Package: SVARtca (via r-universe)

May 28, 2026

Type Package

Title Transmission Channel Analysis in Structural VAR Models

Version 1.0.2

Date 2026-03-15

Description Implements Transmission Channel Analysis (TCA) for structural vector autoregressive (SVAR) models following the methodology of Wegner, Lieb, and Smeekes (2025) <doi:10.48550/arXiv.2405.18987>. TCA decomposes impulse response functions (IRFs) into contributions from distinct transmission channels using a systems form representation and directed acyclic graph (DAG) path analysis. Supports overlapping channels, exhaustive 3-way and 4-way decompositions via inclusion-exclusion principle. This is a parallel R implementation of the 'tca-matlab-toolbox' (<https://github.com/enweg/tca-matlab-toolbox>).

License MIT + file LICENSE

URL https://github.com/muhammedalkhalaf/SVARtca

BugReports https://github.com/muhammedalkhalaf/SVARtca/issues

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports Matrix, ggplot2, rlang

Suggests testthat (>= 3.0.0), knitr, rmarkdown, vars

Config/testthat/edition 3

Repository https://muhammedalkhalaf.r-universe.dev

Date/Publication 2026-04-27 20:56:52 UTC

RemoteUrl https://github.com/muhammedalkhalaf/svartca

RemoteRef HEAD

RemoteSha fefdff7db21c0abee996756d0dcdff46c06f01d7

Contents

`SVARtca-package` *Transmission Channel Analysis in Structural VAR Models*

Description

The SVARtca package implements Transmission Channel Analysis (TCA) for decomposing impulse response functions in structural vector autoregressive (SVAR) models. The package follows the methodology presented in Wegner, Lieb, Smeekes (2025).

TCA enables researchers to decompose the total impulse response to a shock into contributions from different transmission channels. Supported channels include effects passing through specific intermediate variables, allowing analysts to distinguish direct effects from indirect effects that operate through identified transmission mechanisms.

Key features:

- Multiple decomposition modes: overlapping, exhaustive 3-way, and exhaustive 4-way.
- Support for VAR and SVAR models via integration with the **vars** package.
- Visualization of channel contributions using **ggplot2**.
- Diagnostic tools for validating decomposition accuracy.

Main Functions

`tca_systems_form` Build the systems form representation (B and Omega matrices) from VAR coefficients and structural impact matrix.

`tca_analyze` Run the main TCA analysis with specified decomposition mode.

`tca_decompose_binary` Decompose IRF into "through" and "not_through" components for a single variable.

`tca_validate_additivity` Validate that the binary decomposition satisfies exact additivity.

`tca_from_var` Convenience wrapper to run TCA directly from a fitted VAR model object.

`plot_tca` Create publication-quality visualizations of channel contributions.

Identification

The package supports two identification schemes:

- **Cholesky**: Lower-triangular Cholesky decomposition of the residual covariance matrix. Set `Phi0 = t(chol(Sigma))`.
- **Manual**: User-supplied structural impact matrix for alternative identifications (e.g., sign restrictions, zero restrictions).

Citation

If you use the SVARtca package in your research, please cite the original methodology paper:

Wegner, E., Lieb, L., Smeekes, S. (2025). *Transmission Channel Analysis in Dynamic Models.* arXiv:2405.18987. https://github.com/enweg/tca-matlab-toolbox

Author(s)

Muhammad Alkhalaf

References

Wegner, E., Lieb, L., Smeekes, S. (2025). Transmission Channel Analysis in Dynamic Models. arXiv:2405.18987.

---

plot_tca — *Plot TCA Channel Decomposition*

Description

Creates a stacked bar chart (for exhaustive modes) or a line chart (for overlapping mode) showing channel contributions over horizons.

Usage

```
plot_tca(x, target = NULL, type = NULL, title = NULL,
  colors = NULL)
```

Arguments

`x`	A `tca_result` object from `tca_analyze`.
`target`	Integer index of the response variable to plot (default: first variable after shock).
`type`	Plot type: `"bar"` for stacked bar chart (default for exhaustive modes), `"line"` for overlaid lines (default for overlapping mode). If `NULL`, chosen automatically.
`title`	Custom plot title (optional).
`colors`	Named character vector of colours for channels (optional).

