

## *Review*

# **A review of tools, models and techniques for long-term assessment of distribution systems using OpenDSS and parallel computing**

**Gerardo Guerra and Juan A. Martinez-Velasco\***

Departament d'Enginyeria Electrica, Universitat Politecnica de Catalunya, Barcelona, Spain

\* **Correspondence:** Email: [martinez@ee.upc.edu](mailto:martinez@ee.upc.edu); Tel: +34934016725.

**Abstract:** Many distribution system studies require long-term evaluations (e.g. for one year or more): Energy loss minimization, reliability assessment, or optimal rating of distributed energy resources should be based on long-term simulations of the distribution system. This paper summarizes the work carried out by the authors to perform long-term studies of large distribution systems using an OpenDSS-MATLAB environment and parallel computing. The paper details the tools, models, and procedures used by the authors in optimal allocation of distributed resources, reliability assessment of distribution systems with and without distributed generation, optimal rating of energy storage systems, or impact analysis of the solid state transformer. Since in most cases, the developed procedures were implemented for application in a multicore installation, a summary of capabilities required for parallel computing applications is also included. The approaches chosen for carrying out those studies used the traditional Monte Carlo method, clustering techniques or genetic algorithms. Custom-made models for application with OpenDSS were required in some studies: A summary of the characteristics of those models and their implementation are also included.

**Keywords:** clustering technique; distributed energy resource; distribution system; genetic algorithm, Modeling; Monte Carlo method; optimization; parallel computation; reliability

---

## **1. Introduction**

Distribution system analysis has been traditionally perceived as the study of small radially-connected systems by means of dedicated simple power flow methods. Actually, modern distribution systems are very complex, and software packages for realistic analysis of distribution

systems should include a large quantity of specialized capabilities [1–3]: Mixture of three-phase, two-phase and single-phase distribution system component models; capabilities to perform simulations of radial and meshed networks over periods of time (e.g., one year); built-in ZIP models of end use loads; representation of distributed generation and energy storage devices.

The list of studies covered by distribution system simulators includes power flow solutions, steady-state short circuit calculations, and reliability analysis. In general, dynamics analysis is not commonly implemented in tools for distribution system analysis, although some include harmonic analysis and capabilities for simulating and coordinating protection devices.

It is assumed that the future (smart) distribution system will offer improved reliability, higher asset utilization, better integration of distributed energy resources (DERs) and new loads (e.g., plug-in hybrid electric vehicles), reduced operating costs, increased efficiency and conservation, and lower greenhouse gas emissions.

An important aspect of the future distribution system would be the high penetration of DERs which can be used for supporting voltage, reducing losses, providing backup power, improving local power quality and reliability, providing ancillary services, and deferring distribution system upgrade [4–6]. Aspects to be considered when analyzing the connection of DERs are the great variety of generating and energy storage devices [5,7], the fact that some DER devices are connected to the utility network via a static converter [8], and the intermittent nature of some renewable sources. Modeling of DERs raises several challenges to distribution power flow calculations since capabilities for representing intermittent generators, energy storage devices, voltage-control equipment, or multi-phase unbalanced systems will be needed. In addition, studies of systems with intermittent generation requires a probabilistic approach and calculations performed over an arbitrary time period that could cover several years.

Load representation is an important issue since voltage-dependence and time-variation of loads must be also accounted time-driven simulations. These issues complicate the study since software tools have to combine new analysis capabilities with a high number of models for representing the various generation and energy storage technologies, besides the conventional distribution system components, and include capabilities for time mode calculations [1].

The essential requirements for future distribution system analysis software may be summarized as follows [1,9,10]: (1) perform fast solutions, (2) be able to capture and process voluminous amounts of simulation results, (3) have robust solution methods for simulating both radial and meshed networks, (4) be able to perform time series simulations with time steps as short as 1 second during periods of several years, (5) solve very large distribution networks, (6) be able to include higher voltage sub-transmission and transmission systems, (7) provide data on end-use patterns for better quantification of distribution system efficiency and improving automation simulations.

This paper summarizes the work carried out by the authors to implement custom-made models and procedures aimed at performing long-term evaluation of large distribution systems using a multicore installation. The goals of the studies were the optimal allocation of distributed generation [11–16], the reliability assessment of distribution systems [17–19], the optimal rating of energy storage systems [20], optimal day-ahead dispatch of a virtual power plant [21], application of a parallel sparse matrix solver to power flow solution algorithms [22], or the analysis of distribution systems with solid state transformers [23].

The simulation tool used has been OpenDSS, a freely available distribution system simulator, whose modelling and calculation capabilities allow users to represent the most important distribution

components and perform studies considering both deterministic and probabilistic calculations [24,25]. OpenDSS is a simulation tool implemented as both a stand-alone executable program and a COM DLL that can be driven from some software platforms. The executable version adds a basic user interface onto the solution engine to assist users in developing scripts and viewing solutions. Built-in solution capabilities include snapshot and time-mode power flow, harmonics, fault current study, dynamics, parametric and probabilistic studies. OpenDSS can be used for distribution planning and analysis, using a general multi-phase AC circuit model, including models for renewable energy resources, energy storage devices, or unusual transformer configurations. Other solution techniques and capabilities (e.g., graphics) can be added through the COM interface.

The paper presents a summary of modeling guidelines to be used in long-term simulations of distribution systems when using a power flow simulator, the tools used and developed by the authors in their studies with OpenDSS, the custom-made models implemented for some studies, the techniques selected for each study and some simulation results aimed at illustrating the results derived from OpenDSS simulations.

## 2. Modeling guidelines

### 2.1. Introduction

Models to be used for representing multi-phase distribution system components should be adequate for power flow calculations under both balanced and unbalanced operating conditions, and can be classified into the following groups [24]:

- (1) *Power delivery components*, whose basic function is to transport energy. Common distribution power delivery components are lines, cables, transformers and voltage regulators. They can be represented by either their admittance or their impedance matrices, depending on the solution technique used for load flow calculations.
- (2) *Power conversion components*, which convert electrical energy to other form of energy, or vice-versa. They are represented as a single multiphase terminal block. This group of components includes generation units, energy storage devices and loads. Their description may depend on the solution technique and the type of simulation. For instance, a generation unit can be modeled as a fixed ideal voltage source, a PV source, a negative PQ load, or as a Thevenin/Norton equivalent.
- (3) *Protective devices*, which monitor the electrical variables (e.g. current or voltage) of circuit elements and isolate them from the system in the occurrence of an abnormal event. Protective devices found in distribution systems include: Circuit breakers, relays, fuses, reclosers, and sectionalizers. These devices can be modeled as either single or multiphase and, except for sectionalizers, their operation corresponds to the time-curves assigned to them; sectionalizers do not operate based on time-curves but are controlled according to the action of upstream protection devices.

The approaches applied to date for representing distribution system components in three-phase load flow calculations are summarized in the following subsections.

## 2.2. Power delivery components

- (1) *Lines and cables*: The most adequate approach is a multiphase pi representation with parameters calculated at power frequency and considering the effect of earth return currents [26]. In general, a Kron reduction is applied to multi-wire models in order to derive a three-phase model. Models for one-phase lines/cables can be also required. Approximate models of overhead distribution lines neglect the effect of the shunt capacitances, or are derived from the positive and zero sequence impedances [26].
- (2) *Transformers*: A significant effort has been made during the last two decades on the development of three-phase transformer models for distribution load flow calculations [27–31]. The approaches developed for modeling the transformer windings compute either the impedance matrix [28,29] or the admittance matrix [27,30,31]. In some solution methods the core is ignored [28], or treated as a load that can be connected at either side of the transformer [27,29,30]. The core model can be represented as the usual linear parallel  $R$ - $X$  model or by empirically-determined nonlinear functions [27].
- (3) *Voltage regulators*: A voltage regulator consists of an autotransformer and a tap-changing mechanism whose position is determined by a control circuit [26]. The implementation of a voltage regulator model must include the control parameters of the compensator circuit (voltage level, bandwidth,  $R$ - $X$  settings, time delay) [32]. Most load flow solutions use a simplified model where the voltage regulator is represented as an impedance in series with an ideal tapped transformer having taps in the secondary. The way in which the tap is determined depends on the solution method [33,34].
- (4) *Capacitors*: They are usually installed to control voltage. A three-phase capacitor bank may be delta- or wye-connected (grounded or ungrounded). In load flow equations, a capacitor bank can be represented as a PV node or as a constant impedance/admittance.

## 2.3. Power conversion components

- (1) *Generation units*: A generator has been traditionally modeled in load flow calculations as a fixed voltage source (as slack node), as a PV source (with known terminal voltage), or as a negative PQ node (with unknown terminal voltage). Although these models neglect the internal impedances, the internal impedances of a rotating machine based unit must be included when calculating unbalanced three-phase power flows since they can have some effect on system unbalance (i.e. injected P and Q of each phase can be different). Several models have been developed for representing a wind generator as a balanced PQ node [35–38], while reference [39] proposed an induction generator model for unbalanced three-phase load flow. A synchronous generator can be represented by its Norton or its Thevenin equivalent [27,31]. Reference [40] proposes single-phase DG models which can be classified as constant power factor, constant voltage, or variable reactive power model.
- (2) *Energy resources*: A first classification considers two groups: Renewable and non-renewable resources. However, for power system calculations it is better to classify these resources as random (or intermittent) and non-random. All intermittent resources (wind, sun) are renewable, but not all renewable resources are intermittent (e.g., biomass is renewable but not intermittent).

*Wind*: It is basically characterized by the wind speed. Several approaches have been used for wind speed forecast, but in general either a Weibull or a Rayleigh probability density function is used [41,42]. The wind resource is affected by the air density and the nature of the turbulence, as well as the auto-correlation factor, the diurnal pattern strength, and the hour of peak wind speed. Several approaches have been developed to obtain wind power from wind forecast [35,41–46]. However, wind turbine output can be strongly correlated among adjacent wind farms due to the similar wind speed at the area [44–46].

*Sun*: It is characterized by the measured insolation and some site parameters, which may be geographical (site latitude and longitude, ground reflectance) and environmental (ambient temperature). Insolation may be expressed as either average total solar irradiance on the horizontal surface ( $\text{kW/m}^2$ ) or average clearness index [41,42,47–49]. This index is the ratio of the total horizontal radiation at ground surface to the extraterrestrial radiation. The insolation on the panel surface is composed of direct and diffuse radiation, each of which depends on the clearness index.

- (3) *Loads*: Actual loads may be unbalanced, are time-varying, and exhibit some randomness and voltage dependence [50]. Aggregate load models are usually balanced; however, single-phase tapped lines add unbalance. The statistical characterization of loads, including phase unbalance, can be derived from measurements, assumed to be Gaussian [51] or beta distribution [52]. Deterministic detailed residential models representing individual appliances and human behavior patterns were presented in [53–55].

The electric vehicle is a load type that could dramatically change the analysis and design of distribution systems. The model of an electric vehicle has to include a battery model and a statistical distribution of the number of plug-in vehicles in one node. Although several battery and electric vehicle models are available, not many have been developed for load flow studies. The model presented in [56] represents the electric vehicle load as a PQ bus with stochastic behavior based on the queuing theory.

- (4) *Storage devices*: Although the characterization of energy storage devices is not an easy task due to the various storage technologies [57–60], rather simple models can be used [61–63]. The main characteristics of an energy storage unit are energy capacity, charging and discharging power capacity, charging and discharging efficiency, minimum and maximum state of charge, and control strategy. An energy storage device must be linked to a control algorithm that will decide the power flow and its direction. A simple control strategy can be used when an intermittent source is connected to a load and an energy storage unit; the load power is compared to the source power, and the difference is fed or extracted from the storage unit. In reality, the control strategy can be much more complex, mainly when the energy storage unit is used for selling and purchasing energy.

#### 2.4. Protective devices

- (1) *Circuit breakers*: A circuit breaker is a switching device designed to open and close a circuit by non-automatic means and to open the circuit automatically upon a signal sent by an external controller [64]. Closed switching devices can be modeled as a small impedance or by internally fusing the buses connected by the switch [65]. On the other hand, a branch connected to an open

switch can be modeled by disabling the branch element, switching it to an auxiliary bus, or forcing a zero-current on the terminal connected to the open switch [65,66].

