What is the SIaD tool?

The Stability and Interaction assessment in the frequency-Domain tool (SIaD) is a open-source tool designed to analyze small-signal stability and interactions in large modern power systems. It helps identify potential interactions and assess closed-loop stability using advanced frequency-domain analysis techniques.

SIaD consists of two main modules:

1. Scanning Module: Determines the system's frequency response through impedance and admittance matrices.
2. Analysis Module: Applies the Generalized Nyquist Criterion and other specialized algorithms to evaluate system stability and interactions.

Key features and capabilities:

- Identification of interaction frequencies and stability across a wide frequency range (including subsynchronous and harmonic regimes).
- Determination of stability limits between interconnected systems.
- Prediction of oscillations and their damping levels.
- Analysis of multiple operating points.
- Open-source code and interface for Transmission System Operators (TSOs) and academic institutions.

Where to Get the SIaD tool?

The SIaD tool is freely available for download from the following GitHub repository:

https://github.com/luisangelgare/SIaD-Tool

As this is an open-source code, any use, modification, or contribution based on this tool must properly acknowledge the original authors and contributors. To cite this work, please use the following reference:

García-Reyes, L. A., Gomis-Bellmunt, O., Prieto-Araujo, E., Lacerda, V. A., & Chea-Mañé, M., "SIaD Tool: A Comprehensive Frequency-Domain Tool for Small-Signal Stability and Interaction Assessment in Modern Power Systems", in IEEE Transactions on Power Delivery, submitted for review.

Email: luis.reyes@upc.edu

Requirements to Start Using the SlaD Tool

MATLAB Version:

- MATLAB R2022b or newer
- Simulink R2022b or newer
- Control System Toolbox
- Simscape Electrical

Recommended Hardware:

- Intel Core i5 processor or higher
- 8 GB RAM or more

How to Use the SlaD Tool?

Once the **SlaD** tool source files have been downloaded and extracted, the usage process is divided into two separate guides depending on the working environment:

Using MATLAB/Simulink:

After extracting the contents of the .zip file, you will find four main files, as shown in Figure 1.

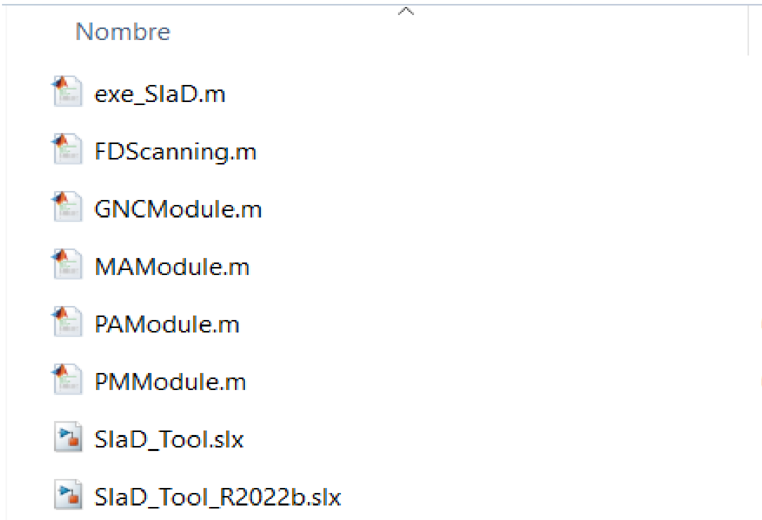


Figure 1. Files included in the **SlaD** tool for MATLAB/Simulink version.

Each file serves the following purpose:

- *exe_SlaD.m*: The main executable script that launches the **SlaD** Tool and contains all necessary settings and parameters for scanning and stability analysis.

Email: luis.reyes@upc.edu

- *FDScanning.m*: The master script containing the full algorithm for open-loop system scanning. This file must be placed in the working directory without modification.
- *GNCModule.m*: Executes the Generalized Nyquist Criterion (GNC) with all configuration options preloaded. This file must remain unmodified in the working directory.
- *MAModule.m*: Modal analysis module used to perform eigenvalue-based and participation factor-based small-signal analysis. This file must remain unmodified in the working directory.
- *PAModule.m*: Passivity Assessment module for evaluating both systems. This file must remain unmodified in the working directory.
- *PMModule.m*: Phase Margin Analysis module. This file must remain unmodified in the working directory.
- *SlaD_Tool.slx*: A Simulink file that includes the tool block. It contains essential parameter configuration information for proper use of the **SlaD** tool. Versions are available for R2024b and R2022b.

