iTesla

Innovative Tools for Electrical System Security within Large Area

Grant agreement number	283012	**Funding scheme**	Collaborative projects
Start date	01.01.2012	**Duration**	48 months
Call identifier	FP7-ENERGY-2011-1		

Deliverable D4.3 (ANNEX)
Modifications to the Impact Assessment Matlab Functions

Dissemination level		
PU	Public.	
TSO	Restricted to consortium members and TSO members of ENTSO-E (including the Commission Services).	
RE	Restricted to a group specified by the consortium (including the Commission services).	
CO	Confidential, only for members of the consortium (including the Commission services).	**X**

Document Name :	D4.3_Annex_v.1.1
Work Package:	WP4
Task	4.3
Deliverable	D4.3
Responsible Partner:	KTH

Author		Approval	
Name	Visa	Name	Date
[KTH] Dr. Rafael Segundo Dr. Luigi Vanfretti			

DIFFUSION LIST	
For action	For information
RTE Quinary	Confidential, only for members of the consortium (including the commission services)

DOCUMENT HISTORY

Index	Date	Author(s)	Main modifications
1.0	June 11th 2014	KTH	Initial Draft, modifications to the small-signal stability index
1.1	June 25th 2014	KTH	Incorporation of modification to the static and transient stability indexes

Table of Contents

1 Introduction

The present document presents in detail, all the changes and modifications applied to the static and most of the dynamic impact assessment indexes developed for iTesla, which are described on [1]. The changes where based on the feedback received on the integration stage of the original functions and the issues raised by the different partners.

2 Improvements to the Static Indexes: Overload and Over\Under Voltage

Some simples but relevant modifications where applied to both static functions, which in terms of methodology are similar. A summary of the main changes are described below:

(i) Less input parameters are required.

(ii) The functions automatically select the initial and final values of the signals, information useful to compute the index.

(iii) The function assumes uniform weights in all lines or buses.

(iv) The index is based only on average values.

(v) The functions neglects lines or buses which are out of service, signals with post-fault average values equal to zero.

(vi) Since the output of both functions will be used in the decision tree, no plotting options are available.

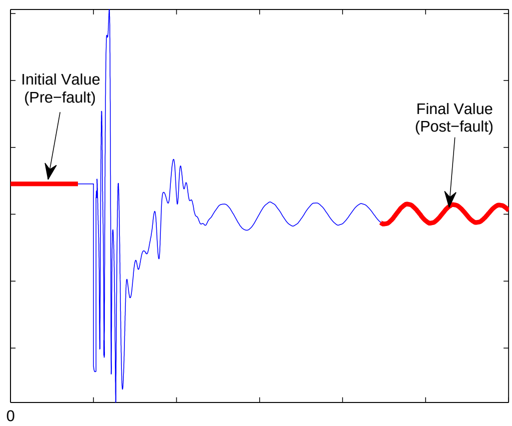

Figure 1: Pre-fault and post-fault sections automatically selected for the function.

The fundamental part of both functions (methodology) has not been changed and the functions provide the same outputs. The procedure to extract the information required to calculate the index has been changed.

Regarding point (ii), previously, the user had to provide a three element vector t_1:

$$t_1 \quad = \quad [t_{init}\,,\; t_{end}\,,\; \Delta T] \tag{1}$$

where t_{init} was the initial time of the simulation, t_{end} was the final time and Δt was the desired time steps. Since most of the time, the user does not know a priory these information, the function was simplified removing these input paraments and automatically selecting them directly from the time series. Figure 1 depicts in red the pre-fault and post-fault sections that are automatically selected for each signal. The mean values of each red section are then used to compute the overload and over\under voltage indexes.

Since uniform weighting factors are assumed, the unitary matrices wf_i and wv_i have been removed from the input parameters. With all the modifications described before, the function has been drastically simplified and the final version is shown on section A.1.

3 Improvements to the Transient Stability Index ISGA

The transient stability (TS) index, for its simplicity, it is considered the most robust index among the dynamic indexes. For this reason, minor modification has been required to apply respect to the original version described on [1].

