SFDEsim
Simulator user manual

User manual for the SFDEsim simulator.
Explains the basic use of the simulator program.

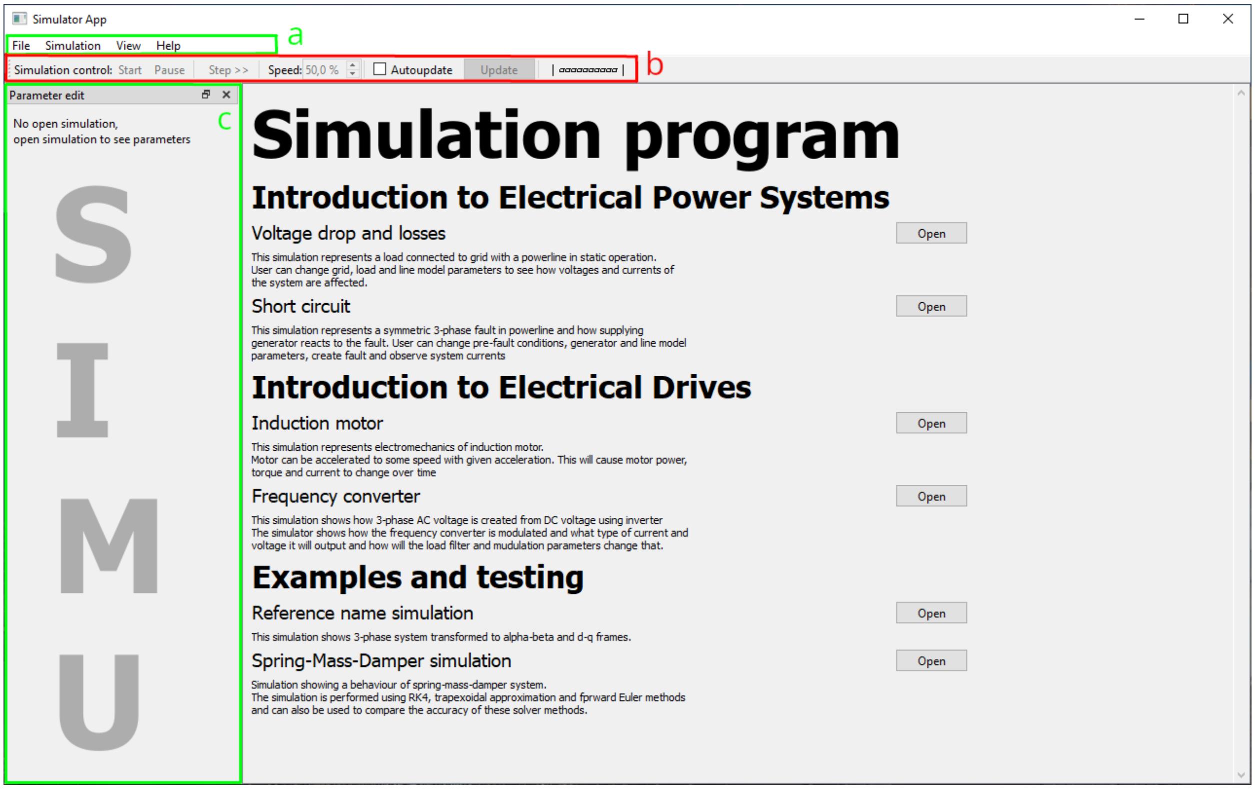
Table of contents

Introduction

SFDEsim-Simulator is made as a part of master's thesis project about usage and benefits of simulation in bachelor's level electrical engineering courses. The SFDEsim-simulator's goal is to provide visual and interactive learning opportunities in simple cases requiring a minimal amount of user knowledge of the subject for use. This manual instructs on general use of the simulator, for setup, see the setup manual and for simulation specific information, see the simulation info sheets. The program is supported on Windows and Linux operating systems.

