# Summary

This technical note is a guide for the Threshold Vector Autoregressive (TVARX) MATLAB Toolkit.[[1]](#footnote-1) Section II presents a brief introduction to the class of model this toolkit can estimate. Section III introduced the functions from the toolkit and Section IV presents the utility functions needed by the toolkit but not documented.

N.B.: MATLAB Statistics Toolbox is required for *quantile.m*. This dependency will be removed in future release.

N.B.: This toolbox use codes available on internet from different sources. If you find your piece of code and find the copyright is not properly stated, write me an email and I will fix the situation. If you find your piece of code and want to be removed from the distribution of this toolbox, write me an email and I will remove the dependency to your code.

Contents

[I. Summary 1](#_Toc369009972)

[II. Models 1](#_Toc369009973)

[III. MATLAB Threshold Vector Autoregressive Toolkit 3](#_Toc369009974)

[IV. Included but not documented functions 8](#_Toc369009975)

[V. Bibliography 9](#_Toc369009976)

# Models

## Linear Vector Autoregressive (VARX) model

The structural form of the VAR to be estimated is

where is a vector of endogenous macroeconomic variables at time , is a matrix of contemporaneous coefficients, is a matrix of lag coefficients on , and . Finally, and are a vector of stochastic exogenous variables and a vector of deterministic exogenous variables at time , respectively, with and being their respective matrix of coefficients. Multiplying by yields:

The reduced form of the VAR that is actually estimated by ordinary least squares (OLS) is

where.

This leads to the well-known identification problem in VAR's, which comes from the fact that there are more free parameters in the structural form than there are estimated parameters in the reduced form. Therefore, the researcher must impose restrictions on the structural form to fully identify the structural-form parameters of interest. Perhaps the most popular form of restrictions is the Cholesky decomposition, which imposes a (contemporaneously) recursive structure on the model.

## Threshold Vector Autoregressive (TVARX) model

TVARX are typically applied to multivariate time series models as an extension of linear VARX, in order to allow for higher degree of flexibility in model parameters through a regime switching behaviour. It is a tool for understanding and predicting future values of time series in this model, assuming that the behaviour of the series changes once the series enters a different regime. The reduced form and structural parameters of a VAR may differ if the current or past value of some known threshold variable is higher than some value that is unknown, a priori. The system of equations to be estimated for a reduced-form VAR, in terms specified in the previous section, with one threshold is

where is an indicator function which equal one if the threshold variable at lag order (the delay parameter) is greater than the threshold value and zero otherwise. Equivalently, heteroscedasticity can be assumed across the two regimes[[2]](#footnote-2):

If is known, the model reduces to an OLS on two distinct samples. If is not known, it required to be determined endogenously by a joint estimation of optimal parameters by Grid Search. The model is not differentiable with respect to . For a given , we estimate the coefficients matrices and the is the one that maximize the likelihood function.

# MATLAB Threshold Vector Autoregressive Toolkit

Threshold Vector Autoregressive (TVARX) Functions[[3]](#footnote-3):

* tvarx: TVARX estimation
* tvarxbacktest: TVARX prediction and forecast accuracy
* tvarxcompmat: TVAR(1) companion form of TVARX model
* tvarxic: TVARX model selection
* tvarxlbckillian: TVARX linear (regime-specific) bias-correction
* tvarxlirf: TVARX linear (regime-specific) impulse response functions
* tvarxlirfplt: Plot linear (regime-specific) impulse response functions
* tvarxlogdet: TVARX log determinant
* tvarxloglik: TVARX total and conditional log likelihoods
* tvarxlsim: TVARX linear (regime-specific) sample path simulation
* tvarxnlsim: TVARX nonlinear sample paths simulation
* tvarxplt: Plot TVARX model actual against predicted values and residuals
* tvarxprt: Print TVARX model output
* tvarxresid: TVARX innovation arrays simulation
* tvarxstab: TVARX stability test
* varxvcov: VARX variance-covariance matrix

## Threshold Vector Autoregressive Estimation (tvarx.m)

Estimate a threshold vector autoregressive with (optionally) exogenous variables model by least squares. Unrestricted TVARX models are estimated by sequential conditional ordinary least squares (equivalent to maximum likelihood) and restricted TVARX models are estimated by sequential conditional generalized least squares (equivalent to restricted maximum likelihood).