Similar to the static indexes, the TS function required manual input of the initial t_{init} and final t_{end} time. This feature has have been removed from the function and now these two parameters are automatically assigned from the time-series, similar as in Figure 1.

$$t_1 = [t_{init}, t_{end}] \tag{2}$$

A positive aspect about the modification is that the function has been simplified and the user does not require to know any information a priori about the time-series, on the other hand, a possible drawback is that in a multy contingency case, the user cannot compute the TS index over a desired interval of time and the index will always calculate the integral over the full time period of the series.

Since the input paraments to this function are generator angles, the function neglects machines that have been shut down, which means, machines whose angles tend to minus infinity following the disconnection.

4 Improvements to the Small-Signal Stability function

Due to some challenges faced in the integration of the small-signal stability function into the iTesla platform, described in [1], major modifications have been applied to the original function and the most important changes are summarized below:

- The function assumes that the time-series used were obtained using variable time step.
- If the input signals lacks of significant oscillation (flat lines), the function will provide empty matrices.
- The actual small-signal analysis is applied to a section of the signals automatically selected. No manual input of window time is required.
- Input signals are analyzed based on variance $\sigma^2(t)$, which means that flat lines with $\sigma^2(t) = 0$ are discarded.
- Prony approach is the only ringdown method available to estimate modes on input signals.
- The output of the function is the three layer small-signal-stability index SMI, AMI and GMI as before.

4.1 Automatic contingency detection

The modifications to the SSS function are described in this section. In the previous function [1], three input parameters related to time events were requited a priori:

$$t_{input} = [t_{init}, t_{end}, t_{small}] \tag{3}$$

where t_{init} and t_{end} were the initial and final time in seconds of the signal to analyze and t_{small} defined the initial time, within the selected time frame, where the ringdown method was applied. Some of the drawbacks of defining these parameters a priori were:

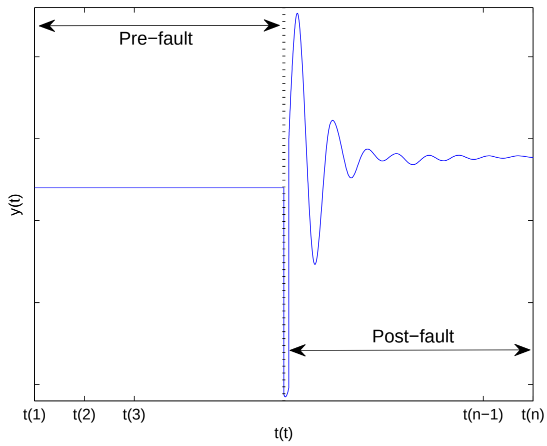

Figure 2: Typical input signal to analyze

- Pre-define a section with not enough oscillations or not oscillation at all.
- Select an initial time or ending time (t_{init} or t_{end}) which did not exist in the time vector, causing an error in the function.
- Lack of signal filter when signals had not oscillation.

In the new function the contingency or oscillations are automatically detected from the time vector so the ringdown method is applied automatically only to the oscillatory section of the signals. The following part of this section, illustrates how the new function operates.

Consider the signal depicted on Figure 2, where the time vector $t \in \Re^{n\times 1}$, the actual signal $y \in \Re^{n\times 1}$ and the distance between sampling time $\Delta t \in \Re^{n\times 1}$ are described as

$$\begin{aligned} y &= \begin{bmatrix} y(1) & y(2) & y(3) & \cdots & y(n-1) & y(n) \end{bmatrix}^T \\ t &= \begin{bmatrix} t(1) & t(2) & t(3) & \cdots & t(n-1) & t(n) \end{bmatrix}^T \end{aligned} \tag{4}$$

$$\begin{aligned} \Delta t &= \begin{bmatrix} t(1)-t(0), & t(2)-t(1), & t(3)-t(2), & \cdots, & t(n)-t(n-1) \end{bmatrix}^T \\ \Delta t &= \begin{bmatrix} \Delta t(1) & \Delta t(2) & \Delta t(3) & \cdots & \Delta t(n) \end{bmatrix}^T \end{aligned} \tag{5}$$

The assumption is that the elements of t are not equally spaced and as consequence, the elements of Δt in (5) are different to each other. Figure 3 describes the distribution in the elements of t and Δt in a typical time-series.