- (2) *Relays*: They are external controllers used to automatically operate circuit breakers and are characterized by their time-curve [24]. Time-curves reflect the switch opening time as a function of the monitored electrical variable; opening times are defined to prevent any damage to the system elements or the general public. Time-curves can be device-specific (developed by the device's manufacturer) or calculated according to a set of equations, see [67]. Relays can be used to monitor voltages, currents, or any other internal element variable that is relevant to the system's protection scheme.
- (3) *Fuses*: Although real fuse operation is determined by two different time-curves (minimum melting time and maximum clearing time), their representation follows the same principle as relays; that is, it is a switching device whose operation is defined by a single time-curve [24]. However, care must be taken that the assigned time-curves reflect the real behavior of fuses (see the K and T-link curves provided with OpenDSS).
- (4) *Reclosers*: They are switching devices that have the capability to return to their original closed position after a waiting time (referred to as dead time) has passed since they opened [68]. A recloser can perform several opening operations (usually between 2 and 4) before it locks out in an open position. Reclosers are defined by two time-curves (one fast and one slow), the number of opening operations (according to the fast and slow time-curves), and dead time between the opening and reclosing operation [24].
- (5) *Sectionalizers*: Line sectionalizers are switching devices that do not possess the capability to interrupt fault currents; therefore, they must be coordinated with an upstream recloser [68]. A sectionalizer counts the number of times the monitored current surpasses a threshold value and locks out in an open position after a predefined number of operations have occurred (usually 2 or 3). The opening operation is performed during the upstream recloser's dead time.

### 3. Tools

The analysis of modern power distribution systems requires the use of different tools that cover a wide range of tasks within the solution scheme. Among the tasks that need to be completed are: Solution of the electrical system, data collection and handling, random number generation, data visualization, and high performance computing. This section presents some of the tools used by the authors to develop applications for the analysis and solution of power distribution systems.

#### 3.1. OpenDSS

OpenDSS (Open Distribution System Simulator) is a power distribution system simulator released by EPRI (Electrical Power Research Institute). It is a frequency-domain simulation engine with many characteristics found in other commercial simulation tools, as well as many additional features aimed at supporting ongoing research efforts on distribution system simulation. Built-in solution capabilities include snapshot and time-mode power flow, harmonics, fault current study, dynamics, parametric and probabilistic studies [24]. OpenDSS can be used for planning and analysis of multi-phase distribution systems, analysis of distributed generation interconnection, annual simulations, storage modeling and analysis, and other studies. OpenDSS was designed to receive

instructions via scripts, allowing greater flexibility to the user. The program can be accessed through a stand-alone executable version and the COM interface implemented on the in-process server DLL. The stand-alone version provides a text scripting interface, which allows a complete interaction with the program. OpenDSS can be linked to other software platforms (e.g. MATLAB) using the COM interface; it provides the same functionalities as the stand-alone version in addition to greater control and data handling capabilities.

### 3.2. MATLAB

MATLAB (MATrix LABoratory) is a well-known software tool used in engineering and scientific analysis and design [69]. Designed for numerical computing, it integrates computation, visualization, and programming environments. MATLAB's matrix-based programming language provides a natural approach to computational mathematics: Solving systems of linear equations, computing eigenvalues and eigenvectors, factoring matrices, and more. The programming environment allows users to develop their own scripts as well as perform operations through the interactive console; furthermore, users can make use of the built-in editing and debugging tools in order to improve their applications. In addition to user-developed functions, MATLAB toolboxes offer a large number of pre-compiled routines that can be readily included into the user's application. These routines cover areas such as machine learning, signal processing, image processing, computer vision, communications, computational finance, control design, robotics, and others.

### 3.3. Multicore for MATLAB

Parallel-oriented software is designed to exploit multiple computer resources (multiple CPUs); the solution of a computational problem requires partitioning it into discrete parts that can be solved simultaneously [70]. In general, these problems (or jobs) can be categorized as follows [71,72]: Task-parallel jobs, Data-parallel jobs, Job Scheduling.

MATLAB's native capabilities can be used to execute parallel applications in a multicore installation [73–75]; however, this usually requires purchasing special toolbox licenses. The “Multicore for MATLAB” library developed by M. Buehren consists of a set of *.m* files which provides the user access to all cores in a multicore installation [76]. This library can be used to execute task-parallel jobs by distributing task executions among system cores in an equivalent manner to MATLAB's *parfor* function. In order to access all available cores one MATLAB session must be run for each parallel execution; one session will serve as master, whereas the rest will work as slave sessions. The maximum number of concurrent MATLAB sessions must be equal to the number of cores at the user's disposal.

The use of the “Multicore for MATLAB” library is not constrained to a single multicore computer; its capabilities can also be used to control a group of computers (i.e. a computer cluster); this possibility will grant the user access to larger computational resources. A computer cluster can be easily expanded by adding more nodes to its original structure, thus increasing the number of available cores for task executions. This increment in available cores will allow the user to achieve a greater reduction in execution times, since the number of task executions per core will be drastically reduced.

### 3.4. Intel MKL and Intel MKL PARDISO

The Intel Math Kernel Library (MKL) is a package of highly optimized and extensively threaded routines for high-performance applications [77]. Optimized for Intel processors, Intel MKL provides FORTRAN and C/C++ programming interfaces, as well as a DLL that can be used for software redistribution. Capabilities included in the library cover areas such as linear algebra routines, distributed processing linear algebra routines, sparse solver routines, fast Fourier transform functions, vector mathematics routines, data fitting library, and eigensolvers. Intel MKL PARDISO is a direct sparse solver routine based on the PARDISO solver [78], a high-performance software package for solving large sparse linear systems of equations on shared-memory multiprocessors [79]. It can cope with a variety of sparse matrices, including real and complex, symmetric and nonsymmetric, positive definite and indefinite sparse linear systems of equations. For sufficiently large-size problems, numerical experiments demonstrate that the scalability of the parallel algorithm is nearly independent of the shared-memory multiprocessing architecture. The execution of Intel MKL PARDISO can be divided into the following phases: (i) fill-reduction analysis and symbolic factorization; (ii) numerical factorization; and (iii) forward and backward solve. These stages can be executed in one sequence or individually at different times.

## 4. Power distribution system models

Modeling and simulation of power distribution systems raises several challenges to load flow calculations since capabilities for representing intermittent generators, voltage-control equipment, or multi-phase unbalanced systems are required. This section introduces both native OpenDSS element models and custom-made models that can be used in different studies of power distribution systems. Furthermore, it presents three algorithms aimed at the generation of system curve shapes, which can be used in OpenDSS time-driven simulations.

### 4.1. OpenDSS built-in models

A great number of element models are available in OpenDSS; among those models the user can find power conversion elements, power delivery elements, protection devices, and control models [24]. All element objects are initialized with a default set of parameter values; when a new instance of an element is created, the default values are overwritten by those specified in the new instance definition. Due to the default parameter initialization, it is not necessary to define all parameter values for a new element, only those that are necessary. Tables 1 to 4 present some of the most important models available in OpenDSS as well as a short description and the main configuration properties [24].



**Table 1.** Power conversion models.

Object	Description	Parameters
Load	It represents the power consumption of system customers. It is defined by its nominal kW and PF or its kW and kvar.	bus1, conn, kV, kW, pf, model, phases, yearly.
Generator	Its rating is defined by its nominal kW and PF or its kW and kvar. For power flow studies, the generator is usually modeled as a negative load that can be dispatched.	bus1, conn, kV, kW, pf, model, phases, yearly.
PVSystem	It aims to replicate the behavior of a photovoltaic generator. The model output will depend on the solar irradiance, panel temperature, and panel and inverter efficiency.	bus1, conn, irradiance, kV, kVA, Pmpp, pf, yearly, Tyearly.
Storage	It is a generator that can be dispatched to either produce power (discharge) or consume power (charge) within its power rating and stored energy; it can either act independently or be controlled by a Storage Controller element.	bus1, conn, kV, kVA, kWhrated, DispMode, %reserve, EffCharge, EffDischarge.

**Table 2.** Power delivery models.

Object	Description	Parameters
Line	It is used to model most multi-phase, two-port lines or cables. Impedances may be specified by symmetrical component values or by matrix values. Alternatively, the user may refer to an existing LineCode or Geometry object.	bus1, bus2, phases, R1, X1, C1, linecode, geometry.
Transformer	It is implemented as a multi-terminal power delivery element; it consists of two or more windings, each defined by one of the available connections (Wye or Delta).	buses, conns, kVs, kVAs, phases, windings, Xhl, %Rs.
Capacitor	It is implemented as a two-terminal. If a connection for the second bus is not specified, it is by default the ground reference of the same bus to which the first terminal is connected.	bus1, conn, kV, kvar, phases.

#### 4.2. Custom-made models

Despite the great number of native models, some studies require new custom-made models that are not available in OpenDSS; these new models can belong to any of the existing model categories (e.g. power delivery elements, protection devices). Custom-made models can be classified as: (i) compiled models, and (ii) external models. Information on the technical details for the implementation of custom-made models can be found in [66].

##### 4.2.1. Compiled models

OpenDSS is an open source software, which grants users access to the source code and the possibility to modify and expand the capabilities of the original package. Compiled models refer to those models that have been included in the source code; as a result, they are compiled within the program and can be defined, modified, and accessed in the same manner as other built-in objects. This section presents the custom-made compiled models that have been developed by the authors.

**Table 3.** Protection device models.

Object	Description	Parameters
Relay	It is a protective device that is connected to one terminal of a circuit element; it can monitor currents and voltages at the terminal to which it is connected. It uses the current and voltage values to operate the terminal switches of the switched circuit element; it also has the capability to perform reclosing actions. The monitored and switched circuit element must be explicitly defined.	kvbase, MonitoredObj, MonitoredTerm, Phasecurve, PhaseTrip, RecloseIntervals, Shots, SwitchedObj, SwitchedTerm, Type, Overvoltcurve, Undervoltcurve.
Fuse	It is a protective device that controls the terminal switches according to the current flowing through the monitored element. As in the Relay object, the monitored and switched element must be defined explicitly.	FuseCurve, RatedCurrent, MonitoredObj, MonitoredTerm, SwitchedObj, SwitchedTerm.
Recloser	Similar to the Relay and Fuse objects, it controls the terminal switches of the switched element according to the current flowing through the monitored element. The Recloser object also has the capability to perform reclosing actions; furthermore, two different time-current curves can be defined (one fast and one slow/delayed).	Numfast, MonitoredObj, MonitoredTerm, SwitchedObj, SwitchedTerm, PhaseTrip, RecloseIntervals, Shots, PhaseFast, PhaseDelayed.

**Table 4.** Other models.

Object	Description	Parameters
Vsource	It is a two-terminal, multi-phase Thevenin equivalent. The data are specified as for a power system source equivalent: Line-line voltage (kV) and short circuit MVA.	angle, basekv, bus1, frequency, phases, pu, MVAsc1, MVAsc3.
Loadshape	It consists of a series of multipliers (typically ranging from 0.0 to 1.0) that are applied to the base kW values of the load to represent variation of the load over some time period.	mult, npts, interval, hour, UseActual.
GrowthShape	It consists of a series of multipliers that represent the nominal load growth for time periods that span over several years.	mult, npts, year.

#### 4.2.1.1. Storage

OpenDSS provides a technology-independent energy storage (ES) model that can be employed to perform planning studies under different operation modes [24]. The use of this model allows the user to set limits for the storage's power capacity and stored energy; moreover, it automatically calculates losses and updates the stored energy value.

The Storage model has been modified to add new features to its native capabilities [20]:

- The efficiency is not assumed constant but defined by a function that depends on its loading with respect to its power rating (in kVA).
- A minimum charge/discharge power as a percentage of the power rating (in kVA) has been defined. Regardless of the dispatched power, the storage element will not consume/produce power unless this value is greater than the minimum charge/discharge power.

- The model defines a maximum charge/discharge power based on the stored energy and the simulation time-step. The objective is aimed at ensuring that the minimum/maximum stored energy values are not violated. For example, assume the present stored energy is 100 kWh and the minimum acceptable value is 80 kWh; given a 1-hour time-step and a discharge efficiency of 100%, the ES unit should only be able to inject a power of 20 kW for a period of 1-hour. Regardless of the predefined dispatched power, the model will enforce the 20 kW limit.