Value

A ggplot object.

Examples

```
K <- 4
A1 <- matrix(c(0.7,-0.1,0.05,-0.05, -0.3,0.6,0.10,-0.10,
               -0.2,0.1,0.70,0.05, -0.1,0.2,0.05,0.65), K, K, byrow=TRUE)
Sigma <- matrix(c(1,0.3,0.2,0.1, 0.3,1.5,0.25,0.15,
                  0.2,0.25,0.8,0.1, 0.1,0.15,0.1,0.6), K, K, byrow=TRUE)
Phi0 <- t(chol(Sigma))
sf <- tca_systems_form(Phi0, list(A1), h = 20)
res <- tca_analyze(from = 1, B = sf$B, Omega = sf$Omega,
                   intermediates = c(2, 4), K = K, h = 20,
                   order = 1:K, mode = "exhaustive_4way",
                   var_names = c("IntRate","GDP","Inflation","Wages"))
plot_tca(res, target = 3)
```

---

print.tca_result *Print TCA Result*

Description

S3 print method for tca_result objects. Displays a formatted table of transmission channel contributions across horizons.

Usage

```
## S3 method for class 'tca_result'
print(x, target = NULL, ...)
```

Arguments

x	A tca_result object.
target	Target variable to display (default: first intermediate).
...	Additional arguments (ignored).

Value

Invisibly returns x.

tca_analyze *Transmission Channel Analysis*

Description

Decomposes impulse response functions into transmission channel contributions using the methodology of Wegner, Lieb, Smeekes (2025).

Three decomposition modes are supported:

`"overlapping"` Each channel is through(j) = total - not_through(j). Channels may overlap, so their sum may differ from the total.

`"exhaustive_3way"` (2 intermediates only) Non-overlapping: (1) through var1 inclusive, (2) through var2 only, (3) direct. Sum equals total.

`"exhaustive_4way"` (2 intermediates only) Full inclusion-exclusion: (1) var1 only, (2) var2 only, (3) both, (4) direct. Sum equals total.

Usage

```
tca_analyze(from, B, Omega, intermediates, K, h, order,
  mode = "overlapping", var_names = NULL)
```

Arguments

`from`	Shock variable number (1-based).
`B`	Systems form B matrix (from `tca_systems_form`).
`Omega`	Systems form Omega matrix.
`intermediates`	Integer vector of intermediate variable numbers (1-based, original ordering).
`K`	Number of variables.
`h`	Maximum horizon.
`order`	Transmission ordering vector.
`mode`	Decomposition mode: `"overlapping"`, `"exhaustive_3way"`, or `"exhaustive_4way"`.
`var_names`	Character vector of variable names (optional).

Value

A list of class `"tca_result"` with components:

irf_total Matrix (h+1) x K of total IRFs.

irf_channels Named list of channel IRF matrices, each (h+1) x K.

channel_names Character vector of channel names.

mode Decomposition mode used.

from Shock variable number.

K Number of variables.

h Maximum horizon.

order Transmission ordering.

var_names Variable names.

References

Wegner, E., Lieb, L., Smeekes, S. (2025). *Transmission Channel Analysis in Dynamic Models.* arXiv:2405.18987.

Examples

```
# Monetary policy model
K <- 4
A1 <- matrix(c(0.7,-0.1,0.05,-0.05, -0.3,0.6,0.10,-0.10,
               -0.2,0.1,0.70,0.05, -0.1,0.2,0.05,0.65), K, K, byrow=TRUE)
Sigma <- matrix(c(1,0.3,0.2,0.1, 0.3,1.5,0.25,0.15,
                  0.2,0.25,0.8,0.1, 0.1,0.15,0.1,0.6), K, K, byrow=TRUE)
Phi0 <- t(chol(Sigma))
sf <- tca_systems_form(Phi0, list(A1), h = 20)
result <- tca_analyze(from = 1, B = sf$B, Omega = sf$Omega,
                      intermediates = c(2, 4), K = K, h = 20,
                      order = 1:K, mode = "exhaustive_4way",
                      var_names = c("IntRate","GDP","Inflation","Wages"))
print(result)
```

`tca_decompose_binary` *Binary Decomposition: Total = Through + Not-Through*

Description

Decomposes the total IRF into the effect passing through a variable and the effect not passing through it. This is an exact decomposition (residual is zero at machine precision).