Syntax

*results = tvarx(y,numLags,numThresh,transVar)*

*results = tvarx(y,numLags,numThresh,transVar,options)*

*results = tvarx(y,numLags,numThresh,transVar,struct('cons',true,'trend',true))*

Inputs (required and optional) and Output arguments

*help tvarx*

## Prediction and Forecast Accuracy (tvarxbacktest.m)

Determine the prediction and forecast accuracy of TVARX models. It assesses the quality and adequacy of the models in a three-step procedure. First, it divides the response data into three periods: presample, estimation and prediction or forecast. Second, it fits the models to the estimation data, using the presample periods to provide lagged data. Finally, it performs a Monte-Carlo simulation with a specified number of sample paths and compares the predicted or forecasted values of the fitted models to the observations using goodness-of-fit measures. The predicted values are in-sample and forecasted values are out-of-sample. Goodness-of-fit measures can be calculated as a numerical summary of the predictive performance. When choosing among competing models, one can look at their respective goodness-of-fit to compare predictive performance.

Syntax

*backtest = tvarxbacktest(results,ystart,yend,test,options)*

Inputs (required and optional) and Output arguments

*help tvarxbacktest*

## Companion Form (tvarxcompmat.m)

Construct TVAR(1) companion form of a TVARX(p) model. It is required to perform stability test.

Syntax

*CompMat = tvarxcompmat(results)*

Inputs (required and optional) and Output arguments

*help tvarxcompmat*

## Model Selection (tvarxic.m)

Evaluate information criteria to select the order, time delay and threshold number of TVARX models. Information criteria are likelihood-based measures of model fit that include a penalty for complexity (specifically, the number of parameters). It performs four tests: Akaike, Schwarz-Bayesian, Hannan-Quinn and Final Prediction Error.

Syntax

*[icVar,infocrit,tresh] = tvarxic(y,nARmax,nThreshmax,transVar,options\_ic,options\_tvarx)*

Inputs (required and optional) and Output arguments

*help tvarxic*

## Linear (Regime-Specific) Bias-Correction (tvarxlbckillian.m)

Apply Kilian (1998) bootstrapped-based bias-correction for linear (regime-specific) impulse response functions of TVARX models.

Syntax

*[bcresults,biasConstantfinal,biasTrendfinal,biasExofinal,biasARfinal] = tvarxlbckillian(results,options)*

Inputs (required and optional) and Output arguments

*help tvarxlbckillian*

## Linear (Regime-Specific) Impulse Response Functions (tvarxlirf.m)

Compute linear (regime-specific) impulse response functions and (optionally) confidence intervals of TVARX models.

Syntax

*[IRF,bootIRFLowerCI,bootIRFUpperCI,bootIRFMedCI,bootIRFAll,impact] = tvarxlirf(results,options)*

Inputs (required and optional) and Output arguments

*help tvarxlirf*

## Plot Linear (Regime-Specific) Impulse Response Functions (tvarxlirfplt.m)

Plot linear (regime-specific) impulse response functions and (optionally) confidence intervals computed with TVARXLIRF.

Syntax

*tvarxlirfplt(results,IRF,bootIRFLowerCI,bootIRFUpperCI,vnames,variables,shocks,filename,fontsize,suptit)*

Inputs (required and optional)

*help tvarxlirfplt*

## Log Determinant (tvarxlogdet.m)

Compute log determinant of TVARX models.

Syntax

*logDet = tvarxlogdet(resid,commonvar)*

Inputs (required and optional) and Output arguments

*help tvarxlogdet*

## Total and Conditional Log Likelihoods (tvarxloglik.m)

Compute total and conditional log likelihoods of TVARX models.

Syntax

*[logL,logCL] = tvarxloglik (resid,commonvar)*

Inputs (required and optional) and Output arguments

*help tvarxloglik*

## Linear (Regime Specific) Sample Path Simulation (tvarxlsim.m)

Simulate linear (regime specific) sample paths of TVARX models for given innovation arrays.

Syntax

*Y = tvarxlsim(results,W,Y0,X)*

Inputs (required and optional) and Output arguments

*help tvarxlsim*

## Non-linear Sample Path Simulation (tvarxnlsim.m)

Simulate non-linear sample paths of TVARX models for given innovation arrays.

Syntax

*[Y,checkRegimes] = tvarxnlsim(results,W,transVar,Y0,X)*

Inputs (required and optional) and Output arguments

*help tvarxnlsim*

## Plot Model Actual against Predicted Values and Residuals (tvarxplt.m)

Plot TVARX models actual against predicted values and residuals.