In a variable time-step simulation, Δt_i becomes very small following a disturbance in the system. On the other hand, Δt_i increase if the system is or has returned to steady state. Regardless the number of signals in the time-series, there is only one time vector of variable step. The idea is to exploit the information provided by the time vector to detect the precise moment that an event has occurred:

Built $t_k \in \Re^{n\times 1}$ using t as follows

$$t_k = \begin{bmatrix} 0, & t(1), & t(2), & \cdots, & t(n-2), & t(n-1) \end{bmatrix}^T \tag{6}$$

Subtract $t - t_k$ to obtain $t_m \in \Re^{n\times 1}$

$$\begin{aligned} t_m &= \begin{bmatrix} t_m(1) & t_m(2) & t_m(3) & \cdots & t_m(n-1) & t_m(n) \end{bmatrix}^T \\ t_m &= \begin{bmatrix} 0, & t(2)-t(1), & t(3)-t(2), & \cdots, & t(n)-t(n-1) \end{bmatrix}^T \end{aligned} \tag{7}$$

and note that although $t_m(1) = t(1) - 0$, the first element of t_m has been replaced by 0. Next, define t_r as a vector with the number of elements in t_m

$$t_r = \begin{bmatrix} 1 & 2 & 3 & \cdots & n \end{bmatrix}^T \tag{8}$$

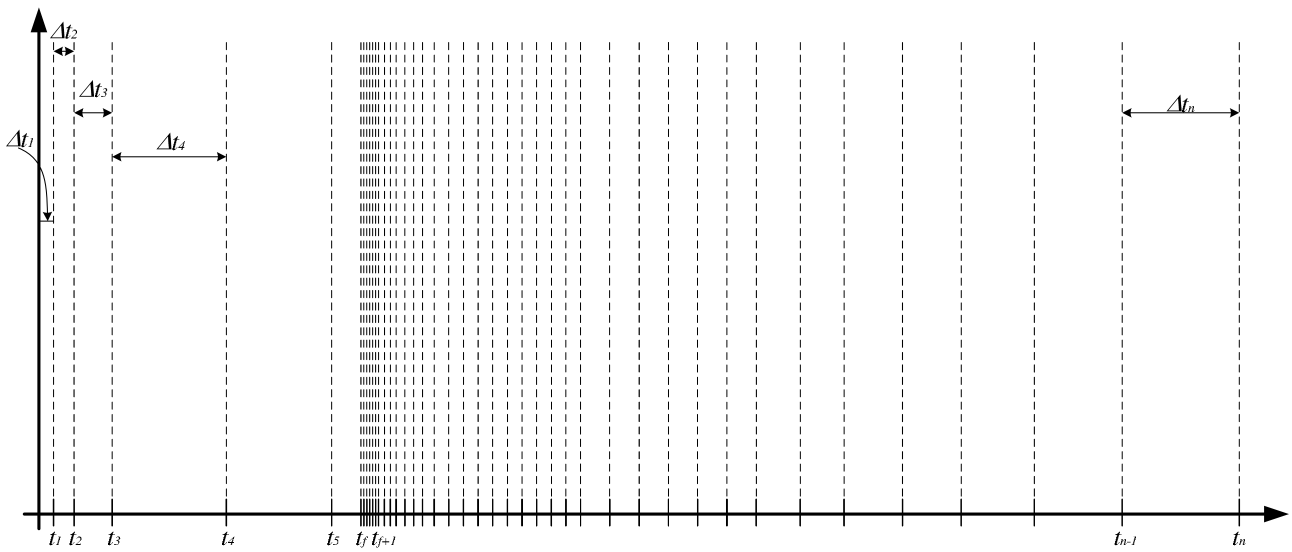

Figure 3: Variable time-step vector

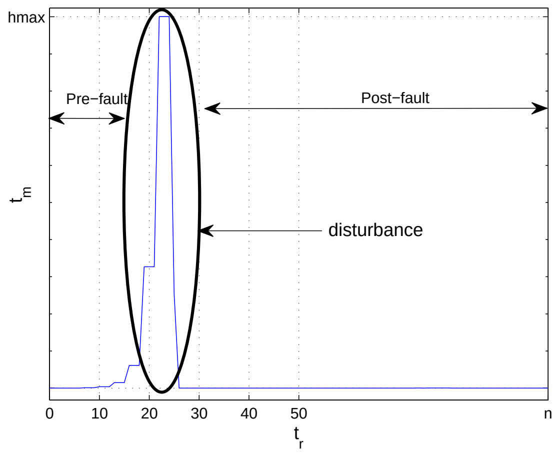

Figure 4: Disturbance detection from variation in the time-step, where h_{max} is the greatest element of t_m

Figure 4 depicts the plot of t_r vs t_m, it can be observed a pick in the plot that corresponds to a variation of the time step in the time vector indicating the origin of an event in the time series.

