Delivered in partnership with

WP2 Report

SSO Identification Tool

Document Reference: 15167-002-D1

Quality Assurance

TNEI Services Ltd and TNEI Africa (Pty) Ltd. ("TNEI") operates an Integrated Management System and is registered with Ocean Certification Limited as being compliant with ISO 9001(Quality), ISO 14001 (Environmental) and ISO 45001 (Health and Safety).

Disclaimer

National Grid Electricity System Operator ("NGESO") has endeavoured to prepare the published report ("Report") in respect of Probabilistic Stability ("Project") in a manner which is, as far as possible, objective, using information collected and compiled by NGESO and its Project partners ("Publishers"). Any intellectual property rights developed in the course of the Project and used in the Report shall be owned by the Publishers (as agreed between NGESO and the Project partners). The Report provided is for information only and viewers of the Report should not place any reliance on any of the contents of this Report including (without limitation) any data, recommendations or conclusions and should take all appropriate steps to verify this information before acting upon it and rely on their own information. None of the Publishers nor its affiliated companies make any representations nor give any warranties or undertakings in relation to the content of the Report in relation to the quality, accuracy, completeness or fitness for purpose of such content. To the fullest extent permitted by law, the Publishers shall not be liable howsoever arising (including negligence) in respect of or in relation to any reliance on information contained in the "Report".

Document Control

Revi	Status	Prepared by	Checked	Approved	Date
D0	DRAFT	JT, MM, NM, IW, DC	DC		30/05/2023
D1	DRAFT	JT, DC	DC		25/10/2023

TNEI Services Ltd

Company Registration Number: 03891836 VAT Registration Number: 239 0146 20

Registered Address
Bainbridge House
86-90 London Road
Manchester
M1 2PW
Tel: +44 (0)161 233 4800

7th Floor West One
Forth Banks
Newcastle upon Tyne
NE1 3PA
Tel: +44 (0)191 211 1400

7th Floor
80 St. Vincent Street
Glasgow
G2 5UB
Tel: +44 (0)141 428 3180

TNEI Ireland Ltd

Registered Address: 104 Lower Baggot Street, Dublin 2, DO2 Y940

Company Registration Number: 03891836 VAT Registration Number: 239 0146 20

For enquires and general
Correspondence - use
Manchester
office details
Tel: +353 (0)190 36445

TNEI Africa (Pty) Ltd

Registered: Mazars House, Rialto Rd, Grand Moorings Precinct,7441 Century City, South Africa

Company Number: 2016/088929/07

1st Floor Willowbridge Centre
39 Carl Cronje Drive
Cape Town
South Africa, 7530
Tel: +27 (0)21 974 6181

Contents

APPENDICES

TABLES

FIGURES

1 Introduction

Sub-synchronous oscillations (SSO) in a system arise primarily from the presence of poorly damped oscillatory modes resulting from the interaction between different technology types. Addressing these interactions poses several challenges: firstly, the complex nature of the interactions makes it difficult to identify them in advance without thorough studies. Secondly, pinpointing the specific components or assets involved in the interaction can be a challenging task. Furthermore, the introduction of new elements into the network, such as the connection of new sites or network reinforcements, can potentially introduce additional poorly damped modes into the system.

The utilization of a state-space approach is a valuable technique for identifying oscillatory modes within a power system and determining their underlying causes. This involves examining the states that contribute significantly to these oscillatory modes and implementing control measures on the relevant power system components to effectively mitigate the modes. Linear stability analysis tools such as eigen decomposition, participation factor analysis, eigenvalue sensitivity analysis, and pole-placement enable the analysis of system instability, design of damping controllers, and related tasks. However, achieving a comprehensive understanding of the system is essential for successful state-space modelling. In traditional systems with synchronous generators and associated controllers like exciters and power system stabilizers, this understanding was attainable. In modern systems, the presence of converter-interfaced sources results in control algorithms that are vendor-specific and treated as confidential commercial assets. As a result, these algorithms are typically provided as black-box models, disclosing only input-output relationships without internal details. Manufacturers share these models as binary files (e.g., .lib and .dll) for use with numerical solvers in conducting time-domain simulations. However, such simulations, particularly for large practical power systems and electromagnetic transient studies, can be highly complex and exhaustive, often lacking a clear understanding of the root cause.

Considering this context, the objective of this project is to create a framework that enables system operators to efficiently analyze numerous scenarios and identify potential sub-synchronous oscillation (SSO) phenomena. The proposed framework will incorporate a combination of automation techniques, frequency domain analysis methods and machine learning. By automating the entire process with minimal reliance on manual supervision by engineers, the framework aims to streamline the identification of SSO events.

This report focuses on the development and implementation of an SSO identification tool, which aims to provide efficient methods for detecting and analysing sub-synchronous oscillations in power systems using frequency domain techniques. The purpose of this report is to introduce the overall process involved in the implementation of the tool. This report does not include any case study results and it does not serve as a manual to use the tool. WP3 report will include a detailed guide to use the tool along with case studies carried out on the Cigre Benchmark System.

The report is structured into several sections, starting with an exploration of frequency domain methods in Section 2. This section includes a review of impedance scan methods (Section 2.1) and a detailed explanation of the SSO scenarios screening process (Section 2.2). Section 3 introduces the grey box method, which plays a crucial role in the identification of SSO phenomena. Moving on to Section 4, the CIGRE Benchmark and its corresponding PSCAD model are discussed, providing an overview of the modelling effort along with the frequency scanning component. In Section 5, the automation of PSCAD code is explained, describing the classes and methods available to interact with PSCAD's python API and automate the studies. Section 6 focuses on the user interface and back-end deployment solutions, outlining the structure and feature set of the tool, including scenario selection, parameter/configuration definitions, simulation execution, progress visualization, result visualization, user assistance, and support. Section 7 examines the feature engineering and machine learning classification, highlighting the computation of features and the structure of the classification module in the tool. Section 8 covers software testing, encompassing unit testing, integration testing, system testing, and acceptance testing. These sections collectively provide a comprehensive overview of the project and its various components, highlighting the approach taken and the outcomes achieved.

2 Frequency domain methods

Several analysis methods are available in the time domain and frequency domain. While time domain analysis has clear benefits like the ability to model non-linear dynamics etc, it is a time-intensive process and especially in the future context where several scenarios need to be studied, time domain analysis on its own does not stand out as the winner. Therefore, it is necessary to investigate frequency domain methods to quickly filter out scenarios and then apply time domain analysis for only a limited number of scenarios.

Introduced in the WP1 report [1], Figure 2-1 depicts an overview of the SSO identification process; the figure shows that the process has several steps or sub-processes. After receiving an initial baseline model, the first subprocess implements a screening method based on frequency domain analysis and builds a grey box representation of the model. In this subprocess the user defines a set of attributes, such as gid strength, active and reactive power dispatch, or the number of WTG in service, to build multiple scenarios and screen for potential SSO events. The grey box module is included in this subprocess to enable root cause analysis by identifying participation of IBRs in poorly damped subsynchronous oscillatory modes. The main goal of this subprocess is to reduce the initial list of SSO scenarios and focus only on those that have the potential to cause SSO events; in this way, the computational burden of multiple unnecessary EMT simulations will be reduced.

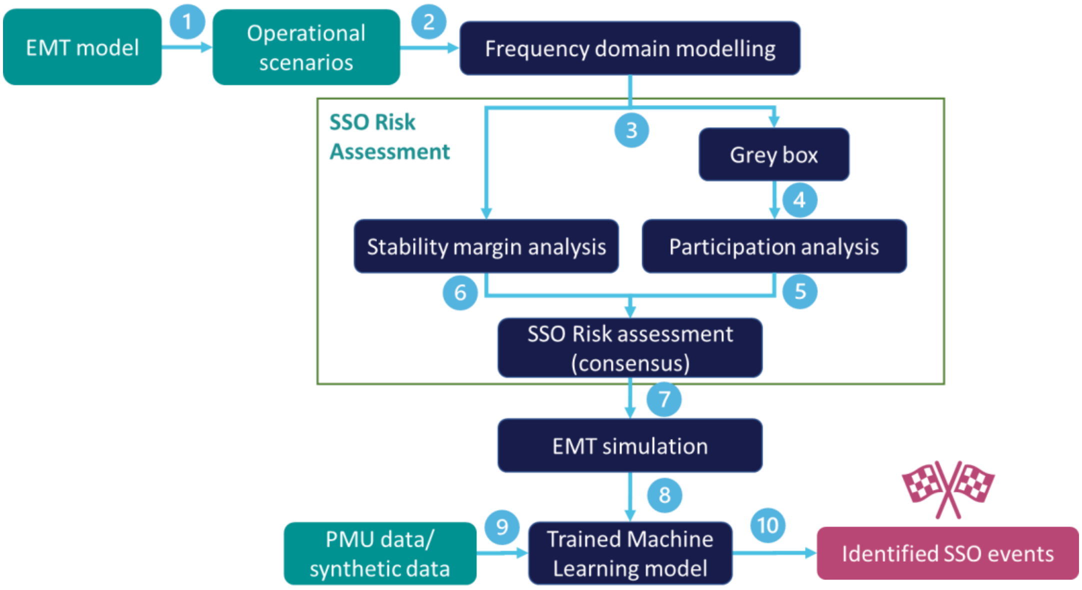


Figure 2-1 Overview of the SSO identification process.

As seen in Figure 2-1, the SSO identification process contains several sub-processes or tasks that enable the detection of SSO events. The interaction between tasks is enabled by the transfer of different data objects. For instance, the Frequency Domain Modelling (FDM) receives a set of operational scenarios defined by modification of parameters of the base-case model. The FMD runs the impedance scan creating PSCAD case files and setting the corresponding parameters. The output of the FDM are sets of impedance values in sequence domain for the frequency range under study, including frequency coupling results. The corresponding data, depending on the type of scan, is used in the stability margin and grey-box analyses. The data exchanged in each step of the process is listed in Table 2-1

Table 2-1 List of data exchanged in each step of the SSO identification process from Figure 2-1

Step	Exchanged data
1	PSCAD model file read by the SSO tool UI.
2	PSCAD model file updated by the tool based on user selected operational scenarios and global settings.
3	Frequency vs complex Impedance values in modified sequence domain. Results saved in frequency_results.db database file.
4	Identified modes based on system identification.
5	Participation of different apparatus in the identified modes.
6	Damping and phase margin calculation based on reactance crossover and impedance intersection methods. Results saved in frequency_results.db database file.
7	Consensus based on SSO/non-SSO outcome from each of the three frequency domain methods. All scenarios tagged as SSO risk, No SSO risk or Inconclusive and saved in tagged_scenarios.db file.
8	SSO risk scenarios (including inconclusive scenarios) are studied in PSCAD and instantaneous current and voltage measurements are passed to the ML model.
9	Instantaneous data from different sources used to train an ML model to identify SSO in any time domain signal.

10	Scenarios studied in step 8 are further labelled as SSO or non-SSO based on conclusive EMT simulations.

2.1 Review of Impedance Scan Methods

In the literature review of WP1, several methods were identified with the potential to be used in the screening method of the SSO identification tool. After a critical review, WP2's focus centred on the impedance-based analysis. Impedance scan analysis is a valuable technique for studying the frequency domain behaviour of nonlinear devices, particularly in the context of wind farm controllers. This analysis provides insights into the dynamic behaviour and impedance characteristics of the controller, enabling the evaluation of its performance in regulating power output and the detection of potential issues. The nonlinear nature of wind farm systems should be considered, as it can impact the results of impedance scans. Conducting scans under various operating conditions and power output levels is crucial for accurate assessment. Moreover, impedance scan analysis facilitates the identification of interactions between the wind farm controller and other power system components, such as stabilizers and regulators, which aids in assessing overall system stability.

The critical review of relevant literature enabled the design of the process implemented in the SSO tool. A summarised overview comparing the advantages and drawbacks of the reviewed methods can be seen in Table 2-2; a detailed review can be found in Appendix B. The comparison and analysis allowed the identification of which methods are more suitable in the context of SSO identification. In particular, papers [2] [3] [4] [5] offer advantages that are useful and applicable; therefore, they were used in the design process.

Table 2-2 Comparison of reviewed frequency domain methods in literature

Method in	Main feature	WTG Model type	Stability assessment method	Advantages	Drawbacks
Paper [2]	Sequence domain impedance matrix	Type 4	Generalised Nyquist criteria	Identification of the influence (coupling) between positive and negative sequence	The frequency injection procedure is not clear

Paper [3]	Sequence domain admittance matrix	Type 3 and 4	Generalised Nyquist criteria (MIMO). Nyquist or Bode plot (SISO).	If the Grid impedance does not have coupling the MIMO system can be analysed as SISO.	The mathematical derivation of the SISO form is not fully described.
Paper [6]	Frequency coupling between AC and DC networks	Type 4	Bode plots	Defines the DC and AC side impedances, which is relevant for HVDC	Requires detailed modelling of the VSC. The method is based on injections on the DC and AC sides.
Paper [4]	Impedance-based stability criterion	Type 3 and 4	Verification of the intersection of the grid side and IBR impedance	It is a simple method to verify the damping level.	Requires independent frequency scans of the system and IBR.
Paper [7]	Reactance-based condition for potential SSO	Type 3	Verification of the value of the WTG resistance at reactance crossover frequencies.	The method is simple and easy to understand. It can be used for radial and non-radial networks	Requires two sets of harmonic injections. Requires independent frequency scans of the system and WTG.
Paper [5]	Improved reactance-based scan considering frequency coupling	Type 3	Verification of the value of the system impedance resistance at reactance crossover frequencies.	The method is more accurate by including a frequency coupling term to compute the impedance. It does not require	Requires two sets of harmonic injections.

				scanning the WTG and grid sides independently	

The usefulness or accuracy of the impedance-based analysis depends on the frequency scanning technique. The modified sequence domain method used in the tool is particularly designed to measure impedance of VSC converters. So, the expectation is that the calculated impedance will be accurate irrespective of the technology type (e.g., WTG, STATCOM, HVDC) as long as it is a voltage source converter. We did not come across any papers investigating the same for LCC technologies so we are not confident about how well the tool will work when used for scanning LCC HVDC links but something we can investigate as part of a future project. Once the impedance is calculated, the second step is to apply the intersection method to assess the phase margin. This theory is applicable to GFL converters only. We need to investigate how to extend the concept to GFM converters. So, to summarise, the current version of the SSO tool will work with any technology types as long as these are VSCs operating in GFL mode.

2.2 SSO scenarios screening process

The review of relevant frequency domain methods (see Table 2-2) provided the information required to design the screening process within the SSO identification. Figure 2-2 describes the screening process based on frequency domain analysis to focus only on those SSO scenarios that have the potential to trigger SSO events. In the first stage, the screening process receives an SSO scenario characterised by a series of parameters defined by the user in combination with a PSCAD baseline model. A frequency injection component is used in PSCAD to inject perturbations at different frequencies. In the second stage, the time domain voltage and current measurements are used to determine the impedance of a given frequency; the goal is to obtain the positive sequence and coupling impedances.

Once the impedance components have been computed for all the frequencies in the frequency range of interest it is possible to assess the stability; in combination with stability margin thresholds, the process will not only verify the stability margin in the sub-synchronous range by performing standard frequency domain analysis, like bode and Nyquist, but also using total reactance crossover and system-WTG impedance intersection. A consensus of different stability assessments will identify if the system is at risk of SSO based on frequency domain (small signal) analysis.

Default threshold values have been defined for each frequency domain method to implement the consensus approach. For grey box method, a 5% damping factors is set as the threshold to identify poorly damped modes. For impedance intersection, a 90° phase margin is set as the stability threshold and for the reactance crossover a 20Ω resistance is considered as minimum damping requirement in the system. These values are ballpark figures and used as default parameters in the tool. Through extensive testing under different operational scenarios and different proprietary model, practical threshold values will need to be identified. This will be partly covered in WP3, but the majority of the validations will be done by ESO using models supplied by different vendors.

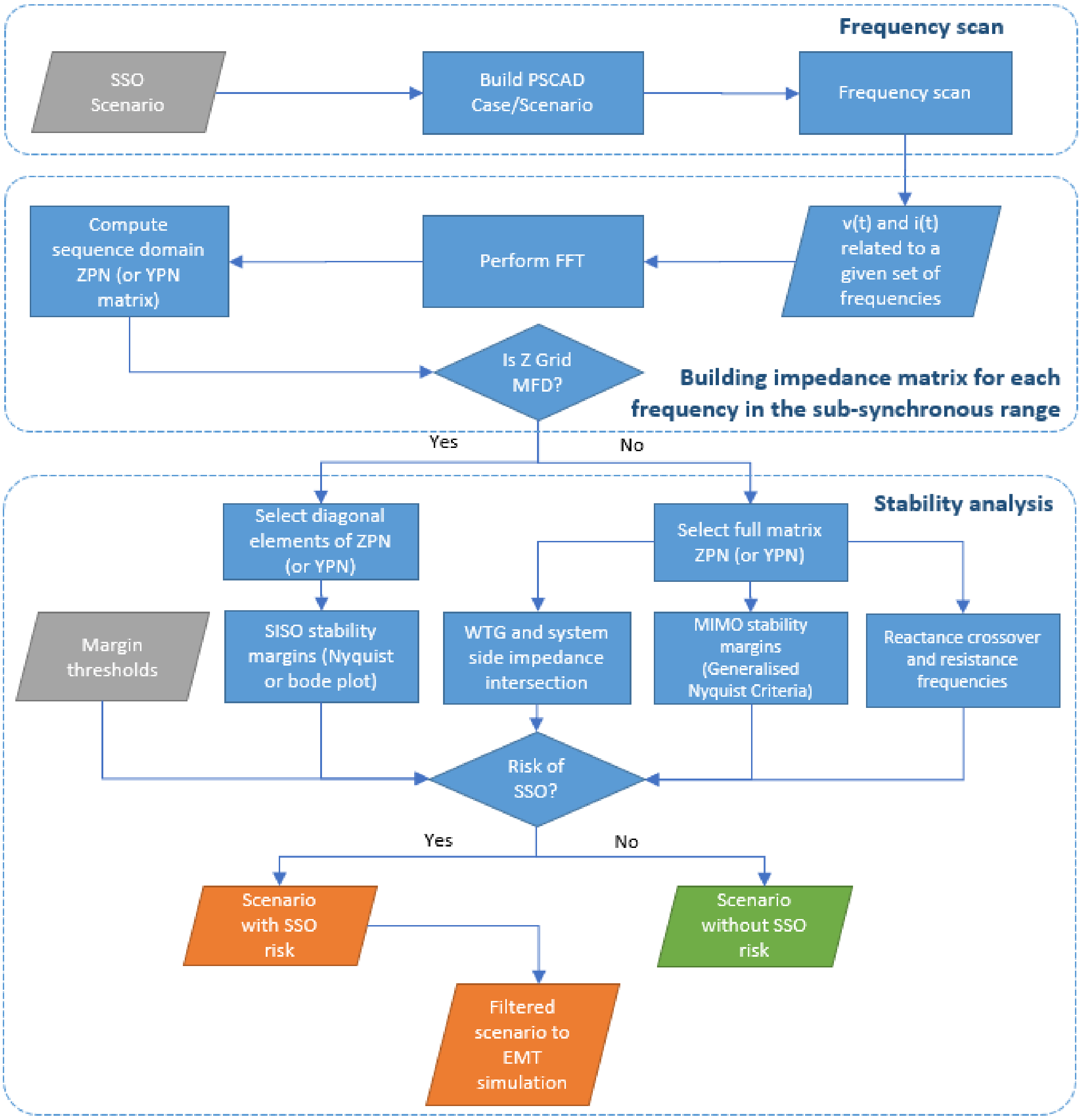

Frequency scan
SSO Scenario
Build PSCAD Case/Scenario
Frequency scan
Compute sequence domain ZPN (or YPN matrix)
Perform FFT
v(t) and i(t) related to a given set of frequencies
Is Z Grid MFD?
Building impedance matrix for each frequency in the sub-synchronous range
Yes
No
Stability analysis
Select diagonal elements of ZPN (or YPN)
Select full matrix ZPN (or YPN)
Margin thresholds
SISO stability margins (Nyquist or bode plot)
WTG and system side impedance intersection
MIMO stability margins (Generalised Nyquist Criteria)
Reactance crossover and resistance frequencies
Risk of SSO?
Yes
No
Scenario with SSO risk
Scenario without SSO risk
Filtered scenario to EMT simulation

Figure 2-2 Flowchart screening process based on frequency domain analysis.

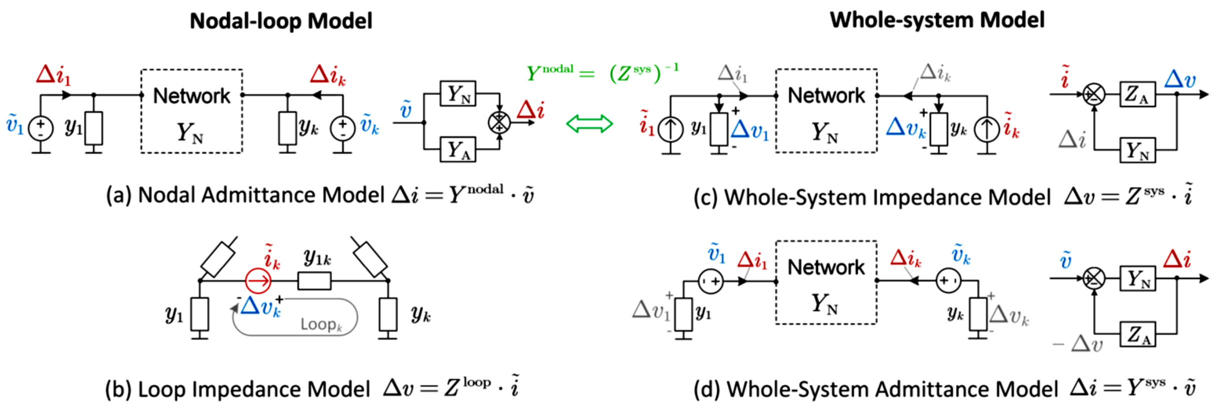
3 Grey box method

3.1 Overview of the grey box approach.

This section provides a brief introduction to the grey-box analysis technique and some initial results from its implementation on an IEEE 14 bus test system. The WP1 report [1] can be referred to for more detail: Section 5.2 for the whole-system modelling methodology and Section 5.3 for the grey-box technique.