Syntax

*tvarxplt(results,vnames)*

Inputs (required and optional)

*help tvarxplt*

## Print Model Output (tvarxprt.m)

Print TVARX models output.

Syntax

*tvarxprt(results,vnames,fid)*

Inputs (required and optional)

*help tvarxprt*

## Innovation Arrays Simulation (tvarxresid.m)

Create innovation arrays for simulation or impulse responses functions by Monte Carlo or bootstrap method.

Syntax

*W = tvarxresid(results,options)*

Inputs (required and optional) and Output arguments

*help tvarxresid*

## Stability Test (tvarxstab.m)

Perform TVARX model stability test. It checks if all eigenvalues of the companion form are smaller than one in modulus.

Syntax

*[isStable,eigARmax] = tvarxstab(results)*

Inputs (required and optional) and Output arguments

*help tvarxstab*

## Variance-Covariance Matrix (varxvcov.m)

Estimate the variance-covariance matrix of TVARX model estimated parameters under one of these conditions:

* Conditionally homoskedastic and uncorrelated errors
* Conditionally homoskedastic but correlated errors
* Heteroskedastic but conditionally uncorrelated errors
* Heteroskedastic and correlated errors

Syntax

*vcov = varxvcov(x,resid,het,corre)*

Inputs (required and optional) and Output arguments

*help varxvcov*

# Included but not documented functions

The toolkit comes with some functions that are used to support other functions. Please consult the help contained within the function for more details[[4]](#footnote-4).

Vector Autoregressive (VARX) Functions

* varx: VARX estimation
* varxcompmat: VAR(1) companion form of a VARX model
* varxloglik: VARX total and conditional log likelihoods
* varxsim: VARX sample path simulation
* varxstab: VARX stability test

Statistics Functions

* bootblock: Circular block bootstrap
* bootcont: Continuous-path block bootstrap
* bootiid: *IID* bootstrap
* bootoptblock: Optimal block length for the stationary or circular block bootstrap
* bootstationary: Stationary bootstrap
* invprctile: Inverse percentiles
* mvlinreg: Multivariate multiple linear regression
* mvolsfit: Multivariate multiple linear regression using OLS
* mvrlsfit: Multivariate multiple linear regression using restricted GLS

Plot and Print Functions

* FigFont: Font size and style of a chart’s font
* maxfig: Maximize a figure window to fill the entire screen
* mprint: Print a matrix in formatted form
* plt: Plots results structures
* prt: Prints results structures
* rgb: Translate a colour from multiple formats into MATLAB format
* subtightplot: Subplot function with additional functionality
* suptitle: Puts a title above all subplots
* tsdate: Produce a time-series date string
* tsplot: Time-series plot with dates and labels
* tsprint: Print time-series matrix or vector with dates and column labels
* export\_fig: Export figures suitable for publication

Utilities Functions

* cal: Time-series calendar
* demean: Demean a vector or matrix
* getOptions: Returns options values in the workspace
* gfit: Goodness-of-fit measures
* ical: Finds observation number given a cal() structure
* ind2subv: Subscript vector from linear index
* infocriterion: Information criteria for model selection
* lagmat: Create matrix of lagged time series
* mdmax: Find the largest component in a multidimensional array
* mdmin: Find the smallest component in a multidimensional array
* vec: VEC operator

# Bibliography

Kilian, Lutz. "Small-Sample Confidence Intervals For Impulse Response Functions." *The Review of Economics and Statistics*, 1998: 218-230.

1. This toolbox was developed with MATLAB R2012a. [↑](#footnote-ref-1)
2. If the variance-covariance matrix of the overall residuals is used to draw shocks (either from the reduced form errors or from structural errors), homoscedasticity is assumed. This is however a huge assumption in the case of regime-switching models and complicates the identification of the structural interrelation of shocks in multivariate systems. A threshold model would - more or less by construction - imply that errors and thus the contemporaneous relations of shocks behave differently in the two regimes. It is thus possible to estimate the contemporaneous relations for each regime separately. [↑](#footnote-ref-2)
3. The internal Help Guides of each code will guide the user to run the routines. [↑](#footnote-ref-3)
4. Help for functions within MATLAB, type “help *name*” in the command window [↑](#footnote-ref-4)