{-1} H'y$$

where λ is a smoothing parameter, $y = (y_1, \ldots y_t)$ the observable series, $\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_t, \tilde{y}_{t+1}, \tilde{y}_{t+2})$, the trend, $H = (I_{t \times t}, 0_{t \times 2})$,

$$Q_{t \times (t+2)} = \begin{bmatrix} 1 & -2 & 1 & 0 & ... & ... & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & ... & ... & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & ... & ... & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & ... & ... & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & ... & ... & 1 & -2 & 1 \end{bmatrix}$$

and the cycle is the difference between the level and the HP trend. The commands to run the decomposition are:

```
dt = Hpfilter(y,lam);        dc = y-dt;
```

where `y` is the data, `lam` is the smoothing parameter. The output of the `hpfilter` function is the estimated trend `dt`. The typical values of λ are 1,600 for quarterly data, 6.25 for annual data and 129,000 for monthly data. Ravn and Uhlig (2002) showed that with these values the estimated cycles in monthly, quarterly and annual data displays the same ups and downs and have a periodicity of approximately 4-8 years.

Because the gain function of the standard HP cyclical filter knocks out almost all power at low frequencies (8-16 years), and partially eliminate power at business cycle frequencies (2-8 years), and because these frequencies may be important for understanding economic phenomena, one may want to consider alternatives which allow in the cycles also contain relevant medium term cyclical variations. For example, for quarterly data $\lambda = 51,200$, could be used, a value close to the BIS recommendations, see Borio (2012).

• One-sided HP trend

The HP-filter is two-sided and thus not very useful for real time analysis and forecasting. In addition, by construction, the cycle at t artificially predicts y_{t+2}. There is a one-sided version of the HP filter which does not feature future predictability. The specification is:

$$y_t = y_t^x + y_t^c$$
$$y_t^x = 2y_{t-1}^x - y_{t-2}^x + \epsilon_t$$

where y_t^x, y_t^c are the trend and the cycle and ϵ_t, y_t^c are uncorrelated white noise sequences. Stock and Watson (1999) show that such a model has the following state space representation:

1. State Equation
$$\begin{bmatrix} y_{t|t}^x \\ y_{t-1|t}^x \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1|t-1}^x \\ y_{t-2|t-1}^x \end{bmatrix} + \begin{bmatrix} \epsilon_t \\ 0 \end{bmatrix}$$

2. Observation Equation

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t|t}^x \\ y_{t-1|t}^x \end{bmatrix} + \begin{bmatrix} y_t^c \\ 0 \end{bmatrix}$$

The trend and the cycle can be estimated with standard Kalman filter/EM algorithm iterations. One can restrict $\lambda = \frac{\sigma_c^2}{\sigma_\epsilon^2}$ with a prior, e.g. $\lambda \sim N(1600, 10)$. In this case, MCMC methods need to be used.

The toolkit uses the serial implementation of Meyer-Gohde (2010). It is much faster than other procedures and gives almost identical results. In practice, the one-sided HP trend is obtained calculating for each $t-2$ the standard HP filtered trend with data up to $t-2$ and equating the one-sided HP trend at $t-2$ with the standard HP trend value for period $t-2$

120

(i.e., we compute T two-sided HP filters trends, one for each $2 < t < T - 2$). The command to run this decomposition is:

```
[dt,dc] = one_sided_hpfilter_serial(y,lam,disc);
```

where `y` is the data, `lam` is the smoothing parameter, `disc` the number of unused observations (typically `disc = 0`). The output of the function is the estimated trend `dt` and the estimated cycle `dc`.

Example 37 (Two- and one-sided HP filter) *We consider quarterly Euro Area log GDP from 1999Q1 until 2017Q4 available in the AWM DATABASE. We apply the one- and two-sided HP filter to the (log of) raw data. Figure 31 reports the log level data, the two trend and cycle decompositions. Note that the one-sided HP gives a much more optimistic view of the cycle after 2013.*

Figure 31: One- and two-sided HP filter of the EA log of real GDP.
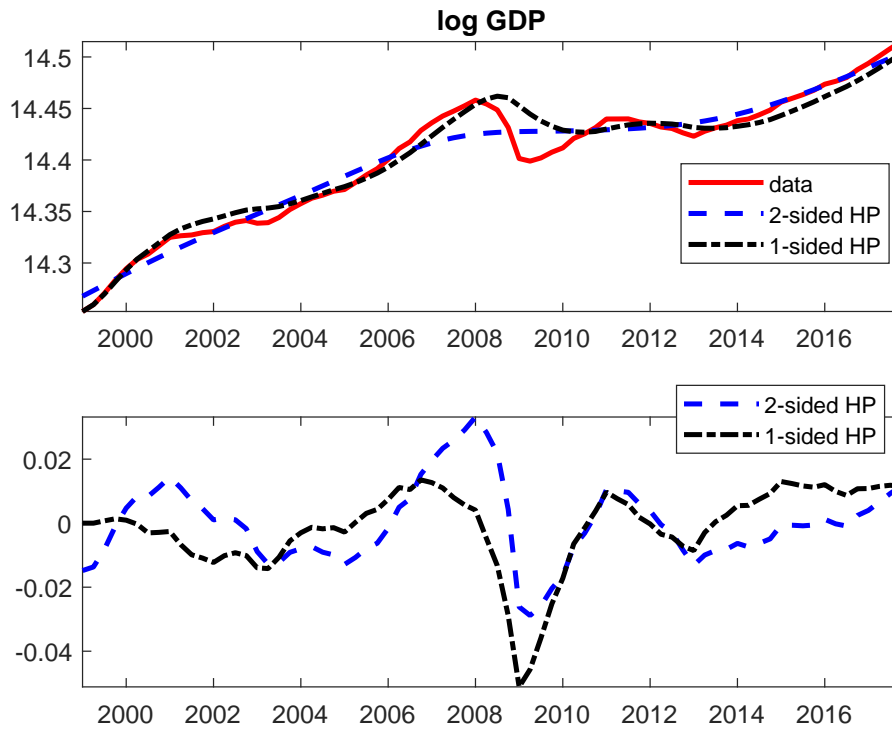
• Band pass (BP) filter

BP filters are a combination of high pass and low pass filters. A band pass filter has a squared gain function equal to 1 for all frequencies in some interval (ω_1, ω_2) and 0 otherwise. Thus, it passes unchanged all the variability of the input between (ω_1, ω_2) and wipes out the variability outside of it. The time series versions of these filters are:

Low pass: $\mathcal{B}_0^{lp} = \frac{\omega_1}{\pi}$; $\mathcal{B}_j^{lp} = \frac{sin(j\omega_1)}{j\pi}$; $\quad 0 < j < \infty$, some ω_1.

High pass: $\mathcal{B}_0^{hp} = 1 - \mathcal{B}_0^{lp}$; $\mathcal{B}_j^{hp} = -\mathcal{B}_j^{lp}$; $\quad 0 < j < \infty$.

121

Band pass: $\mathcal{B}_0^{bp} = \mathcal{B}_j^{lp}(\omega_2) - \mathcal{B}_j^{lp}(\omega_1); \quad 0 < j < \infty, \ \omega_2 > \omega_1$.

Note that j must go to infinity to have the required gain function. Hence, for finite T, only approximations to these filters can be computed. The two most common ones have been suggested by Baxter and King (1999) (BK), who cut j at some $\bar{J} < \infty$ (usually 3 years of data) and Christiano and Fitzgerald (2003) (CF) who use a non-stationary, asymmetric approximation which is optimal in the sense of making the approximation error as small as possible. In this approximation the coefficients of the filter change magnitude and even sign at each t.

The CF version has better spectral properties (by construction the approximation error is smaller) but one needs to know if the input series is $I(1)$ or $I(0)$ before applying the filter. Furthermore, phase shifts may occur (the timing of ups and downs of the output do not necessarily coincide with the ups and downs of the input), since the filter is asymmetric. The two approximations produce similar timing of recessions and expansions when applied to US macroeconomic time series, except at the beginning and at the end of the sample.

The toolkit features both the BK and the CF approximations. The BK decomposition can be called using:

$$\texttt{dc = bkfilter(y,bp1,bp2);} \qquad \texttt{dt = y-dc;}$$

where the first command calculates the cycle and the second the trend. `bp1` is the upper limit of the frequency band and `bp2` is the lower limit of the frequency band over which the squared gain function is 1. Typical choices for quarterly data are `bp1 = 8` and `bp2 = 32`. For monthly (annual) data the typical choices are `bp1 = 24` (2) and `bp2 = 96` (8).

The command for the CF decomposition is instead:

$$\texttt{dc = cffilter(y,bp1,bp2,cf1,cf2,cf3);} \qquad \texttt{dt = y-dc;}$$

where `cf1` is an indicator function describing the time series properties of the input (`cf1 = 0` no unit root); `cf2` is an indicator function for whether there is a drift in the unit root or a deterministic trend (`cf2 = 0` no drift); `cf3` chooses the format of the filter: `cf3 = 0` uses the basic asymmetric filter; `cf3 = 1` uses the symmetric filter; `cf3 = 2` uses a fixed length symmetric filter; `cf3 = 3` uses a truncated, fixed length symmetric filter (the BK choice); and `cf3 = 4` uses a trigonometric regression filter. If `cf1`, `cf2` and `cf3` are omitted, the default values are `cf1 = 1`, `cf2 = 1` and `cf3 = 0`.