After identifying the disturbance in the time vector, the next step is to keep the section identified as post-fault as shown in Figure 5. The post-fault signal $y_f(t) \in \Re^{k\times 1}$ and post-fault time $t_f(t) \in \Re^{k\times 1}$ vectors are redefined in (9), note that $k < n$, the post-fault vectors are of smaller dimension.

$$\begin{aligned} y_f &= \begin{bmatrix} y_f(1) & y_f(2) & y_f(3) & \cdots & y_f(k-1) & y_f(k) \end{bmatrix}^T \\ t_f &= \begin{bmatrix} t_f(1) & t_f(2) & t_f(3) & \cdots & t_f(k-1) & t_f(k) \end{bmatrix}^T. \end{aligned} \tag{9}$$

The first input parameter to the function is ”`step_min`”, a constant number in the range of 0 and 0.5. The function will automatically select the post-fault section of the signal, however, it is required to manually specify the smallest time-step from which the post-fault signal will be selected. From experience, a good estimation of `step_min`$= 0.04$, this value will allow to ignore nonlinearities at the beginning of the signal and will guarantee to keep only the oscillatory part.

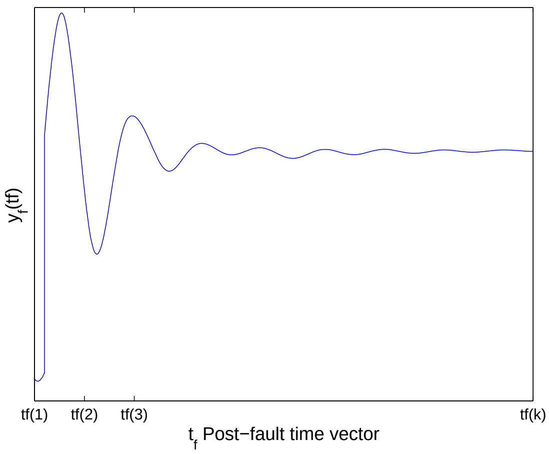

Figure 5: Automatic post-fault signal selection

4.2 Filter signals base on variance σ^2

To illustrate the procedure, we require a case where multiple signals are under analysis. Consider the signals depicted on Figure 6, where for this particular example, the time vector $t \in \Re^{n\times 1}$ and the actual signals $y \in \Re^{n\times nL}$ are defined in 10, with nL indicating the total number of signals.

$$\begin{aligned} y &= \begin{bmatrix} y_1(1) & y_1(2) & y_1(3) & \cdots & y_1(n) \\ y_2(1) & y_2(2) & y_2(3) & \cdots & y_2(n) \\ \vdots & & & & \vdots \\ y_{nL}(1) & y_{nL}(2) & y_{nL}(3) & \cdots & y_{nL}(n) \end{bmatrix}^T \\ t &= \begin{bmatrix} t(1) & t(2) & t(3) & \cdots & t(n) \end{bmatrix}^T \end{aligned} \tag{10}$$

Note that:

- nL is the number of signals analyzed.
- Each signal has different oscillation content.
- In this case, one signal within the set has no oscillation (is a flat line) and is highlighted in red.

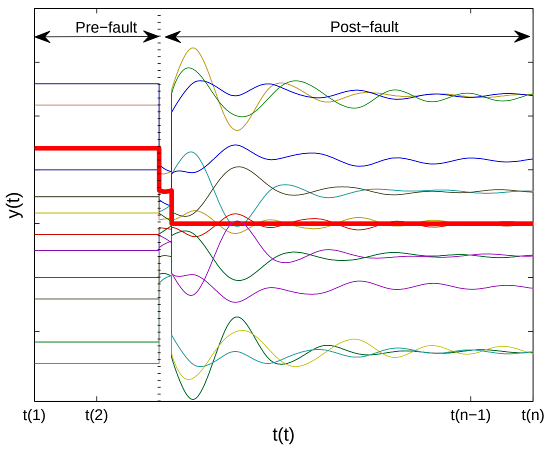

Figure 6: Multiple signals to analyze

After applying the automatic procedure to select the post-fault section of the signals $y_f \in \Re^{k\times nL}$ and redefining the time vector $t_f \in \Re^{k\times nL}$, the signals are set to the same reference (detrended) as described on Figure 7