Usage

```
tca_decompose_binary(from, B, Omega, var_idx, K, h, order)
```

Arguments

`from`	Shock variable (1-based).
`B`	Systems form B matrix.
`Omega`	Systems form Omega matrix.
`var_idx`	Variable to decompose through (1-based).
`K`	Number of variables.
`h`	Maximum horizon.
`order`	Transmission ordering.

Value

A list with matrices `total`, `through`, `not_through` (each (h+1) x K).

tca_from_var — *Run TCA from a VAR Estimation Object*

Description

Extracts coefficient matrices and residual covariance from a fitted VAR model (from the **vars** package) and runs TCA.

Usage

```
tca_from_var(var_model, from, intermediates, h = 20,
  order = NULL, mode = "overlapping",
  identification = "cholesky", Phi0 = NULL)
```

Arguments

`var_model`	A fitted VAR model object from `VAR`.
`from`	Shock variable (integer or name).
`intermediates`	Integer vector or character vector of intermediate variable names.
`h`	Maximum horizon (default: 20).
`order`	Transmission ordering (default: variable ordering in the VAR).
`mode`	Decomposition mode: `"overlapping"`, `"exhaustive_3way"`, or `"exhaustive_4way"`.
`identification`	Identification scheme: `"cholesky"` (default) or `"manual"`.
`Phi0`	Manual impact matrix. Required if `identification = "manual"`.

Value

A `tca_result` object (see `tca_analyze`).

Examples

```
library(vars)
data(Canada)
var_est <- VAR(Canada, p = 2, type = "const")
result <- tca_from_var(var_est, from = "e",
                        intermediates = c("prod", "rw"),
                        h = 20, mode = "exhaustive_4way")
plot_tca(result, target = "U")
```

tca_systems_form *Build Complete Systems Form*

Description

Constructs both B and Omega matrices for the systems form representation x = Bx + Omega*epsilon, following the methodology of Wegner, Lieb, Smeekes (2025).

Usage

```
tca_systems_form(Phi0, As, h, order = NULL, Psis = NULL)
```

Arguments

`Phi0`	Structural impact matrix (K x K). For Cholesky identification, use `t(chol(Sigma))`.
`As`	List of VAR coefficient matrices (A_1, A_2, ..., A_p).
`h`	Maximum IRF horizon.
`order`	Transmission ordering vector (default: `1:K`).
`Psis`	List of reduced-form MA coefficients (optional, for DSGE/VARMA models).

Value

A list with components:

B Systems form B matrix (K*(h+1) x K*(h+1)).

Omega Systems form Omega matrix (K*(h+1) x K*(h+1)).

References

Wegner, E., Lieb, L., Smeekes, S. (2025). *Transmission Channel Analysis in Dynamic Models.* arXiv:2405.18987. https://github.com/enweg/tca-matlab-toolbox

Examples

```
# 2-variable VAR(1)
Phi0 <- matrix(c(1, 0.3, 0, 0.95), 2, 2)
As <- list(matrix(c(0.5, -0.1, 0.2, 0.4), 2, 2))
sf <- tca_systems_form(Phi0, As, h = 10)
dim(sf$B)  # 22 x 22
```

tca_validate_additivity *Validate Binary Additivity*

Description

Tests that total = through(j) + not_through(j) holds for all variables at machine precision. This is a diagnostic to verify correct implementation.

Usage

```
tca_validate_additivity(from, B, Omega, K, h, order,
  var_names = NULL, verbose = TRUE)
```

Arguments

`from`	Shock variable (1-based).
`B`	Systems form B matrix.
`Omega`	Systems form Omega matrix.
`K`	Number of variables.
`h`	Maximum horizon.
`order`	Transmission ordering.
`var_names`	Character vector of variable names (optional).
`verbose`	Logical; print results? Default `TRUE`.

Value

Invisibly returns `TRUE` if all tests pass.

Index