Startup menu

The simulator launches to the main screen seen in figure 1. From the main screen, simulations can be opened, additionally all other elements are loaded into the window, including **Program menu** (figure 1, a), **Control panel** (figure 1, b) and **Parameter edit** (figure 1, c). When the simulation is opened, all simulation specific elements are loaded into view.



No open simulation,
open simulation to see parameters

Simulation program

Introduction to Electrical Power Systems

Voltage drop and losses

This simulation represents a load connected to grid with a powerline in static operation.
User can change grid, load and line model parameters to see how voltages and currents of the system are affected.

Short circuit

This simulation represents a symmetric 3-phase fault in powerline and how supplying generator reacts to the fault. User can change pre-fault conditions, generator and line model parameters, create fault and observe system currents

Introduction to Electrical Drives

Induction motor

This simulation represents electromechanics of induction motor.
Motor can be accelerated to some speed with given acceleration. This will cause motor power, torque and current to change over time

Frequency converter

This simulation shows how 3-phase AC voltage is created from DC voltage using inverter
The simulator shows how the frequency converter is modulated and what type of current and voltage it will output and how will the load filter and mudulation parameters change that.

Examples and testing

Reference name simulation

This simulation shows 3-phase system transformed to alpha-beta and d-q frames.

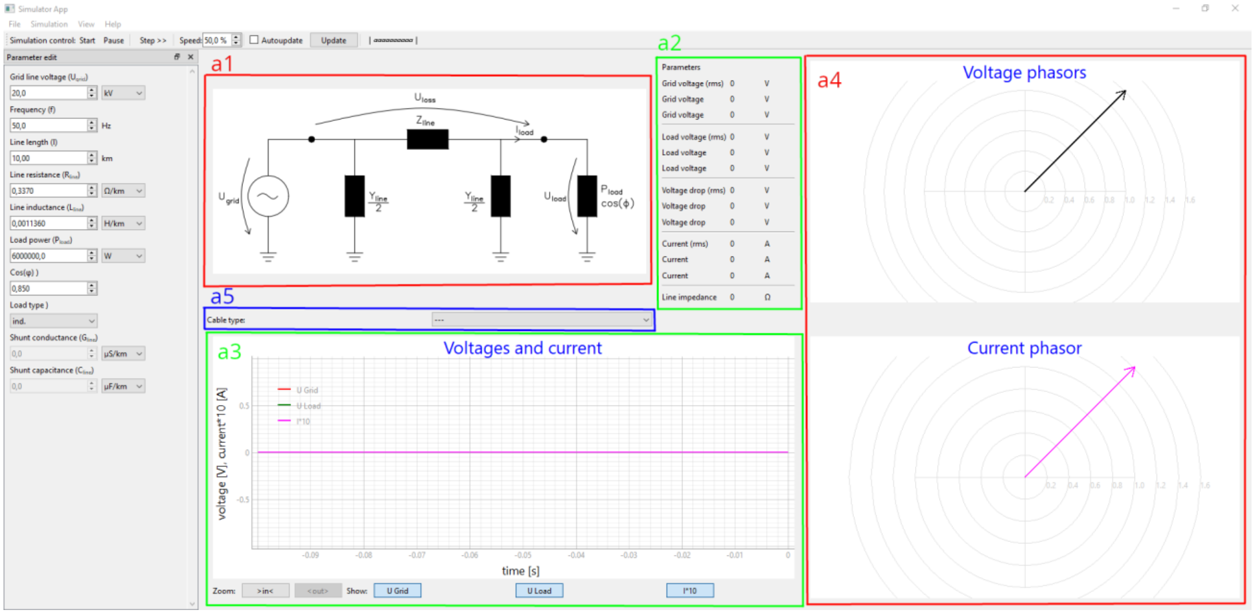
Spring-Mass-Damper simulation

Simulation showing a behaviour of spring-mass-damper system.
The simulation is performed using RK4, trapexoidal approximation and fprward Euler methods and can also be used to compare the accuracy of these solver methods.