$$y_f = \begin{bmatrix} y_{f1}(1) & y_{f1}(2) & y_{f1}(3) & \cdots & y_{f1}(k) \\ y_{f2}(1) & y_{f2}(2) & y_{f2}(3) & \cdots & y_{f2}(k) \\ \vdots & & & & \vdots \\ y_{fnL}(1) & y_{fnL}(2) & y_{fnL}(3) & \cdots & y_{fnL}(k) \end{bmatrix}^T \tag{11}$$
$$t_f = \begin{bmatrix} t_f(1) & t_f(2) & t_f(3) & \cdots & t_f(k) \end{bmatrix}^T.$$

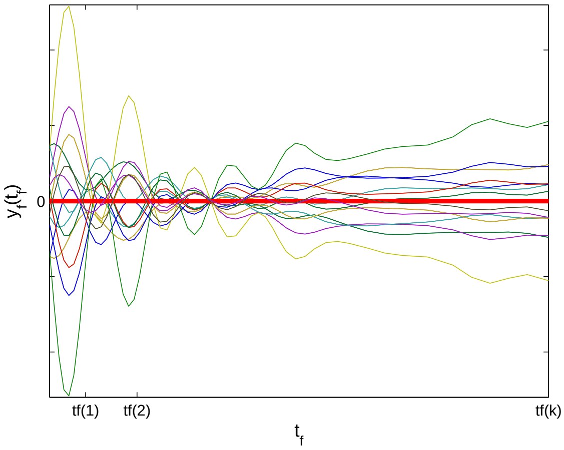

Figure 7: Automatic post-fault selection and detrend when multiple signals are considered.

Now that the signals are in the same reference and only the post-fault section is used, the variance of each signal is calculated using (12)

$$\sigma^2 = \frac{\sum_{i=1}^{k} \left(y_f(t_f)-\mu\right)^2}{k}, \quad \text{and} \quad \mu = \frac{\sum_{i=1}^{k} y_f(t_f)}{k} \tag{12}$$
$$\sigma^2 = \begin{bmatrix} \sigma^2(1), & \sigma^2(2), & \cdots, & \sigma^2(k) \end{bmatrix}$$

where $\sigma^2(i)$ is the variance of the ith signal. Then, the variance is normalized $\hat{\sigma}^2 = \sigma^2/max(\sigma^2)$ and only signals with a normalized variance greater than ε_0, which is pre-defined by the user in the parameter ”`epsilon0`” of the function, are considered for ringdown analysis.

$$\text{if} \quad \hat{\sigma}^2(i) >= \varepsilon_0 \quad \rightarrow \quad \hat{y} = y_{fi} \tag{13}$$

The filtered signals are stored on $\hat{y} \in \Re^{q\times k}$ as shown in (14), where q is the number of signals with normalized variance greater then ε_0, note that $q < nL$. For instance, lets consider the signals shown in Figure 7. If the epsilon is defined as $\varepsilon_0 = 0.1$, this means that only signals with a normalized variance grater than 10% are considered for ringdown analysis. The results are shown in Figure 8 and the following remarks are presented:

- ◇ Only signals with significant oscillations pass through the filter
- ◇ Flat lines and minor oscillatins are ignored (as seen on Figure 8)
- ◇ The value of ε_0 is a constant number between 0 and 1 ($0 < \varepsilon_0 \leq 1$).
- ◇ Manipulating ε_0 the user reduce the number of signals analyzed in the ringdown method, regardless the number of signals given to the function.
    - If $\varepsilon_0 = 0$ then $q = nL$. The variance is ignored and the ringdown method is applied to all signals. High possibility of malfunction of the SSS index and errors can be expected.