The "grey-box" method [8] allows some insight into the relationship between the whole-system oscillation modes and the impedance of connected apparatus using only black-box frequency response data. The "whole-system" terminology refers to the system which is formed when apparatus are connected to the grid, so considers the response of the entire system at once rather than analysing apparatus in isolation (which would require a separate model or in practice disconnection of apparatus). It is formed by the feedback loop between the grid and apparatus shown in Equation 3.1. Here, $Z_A(s)$ is the block diagonal matrix of 2x2 apparatus *dq* impedances and $Y_N(s)$ is the network bus admittance matrix.

$$
\begin{aligned}
Z^{sys}(s) &= Z_A(s)\big(I + Y_N(s)Z_A(s)\big)^{-1} \\
Y^{sys}(s) &= \big(I + Y_N(s)Z_A(s)\big)^{-1}Y_N(s)
\end{aligned}
\qquad 3.1
$$



(a) Nodal Admittance Model $\Delta i = Y^{\text{nodal}} \cdot \tilde{v}$

(b) Loop Impedance Model $\Delta v = Z^{\text{loop}} \cdot \tilde{i}$

(c) Whole-System Impedance Model $\Delta v = Z^{\text{sys}} \cdot \tilde{i}$

(d) Whole-System Admittance Model $\Delta i = Y^{\text{sys}} \cdot \tilde{v}$

Figure 3-1 Whole-system models vs nodal admittance and loop impedance modes [9].

The whole-system admittance transfer function matrix represents the input/output relationship between series voltage injections at each bus and resultant changes in current from the grid to the apparatus. A series voltage is used because the bus voltages are generally set by strong voltage sources which would not allow a parallel voltage disturbance.

The whole-system impedance transfer function matrix represents the bus voltage response to parallel current injection, which is used because it would be impractical to inject a current disturbance in series with a strong current source. Both whole-system transfer function matrices are proper and have frequency responses which can be measured online with the injection of probing disturbances. Both $Z^{sys}(s)$ and $Y^{sys}(s)$ are transfer function matrices since there are 2N input locations and 2N output locations, where N is the number of buses (and 2N since the system is modelled in the *dq* reference frame). The 2x2 matrix transfer functions on the diagonal are the responses which can be measured locally.

Either the whole-system impedance or the whole-system admittance can be used in the grey-box approach so the admittance $Y^{sys}(s)$ shall be used for demonstration here.

Equation 3.2 describes the relation between the whole system admittance model $Y^{sys}(s)$ and the state space representation of the system; where A, B and C are the state matrix, input matrix and output matrix. x, u and y are the state vector. Through modal-decomposition and coordinate transformation, we consider $z = \Psi x$, and $x = \Phi z$, where Ψ, Φ correspond to the left eigenvector matrix and right eigenvector matrix, respectively, and $\Psi = \Phi^{-1}$. Every entry in $Y^{sys}(s)$ shares the same poles with the state matrix of the whole system, which can be found using the standard eigen-decomposition of its corresponding state-space A matrix such that the diagonal matrix of eigenvalues $\Lambda = \Psi A \Phi = diag(\lambda_1, \lambda_2, \dots \lambda_N)$. Here the eigenvalues $(\lambda_1, \lambda_2, \dots \lambda_N)$ describe the whole-system oscillation modes, where the imaginary part represents the mode oscillation frequency and the real part represents the rate of decay.

$$Y^{sys}(\mathrm{s}) = C(sI - A)^{-1}B = C\Phi(sI - \Lambda)^{-1}\Psi B \qquad 3.2$$

From this the *k*th element of $Y^{sys}(s)$ can be found as per Equation 3.3, where ϕ_i is the right eigenvector and ψ_i is the left eigenvector corresponding to the eigenvalue λ_i.

$$Y_{kk}^{sys}(\mathrm{s}) = \sum_{i=1}^{m} \frac{c_k \phi_i \psi_i b_k}{s - \lambda_i} = \sum_{i=1}^{m} \frac{R_{ki}}{s - \lambda_i} \qquad 3.3$$

The terms R_{ki} are known as the system *residues* and are critical to the grey-box analysis technique. The residues can be extracted from a measured frequency response (obtained from measurement or a black-box model) by applying a method of system identification known as vector fitting [10] [11].

Vector fitting is essentially a curve-fitting algorithm which uses the frequency response data to iteratively solve linear problems for the pole locations and residue values until convergence is achieved [12]. Since the problem is linear it can be solved computationally efficiently [10] [13], and has been applied widely across the fields of microwave engineering, electronics and high-voltage power systems. For more details see [12] or [11].

It was found in [8] that the residues indicate the sensitivity of the corresponding whole-system oscillation mode to a change in the impedance of the apparatus connected at the selected bus, as shown in Equations 3.4 and 3.5 [10]. Large residue magnitudes indicate that a small change in apparatus impedance can significantly affect the mode.

$$\Delta\lambda_i \approx \langle -R_{ki}, \Delta Z_{Ak}(\lambda_i)\rangle_F \tag{3.4}$$

$$\Delta\lambda_i \approx -\left(R_{ki}\Delta Z_{Ak,dd}(\lambda_i) + R_{ki,dq}\Delta Z_{Ak,qd}(\lambda_i) + R_{ki,qd}\Delta Z_{Ak,dq}(\lambda_i) + R_{ki,qq}\Delta Z_{Ak,qq}(\lambda_i)\right) \tag{3.5}$$

If the residues of the whole-system impedance are used instead, the deviations in apparatus impedance $\Delta Z_{Ak}(\lambda_i)$ terms should be replaced by the corresponding deviations in apparatus admittance $\Delta Y_{Ak}(\lambda_i)$.

If there is further information known about the apparatus connected to a bus where the residues are measured, information previously only calculable with a whole-system white box model can be found. For example, state-participation factors or the sensitivity of a whole-system mode to a parameter within the apparatus.

These have been characterised as ‘layers’, where Layer 1 just compares the norms of the 2x2 matrix of residues at each bus and requires no further information about the apparatus connected. Layer 2 requires prior knowledge of the impedance of the apparatus at the bus and uses the residues to determine whether scaling the impedance, increasing the number of connected apparatus (in series or parallel), will improve or worsen the mode damping.

Finally, Layer 3 uses the sensitivity of the apparatus impedance to an internal parameter (e.g., controller gains or delays) to determine the effect on mode damping of changing a parameter. In the same manner that the state participation factor is the sensitivity of the mode to the corresponding diagonal component of the state ‘A’ matrix, these layers have been likened to impedance and parameter participation factors. The layers are illustrated in Figure 3-2.

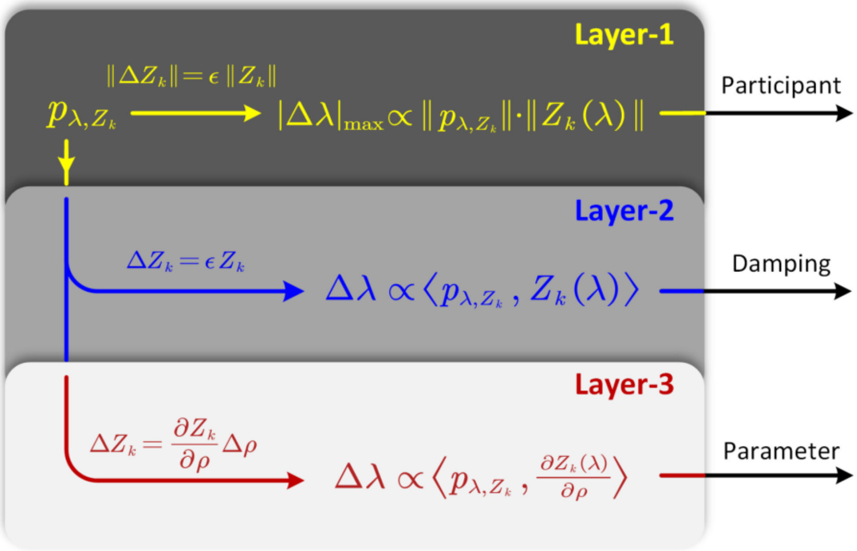

Layer-1
$\|\Delta Z_k\| = \epsilon\,\|Z_k\|$
p_{λ,Z_k}
$|\Delta\lambda|_{\max} \propto \|p_{\lambda,Z_k}\| \cdot \|Z_k(\lambda)\|$
Participant
Layer-2
$\Delta Z_k = \epsilon Z_k$
$\Delta\lambda \propto \langle p_{\lambda,Z_k}, Z_k(\lambda)\rangle$
Damping
Layer-3
$\Delta Z_k = \frac{\partial Z_k}{\partial \rho}\Delta\rho$
$\Delta\lambda \propto \langle p_{\lambda,Z_k}, \frac{\partial Z_k(\lambda)}{\partial \rho}\rangle$
Parameter

Figure 3-2 The grey-box layers [8].

3.2 Validation of the grey box approach based on IEEE 14 Bus network.

Vector fitting requires the specification of the number of modes and residues to extract from the system frequency response data, which in practice is often guided by the number of resonant peaks present in the data. There must be at least as many frequency sample points as the fitting order and specifying too high an order risks overfitting to noise, so the order used will be lower than the actual order of the system. This means that a well-damped mode $(-a + jb)$ will have a large and positive damping factor $-a/\sqrt{a^2 + b^2}$ due to their large negative real part, and will not be picked up by the vector fitting – however problematic SSO modes are usually poorly damped. The level of accuracy with which vector fitting can extract the residues with noisy data and underspecified system order is an ongoing area of research.

A state-space (white-box) model of an IEEE 14 bus network [9] has been developed by Southampton University and is used here for validation of the grey-box approach. The state-space model is used as a black-box model to generate frequency response data (i.e., $Z^{sys}(j\omega)$). The vector fitting [11] method has been applied to $Z^{sys}(j\omega)$ to extract the residues and perform the grey-box analysis. For the IEEE14 Bus network, the order of the state-space system is 245. The order of the vector fitting will be determined as the minimum order which gives an acceptable fit to the frequency response data. This will generally be at least as many peaks as there are present in the data. Therefore, the order must be greater to get an acceptable fit, although care should be taken not to overfit to noise/inaccuracies.

Figure 3-3 shows the bode plot for several components of the whole-system impedance frequency response matrix. The components chosen are the *dd* quadrants for a subset of the system busses which correspond to those in the reference publication [9]. The modes picked up by vector fitting with an order of 10 are plotted in Figure 3-4(a) and (b), where the poorly damped 20 Hz SSO mode is visible (it can also be seen as a peak in all the responses in Figure 3-3).

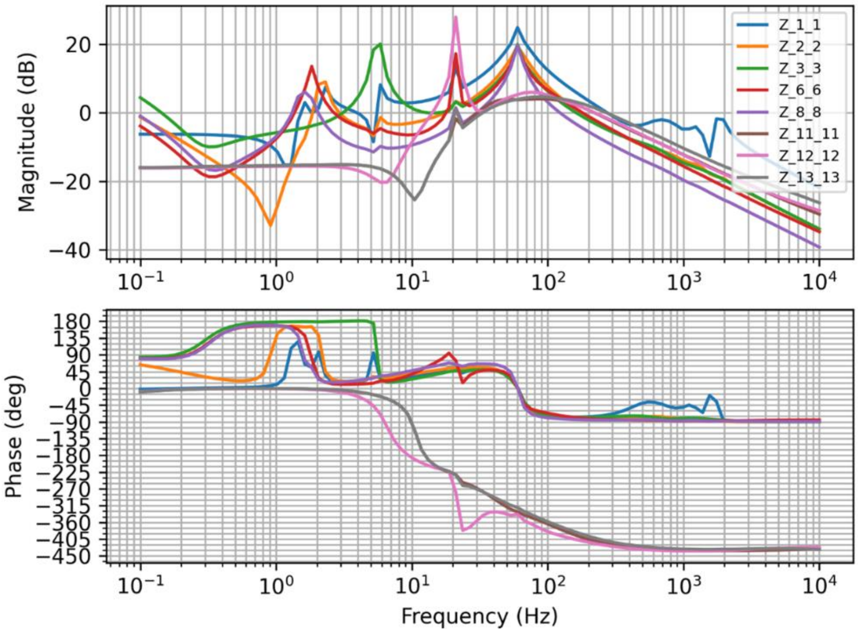


Figure 3-3 Whole system frequency response Z $(j\omega)$ for various buses

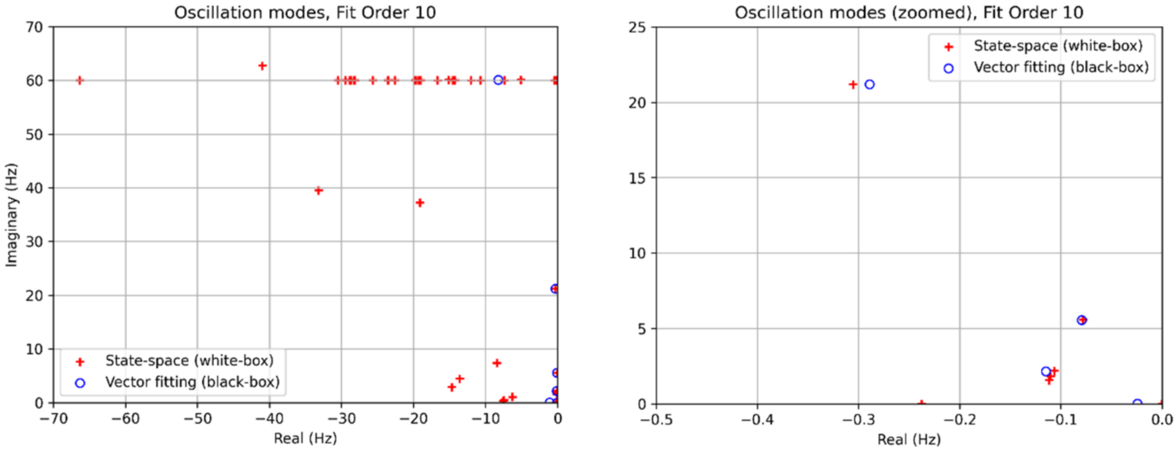


Figure 3-4 System oscillation modes vs the modes identified by the vector fitting (a); zoomed to SSO eigenmodes (b)

The Layer 1 & Layer 2 results for the 20 Hz SSO mode generated using the white box MATLAB model are shown in Figure 3-5. Using the grey-box method by extracting the residues from the frequency response gives the Layer 1 results shown in Figure 3-6, where it can be seen that the vector fitting correctly identifies Apparatus 12 as the main contributor to the SSO mode. This is confirmed with the state-participation analysis (using the white-box model) shown in Figure 3-7 (note the states are only described by the Apparatus they belong to).

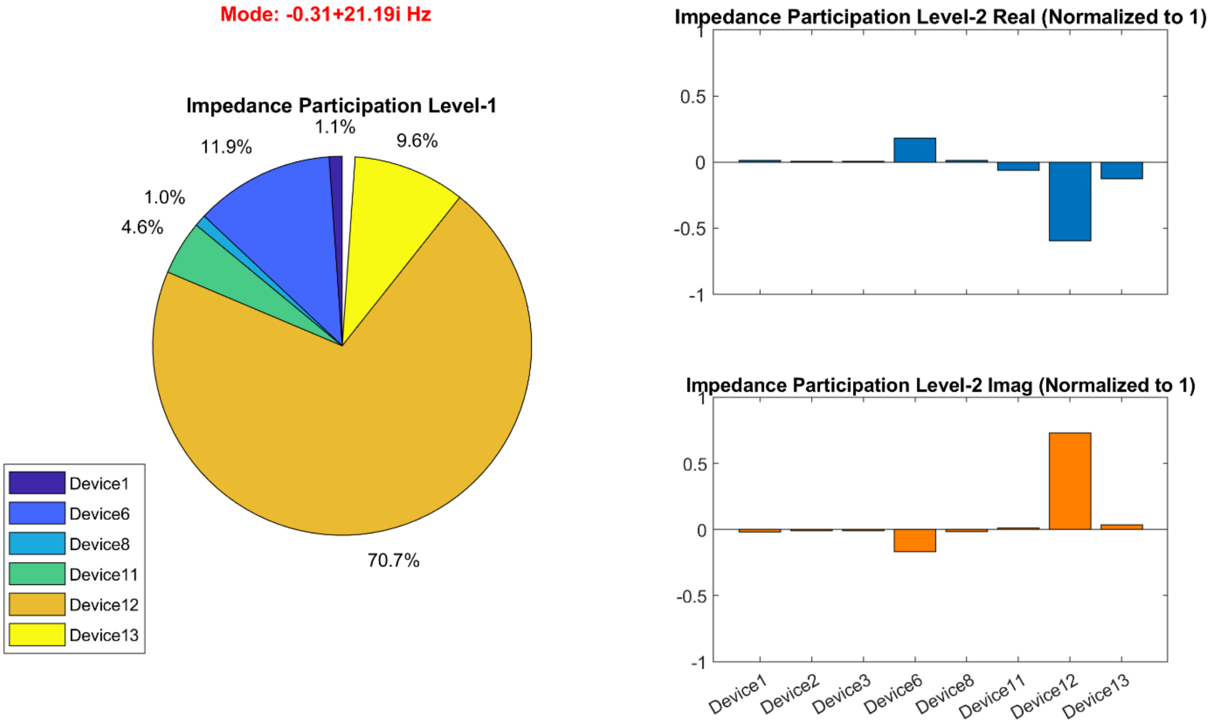


Figure 3-5: Layer 1 & Layer 2 analysis using the white box model.

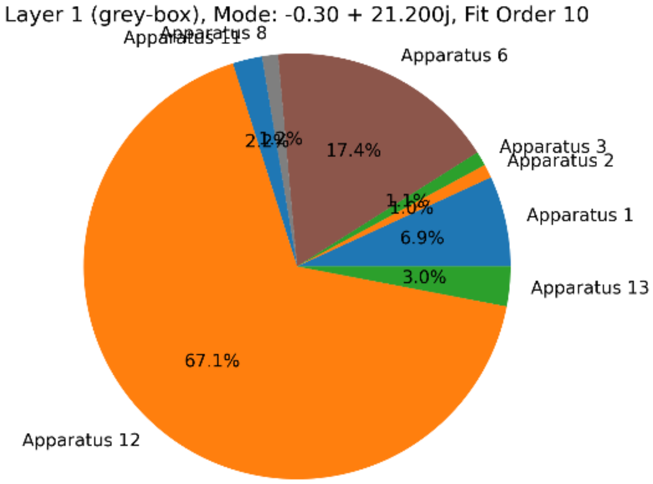


Figure 3-6: Layer 1 analysis using the vector fitting results and the grey box method.

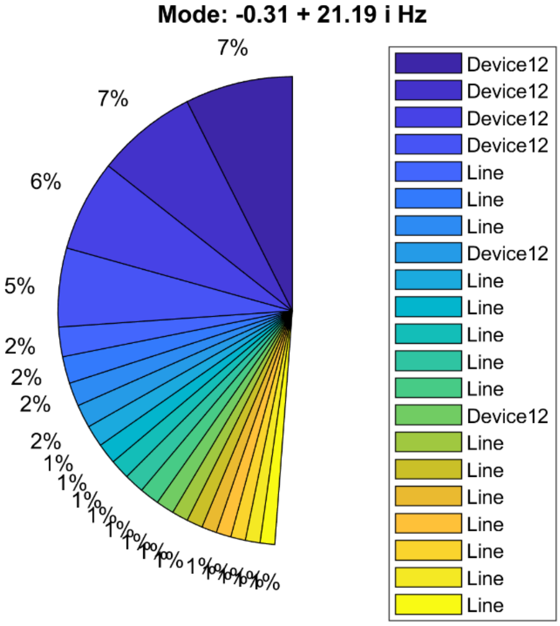


Figure 3-7: State participation factor analysis for the 20 Hz SSO mode (note the state names indicate which apparatus the states belong to)

Appendix C details the results for other modes and shows that with a fitting order of 10, even though the model order is 245, the vector fitting can reproduce the Layer 1 results well for poorly damped, isolated modes. If there are multiple modes close together the accuracy of the Layer 1 results is degraded. [9] [8]

To improve the accuracy of the vector fitting to distinguish the modes which are close together in frequency and damping (or to analyse any of the other modes which were missed) the order of the vector fitting should be increased. Increasing the order will increase the risk of overfitting to noise or process errors, although this risk should be lower when using black box model results rather than real measurements.

The next steps are to use the information on the apparatus impedance to implement the layer 2 grey box analysis and use the grey-box technique to analyse the CIGRE Benchmark system. Further thought will also be given to the process of choosing the most suitable order for the vector fitting.

4 CIGRE Benchmark – PSCAD model

4.1 Overview of the modelling effort

CIGRE working group C4.49, which studies the multi-frequency stability of converter-based modern power systems, has developed a benchmark model to study SSO phenomena. An overview of the system and its characteristics is provided in Appendix A.

The CIGRE benchmark system is developed by the CIGRE working group in MATLAB/Simulink. For this project, an equivalent representation of the benchmark system is required in PSCAD. The objective is to develop a model comprised of similar building blocks which yield a similar steady-state and dynamic response.

A typical challenge encountered when replicating models across different software packages is the lack of one-to-one matches in components, as well as nuances in implementation between similar components. As an example, for implementation of a specific component/function, one software package may have composite black-box modules (with inputs and outputs) available which performs a certain function, whereas in another this composite module must be built up from first principles using elementary components.

An attempt has been made to replicate the hardware topology and control system parameters of the CIGRE benchmark system in PSCAD to as accurate an extent as practically possible. The PSCAD model has been constructed using components and models available in PSCAD's master library.

The system consists of a 400 kV transmission grid, modelled as an ideal voltage source behind the grid impedance, as well as a grid consisting of transmission lines and transformers which step the voltage down from 400 kV to 220 kV, and from 220 kV to 66 kV. Two GFL wind farms, one being 240 MW and the other 180 MW, connect to the grid via 66/0.69 kV transformers. The measurement and control implementation at the two wind farms are similar.