Steps to Install and Use the SlaD Tool (MATLAB/Simulink)

Step 1: Set Up the Working Environment

Copy the *FDScanning.m, GNCModule.m, MAModule.m, PAModule.m* and *PMModule.m* files into your main working directory (see Figure 2). This folder should include all the necessary files to run the simulation, such as:

- Scripts for pre-calculating simulation parameters.
- Auxiliary functions used by the main script and its dependencies.
- The Simulink EMT model file (.slx) of the system to be identified.

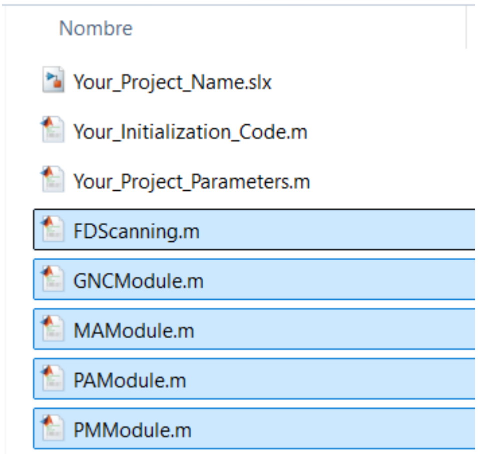


Figure 2. Copying the master files to your working directory.

Email: luis.reyes@upc.edu

Step 2: Configure the Execution Script

Also copy the *exe_SlaD.m* file into the same working directory mentioned earlier (see Figure 3). Once the file is in place, open it to define the scanning and system identification parameters.

This script allows you to configure key aspects of the analysis, such as the frequency range, steady-state, type of perturbation, and other parameters required to successfully run the identification process.

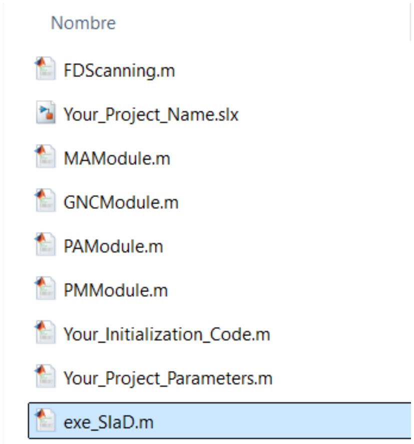


Figure 3. Copying the file *exe_SlaD.m* to your working directory.

Step 3: Define Global Variables in the Execution Script

Open the *exe_SlaD.m* file. Inside the script, you will find a section for defining the global variables used by the tool. These variables must be **unique within the working environment**, so make sure they are **not duplicated** in your initialization scripts, parameter files, or especially within the Simulink EMT model.

3.1 Simulation Parameters

The first set of parameters in the *exe_SlaD.m* script is divided into two sections. The variables to be defined are shown in Figure 4 and are explained below.

```
%% Simulation parameters

program='Your_Project_Name.slx'; % Simulink program selection
model='Your_Project_Name'; % Main canvas name of your project (same as program usually)
Tinit=6; % Initialization time (arbitray for select the steady state)
fs=1; % Resolution frequency for FTT (specify the value in Hz for minimal freq)
delta_t=25E-6; % Fixed step time
%%% ----------------------------------------------------------------------
% Frequency band selection: choose one and comment the other
% --- Specify the frequency vector for SINGLE-TONE perturbations
fd0=unique(round(logspace(0,log10((1/delta_t)/4),3)));  % Perturbation frequencies in Hz
% --- Specify the frequency band for MULTI-TONE or PRBS perturbations
% fd0=[0.001 6*50];  % Frequency band in Hz (RECOMMENDED 6 TIMES THE FUNDAMENTAL)
%%% ----------------------------------------------------------------------
```

Figure 4. First block of variables definition in *exe_SlaD.m*.

Email: luis.reyes@upc.edu

The first section includes the simulation parameters, which must be carefully defined to ensure proper execution of the analysis.

- `program`: This is where you insert the name of the .slx file containing the EMT model of the system to be analyzed. For example, in the application shown in Figure 3, the file name would be "*Your_Project_Name.slx*".
- `model`: Repeat the same name as above, but without the .slx extension, assuming the main canvas in Simulink uses the same name.
- `Tinit`: This parameter defines the time at which the system reaches a steady state. It is set arbitrarily by the user based on prior observation of the system's transient behavior. In other words, it should correspond to the moment when the initial transient has passed and all controllers and system dynamics are active. Figure 5 shows a practical example of how to identify this time in a simple transient. As illustrated, T_{init} would be the point where the system is stable and ready to receive perturbations for small-signal identification.

