Package: kwb.resilience (via r-universe)

September 10, 2024

Title R Package for the Quantification of Technical Resilience

Version 0.1.0

Description kwb.resilience allows quantification of a number of resilience indicators. Calculation requires a time series of performance values of a technical system, as well as values for acceptable and worst case performance.

License MIT + file LICENSE

URL https://github.com/KWB-R/kwb.resilience

BugReports https://github.com/KWB-R/kwb.resilience/issues

Depends R (>= 3.0)

Imports kwb.event

Suggests covr, testthat, knitr, rmarkdown, kwb.utils

VignetteBuilder knitr

Remotes github::KWB-R/kwb.event, github::KWB-R/kwb.utils

ByteCompile true

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

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

RemoteUrl https://github.com/KWB-R/kwb.resilience

RemoteRef HEAD

RemoteSha 24c604bc3c3a9e94a2acab6114946836d8946217

Contents

oxygen *Simulated Dissolved Oxygen in River Spree*

Description

Data series "oxygen" are included in kwb.resilience as test data for the use of the package. The included data is referred to in the supplied package tutorial (see vignettes), as well as in the supporting research paper
The data are simulated concentrations of dissolved oxygen (DO) in mg/l for different management scenarios. Simulations and assumptions are described in detail in Riechel et al. 2016.

Usage

```
data(oxygen)
```

Format

A data frame with 23425 rows and 5 variables

Details

Columns in data.frame:

- timestamp (POSIXct).
- S2_storage_2020 (numeric).
- S3_storage_increase (numeric).
- S4_red_Imp_Surface (numeric).
- S5_increase_in_DO (numeric).

resilience.events *Calculate Resilience by Failure Event*

Description

Calculates resilience indices (see Matzinger et al. 2018) for each failure event in time series of performance P(t). Failure is defined by acceptable performance Pa and maximal failure Pmax.

Usage

```
resilience.events(time_stamp, Pt, Pa, Pmax, evtSepTime, signalWidth)
```

Arguments

`time_stamp`	vector containing timestamp (sorted in ascending order)
`Pt`	vector with performance P(t) (same length as timestamp)
`Pa`	accpetable performance
`Pmax`	maximal failure
`evtSepTime`	"event separation time" in seconds. Maximal allowed time difference between two consecutive timestamps within the same event.
`signalWidth`	"signal width" in seconds. Length of time interval that one timestamp is representing, e.g. 5*60 = 300 if each timestamp respresents a time interval of five minutes (as e.g. a time series is recorded on a five minute time scale). This parameter is needed to calculate event durations.

Value

Returns data.frame containing one row by failure event. First columns are identical to kwb.event::hsEvents. Following columns are additional resilience indices:

- Sev: severity by event
- Res0: resilience index by event
- trec: recovery time in seconds
- trec_percent: trec relative to event duration in per cent
- worst_P: P(t) closest to Pmax within event

script is stopped if no failure event with message "Pa never exceeded"

`resilience.severity` *Calculate Severity*

Description

calculates severity Sev (see Matzinger et al. 2018) of failures for time series of performance P(t). Entire time period is used, failure is defined by acceptable performance Pa and maximal failure Pmax.

Usage

```
resilience.severity(time_stamp, Pt, Pa, Pmax, integral_method = 2)
```

Arguments

time_stamp	vector containing timestamp (sorted in ascending order)
Pt	vector with performance P(t) (same length as timestamp)
Pa	accpetable performance
Pmax	maximal failure (worst case)
integral_method	either 1 or 2. Switches between two different versions of integral calculation. Both methods should return the same but method 2 should be much faster when applied to long vectors. Default is method 2.

Value

Returns severity integrated over entire time series (one number)

---

resilience.summary *Calculate Overall Resilience Index for One or Several Performance Time Series*

Description

Calculates resilience indices (see Matzinger et al. 2018) for entire time series of performance P(t). Failure is defined by acceptable performance Pa and maximal failure Pmax. Entire time series is considered

Usage

```
resilience.summary(time_stamp, Pt, Pa, Pmax, evtSepTime, signalWidth)
```

Arguments

time_stamp	vector containing timestamp (sorted in ascending order)
Pt	vector or data.frame (if several colums) with performance P(t) (same length as timestamp)
Pa	accpetable performance
Pmax	maximal failure
evtSepTime	"event separation time" in seconds. Maximal allowed time difference between two consecutive timestamps within the same event.
signalWidth	"signal width" in seconds. Length of time interval that one timestamp is representing, e.g. 5*60 = 300 if each timestamp respresents a time interval of five minutes (as e.g. a time series is recorded on a five minute time scale). This parameter is needed to calculate event durations.

Value

Returns data.frame containing one row by time series. Columns are:

- num_events: number of failure events in time series
- worst_P: P(t) closest to Pmax within time series
- total_dur: total duration of failure events in seconds
- total_trec: total recovery time of failure events in seconds
- mean_trec_percent: trec relative to event duration in per cent, averaged over all failure events in time series
- Sev: severity over entire time series (=0 if no exceedance of Pa)
- Res0: resilience index over entire time series (=1 if no exceedance of Pa)

Index