The following measurements are taken at each wind farm: The instantaneous values of voltage and current at the 66 kV side of the 66/0.69 kV step-up transformer, the instantaneous values of voltage and current in the middle of the converter output filter (between the series inductor and the shunt capacitor), as well as the DC input voltage to the converter. These five measurement signals are passed through an anti-aliasing filter and then sampled, where they are translated from the analogue into the digital domain. The control models use the measured grid voltage as input to a phase-locked loop (PLL), where the angular frequency ω and phase angle ωt are computed. The phase angle ωt is an essential parameter that is required to implement abc-to-dq and dq-to-abc domain transfers. These domain transformations take sinusoidal three-phase quantities and essentially translate them into dc quantities for easier implementation in a digital control system. The control system includes a reactive power control loop, a current control loop, and a DC power control loop. The control system synthesizes a reference dq voltage which is transferred into the three-phase domain using a dq-to-abc transformation. This reference voltage waveform is compared with a triangular carrier waveform to perform pulse width modulation (PWM). The PWM-generated pulses are passed as gate signals to a six-pulse IGBT-based converter bridge, which fires the IGBTs. The phase and amplitude of this reference voltage waveform are manipulated by the control system to inject the desired amount of active and reactive power into the grid.

A high-level overview of the overall PSCAD model, showing the external grid voltage source, transmission lines, transformers, and the wind farm modules to the far right, is shown in Figure 4-1 below. Figure 4-2 provides a high-level overview of the wind farm model, showing the converter output filter, as well as sub-modules for the converter bridge, the DC side, as well as the controller module and its measurement signals.

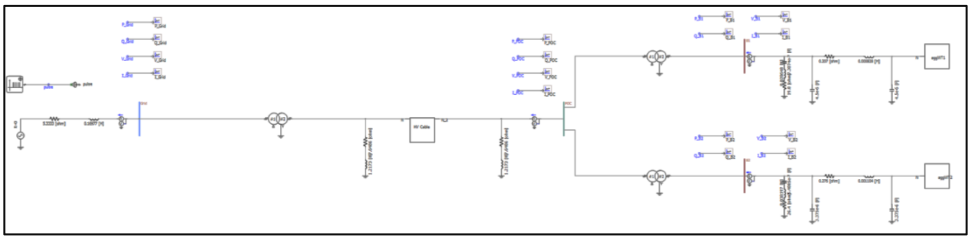


Figure 4-1 High-Level Overview of Overall PSCAD Benchmark Model

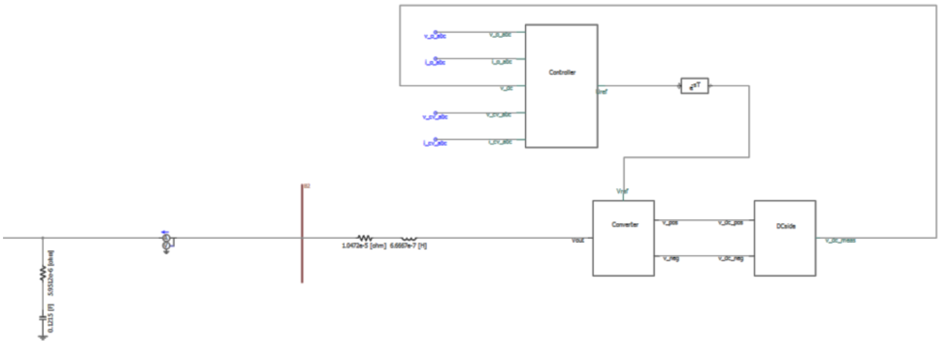


Figure 4-2 High-Level Overview of Wind Farm Module

Following extensive piece-wise building and testing of the PSCAD model, the entire system, comprised of the external grid, both wind farms and all control systems, was tested under steady-state and perturbed conditions to compare the response to the MATLAB/Simulink model. Some tuning and adjustment were then affected until a satisfactory response was yielded from the PSCAD model.

4.2 Frequency scanning component

PSCAD's built-in *Scenarios* feature can be used to perform parameter sweeps on the model to investigate their influence on the stability of different modes. The Scenarios feature enables the user to define parameter sets using PSCAD's dial, switch, and slider components. Each parameter set saves the specific dial, switch, and slider component input values to a specific scenario.

PSCAD is also used to perform frequency scans on the converter models. This screening technique aims to determine if the converters offer any negative damping to sub-synchronous frequency currents, which provides insight into their potential SSO behaviour.

Traditional frequency scan tools, which are typically built-in modules in power systems analysis software packages, are not appropriate for this purpose. Power electronic converters exhibit active and highly non-linear behaviour which cannot be accurately captured by traditional frequency sweep tools. To obtain an accurate approximation for the frequency domain behaviour of a converter, a frequency scan method for active/non-linear elements needs to be applied.

Extensive research has been conducted in this domain [14], [15], [16] and [17], which lead to the development of frequency scan techniques for active/non-linear elements which have been validated in conjunction with leading original equipment manufacturers (OEMs).

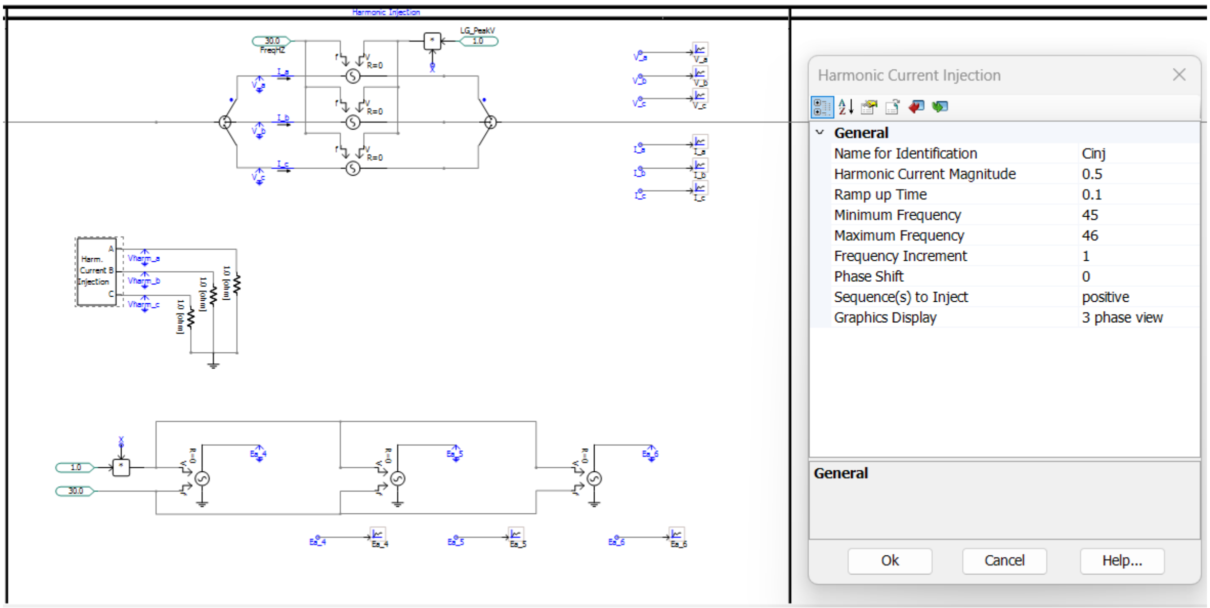
Badrzadeh et al. [17] provide several practical considerations on its application, as well as explanations as to why some other techniques fail in correctly capturing the converter's sub-synchronous impedance behaviour.

The method is based on a small-signal white noise excitation with a sub-synchronous frequency spectrum that is superimposed on the steady-state, fundamental frequency excitation waveform. For the application of this technique, the exact modelling of the network is not important, and the network is best represented by a simple voltage source. This also prevents potential background interference from the network side from distorting the results.

During the application of the technique, parameters must be tuned to obtain a white noise signal with a magnitude that is sufficiently large to obtain meaningful results, and small enough so that it does not introduce instabilities and interfere with the nominal operating point of the converter. Key to this is determining a sufficient amplitude of the injection, as well as tuning the individual phase angles of the injected harmonics. This is needed to counteract the phenomenon associated with beat frequency interference, where unwanted superposition can cause large spikes in the white noise signal which could potentially interfere with the nominal operating point of the converter. To this end, a quadratic phase shift, proposed by Jiang and Gole [18], can be applied. It is also recommended to inject the harmonics in small bands, rather than the entire spectrum at once since the effect of beat frequency interference becomes more pronounced with larger bands of harmonics, and this can cause interference with the nominal operating point of the converter.

Once sufficient white noise signals have been synthesized, the injected signals should be analysed in the stable region, where their RMS values have settled to constant values. The white noise voltage and corresponding white noise current that enters the converter can then be used to estimate the virtual impedance, and consequently the virtual resistance (damping), for sub-synchronous frequency ranges.

As part of the PSCAD model, a tool which performs the above-mentioned frequency scanning method has been developed. In particular, an overview of the frequency scan module is shown in Figure 4-3; the figure shows the schematics of the module and the interface to modify the injection parameters, such as frequency range and increment, ramp-up time, current magnitude, type of sequence, and phase shift. The frequency scanning tool is fully built in PSCAD and requires no external modules to perform its function. The frequency scanning module is also interfaced with the CIGRE benchmark model that has been developed in PSCAD. The reader can find a review on the selection of point of injection/measurement and type of injection in Section 4 of the WP1 report [1].



Harmonic Current Injection

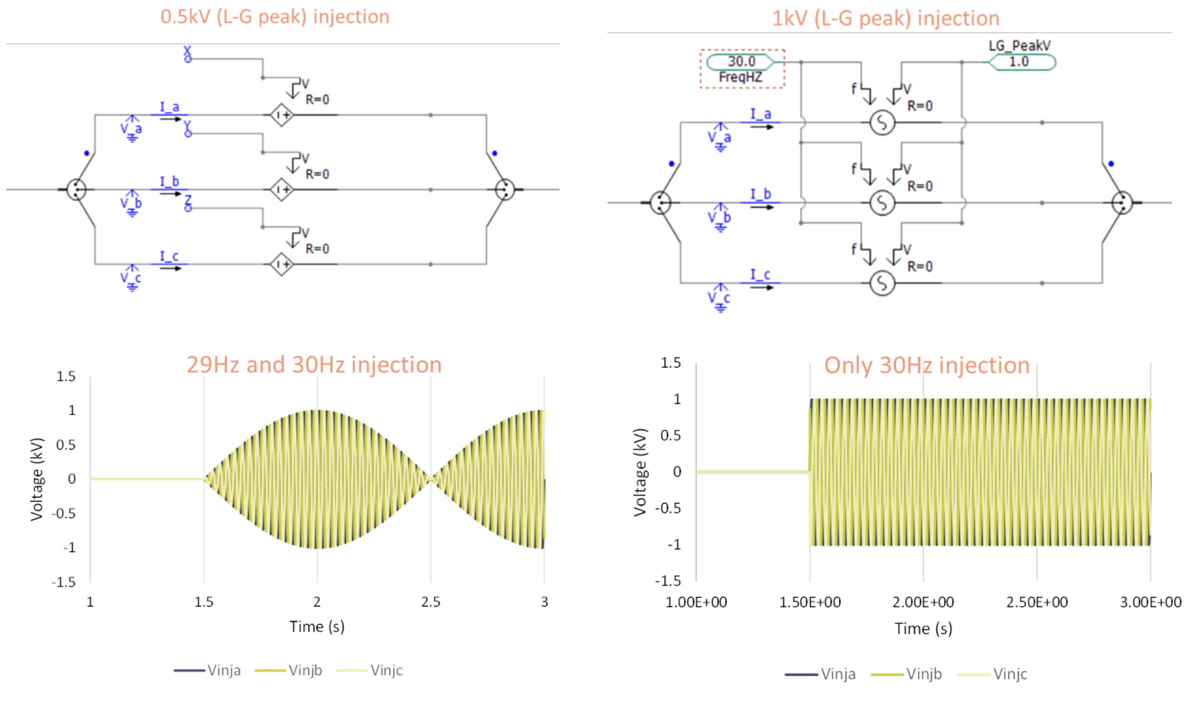
General

Name for Identification	Cinj
Harmonic Current Magnitude	0.5
Ramp up Time	0.1
Minimum Frequency	45
Maximum Frequency	46
Frequency Increment	1
Phase Shift	0
Sequence(s) to Inject	positive
Graphics Display	3 phase view

Figure 4-3 Overview of the frequency scan module in PSCAD

The developed frequency injection module allows users to select between the two types of injection shown in Figure 4-4. The DC injection method can inject multiple frequencies at one go while the AC injection method injects only a single frequency. To give an idea about the rough scale of difference in the frequency scanning time, let’s assume that we want to find out only the positive sequence impedance from 1Hz to 20Hz. The DC method is configured to inject 2 frequencies at one go[1] so to scan up to 20Hz, there will be 10 injections from the frequency scan module (Figure 4-3). If each injection takes 60 secs, then the total time will be 600 secs to get values from 1Hz to 20Hz. Compared to this method, the AC injection will inject individual frequencies 20 times. The simulation time for each injection is comparable for both methods. Therefore, it will take 1200 secs for the AC injection method to get values from 1Hz to 20Hz. In reality, the second method will only be used when there is coupling between sequence components. However, to accurately capture the coupling, two separate injections are necessary for each frequency. This means that to perform a scan from 1Hz to 20Hz using the AC injection method, the total time taken would be 2400secs.

[1] The minimum setting is 2 frequency injections. This is a restriction from PSCAD. There is no upper limit for the number of frequencies to inject in a single burst. However, due to the phenomenon of ‘beat frequency’, we advise not to inject more than 5 frequencies in a single burst. The frequencies should be injected with quadratic phase shift.



(a) (b)

Figure 4-4 Two types of frequency injection technique in PSCAD, (a) DC injection (multi-frequency) and (b) AC injection (single frequency)

5 PSCAD Automation Module

At the core of the SSO Identification tool is the PSCAD automation module which takes user selected options from the user interface and applies those changes to the network model. Several steps are involved in the process, which are handled by the 'pscad' class in the automation module. This is the only module in the SSO identification tool which interfaces with the PSCAD API. Figure 5-1 provides an overview of the module and shows the different functions available to interact with the software.

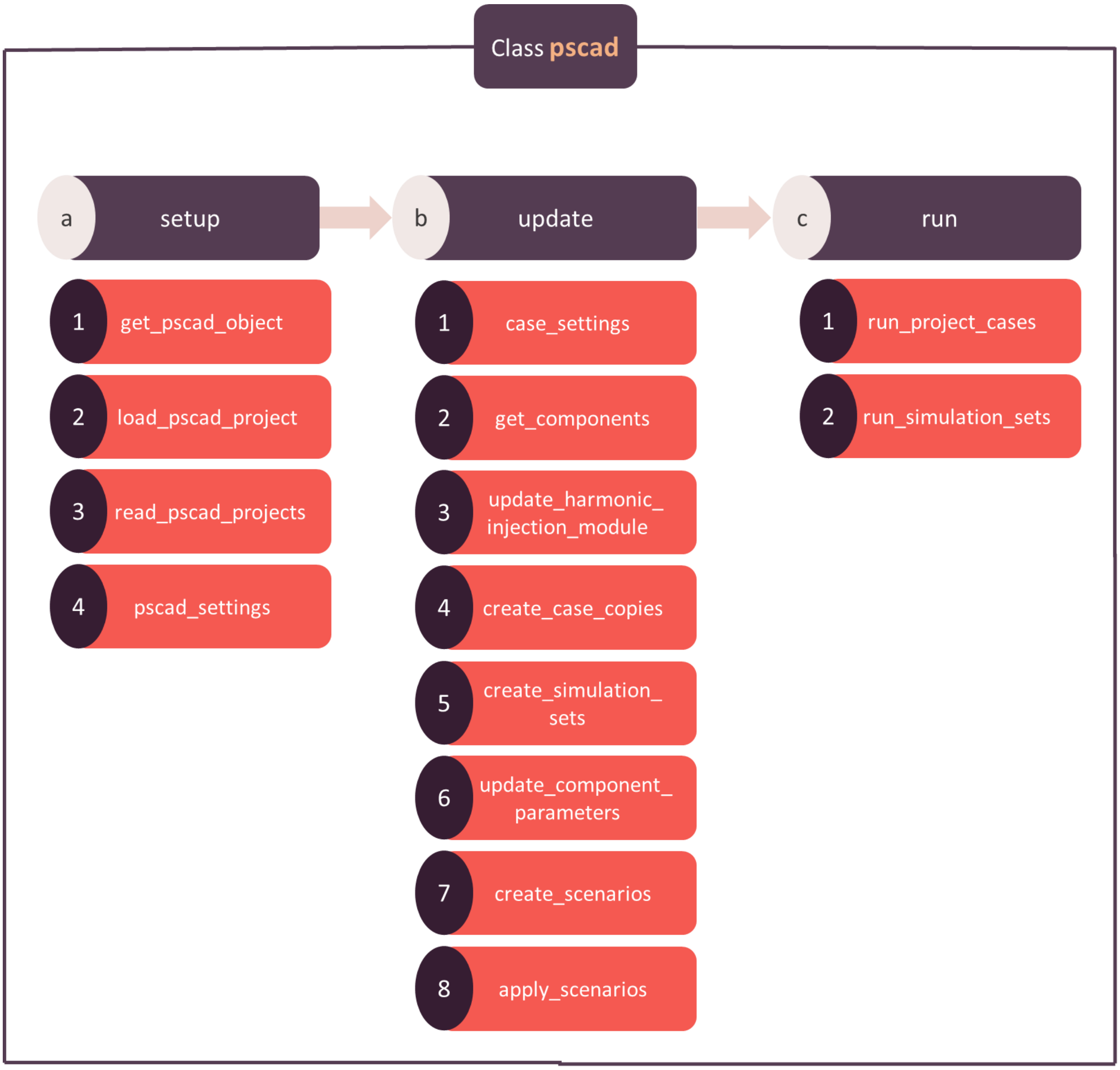


Figure 5-1 Overview of functions in the PSCAD automation module

The ‘pscad’ class in the automation module has 14 different functions to break down the overall tasks into smaller operations. All these functions are written as private i.e., they are not intended to be accessed from any other module. The only publicly available functions are a) ‘setup’, b) ‘update’ and c) ‘run’ and only these are available to any module interfacing with the automation functions.

1. Setup

The setup function provides access to four private functions that are listed in the diagram from 1 to 4.

The first function a.1 is the first step in the automation process and interfaces with PSCAD’s python API to access the PSCAD object.

The second function a.2 loads the base case PSCAD project selected by the user from the user interface.

Function a.3 reads the PSCAD project and accesses all necessary project settings, parameters, and network components.

The last function a.4 applies the user specified (super user) PSCAD settings to the project such as number of concurrent simulations, compiler selection etc.

2. Update

The update function provides access to eight private functions. Function b.1 applies PSCAD ‘case’ specific settings such as simulation time, simulation time step etc.

Function b.2 is an optional function and depends on user selection. When enabled, all components in the base case network model are returned along with their respective parameters as a csv output file.

Function b.3 updates the parameters of the frequency scan block in the network model.

Function b.4 creates copies of the base case model depending on the number of operational scenarios created by the user in the user interface.

Function b.5 creates simulation sets and assigns the case copies equally across different sets. Currently, the functionality is restricted to just one simulation set. In the next work package, we will explore the optimal allocation of cases in simulation sets to speed up the simulation time.

Function b.6 updates all component parameters requested by the user through the scenario generator feature in the user interface of the tool.

Function b.7 creates 'scenarios' within PSCAD[2] based on the user defined operational scenarios requested through the user interface.

Function b.8 applies the created scenarios in b.7 to the respective PSCAD cases.

3. Run

The run function offers access to two private functions. Function c.1 allows users to run specific project cases sequentially. This is useful if the user wants to run only a few cases for debug purposes.

Function c.2 runs the simulation set thereby implementing parallel run of PSCAD cases. The number of parallel cases will depend on the specific PSCAD license available to the user. The default number of parallel runs is set to eight.

[2] Scenario feature in PSCAD is different from the scenario generator in the user interface of the tool. To avoid confusion, we have used PSCAD-scenarios whenever to refer to native scenarios feature within PSCAD.

6 User Interface and Back-end Deployment Solutions

The user interface (UI) of the SSO tool serves as the core implementation that will be refined and developed further in WP3 (Work Package 3). The UI acts as the primary means for users to interact with the tool and access its features. It provides a graphical representation of the tool's functionality and allows users to input data, configure settings, and view results. This section describes the design of the UI and some details around implementation.

6.1 Overview and structure

The importance of the UI lies in its role as a user-friendly gateway to the SSO tool. It allows users, who may not have technical expertise or knowledge of the underlying code, to effectively utilize the tool's features. By providing a visually appealing and intuitive interface, the UI enhances the user experience and increases the accessibility of the SSO tool.

The UI also enables the underlying code to be utilized for engineering more bespoke features and processes. While the UI may provide a predefined set of features, the flexibility of the underlying code allows for customization and extension of functionality. This means that as the SSO tool evolves and new requirements arise, the UI can be modified and expanded to accommodate these changes, while still leveraging the existing codebase.

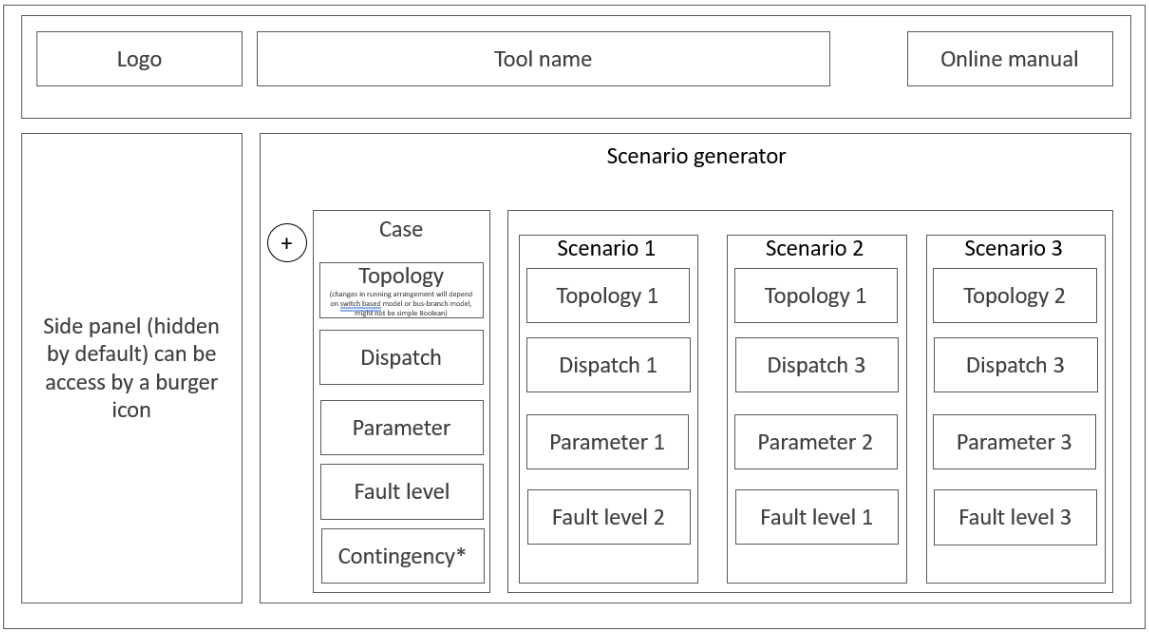


Figure 6-1 Scenario generator page

This wireframe diagram represents the page where users can create custom simulation scenarios using a kanban-style approach. It allows users to specify configurations and group them into cases for simulation and SSO identification. The scenario selection feature enables comprehensive exploration of all available and configurable parameters, providing flexibility in designing simulation cases.