#### 4.2.1.2. Virtual power plant

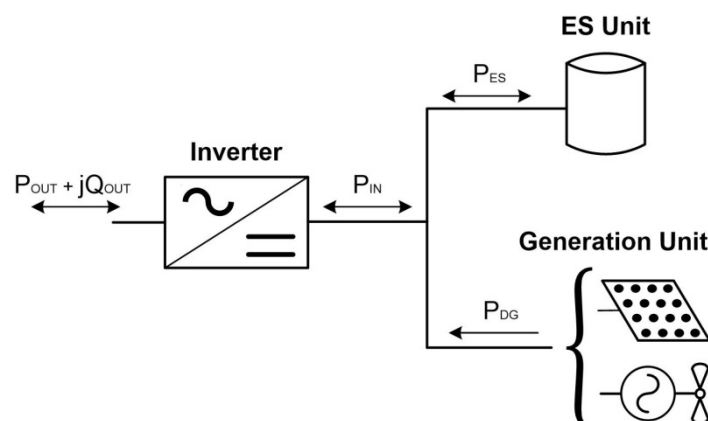
The Virtual Power Plant (VPP) object is the aggregation of energy storage and renewable generation (either solar or wind) models; see Figure 1 [21]. It is based on the built-in Storage object element and is adequate for distribution system studies and quasi-static (time-driven) simulations.

The VPP is composed by a generation unit, an energy storage unit, and an inverter; see Figure 1. The generation unit can be either solar or wind generation. Solar generation is modeled after the OpenDSS PVSystem object, while wind generation is based on the built-in Generator object and its output power is calculated according to [14].

Two different operation modes have been implemented inside the model: Follow and Dispatch. Under the Follow mode the model behaves according to the curves assigned to the generation unit and Storage object, whereas in Dispatch mode the VPP acts as a dispatchable unit.

#### 4.2.2. External models

OpenDSS can be linked to other software platforms (such as MATLAB, Python, C#, and other languages) using the COM interface of the in-process server DLL. External models rely on this interface to implement custom-made features, models, and solutions. The following models have been developed by the authors making use of MATLAB as a tool to control built-in OpenDSS objects via the COM interface.



**Figure 1.** Basic configuration of the Virtual Power Plant.

#### 4.2.2.1. Solid state transformer

The Solid State Transformer (SST) is modeled using two separate objects: LV voltage sources and MV loads, which will be respectively referred to as SST load and SST voltage source [23]. The SST voltage source provides the active and reactive power demanded by the LV loads while maintaining a constant voltage value at the secondary terminal (e.g. 1 pu), whereas the SST load demands only active power from the primary MV side. Since active power can flow through the SST in two directions, two operating modes are distinguished: *Load* and *Generation* modes. Figure 2 shows the diagram corresponding to a single-phase SST running in *Load* mode.

The procedure implemented in MATLAB-OpenDSS to obtain a load flow solution when at least one SST is included in the system model may be summarized as follows:

- (1) Solve the system using initial “dummy” values for the SST loads (see Figure 2); the initial values can also be inherited from a previous solution. These initial values will not interfere with the loads connected to the secondary terminal of the SST, as they are independently served by the SST voltage source.
- (2) Read the total active and reactive power provided by the SST voltage source.
- (3) Calculate the power demanded by the SST from the MV network by adding losses.
- (4) Update power values for the SST loads.
- (5) Solve the system with correct values.

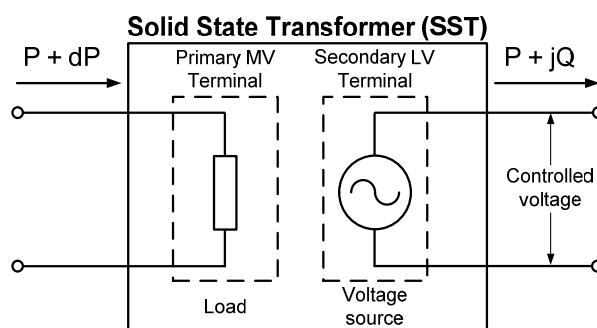
If the system solution is required as a function of time, this procedure is repeated for every time step.

#### 4.2.2.2. Sectionalizer

The Sectionalizer model is based on a Fuse object that is externally commanded by a control algorithm implemented in MATLAB. The control algorithm is responsible for monitoring the current flowing through the protected line, counting the number of upstream operations, and sending the opening signal via the COM interface. The model was implemented for time-driven mode in order to provide a correct coordination with the upstream protection devices during transient simulations.

The most important aspects of the implementation are:

- The current flowing through the distribution lines, to which the sectionalizers are connected, must be monitored for every simulation time-step.



**Figure 2.** SST model as implemented in OpenDSS (SST operates in Load mode).

- The control algorithm tracks the number of times the monitored current goes over and falls back under a predetermined threshold; see Figure 3.
- The time-curves assigned to the Fuse objects have been designed to prevent fuse operations during a fault.
- More than one sectionalizer can be installed on the same lateral circuit. Under this condition, the control algorithm is capable of identifying which sectionalizer must operate according to its location with respect to the fault.

#### 4.3. Load and generation curve shapes

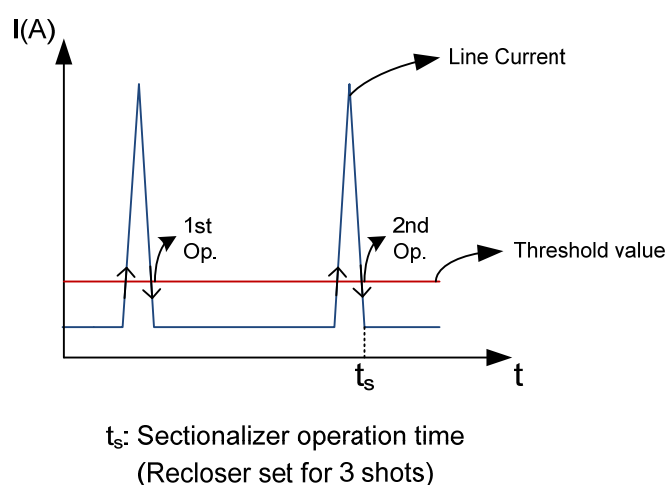
Time-driven (or quasi-static) solutions permit a more realistic representation of power distribution systems, since they allow the user to carry out simulations where system element behavior (e.g. load and generation) varies with time. In addition to system equipment (lines, cables, transformers, voltage regulators, loads, capacitors, generators, etc.), time-driven solutions require information related to system performance over the evaluation period. This performance will be described by a set of curves that will dictate the behavior of loads, generators, and other elements in the system. The following algorithms have been implemented in MATLAB by the authors for the generation of system curve shapes.

##### 4.3.1. Node load profiles

Load profiles describe hourly behavior of system node demands. They are generated using realistic base load curves, including the combination of different types of curves. Base load curves represent typical behavior of different types of customers. New load curves are synthetically generated to have different sample curves that follow the same general pattern but present differences in hour-to-hour values; they are based on realistic base load curves, which serve as starting point to ensure the new curves reflect the behavior of real costumers.

The procedure for the synthetic generation of load curves is as follows [80]:

- (1) Choose a base load curve.



**Figure 3.** Sectionalizer operation sequence.

- (2) Determine yearly trend using Fourier series.
- (3) Obtain weekly cycles using average values.
- (4) Reconstruct daily average cycles.
- (5) Generate an approximate curve overlaying the effect of yearly trend, weekly cycles, and daily average values.
- (6) Calculate the error of the approximate curve.
- (7) Estimate the error's probability density function (PDF) for every hour and month of the year.
- (8) Generate random error values using the error PDF.
- (9) Aggregate generated errors to the approximate curve.

System node demands can consist of different types of customers and their consumption pattern must account for the contribution of all customers. Load profiles will be generated by combining synthetically generated load curves, which represent each of the customers comprising the node demand.

The steps to create new load profiles are as follows [14]:

- (1) Establish what percentage of nominal power is assigned to each type of customer. Customers are represented by their base load curves.
- (2) Synthesize load curves used for the new load profile.
- (3) Create the new load profile by calculating the weighted sum of the generated load curves.
- (4) Define an hourly variation factor.
- (5) Randomly generate variation factors for every hour using a uniform distribution.
- (6) Multiply new load profile hourly values by the randomly generated variation factors.
- (7) Normalize load profile values.

#### 4.3.2. Wind generation curves

Wind generation curves are obtained from wind speed curves. These curves may be synthesized by adding a deterministic component, which defines the general annual trend, and a stochastic component, responsible for introducing a random element in hour-to-hour values [81]. The resulting wind speed values are adjusted to match statistical properties of the evaluation site. In addition, wind speed values must be corrected to match the actual height of the wind turbine. The power generated by the turbine/generator is calculated using wind-power curves. Finally, the turbine power is adjusted to take into account the actual air density.

The main steps for generating wind generation curves are [14]:

- (1) Calculate deterministic curve of wind speed values.
- (2) Generate wind curve's stochastic component.
- (3) Sum of deterministic and stochastic components.
- (4) Final probability density function transformation.
- (5) Height correction.
- (6) Calculation of generated power.
- (7) Air density correction.

Input parameters for the estimation of wind generation curves are: Reference height for wind speed measurements, site altitude over sea level, monthly average wind speed values, site roughness length in wind direction, wind speed curve characteristics, generator rating in kW and hourly ambient temperature values.

### 4.3.3. Photovoltaic generation curves

Photovoltaic generation curves are generated by providing the OpenDSS built-in PVSystem object with the corresponding irradiance and panel temperature curves for the evaluation period under consideration; the PVSystem object calculates the generated power according to the values in these curves.

#### 4.3.3.1. Generation of the yearly solar irradiance curve

The following steps have been implemented to generate hourly values of incident irradiance on a tilted surface [12]:

- (1) Calculation of the daily clearness index.
- (2) Calculation of the hourly clearness index for a given day.
- (3) Calculation of the hourly solar irradiance.

#### 4.3.3.2. Generation of the yearly curve of panel temperature

The procedure implemented to obtain the panel temperature is divided into three steps [12]:

- (1) Calculation of minimum and maximum daily temperatures.
- (2) Calculation of hourly temperatures.
- (3) Calculation of average panel temperature.

To obtain these curves the user has to specify the average monthly clearness index, the slope angle and the geographical coordinates (latitude, longitude, and time zone) of the panel, the normal operating cell temperature and the average monthly values of the minimum and maximum temperatures.