Figure 1. Simulator startup menu, where program menu (a), control panel (b), is parameter edit (c).

Simulation view

Simulation view of Voltage drop and losses -simulation is used as an example and can be seen in figure 2, having all the same main elements visible with an addition of the simulation elements visible at the main screen.



Speed: 50,0 %

☐ Autoupdate

Grid line voltage (U_{grid}): 20,0 kV

Frequency (f): 50,0 Hz

Line length (l): 10,00 km

Line resistance (R_{line}): 0,3370 Ω/km

Line inductance (L_{line}): 0,0011360 H/km

Load power (P_{load}): 6000000,0 W

Cos(φ)): 0,850

Load type): ind.

Shunt conductance (G_{line}): 0,0 µS/km

Shunt capacitance (C_{line}): 0,0 µF/km

Cable type: ---

Parameters		
Grid voltage (rms)	0	V
Grid voltage	0	V
Grid voltage	0	V
Load voltage (rms)	0	V
Load voltage	0	V
Load voltage	0	V
Voltage drop (rms)	0	V
Voltage drop	0	V
Voltage drop	0	V
Current (rms)	0	A
Current	0	A
Current	0	A
Line impedance	0	Ω

Figure 2. Voltage drop simulation view, where circuit diagram (a1), parameter view (a2), time-domain graph (a3), phasor-domain graph (a4), dropdown menu (a5).

Simulation main screen elements

The simulation main screen (figure 2, a) consists of five types of elements.

- The circuit diagram of the circuit (a1) is a static picture and does not have any interactive elements.
- The parameter view (a2) has numerical calculated parameters resulting of simulation calculations, these values update at specific times.
- The simulator has two types of graphs, time domain (a3) and phasor domain (a4) graphs. Both types of graphs have similar basic operations.
    - The graphs can be zoomed using the mouse wheel and panned by clicking a dragging.
    - However, the time domain graphs are commonly updated in real time meaning that the graph will reset itself in x axis if simulator is running. For this,

    the time domain graphs have x-axis zoom button which can zoom on the x-axis without it resetting.
    - The time domain graphs also have buttons to toggle individual plot lines on and off.
    - Graphs additionally have right click options which can be used including zooming, view and export options.
- "Cable type" dropdown menu (a5), can be select predefined parameter sets. This type of parameter selections require clicking the update button in the control button after selecting value. This will also overwrite same parameters in the parameter edit.
- The simulation main screen may include other elements, such as buttons or sliders, these elements, behave as expected. Buttons activate something when clicked, and sliders change some value.

Graphs

Graphs can be panned by clicking and dragging and zoomed with mouse wheel. In addition, Graphs have several extra features, most available through the context menu options. Context menu can be opened by right clicking any graph. Context menu is shown in figure 3. A marker can be added to a graph left clicking the graph close to a plotline. The marker is set to the closest data point on the plotline and colored with the same color as the plotline. In addition to this, the x and y coordinates of the marker are shown. The markers are only available in time-domain graphs.

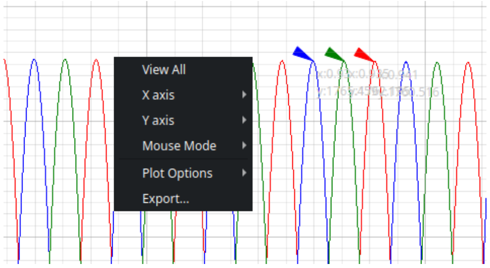


Figure 3. Graph context menu and markers.

The context menu includes the built-in features of PyQtGraph. View All option reframes the graph so that all data is seen on the graph. The X and Y axis submenus can be used to adjust the axis. Mouse mode can be used to switch between three and one button modes. Plot Options have options for transforming the graph, sampling etc. however many of these options are not currently supported as a part of this simulator. And the export menu allows exportation of data from the graph in numeric format (.csv) and visual forms (.jpeg, .png, SVG format).