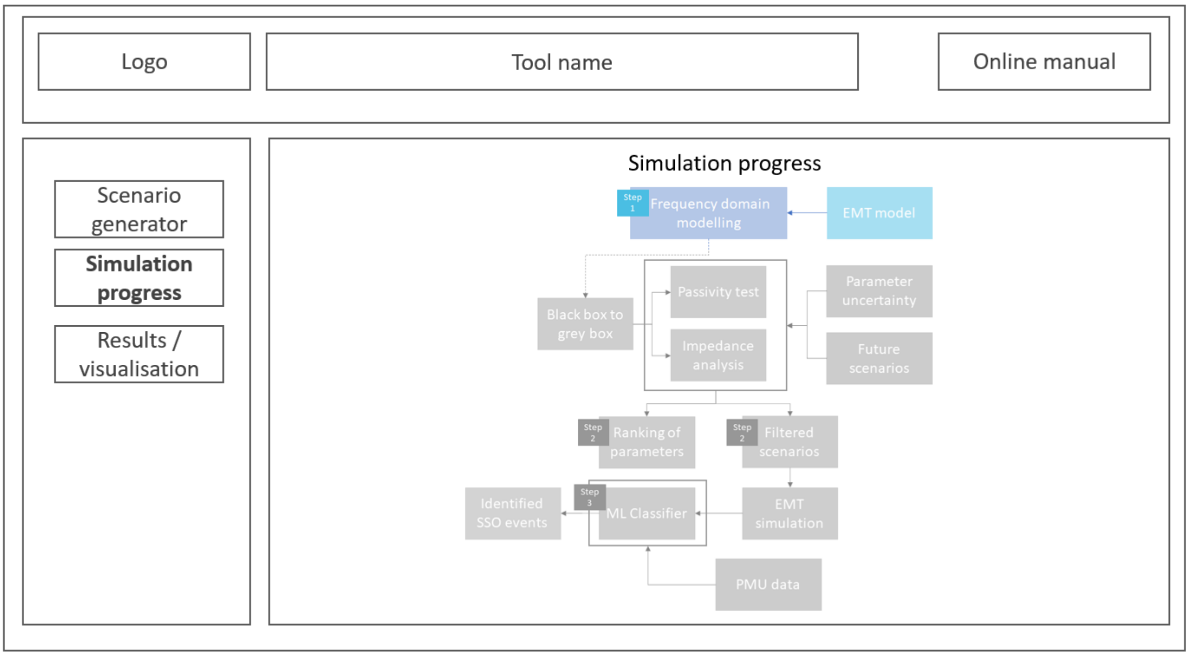


Figure 6-2 Simulation progress page

This wireframe diagram illustrates the page that shows the progress of simulations and SSO identification processes. As these processes can be time-consuming, the UI provides a progress visualization feature to keep users informed about the current stage of the overall process. This feature aligns with the flow chart in Figure 2-1, ensuring transparency and clarity in the simulation workflow.

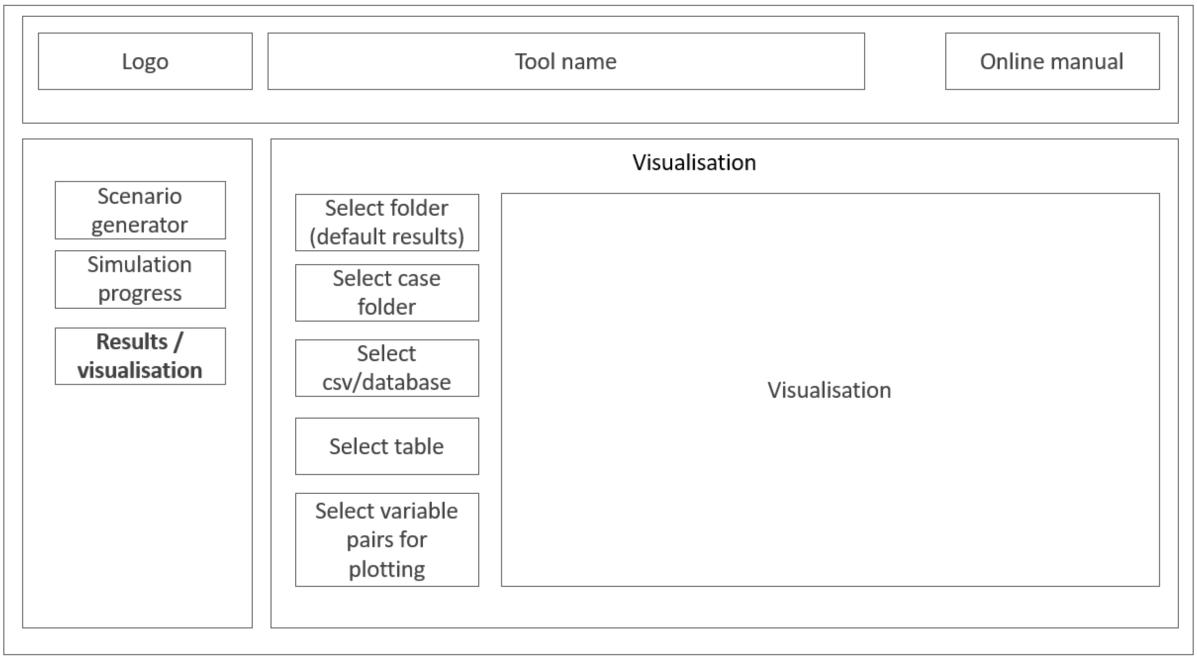
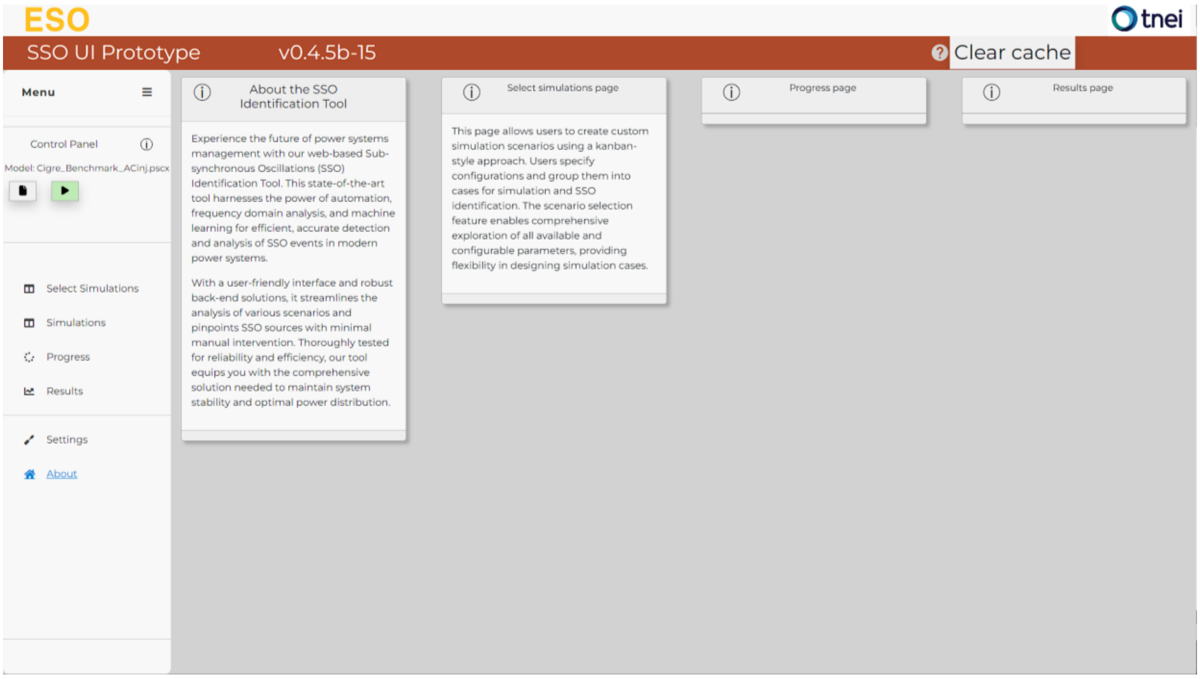


Figure 6-3 Data visualisation page

This wireframe diagram showcases the page dedicated to visualizing and comparing results from simulations and identification processes. The UI includes parallel plots, implemented as matplotlib figures initially, which can be extended to use more interactive libraries such as Plotly. This enables users to explore and analyse the simulation data in a more detailed and interactive manner, enhancing their understanding of the results.

A screenshot of the actual user interface home page is shown below.



About the SSO Identification Tool

Experience the future of power systems management with our web-based Sub-synchronous Oscillations (SSO) Identification Tool. This state-of-the-art tool harnesses the power of automation, frequency domain analysis, and machine learning for efficient, accurate detection and analysis of SSO events in modern power systems.

With a user-friendly interface and robust back-end solutions, it streamlines the analysis of various scenarios and pinpoints SSO sources with minimal manual intervention. Thoroughly tested for reliability and efficiency, our tool equips you with the comprehensive solution needed to maintain system stability and optimal power distribution.

Select simulations page

This page allows users to create custom simulation scenarios using a kanban-style approach. Users specify configurations and group them into cases for simulation and SSO identification. The scenario selection feature enables comprehensive exploration of all available and configurable parameters, providing flexibility in designing simulation cases.

6.2 Feature set

6.2.1 Scenario selection

This feature allows users to design custom simulation cases according to their unique requirements. This is made possible through a kanban style interface which enables a smooth organization and visualization of tasks. Each task represents a specific configuration that is grouped into cases for simulation. In essence, this feature makes it possible to conduct a comprehensive exploration of all configurable parameters and identify possible SSO events.

6.2.2 Parameter/configuration definitions

Figure 6-4 depict an overview of the features that have been analysed to be considered of parameters to define different SSO scenarios. These features can be classified into four distinct categories, providing a structure for system adjustments. (i) Dispatch setpoints involve tuning generator and compensation equipment parameters. (ii) Manual adjustments of physical parameters facilitate physical modifications to the system. (iii) Topological adjustments mean altering the system structure by adding or removing components. (iv) Fault perturbation conditions handle situations where there are large or small disturbances in the system.

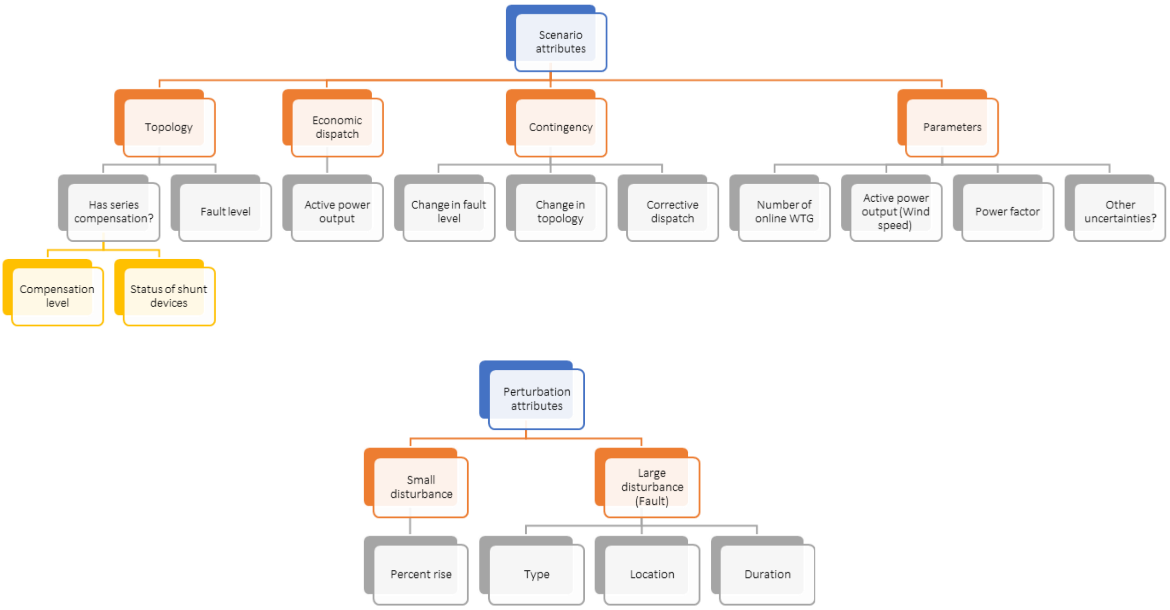
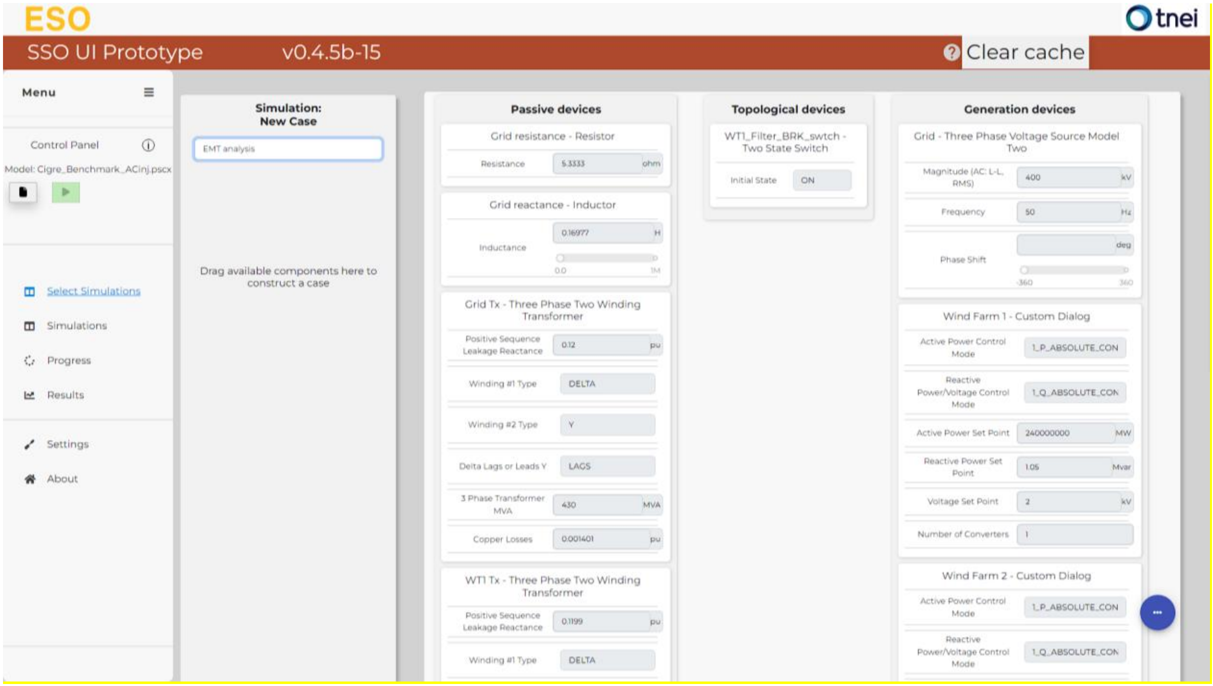


Figure 6-4 Overview of preliminary SSO scenario features

Below we have included a screenshot from the 'Select Simulations' page that shows how the different scenario attributes mentioned in Figure 6-4 can be selected from the UI.



Model: Cigre_Benchmark_ACinj.pscx

Simulation: New Case

EMT analysis

Drag available components here to construct a case

Passive devices

Grid resistance - Resistor

Resistance: 5.3333 ohm

Grid reactance - Inductor

Inductance: 0.16977 H (0.0 – 1M)

Grid Tx - Three Phase Two Winding Transformer

Positive Sequence Leakage Reactance: 0.12 pu

Winding #1 Type: DELTA

Winding #2 Type: Y

Delta Lags or Leads Y: LAGS

3 Phase Transformer MVA: 430 MVA

Copper Losses: 0.001401 pu

WT1 Tx - Three Phase Two Winding Transformer

Positive Sequence Leakage Reactance: 0.1199 pu

Winding #1 Type: DELTA

Topological devices

WT1_Filter_BRK_swtch - Two State Switch

Initial State: ON

Generation devices

Grid - Three Phase Voltage Source Model Two

Magnitude (AC: L-L, RMS): 400 kV

Frequency: 50 Hz

Phase Shift: deg (-360 – 360)

Wind Farm 1 - Custom Dialog

Active Power Control Mode: 1_P_ABSOLUTE_CON

Reactive Power/Voltage Control Mode: 1_Q_ABSOLUTE_CON

Active Power Set Point: 240000000 MW

Reactive Power Set Point: 1.05 Mvar

Voltage Set Point: 2 kV

Number of Converters: 1

Wind Farm 2 - Custom Dialog

Active Power Control Mode: 1_P_ABSOLUTE_CON

Reactive Power/Voltage Control Mode: 1_Q_ABSOLUTE_CON

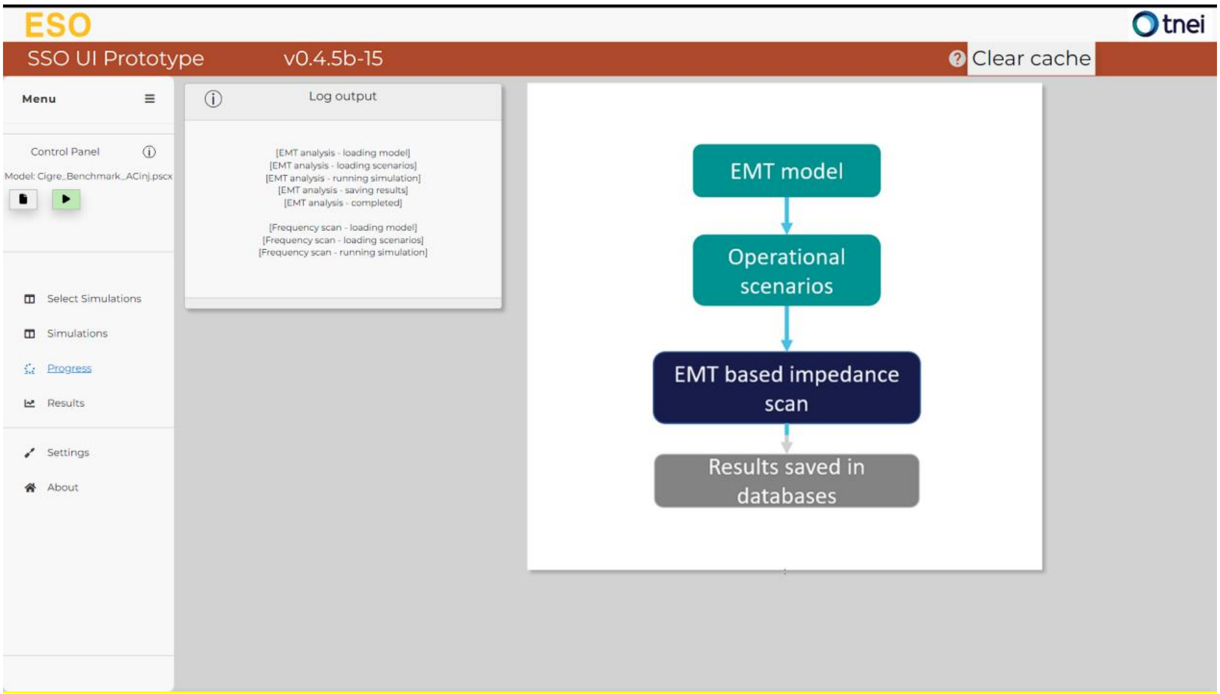
6.2.3 Simulation execution

With this feature, users can select from various processes provided by the SSO tool engine. This offers flexibility and adaptability in the simulation process, enabling the use of different algorithms or methods to perform SSO identification.

6.2.4 Progress visualisation

Given the potential for lengthy simulations, this feature provides users with updates on the progress of their simulations and SSO identification processes. This reduces uncertainty about the stage of processing and keeps users informed about ongoing tasks.

A screenshot of the 'Progress' page from the UI. This page has a log output window which updates the user on the progress of the simulation. We have included a flow chart as well which highlights each process in the analysis as the tool steps through the individual modules.