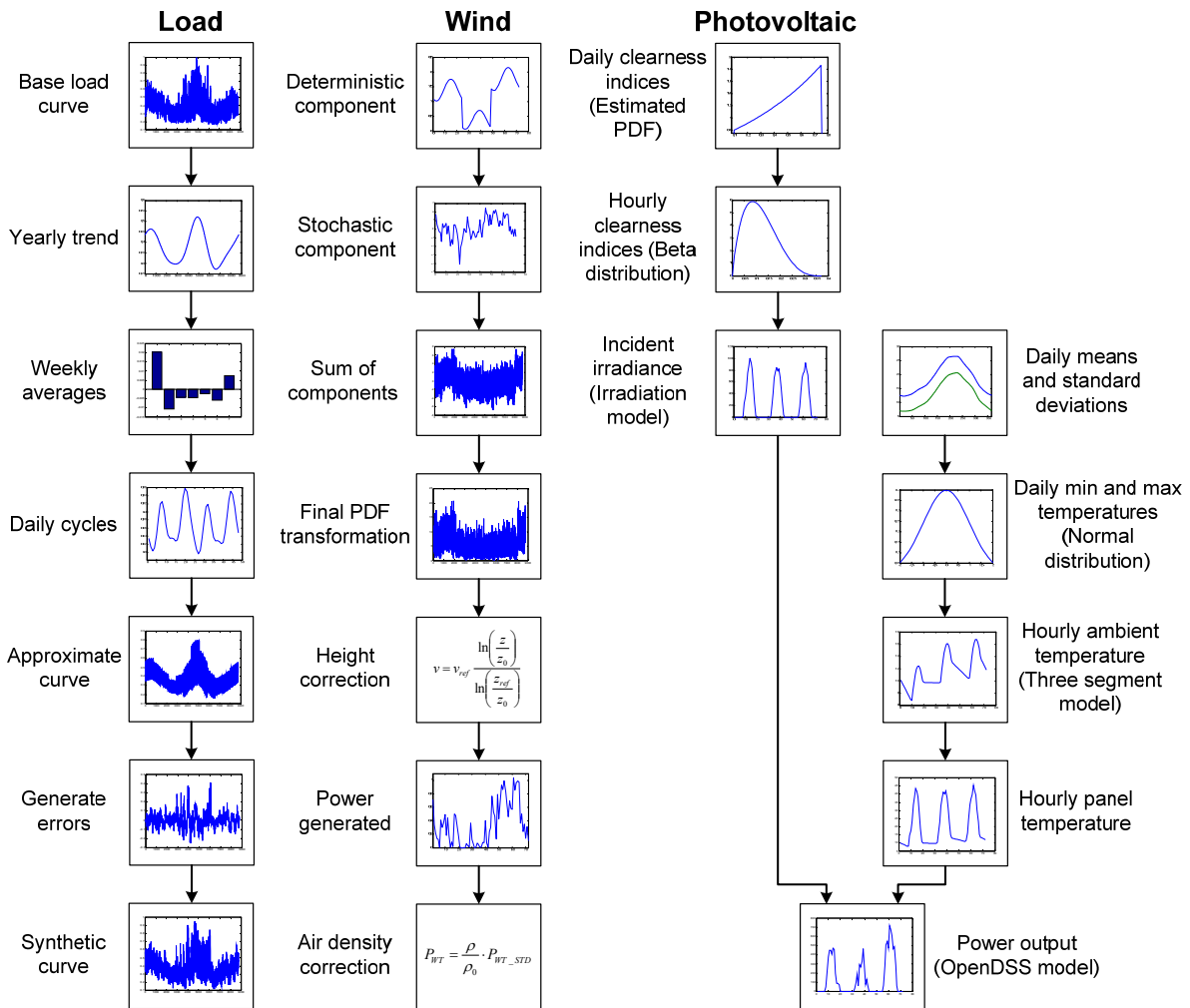
Figure 4 illustrates the main steps implemented in the development of the three algorithms for the generation of system curve shapes.

### 4.3.4. Case study

Figure 5 shows the diagram of the test system used to check the usefulness of the developed procedures. Some important numbers about the system are given below:

- Rated system voltage: 4.16 Kv.
- Rated power of substation transformer: 5000 kVA.
- Number of load nodes: 51.
- Initial maximum coincident load measured at the substation: 1700 kVA.
- Overall line length: 26.36 km.

The study is aimed at analyzing the long-term impact of the DG on the system performance; that is, knowing the yearly load shapes at all system nodes and the yearly variation of loads, the goal is to estimate the resulting voltage values and energy losses, or the yearly energy that will be demanded from the HV system after connecting DG units.



**Figure 4.** Generation of yearly curve shapes for distribution system analysis.

The ratings of the DG units, their locations and the year of connection are as follows:

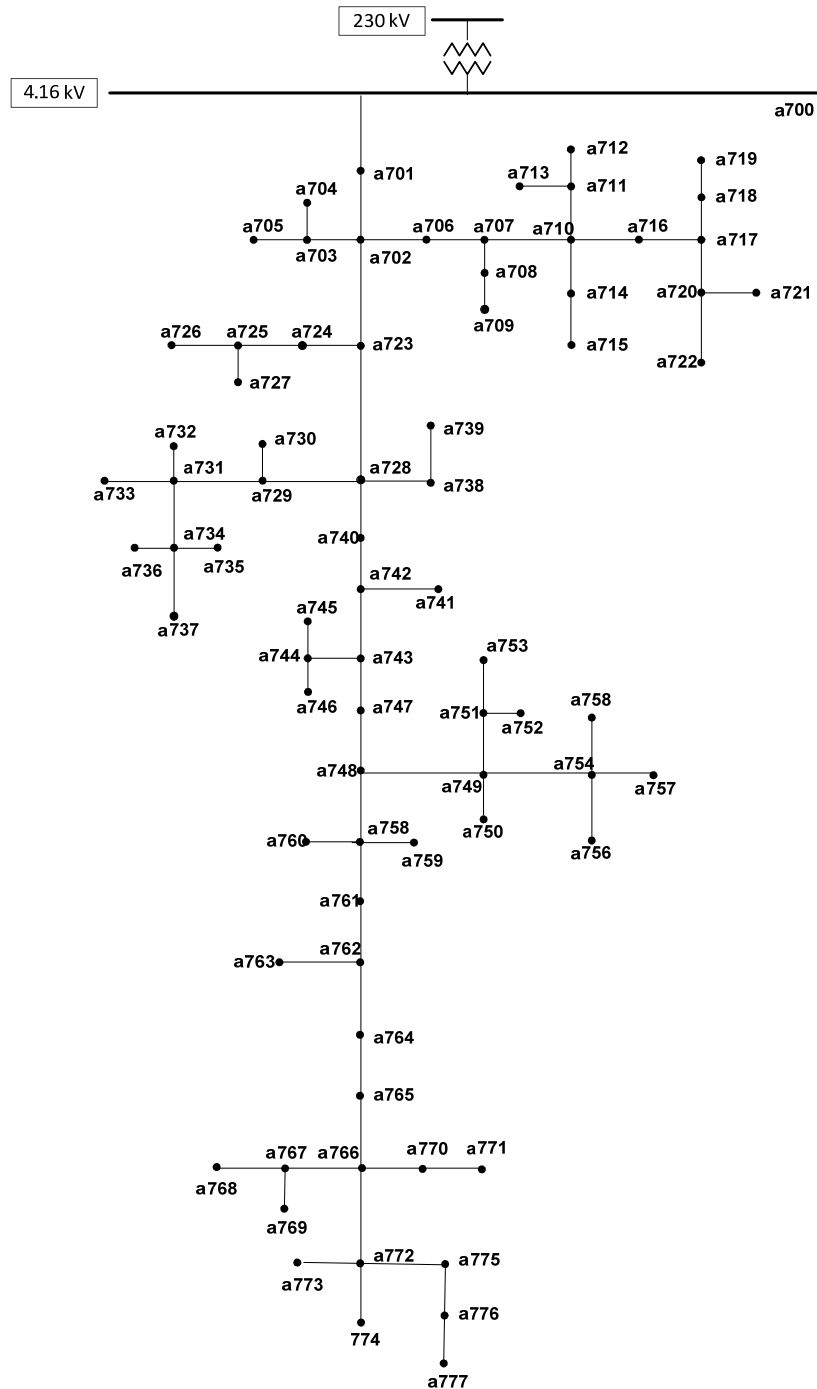
- Unit 1: PV, 100 kVA, connected at node A716, year 2.
- Unit 2: Wind, 150 kVA, connected at node A728, year 3.
- Unit 3: PV, 150 kVA, connected at node A758, year 5.
- Unit 4, Wind, 100 kVA, connected at node A766, year 6.

Given that generation units by independent producers can be randomly located, as long as they fulfill the interconnection requirements, the main results of the study can be useful for the utility to decide whether the ratings and locations of the DG units are adequate or not.

The study will be carried out for a period of ten years, and assuming the predicted variation of the load is not the same for all nodes. As for the distributed generators, they will not be simultaneously connected to the system.

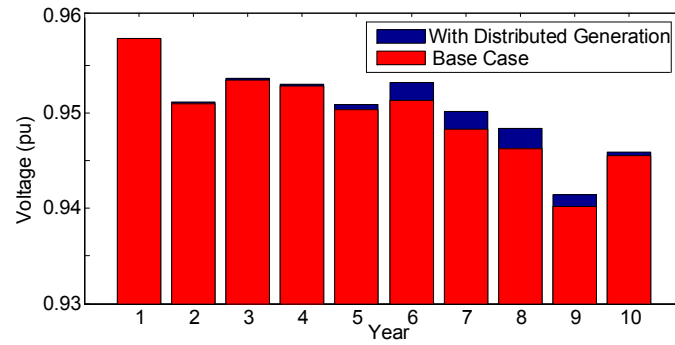
It is assumed by default that the connection of each unit is made at the beginning of the corresponding year, and the generators will only inject active power.



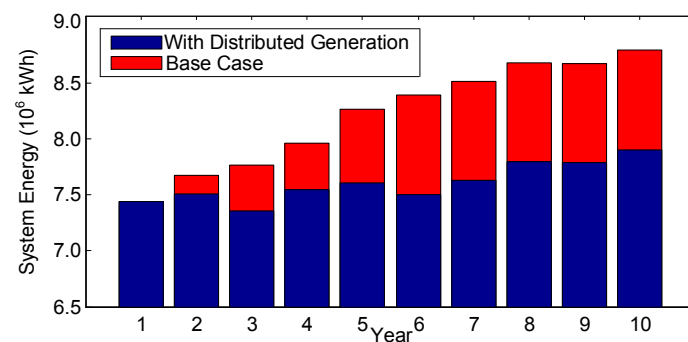


**Figure 5.** Diagram of the test system.

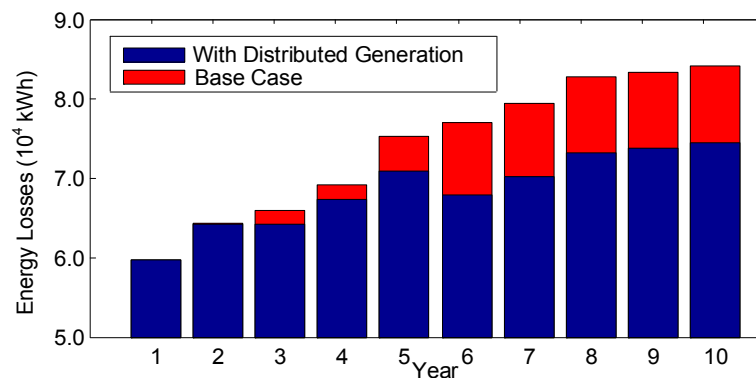
Figures 6 to 8 show some results of the study when using a constant power model for representing node loads. It is obvious from these results that, as expected, there is a positive impact of the DG on the node voltages (although the minimum voltage at certain year is below 0.95 pu even with DG), and a significant reduction in the system losses and in the energy required from the HV system is achieved. Note that the red bars, which refer to the “Base Case”, are plotted behind the blue bars (“With Distributed Generation”). For the first year of the study, both bars present the same height and therefore, only the blue bar can be observed. Starting from the second year, the introduction of DG creates a difference between them, which makes the red bar visible.



**Figure 6.** Impact of DG on the minimum voltage values.



**Figure 7.** Energy required from the HV system (measured at the secondary substation terminals).



**Figure 8.** Energy losses (without considering substation losses).

Some interesting conclusions can be derived from the results presented in these figures. For example, variations in minimum voltage values are a consequence of randomness in load and generation values. Furthermore, this random behavior causes a non-uniform growth of annual energy and energy losses. On the other hand, in spite of randomness, the contribution of the generators remains basically constant once that last generator has been connected.

## 5. Solution techniques

### 5.1. OpenDSS solution methods

OpenDSS has several solution methods that allow users to apply this tool to a wide range of studies [24]. These solution methods present many configuration options, which give the user great control over them. Moreover, Monitor and EnergyMeter elements can be used to collect all the necessary information resulting from the study under consideration.

A short summary of the tasks the OpenDSS capabilities allow to perform is presented below.

#### 5.1.1. Power flow calculations

OpenDSS is designed to perform a basic distribution-style power flow in which the bulk power system is the dominant source of energy. However, unlike traditional radial circuit solvers, it can cope with both radial and meshed networks; thus, it can be used for the solution of circuits that include a representation of transmission or subtransmission systems.

The circuit model employed can be single-phase, full multi-phase model or a simplified positive sequence model, while loads and generators can be modeled as injection sources or admittances in the system admittance matrix. The power flow can be executed as a single Snapshot or Time-driven solution, where the load varies as a function of time. Time-driven solution depends on the evaluation period and includes the following modes: Daily, Yearly, Time, and Dutycycle. Furthermore, the circuit can be solved assigning random values to system loads under the Monte Carlo mode, see [14].