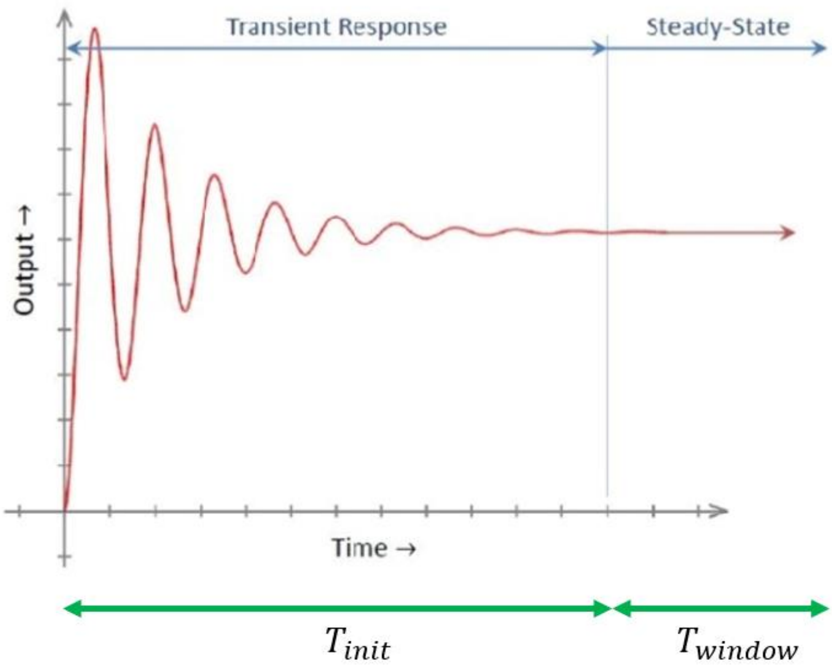


Figure 5. Details in the selection of T_{init}.

- `fs`: This parameter defines the desired frequency resolution in the frequency domain, i.e., the minimum frequency to be identified in the system. It directly determines the time window (T_{window}) used during the simulation, based on the inverse relationship:

$$fs = \frac{1}{T_{window}}$$

Email: luis.reyes@upc.edu

Therefore, a lower resolution frequency `fs` requires a longer time window T_{window} to accurately capture the system dynamics within that frequency range. When the `fs` parameter is set, **the tool automatically** calculates the additional simulation time required to cover the necessary observation window. This total simulation time is referred to as `Tobs`, and is defined as:

$$Tobs = T_{init} + T_{window}$$

For example:

- If a 1 Hz spectral resolution is desired, the time window will be 1 second, so the simulation must run until $T_{init} + 1\ s$.
- If a 0.1 Hz resolution is required, the window will be 10 seconds, and the total simulation time will be $T_{init} + 10\ s$.

This parameter can also be interpreted as the **minimum frequency to be identified**, and its selection depends on the type of study being conducted, such as:

- Subsynchronous interactions (require lower frequencies, hence longer windows).
- Harmonic interactions (may require higher frequencies, with shorter windows).

- `delta_t`: This is the discrete integration step defined by the user for EMT simulations and system identification. While arbitrary, it must be consistent with the type of study:
  - For harmonic interactions (high frequency), a high resolution is recommended, e.g., 10 µs.
  - For low-frequency interactions, a larger step can be used to speed up the study, with a maximum recommended value of 50 µs.
- `fd0`**:** This is a frequency vector defined by the user to select the frequency band for system identification. By default, a *single-tone* identification is used, with a logarithmic scale vector from 1 Hz up to the maximum sampling frequency allowed. It is crucial not to exceed the sampling limit to avoid aliasing errors:

$$fd_{max} < \frac{fsampling}{4}\ ;\ where\ fsampling = \frac{1}{delta_t}$$

This vector can be modified to specify all desired frequencies for system perturbation.

Email: luis.reyes@upc.edu

When multi-tone perturbations are used, the vector format must be adjusted to include only the initial and final frequencies. This allows the synthesis of either a sinusoidal or random binary waveform that spans the specified frequency range. For improved identification at low frequencies, a narrow spectrum is recommended (e.g., 1 to 300 Hz). However, this approach is not recommended for high-frequency analysis.

It is important to ensure consistency between the selected perturbation signal type and the corresponding configuration. If both settings are not aligned, the program will return an error. Therefore, one configuration must be commented out when the other is active.

Recommendation: The authors suggest using *single-tone* perturbations due to their higher accuracy.