• Wavelet filters

A Wavelet filter is build using the same principles of a BP filter but the implementation is directly in time domain. Furthermore, at each t the filter uses only past information to separate the two components. Thus, it is a one-sided as opposed to a two-sided filter, such as the BP or HP. How far one looks back in the past to construct the filtered series depends on the cycles one wants to extract.

Wavelet filters can be used on stationary and non-stationary inputs. The filter assumes the following representation for the observable time series:

$$y_t = \sum_{j=1}^{J} D_{jt} + S_{J,t}$$

$$D_{jt} = 1/(2^j)\left(\sum_{i=0}^{2^{j-1}-1} y_{t-i} - \sum_{i=2^{j-1}}^{2^j-1} y_{t-i}\right)$$

$$S_{J,t} = 1/(2^J)\left(\sum_{i=0}^{2^J-1} y_{t-i}\right)$$

where typically $J = 5$. $j = 1, 2$ capture high frequency; $j = 3, 4$ business cycles and $j = 5$ low frequencies. S_{Jt} captures the long run component. In the toolkit, D_{3t}, D_{4t}, D_{5t} are computed using the Haar wavelet representation described by Lubik, Matthes and Verona (2019). The representation for the relevant cycles are:

8-16 quarters cycles:

$$D_{3t} = 1/8 \left(y_t + y_{t-1} + y_{t-2} + y_{t-3} - y_{t-4} - y_{t-5} - y_{t-6} - y_{t-7}\right)$$

16-32 quarters cycles:

$$D_{4t} = 1/16 \left(y_t + y_{t-1} + y_{t-2} + y_{t-3} + y_{t-4} + y_{t-5} + y_{t-6} + y_{t-7}\right.$$
$$\left. - y_{t-8} - y_{t-9} - y_{t-10} - y_{t-11} - y_{t-12} - y_{t-13} - y_{t-14} - y_{t-15}\right)$$

32-64 quarters cycles:

$$D_{5t} = 1/32 \left(y_t + y_{t-1} + y_{t-2} + y_{t-3} + y_{t-4} + y_{t-5} + y_{t-6} + y_{t-7}\right.$$
$$+ y_{t-8} + y_{t-9} + y_{t-10} + y_{t-11} + y_{t-12} + y_{t-13} + y_{t-14} + y_{t-15}$$
$$- y_{t-16} - y_{t-17} - y_{t-18} - y_{t-19} + y_{t-20} - y_{t-21} - y_{t-22} - y_{t-23}$$
$$\left. - y_{t-24} - y_{t-25} - y_{t-26} - y_{t-27} - y_{t-28} - y_{t-29} - y_{t-30} - y_{t-31}\right)$$

The wavelet filter decomposition can be obtained using the command:

```
[dc1,dt1,dc2,dt2] = wavefilter(y,gg);
```

where `dc1` and `dc2` are the cyclical components computed focusing attention on 8-32 ($D_{3t} + D_{4t}$) or 8-64 ($D_{3t} + D_{4t} + D_{5t}$) quarters cycles and `dt1 = y-dc1`, `dt2 = y-dc2` are the corresponding trend estimates; `y` is the input series, and `gg` is an indicator function for whether the cyclical components are plotted (for `gg = 1` plots are shown).

Example 38 (BP and Wavelet filters) *Consider the same data of exercise 37. We apply the BP and Wavelet filter to the logarithm of the raw data. Figure 32 reports the trend and cycle decompositions obtained with BK, CF, and the two wavelet decompositions.*

Figure 32: BP and Wavelet filters of the EA log of real GDP.
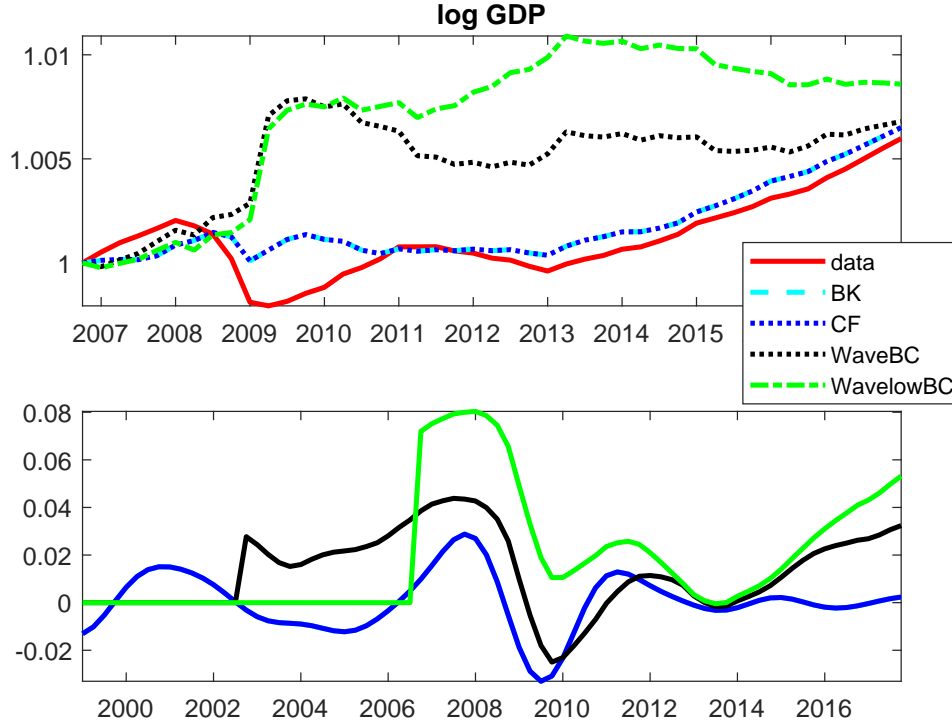
• Butterworth filters

Butterworth are a class of filters which can be designed by engineers to have low pass, high pass, band pass and even stop pass characteristics. They are very flexible and can be specified so that the cycle captures medium and low frequency variations; or to eliminate unit roots without affecting medium frequencies.

The squared gain function of the filter is: $G(\omega) = \frac{G_0}{1+(\frac{\omega}{\omega_c})^{2n}}$, where G_0 is the gain at the zero frequency, n is the (polynomial) order of the filter and ω is a selected frequency and ω_c a reference frequency (typically $\omega_c = 1$). By playing around with these three parameters, one can give to the squared gain a variety of shapes.

Canova (2020) finds the specification useful to cut the spectrum of the observed series horizontally, so that the estimated latent components have power at all frequencies, rather than cutting it by frequencies as, for example, the BP filter does. Butterworth filters are particularly useful when not all low frequency variations should be attributed to the trend.

There is no specific command in the toolkit to implement Butterworth filters, but simple Matlab functions can be used if the signal processing toolbox is installed.

Here is a simple set of commands which allow to build a Butterworth filter:

1. Creating the ARMA weights

$$[a,b] = \text{butter}(n,\text{cutoff},\text{type})$$

where `n` is the degree of the polynomial, `cutoff` is the point in radians where the squared gain falls and `type` is an indicator for the type of filter you want (could be

low, high, stop-pass). If `cutoff` is a vector with two values, `butter` computes band pass weights; `a` are the AR weights and `b` are the MA weights of the filter.

2. Filtering the input with an ARMA(a,b)

$$yc = \text{filtfilt(a,b,y)}$$

This command creates a filtered (output) series `yc` using an $ARMA(a, b)$ on y as input. With these commands $G_0 = 1$ is automatically set. For a different G_0, one needs to rescale the `b` coefficients (if they are scaled up, the squared gain at ω_0 is lower).

• Unobservable component (UC) decompositions

UC methods are based on a state space formulation of the latent variable problem. Thus, there is a measurement equation linking the observable to the latent components and one or two state equations describing the time evolution of the latent variable(s). Typically, the trend is assumed to be random walk, with or without drift depending on whether the input data is raw or demeaned, while the cycle is assumed to be an AR process. In some specifications, the cycle is assumed to be a 2×1 vector of trigonometric functions which load at a particular frequency ω (see e.g. Runstler and Vlekke (2018)). The measurement equation may feature a measurement error. Typically, the innovations in the two latent variables are assumed to be uncorrelated and the parameter of the model are estimated by a Kalman filter, EM routine. Recently Grant and Chan (2017) described an approach which allows them to be correlated. Estimation in this case is performed with MCMC methods.

The standard time series specification can be boosted up with regressors, affecting the law of motion of the latent variables, and complicated structure can be imposed on the innovations to the latent variables, see e.g. Stock and Watson (2016). The procedure is typically run on a single time series, but the setup can be easily made multivariate, see e.g. Gonzalez-Astudillo and Roberts (2016), or Grant and Chan (2017). In this case, one can assume that there are common trends, or common cycles, or both.

One UC setup the toolkit accepts is the following:

$$y_t = \tau_t + c_t$$
$$\tau_t = \tau_{t-1} + \mu + \eta_t$$
$$c_t = \varphi_1 c_{t-1} + \varphi_2 c_{t-2} + \epsilon_t$$

Estimates of $(\theta_1, \theta_2, \mu, \sigma_u^2, \sigma_\eta^2, \sigma_\epsilon^2)$, and $\rho = corr(\eta_t, \epsilon_t)$ can be obtained KF-ML approach or by MCMC with flat prior. The implementation employed uses the codes of Grant and Chan (2017) and can be activated with the following command

$$[\text{yt,yc}] = \text{uc1\_(y,opts)};$$

where `y` is the data and `options` is a field structure that allows to customize various the settings which are listed below:

- `opts.figg`: when equals 1 it plots the series, the trend, and the cycle.

- `opts.noprint`: when equals 1 it does not print estimates.