#### 5.1.2. Fault studies

OpenDSS can be used to perform short-circuit fault studies. This type of studies can be done following different approaches:

- Faultstudy mode: The program places short-circuit faults in all system buses and returns the resulting currents and voltages.
- Single fault: The user places one or more fault elements in the circuit. The program solves the circuit considering the user-defined faults.
- MF mode: The circuit is solved considering randomly applied faults throughout the system.

#### 5.1.3. Harmonic flow analysis

OpenDSS is a general-purpose frequency-domain circuit solver and can perform solutions for other frequencies than the fundamental. The frequency behavior of power conversion elements (e.g. loads, generators, voltage and current sources, etc.) is modeled using various harmonic spectra. The program can also perform frequency sweeps for user-defined frequencies (expressed as harmonics of the fundamental); the contribution of power conversion elements to the harmonic solution is calculated according to the assigned spectra.

#### 5.1.4. Dynamics studies

OpenDSS can perform basic electromechanical transients, or dynamic, simulations using a simple predictor-corrector method with one correction step. This capability of OpenDSS has been steadily improved due to needs in inverter modeling and other applications where machine dynamics are important. The generator object uses a single-mass model for dynamic simulations; however, users can develop more sophisticated models and compile them into DLLs for their use with OpenDSS.

#### 5.1.5. Protection studies

OpenDSS can be used to perform coordination studies for protection devices; these studies can include built-in models and new custom-made devices developed by the user. Coordination studies can be conducted considering steady-state fault currents and a static control for protection devices, that is, simulation time is increased internally and the user only observes the system's initial and final state. Furthermore, dynamic simulations can be carried out to calculate the generators and other components' contribution to the transient fault currents and observe the protection system's operation as a function of time.

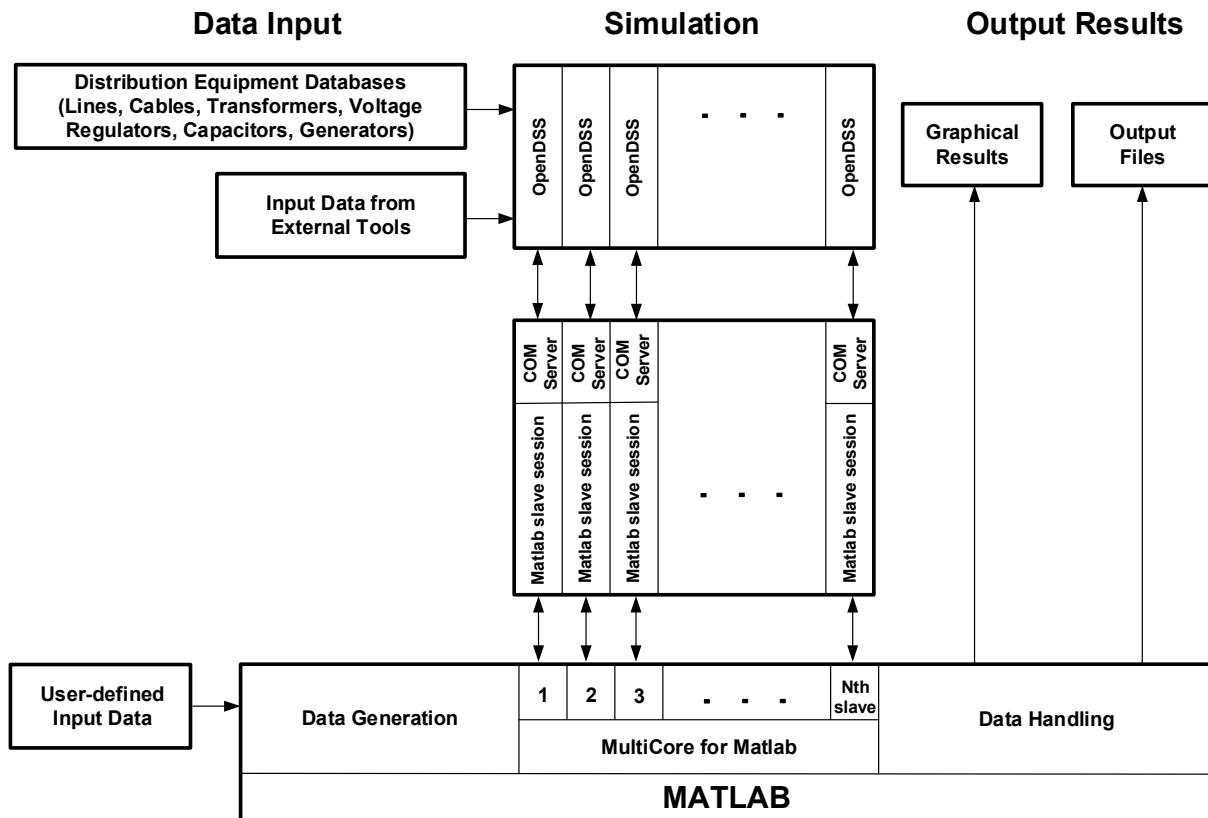
### 5.2. *Creation of custom solution methods*

The native OpenDSS solution methods can be readily expanded through the use of the tools presented in Section 3. These custom solution methods can be implemented either using the COM interface or by directly modifying the OpenDSS source code. This section presents the solution methods used and/or developed by the authors to add flexibility and new features to the built-in solution methods.

#### 5.2.1. Task-parallel executions

Planning studies may require simulating the same system many times with only small variations in certain design parameters; although individual OpenDSS execution times are relatively short, total simulation times can be rather long if the number of required simulations for a certain study is too high. In many of these studies each execution can be performed independently and is not affected by the results obtained from other executions; these characteristics fit the definition of a task-parallel job [71]. The task-parallel nature of these studies can be exploited through the application of parallel computing in order to produce a reduction in total simulation times. For this purpose, OpenDSS can be used in conjunction with the "Multicore for MATLAB" library for developing an application that introduces parallel computing to the simulation of power distribution systems.

Figure 9 presents an overview of the interaction between OpenDSS and MATLAB in the application of parallel computing to the simulation of power distribution systems; it shows how the information is exchanged among the different tools present in the implementation. According to this diagram, MATLAB is responsible for managing the complete process; it generates the input information required for the execution, passes information to OpenDSS via the COM Server, distributes the tasks among the available cores in the cluster, and collects and processes the information obtained from the simulations.



**Figure 9.** Block diagram of OpenDSS-MATLAB interaction.

Details related to input data generation and handling will depend on the objective of the study, namely, what input information is required and how output data will be analyzed. The figure also shows how one instance of OpenDSS is executed for every MATLAB slave session and system definition is loaded directly from *.dss* files; OpenDSS may require information generated by external tools (e.g. HOMER [82]) for specific applications, such as time-driven simulations. Finally, the “Multicore for MATLAB” library is embedded within MATLAB and is executed as part of the main script that controls the complete execution.

### 5.2.2. Monte Carlo method

The Monte Carlo method is a natural approach when uncertainties are involved and some variables present random behaviors; it can be applied to a large number of problems in electrical engineering, including: Insulation coordination, reliability analysis, probabilistic power flow, and optimization.

OpenDSS possesses built-in capabilities that allow the user to perform certain Monte Carlo studies; however, these capabilities are rather limited and cannot be used in more complex scenarios. In order to circumvent this drawback, OpenDSS can be used in conjunction with MATLAB to conduct Monte Carlo studies, in which MATLAB handles the generation of random numbers, execution control, and data collection, while OpenDSS is responsible for the solution of the electrical system. Information between OpenDSS and MATLAB is exchanged via the COM interface. The main benefits of relying on MATLAB for the control of the Monte Carlo method may be summarized as follows:

- Monte Carlo studies are not limited to probabilistic power flow and random fault studies.
- The generation of random numbers can be done following any of the distributions available in MATLAB, including non-parametric distributions.
- Random numbers for different variables can be generated taking into account the correlation among them.
- Any action or parameter variation that results from the random number generation is applied on the test system using the COM interface.
- Monte Carlo runs (or samples) can be distributed among the available cores in a multicore installation in order to reduce total execution times.

### 5.2.3. Genetic algorithms

A *genetic algorithm* (GA) is an optimization technique that evaluates more than one area of the search space and can discover more than one solution to a problem [83]. A complete GA procedure must include the following characteristics [84]: (i) a criterion to create the initial population; (ii) an evaluation function to obtain the fitness of all possible solutions; (iii) a genetic operator to obtain the next generation; and (iv) a stopping criterion.

GAs are prone to parallelization, depending on the approach used to evaluate the fitness function and how the genetic operators are applied. Several types of General Parallel Genetic Algorithms (GPGA) have been proposed: Global single-population master-slave GAs, single-population fine-grained, multiple-population coarse-grained GAs [85]. In global single-population GAs only one population is created and the evaluation of each member of a given generation is independent from the rest of the members; therefore, the evaluation of a generation can be seen as a task-parallel problem [86]. This approach yields a reduction of the total execution times by distributing the evaluation of individuals in a generation among available cores in a multi-core installation.

Planning and design studies may require determining the optimal value of specific parameters that seek to either minimize or maximize a feature or variable in a power distribution system. For that purpose, a GPGA can be implemented in MATLAB in order to determine the parameter's optimal value. The use of OpenDSS to simulate the distribution system's response will allow for a detailed modeling of the distribution system and an accurate estimation of network conditions. The distribution of individual executions among available cores can be done using the library developed by M. Buehren [76].

### 5.2.4. Power curves clustering

The analysis of consumption patterns (i.e. power curves) can be used for categorizing the behavior of distribution customers [87]; moreover, it can be used for characterizing the behavior of the entire distribution system. Based on field indices (e.g. ratio of average to maximum daily load) or the active power profile, consumption patterns can be grouped into clusters according to the similarity among them.

A hierarchical clustering algorithm groups a collection of individuals by forming a cluster tree [88]. The cluster tree does not separate the individuals into clusters but rather shows the connections among them at several hierarchical levels. Once the cluster tree has been built, it must be cut in

order to group individuals into clusters; depending on how the tree is separated, different clusters will be generated.

Taking advantage of the MATLAB-OpenDSS link, information resulting from an OpenDSS simulation can be readily fed into a hierarchical clustering algorithm in order to perform data reduction, hypothesis generation, hypothesis testing or prediction based on groups [89]. Figure 10 shows the general procedure for performing hierarchical clustering based on information collected from the solution of a distribution system.

### 5.2.5. Data-parallel executions

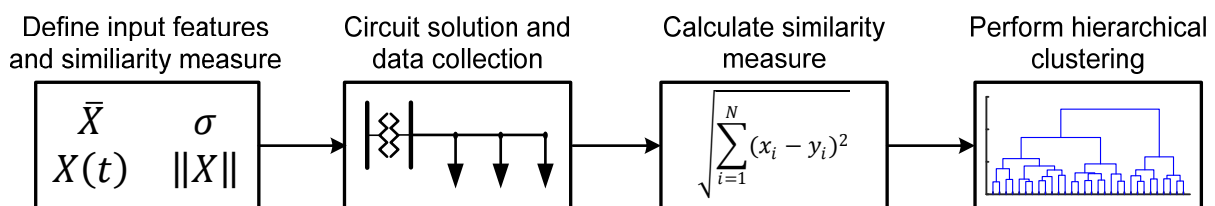
Sparse matrix solvers are optimized routines that seek to exploit the sparsity of linear systems represented by the following equation:

$$[b] = [A] [x] \quad (1)$$

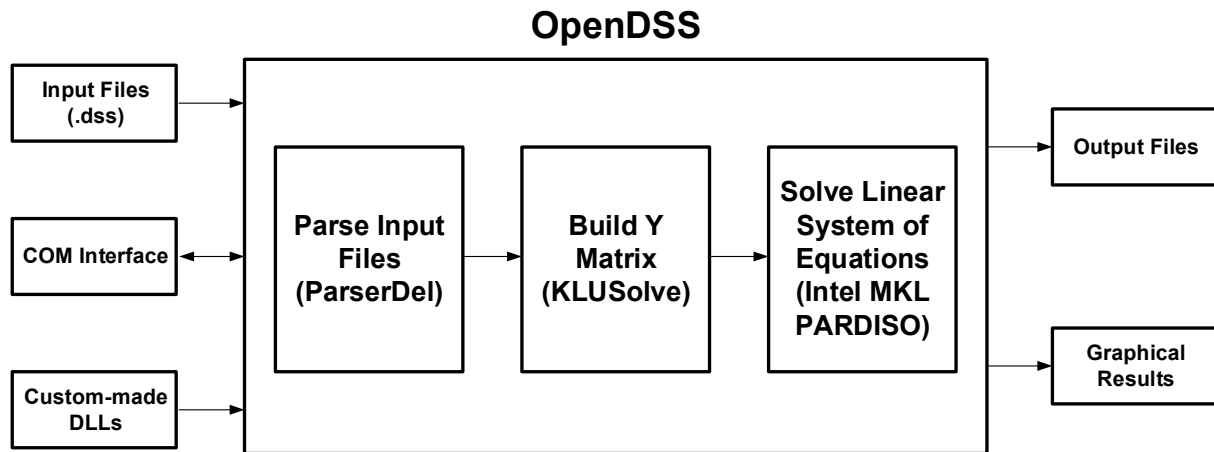
where  $A$  is a sparse matrix, and  $b$  and  $x$  are vectors. Furthermore, parallel sparse matrix solvers take advantage of the resources present in multi-core computers in order to speed up calculations. The parallel solution of linear systems of equations falls under the concept of data-parallel problems [71]. The form in Eq 1 can be recognized in the iterative solution of the Default OpenDSS [25] power flow algorithm; therefore, the application of the Intel MKL PARDISO to the solution of the aforementioned algorithm is straightforward.