Model: Cigre_Benchmark_ACinj.pscx

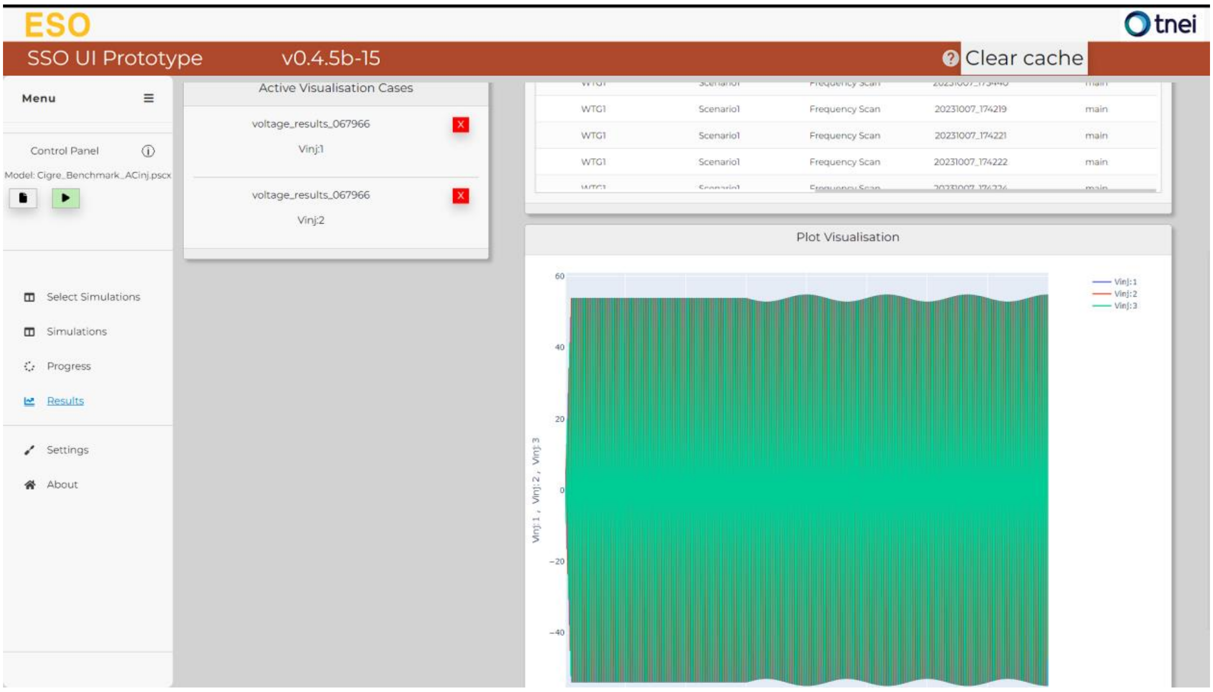
```
[EMT analysis - loading model]
[EMT analysis - loading scenarios]
[EMT analysis - running simulation]
[EMT analysis - saving results]
[EMT analysis - completed]

[Frequency scan - loading model]
[Frequency scan - loading scenarios]
[Frequency scan - running simulation]
```

6.2.5 Result visualisation and comparison

This feature enables users to visually compare simulation and identification process results through parallel plots. It supports interactive libraries such as Plotly for an enriched user experience, aiding users in exploring their data more effectively.

A screenshot of the 'Results' page from the UI. This page allows users to visualise both time domain and frequency domain results of simulations. The different scenarios can be explored in the meta table and then the user can either visualise a particular scenario or plot multiple scenario results in a single graph to compare the responses.



voltage_results_067966 Vinj:1

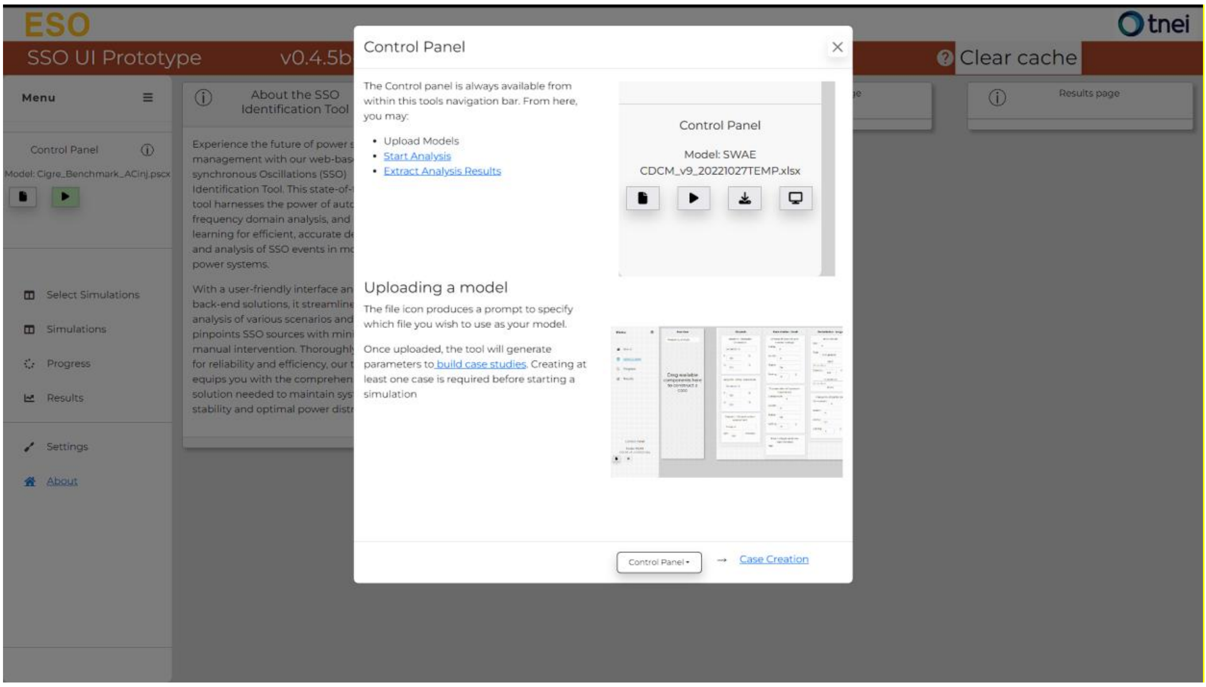
voltage_results_067966 Vinj:2

WTG1	Scenario1	Frequency Scan	20231007_174219	main
WTG1	Scenario1	Frequency Scan	20231007_174221	main
WTG1	Scenario1	Frequency Scan	20231007_174222	main

6.2.6 User assistance and support

This feature enhances the user experience with helpful tool tips, a bug-reporting mechanism, and access to system logs. Tool tips describe tool functionality and process outputs when the user hover over the respective inputs in the UI. The bug reporting mechanism allows users to flag issues for attention, while the log access provides visibility into the detailed workings of the SSO simulation and identification engine.

The tool includes a control panel which guides users interactively through a step-by-step process of using the different features of the tool. Screenshots from the control panel are included below.



Control Panel

The Control panel is always available from within this tools navigation bar. From here, you may:

- Upload Models
- Start Analysis
- Extract Analysis Results

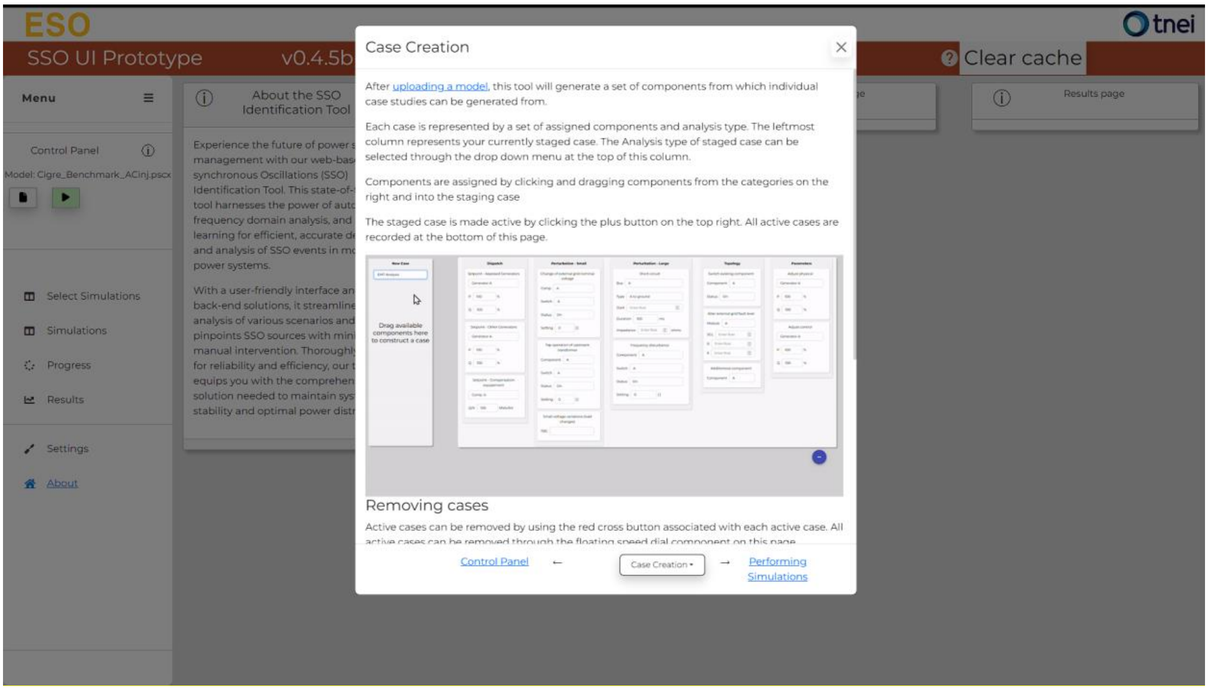
Control Panel

Model: SWAE CDCM_v9_20221027TEMP.xlsx

Uploading a model

The file icon produces a prompt to specify which file you wish to use as your model.

Once uploaded, the tool will generate parameters to build case studies. Creating at least one case is required before starting a simulation



Case Creation

After uploading a model, this tool will generate a set of components from which individual case studies can be generated from.

Each case is represented by a set of assigned components and analysis type. The leftmost column represents your currently staged case. The Analysis type of staged case can be selected through the drop down menu at the top of this column.

Components are assigned by clicking and dragging components from the categories on the right and into the staging case

The staged case is made active by clicking the plus button on the top right. All active cases are recorded at the bottom of this page.

Removing cases

Active cases can be removed by using the red cross button associated with each active case. All active cases can be removed through the floating speed dial component on this page

7 Feature engineering and ML classifier

The automatic detection of SSOs is a critical part of the process as this will increase engineering efficiency. This process is realised through a two-step process: 1) Feature engineering based on measurements and 2) the application of a machine learning classifier to distinguish between SSO and non-SSO events. Figure 7-1 depicts an overview of the automated SSO detection within the overall identification tool. The figure shows that EMT simulation results of SSO scenarios in the form of time domain signals will be fed to the detection process to identify any oscillations in the signal (more details in Figure 7-3). As explained below, the automated detection is based on signal processing and the algorithm can be used to detect presence of SSO in time-domain signals from any sources such as EMT simulation results or PMU data.

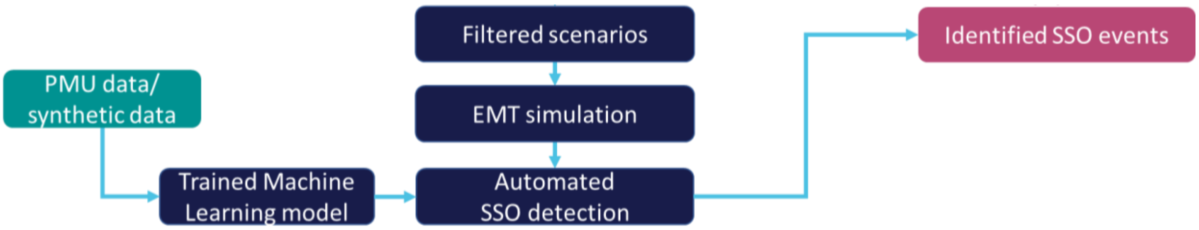


Figure 7-1 Overview of automated SSO detection.

This section provides an overview of these two crucial steps, emphasising the importance and detailing the process of feature computation and machine learning classification.

7.1 Feature Computation

The first step in the automatic detection of SSOs involves feature computation. This process is achieved by segmenting the signal into overlapping parts of a predetermined duration. This segmentation not only reduces the memory size occupied by the features but also increases resilience against fluctuations in the signal, thereby enhancing the efficiency and accuracy of the detection process. Four distinctive features are engineered based on the measurements, namely Feature 1, Feature 2, Feature 3, and Feature 4.

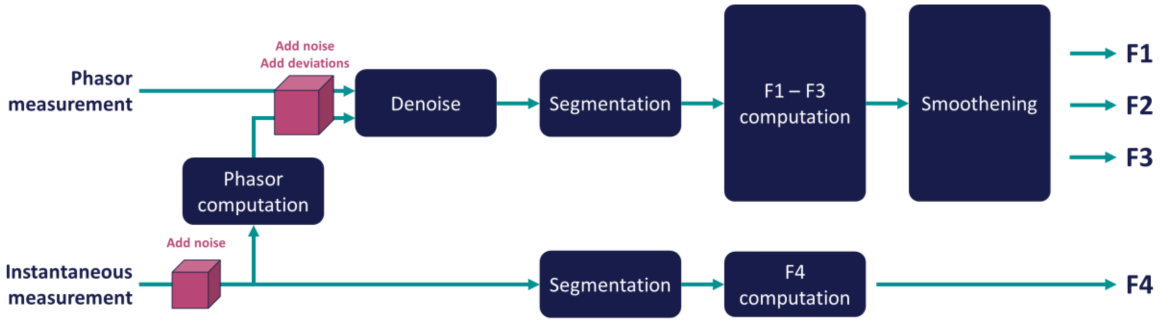


Figure 7-2 Feature engineering pipeline extended with perturbation sources.

Feature 1, or the **'Trend'**, is a key component in distinguishing oscillatory signals from non-SSO data due to its ability to identify periodicity. It decomposes a signal into regions of constant or increasing and decreasing magnitudes, looking for a repeating trend. However, it's crucial to use phasor measurements instead of time measurements to avoid misinterpretation of trends in current and voltage signals. To handle real-world data, improvements include checking sequences for points above a certain threshold to mitigate the impact of random noise fluctuations, using a peak detection and sequence-removing algorithm to address non-significant sequences around maxima and minima in the phasor, and adjusting the original method to accommodate non-integer multiples of the sampling period in the SSO.

Feature 2, also known as the '**Envelope Variation Index (EVI)'**, is designed to detect constant magnitude changes within the repetition period of SSO signals. It identifies variations in the signal's envelope by detecting peaks and valleys in a signal segment. The variance in these peaks and valleys, relative to the data variance, and their quantity, contribute to the value of Feature 2. In SSO signals, the phasor signal is periodic with a well-defined envelope, resulting in fewer peaks and valleys and less variance, hence a lower Feature 2 value. Non-SSO signals, on the other hand, have more peaks and valleys with significant variance, resulting in a higher Feature 2 value. Adjustments are made to handle real-world data, such as evaluating different scenarios to infer the value of Feature 2 when variances are undefined due to too-small segmentation.

The third feature, the' **Length of Stationary Subsequences (LSS)'**, measures the extent of change in adjacent amplitudes of the phasor. During Subsynchronous Oscillation (SSO) events, the amplitude changes significantly, resulting in fewer stationary subsequences and a lower value for this feature. In contrast, non-SSO measurements have more stationary subsequences due to minor changes in the phasor magnitude caused by noise, resulting in a higher feature value. The threshold for determining stationary subsequences is highly dependent on the sample rate and data range, requiring careful selection for each source.

The fourth feature, the '**Fourier Transform Index (FTI)'**, is computed based on the periodogram of a signal and is used to identify the presence of SSOs in a signal. The FTI is calculated by comparing the Power Spectral Density (PSD) of significant frequencies in the signal to the PSD of the fundamental grid frequency. For signals without SSOs, the FTI will be zero as there are no additional peaks in the frequency spectrum. However, for signals with SSOs, the FTI will be non-zero, with a higher value indicating more significant frequencies or higher PSDs. The computation of the FTI involves several steps, including the application of a windowing function to reduce spectral leakage, zero-padding to improve frequency estimation and the identification of significant frequencies based on a defined threshold. The FTI is robust against varying noise characteristics and is effective in indicating the presence of SSO events in a signal.

The first three features are computed on the phasor measurement, focusing on the trend, the volatility of the envelope, and the presence of stationary subsequences in the signal. These features exploit the inherent differences between oscillatory and non-oscillatory signals, thereby enhancing the ability to detect SSOs. The fourth feature, known as the Fourier Transform Index (FTI), is based on the frequency spectrum. The FTI provides a measure of how the energy of the off-nominal frequencies relates to that of the fundamental frequency, with a higher value indicating an SSO region.

7.2 Machine Learning Classification

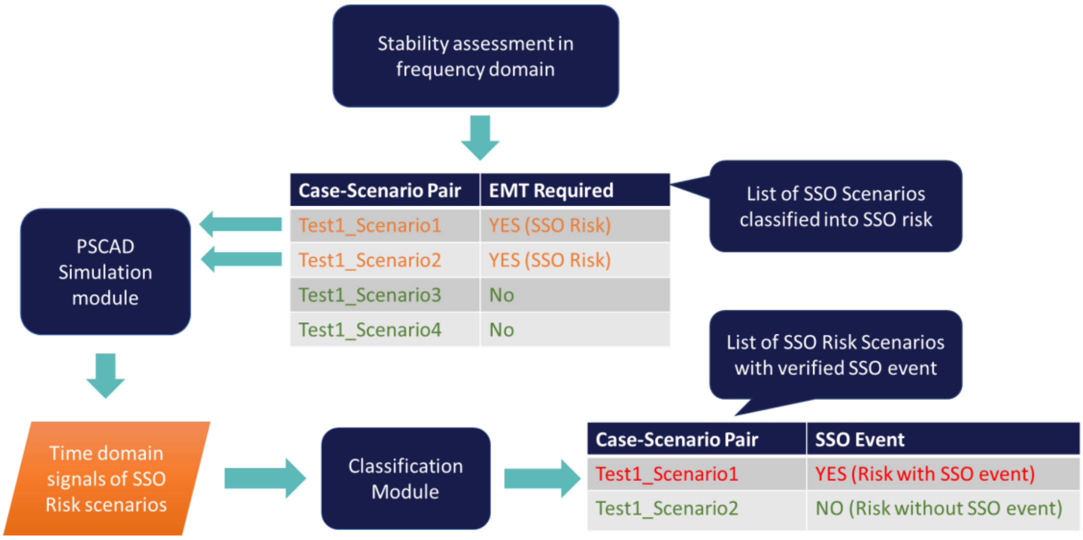
The second step in the detection process involves the application of a machine learning classifier. This classifier is used to distinguish between the feature values during SSO and non-SSO regions. The classifier's pipeline consists of three stages: scaling, classification, and voting. The scaling step is crucial as it ensures the proper functioning of the detector across different sources by normalising the output range of the features.

Two classifiers are implemented to handle the different durations of the segments. The first classifier, based on Support Vector Machines (SVMs), predicts using features 1 to 3. The second classifier, based on an optimised Random Forest (RF) structure, works on feature 4. Both classifiers demonstrate outstanding performance in predicting SSOs on both instantaneous and phasor measurements, thereby meeting the requirements for scalability and explainability.

7.3 Structure of the Classification Module in the tool

The classification module within the tool is designed to execute the entire classification pipeline for predicting SSO events, which includes feature extraction, constructing the sample-feature matrix, training the classifier, and predicting SSO events on a new measurement.

The classification module is initiated by reading the results from the PSCAD Automation run. This process reads from a CSV file that indicates if EMT analysis is required for classification. The CSV file contains two columns: 'Case_scenario pair' and 'EMT Required' and contains information related to all SSO scenarios (case_scenario pairs) under study. If a case_scenario has been identified as SSO risk, it will be flagged as EMT_Required=True. If the corresponding row for 'EMT Required' is 'YES', then an EMT simulation will be run for the specific case_scenario pair. The classification module takes the simulation results from the case scenario pair and process the corresponding time-domain signals to verify if the signals indeed have an SSO event. At the end of the process, a CSV file with a list of SSO risk scenarios is generated indicating any identified SSO events.


Stability assessment in frequency domain
List of SSO Scenarios classified into SSO risk
PSCAD Simulation module
List of SSO Risk Scenarios with verified SSO event
Time domain signals of SSO Risk scenarios
Classification Module

Case-Scenario Pair	EMT Required
Test1_Scenario1	YES (SSO Risk)
Test1_Scenario2	YES (SSO Risk)
Test1_Scenario3	No
Test1_Scenario4	No

Case-Scenario Pair	SSO Event
Test1_Scenario1	YES (Risk with SSO event)
Test1_Scenario2	NO (Risk without SSO event)

Figure 7-3 Classification module integration with the SSO Identification Tool.

The classification module is implemented using Python and is structured into four main steps:

- Step 1 - Feature Extraction: The initial step of the classification module is feature extraction, which is crucial for determining optimal parameters. The objective here is to adjust the parameters so that the features yield distinct values during periods of SSO and non-SSO events. To achieve this, the feature extractor class is instantiated, and the *run_feature_computations* function is invoked. This process ensures that the features are accurately extracted, setting the foundation for the subsequent steps in the classification module.
- Step 2 - Construct the Sample-Feature Matrix: The second step involves constructing the sample-feature matrix and the label vector. In this phase, features are computed for all files in the provided data folder, using either default settings or user-specified parameters. The same settings that yielded the best results in the feature extraction

phase are employed here. The construction of the sample-feature matrix and label vector is a pivotal step as it prepares the data for the training of the classifier.

- Step 3 - Train the Classifier: The third step is dedicated to training the classifier. Once the features have been extracted and the sample-feature matrix constructed, the classifier can be trained using this data. The classifier is created, and the data is inspected. The classifier is then fine-tuned and fitted to the data. This step is crucial as it ensures that the classifier is optimally trained to distinguish between SSO and non-SSO events, thereby enhancing the accuracy of the predictions in the final step.
- Step 4 - Load a New Measurement and Predict: The final stage of the classification module involves applying the trained classifier to new data for SSO prediction. Features are computed on different segments of the new measurement, mirroring the process undertaken during training. Each segment is then individually assessed by the classifier to predict the occurrence of an SSO event.

A voting mechanism is subsequently implemented to derive a conclusive label for the entire measurement, based on the number of segments predicted as SSO events. This final prediction indicates whether an SSO event is likely to have occurred, showcasing the classification module's capability to accurately apply the trained classifier to new data. A detailed discussion on the verification/validation of the machine learning classifier was included in Section 8.5 of the WP1 report [1].

8 Software Testing

Software testing is a critical process that ensures the quality and reliability of software applications. It involves systematically examining software components and functionalities to identify defects and errors. The importance of software testing lies in its ability to minimize risks, enhance user experience, and improve overall software quality. The advantages of software testing include early defect detection and validation of software requirements. However, testing can be time-consuming and resource-intensive, and it cannot guarantee the absence of all defects. A balanced approach that combines rigorous testing with effective development practices is necessary to achieve optimal results.

V-model for software development and testing (see Figure 8-1) establishes a strong connection between software testing and the different stages of software development. It emphasizes the parallelism between development activities and their corresponding testing activities, ensuring that testing is integrated throughout the entire software development life cycle (SDLC).