- If $\varepsilon_0 = 1$ then q=1. Only one signal will pass through (with highest oscillation), regardless the number of input signals in the function.

$$\hat{y} = \begin{bmatrix} y_1(1) & y_1(2) & y_1(3) & \cdots & y_1(k) \\ y_2(1) & y_2(2) & y_2(3) & \cdots & y_2(k) \\ \vdots & & & & \vdots \\ y_q(1) & y_q(2) & y_q(3) & \cdots & y_q(k) \end{bmatrix}^T \tag{14}$$

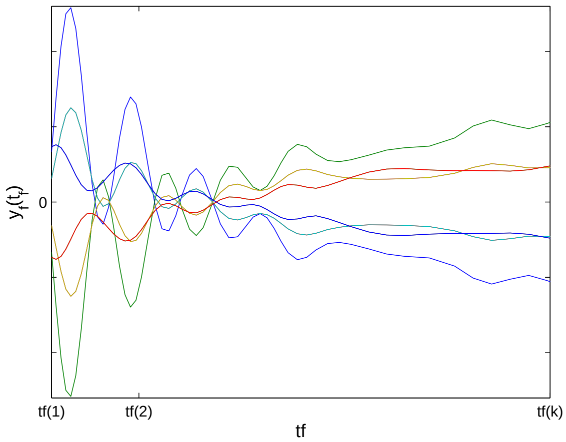

Figure 8: Signals with normalized variance greater than 10%.

4.3 Ringdown Analysis: Prony

Prony approach has been described in [1], full details of this method are also available in [2], [3] and much others. In summary, Prony was designed to estimate the exponential parameters of (15) by fitting a function to an observed measurement. Prony's method is a "polynomial" method and includes the process of finding the roots of a characteristic polynomial.

$$y_i(t) = \sum_{i=1}^{n} a_i e^{\sigma_i t} \cos(\omega_i t + \theta_i) \tag{15}$$

Continuing with the illustrative example, the signals $\hat{y}(t_f)$ described in Figure 8 are reconstructed using Prony approach. The results are shown in Figure 9 and the reconstructed signals are marked with dots. Table 1 displays the number of modes found, the frequencies in Hz and the damping ratios.

Mode No.	Freq (Hz)	Damping (%)
1	1.58	12.40

Table 1: Identified modes within frequencies of: 0.1 and 2.5 Hz

After applying prony's approach, information about the signals such as the identified modes is available and is used to compute the three layer small-signal stability index. The function displays only modes that lie within a predefined range of frequencies f_{min} and f_{max}, indicating the damping value in % as shown on Table 1.

4.4 Main Output: SMI, AMI and GMI

The main output of the function is the matrix SMI, the vector AMI and the constant GMI that form the three layer SSS index. Full details about the layers is available in [1] and [4].

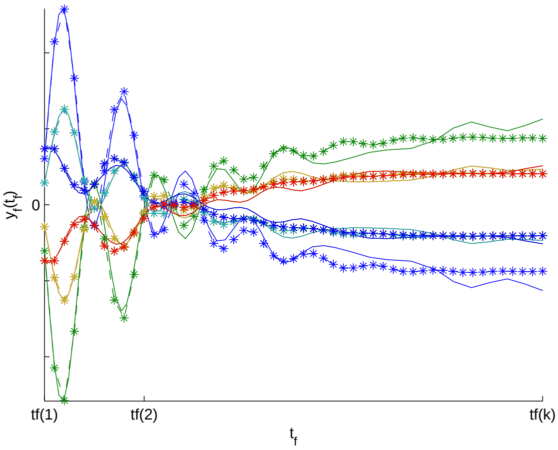

Figure 9: Reconstructed signals after applying Prony to $\hat{y}(t_f)$.

4.5 Summary of changes

- Less input parameters are required.
- The function automatically selects the section where the ringdown method is applied.
- The energy of the signals is not calculated, the variance is calculated instead.
- There is not frequency screening. In the previous function, Fast Fourier Transform (FFT) was computed before the ringdow method. In the new function, the FFT is omitted to simplify the function and reduce the possibility of computational errors.
- Eigensystem Realization Algorithm (ERA) is not longer available for ringdown analysis. To improve the robustness of the function, only Prony is available.
- The actual small-signal stability index (SMI, AMI and GMI) is the same.
- The output provides the same information as before.