One important aspect of the OpenDSS algorithm is the use of constant matrices for the iterative solution and therefore, must be calculated and triangulated only once. Thanks to Intel MKL PARDISO's capability of dividing the solution process in different stages (analysis, numerical factorization, and solve), the iterative scheme remains unchanged and the solution process can be carried out using the triangulated matrix. OpenDSS uses the single-core routine KLUSolve for the construction of the system admittance matrix and solution of the resulting sparse matrix system [90]. Contrary to Intel MKL PARDISO which requires that the sparse matrix be defined using the Compressed Sparse Row (CSR) format, the built-in KLUSolve functions only return the Compressed Sparse Column (CSC) format; this situation does not pose a problem if the correct parameters are used for matrix factorization.

Figure 11 shows the general structure of OpenDSS; within the main block, some of the most important internal tasks, and the tool in charge of conducting them, are represented. In this diagram Intel MKL PARDISO has replaced KLUSolve as the tool responsible of solving the linear system of equations (which is part of the iterative algorithm).



**Figure 10.** General procedure for performing hierarchical clustering.



**Figure 11.** OpenDSS structure with Intel MKL PARDISO.

## 6. Applications

This section presents the applications developed by the authors to perform different studies on power distribution systems. These applications are based on the tools and models presented in the previous sections. The topics included in this section cover the optimal allocation and sizing of DER, reliability of power distribution systems, impact of new technologies, and parallel solution of power flow algorithms.

### 6.1. Optimal allocation of distributed generation

The developed procedure presented in [11–16] may be defined as a Parallel Monte Carlo method aimed at estimating the location and size of one or more generation units to minimize the system losses. System energy losses are computed for an evaluation period that can span over several years. Optimal allocation studies can be categorized according to the length of the evaluation period as short-term and long-term studies. A one-year evaluation period has been considered for short-term studies, whereas evaluation periods of 10 or more years are contemplated for long-term studies. The approach for the allocation of two or more distributed generators can also depend on the length of the evaluation period. For short-term studies it can be assumed generation units are connected simultaneously; however, the connection of these units will be carried out in a sequential manner for long-term evaluations.

Input data include system parameters and time variation of loads and generations. Random variables to be generated during the application of the Monte Carlo method are locations and sizes of the generation units.

#### 6.1.1. Short-term optimal allocation of distributed generation

The goal of a short-term study is to apply the Monte Carlo procedure in order to obtain the optimal allocation of generation units for an evaluation period of one year.

In the conducted studies, the Monte Carlo procedure was applied using different number of runs according to the number of generation units to be allocated. The number of executed runs must be



large enough to ensure that optimum energy losses are near the global minimum. It is assumed the method has converged when the variations of the target variable (system energy losses) are within a margin of 1%.

The differences in energy losses among consecutive executions of the procedure are very small and in some cases negligible (always within the 1% margin). Therefore, it can be expected that global minimum will not present a significant variation with respect to the values found with the procedure. Results show that more runs than those used are required to establish a well-defined tendency of locations and ratings for more than two generation units. However, one must consider if the increment in computational time is justified when assessing the improvement in energy losses obtained with a larger number of runs.

#### 6.1.2. Refined Monte Carlo method

The goal of the Refined Monte Carlo method is to check how much accuracy is lost and how much reduction of time is achieved by neglecting some runs. Not much difference between results should be expected when the combination of the two random values gives a point that is closely located to a previously obtained and simulated point; this appears to be particularly true for points located close to the optimum. The approach is basically the same when more than one generation unit has to be connected. In such case all combinations located within a Euclidean distance equal or shorter than 5% are discarded [13].

The results in [13–16] prove that a significant reduction in computing time can be achieved; however, the accuracy of the new approach decreases as the number of generation units increases. This is an expected result, and it is evident that the accuracy can be improved by checking the differences between rated powers, not only between distances to the substation. Although some differences between location and power values between the two approaches are very significant, the energy losses obtained when applying the Monte Carlo method with the new rules are basically the same that were obtained with the original method.

#### 6.1.3. Optimal allocation of distributed generation using long-term evaluation

The goal is to estimate the optimal size and location of generation units when the target is to minimize the energy losses during a long period (i.e. equal or longer than one year) and the generating units are sequentially connected [15].

The optimal allocation will be carried out in a sequential manner and requires information regarding the number of generation units to be allocated, the year when each unit will be connected to the system and the length of the evaluation period (i.e. number of years). It is assumed that generation units are connected at the beginning of the year and that the length of the evaluation period is the same for all units.

In the conducted studies, the larger values of the rated powers correspond to the first units to be allocated; that is, the rated powers of the unit to be connected at the beginning of the first year are larger than the rated powers of any unit to be connected in subsequent years, irrespectively of the load model. However, due to the yearly variation of loads, when a unit is to be connected to a feeder in which other generation units have been previously connected, the optimum rated power of the new PV generation unit will not be always smaller than any other generator in operation because the

energy losses to be compensated for a certain evaluation term could be larger than for a previous term. The resulting energy losses are basically the same with the two Monte Carlo approaches (complete and refined). This behavior shows that a quasi-optimal solution can be reached by considering different combinations of locations and rated powers because the optimal reduction of energy losses is not very sensitive with respect to rated powers and locations of generators.

#### 6.1.4. Application of “Divide and Conquer” principle

An approach based on “Divide and Conquer” principle can be used to simplify the optimal allocation procedure for long-term evaluations in multi-feeder systems by predicting the feeder to which a generation unit must be connected based on the potential reduction of energy losses. The potential reduction of losses is estimated through calculations for a short-term study.

The procedure can be summarized as follows [15]:

- (1) Estimate the energy loss reduction factors for every feeder of the system using a one-year evaluation.
- (2) Estimate the feeders to which the first and subsequent units will be connected.
- (3) Obtain the energy loss reduction for the evaluation period every time a generation unit is to be connected.

The connection patterns with this approach are basically the same, when compared to the complete procedure, and the energy losses at the end of the period are very similar for the two Monte Carlo methods. However, the simulation time required with the new methodology is significantly shorter when the refined method is applied. The “Divide and Conquer” approach allows for the determination of a quasi-optimal solution without having to include the entire distribution system in the optimization procedure: If generators are to be allocated one by one during a given period, the study can be carried out by analyzing only the feeder in which the highest energy loss reduction can be achieved; then the refined approach can be advantageously used to significantly reduce the simulation time.

### 6.2. Reliability analysis of distribution systems

The procedure developed by the authors may be defined as a parallel Monte Carlo method aimed at estimating the probability density functions of reliability indices related to frequency of interruptions, duration of interruptions and non-supplied energy [17–19]. It can be applied to distribution systems (either overhead or underground) with or without distributed generation.

Input data include system parameters (i.e., network topology, component parameters, including setting of protection devices) and yearly variation of loads and generations. A failure is caused by a fault, a term to which several random variables are related: location, time of occurrence, duration, type. Random variables to be generated during the application of the Monte Carlo method implemented for this work are those related to failures rates, fault characterization and reconfiguration times. The use of a power flow simulator allows the estimation of the non-supplied energy.

Three scenarios, which reflect different levels of automation, have been considered for the reliability evaluation: (i) only protective devices operate; (ii) the faulted section is isolated by means of switching operations; and (iii) transfer switches are used for system reconfiguration and load transfer [17].

Results in [19] show that index values for the first scenario do not depend on the times required to isolate the failed component or to reconfigure the system, since it is assumed that service is restored (to nodes downstream of the protective device that opens) after the failed equipment is repaired. When service is restored after some switching operations and the times required for these operations are shorter than repair times, some reliability indices significantly improve, being the performance better when service may be restored both downstream and upstream of the faulted section.

The results in [18] prove that a significantly shorter simulation time is required when the calculation of reliability indices is based only on the simulation of sustained interruptions without a loss of accuracy. The differences are due to the fact that when performing reliability calculations, in addition to the reduction of the time-step size during a faulted condition, some variables (e.g., node voltages) are permanently monitored and this type of tasks slows down the procedure.

The implemented approach offers some important advantages:

- The model can be realistic and detailed, and results can be very accurate.
- Simulation results provide a significant amount of information that can be used for other purposes (e.g., optimum location of DG units).
- The information can also be used for estimating other performance indices than the typical SAIFI, SAIDI and CAIDI.

### 6.3. *Optimal sizing of energy storage*

The procedure presented in [20] uses a GPGA to determine the optimal ratings of multiple fixed-location ES units and PV generators in order to minimize distribution system energy losses, annual energy and peak power supplied by the substation transformer. ES units and PV generators are operated in conjunction, and from utility's perspective (i.e. aiming to improve network conditions).

The operation of ES units is based on the variation of load and photovoltaic generation over the evaluation period; the strategy assumes that the operation curve is the same for all ES units while the days of the evaluated period (i.e. one year) are grouped according to different classification methods in order to reduce the dimension of the optimization problem.

Three classification applications based on clustering techniques have been implemented for classifying and grouping the days of the valuation period: (i) daily served energy and PV generation; (ii) time-series clustering; and (iii) daily-values clustering. The first method uses a fixed number of groups, whereas the other two can be executed considering different numbers of classification groups. For the Time-series and Daily-values clustering, the optimal number of clusters can be calculated by means of the so-called CH index [91]. With these classifications, all days belonging to the same group will share the same operation parameters so the total number of required parameters for defining ES operation will consequently be reduced.

Results in [20] show substantial improvements regarding the peak power supplied by the substation transformer and standard deviation of the yearly power measured at the MV side of the substation. However, the procedure does not have the same effect on annual energy and energy losses. Annual energy exhibits limited improvement, mostly due to the relatively low penetration factor of DG (remember that PV generation can only help during day-time hours). Yearly energy losses are almost unaffected, and in some cases increase with respect to the case without PV generation and ES can occur; this situation is likely to be caused by the limited weight that the

energy losses reduction has on the fitness function evaluation and due to the introduction of a new source of losses (i.e. interconnection transformers).

#### 6.4. Optimal day-ahead dispatch for a virtual power plant

Day-ahead dispatch is based on the system's forecast for the following day; in order to maximize the potential benefits to the distribution system the VPP operation must be optimized according to these forecasted load and renewable resource curves. Reference [21] presented a procedure for determining the optimal hourly day-ahead dispatch of multiple VPPs from the utility's perspective.

The VPP optimal dispatch is determined by means of a GPGA, which will seek to determine the optimal values for every point of the VPP operation curve that minimize the following operation variables:

- Distribution energy losses, without including losses in substation transformer.
- Energy supplied from the MV-side of the substation transformer.
- Peak power supplied by the substation transformer from the MV-side terminals.

For a day-ahead dispatch with a 1-hour time step, the VPP operation curve will consist of 24 values, which can be either positive or negative (according to the power injection or absorption). Note that the procedure can cope with multiple dispatch curves; that is, different VPPs can follow different operation curves.

