

ReadMe File for Replication Package

”Learning from crises: A new class of time-varying parameter VARs with observable adaptation”

In-Sample Analysis: U.S. Monthly Data and Euro Area Quarterly Data

1 Purpose and Scope

This document provides instructions for replicating the **in-sample analysis** conducted for both the U.S. monthly dataset and the Euro Area quarterly dataset. The in-sample component of the replication package includes:

- estimation of three time-varying parameter models (AVP–VAR, TVP–VAR–EB, TVP–VAR–FB),
- estimation of univariate UCSV models for each macroeconomic variable,
- comparison of intercept dynamics across multivariate and univariate models,
- standardized and unstandardized intercept comparisons,
- in-sample residual analysis for all three VAR specifications.

All computations are performed using the scripts:

- `INSAMPLE_models.m` (U.S. application),
- `main_INSAMPLE_EA.m` (Euro Area application),
- `plot_UCSV_TVintercepts.m`,
- `plot_residuals.m`.

The outputs generated by these scripts form the basis of the figures shown in the Appendix of the paper.

2 Software and Requirements

The in-sample routines require only MATLAB. All computations have been tested using:

- MATLAB R2024b and R2025a,
- no proprietary MathWorks toolboxes.

Required folders are automatically added using:

```
addpath('functions')
addpath('data')
```

3 Data Loading and Transformation

3.1 Endogenous Variables

The U.S. in-sample script uses:

```
select = {INDPRO, PCEPI, FEDFUNDS}
```

Series are transformed according to:

```
tcode = (5, 5, 2),
```

corresponding to:

- INDPRO: log-difference,
- PCEPI: log-difference,
- FEDFUNDS: first difference.

The Euro Area script follows the exact same logic, with transformations defined in `main_INSAMPLE_EA.m`.

3.2 Drivers (Instruments)

The AVP-VAR uses a rich set of observable instruments:

```
selectZ = {USEPU, GPR, GECON, GZ, EBP, JLN12, ...}.
```

Drivers are typically used in levels:

```
tcodeZ = (1, 1, ..., 1),      standardZ = 1.
```

Data are loaded using:

```
[Y, Z, T, varnum, data, dataZ, dates, varnames] = ...
load_data_Instruments(select, selectZ, tcode, tcodeZ, standard, standardZ);
```

4 Model Inputs and MCMC Settings

All three in-sample models use the same specification:

- VAR lag length: $p = 1$,
- number of latent factors: $r = 1$,
- MCMC settings:

```
nsave = 95,000, nburn = 5,000, nthin = 5.
```

Stored posterior draws:

$$N_{\text{kept}} = \frac{\text{nsave}}{\text{nthin}}.$$

5 Models Estimated In-Sample

This section documents the **exact MATLAB functions** used to estimate each model, their key features, and their inputs and outputs.

Model 1: AVP–VAR (function APVAR.m)

```
[yfore_save, beta_save1, ~, ~, residuals_save1] = ...  
APVAR(Y(1:sample,:), Z(1:sample,:), h, p, r, nsave, nburn, nthin, 'SV');
```

Key features:

- AVP–VAR with innovations driven by observable instruments Z ,
- Gaussian-mixture stochastic volatility ('SV'),
- returns:
 - `beta_save1`: posterior draws of coefficients,
 - `residuals_save1`: time-varying residuals,
 - `yfore_save`: in-sample fitted values.

Model 2: TVP–VAR–EB (function TVP_RW_EB_residuals.m)

```
[residuals_save2, ~, beta_save2] = ...  
TVP_RW_EB_residuals(Y(1:sample,:), p, nsave, nburn, nthin);
```

Key features:

- Empirical-Bayes Primiceri-style TVP specification,
- stochastic volatility and random-walk states,
- aligned with pre-sample EB shrinkage,
- returns:
 - `beta_save2`: time-varying coefficients,
 - `residuals_save2`: TVP residuals.

Model 3: TVP–VAR–FB (function `TVP_VAR_FB.m`)

```
[yfore_save3, beta_save3, invalid_runs, ~, residuals_save3] = ...
TVP_VAR_FB(Y(1:sample,:), h, p, r, nsave, nburn, nthin);
```

Key features:

- full Bayesian TVP model with factor stochastic volatility,
- random-walk coefficients and loadings,
- horseshoe shrinkage,
- returns:
 - `beta_save3`,
 - `residuals_save3`.

Univariate Benchmark: UC-SV

For each variable $i = 1, 2, 3$, the script runs:

```
[y_fitted(:,i), y_forecast(:,i), ~, tauhat(:,i), residualsUCSV(:,i)] = ...
UCSV_gam_Nico(Y(1:sample,i), nsave, nburn, nthin, h);
```

Outputs:

- τ_t : trend component used as a benchmark for intercept comparisons,
- univariate SV residuals.

6 Saving Results

Each in-sample estimation script stores results as:

```
insample_results_p1.mat % U.S. monthly data  
insample_results_p2.mat % Euro Area data
```

The saved objects include:

- `beta_save1`, `beta_save2`, `beta_save3`,
- `residuals_save1`, `residuals_save2`, `residuals_save3`,
- `tauhat` (UCSV),
- `Y`, `Z`, `dates`, `T_full`.

7 Generating Intercept Comparison Figures

Intercept figures are produced using:

```
>> plot_UCSV_TVintercepts
```

This script:

- aligns TVP coefficients from all three models,
- standardizes them using `zscore`,
- produces AVP vs TVP–VAR vs UCSV and AVP vs TVP–VAR–EB vs UCSV,
- generates both standardized and unstandardized versions,
- shades recession episodes (GFC and COVID-19).

Generated figures:

```
Figures/intercepts_US_FB.pdf  
Figures/intercepts_US_EB.pdf  
Figures/intercepts_US_FB_unstand.pdf  
Figures/intercepts_US_EB_unstand.pdf
```

8 Generating Residual Comparison Figures

In-sample residual diagnostics are produced using:

```
>> plot_residuals
```

This script:

- loads in-sample results,
- computes cumulative absolute residuals:

$$\text{cumres}(t) = \sum_{s=1}^t |e_s|,$$

- plots AVP–VAR, TVP–VAR–FB, and TVP–VAR–EB,
- highlights crisis periods.

Output:

`Figures/residuals_US.pdf`

9 Summary of Script Workflow

Task	Script	Output
Estimate in-sample models (US)	<code>INSAMPLE_models.m</code>	<code>insample_results_p1.mat</code>
Estimate in-sample models (EA)	<code>main_INSAMPLE_EA.m</code>	<code>insample_results_p2.mat</code>
Intercept comparison figures	<code>plot_UCSV_TVintercepts.m</code>	4 PDF figures
Residual comparison figures	<code>plot_residuals.m</code>	<code>residuals_US.pdf</code>