4.6 Possible drawbacks of the function

- If the signals used were computed using a fixed time-step solver, the results of the function will not be correct or the function will not work.
- The larger the number of signals provided, the less accurate the results of the index.
- A maximum of one contingency is assumed to exist on each set of time-series. If the time-series contains more than one contingency or event, the function will not work.
- If the contingency is severe and the post-fault time period is short the function might not work. This means that a severe disturbance in the system has occurred at time t_f and the simulation finished close to t_f. In such event, the time-series is highly probable to crush the function application.
- Reverse case of the previous point. A minor contingency has occurred and the post-fault time period is long. This means that a little problem in the system occurred at time t_f, the system reach steady state quickly and the simulation ends much later than t_f. In such event, the time-series is highly probable to be ignored, the index will not be calculated.

A Appendix A

A.1 Input/output data specification for functions execution

This section illustrates how the MATLAB functions for all indexes are called, what are the inputs required and outputs provided. An example using a MATLAB script is given for each function.

A.1.1 Overload Index

The inputs to the overload index are defined as follows

```
%% Static OVERLOAD index inputs
t;              % Time vector
S   = S1;       % Input signal, Apparent power (S) size t x N
p   = 3;        % Exponent used to magnify problems, value between 1 to "inf"
d   = 10;       % Maximum variation allowed from the nominal value in %, e.g, 10 for 10%

% Compute the index
f_x = static_overload(t,S,p,d);

% outputs
%       f_x        overload index
%% eof
```

A.1.2 Under/Over Voltage Index

The inputs to the under/over voltage index are defined as follows

```
% Static UNDER/OVER VOLTAGE index inputs
t;                % time vector
V     = BVm;      % Voltage mangitudes - Matrix of size t x N
p     = 3;        % Exponent used to scale the index, value between 1 to "inf"
d     = 1;        % Maximum variation allowed from the nominal value in %, e.g. 2 for 2%

% Compute the index
[v_x] = static_voltage(t,V,p,d);

% outputs
%       v_x         over\under index
%% eof
```

A.1.3 Transient Stability Index

The inputs to the transient stability index are defined as follows

```
% TRANSIENT STABILITY inputs
t;       % Time vector size t x 1
delta;   % Angle of the machines size t x N;
M;       % Two time interia (H) of the machines (M=2*H), size 1 x N

J=dynamic_transient(t,delta,M)

% outputs
%       J        TS index
```

A.1.4 Small-Signal Stability Index

The inputs to the small-signal stability index are defined as follows

```
%% SMALL-SIGNAL inputs

time   = ts;      % Time vector of order Nx1 with variable time step
signal  = P_line; % Active power flow on relevant lines of order NxM,
                  % where M is the number of signals

step_min = 0.04;  % Minimum step size
epsilon0 = 0.3;   % [0-1] Filter signals with normalized variance
                  % smaller than "epsilon0"
f = [0.1,2.5];    % [fmin,fmax] Range of  frequencies of interest in Hz
d =[0,5,10];      % Damping values in percent (%)
                  % where the index distances will be calculated
Nm  =10;          % Number of modes used in Prony to reconstruct the signals

[ss,y0,G,det]=sssi(signal,time,step_min,epsilon0,f,d,Nm);

ss.smi
ss.ami
ss.gmi

%% Outputs
%
% ss is a structure with the actual small-signal-stability index
%       ss.smi single mode indicator
%       ss.ami all modes indicator
%       ss.gmi global mode indicator
%
% y0 is a structure with the section of signals that where used as input in Prony
%       y0.t   time vector
%       y0.y   signals
%
% G is a structure with Prony output
%       G.Poles detected modes
%       G.Res   residue of each mode
%       G.K     gain of each mode
%       G.that  time vector
%       G.yhat  re-constructed signal
%
```

References

[1] R. Segundo and L. Vanfretti, "Power system stability impact assessment using time series from phasor-time-domain simulations," *iTesla Deliverable D4.3*, pp. 1–44, November 2013.

[2] J. Sanchez-Gasca and D. Trudnowski, "Identification of electromechanical modes in power systems," .*IEEE Task Force Report, Special Publication TP462*, June 2012.

[3] J. Hauer, C. Demeure, and L. Scharf, "Initial results in prony analysis of power system response signals," *Power Systems, IEEE Transactions on*, vol. 5, no. 1, pp. 80–89, Feb 1990.

[4] R. Segundo and L. Vanfretti, "A small-signal stability index for power system dynamic impact assessment using time-domain simulations," *IEEE PES GM Washington DC, USA*, July 2014.