The results show a clear improvement on the energy and peak power served by the substation transformer. On the other hand, it was not possible to produce a reduction on the energy losses; however, the energy losses for the base case did not include the losses of the VPP interconnection transformer, which represent a new and significant source of losses. Results also demonstrate that the VPPs are capable of accurately following the dispatch curves that result from the GPGA.

#### 6.5. Evaluation of OpenDSS power flow calculations using parallel computing

Intel MKL PARDISO can be embedded into the solution scheme of OpenDSS to attempt the reduction of execution times when solving large distribution systems [22]. For data-parallel executions of OpenDSS, KLUSolve is replaced by Intel MKL PARDISO in the solution of the linear system of equations of the iterative algorithm. The results obtained with OpenDSS show that the Intel MKL PARDISO routine can lead to time increments of up to 47%, causing an important loss in performance when compared to KLUSolve. Results also show that increasing the number of threads used in the parallelization of the Intel MKL PARDISO routine causes an increment in solution times. This behavior is typical of small systems where the overhead caused by partitioning the problem (according to the number of threads) outweighs the time gain of the parallel solution. Although there is clear diminishment in the performance of OpenDSS, this behavior may be overturned when solving very large systems. Moreover, there are still ongoing efforts to develop more powerful and efficient parallel sparse matrix solvers.

#### 6.6. SST impact on distribution systems

The SST is a versatile device that can provide new power quality solutions to the future smart grid. The study presented in [23] explores the effect that the replacement of conventional transformers with SSTs could cause on voltages and energy losses of a distribution system. For this

study the custom-made SST model presented is used to evaluate the effect of the efficiency curve and sizing of the SST.

The obtained results show a rise in both the energy losses and the active power to be supplied from the substation when conventional transformers are replaced by SSTs. The increase of SSTs losses exceeds the reactive power compensation effect provided by the MV side of SSTs in the MV-level system. On the other hand, since voltages at the secondary terminals of SSTs are controlled to 1 pu, load voltages improve when SSTs are installed.

From the results presented in [23], it is evident that the SST efficiency is one of the main drawbacks: Considering the current achievements, very rarely the efficiency obtained with SSTs will be better than that obtained with conventional transformers. It is important to observe that oversizing the SST will lead to a lower efficiency since it will operate at low load levels during most of the time. However, it is expected that advances in semiconductor technology (e.g. wideband silicon carbide) will help to improve the SST efficiency, thus closing the gap with respect to the conventional transformer.

Voltage control provided from the SST LV terminals can be used to both increase and decrease voltage, which will also affect the energy to be supplied to loads. In addition, the reactive power compensation provided from the MV terminal can be used to reduce energy losses and control voltage at MV levels. Since SST capabilities can be used to perform voltage control in both directions, SST can simultaneously optimize the voltage and the energy required by loads. Obviously, voltage control of LV networks can also be achieved by using tapped conventional transformers; the advantage of the SSTs is its fast response: it can react to voltage variations in a few power-frequency cycles.

## 7. Conclusions

This paper has summarized the work carried out by the authors on distribution systems analysis by using OpenDSS and parallel computing, with emphasis on long-term assessment. The applications presented in this paper represent a sample of the new capabilities and resources that will be required to accurately analyze and assess the performance of the future smart grids. The paper has reviewed the modeling guidelines to be followed when performing long-term studies with a power flow simulator such as OpenDSS, the software tools used in addition to OpenDSS, the models available in OpenDSS and those created by the authors for OpenDSS implementation, the solution techniques that were required for long-term studies, and some of the studies presented by the authors during the last years. The main aspects of authors' work have been summarized in Tables 5 to 8. The studies cover a wide range of applications, see Table 8. The use of OpenDSS, whose capabilities have been expanded to include the data processing capabilities of MATLAB, has a clear advantage: It allows for the implementation of complex algorithms for data generation and post-processing without sacrificing the accuracy in the solution of the electric system. Through this approach it is possible to generate large amounts of information that can be used not only to solve the problem at hand but to support other applications.

**Table 5.** Tools summary.

Tool	Role	Main features	Studies
OpenDSS	Solution of electric systems.	Different solution methods; highly optimized solution algorithm; large number of built-in models for system components and control devices; link to external tools.	Power flow, dynamic simulations, harmonic solution.
MATLAB	Generation of input data, output data collection, and graphical outputs.	Matrix operations; large number of precom-piled functions; random number generator; graphics processor.	Applications that require custom-made solution techniques.
Multicore for MATLAB	Distribution of task executions among system cores.	It does not require a special license; it can be readily expanded for its use with a computer cluster.	Task-parallel executions, Monte Carlo method, GA.
Intel MKL PARDISO	Parallel sparse matrix solver for linear system of equations.	It does not require a deep knowledge of programming or parallel algorithms; it can be considered as a plug-and-play tool.	Parallel power flow.

**Table 6.** Custom-made models summary.

Model	Description	Main features
Storage	Adds new features to the OpenDSS built-in Storage object.	Non-constant efficiency; minimum discharge and charge power; maximum discharge and charge power based on stored energy.
VPP	It is the aggregation of renewable generation and energy storage; it can be operated as a dispatchable power source.	Non-constant efficiency; two operation modes; active and reactive power dispatch; power curtailment; idle power modeled as ZIP load.
SST	It provides the power demanded by the loads served through its secondary side while maintaining a constant voltage value. It only demands active power from the primary side.	Non-constant efficiency; bidirectional power flow; secondary side voltage can vary according to SST's overload.
Sectionalizer	It must be coordinated with an upstream protection device with reclosing capabilities.	Control can effectively operate more than one sectionalizer on the same lateral circuit.
Node load profile	It describes hourly behavior of system node demands for time-driven simulations.	Base curves represent the behavior of real customers; they account for randomness and temporal patterns of electricity consumption.
Photovoltaic generation curves	It describes hourly behavior of solar generation for time-driven simulations.	They are based on geographical coordinates and environmental conditions of the location site; they also account for the stochastic behavior of the solar resource.
Wind generation curves	It describes hourly behavior of wind generation for time-driven simulations.	They are based on environmental conditions of the location site; they introduce the stochastic behavior and temporal patterns of the wind resource.

**Table 7.** Solution techniques summary.

Techniques	Applications	Drawback	Remarks
Task-parallel executions	Reduction of execution times for applications where the same task must be performed many times with variations of input parameters.	Overhead required for distributing executions can become significant for very short individual executions.	Grouping multiple executions and performing them as a batch can help to avoid this drawback.
Monte Carlo method	Characterization of systems whose inputs present stochastic behavior.	Simulation times can become prohibitively large due to system size and number of samples.	Simulation times can be reduced through the application of parallel computing.
Genetic algorithms	Problem optimization through the evaluation of more than one area of the search space. It can cope with complex fitness functions.	Genetic algorithms are considered near-optimum algorithms and rarely find the real global optimum.	For many studies a near-optimum solution is considered as a good enough approximation.
Power curves clustering	Discovery of natural groupings among similar individuals in a population.	Clustering results depend greatly on the chosen input criteria and similarity measure.	Clustering can be used as a dimensionality reduction technique for optimization problems.
Data-parallel executions	Reduction of execution times for single tasks that can be partitioned into smaller sub-problems which can be solved simultaneously.	Small size systems of linear of equations can lead to performance reduction with respect to single-core algorithms.	Graphics processing units (GPUs) represent an option for the implementation of data-parallel algorithms.

Although the results obtained from these studies can be considered as satisfactory, there still exist key-points that should be taken into account in order to improve their accuracy and ensure a precise representation of a real-world distribution system. For instance, simulation time-step is one aspect that can play an important role on the expected solution for time-driven simulations; in some cases time-steps of 1 minute or less may be necessary to capture the effect of output variations in renewable generation due to the random behavior of the primary energy source (e.g. sun or wind). On the other hand, the long-term optimization of energy storage units and virtual power plants operation will lead to high-dimension optimization problems; so far, this drawback has been circumvented through the application of dimensionality reduction techniques; however, finding the real global optimum will require the application of optimization algorithms that do not rely on such techniques.

The use of a multicore environment and the development/implementation of some software OpenDSS-based applications have allowed a significant reduction of the simulation time required for long-term studies. However, it is evident that total execution times will rapidly grow as the systems under study become larger and the applied solution techniques involve a larger number of executions to produce accurate results. Therefore, improvements in parallel-computing based techniques, in addition to other approaches that can lead to a reduction of simulation times, are still needed, especially when working with large realistic system models and more complex solution techniques.

**Table 8.** Applications summary.

Study	Tools	Techniques	Advantages	Drawbacks
DG optimal allocation	OpenDSS, MATLAB, Multicore for MATLAB.	Monte Carlo method, Task-parallel executions.	Easy to implement; test systems can be of any size and include detailed models.	Simulation times grow rapidly based on the system size and number of generation units to be allocated.
Distribution reliability	OpenDSS, MATLAB, Multicore for MATLAB.	Monte Carlo method, Task-parallel executions.	Very accurate; it can produce large amounts of information and include time-dependent actions.	Individual execution times can be significant even for small and mid-size systems.
ES optimal sizing	OpenDSS, MATLAB, Multicore for MATLAB.	Genetic algorithms, Task-parallel executions, Power curves clustering.	It can cope with multiple ES units; test systems can be of any size and include detailed models.	Reaching the true global optimum requires optimizing every single ES operation point.
VPP day-ahead dispatch	OpenDSS, MATLAB, Multicore for MATLAB.	Genetic algorithms, Task-parallel executions.	It optimizes every VPP operation point; it can cope with multiple operation curves.	A smaller time-step might be required for a more accurate solution.
Parallel power flow	OpenDSS, MATLAB, Intel MKL PARDISO.	Data-parallel executions.	It can be easily introduced into the solution scheme of different power flow algorithms.	It produces no improvements for system sizes similar to those tested.
SST impact	OpenDSS, MATLAB.	-	It provides a flexible platform for model and control algorithm prototyping.	The external SST control introduces a loss of computational performance.

EPRI has recently released two new OpenDSS-based tools that can be extremely useful for users: OpenDSS-PM and OpenDSS-G. OpenDSS-PM allows the user to load multiple circuits into the shared memory of a single multi-core processor and solve them in an individual or parallel manner [91]. On the other hand, OpenDSS-G provides a user-friendly graphical interface where the user can develop new circuit models and make use of all the features present in the OpenDSS solution engine [92]. Both tools have been developed as expansions of OpenDSS's original capabilities.

In summary, the long-term assessment based on a tool such OpenDSS offers some important advantages: (i) the test systems can be realistic and include detailed models for system components, and results can be very accurate; (ii) evaluation periods can span over several years and take into account individual growth curves for system loads' rated power; (iii) simulation results provide a significant amount of information that can be used for other purposes; (iv) MATLAB's capabilities can be used to develop complex algorithms for data generation and post-processing; (v) parallel computing can be applied in order to reduce total simulation times; (vi) custom-made models can be developed and readily included into the test systems.



## Conflict of interest

All authors declare there is no conflict in this paper.