3.2 Steady-State and Disturbance Source Data

This section includes variables required for the **ideal source** to properly identify the operating point of the element under study. In Figure 6 the available parameters to set the steady state data are shown. Here you must enter the general system parameters at the point of scan: `f0`, `Sbase` and `Vbase`.

```
%% Steady state and disturbance sources data

f0=50; % Fundamental base frequency
Sbase=2.75E6; % Base power (where the tool is connected)
Vbase=1; % Base voltage (V)
Vq_ss=0; % q-component steady state voltage at bus under analysis
Vd_ss=0; % d-component steady state voltage at bus under analysis
theta_ss=0; % Steady state angle at bus under analysis
Iq_ss=0; % q-component steady state current at link under analysis
Id_ss=0; % d-component steady state current at link under analysis
Vperturbation=0.01; % Percentage of nominal voltage perturbation
Iperturbation=0.01; % Percentage of nominal current perturbation
```

Figure 6. Second block of variables definition in *exe_SIaD.m*.

There are two available methods to configure the steady-state operating point:

1. **Manual configuration (`ss_cal = 0`):** if `ss_cal = 0` (see Step 4 for details on this variable), the open-loop option is selected. In this case, you must manually enter the constant values used in the EMT simulation: `Vq_ss, Vd_ss, theta_ss, Iq_ss, Id_ss;` previously measured under steady-state conditions. These values must be consistent with the units used in the Simulink model:

Email: luis.reyes@upc.edu

- If the .slx model uses per-unit (p.u.) values, enter them in p.u.
- If real units are used (e.g., volts or amperes), maintain those units.

2. **Automatic calculation (`ss_cal = 1`):** If ss_cal = 1, it is not necessary to define `Vq_ss, Vd_ss, theta_ss, Iq_ss, Id_ss`, as the tool will automatically determine the steady-state operating point. The variables `Vperturbation` and `Iperturbation` define the percentage of disturbance applied to voltage and current, respectively. These values should always remain between **0.01 and 0.03** to preserve linearity and ensure the system remains close to its nominal operating point.

NOTE: In both cases, it is mandatory to define `f0, Sbase` and `Vbase.`

Step 4: Select the Scanning Options

Next, you must select the available scanning options within the script. Figure 7 shows the steady-state and scanning methodologies integrated into the tool. All available combinations can currently be used. However, to properly tailor each methodology to your specific study, it is highly recommended to consult the reference article associated with the tool. Doing so will help you achieve more accurate and relevant results for your analysis.

```
%% Scanning options available

% Steady-state calculation:
% 1 -> Yes, take from 1st simulation (closed loop)
% 0 -> No, I addded the d- and q-component previously (open loop)
ss_cal=1;

% Scanner settings:
% 1 -> Voltage perturbation
% 2 -> Current perturbation
scanner_type=1;

% 1 -> Single-tone perturbation
% 2 -> RBS perturbation
% 3 -> Multi-tone perturbation
signal_type=3;

% 1 -> ABC scan
% 2 -> qd0 scan
% 3 -> pn0 scan
scanner_selector=2;

% 1 -> linear response exists
% 0 -> no linear response is available
linear=0;
```

Figure 7. Scanning options available in the SIaD tool.

Email: luis.reyes@upc.edu

Users can choose between different types of perturbation and identification configurations:

- Steady-state calculation:
    - **Automatically measured** from the first simulation (closed loop).
    - The user manually enters the values in the *dq* reference frame (open-loop).
- Perturbation type:
    - **Voltage-based** or **current-based**.
    - Identification signals can be either **random binary** or **multi-frequency sine**.
- Reference sequence:
    - Identification can be performed in different **reference sequences** (e.g., positive-negative or dq), depending on the analysis requirements.
- Linear validation:
    - If a **theoretical linear response** of the system is available, the "linear" option can be enabled to perform an additional validation of the identification results.

All available options are clearly numbered and described within the script, allowing for a very straightforward and user-friendly selection process.

Step 5: Finalize the Executable Script Configuration

The final part of the *exe_SIaD.m* script includes the necessary components to properly execute the identification process (see Figure 8):

1. Load the system parameter program: If your project requires **pre-processing calculations** before initialization or simulation (e.g., parameter estimation, controller setup), this is where you should include the corresponding script.
2. Project initialization: If a **power flow** or other initialization process is required before running the EMT simulation, it should be added here.
3. Finally, users must specify which analysis modules to activate in order to evaluate the stability and interaction characteristics of the identified matrices for both systems. The available modules include various techniques for small-signal analysis, passivity assessment, and frequency-domain stability evaluation.