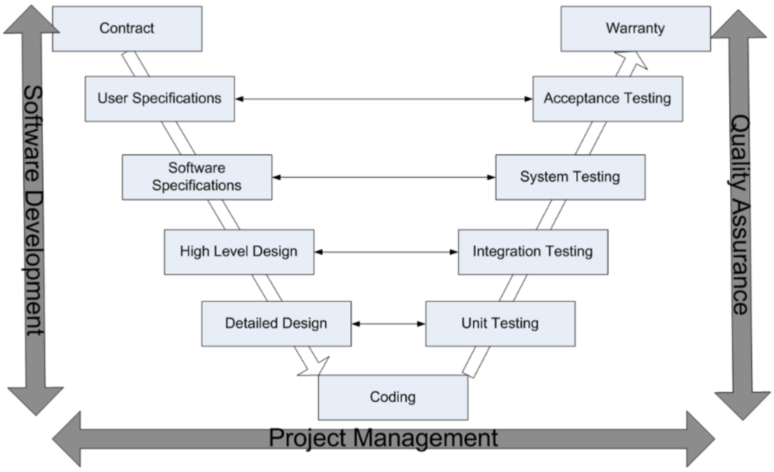


Figure 8-1 V-model for software development and testing.

The V-model highlights the importance of early and continuous testing by mapping testing activities to each development stage. As the development progresses from requirements gathering and analysis to system design, module design, and coding, the corresponding testing activities move from test planning to test design, test execution, and verification/validation.

By following the V-model, software testing becomes an integral part of the development process, enabling early defect detection and addressing issues at their root. The V-model's structured approach ensures that testing is aligned with the defined requirements and specifications, facilitating better risk management and quality assurance. It promotes collaboration between developers and testers, as both parties are involved throughout the process, leading to improved communication and a shared understanding of the software's functionality and quality objectives.

Overall, the V-model strengthens the connection between software testing and development by emphasizing their parallel nature and promoting the integration of testing activities across the SDLC. This approach helps ensure that software testing is not an isolated activity but an integral part of the development process, contributing to the delivery of high-quality software products.

8.1 Unit testing

Unit testing is a software testing technique that focuses on evaluating the individual components or units of a software application in isolation. It involves testing the smallest testable parts, such as functions, methods, or classes, to ensure their correct behaviour and functionality. Unit tests are typically written by developers and executed frequently during the development process to detect defects or errors within the units themselves and to verify that they produce the expected outputs when provided with specific inputs. Unit testing helps improve code quality, facilitates early bug detection, supports refactoring efforts, and promotes modular and maintainable software design.

8.2 Integration Testing

Integration testing is a software testing approach that validates the interactions and interfaces between different components or modules of a software system. It focuses on testing the integration points where the individual units come together to form a cohesive system. Integration testing aims to uncover defects or issues that may arise due to the interactions between these components, such as data inconsistencies, communication failures, or functionality conflicts. It ensures that the integrated components work correctly together and that the system functions as intended. Integration testing plays a crucial role in identifying and resolving integration-related problems early in the software development life cycle, ensuring a smooth and seamless integration of various components and minimizing the risk of potential failures in the production environment.

8.3 System Testing

System testing is a comprehensive software testing process that assesses the behaviour and functionality of an integrated software system. It aims to validate that the entire system meets the specified requirements and performs as expected in different real-world scenarios. System testing covers end-to-end testing, where the system is tested in a simulated production-like environment, including all components, interfaces, and external dependencies. It focuses on evaluating the system's functional and non-functional aspects, such as performance, security, reliability, and usability. System testing helps ensure that the software system meets the desired quality standards and is ready for deployment, providing confidence to stakeholders that it will meet user needs and expectations.

8.4 Acceptance testing

Acceptance testing is the final phase of software testing, typically performed by end-users or stakeholders, to determine whether a software system meets the specified requirements and is ready for deployment. The primary purpose of acceptance testing is to validate that the system satisfies user needs and expectations and can be accepted for actual use. It focuses on verifying that the software functions correctly in real-world scenarios, meets business requirements, and aligns with the intended use cases. Acceptance testing ensures that the software system is reliable, usable, and fulfils the desired functionality from the perspective of the end-users. This testing phase acts as a crucial gatekeeper before the software is deployed, allowing stakeholders to make an informed decision regarding its acceptance and readiness for production use.

9 Conclusions

In conclusion, this project report has presented the development and implementation of an SSO identification tool, which aims to address the challenges associated with detecting and analysing sub-synchronous oscillations in power systems. Through the exploration of frequency domain methods, including impedance scan methods and the grey box approach, the tool provides valuable insights into the dynamic behaviour and impedance characteristics of the system. The integration of the CIGRE Benchmark and PSCAD model further enhances the tool's capabilities, allowing for accurate frequency scanning and modelling efforts. The user interface and back-end deployment solutions offer a user-friendly experience, enabling efficient scenario selection, parameter configuration, simulation execution, and result visualization. Leveraging feature engineering and machine learning classification, the tool provides advanced techniques for identifying SSO phenomena and enhancing the understanding of power system stability. The comprehensive software testing ensures the reliability and accuracy of the tool. Overall, this project contributes to the field of SSO analysis by providing an effective and practical solution for power system engineers and operators to detect, analyse, and mitigate sub-synchronous oscillations, thereby improving the stability and reliability of power systems.

10 References

[1] T. Services, "WP1 Report," TNEI Services, Manchester, 2023.

[2] A. Rygg, M. Molinas, C. Zhang and X. Cai, "A Modified Sequence-Domain Impedar Definition and Its Equivalence to the dq-Domain Impedance Definition for the Stabi Analysis of AC Power Electronic Systems," *IEEE Journal of Emerging and Selected Top in Power Electronics,* p. 1383–1396, 2016.

[3] S. Shah, P. Koralewicz, V. Gevorgian and R. Wallen, "Sequence Impedance Measureme of Utility-Scale Wind Turbines and Inverters - Reference Frame, Frequency Coupli and MIMO/SISO Forms," *IEEE Transactions on Energy Conversion,* p. 75–86, 2022.

[4] J. Sun, "Impedance-based stability criterion for grid-connected inverters," *I Transactions on Power Electronics,* p. 3075–3078, 2011.

[5] W. Ren and E. Larsen, "A Refined Frequency Scan Approach to Sub-Synchronous Cont Interaction (SSCI) Study of Wind Farms," *IEEE Transactions on Power Systems,* p. 390 3912, 2016.

[6] S. Shah and L. Parsa, "Impedance Modeling of Three-Phase Voltage Source Convert in DQ, Sequence, and Phasor Domains," *IEEE Transactions on Energy Conversion,* p. 113 1150, 2017.

[7] Y. Cheng, M. Sahni, D. Muthumuni and B. Badrzadeh, "Reactance scan crossover-bas approach for investigating SSCI concerns for DFIG-based wind turbines," *I Transactions on Power Delivery,* p. 742–751, 2013.

[8] Y. Zhu, Y. Gu, Y. Li and T. C. Green, "Participation Analysis in Impedance Models: The Gre Box Approach for Power System Stability," *IEEE Transactions on Power Systems,* vol. no. 1.

[9] Y. Zhu, Y. Gu, Y. Li and T. C. Green, "Impedance-Based Root-Cause Analysis: Comparat Study of Impedance Models and Calculation of Eigenvalue Sensitivity," *IEEE Transactic on Power Systems,* vol. 38, no. 2.

[10] B. G. a. A. Semlyen, “Rational approximation of frequency domain responses by vec fitting,” *IEEE Trans. Power Delivery,* vol. 14, no. 3, July 1999.

[11] A. Arsenovic, “scikit-rf: An Open Source Python Package for Microwave Network Creati Analysis, and Calibration [Speaker’s Corner],,” *IEEE Microwave Magazine,* vol. 23, nc Jan 2022.

[12] B. Guvstavsen, “The Vector Fitting Website,” [Online]. Availak www.sintef.no/projectweb/vectorfitting.

[13] H. D. S. B. Salarieh, “Review and comparison of frequency-domain curve-fitti techniques: Vector fitting, frequency-partitioning fitting, matrix pencil method a loewner matrix,” *Electric Power Systems Research.*

[14] M. Sahni, D. Muthumuni, B. Badrzadeh, A. Gole and A. Kulkarni, “Advanced Screeni Techniques for Sub-Synchronous Interaction in Wind Farms,” in *PES T&D*, Orlando, 20

[15] M. Sahni, B. Badrzadeh, D. Muthumuni, Y. Cheng, H. Yin, S.-H. Huang and Y. Zhou, “Su synchronous Interaction in Wind Power Plants - Part II: An ERCOT Case Study,” in *2 IEEE Power and Energy Society General Meeting*, San Diego, 2012.

[16] B. Badrzadeh, M. Sahni, D. Muthumuni, Y. Zhou and A. Gole, “Sub-synchronous Interacti in Wind Power Plants - Part I: Study Tools and Techniques,” in *2012 IEEE Power and Ener Society General Meeting*, San Diego, 2012.

[17] B. Badrzadeh, M. Sahni, Y. Zhou, D. Muthumuni and A. Gole, “General Methodology Analysis of Sub-Synchronous Interaction in Wind Power Plants,” *IEEE Transactions Power Systems,* vol. 28, no. 2, pp. 1858-1869, 2013.

[18] X. Jiang and A. Gole, “A Frequency Scanning Method for the Identification of Harmo Instabilities in HVDC Systems,” *IEEE Transactions on Power Delivery,* vol. 10, no. 4, pp. 187 1881, 1995.

[19] B.-G. R. B. C. K. J. L. M. S. Y. W. X. Kocewiak L, “Practical Aspects of Small-signal Stabi Analysis and Instability Mitigation,” 2022.

[20] B.-G. R. B. C. K. J. L. M. T. A. S. Y. W. X. Kocewiak Ł, “Instability mitigation methods in mode converter-based power systems,” *20th International Workshop on Large-Sc*

Integration of Wind Power into Power Systems as well as on Transmission Networks Offshore Wind Power Plants (WIW 2021), pp. 464-480, 2021.

[21] Manitoba Hydro International Ltd., "Sub Synchronous Oscillations SSO Investigation a Applications," Manitoba Hydro International Ltd., 2016 November 2016. [Onlir Available: https://www.youtube.com/watch?v=d6FZxZYTLg8&t=2282s. [Accessed September 2022].

[22] R. N. Damas, Y. Son, M. Yoon, S.-Y. Kim and S. Choi, "Subsynchronous Oscillation a Advanced Analysis: A Review," *IEEE Access,* vol. B, pp. 224020-224032, 2020.

[23] D. Kidd and P. Hassink, "Transmission Operator Perspective of Sub-Synchronc Interaction," in *PES T&D*, Orlando, 2012.

[24] A. Ostadi, A. Yazdani and R. Varma, "Modeling and Stability Analysis of a DFIG-Bas Wind-Power Generator Interfaced With a Series-Compensated Line," *IEEE Transactic on Power Delivery,* vol. 24, no. 3, pp. 1504-1514, 2009.

[25] X. Xie, X. Zhang, H. Liu, Y. Li and C. Zhang, "Characteristic Analysis of Subsynchronc Resonance in Practical Wind Farms Connected to Series-Compensat Transmissions," *IEEE Transactions on Energy Conversion,* vol. 32, no. 3, pp. 1117-1126, 2(

[26] M. Beza and M. Bongiorno, "On the Risk for Subsynchronous Control Interaction in Ty 4 Based Wind Farms," *IEEE Transactions on Sustainable Energy,* vol. 10, no. 3, pp. 14 1418, 2019.

Appendix A: Cigre benchmark model

CIGRE WG on *Multi-frequency stability of converter-based modern power systems* has developed a benchmark system to study the SSO phenomenon. In [19] [20] , various methods are described that can identify and possibly mitigate frequency instabilities in converter-based power systems. The CIGRE benchmark model is used in MATLAB Simulink to investigate and validate the effect of the proposed methods.

Overview

The benchmark system consists of two identical GFL Wind Turbine Generator (WTG) strings, and inverters connect both to the grid. The grid is modelled as a voltage source with a grid impedance. In [19] [20], various methods are described that can identify and possibly mitigate frequency instabilities in converter-based power systems. A test bench was created to investigate and validate the effect of the proposed methods. The system is depicted in Figure A1.

The testbed is at its "stability boundary, i.e., the system is poorly damped, and even small parameter variations can lead to instability" [20]. This is also seen from the eigenvalue analysis, showing multiple undamped oscillatory modes (Figure A.3). Note that the modes are not exclusively from the SSO range but also include interactions in the harmonic range.

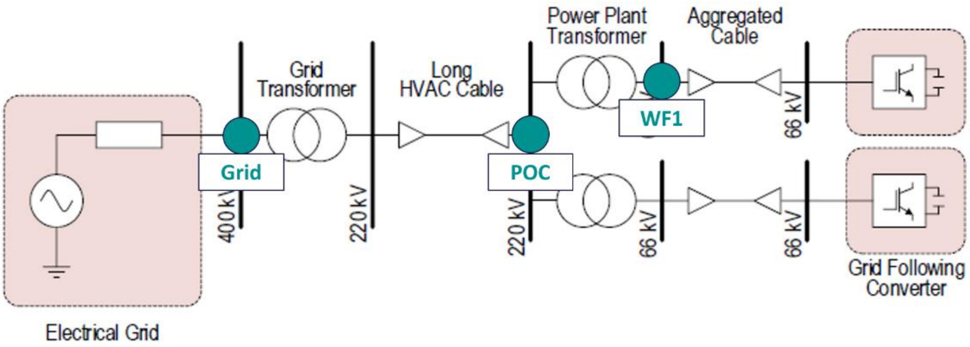


Figure A. 1 Simple representation of the CIGRE test bench. Image from [20].

The authors list the grid's Short-Circuit Ratio (SCR) as the main contributor to the SSO. The damping of the DC voltage control loops in the aggregated WTG strings and the Phase Lock Loop (PLL) decreases with the SCR and becomes unstable below $S_n \leq 1100$ MVA and $S_n \leq 750$ MVA respectively [20]. This means that by varying S_n, different scenarios with oscillations can be created.

Steps are applied in the grid voltage, AC and DC voltage references of the inverters, and the DC side's power to excite these oscillations. Measurements are recorded at three points (see Figure A.1): one WTG string (WF1), the point of coupling of the strings (POC) and the grid side (Grid). Moreover, both the current and voltage are measured. This is done to test the features on different signals at different points in the system where the SSOs might be more or less prominent. This means that six measurements are recorded per scenario.

General controller architecture

Both wind farm models use PLL to calculate the angle used for the dq0-transformation.

Also, in the first case, the (AC) voltage is not regulated in a separate block. In the second case, a separate PI controller exists for this purpose. An overview of the grid-following controller is seen in Figure A2.

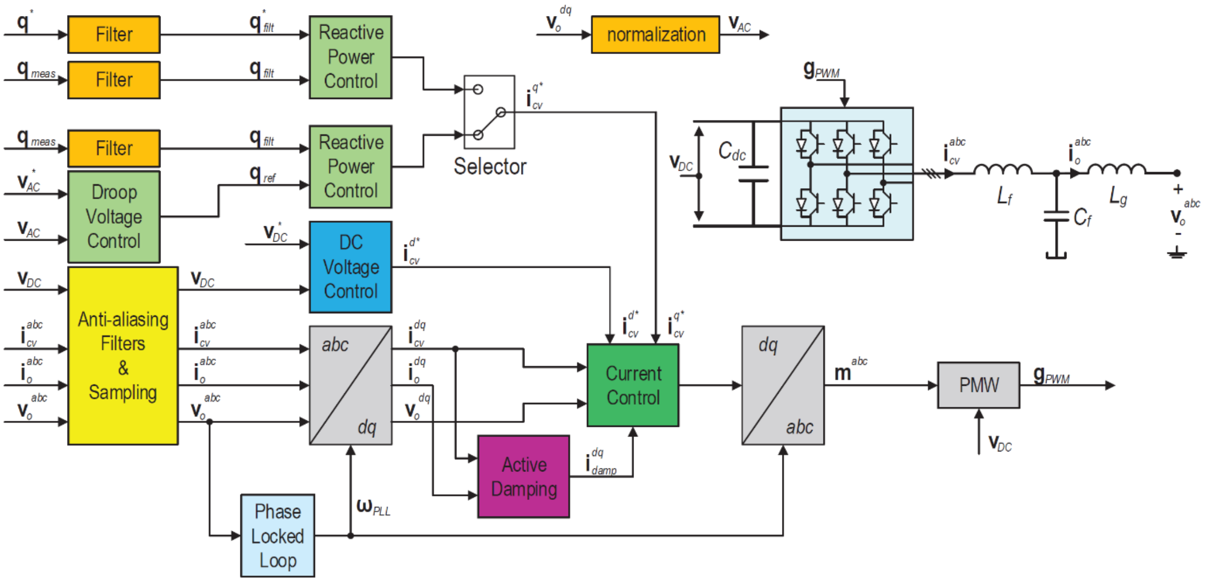


Figure A. 2 The implemented grid-following controller of the first case. Image from [19].

The reactive power control loop

The reactive power setpoint is computed using the voltage droop controller and the measured reactive power in both models.

The method is not important for the creation of oscillations. However, case 1 allows defining the setpoints from a MATLAB script, while in case 2, the setpoints must be set inside Simulink. This makes case 2 unattractive for the simulation of multiple scenarios.

The current control loop

The current control loop is used to implement the desired power outputs (setpoints in i_{dq}) in the system. These setpoints are computed from the AC-voltage droop controller and the DC-voltage (1st model) or directly from PQ setpoints passed through a voltage regulator (2nd model).

The current control loop in the first model has a variable gain K_{ffv} which can be changed to move the poles into a stable region (see the section below). In the second case, this parameter is absent, meaning an equivalent gain of $K_{ffv} = 1$. This should result in unstable operation.

Also, the difference between the measured current and the current setpoint is handled differently. In the first case, it is passed through a PI controller; in the second case, a P controller is used.

Moreover, in the first model, the difference in current over the transformer is passed through the active damping gain and used as well in the computation of the reference *dq*-voltage. In the second model, this is done only over i_{cv} (the current output at the WTG string).

Finally, a decoupled i_{cv} is used in the first model to compensate for the interaction between the axes. If properly implemented, this should reduce the effect that a change on one axis has on the other axis by predicting the interaction and subtracting it from the other. Note: this only works well under balanced conditions. If the system is imbalanced, some coupling will occur. This is not done in the second model.

The output stage

An output stage converts the computed dq0-voltage setpoint into an AC voltage by computing a duty cycle m for the PWM modulator.

The DC power control loop

In the first model, the DC side is implemented with a variable power output which is given by a step function passed through the transfer function $H(s) = \frac{1}{0.5s+1}$ and converted back into a current setpoint. In the second model, the DC side is modelled as a constant voltage source. This difference in implementation means changing the DC side power is only possible in the first model.

Oscillatory modes

The testbed is at its *"stability boundary, i.e. the system is poorly damped, and even small parameter variations can lead to instability"* [20] . This is also seen from the eigenvalue analysis, showing multiple undamped oscillatory modes (see Figure A. 3). There are a couple of unstable high-frequency modes and some low-frequency modes. The latter is in the SSO range.

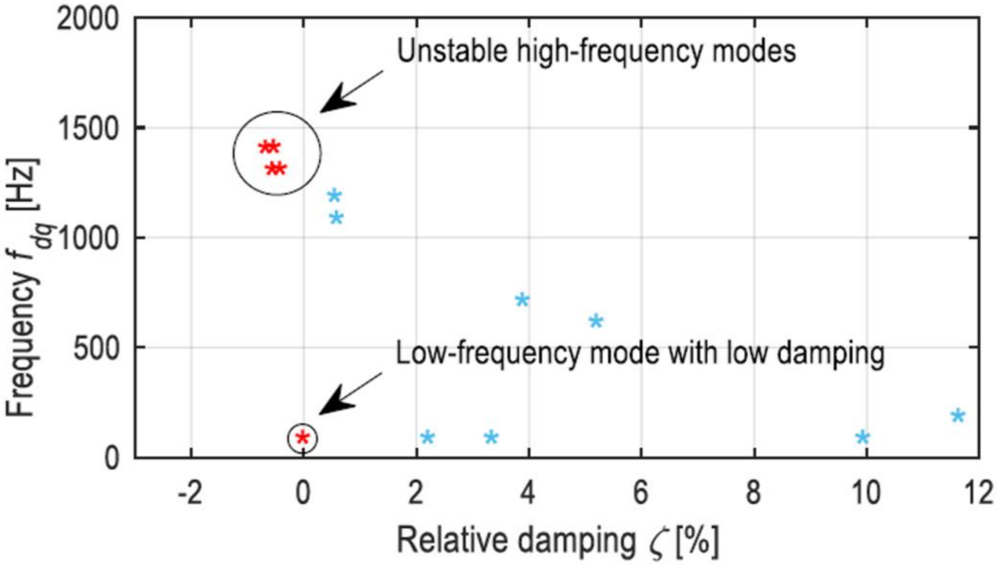


Figure A. 3 Eigenvalue analysis for the base case configuration of the CIGRE test bench. Image from [20] .

It is possible to mitigate some of these modes by changing system parameters. The current control gain K_{pc} is located inside the PI controller of the current controller of the control system. It multiplies the difference between the reference current from the power control block and the actual output current with this gain to find (part of) the output *dq*-reference voltage. This means that this part ensures that the reference setpoint of the power control loop is satisfied. This parameter has a limited effect on the oscillatory modes.

The voltage feedforward gain K_{ffv} is located inside the current controller of the control system. It multiplies the voltage at the output of the WF (after the transformer) with this gain to calculate the output *dq*-reference voltage. This term is used to ensure the desired voltage after the transformer. Decreasing this parameter from 1 to 0.7 p.u. damps the oscillations at 1271 and 1370 Hz (see Figure A. 4).

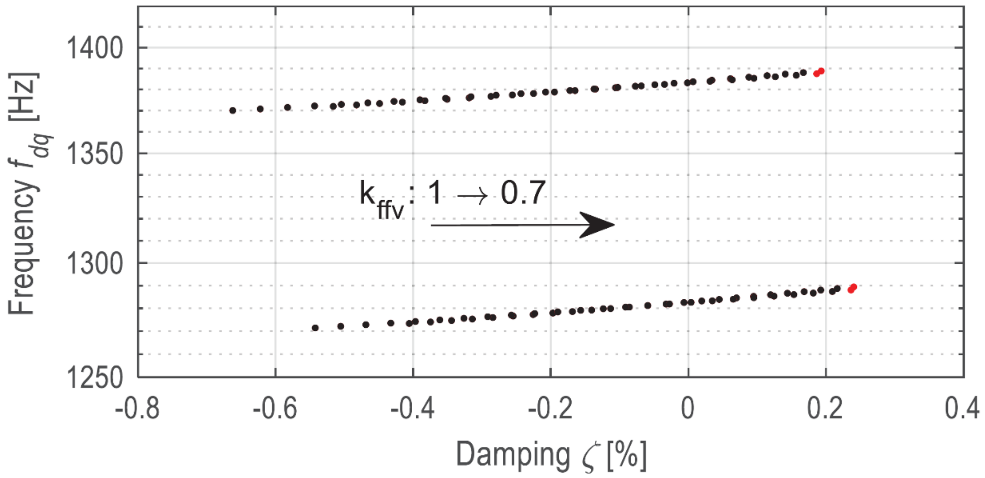


Figure A. 4 Parameter sweep of $\boldsymbol{K_{ffv}}$. Image from [20] .

The active damping gain K_{AD} is located inside the current controller of the control system. It multiplies the difference in current over the transformer with this gain to find (part of) the output dq-reference voltage. This term is used to filter the harmonics across the transformer actively. Increasing this parameter stabilises the higher-order oscillatory modes. Increasing this parameter from 0.2 to 0.5 p.u. successfully damps the oscillatory modes at 1271, 1273, 1370 and 1372 Hz (see Figure A. 5).

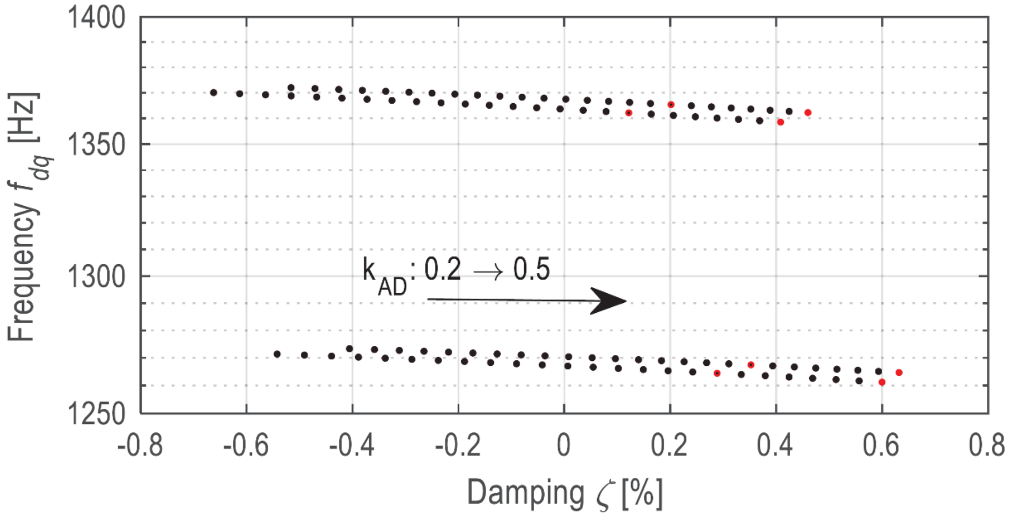


Figure A. 5 Parameter sweep of $\boldsymbol{K_{AD}}$. Image from [20] .

The strength of the grid influences the oscillatory modes in the SSO range. A weak grid, e.g., $S_n = 600$ MVA, has a Short-Circuit Ratio (SCR) of 1.43. On the other hand, a strong grid with $S_n = 3000$ MVA has an SCR of 7.14.

In Figure A. 6, the impact of the SCR on the lower frequency modes is seen. The DC voltage control loop causes the top SSO, while the PLL is responsible for the lower mode [20]. The modes become damped when the SCR increases.

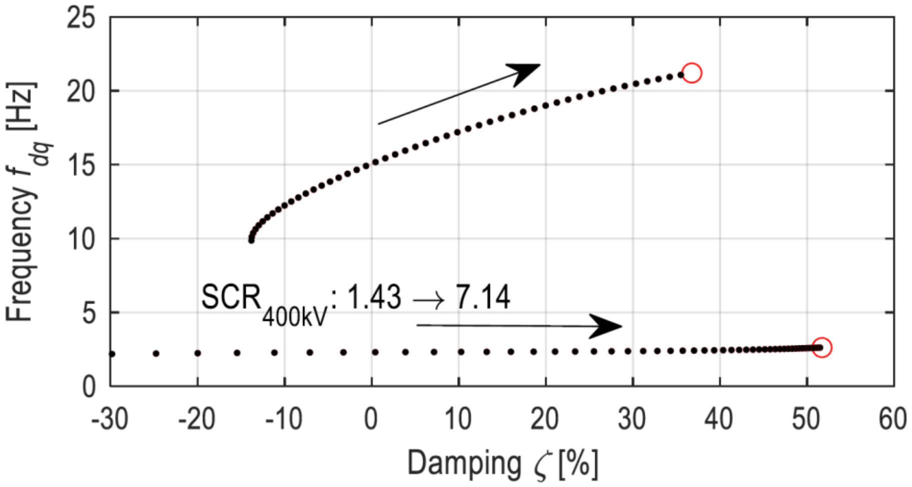


Figure A. 6 Parameter sweep of the grid's SCR. Image from [20].

Unfortunately, changing the control parameters did not seem to reduce the oscillations, as suggested in [19] . Possibly, the sweep was done on a different model. Alternatively, the effect of varying the parameters is derived from the system's state-space representation without it being reflected by the model. The model produces both subsynchronous and harmonic modes and it is not possible to create only subsynchronous modes in the system. Since we are focusing on SSO only, we filtered the time domain results to keep only SSO frequencies during the machine learning model training. A low-pass filter could be applied to isolate the low frequencies from the measurements.

Changing the SCR did show the expected results, showing no oscillations at high SCRs (Figure A. 77), while the spectrum is full of them at lower SCRs. The spectrogram diagrams are showing the frequency content of a time domain signal when a window of a certain size is moved along the signal. The x-axis is showing the different frequencies, and the y-axis is showing the presence of these frequencies at different time instance of the signal.

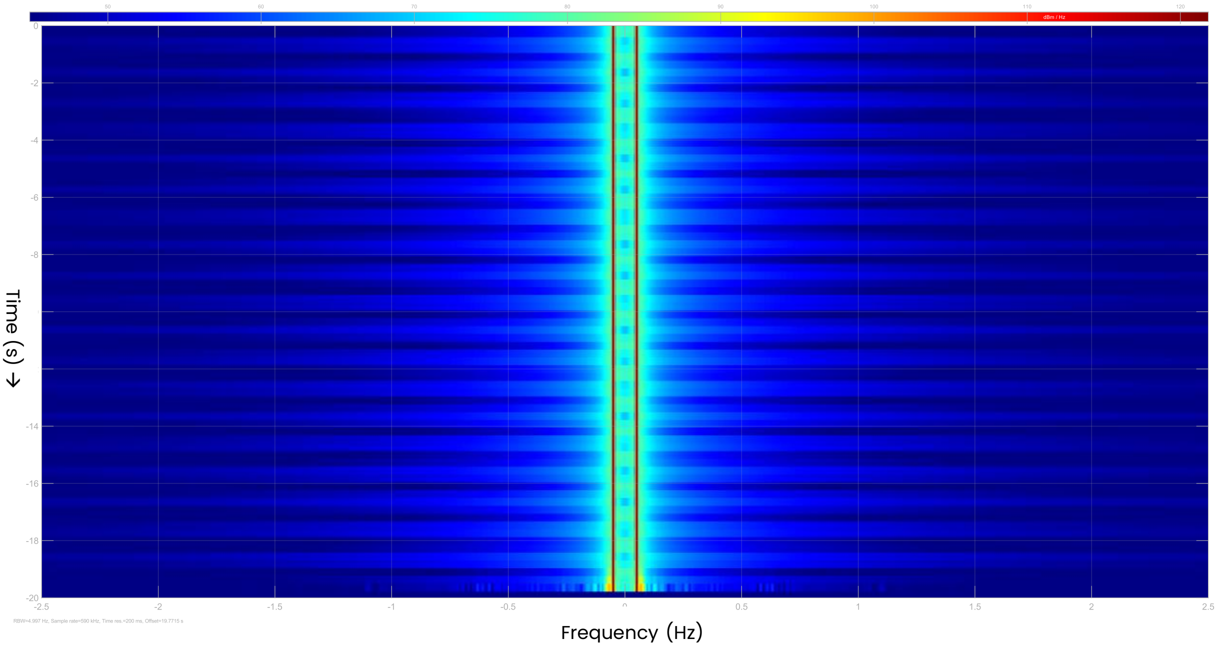


Figure A. 7 Spectrogram of the voltage at POC for $S_n = 3000\ MVA$

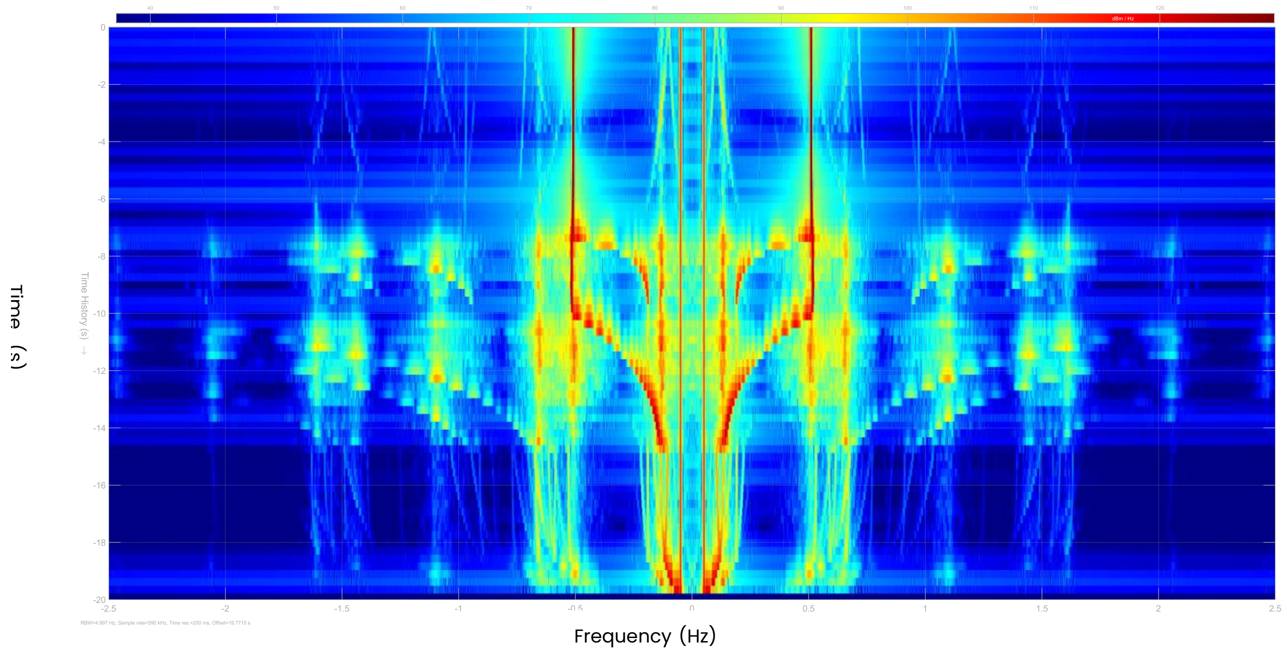


Figure A. 8 Spectrogram of the voltage at POC for $S_n = 900\ MVA$.

Events

There are several events added to the test bench which excite the oscillations. Initially, the system is marginally stable, and events require the network to move towards a different state. Because the system is hardly stable, even small perturbations are sufficient to induce oscillations.

There are four possible events: a step change in the grid voltage, WF power output change, AC voltage reference change and DC voltage reference change. Their meaning is discussed below.

Step in the grid voltage

A step in the grid voltage indicates an event on the grid side. The grid-following converters will try to follow this voltage level.

Step in the WF power

The power output of the WF can change. This change can be natural when the availability of wind causes it. However, the grid operator can also request an amount of active power to be delivered. Some turbines may be turned off, and the setpoint changed for the other turbines to satisfy the operational point. The mechanical power also needs to be adjusted to reflect this new setpoint. Because the pitch control is slow to react, there is a temporary mismatch causing a change in voltage. The converters must cope with the increased voltage level; if not, oscillations will occur.

In the test bench, it is possible to change the active power output of each wind farm individually.

Step change in AC voltage reference

A step in the AC voltage reference is effectively a step in the reactive power setpoint. A change in this setpoint could be caused by a transformer tap action, as required by a contingency, or to minimise transmission losses. .

Step in the DC voltage reference

When a fault occurs on the AC side, the power output drops, but DC power persists because of the mechanical input (wind). The DC power is ramped down but slower, meaning that there is much more power produced than there can be delivered to the grid. This energy needs to go somewhere and results in the capacitors being charged, causing the DC voltage to go up. This is simulated by changing the DC voltage reference. This increased level persists until a crowbar is connected to discharge the system.

Default events

The oscillations in the test bench are excited by chaining all events together.

First, a grid step of 0.02 p.u. is applied at 1 second. Next, the WF MW outputs are increased from 0 to 1 p.u. at 5 seconds for WF1 and at 10 seconds for WF2, respectively. This corresponds to increasing the power output of each string to its nominal power. After that, the AC voltage reference is decreased from 1.05 to 1.03 p.u. because some tap changes are needed for power transport. This happens 12 seconds into the simulation. Finally, at 15 seconds, the DC reference voltage reference is reduced from 2 to 1.98 p.u. This is because the new active and reactive power setpoints cause a temporary mismatch between the requested and delivered power.

One of the resulting spectrograms is seen in Figure A. 88. It is possible to see how the various events excite the modes differently.

Appendix B: Review of impedance scan methods

1. A Modified Sequence-Domain Impedance Definition and Its Equivalence to the dq-Domain Impedance Definition for the Stability Analysis of AC Power Electronic Systems [2]

- The main contribution of the paper is a method to determine the sequence impedance of a Source-Load system, enabling the identification of frequency coupling in the sequence domain.
- Frequency coupling has an impact on the stability performance of the Source-Load system. **A system with frequency coupling will be less stable in comparison with a mirror frequency decoupled (MFD) system**.
- Sources of frequency coupling in IBRs are PLL, asymmetrical implementation of dq axis controllers, DC-bus voltage control, and active and reactive power controllers.
- Sources of frequency coupling on synchronous machines are rotor saliency, poorly tuned PSSs, and frequency and voltage control loops.
- The eigenvalues of the d-q impedance matrix are equal to those obtained by the modified sequence domain impedance (MSDI) matrix.

Key features of the reviewed method

- The method relies upon the definition of the modified sequence domain impedance (MSDI) matrix, which uses the relation between phasors in the dq domain with the corresponding phasors in the sequence domain.
- The off-diagonal components of the MSDI matrix Z_{pn} (below) allows the identification of the influence of a positive sequence current on the negative sequence voltage, and vice versa.
- The eigenvalues of the dq impedance matrix are equal to the MSDI matrix.
- Figure B. 1 shows the circuit equivalents of the impedance matrices in both domains. If a system is mirror frequency decoupled the off-diagonal components of $\boldsymbol{Z_{pn}}$are zeros; the current dependant voltage sources in the sequence domain are not necessary.

- Simulation results showed that systems which have mirror frequency coupling have less stability margin in comparison with those mirror frequency decoupled. Sources of frequency coupling are PLL; converter current controllers with unequal structure of parameter values in the d and q axes; dc-link voltage control systems; active and reactive power controllers; and salient poles synchronous machines.
- A drawback is that the authors did not specify the selection of the perturbation signals. They only mentioned that the two perturbation signals must be linearly independent; for that, the perturbations defined in dq domain must have different frequencies.
- Another drawback is the lack of explanation of the stability analysis process. There is no clear procedure to process the MIMO system (2x2 sequence impedance matrix), to produce the stability analysis (multiple Nyquist plots) and the application of the generalised Nyquist criteria.

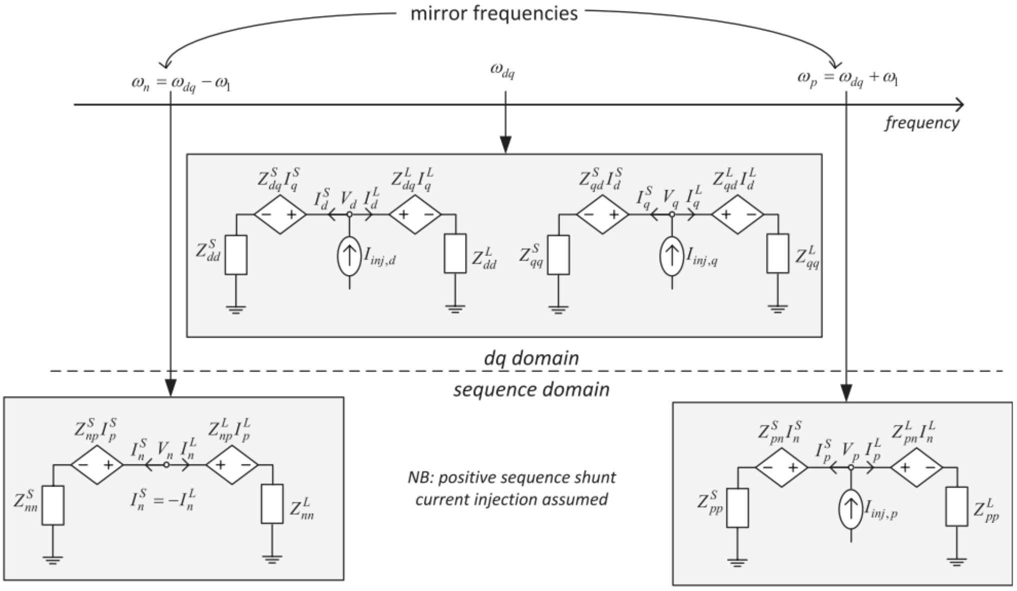


Figure B. 1 Circuit representation of frequency coupling using dq and sequence domain.

Summary of the proposed method

1. The method requires the injection of two test signals in abc domain.
2. Storing the time-series response (voltage and current) of source and load subsystems.
3. Transform the stored signals to dq domain and perform FFT.
4. Extract the magnitude of the components at the frequency of interest.
5. Calculate the dq impedance matrix.

6. Obtain the sequence impedance matrix using *dq* to *abc* transformation.

2. Sequence Impedance Measurement of Utility-Scale Wind Turbines and Inverters - Reference Frame, Frequency Coupling, and MIMO/SISO Forms [3]

- Active devices like IBRs have a frequency-coupled response due to their configuration; in practical terms, an injection of a positive-sequence voltage perturbation produces a negative-sequence current, in addition to the positive-sequence response. Therefore, the off-diagonal elements of a matrix transfer function (impedance or admittance) will likely be not zero for IBRs.
- The frequency coupling is significant at low frequencies, below a couple of hundred Hertz; therefore, the SSO analysis must be considered in the screening method.
- The sequence impedance of active devices has a reference frame. To ensure the phase alignment with the phase-A voltage, a correction of the Fourier components was introduced. The correction aligns the starting point of the data window of the FFT with the fundamental component of phase A voltage.

Key features of the reviewed method

- The method enables the computation of the sequence domain admittance matrix containing frequency coupling components (off-diagonal terms).
- The 2x2 matrix is a MIMO system that can be reduced to a SISO system. If the impedance from the grid side Z_g (see Figure B. 2)does not show frequency coupling, it is possible to reduce the system to use Nyquist and Bode plots to study the stability of the system.
- The method provides a compensation procedure to align the FFT components of the data corresponding to the measurements, with the peak of the fundamental component of the phase A voltage.

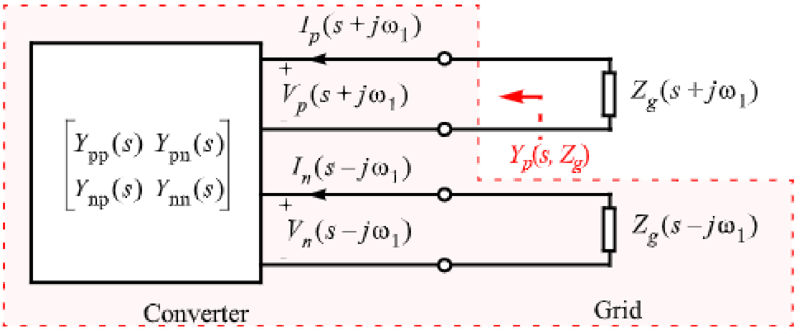


Figure B. 2 SISO representation of the system.

Summary of the proposed method

1. Select a frequency s and inject two test signals in the sequence domain. One at positive sequence frequency $(s + j\omega_1)$ and another in negative sequence frequency $(s - j\omega_1)$.
2. Perform FFT analysis and extract 4 sets of components for both injections.
   - For the first injection; positive sequence injection.
   - Voltage $V_p(s + j\omega_1)_{(1)}$ at positive sequence perturbation frequency
   - Current $I_p(s + j\omega_1)_{(1)}$ at positive sequence perturbation frequency
   - Voltage $V_n(s + j\omega_1)_{(1)}$ at negative sequence perturbation frequency
   - Current $I_n(s + j\omega_1)_{(1)}$ at negative sequence perturbation frequency
   - Repeat for the second injection, negative sequence injection.
3. Estimate the correction factor and apply the compensation to the phases of Fourier components.
4. Obtain the sequence domain admittance matrix Y_{PN}
5. Select another frequency s from the frequency interval of interest.

3. Impedance Modelling of Three-Phase Voltage Source Converters in DQ, Sequence, and Phasor Domains [6]

- The coupling between grid side impedance (AC and DC) and DC side impedance cannot be neglected. The coupling plays an important role in low-frequency resonance, which is not captured by decoupled methods.

- The stability analysis of the frequency response converting sequence domain into dq domain is crucial to produce an accurate assessment.
- Poorly tuned PLL introduce unstable resonance between the control loop and the grid impedance. Those comparisons are possible using the proposed linear transformation between different domains (sequence, dq, and phasor)

4. Impedance-based stability criterion for grid-connected inverters [4]

- The effects of grid impedance on grid-connected inverter stability can be studied by using a simple small-signal model involving the grid impedance and the inverter output impedance.
- Under the assumption that the inverter is controlled as a current source and is stable when operating with an ideal grid, it will remain stable with a nonideal grid if the ratio of the grid impedance to the inverter output impedance satisfies the Nyquist criterion.
- The analysis also indicates that grid-connected inverters should be designed to have as high output impedance as possible to be able to operate with a wide range of grid impedances.

Key features of the reviewed method

- The method provides a stability assessment criterion based on grid and IBR output impedance scan.
- Once the grid side and IBR impedances are obtained from frequency scans, the bode plot of both impedances can be computed. The phase difference needs to be checked at frequencies where the magnitude of the grid side impedance is equal to the IBR impedance. If the phase difference between the grid side and the IBR side is closer to 180 degrees, the system will have lower damping for harmonics at that frequency.

5. Reactance scan crossover-based approach for investigating SSCI concerns for DFIG-based wind turbines [7]

- Decoupled analysis for WTG side and system side. The analysis focuses on the crossover reactance on the network side (driving point reactance scan) to define a potential SSO, checking the resistance of the WTG side.
- The analysis showed potential SSO not only for radial connections but also for meshed (close to radial).
- There is a higher SSO risk for scenarios with faults, in comparison to those only with outages.

Key features of the reviewed method

- The method provides a tool to identify potential SSCI by combining two scans. The driving point scan to obtain the system impedance is performed using PSCAD harmonic injection (Method 1 in Figure B. 3). The method of harmonic injection is suitable for systems with active components like WTG.
- If the system reactance has crossover frequencies, which means that the reactance crosses from positive to negative, and vice-versa, the system may be at risk of SSO.
- The system reactance scan is combined with the WTG scan. In this case, the WTG is disconnected from the system and connected to an ideal source.
- The condition for SSO risk for a given system condition (dispatch, network outages, etc) can be defined as follows "If there are reactance crossover on the system and those frequencies have negative resistance on the WTG scan, then there is a risk of SSO" (see Figure B. 4).
- The method is simple and easy to implement and interpret.
- The method requires performing two sets of harmonic injections; one with the system with the WTG, and another with the WTG connected to an ideal voltage source. The drawback is the amount of computation required to perform the scans.

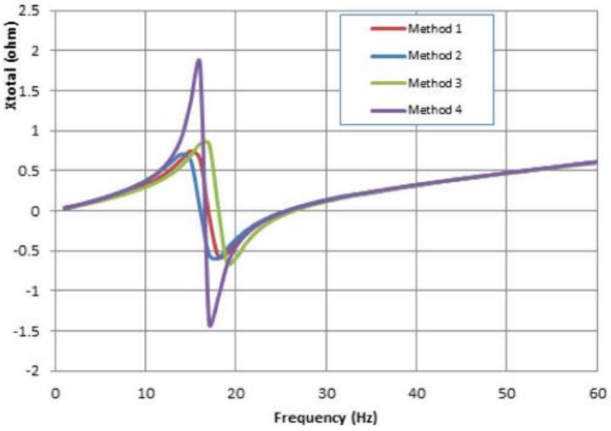


Figure B. 3 Total reactance of the network

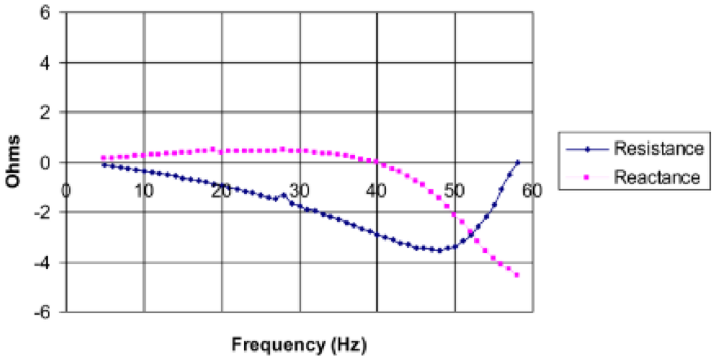


Figure B. 4 Resistance and reactance of the WTG.

Summary of the proposed method

1. Define system condition. Introduce changes on WTG dispatch or the status of branch elements in the network.
2. Perform a harmonic impedance scan and obtain the system reactance as a function of frequency.
3. Perform a harmonic impedance scan of the turbine connected to an ideal voltage source and obtain the WTG resistance as a function of frequency.
4. Verify the if the frequency of the system reactance crossover also has a negative resistance on the WTG impedance. If so, the system has a potential risk of SSO under the given condition.

6. A Refined Frequency Scan Approach to Sub-Synchronous Control Interaction (SSCI) Study of Wind Farms [5]

- A detailed mathematical explanation of frequency coupling due to d-q saliency. In the analysed DFIG WTG the effect of different gain values Kd and Kq results in the creation of new harmonics components (frequency coupling).
- The derivation of impedance considering frequency coupling improves the accuracy of the detection of potential SSO risks.

Key features of the reviewed method

- The method can identify when there is a source of mirror frequency coupling on the WTG side. The inclusion of a term accounting for frequency coupling makes the method more accurate in comparison with standard methods that assume that the frequencies are decoupled.
- The method is easy to implement, but it is computationally intensive; it requires two perturbations for each frequency under study. However, it does not need to scan the grid and WTG independently.
- The new method (considering frequency coupling) also uses the reactance crossover frequency based on the total impedance to identify SSO risk (see Figure B. 5).

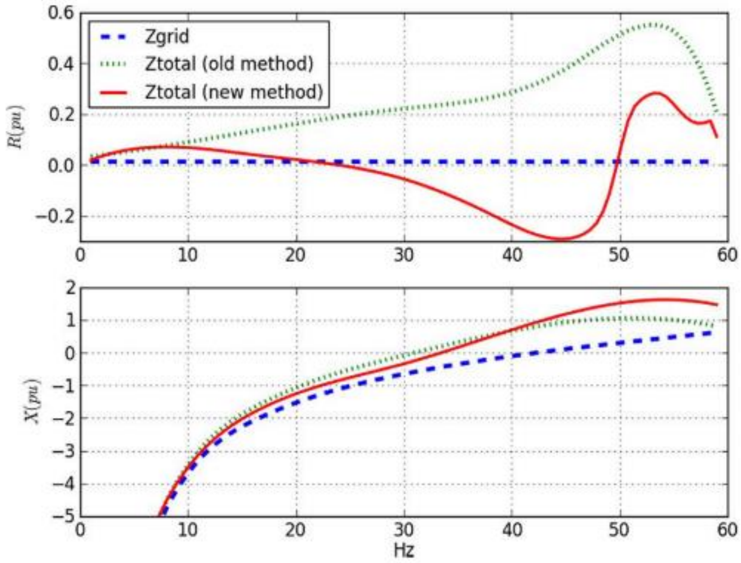


Figure B. 5 Resistance and reactance of the whole system.

Summary of the proposed method

1. Establish a steady state operating condition of the system under test (e.g., power, and reactive power out of the WTG (see Figure B. 6), terminal voltage, frequency). Record the steady state quantities with a known Z_{test}^{1}.

2. Inject a perturbation signal at $f1$, and measure voltage and current at the same frequency.
3. Inject another perturbation at $f2$, which must be at the coupling frequency of $f1$, and measure the voltage and current at the same frequency.
4. Repeat until $f1$ sweeps the range of frequencies under study.
5. Define another known test impedance Z_{test}^{2} and repeat the previous injection steps.
6. Solve the equation for self-impedances.
7. Solve the equation for coupling impedances.
8. Finally, obtain the total system impedance.

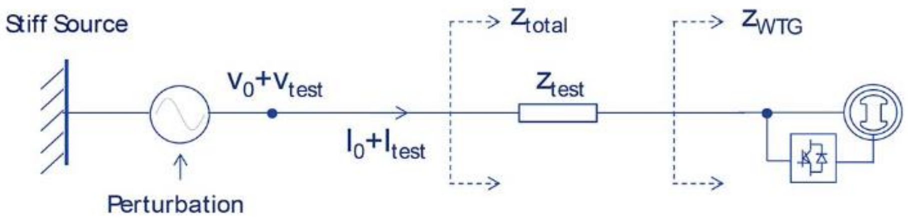


Figure B. 6 Setting for WTG scan.

Appendix C: Additional grey-box results

The white box reference results for the Layer 1 analysis of the 20 Hz SSO mode were discussed in Section 3, where the Layer 1 analysis correctly identified that Apparatus 12 is the main apparatus participating in the mode.

The other poorly damped mode which is well captured by the vector fitting is at approximately 5.5 Hz. Figure C. 1 and Figure C. 2 show that the vector fitting extracted the residues and generated the Layer 1 results well, identifying that Apparatus 3 is the main contributor to the mode (with the state-participation factors confirming the analysis in Figure C. 3).

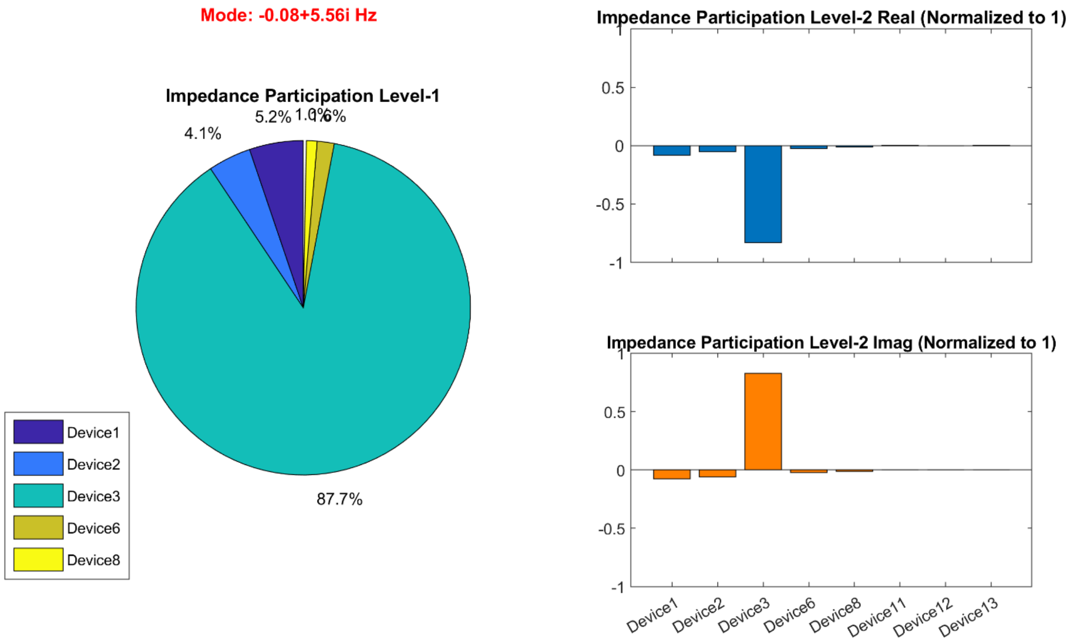


Figure C. 1 Layer 1 & Layer 2 analysis using the white box model.

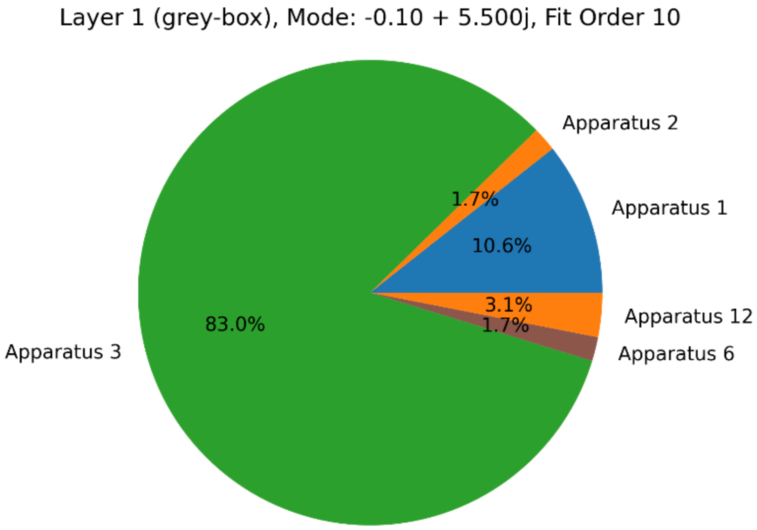


Figure C. 2 Layer 1 analysis using the vector fitting results and the grey box method.

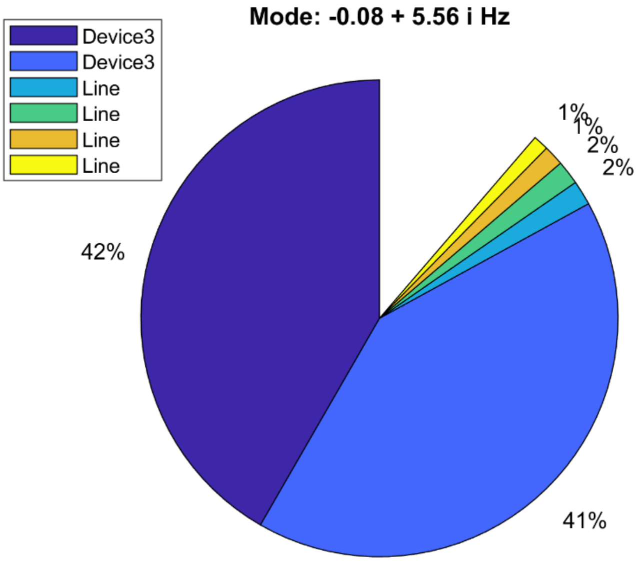


Figure C. 3 State participation factor results for the 5.5 Hz Mode

However, there is also a scenario where the vector-fitting does not perform as well. It can be seen in the zoomed mode plot (Figure 3-4 (b)) that the vector fitting only identifies one mode where there is a cluster of three modes. As might be expected, the identified residues are incorrect and the Layer 1 analysis identifies the Apparatus 1, 2, 6 and 8 as all contributing to the mode. As can be seen in Figures C. 5 to C. 7 this is a rough amalgamation of the individual results for the three modes.

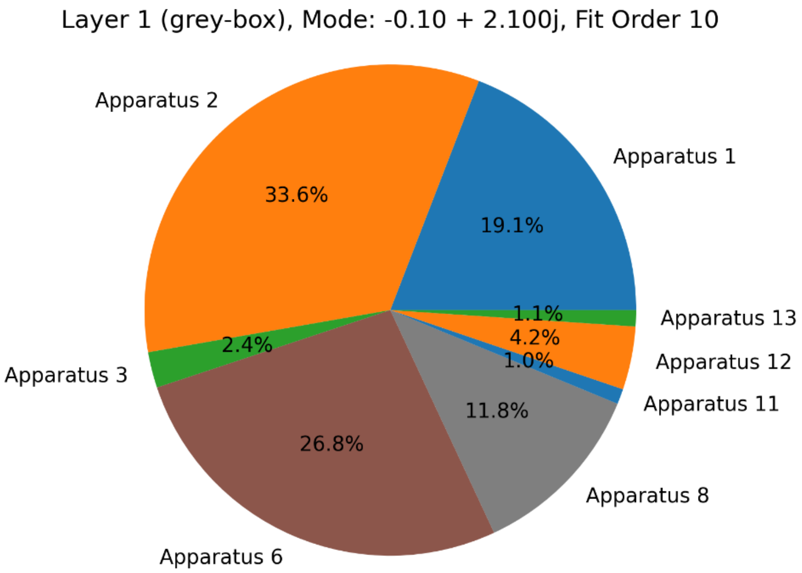


Figure C. 4 Layer 1 analysis using the vector fitting results and the grey box method.

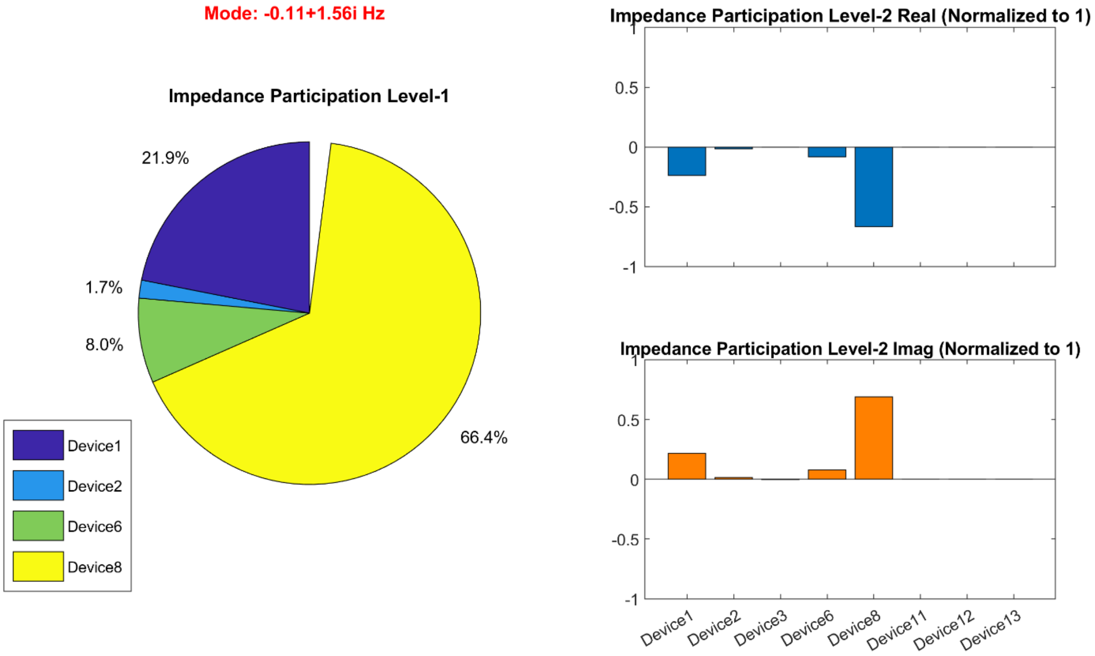


Figure C. 5 Layer 1 & Layer 2 analysis using the white box model.

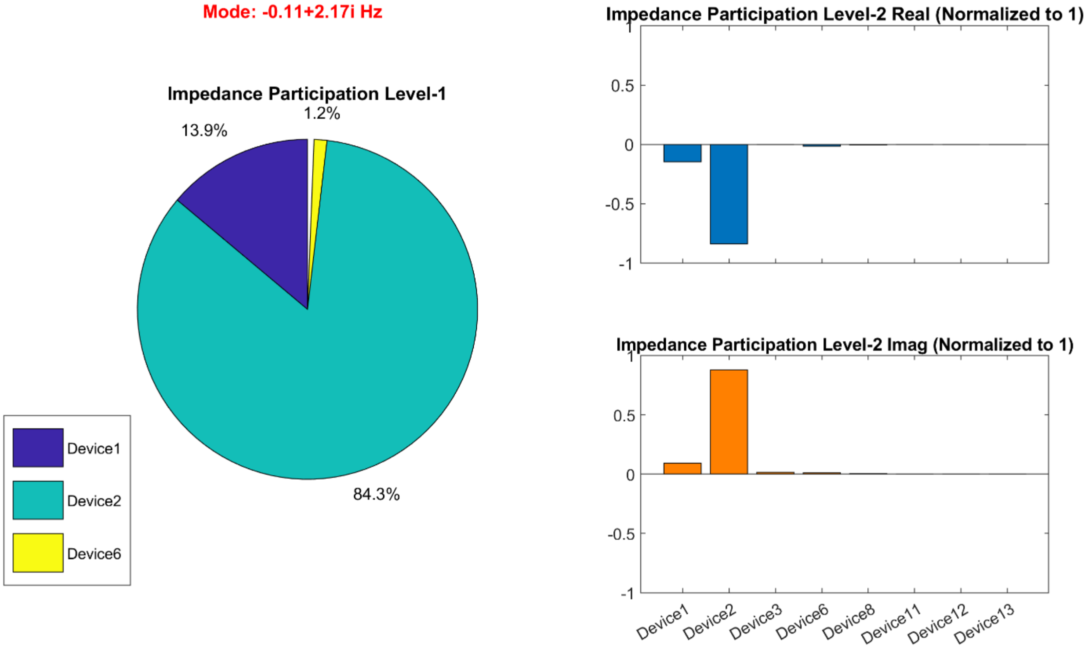


Figure C. 6 Layer 1 & Layer 2 analysis using the white box model.

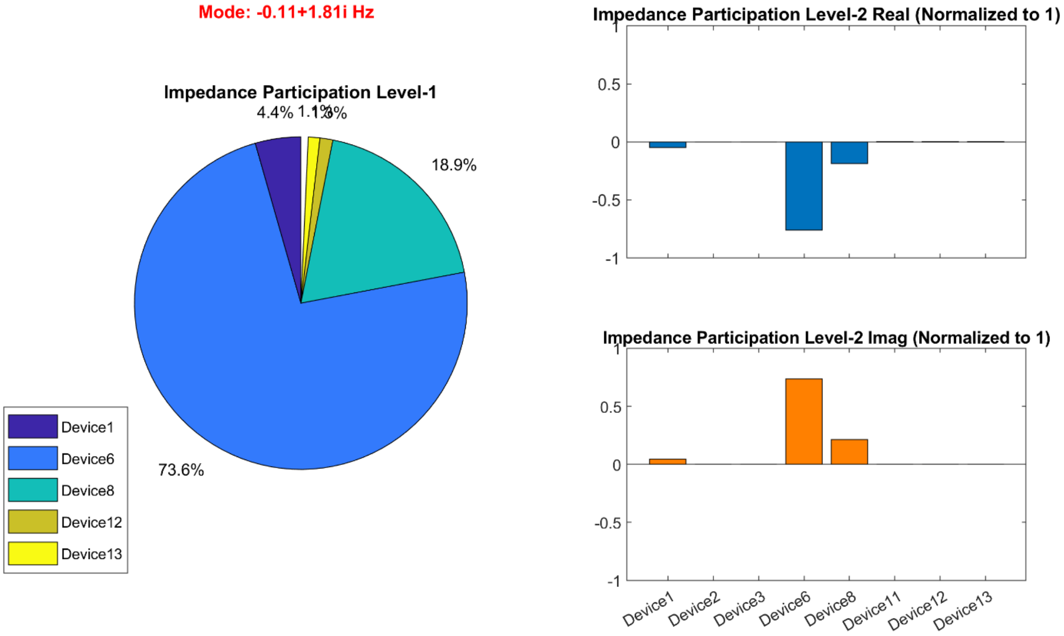


Figure C. 7 Layer 1 & Layer 2 analysis using the white box model.