## References

1. Dugan RC, Arritt RF, Mcdermott TE, et al. (2010) Distribution system analysis to support the smart grid. *Power Energy Soc Gen Meet* 2010: 1–8.
2. Martinez JA, Martin-Arnedo J (2011) Tools for analysis and design of distributed resources-part I: Tools for feasibility studies. *IEEE T Power Delivery* 26: 1643–1652.
3. Martinez JA, León FD, Dinavahi V (2010) Simulation tools for analysis of distribution systems with distributed resources. Present and future trends. *Power Energy Soc Gen Meet* 39: 1–7.
4. Ackermann T, Andersson G, Söder L (2001) Distributed generation: A definition. *Electr Power Syst Res* 57: 195–204.
5. Farret FA, Godoy Simões M (2006) *Integration of Alternative Sources of Energy*. John Wiley.
6. Sandia National Laboratories and NRECA (2015) *DOE/EPRI Electricity Storage Handbook*.
7. New York Independent System Operator (2014) *A Review of Distributed Energy Resources*.
8. Carrasco JM, Franquelo LG, Bialasiewicz JT, et al. (2006) Power-electronic systems for the grid integration of renewable energy sources: A survey. *IEEE T Ind Electr* 53: 1002–1016.
9. Arritt RF, Dugan RC (2011) Distribution system analysis and the future smart grid. *IEEE T Ind Appl* 47: 2343–2350.
10. Taylor J, Smith JW, Dugan RC (2012) Distribution modeling requirements for integration of PV, PEV, and storage in a smart grid environment. *Power Energy Soc Gen Meet* 2011: 1–6.
11. Martinez JA, Guerra G (2012) Optimum placement of distributed generation in three-phase distribution systems with time varying load using a Monte Carlo approach. *IEEE Power Energy Soc Gen Meet* 2012: 1–7.
12. Guerra G, Martinez JA (2014) A Monte Carlo method for optimum placement of photovoltaic generation using a multicore computing environment. *Pes Gen Meet Conf Expo* 2014: 1–5.
13. Martinez JA, Guerra G (2014) A parallel Monte Carlo method for optimum allocation of distributed generation. *IEEE T Power Syst* 29: 2926–2933.
14. Martínez-Velasco JA, Guerra G (2015) Analysis of large distribution networks with distributed energy resources. *Ingeniare* 23: 594–608.
15. Guerra G, Martínez-Velasco JA (2016) Optimum allocation of distributed generation in multi-feeder systems using long term evaluation and assuming voltage-dependent loads. *Sust Energy Grids Netw* 5: 13–26.
16. Martínez-Velasco JA, Guerra GA (2016) Allocation of distributed generation for maximum reduction of energy losses in distribution systems, In: *Energy Management of Distributed Generation Systems* (L. Mihet-Popa, ed.), ch. 05. InTech.
17. Martinez JA, Guerra G (2013) A Monte Carlo approach for distribution reliability assessment considering time varying loads and system reconfiguration. *Power Energy Soc Gen Meet* 23: 1–5.
18. Martínez-Velasco JA, Guerra G (2014) Parallel Monte Carlo approach for distribution reliability assessment. *IET Gen Trans Dis* 8: 1810–1819.

19. Martinez-Velasco JA, Guerra G (2016) Reliability analysis of distribution systems with photovoltaic generation using a power flow simulator and a parallel Monte Carlo approach. *Energies* 9: 537.
20. Guerra G, Martinez-Velasco JA (2018) Optimal sizing and operation of energy storage systems considering long term assessment. *AIMS Energy* 6: 70–96.
21. Guerra G, Martinez-Velasco JA (2017) A virtual power plant model for time-driven power flow calculations. *AIMS Energy* 5: 887–911.
22. Guerra G, Martinez-Velasco JA (2017) Evaluation of MATPOWER and OpenDSS load flow calculations in power systems using parallel computing. *J Eng* 2017: 195–204.
23. Guerra G, Martinez-Velasco JA (2017) A solid state transformer model for power flow calculations. *Int J Electr Power Energy Syst* 89: 40–51.
24. Dugan RC (2016) *Reference Guide. The Open Distribution System Simulator (OpenDSS)*. EPRI.
25. Dugan RC, Mcdermott TE (2011) An open source platform for collaborating on smart grid research. *Power Energy Soc Gen Meet* 5: 1–7.
26. Kersting WH (2007) *Distribution System Modeling and Analysis*. CRC Press.
27. Chen T, Chen M, Inoue T, et al. (1991) Three-phase cogenerator and transformer models for distribution system analysis. *IEEE T Power Delivery* 6: 1671–1681.
28. Daran ME, Staton EA (1997) Distribution transformer models for branch current based feeder analysis. *IEEE T Power Syst* 12: 698–703.
29. Kersting WH, Phillips WH, Carr W (1999) A new approach to modeling three-phase transformer connections. *IEEE Trans Ind Appl* 35: 169–175.
30. Xiao P, Yu DC, Yan W (2006) A unified three-phase transformer model for distribution load flow calculations. *IEEE T Power Syst* 21: 153–159.
31. Peralta JA, de León F, Mahseredjian J (2008) Unbalanced multiphase load-flow using a positive-sequence load-flow program. *IEEE T Power Syst* 23: 469–476.
32. Kersting WH (2009) The modeling and application of step voltage regulators. *Power Syst Conf Expo* 2009: 1–8.
33. Cheng CS, Shirmohainmadi D (1995) A three-phase power flow method for real-time distribution system analysis. *IEEE T Power Syst* 10: 671–679.
34. Garcia PAN, Pereira JLR, Carneiro S (2001) Voltage control devices models for distribution power flow analysis. *IEEE T Power Syst* 16: 586–594.
35. Hatziargyriou ND, Karakatsanis TS, Papadopoulos M (1993) Probabilistic load flow in distribution systems containing dispersed wind power generation. *IEEE T Power Syst* 8: 159–165.
36. Feijóo AE, Cidrás J (2000) Modeling of wind farms in the load flow analysis. *IEEE T Power Syst* 15: 110–115.
37. Divya KC, Rao PSN (2006) Models for wind turbine generating systems and their application in load flow studies. *Electr Power Syst Res* 76: 844–856.
38. Eminoglu U (2009) Modeling and application of wind turbine generating system (WTGS) to distribution systems. *Renew Energy* 34: 2474–2483.
39. Caramia P, Carpinelli G, Pagano M, et al. (2007) Probabilistic three-phase load flow for unbalanced electrical distribution systems with wind farms. *IET Renew Power Gener* 1: 115–122.
40. Teng J (2008) Modelling distributed generations in three-phase distribution load flow. *IET Gener Trans Distrib* 2: 330–340.

41. RETScreen International, Clean Energy Decision Support Centre, 2005. *Clean Energy Project Analysis, RETScreen Engineering and Case Textbook*. Minister of Natural Resources Canada.
42. Manwell JF, Rogers A, Hayman G, et al. (1998) Hybrid2—A Hybrid System Simulation Model, Theory Manual. University of Massachusetts.
43. Vlachogiannis JG (2009) Probabilistic constrained load flow considering integration of wind power generation and electric vehicles. *IEEE T Power Syst* 24: 1808–1817.
44. Feijóo AE, Cidrás J, Dornelas JLG (1999) Wind speed simulation in wind farms for steady-state security assessment of electrical power systems. *IEEE T Energy Convers* 14: 1582–1588.
45. Papaefthymiou G, Schavemaker PH, Sluis LVD, et al. (2006) Integration of stochastic generation in power systems. *Int J Electr Power Energy Syst* 28: 655–667.
46. Papaefthymiou G, Kurowicka D (2009) Using copulas for modeling stochastic dependence in power system uncertainty analysis. *IEEE T Power Syst* 24: 40–49.
47. Duffie JA, Beckman WA (1991) *Solar Engineering of Thermal Processes*. John Wiley.
48. Graham VA, Hollands KGT (1990) A method to generate synthetic hourly solar radiation globally. *Sol Energy* 44: 333–341.
49. Conti S, Raiti S (2007) Probabilistic load flow using Monte Carlo techniques for distribution networks with photovoltaic generators. *Sol Energy* 81: 1473–1481.
50. Price WW, Taylor CW, Rogers GJ, et al. (1995) Standard load models for power flow and dynamic performance simulation. *IEEE T Power Syst* 10: 1302–1313.
51. Caramia P, Carpinelli G, Varilone P, et al. (1999) Probabilistic three-phase load flow. *Int J Electr Power Energy Syst* 21: 55–69.
52. Heunis SW, Herman R (2002) A probabilistic model for residential consumer loads. *IEEE T Power Syst* 17: 621–625.
53. Lu N, Chassin DP (2004) A state-queueing model of thermostatically controlled appliances. *IEEE T Power Syst* 19: 1666–1673.
54. Lu N, Chassin DP, Widergren SE (2005) Modeling uncertainties in aggregated thermostatically controlled loads using a state queueing model. *IEEE T Power Syst* 20: 725–733.
55. Schneider KP, Fuller JC (2010) Detailed end use load modeling for distribution system analysis. *Power Energy Soc Gen Meet* 2010: 1–7.
56. Garcia-Valle R, Vlachogiannis JG (2009) Electric vehicle demand model for load flow studies. *Electr Power Comp Syst* 37: 577–582.
57. Kondoh J, Ishii I, Yamaguchi H, et al. (2000) Electrical energy storage systems for energy networks. *Energy Convers Manage* 41: 1863–1874.
58. Ibrahim H, Ilinca A, Perron J (2008) Energy storage systems-characteristics and comparisons. *Renew Sust Energy Rev* 12: 1221–1250.
59. Divya KC, Østergaard J (2009) Battery energy storage technology for power systems-an overview. *Electr Power Syst Res* 79: 511–520.
60. Barton JP, Infield DG (2004) Energy storage and its use with intermittent renewable sources. *IEEE T Energy Convers* 19: 441–448.
61. Korpaas M, Holen AT, Hildrum R (2003) Operation and sizing of energy storage for wind power plants in a market system. *Int J Electr Power Energy Syst* 25: 599–606.
62. Klöckl B, Stricker P, Koeppel G (2005) On the properties of stochastic power sources in combination with local energy storage. In CIGRE Symposium on Power Systems with Dispersed Generation, Athens, 1–4.

63. Koepfel G, Korpaas M (2006) Using storage devices for compensating uncertainties caused by non-dispatchable generators. *Int Conf Probab Methods* 2006: 1–8.
64. IEEE Recommended Practice for Protection and Coordination of Industrial and Commercial Power Systems, IEEE Standard 242-2001.
65. Thurner L, Scheidler A, Schäfer F, et al. (2017) *Pandapower documentation*. Fraunhofer IWES.
66. Dugan RC, OpenDSS Tech notes, 2018. Available from: <https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/Doc/TechNotes>.
67. Benmouyal G, Meisinger M, Burnworth J, et al. (1999) IEEE Standard Inverse-Time Characteristic Equations for Overcurrent Relays. *IEEE T Power Deliver* 14: 868–872.
68. Gönen T (2008) *Electric Power Distribution System Engineering*, 2nd Edition, CRC Press.
69. The MathWorks Inc. MATLAB 9.2 (R2017a). The MathWorks Inc., 2017.
70. Grama A, Gupta A, Karypis G, et al. (2003) *Introduction to Parallel Computing*, 2nd Edition, Pearson Education.
71. Boston University. *MATLAB Parallel Computing Toolbox Tutorial*. Available from: <http://www.bu.edu/tech/support/research/training-consulting/online-tutorials/matlab-pct>.
72. Kepner J (2009) *Parallel MATLAB for Multicore and Multinode Computers*. SIAM.
73. Moler C (2007) Parallel MATLAB: Multiple processors and multiple cores. *The MathWorks News & Notes*.
74. Luszczek P (2008) Enhancing multicore system performance using parallel computing with MATLAB. *MATLAB Digest* 17.
75. Buehren M (2007) *MATLAB Library for Parallel Processing on Multiple Cores*. Available from: <http://www.mathworks.com>.
76. Intel Math Kernel Library for Windows OS (Developer Guide). Available from: <https://software.intel.com/en-us/intel-mkl-support/documentation>.
77. Schenk O, Gärtner K, Fichtner W (2000) Efficient sparse LU factorization with left-right looking strategy on shared memory multiprocessors. *BIT Num Math* 40: 158–176.
78. Intel Math Kernel Library (Developer Reference). Available from: <https://software.intel.com/en-us/intel-mkl-support/documentation>.
79. Magnano L, Boland JW (2007) Generation of synthetic sequences of electricity demand: Application in South Australia. *Energy* 32: 2230–2243.
80. Carapellucci R, Giordano L (2013) A methodology for the synthetic generation of hourly wind speed time series based on some known aggregate input data. *Appl Energy* 101: 541–550.
81. Homer Energy. *HOMER Pro*. Available from: <https://www.homerenergy.com>.
82. Passino KM (2005) *Biomimicry for Optimization, Control, and Automation*. Springer Science & Business Media.
83. Michalewicz Z (1992) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.
84. Yeh EC, Venkata SS, Sumic Z (1995) Improved distribution system planning using computational evolution. *Power Syst IEEE Trans* 11: 668–674.
85. Konfrst Z (2004) Parallel genetic algorithms: Advances, computing trends, applications and perspectives. *Int Parallel Distrib Process* 2004: 162.
86. Chicco G, Napoli R, Postolache P, et al. (2003) Customer characterization options for improving the tariff offer. *IEEE T Power Syst* 18: 381–387.
87. