Please read this file very carefully before you start using the code. This folder contains the following files:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) OTHER CODE:

B. TVP-FAVAR : Estimate a single TVP-FAVAR. This code is for DEMONSTRATION

only, and it should be used as a starting point in order to

understand how estimation works (before going to the multiple

model case using DMA)

2) FORECASTING CODE:

A. Competing FCIs.m : Forecasts from the 4 existing FCIs we have collected from

Federal Reserve Banks

B. DMA\_TVP\_FAVAR.m : Forecasts from Dynamic Model Averaging/Selection (DMA/DMS),

with relatively noninformative prior

C. DMA\_TVP\_FAVAR\_TS.m : Forecasts from Dynamic Model Averaging/Selection (DMA/DMS),

with training sample prior

(This code is used only in the sensitivity analysis in the

online appendix)

D. FAVAR\_PC\_DOZ.m : Forecasts from the homoskedastic FAVAR with principal

components and Doz et al. (2011)estimation of the factors

3) FULL-SAMPLE CODE:

A. DMA\_probabilities : Plots the time-varying DMA probabilities, expected number of

variables, and FCI implied by DMA

(Use this code to replicate Figures 4 & 5)

Additionally the folder “functions” contains useful functions that are being called during estimation (e.g. mlag2.m creates VAR lags, and Minn\_prior\_KOOP.m implements our Minnesota-type prior on the coefficients ).

The folder data contains - guess what. However, be careful as there are two datasets in there. The first is the one used in the article (xdata.dat, other\_FCIs.dat, ydata.dat). However, I also have a dataset with 81 financial variables (xdata\_all.dat) which is only called by the DEMONSTRATION code TVP-FAVAR. In this code I give you the option to load any of the two datasets to extract the FCI (in order to understand how the algorithm works). Names of the variables are in the .mat file xnames.mat.

HOW TO USE THE CODE:

In the beginning of each file I have a section called “USER INPUT”. Please feel free to experiment with it. The default settings are the ones used in the paper, e.g. nlag=4 is the number of lags in the FAVAR).

HOWEVER, the default values of the forgetting/decay factors (called l\_1, l\_2, l\_3, l\_4 in the code, but denoted as in the paper) correspond to the TVP-FAVAR model. In order to estimate the FAVAR and FA-TVP-VAR models (see paper) you need to change the values of the forgetting factors. Setting l\_3=1 (leave l\_1 = l\_2 = 0.96, and l\_4 = 0.99) gives you the FA-TVP-VAR, while setting l\_3 = l\_4 = 1 (leave l\_1 = l\_2 = 0.96) gives you the heteroskedastic FAVAR. You can also obtain the homoscedastic FAVAR by setting all forgetting factors to 1, but this is not a model used in the paper (since as we explain this has inferior forecasting performance).

I have set the code DMA\_probabilities.m in order to print figures shown in the paper, conditional on the model chosen (e.g. the default setting of the forgetting factors will give you the probabilities in the TVP-FAVAR). For the forecasting code things are semi-automatic, since I do not like to set-up MATLAB to calculate forecasting results and print LaTeX tables (more programming means more chances of error, hence, I prefer calculating the averages manually in Excel). In that respect, if you want e.g. the MSFEs these can be found at the end of the code in the array MSFE\_DMA (for the case of DMA, and similarly for other forecasts found in other files). In order to obtain the mean MSFE for all variables, simply use the mean( ) function in MATLAB:

squeeze(mean(MSFE\_DMA(1:end-1,:,1),1))' % for h=1 steps ahead

squeeze(mean(MSFE\_DMA(1:end-2,:,2),1))' % for h=2 steps ahead

squeeze(mean(MSFE\_DMA(1:end-3,:,3),1))' % for h=3 steps ahead

squeeze(mean(MSFE\_DMA(1:end-4,:,4),1))' % for h=4 steps ahead

HEALTH WARNINGS:

While a single TVP-FAVAR is trivial to estimate, you will soon realise that forecasting recursively with 219 = 524,288 models (as we do in DMA), is quite a task. You will need a very strong PC and lots of patience, or alternatively a cluster of servers and MATLAB’s Parallel Processing Toolbox (this is what I actually did, i.e. I was submitting PBS jobs remotely in my University’s central cluster).

Before you try running the DMA code in your PC, I would suggest you time how much it takes to estimate and recursively forecast with a single model. You can do that by using the original DMA\_TVP\_FAVAR.m code. In the USER INPUT there is the setting:

var\_no\_dma = 1;

which chooses which variables should NOT be included in DMA. The setting above takes the first variable (S&P500 – check varnames.mat for the names and order of all 20 variables) and always includes it in each model, thus leaving the code to do DMA in the remaining 219 models. If you set:

var\_no\_dma = 1:20;

then all 20 variables are included in each model, and 0 variables are included in DMA. Hence, this is equivalent to estimating the full model with no DMA. The choice:

var\_no\_dma = [1 3 5 9 12 15];

will always include variables (1,3,5,9,12,15) in each factor model, and ask the code to do DMA in the remaining 14 variables (thus 214 = 16384 models, which can still be cumbersome for old PCs).

PLEASE BE CAREFULL WHEN YOU RUN THE CODE, WE BEAR NO RESPONSIBILITY FOR HARDWARE DAMAGE THAT MIGHT OCCUR IN OLD PCS FROM CPU OVERLOAD… THIS CODE IS NOT IDEAL FOR COMPLETE NOVICES. HOWEVER, LESS EXPERIENCED MATLAB USERS AND/OR PHD STUDENTS SHOULD BE ABLE TO EASILY UNDERSTAND THE ATTACHED CODE IN COMBINATION WITH CAREFUL STUDY OF THE PAPER. WE DO NOT OFFER SUPPORT FOR THIS CODE.

DIMITRIS KOROBILIS,

UNIVERSITY OF GLASGOW

DIMITRIS.KOROBILIS@GLASGOW.AC.UK