- `opts.nsims`: sets the number of MCMC simulations (default 5000).

- `opts.burnin`: sets the number of burn-in draws (default, 30000).

- `opts.rhoyes` when equals 1, it allows correlation between the trend and the cycle (default, 0)

- `opts.mu0`: prior mean trend drift (default, 0.1)

- `opts.Vmu`: prior variance trend drift (default, 1)

- `opts.phi0`: prior mean cycle AR1 and AR2 coefficients (default, `[0.2 0.2]`)

- `opts.Vphi`: prior variance AR coefficients (default, diagonal matrix with 1)

- `opts.sigc2`: prior mean cycle variance (default, 0.5)

- `opts.sigtau2`: prior mean trend variance (default, 1.5)

- `opts.time`: `T` $\times$ 1 array containing the time span. This option can be used when `opts.figg = 1` so that the x-axis display the desired time units.

The law of motions of the latent variables can be made more complicated, for example one can assume a break in the random walk drift or assume a local linear trend specification, see e.g. Harvey (1990). The toolkit also allows to estimate a local linear trend model:

$$y_t = \tau_t + c_t$$
$$\tau_t = \tau_{t-1} + \mu_t + \eta_{2,t}$$
$$\mu_t = \mu_{t-1} + \eta_{1,t}$$
$$c_t = \varphi_1 c_{t-1} + ... + \varphi_p c_{t-p} + \epsilon_t$$

The implementation employed uses a maximum likelihood approach and can be activated with the following command:

$$[\texttt{yt,yc,out}] = \texttt{uc2\_(y,lags,opts)};$$

where `y` is the data, `lags` is the number of lags in the cycle specification. The parameters of the local linear trend model are ordered as follows $[\varphi_1, ..., \varphi_p, \sigma_\epsilon, \sigma_{\eta,1}, \sigma_{\eta,2}]$; they can be estimated or calibrated by setting the appropriate options (see below) and are reported as additional output as subfields in `out`. `opts` is a field structure that allows to customize alternative settings which are listed as follows

- `opts.phi` is a `lags` $\times$ 1 vector that sets the initial value for the φ's (default, 0.5^j for $j = 1, ..., d$).

- `opts.sigma` is a 3 $\times$ 1 vector that sets the initial value for the σ's (default, 0.5).

126

- `opts.index_est` is a row vector that selects the parameters to be optimized. By default, `opts.index_est=1:lags+3`, that is, all parameters are optimized.

- `opts.lb` and `opts.ub` set the lower and upper bounds for the optimization. Both are row array vectors of the same size of `opts.index_est`.

- `opts.max_compute` is a scalar selecting the maximization routine to be employed. The maximization settings are the same as those for the maximization of the Minnesota hyper-parameters (see (3.1.1)).

Multivariate approaches

When the DGP features general equilibrium equations, univariate decompositions are inefficient as they disregard, for example, the presence of a balance growth path when TFP has a unit root, or the fact that cyclical components have similar features (since they are driven by the same disturbances). The next procedure account for these possibilities.

- Beveridge and Nelson (BN) decomposition

The BN decomposition was originally thought for univariate time series but, in most applications, a bivariate process is used. The procedure is flexible and can allow for more than two endogenous variables, as long as sufficient smoothness conditions are imposed, see e.g. Kamber, Morley and Wong (2018). It can also be cast into a state space formulation, which helps to make clear the link with a UC methodology, see e.g. Morley, Nelson and Zivot (2003).

In BN the trend is defined as the long run forecastable component of y_t (recall, in Hamilton local projection approach it was the medium term forecastable component). It assumes that at least one of the components in y_t has a unit root (otherwise the forecastable component is simply the mean of y_t), and the cycle is the difference between y_t and its forecastable component. Estimates of the cycle depend on the lag length of the estimated time series model and the sample size. Thus, we recommend users to experiment with different model specifications.

The univariate setup is the following. Assume that $(\Delta y_t - \bar{y}) = A(L)(\Delta y_{t-1} - \bar{y}) + e_t$, where e_t is iid$(0; \Sigma_e)$, $\bar{y}$ is the mean of y_t and $\det(A(L))$ has all its roots less then 1 in absolute value. The Moving Average representation (MA) is $(\Delta y_t - \bar{y}) = \Delta y_t^* = D(L)e_t = D(1)e_t + \frac{(D(L)-D(1))}{1-L}\Delta e_t$. Cumulating one obtains

$$y_t = (\bar{y} + D(1)\sum_{j=1}^{t} e_j) + \frac{(D(L)-D(1))}{1-L}e_t \equiv y_t^T + y_t^c$$

where y_t^T is the trend and y_t^c is the cycle. Note that, by construction the trend and the cycle are perfectly correlated since they are driven by the same innovation. To call this

decomposition, the user should use the command:

```
[dc,dt] = BNuniv(y,lags,ff,gg, mm);
```

where `y` is the time series, `lags` is the number of lags in the estimated autoregression; `ff` is an indicator function for the constant in the AR model (`ff = 0` no constant); `gg` is an indicator function for whether the estimated cycle is plotted or not (`gg = 0` no plot); `mm` is an indicator function for whether the decomposition is computed using the estimated long run mean from the model or the estimated mean of y_t (`mm = 1` use the estimated mean; `mm = 0` use long run mean).

The multivariate setup is instead the following. Let $y_t = [\Delta y_{1t}, y_{2t}]$, where y_t is a $m \times 1$. Here y_{1t} are $I(1)$ variables; and y_{2t} are I(0). Suppose y_t admits a MA of the form $y_t = \bar{y} + D(\ell)e_t$, where $e_t \sim iid(0, \Sigma_e)$, $D_0 = I$, the roots of $|\det(D(\ell))|$ are equal or greater than one; and that $D_1(1) \neq 0$, where $D_1(\ell)$ is $m_1 \times 1$ (first m_1 rows of $D(\ell)$). Then

$$\begin{pmatrix} \Delta y_{1t} \\ \Delta y_{2t} \end{pmatrix} = \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \end{pmatrix} + \begin{pmatrix} D_1(1) \\ 0 \end{pmatrix} e_t + \begin{pmatrix} (1-\ell)D_1^\dagger(\ell) \\ (1-\ell)D_2^\dagger(\ell) \end{pmatrix} \Delta e_t \tag{10}$$

$D_1^\dagger(\ell) = \frac{D_1(\ell) - D_1(1)}{1-\ell}$ $D_2^\dagger(\ell) = \frac{D_2(\ell)}{1-\ell}$, $0 < \text{rank}[D_1(1)] \leq m_1$ and $y_t^x = [\bar{y}_1 + D_1(1) \sum_s e_s, \bar{y}_2]'$ is the permanent component of y_t.

Note that as in UC, the BN trend is the permanent component of the series. BN however restricts the latent components to be perfectly correlated, while in UC the correlation is a parameter that can be estimated from the data.

• Blanchard and Quah (BQ) decomposition

The Blanchard and Quah decomposition is run on a VAR which has the same format as the one for the multivariate BN decomposition. However, rather than using the innovations to construct estimates of the latent components, one uses rotation matrices to make sure that the shocks of the model have some economic interpretation (typically demand vs. supply shocks). In the typical setup, the decomposition has two variables, where at least one is a $I(1)$ process. The two shocks are identified by the restriction that only one shock affects the $I(1)$ variable in the long run and both affect the $I(0)$ and the $I(1)$ variables in the short run.

When there are more than two variables, balanced growth can be imposed, see King, Plosser, Stock and Watson (1991), and more than one shock can have long run effects on $I(1)$ variables, see Fisher (2006). The standard BQ decomposition identifies supply and demand shocks assuming that they are uncorrelated. Cover, Enders and Hueng (2006) use an alternative normalization and different identification assumptions that allow demand and supply shocks to be correlated.

In the toolkit, the BQ and the (bivariate) BN decompositions can be obtained with one command:

```
[dt1,dc1,dt2,dc2] = BNBQbiv(y,lags,ff,rhoyes,ssts,gg,mm);
```

where `lags` is the number of lags in the VAR; `ff` is an indicator function for whether a constant is included in the VAR (if `ff = 1` there is a constant); `rhoyes` is an indicator function for whether the two disturbances are correlated (if `rhoyes = 1` demand and supply shocks are correlated; this option triggers the Cover et al. (2006) decomposition); `ssts` is an indicator function for whether the second variable of the VAR is also integrated (if `ssts = 1` it is $I(1)$); `gg` is an indicator function for the plots (if `gg = 1` the permanent and transitory component of the first variable are plotted); `mm` is an indicator function for spectral density plots (if `mm = 1` spectral densities plots appear on the screen). Here `y` should be a $T \times 2$ matrix; `dt1`, `dc1` are the permanent and the transitory components of the first series obtained with BN; `dt2`, `dc2` are the permanent transitory components of the first series obtained with BQ.

BN and BQ decompositions are sensitive to the number of lags, the number and the type of variables, whether a constant is included in the VAR and the sample size. We recommend users to experiment with various options before taking one output for inference.

6.2 Dating turning points

There are different concepts of the business cycle researchers use. For example, following Lucas (1977), several macroeconomists think business cycles as the presence variability, serial and cross correlation in a vector of aggregate macroeconomic variables. Such a definition typically requires the separation of observables time series into trends and cycles, and the decompositions presented in the previous subsection can be employed. Other macroeconomists instead prefer to think of business cycles as the sequence of alternating, irregularly spaced turning points generating expansion/recession phases of some minimum duration. Such an approach has been pioneered by Burns-Mitchell (1946) and recently brought back to life by Harding and Pagan (2002, 2006), Kulish and Pagan (2019), and others. The approach can be used to compute cyclical statistics without having to first detrend the data; allows for asymmetries between expansions and recessions, and involves first, dating turning points, and second, computing statistics of business cycle (recession and expansion) phases.

The first step is basically a pattern recognition exercise: one finds turning points under some constraints in the level of y_{it} (or it growth rate). The constraints could be judgmental, as for example NBER/CEPR dating committees use - a persistent period (at least two quarters) of positive/negative growth is called an expansion/recession - or mechanical such as the Bry and Boschen (BB) algorithm, who finds peak and through dates using a procedure to locate local maximum and minimum of a series.

The rules of the BB algorithm are:

1. Peaks and troughs must alternate.

2. Each phase (peak to trough or through to peak) must have a duration of at least six months (two quarters).

3. A cycle (peak to peak or trough to through) must have a duration of at least 15 months (5 quarters).

4. Turning points within six months (2 quarters) of the beginning or end of the series are eliminated. Peaks or troughs within 24 months (8 quarters) of the beginning or end of the sample are eliminated if any of the points after or before are higher (or lower) than the peak (trough).

Note that BB turning point dates may be different from, e.g., NBER/CEPR turning point dates because peaks (troughs) may occur at negative (positive) values and recessions may be uniformly small (there is no sharp through). Note also that turning point dates are sensitive to dating rules 2. and 3. and to minimum amplitude restrictions (e.g. peaks to troughs drops of less than one percent are typically excluded). Thus, experimentation with these parameters is important.

In general, however, the BB algorithm with the basic settings provides a good mechanical rationalization for choices of business cycle dating committees. The table below has been constructed using the BB algorithm present in the toolkit to date GDP turning points in the EURO area. Even if the CEPR dating committee looks at series other than GDP the match seems very reasonable, except at the end of the 1970 beginning of 1980.

Turning point dates: Euro area

Phase	CEPR	Length	BB (GDP)	Length
Peak	1974:3		1973:4	
Through	1975:1	2	1974:1	2
Peak	1980:1	20	1979:2	21
Through	1982:3	10	1979:4	2
Peak	1992:1	38	1991:2	46
Through	1993:3	6	1992:2	4
Peak	2008:1	58	2007:2	60
Through	2009:2	5	2008:3	5
Peak	2011:3	9	2010:4	9
Through	2013:1	6	2012:2	6

The toolkit uses the implementation of the BB algorithm, coded by Harding and Pagan and modified into a usable function by us. The command to date turning points in a series is

```
[dt_] = date_(y, time, freq, tstart, tend, options);
```

where y is the data, time is the time span (same length as y), freq is a string declaring the frequency of the data; that is, 'm' for monthly data and 'q' for quarterly data; tstart (tend) is a (1×2) vector containing the year and the month/quarter when the dating exercise should start (end), e.g. if the sample is 1970Q3-2020Q1, we have tstart = [1970 3] and tend = [2020 1]. The sixth input, options, allows to customize the dating exercise; in particular we have:

- options.phase sets the minimum duration of the estimated phase; default value for quarterly (monthly) data equals 2 (6).

- `options.cycle` sets the minimum length of the estimated cycle; default value for quarterly (monthly) data equals 5 (15).

- `options.thresh` bypasses phase and cycle restriction if peak to trough is larger than threshold; default value equals 10.4.

- `options.complete`. When equals one, it uses only complete cycles, else incomplete cycles (excess still computed on complete cycle). Default value equals one.

- `options.nrep` is used when multiple dataset simulated from a model are considered. So it will run the dating exercise `options.nrep` times rather than one time on `options.nrep` × T observations. Default value equals one.

The output of the function is a field structure containing several objects of interest for business cycles analysis:

- `dt_.st` is a T × 1 a binary array which contains the phases over time, i.e. expansion (`1`) and contraction (`0`).

- `dt_.trinary` is a T × 1 trinary array which contains the cycle states over time, i.e. peak (`1`), trough (`-1`) or none of the two (`0`).

- `dt_.dura` is a 2 × 1 array with the average duration of contractions and the duration of expansions respectively. For more details see below at 6.2

- `dt_.ampl` is a 2×1 array with the average amplitudes of contractions and the amplitude of expansions respectively

- `dt_.cumm` is a 2 × 1 array with the average cumulative change in contractions and expansions respectively

- `dt_.excc` is a 2 × 1 array reporting the percent of excess movements of triangle area for contraction and expansions respectively

- `dt_.durcv` is a 2 × 1 array the coefficient of variation (the inverse of the standard deviation) of the duration of contractions and expansions

- `dt_.amplcv` is a 2 × 1 array the coefficient of variation of the amplitude of contractions and expansions

- `dt_.exccv` is a 2 × 1 array the coefficient of variation of the excess movements in contractions and expansions

If one has more than one series that needs to be dates, the dating function can be put into a loop and run over the available series, e.g.

```
for  qq = 1 : size(Y,2)
    y       = Y(:,qq);
    dt_(qq) = date_(y, time, freq, tstart, tend, options);
    turningpoints(:,qq) = dt_(qq).trinary;
end
```

and `turningpoints` contains as column vectors the turning points of each of the series.

The algorithm works on individual series. Thus, if one has many time series, she has to decide how to aggregate turning point dates and business cycle phases across series to construct "reference" turning point dates and "reference" business cycle phases.

One option is to "aggregate-and-date". This approach involves constructing a coincident indicator of economic activity, e.g. Conference Board (TCB) Indicator in US; picking one relevant series, e.g. GDP, or IP; using a dynamic factor model (DFM) to produce an estimate of f_t; for example:

$$
\begin{aligned}
y_{it} &= \lambda f_t + e_{it} \\
f_t &= a(L)f_{t-1} + u_t \quad u_t \sim (0, \sigma_u^2) \\
e_{it} &= b(L)e_{it-1} + v_{it} \quad v_{it} \sim iidN(0, \sigma_v^2)
\end{aligned}
\tag{11}
$$

or construct an ISD (Index standard deviation) weighting the available time series

$$
I_t = \exp\left(\sum_{i=1}^{N} \alpha_i y_{it}\right)
\tag{12}
$$

where $\alpha_i = \frac{s_i^{-1}}{\sum_{j=1}^{N} s_j^{-1}}$ and s_i is the standard deviation of y_{it}. Then, the dating exercise is performed using TCB, GDP, f_t or I_t. As shown in Stock and Watson (2014), the time paths of ISD, DFM and TCB are similar and thus turning points dates are roughly coincident.

The alternative approach "date-and-aggregate" computes turning points τ_{is} for each series $y_{it}, i = 1, \ldots, n_s$ in episode s and then compute a location measure of the turning points distribution for each identified phase (e.g. NBER recession). If turning points are iid:

$$
\begin{aligned}
n_s^{0.5}(\hat{\tau}_s^{mean} - \tau_s^{mean}) &\xrightarrow{D} N(0, var(\tau_{is})) \\
n_s^{0.5}(\hat{\tau}_s^{median} - \tau_s^{median}) &\xrightarrow{D} N\left(0, \frac{1}{4(g_s(\tau_s))^2}\right) \\
(h^3 n_s)^{0.5}(\hat{\tau}_s^{mode} - \tau_s^{mode}) &\xrightarrow{D} N\left(0, \frac{g_s(\tau_s^{mode})\int[K'(z)]^2 dz}{g_s''(\tau_s^{mode})}\right)
\end{aligned}
$$

where $K(.)$ is a kernel, h the length of the kernel, $g_s(\tau)$ is the distribution of τ in episode s (see Stock and Watson 2014).

When certain types of series are over-represented in the sample relative to the population (e.g. there too many IP series and too few employment series), one can use weights in the construction of the mean:

$$
w_{i,s} = \frac{\pi_{m_i}}{p_{m_i,s}}
\tag{13}
$$

where π_m is the population probability of class m series (IPs, employments, interest rates, etc.) and $p_{m,s}$ is the sample probability of class m in business cycle episode s.

There are a number of other possibilities to compute location measures of the distribution. For example, Harding and Pagan (2006) assume that a "reference phase" is in place if at least 50 per cent of the series are in that particular phase. Kulish and Pagan (2019) suggest

to compute reference turning point dates by minimizing the discrepancy among individual series turning point dates, i.e. if peaks are at 1973:1, 1973:5, 1973:9, the reference peak is 1973:5. One can also construct a (subjectively) weighted average of turning points where the weight depends on importance of individual series (GDP turning points may have more weights than, say, consumption turning points).

Example 39 (Euro area Turning point dates) *We consider quarterly Euro Area series on output, consumption, investment, labor productivity, employment, nominal rate and inflation from 1970Q1 until 2019Q3 available in the AWM DATABASE. We apply the BB algorithm to each of the series and plot the distribution of turning points across series for each data in the sample (+1 peak,-1 through), see Figure 33.*

• 1975:1 is it a though? If we use the 50% rule, it is not, since less than 50% of the series have a through at that date.

• Is there a through in 2008? If we use the minimal distance criteria, the through date is 2009:2 (two series have minimum in 2008:1, two in 2010:3 and three have no minimum in that interval).

As a general suggestion, if one wants to apply these simple rules, it is important to have a good number of coincident series; otherwise a lot of measurement error may be present.

Figure 33: Distribution of Turning point dates: Euro area.
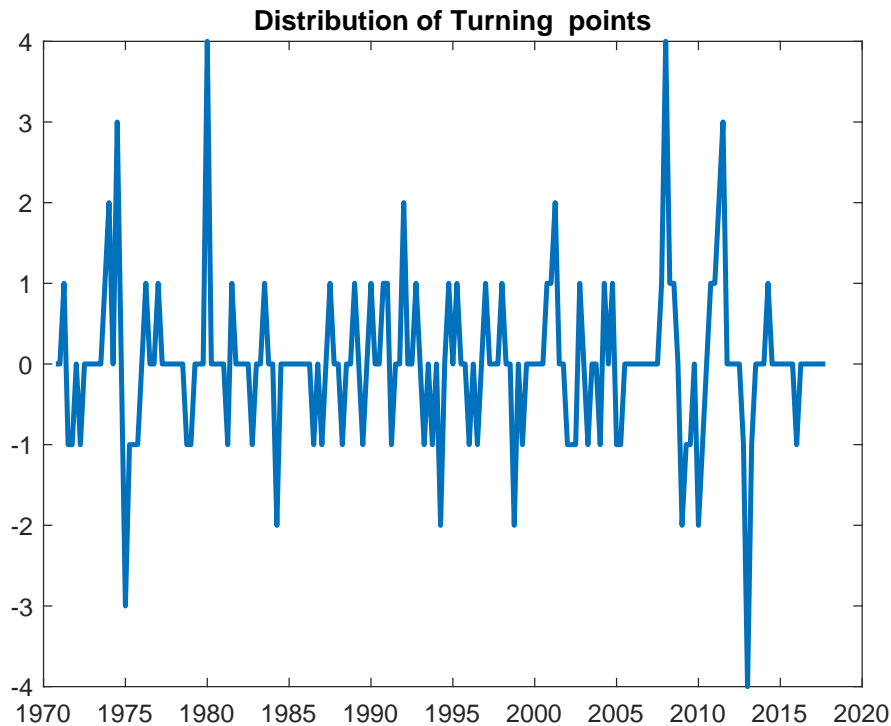
Once turning points and reference phases are constructed one can compute a number statistics:

- Average durations (AD), i.e. the average length of time spent between troughs and peaks or peaks and troughs.

- Average amplitudes (AA), i.e. the average magnitude of the drop between peaks and troughs or of the gain between troughs and peaks.

- Average cumulative changes over phases ($CM = 0.5(AD \times AA)$) and excess average cumulative changes $ECM = ((CM - CM^A + 0.5 \times AA)/AD)$, where CM^A is the actual average cumulative change. This is a measure of the average cost (gains) due to recessions (expansions).

- Concordance index $CI_{j,j'} = n^{-1}[\sum \mathcal{I}_{jt}\mathcal{I}_{it} - (1-\mathcal{I}_{jt})(1-\mathcal{I}_{it})]$. It measures comovements over business cycle phases of two variables, where n is the number of complete cycles and $\mathcal{I}_{it} = 1$ in expansions and $\mathcal{I}_{it} = 0$ in contractions. $CI = 1(= 0)$ if the two series are perfectly positively (negatively) correlated.

As described, the dating function produces as output dates of turning points but also amplitudes, durations, CMs and ECMs. To compute concordances one has two options. Concordances, in fact, can be computed for turning point dates or for business cycle phases. Assume that we have dated the business cycles for 7 different series (GDP, consumption, investment, labor productivity, interest rate and inflation), and the output is collected in `dt_(1)`, ..., `dt_(7)`. The following are simple Matlab commands that allow the computation of concordances.

```
% GDP, C, Inv, Y/N, N, R, pi
zz =  [dt_(1:7).st];
disp('concordance index BC phases')
ma = corr(zz,'type','Spearman');
disp(ma(1, 2 : size(ma,2)))
turn =  [dt_(1:7).trinary];
disp('concordance turning points')
qa = corr(turn,'type','Spearman');
disp(qa(1, 2 : size(qa,2)))
```

where `zz` is the matrix containing business cycle phases and `turn` is the matrix containing business cycle turning points for different series.

Example 40 (Euro area business cycle statistics) *Using With the turning point dates of exercise 39, we compute statistics of each phase. Three facts are evident. First, there are large asymmetries in durations and amplitudes. Second, output and consumption expansions are longer and stronger than in the other series. Third, there is very low concordance between real and nominal series.*

Euro area Business Cycle Statistics

	Average Duration (quarters)		Average amplitude (%)		ECM (%)		$CI_{j,j'}$ (phase)
	PT	TP	PT	TP	PT	TP	
GDP	3.8	33.7	-2.5	20.9	6.7	1.9	
C	5	36.6	-1.5	19.2	9.8	4.4	0.57
Inv	6.7	14.7	-7.2	14.7	14.9	1.1	0.52
Y/N	2.0	18.6	-1.2	8.9	1.7	10.1	0.61
N	9.0	22.8	-1.8	6.1	7.0	11.8	0.45
R	8.4	6.6	-3.1	2.7	10.5	7.8	0.04
π	9.0	6.9	-6.1	5.6	0.34	12.0	0.15

7 Final thoughts

As we have mentioned in the introduction, the toolbox comes as it is, with no support or help on our part. The small scripts we used as examples work well, and have been tested with generations of graduate students and numerous policy rounds. They should be used as blue-prints for building codes, users may want to employ.

If the reader has never seen MATLAB before, it is probably impossible to properly use the toolbox after reading this paper. Thus, we suggest to look at the MATLAB manual first. After that, we encourage the user to modify the examples, for instance using your favourite dataset, your favourite VAR specification, or your favourite identification scheme, to gain confidence. Only after that, it is logic to try to do something on your own. The toolbox has been written with the scope of making life easier for applied researchers who want to do standard stuff, building stylized facts, or running simple checks on the data. In case you use the toolbox to produce something completely new, we will be interested in knowing what twist you have added.

If you are an experienced MATLAB user, you can access the codes and see yourself how things are done. We did not strive for beauty or efficiency; most of the examples take only a few second on standard machines with 32RAM of memory, so efficiency was never a consideration for us. The codes are public. Feel free to modify them as you wish for your own purposes.

It took us quite a long time to put together the routines in a consistent way and we hope that the work will be useful for applied researchers and for teaching purposes. We think that the toolbox could be very valuable when teaching applied macro for advanced MA students or first year Ph.D. students, but some of the functions could be useful also for later courses.

We will take a break for a while, it has been a long road up to here. We plan to go back to programming sometimes in the future and add other tasks as they will became relevant for applied macroeconomists. Suggestions of what these may be are welcome.

Happy trails to all!

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.

Altavilla, C., Brugnolini, L., Gürkaynak, R. S., Motto, R. and Ragusa, G.: 2019, Measuring Euro Area Monetary Policy, *Journal of Monetary Economics* **108**, 162–179.

Altavilla, C., Canova, F. and Ciccarelli, M.: 2020, Mending the broken link: Heterogeneous bank lending rates and monetary policy pass-through, *Journal of Monetary Economics* **110**, 81–98.

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.: 2018, Narrative Sign Restrictions for SVARs, *American Economic Review* **108**(10), 2802–2829.

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.

Bai, J. and Ng, S.: 2002, Determining the Number of Factors in Approximate Factor Models, *Econometrica* **70**(1), 191–221.

Bai, J. and Ng, S.: 2007, Determining the Number of Primitive Shocks in Factor Models, *Journal of Business and Economic Statistics* **25**, 52–60.

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

Baxter, M. and King, R. G.: 1999, Measuring Business Cycles: Approximate Band-Pass Filters For Economic Time Series, *The Review of Economics and Statistics* **81**(4), 575–593.

Ben Zeev, N.: 2018, What can we learn about news shocks from the late 1990s and early 2000s boom-bust period?, *Journal of Economic Dynamics and Control* **87**(C), 94–105.

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.

Borio, C.: 2012, The financial cycle and macroeconomics: what have we learned?, *Technical report*, BIS working paper 2012.

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

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.: 2020, FAQ: How do I extract the output gap, *Working Paper Series DP14943*, C.E.P.R.

Canova, F. and Ciccarelli, M.: 2009, Estimating Multi-country VAR Models, *International Economic Review* **50**(3), 929–961.

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.: 2021, Mind the gap! Stylized dynamic facts and structural models, *American Economic Journal: Macroeconomics* **forthcoming**.

Canova, F. and Pappa, E.: 2007, Price Differentials in Monetary Unions: the role of fiscal shocks, *Economic Journal* **117**(68), 713–737.

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.

Christiano, L. J. and Fitzgerald, T. J.: 2003, The Band Pass Filter, *International Economic Review* **44**(2), 435–465.

Cover, J. P., Enders, W. and Hueng, C. J.: 2006, Using the Aggregate Demand-Aggregate Supply Model to Identify Structural Demand-Side and Supply-Side Shocks: Results Using a Bivariate VAR, *Journal of Money, Credit and Banking* **38**(3), 777–790.

De Graeve, F. and Karas, A.: 2014, Evaluating Theories of Bank Runs with Heterogeneity Restrictions?, *Journal of the European Economic Association* **12**(5), 1166–1188.

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.

Diebold, F. X. and Yilmaz, K.: 2009, Measuring Financial Asset Return and Volatility Spillovers, with Application to Global Equity Markets, *Economic Journal* **119**(534), 158–171.

Diebold, F. X. and Yilmaz, K.: 2012, Better to give than to receive: Predictive directional measurement of volatility spillovers, *International Journal of Forecasting* **28**(1), 57–66.

Dieppe, A., van Roye, B. and Legrand, R.: 2016, The BEAR toolbox, *Working Paper Series 1934*, European Central Bank.

Durbin, J. and Koopman, S. J.: 2001, *Time Series Analysis by State Space Methods*, 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.

Geweke, J.: 1999, Using simulation methods for Bayesian econometric models: inference, development, and communication, *Econometric Reviews* **18**(1), 1–73.

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

Gonzalez-Astudillo, M. and Roberts, J. M.: 2016, When Can Trend-Cycle Decompositions Be Trusted?, *Finance and Economics Discussion Series 2016-099*, Board of Governors of the Federal Reserve System (U.S.).

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

Grant, A. L. and Chan, J. C.: 2017, Reconciling output gaps: Unobserved components model and Hodrick-Prescott filter, *Journal of Economic Dynamics and Control* **75**(C), 114–121.

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

Hamilton, J. D.: 2018, Why you should never use the hodrick-prescott filter, *The Review of Economics and Statistics* **100**(5), 831–843.

Harding, D. and Pagan, A.: 2002, Dissecting the cycle: a methodological investigation, *Journal of Monetary Economics* **49**(2), 365–381.

Harding, D. and Pagan, A.: 2006, Measurement of Business Cycles, *Department of Economics Working Papers Series 966*, The University of Melbourne.

Harvey, A. C.: 1990, *Forecasting, Structural Time Series Models and the Kalman Filter*, number 9780521321969 in *Cambridge Books*, Cambridge University Press.

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

Kamber, G., Morley, J. and Wong, B.: 2018, Intuitive and Reliable Estimates of the Output Gap from a Beveridge-Nelson Filter, *The Review of Economics and Statistics* **100**(3), 550–566.

Kilian, L. and Murphy, D. P.: 2012, Why Agnostic Sign Restrictions Are Not Enough: Understanding The Dynamics Of Oil Market Var Models, *Journal of the European Economic Association* **10**(5), 1166–1188.

King, R. G., Plosser, C. I., Stock, J. H. and Watson, M. W.: 1991, Stochastic Trends and Economic Fluctuations, *American Economic Review* **81**(4), 819–840.

Kulish, M. and Pagan, A.: 2019, Turning point and oscillatory cycles, *CAMA Working Papers 2019-74*, Centre for Applied Macroeconomic Analysis, Crawford School of Public Policy, The Australian National University.

Lenza, M. and Primiceri, G. E.: 2020, How to estimate a VAR after March 2020, *Working Paper Series 2461*, European Central Bank.

Lubik, T. A., Matthes, C. and Verona, F.: 2019, Assessing U.S. Aggregate Fluctuations Across Time and Frequencies, *Working Paper 19-6*, Federal Reserve Bank of Richmond.

Lucas, R. E.: 1977, Understanding business cycles, *Carnegie-Rochester Conference Series on Public Policy* **5**(1), 7–29.

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.

Meyer-Gohde, A.: 2010, Matlab code for one-sided HP-filters, *Technical report*, QM RBC Codes, Quantitative Macroeconomics Real Business Cycles.

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

Morley, J. C., Nelson, C. R. and Zivot, E.: 2003, Why Are the Beveridge-Nelson and Unobserved-Components Decompositions of GDP So Different?, *The Review of Economics and Statistics* **85**(2), 235–243.

Pesaran, H. H. and Shin, Y.: 1998, Generalized impulse response analysis in linear multivariate models, *Economics Letters* **58**(1), 17–29.

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.

Ravn, M. O. and Uhlig, H.: 2002, On adjusting the Hodrick-Prescott filter for the frequency of observations, *The Review of Economics and Statistics* **84**(2), 371–375.

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.

Runstler, G. and Vlekke, M.: 2018, Business, housing, and credit cycles, *Journal of Applied Econometrics* **33**(2), 212–226.

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.: 2020, Svar identification through heteroskedasticity with misspecified regimes, *Technical report*, Mimeo.

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.: 1999, Business cycle fluctuations in us macroeconomic time series, *in* J. B. Taylor and M. Woodford (eds), *Handbook of Macroeconomics*, Vol. 1 of *Handbook of Macroeconomics*, Elsevier, chapter 1, pp. 3–64.

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

Stock, J. H. and Watson, M. W.: 2014, Estimating turning points using large data sets, *Journal of Econometrics* **178**(P2), 368–381.

Stock, J. H. and Watson, M. W.: 2016, Core Inflation and Trend Inflation, *The Review of Economics and Statistics* **98**(4), 770–784.

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 the VAR 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 a VAR to a VMA

The $VAR(p)$ can be written in a VMA format 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 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

$$\Psi_0 = I$$

$$\Psi_1 = \Phi_1$$

$$\Psi_2 = \Psi_1 \Phi_1 + \Psi_0 \Phi_2$$

$$\Psi_3 = \Psi_2 \Phi_1 + \Psi_1 \Phi_2 + \Psi_0 \Phi_3$$

$$\vdots$$

$$\Psi_j = \sum_{i=1}^{j} \Psi_{j-i} \Phi_i \quad \text{for } j < p$$

$$\Psi_j = \Psi_{j-1} \Phi_1 + \Psi_{j-2} \Phi_2 + \cdots + \Psi_{j-p} \Phi_p \quad \text{for } j \geq p$$

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 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

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

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. Fro lag selection, the information criteria considered are

$$AIC(p) = -2 \ln L + \frac{2}{T} pn^2$$

$$HQIC(p) = -2 \ln L + \frac{2 \ln \ln T}{T} pn^2$$

$$BIC(p) = -2 \ln L + \frac{\ln T}{T} pn^2$$

where $\ln L$ is the log likelihood of the data evaluated at the OLS estimator, i.e. $\ln p(Y|X, \widehat{\Phi}, S_{ols})$ see below.

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.11, $Y - X\Phi$ is also multivariate normal and the conditional likelihood has the following expression

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

This function can be conveniently expressed as (see Zellner (1971))

$$p(Y|X, \Phi, \Sigma) = (2\pi)^{-Tn/2} |\Sigma|^{-T/2} \tag{14}$$
$$\exp\left(-1/2tr(\Sigma^{-1}\widehat{S})\right) \times \exp\left(-1/2tr(\Sigma^{-1}(\Phi - \widehat{\Phi})'X'X(\Phi - \widehat{\Phi}))\right)$$
$$= p(\Sigma|Y, X)p(\Phi|\Sigma, Y, X) \tag{15}$$

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:

$$(Y - X\Phi)'(Y - X\Phi) = (Y - X\widehat{\Phi} + X\widehat{\Phi} - X\Phi)'(Y - X\widehat{\Phi} + X\widehat{\Phi} - X\Phi)$$
$$= \widehat{S} + (\widehat{\Phi} - \Phi)X'X(\widehat{\Phi} - \Phi)$$

The likelihood in (15) 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 $(\Phi, \Sigma|Y)$[18]:

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

Therefore,

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

Thus, draws from the posterior of the parameters 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 DISTRIBUTIONS WITH CONJUGATE PRIORS

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

$$\Phi \sim N(\Phi_0, \Sigma \otimes V) \quad = (2\pi)^{-nk/2}|\Sigma|^{-k/2}|V|^{-n/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}(\Phi - \Phi_0)'V^{-1}(\Phi - \Phi_0)\right]\right\}$$

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

---

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

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

$$\Phi|\Sigma, Y, X, \sim N(\overline{\Phi}, \Sigma \otimes (X'X + V^{-1})^{-1})$$

$$\Sigma|Y, X \sim IW(\overline{S}, T + d)$$

where

$$\overline{\Phi} = (X'X + V^{-1})^{-1}(X'Y + V^{-1}\Phi_0)$$

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

$$= \Sigma_0 + (Y - X\overline{\Phi})'(Y - X\overline{\Phi}) + (\overline{\Phi} - \Phi_0)'V^{-1}(\overline{\Phi} - \Phi_0)$$

The process to draw from this joint posterior 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 $N(\overline{\Phi}, \Sigma^j \otimes (X'X)^{-1})$

A.4.1   Derivation the MN-IW Conjugate result

The posterior distribution can be conveniently expressed as

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

$$(2\pi)^{-Tn/2}\, |\Sigma|^{-T/2} \exp\left(-1/2 tr(\Sigma^{-1}\widehat{S})\right) \exp\left(-1/2 tr(\Sigma^{-1}(\Phi - \widehat{\Phi})'X'X(\Phi - \widehat{\Phi}))\right) \quad \text{[Likelihood]}$$

$$(2\pi)^{-nk/2}|\Sigma|^{-k/2}|V|^{-n/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}(\Phi - \Phi_0)'V^{-1}(\Phi - \Phi_0)\right]\right\} \quad \text{[Prior } \Phi]$$

$$\frac{|\Sigma_0|^{d/2}}{2^{dn/2}\Gamma_n(d/2)}|\Sigma|^{-(n+d+1)/2} \exp\left\{-\frac{1}{2}tr\left[\Sigma_0\Sigma^{-1}\right]\right\} \quad \text{[Prior } \Sigma]$$

Rearranging terms

$$p(\Sigma, \Phi|Y) \propto (2\pi)^{-(T-k)n/2}\frac{|\Sigma_0|^{d/2}}{2^{dn/2}\Gamma_n(d/2)}|V|^{-n/2}\times$$

$$|\Sigma|^{-T/2}|\Sigma|^{-k/2}|\Sigma|^{-(n+d+1)/2}\times$$

$$\exp\left\{-\frac{1}{2}tr\left[(\widehat{S} + \Sigma_0)\Sigma^{-1}\right]\right\}\times$$

$$\exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}\left(\underbrace{(\Phi - \widehat{\Phi})'X'X(\Phi - \widehat{\Phi}) + (\Phi - \Phi_0)'V^{-1}(\Phi - \Phi_0)}_{A}\right)\right]\right\}$$

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

$$(\Phi - \widehat{\Phi})'X'X(\Phi - \widehat{\Phi}) + (\Phi - \widehat{\Phi} + \widehat{\Phi} - \Phi_0)'V^{-1}(\Phi - \widehat{\Phi} + \widehat{\Phi} - \Phi_0)$$

$$= (\Phi - \widehat{\Phi})' \left[X'X(\Phi - \widehat{\Phi}) + V^{-1}(\Phi - \widehat{\Phi}) + V^{-1}(\widehat{\Phi} - \Phi_0) \right] + \underbrace{(\widehat{\Phi} - \Phi_0)'V^{-1}(\Phi - \widehat{\Phi})}_{A_1} + \underbrace{(\widehat{\Phi} - \Phi_0)'V^{-1}(\widehat{\Phi} - \Phi_0)}_{A_2}$$

$$= (\Phi - \widehat{\Phi})' \left[X'X\Phi - X'X\widehat{\Phi} + V^{-1}\Phi - V^{-1}\widehat{\Phi} + V^{-1}\widehat{\Phi} - V^{-1}\Phi_0 \right] + A_1 + A_2$$

$$= (\Phi - \widehat{\Phi})' \left[(X'X + V^{-1})\Phi - X'X\widehat{\Phi} - V^{-1}\Phi_0 \right] + A_1 + A_2$$

$$= (\Phi - \widehat{\Phi})'(X'X + V^{-1}) \left[\Phi - \underbrace{(X'X + V^{-1})^{-1}(X'X\widehat{\Phi} + V^{-1}\Phi_0)}_{\overline{\Phi}} \right] + A_1 + A_2$$

$$= (\Phi - \widehat{\Phi})'(X'X + V^{-1}) \left[\Phi - \overline{\Phi} \right] + A_1 + A_2$$

$$= (\Phi - \overline{\Phi} + \overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1}) \left[\Phi - \overline{\Phi} \right] + A_1 + A_2$$

$$= \underbrace{(\Phi - \overline{\Phi})'(X'X + V^{-1}) \left[\Phi - \overline{\Phi} \right]}_{A_0} + (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1}) \left[\Phi - \overline{\Phi} \right] + A_1 + A_2$$

$$= A_0 + (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1})(\Phi - \overline{\Phi}) + (\widehat{\Phi} - \Phi_0)'V^{-1}(\Phi - \widehat{\Phi}) + A_2$$

$$= A_0 + (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1})\Phi - (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1})\overline{\Phi} + (\widehat{\Phi} - \Phi_0)'V^{-1}\Phi - (\widehat{\Phi} - \Phi_0)'V^{-1}\widehat{\Phi} + A_2$$

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

It is easy to show that $A_3 = 0$

$$A_3 = \overline{\Phi}'(X'X + V^{-1}) - \widehat{\Phi}'(X'X + V^{-1}) + \widehat{\Phi}'V^{-1} - \Phi_0'V^{-1}$$

$$= (X'X\widehat{\Phi} + V^{-1}\Phi_0)'(X'X + V^{-1})^{-1}(X'X + V^{-1}) - \widehat{\Phi}'(X'X + V^{-1}) + \widehat{\Phi}'V^{-1} - \Phi_0'V^{-1}$$

$$= \widehat{\Phi}'X'X + \Phi_0'V^{-1} - \widehat{\Phi}'X'X - \widehat{\Phi}'V^{-1} + \widehat{\Phi}'V^{-1} - \Phi_0'V^{-1}$$

$$= 0$$

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

$$A_2 - A_4 =$$

$$= (\widehat{\Phi} - \Phi_0)'V^{-1}(\widehat{\Phi} - \Phi_0) - (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1})\overline{\Phi} - (\widehat{\Phi} - \Phi_0)'V^{-1}\widehat{\Phi}$$

$$= -(\widehat{\Phi} - \Phi_0)'V^{-1}\Phi_0 - (\overline{\Phi} - \widehat{\Phi})'(X'X + V^{-1})\overline{\Phi}$$

$$= -\widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi} + \widehat{\Phi}'(X'X + V^{-1})\overline{\Phi}$$

$$= \widehat{\Phi}'(X'X + V^{-1})\overline{\Phi} - \widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi}$$

$$= \widehat{\Phi}'(X'X + V^{-1})(X'X + V^{-1})^{-1}(X'X\widehat{\Phi} + V^{-1}\Phi_0) - \widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi}$$

$$= \widehat{\Phi}'X'X\widehat{\Phi} + \widehat{\Phi}'V^{-1}\Phi_0 - \widehat{\Phi}'V^{-1}\Phi_0 + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi}$$

$$= \widehat{\Phi}'X'X\widehat{\Phi} + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi}$$

The posterior distribution is then

$$p(\Sigma, \Phi|Y) = \text{const} \times |\Sigma|^{-(T+k+n+d+1)/2} \times$$

$$\exp\left\{-\frac{1}{2}tr\left[\left(\widehat{S} + \Sigma_0 + \widehat{\Phi}'X'X\widehat{\Phi} + \Phi_0'V^{-1}\Phi_0 - \overline{\Phi}'(X'X + V^{-1})\overline{\Phi}\right)\Sigma^{-1}\right]\right\} \qquad [p(\Sigma|Y) \text{ kernel of a IW}]$$

$$\exp\left\{-\frac{1}{2}tr\left[\Sigma^{-1}(\Phi - \overline{\Phi})'(X'X + V^{-1})(\Phi - \overline{\Phi})\right]\right\} \qquad [p(\Phi|Y,\Sigma) \text{ kernel of a MN}]$$

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,

$$\ln p(Y|X) = -(Tn/2)\ln\pi + \ln\Gamma_n\left(\frac{T+d}{2}\right) - \ln\Gamma_n\left(\frac{d}{2}\right) + (d/2)\ln|\Sigma_0|$$

$$- (n/2)\ln|V| - (n/2)\ln|X'X + V^{-1}| - \frac{T+d}{2}\ln|\overline{S}|$$

To avoid numerical issues, we 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, the log marginal likelihood is

$$\ln p(Y|X) = -(Tn/2)\ln\pi + \ln\Gamma_n\left(\frac{T+d}{2}\right) - \ln\Gamma_n\left(\frac{d}{2}\right) - (T/2)\ln|\Sigma_0| - (n/2)\ln|I + F_V'X'XF_V|$$

$$- \frac{T+d}{2}\ln|I + F_0'[(Y - X\overline{\Phi})'(Y - X\overline{\Phi}) + (\overline{\Phi} - \Phi_0)'V^{-1}(\overline{\Phi} - \Phi_0)]F_0|$$

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.

Next we 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

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

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

$$E(\phi_{11,2}\phi_{21,2}) = E((-u_{31}/(\tau\underline{s_1}2^d))(-u_{32}/(\tau\underline{s_1}2^d))) = \Sigma_{12}/(\tau\underline{s_1}2^d)^2$$

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:

$$(\phi_{11,1} + \phi_{11,2}) \sim N(1, \Sigma_{11}/(\lambda\underline{y_1})^2)$$

$$(\phi_{21,1} + \phi_{21,2}) \sim N(0, \Sigma_{22}/(\lambda\underline{y_1})^2)$$

$$E((\phi_{11,1} + \phi_{11,2})(\phi_{21,1} + \phi_{21,2})) = \Sigma_{12}/(\lambda\underline{y_1})^2$$

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

$$\phi_{1,0} = \underline{y_1} - \underline{y_1}(\phi_{11,1} + \phi_{11,2}) - \underline{y_2}(\phi_{12,1} + \phi_{12,2}) - (1/\mu)u_{71}$$

$$\phi_{1,0} \sim N(0, \Sigma_{11}(\lambda^{-2} + \mu^{-2}) + \Sigma_{22}\lambda^{-2})$$

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^*:

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

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

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

A.6   Posterior Distributions for Factor Models

In this section we describe how to generate draws from the posterior distribution of the parameters of static or dynamic factor models. In both cases, the Gibbs sampler used to generate draws from the VAR model parameters needs to be augmented with a steps where we draw from the distribution of the latent states.

Consider the static factor model of section 4.2. When we assume a normal distribution for the factors and for the idiosyncratic errors, the joint distribution of the observable variables and the unobserved factors conditional on time $t-1$ information is given by

$$\begin{pmatrix} y_t \\ f_t \end{pmatrix} | y^{t-1}, \Lambda, \Sigma_e \sim N \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} \Sigma_y & \Lambda \\ \Lambda' & I \end{bmatrix} \right)$$

where $\Sigma_y = \Lambda\Lambda' + \Sigma_e$ and y^{t-1} summarizes the history of past realizations of the observable. Since the model is static, past values do not influence the conditional distribution which is time invariant. By the properties of marginal and conditional multivariate normal (Gaussian)

distribution we have that the distribution of the factors conditional on y_t (and conditional on the parameters) is given by

$$f_t \mid y_t, \Lambda, \Sigma_e \sim N\left(\Lambda' \Sigma_y^{-1} y_t, I - \Lambda' \Sigma_y^{-1} \Lambda\right) \tag{16}$$

Assuming as priors for Λ and Σ_e a multivariate normal distribution and an inverse Wishart for respectively, the posterior distributions of Λ and Σ_e conditional on $\{f_t, y_t\}_{t=1}^T$ is a MN-IW. Hence, posterior draws can be generated using the Gibbs Sampler algorithm, described in A.4, with additional step in which the static factors are drawn randomly from (16). Thus, draws from the posterior of the parameters can be obtained as follows:

1. Conditional on Λ and Σ_e, use (16) to generate a draw of the latent factors for $t = 1, ..., T$

2. Given $\{y_t, f_t\}_{t=1}^T$ generate a draw for the parameters Λ and Σ_e the Gibbs Sampler algorithm described in A.4, substituting the lagged observable variables in X with the factors.