Email: luis.reyes@upc.edu

```
%%% ----------------------------------------------------------------------
% Here is your code for initizalization
Your_Project_Parameters;
Your_Initialization_Code;
%%% ----------------------------------------------------------------------

% Loading the FDs main code...
FDScanning;

% Loading the GNC Module
GNCModule;

% Loading the Modal impedance analysis Module
MAModule; % Do not use it if PRBS or Multi-tone strategy are selected!

% Loading the Phase Margin analysis Module
PMModule;

% Loading the Passivity assessment Module
PAModule;

%%% ----------------------------------------------------------------------
% Save your matrices here in the format you desire!
```

Figure 8. Final setup for the *exe_SlaD.m*.

Step 6: Integrate the SlaD Block in Simulink

To integrate the block that enables system identification within your Simulink model, follow these steps:

1. Open the *SlaD_Tool.slx* file, located in the **root folder** where the tool was downloaded (see **Figure 1** for a reference of the folder structure).
2. Inside this file, you will find the block named **"SlaD Tool"**, colored in **red** for easy identification.
3. This block should be **copied and inserted** into your project's EMT model (.slx) at the point in the system where you want to perform the analysis.

Figure 9 shows the visual location and appearance of the block within the original file.

Email: luis.reyes@upc.edu

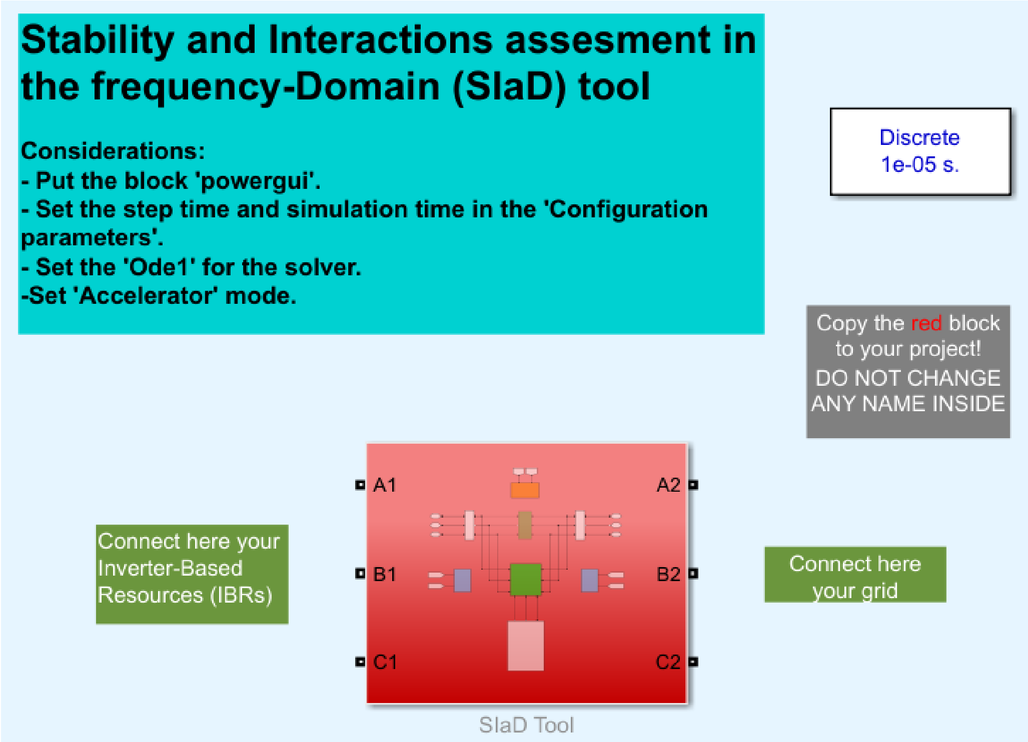


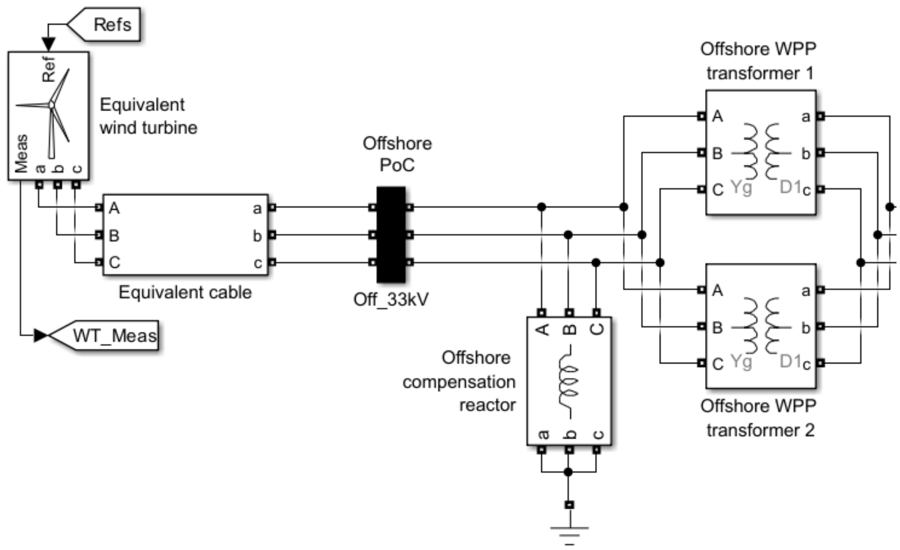
Figure 9. Frequency Scanner block.

Once the **SlaD Tool** block is located, it must be **copied and inserted** into your simulation project: *Your_Project_Name.slx*. Next, you must **properly connect the terminals** of the block to the **three-phase buses of system 1 (suggested for power electronics-based resources) and system 2 (suggested for grid or any other element)** where system identification is to be performed. This connection point should correspond to the location in the system where the frequency response is to be measured. Notice that the tool is made to determine the interactions between power electronics (left side) and the grid (right side), so it is recommended to follow the convention here proposed.

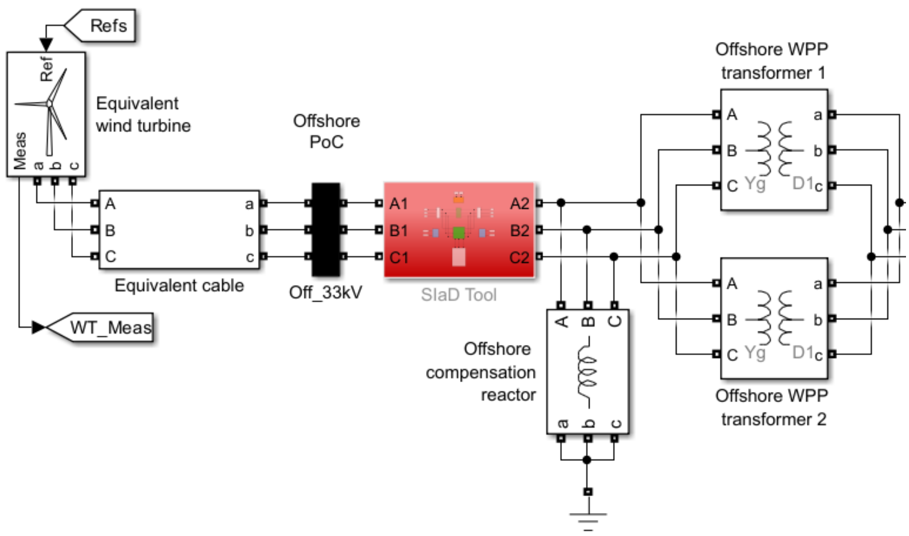
For example, in a typical case study showed in the Figure 10, the scanner is connected at the Point of Connection (PoC) of a wind farm located at an offshore node, in order to identify its dynamic behavior. Then, in Figure 10-a), the initial EMT model is presented and next in Figure 10-b), the connection of the scanner is highlighted. Notice that the ports A1, B1 and C1 are connected to the power electronics resource and the ports A2, B2 and C2 are connected to the rest of the system.

Since the connection points are **three-phase**, it is essential to ensure that **each phase (A, B, and C)** is connected correctly.

Email: luis.reyes@upc.edu



a) Scheme before the SlaD Tool connection.



b) Scheme after the SlaD Tool connection.

Figure 10. Offshore wind farm example of connection.

Step 7: Configuring the Simulink Environment for the SlaD Tool

Following the instructions shown in the **SlaD Tool** block interface (see Figure 9), the following steps must be taken to properly integrate the tool into your project:

7.1. Configure solver parameters

Due to the nature of the identification method, your .slx model must operate in **discrete time**. To do this:

- Open your model in Simulink.

Email: luis.reyes@upc.edu

- Click on the **“Model Settings”** icon ( ) and go to **Solver** options
- Set the variables `Tobs` and `delta_t` (previously defined in the *exe_SlaD.m* script).
- Select the **“Fixed-step”** option.
- Choose the solver **ode1 (Euler)**, which is recommended by the authors for its simplicity and compatibility with the identification method.

Once these parameters are configured, click **“Apply”** to save the changes and then **“OK”**. This process is illustrated in **Figure 11**.

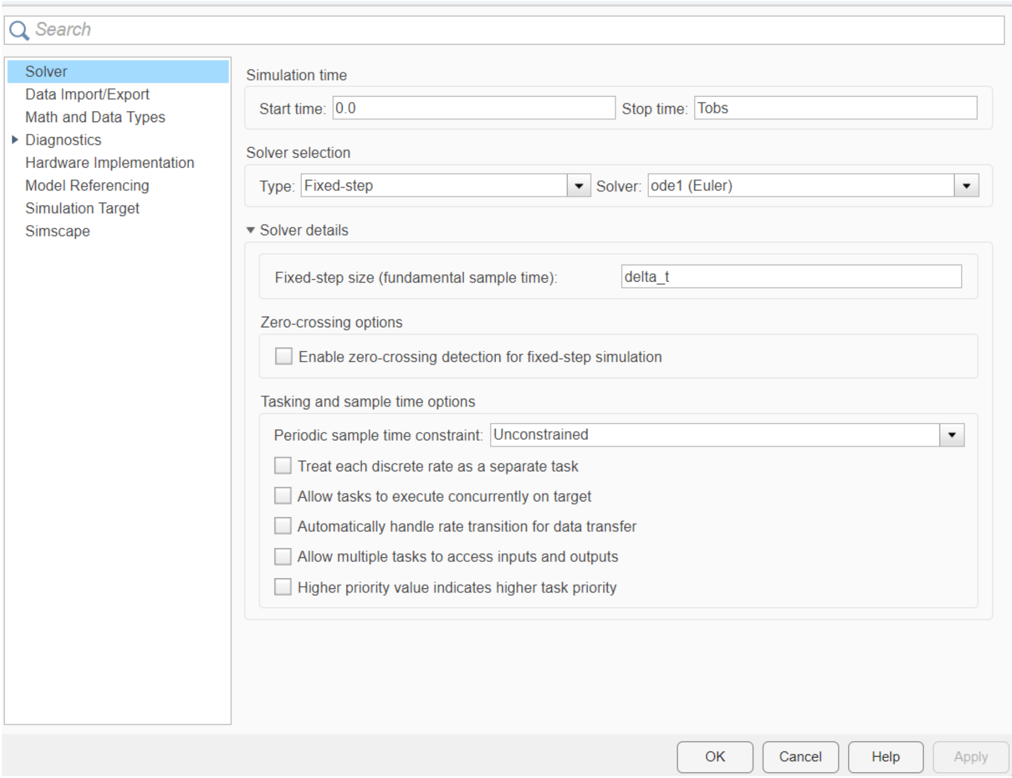


Figure 11. Model settings adjustments for the **SlaD** tool.

7.2. Enable “Accelerator” Simulation Mode

As a final step, it is recommended to enable the **“Accelerator”** simulation mode to improve performance during model execution (see Figure 12).

To do this:

- Go to the **“Simulation”** section in the Simulink top menu.
- Then, under the **“Simulate”** subsection, just below the **“Stop Time”** field, open the dropdown menu and select **“Accelerator”**.

Email: luis.reyes@upc.edu

This setting allows Simulink to compile the model before running it, which can significantly reduce simulation time, especially for complex systems.

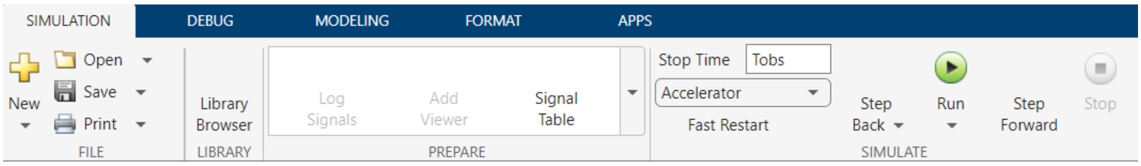


Figure 12. Simulation acceleration option selection.

Once all changes are made, **save the model**. At this point, the environment is fully configured and ready to begin working with the SIaD tool.

Step 8: Running the Study and Viewing Results

Once all **7 steps** of this manual have been completed, click **"RUN"** in the executable script (*exe_SIaD.m*). The tool will automatically perform the identification study and generate the real-time status of the analysis and will print the results in the command window and as **plots with the results**, depending of the modules previously selected for the entire study.

The **identification variables** will be saved in the MATLAB **Workspace**, organized according to the selected **reference frame**. For example:

- Yxx or Zxx, where x can be:
  - ABC (phase sequence),
  - dq0 (synchronous reference frame),
  - 0pn (modal sequence: zero, positive, negative).

These variables will be available for further analysis depending on your study needs. The 3D matrices will be available as well as Y_sys1 and Ysys2.

APPENDIX A: Important generalities within the process of SIaD tool.

In general, the complete identification and stability/interaction assessment workflow is illustrated in the diagram shown in Figure A1, which summarizes the steps from setup to results for the two systems where the tool is connected.

Email: luis.reyes@upc.edu

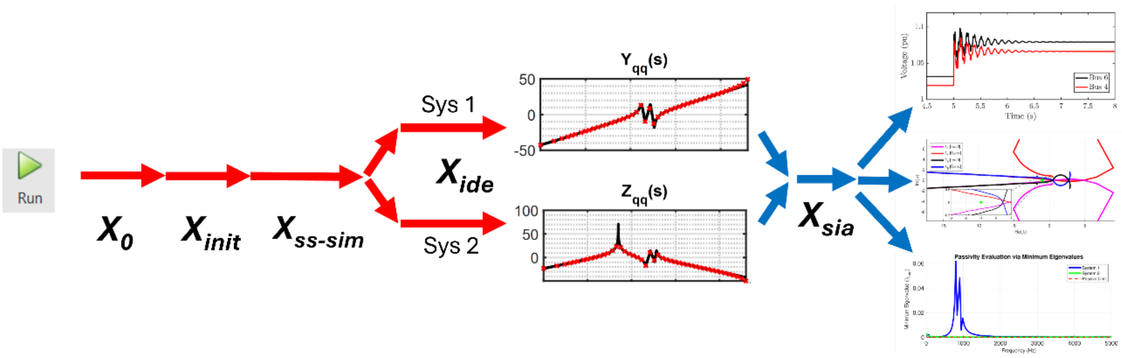


Figure A1. **SlaD** tool full process workflow.

A.1 Steady-State Parameter Calculation (X_0)

The first stage of the identification process, labeled X_0 in Figure A1, corresponds to the calculation of the system's steady-state parameters. This step is equivalent to running your custom script, for example: *Your_Project_Parameters.m*. This script should include all necessary calculations to generate the variables required for the EMT simulation, such as initial conditions, electrical parameters, controller settings, and more.

Executing this step ensures that the system is properly parameterized before starting the simulation and applying perturbations. Figure A2 provides a detailed illustration of this process.

```
%% Insert here your file with your system initialization data (without run)

%------------------
% Here is your code for parameters file and initizalization
Your_Project_Parameters;
Your_Initialization_Code;
%------------------

% Loading the FDs main code...
FDScanning;
```

Figure A2. Initialization of the **SlaD** tool.

Your_Project_Parameters.m **must not contain any commands that start the EMT simulation**, especially if the SlaD block is already connected in the .slx mode (see **Step 6** for reference on integrating the **SlaD** block in Simulink).

Email: luis.reyes@upc.edu

A.2 Stage X_{init}: Model Initialization

If your project requires a **power flow** or other pre-simulation calculation to obtain necessary variables, include tour script: *Your_Initialization_Code.m*. This script must be placed in the **same folder** as the rest of your project files (refer to **Figure 3** for the file placement).

A.3 Stages $X_{ss\text{-}sim}$ and X_{ide}: Simulation and Identification

This script stage automatically executes the last two stages:

- $X_{ss\text{-}sim}$: Runs a simulation to calculate the steady-state values of **voltage and current** at the bus where the **SIaD** tool is connected.
- X_{ide}: Performs the **complete system identification of both sides**.

The duration of this stage depends on the type of perturbation signal used:

- Random binary (RBS) and multi-frequency sine signals allow for faster identification.
- Single-tone signals take longer but provide higher accuracy.

Recommendation: Single-tone signals are recommended for more precise identification results.

A.4 Stage X_{sia}: Stability and Interactions assessment

This script stage automatically performs the small-signal analysis previously selected by the user. It is just necessary that you comment the modules you don't want to use. Once you click, run, the tool automatically print the status and the results in the Command Window. The available plots will be displayed as well.

CITCEA-UPC thanks you for downloading the SIaD Tool. We appreciate your interest and welcome any questions, suggestions, or collaboration proposals related to the use and development of the tool.

Who to Contact for Support or Follow-Up?

For questions, follow-up, or technical support regarding the **SlaD** tool, please contact:

📧 luis.reyes@upc.edu

Important Note

The SlaD tool is still under development. Any **improvements, suggestions, or issues** you identify are highly appreciated and should be reported to the authors to help **refine and enhance** the tool for the benefit of the community.

Acknowledgements

This work has received funding from the ADOreD project under the European Union's Horizon Europe Research and Innovation Programme under the Marie Skłodowska-Curie Grant Agreement No. 101073554.

Email: luis.reyes@upc.edu