Parameter edit

The simulator parameter edit element (figure 1, c) includes the editable simulation parameters. These parameters can be edited during the simulation, by changing the value in the parameter edit and updating the simulation, see **Control panel** for more information.

- The parameter values can be set using either the numerical input or by the up and down arrow buttons.
- When the updated values are set to the simulation, depends on the "Autoupdate" setting in the control panel.
  - If autoupdate checkbox is checked, the update is done after each change to the values.
  - If autoupdate checkbox is unchecked, the update is done by clicking the "Update" button in the control panel.
- Some parameters may have a dropdown menu showing the units. This dropdown menu can be used to select the fitting SI prefix for the input, for example instead of writing 10 000 000 V, it may be clearer to write 10 MV.
- Some of the parameters may be disabled (greyed out). These parameters are outside the scope of the intended use of the simulation but can be toggled on from Program menu -> simulation -> advanced -> advanced edit. See more details at **Program menu**.
- The parameter edit is a moveable element, and its position can be changed by dragging from the bar at the top. The element can be moved to the left or right edge of the window or set to float at any location.
- The parameter edit can be closed from the X at the corner of the element, and opened from Program menu -> View -> Parameter edit
- The width of the parameter edit can be changed by dragging at the dot marking at the vertical middle. This may not be possible if the simulation view requires too much horizontal space.

Control panel

The control panel (figure 1, b) has the main controls of the simulation.

- "Start" button starts or continues the simulation depending on its current state.
- "Pause" button pauses the simulation.

- “Step >>” button takes one graphed simulation step forward. This may consist of multiple computed steps.
- “Speed” input changes the current speed of the simulation.
- “Autoupdate” checkbox changes how parameter edit works. If the checkbox is unchecked the parameters only update when “Update” button is clicked. If the checkbox is checked, the parameter will be updated after the value is changed.
- “Update” button is used to update all the changed parameters at once when the “Autoupdate” checkbox is unchecked.
- The last element on the control panel is the running indicator. This shows when the simulator is running, which may be useful in cases when a high count of iterations is required. In this case no other changes may be happening in the window, but the running indicator keeps still running.

Program menu

The program menu (figure 1, a) has four menus in it; File, Simulation, View and Help. Some of the menu buttons are enabled only in certain situations.

File -menu

- “Open” button can be used to open simulation
- “Close” button closes the currently open simulation and returns to startup menu.
- “Export” options currently not available.
- “Settings” submenu has settings for simulation speed and a save button
    - Simulation speed setting is used to adjust the simulation frequency range changeable by the control panel speed setting.
    - Limits set maximum and minimum amount of simulation steps in second.
    - The percentage value in control panel is approximate percentage between these values.
    - The speed values can be edited in cases when user's computer does not have enough computational power to run simulation at the given speed. This may lead to the program not responding (window stops working and goes gray). Lowering maximum speed may help with this problem. On the other hand, a computer with more power can run the program even faster and the maximum can be increased.
    - Setting the maximum above 333 may lead to problems on Windows computers.

    - The changes made will be reset after restarting the program.
    - The save button can be used to save the new values to memory, and the settings will persist after restart.
- "Exit" button exists the program after showing warning.

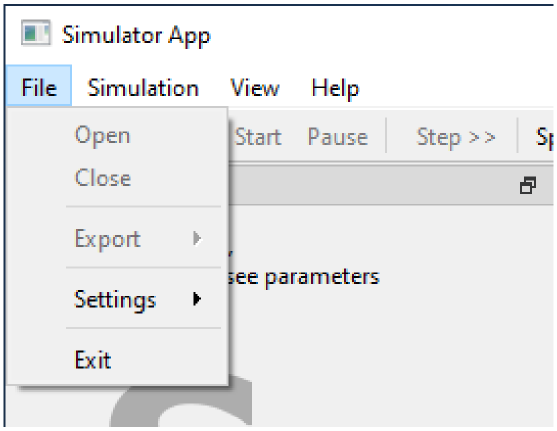


Figure 4. Screen capture of file-menu.

Simulation -menu

- "Start", "Pause" and "Step >>" buttons have same actions as in control panel.
    - "Start" button starts or continues the simulation depending on its current state.
    - "Pause" button pauses the simulation.
    - "Step >>" button takes one graphed simulation step forward. This may consist of multiple computed steps.
- "Reset" button resets the simulation to its original state.
- "Advanced" submenu has two options, Advanced edit and graphing interval.
    - Advanced edit enables editing of all parameters in parameter edit.
    - Graphing interval changes the frequency of simulation main screen elements. The elements, especially graphs, may take a considerable amount of time compared to calculating the simulation steps. However, short step times may be required to accurate modeling. Therefore, not graphing all data points can increase the performance of the program.
    - Graphing interval may be increased if user computer has problems of running the simulator smoothly. Or decreased if available computational power is high enough.
- Simulation step time shows the step time used in the currently open simulation.

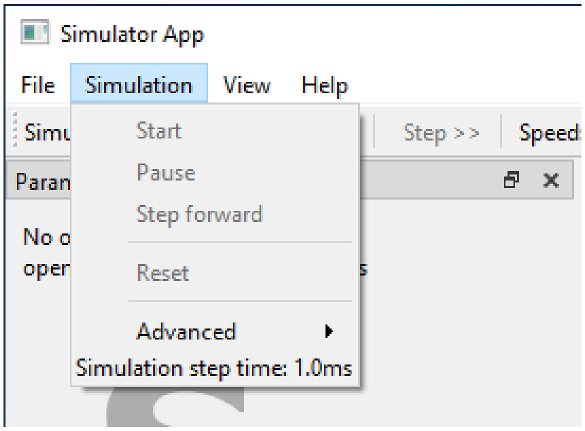


Figure 5. Screen capture of simulation-menu.

View -menu

- Parameter edit re-opens the parameter object element if it is closed.

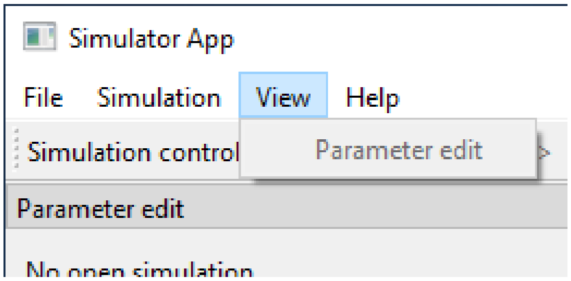
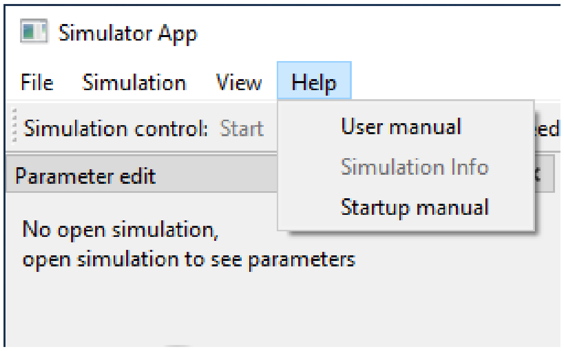


Figure 6. Screen capture of view-menu.

Help -menu

- "User manual" button opens this manual.
- "Simulation info" button opens the info sheet of the currently open simulator.
- "Startup manual" button opens the program installation and startup manual.
- PDFs are opened using the default PDF-reader of user computer.



No open simulation,
open simulation to see parameters

Figure 7. Screen capture of help-menu.

General information

The SFDEsim -simulator is written in Python and uses mostly built in Python modules and libraries; however, it requires installation of three non-standard libraries. These libraries are installed during the setup (see startup manual). These libraries are PySide6, pyqtgraph and NumPy. PySide6 is the official Python module for the Qt for Python project. This library is used for the graphical user interface of the program. Pyqtgraph is a graphics and data visualization module used for all graphing in the simulator. And NumPy is a scientific computation module used computational tasks in the simulator. Installation of these modules is addressed in the Startup manual.

The simulator is build to be modular and to have easy access to creating new simulations or editing already existing ones. The simulator is intended to be published for free use after the completion of the thesis work, under an appropriate license. During the course testing and thesis process all rights are reserved, reading the source code is allowed.

Simulation flow

Simulation flow of the program is separated to simulation on graphing steps. Simulation steps refer to the computational steps when simulating any discretized system and the length of these steps can vary based on the simulation. These simulation steps are not all graphed into the graphs in the simulation view. For every simulation a graphing interval is set, representing how often a computed simulation step is drawn to the graph. This is done since the graphing using these software libraries requires a significant amount of processing power or processor time. This combined with how graphical element updating is handled, updating too many graphs can lead to the program stop responding. Only graphing one in x data points can increase the performance of the simulator without loss of significant amount of data. This way, the number of simulation steps can be kept high to decrease step time and increase simulation accuracy. Graphing interval is primarily set by the programmer of each simulation, however if there are problems with running the simulation smoothly the setting can be changed in simulation menu > advanced.

In addition to only graphing every Xth data point, the simulation flow is artificially slowed down. This slows down can be controlled using the speed setting in control panel and on more advanced level in the settings menu. This slowdown serves two purposes. First the simulation can be run faster or slower depending on what type of situation is observed in the simulator. And second, it can be used to slow down the rate of graphing for previously mentioned performance reasons. The simulation speed change using control panel should be the primary way of changing the simulation speed, but changing the settings can be used to adjust the performance to the level of used computer.

Adding, removing and updating simulations

The program has been designed in a way that simulations are built in a separate files and can be replaced, removed or more can be added without changes to other parts of the program. More information about building a simulation in a separate guide. Simulations are located in PythonModules/Simulation_files folder. The program retrieves the simulations from this directory based on the Simulation_list.json file in the same location. Files in the location can be replaced, for example in a case of an update, without further actions. Adding a simulation requires editing the simulation list file. A .json file can be edited using all common plain text or code editors. The structure of the file follows python dictionary formatting rules. In the dictionary the key structure is as follows:

- Subjects
    - Courses
        - Simulation
            - Filename
            - Abstract

Courses set in which course category the simulation is placed in the startup menu. Simulation separates the different simulations in the course, and this dictionary key is used as the name of the simulation. Filename holds the name of the simulation file. Abstract allows to add a small description of the simulation which is shown in the startup menu. Removing a simulation requires the removal of the simulation from the simulation list. After this the simulation file can be removed from the folder. If the simulation list contains simulations not located in the directory, the simulator will inform the user at launch about missing simulations.

Errors and problems

The simulator can handle most common types of errors, however some errors and problems may arise with a program still in testing phases. Program errors may present as due to a broken or missing module. These conditions are handled by the program and will give an error message for the user. Similar error messages can be shown from a simulation, for example if an iterator fails to converge. The simulator has a log file (log_file.txt), which has a record of different events. Error messages can be read from this log file and can be useful if contacting anyone for support. There is also the possibility of simulation error. Simulation errors may happen and usually lead to a situation where plotting will stop even though the simulation is still running. This requires resetting of the simulation. Please report such errors with an explanation of what happened. If the simulation is choppy, doesn't run smoothly or the program stops responding, slowing down the simulation should help. More about simulation speed in simulation flow section. Changin simulation speed for a long term can be done from Program menu > File > Settings > Simulation speed max, lowering this number and saving the setting should help with choppy simulation or unresponsive program.