Consider now the dynamic factor model of section 4.3. Assume without loss of generality one lag in the transition equation of the latent factors. Let $\vartheta = [\Lambda, \Sigma_e, \Phi, \Sigma]$. As before, with the normality assumptions the joint distribution of observable variables and latent factor is normal. Unlike in the previous case, the joint distribution conditional on time $t-1$ information has time varying first and second moments. Given $f_{t|t-1}$ and $P_{t|t-1}$ we have

$$\begin{pmatrix} y_t \\ f_t \end{pmatrix} \mid y^{t-1}, f_{t|t-1}, P_{t|t-1}, \vartheta \sim N\left(\begin{bmatrix} y_{t|t-1} \\ f_{t|t-1} \end{bmatrix} \begin{bmatrix} \Sigma_{t|t-1} & H_{t|t-1} \\ H'_{t|t-1} & P_{t|t-1} \end{bmatrix} \right)$$

where

$$y_{t|t-1} = E(y_t | y^{t-1}) = \Lambda f_{t|t-1} \tag{17}$$

$$\Sigma_{t|t-1} = E(y_t - y_{t|t-1})(y_t - y_{t|t-1})' = E(\Lambda f_t + e_t - \Lambda f_{t|t-1})(\Lambda f_t + e_t - \Lambda f_{t|t-1})'$$

$$= \Lambda E(f_t - f_{t|t-1})(f_t - f_{t|t-1})'\Lambda' + \Sigma_e = \Lambda\, P_{t|t-1}\, \Lambda' + \Sigma_e$$

$$H_{t|t-1} = E(y_t - y_{t|t-1})(f_t - f_{t|t-1})' = E(\Lambda f_t + e_t - \Lambda f_{t|t-1})(f_t - f_{t|t-1})' = \Lambda P_{t|t-1}$$

$$K_{t|t-1} = H'_{t|t-1} \Sigma_{t|t-1}^{-1} = P_{t|t-1} \Lambda'(\Lambda P_{t|t-1}\, \Lambda' + \Sigma_e)^{-1}$$

The last term represents the Kalman gain, i.e. the covariance between factors and observables scaled by the variance of the observables.

The marginal distribution of the latent factors conditional on the information available at time t is given by

$$f_t \mid y_t, f_{t|t-1}, P_{t|t-1}, \vartheta \sim N\left(f_{t|t-1} + K_{t|t-1}(y_t - y_{t|t-1}), P_{t|t-1} - H'_{t|t-1}\Sigma_{t|t-1}^{-1}H_{t|t-1} \right) \tag{18}$$

$$f_{t+1|t} = \Phi f_{t|t}$$

$$P_{t+1|t} = E(f_{t+1} - f_{t+1|t})(f_{t+1} - f_{t+1|t})' = E(\Phi x_t + u_t - \Phi x_{t|t})(\Phi x_t + u_t - \Phi x_{t|t})' =$$

$$= \Phi P_{t|t}\Phi' + \Sigma$$

Given an estimate of the latent factor, it is fairly straightforward to draw from the conditional distribution of the factor model parameters. With the usual distribution assumptions of normality for the coefficients (Φ and Λ) and inverse Wishart for covariances (Σ_u and Σ_e), the resulting posteriors are MN-IW conditional on $\{f_t, y_t\}_{t=1}^T$. Thus, draws from the posterior of the parameters can be obtained as follows:

1. Conditional on Φ, Λ, Σ_u and Σ_e, use (17)-(18) to get an estimate of the latent factors, $\{f_{t|t}\}_{t=1}^T$. With the Carter-Kohn backward sampling algorithm generate a draw of $\{f_{t|T}\}_{t=1}^T$,

2. Conditional on $\{f_{t|T}\}_{t=1}^T$, generate a draw for the parameters Φ and Σ_u of the transition equation using the Gibbs Sampler algorithm described in A.4,

3. Given $\{y_t, f_{t|T}\}_{t=1}^T$ generate a draw for the parameters Λ and Σ_e in the measurement equation using the Gibbs Sampler algorithm described in A.4, substituting the lagged observable variables in X with the factors.

The Gibbs sampler used to estimate the MFVAR is very similar to the algorithm described above. The only difference lies in step 3. In the MFVAR, we do not need to generate draws from the parameters of the measurement equations since they capture either identities or temporal aggregations.

A.7   PRIORS AND POSTERIORS FOR DIRECT METHODS

Assume that the prior is given by[19]

$$\underset{n \times n}{\Sigma_{(h)}} \sim IW(S_h, d)$$

$$S_h \equiv \sum_{j=0}^{h} F^{h-j} G \Sigma G' F^{h-j'}$$

$$\begin{bmatrix} \beta_{(h)} & \alpha_{(h)} \\ n \times np & n \times 1 \end{bmatrix} \sim N(B_h, \Sigma_{(h)} \otimes \frac{1}{\tau_h} V)$$

$$B_h \equiv \begin{bmatrix} \underbrace{G'F^h}_{n \times np} & \underbrace{G'(I_{np} - F)^{-1}(I_{np} - F^h)F_0}_{n \times 1} \end{bmatrix}$$

where d are the prior degrees of freedom (usually number of regressors minus 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 identity matrix). Because

---

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

the prior is conjugate, the posterior also has a N-IW format:

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

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

where

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

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

A.8   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

$$E(m_t \ \nu'_t) = [\rho, \mathbf{0}]$$
$$E(m_t \ \nu'_t)\Omega' = [\rho, \mathbf{0}] \ \Omega'$$
$$E(m_t \ u'_t) = [\rho \ \Omega_{1,1}, \ \rho \ \Omega'_{2,1}]$$
$$E(m_t \ u'_{2,t})E(m_t \ u_{1,t})^{-1} = \Omega'_{2,1}\Omega_{1,1}^{-1}$$

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

$$\frac{1}{T}\sum m_t u'_{2,t} \left(\frac{1}{T}\sum m_t u_{1,t}\right)^{-1} \rightarrow E(m_t u'_{2,t})E(m_t u_{1,t})^{-1} = \Omega'_{2,1}\Omega_{1,1}^{-1} \tag{19}$$

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 (19) is:

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

$$=\frac{1}{T}\sum m_t u_{2,t}'\left(\frac{1}{T}\sum m_t^2\right)^{-1}\frac{\widehat{\rho}}{\widehat{\rho}^2}$$

$$=\frac{1}{T}\sum(\widehat{\rho}m_t)u_{2,t}'\left(\frac{1}{T}\sum(\widehat{\rho}m_t)^2\right)^{-1}$$

$$=\frac{1}{T}\sum\widehat{u}_{1,t}u_{2,t}'\left(\frac{1}{T}\sum\widehat{u}_{1,t}^2\right)^{-1}$$

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:

$$u_{1,t} = \rho_0 + \rho_1 m_t + e_{1,t} \tag{20}$$

$$u_{2,t} = b_0 + b_1 \widehat{u}_{1,t} + e_{2,t} \tag{21}$$

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

$$b_{1,OLS} = \Omega_{2,1}'\Omega_{1,1}^{-1}$$

$$\Sigma_{1,1} = \Omega_{1,1}^2 - \Omega_{1,2}\Omega_{1,2}'$$

$$\Sigma_{1,2} = \Omega_{1,1}\Omega_{2,1} - \Omega_{1,2}\Omega_{2,2}'$$

$$\Sigma_{2,2} = \Omega_{2,1}\Omega_{2,1}' - \Omega_{2,2}\Omega_{2,2}'$$

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 (20)-(21). 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.9 LONG RUN IDENTIFICATION

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

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

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 respectively

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

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

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.10   Forecast Error Variance Decomposition

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

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

$$= \varphi_h v_t + ... + \varphi_{t+h} v_1 + \overline{\Psi}_{t+h}$$

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 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:

$$E(e_{t+h} e'_{t+h}) = \varphi_0 E(v_{t+h} v'_{t+h}) \varphi'_0 + ... + \varphi_{h-1} E(v_{t+1} v'_{t+1}) \varphi'_{h-1}$$

$$= \varphi_0 E \left(\left(\sum_k \nu_t^k \right) \left(\sum_k \nu_t^k \right)' \right) \varphi'_0 + ... + \varphi_{h-1} E \left(\left(\sum_k \nu_t^k \right) \left(\sum_k \nu_t^k \right)' \right) \varphi'_{h-1}$$

$$= \sum_{\ell=0}^{h-1} \varphi_\ell E \left(\left(\sum_k \nu_t^k \right) \left(\sum_k \nu_t^k \right)' \right) \varphi'_\ell$$

$$= \sum_{\ell=0}^{h-1} \varphi_\ell E \left(\nu_t^1 \nu_t^{1'} \right) \varphi'_\ell + ... + \sum_{\ell=0}^{h-1} \varphi_\ell E \left(\nu_t^n \nu_t^{n'} \right) \varphi'_\ell$$

$$= \sum_{\ell=0}^{h-1} \varphi_\ell J_1 \varphi'_\ell + ... + \sum_{\ell=0}^{h-1} \varphi_\ell J_n \varphi'_\ell$$

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

A.11 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.12 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

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

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 et al. (2011))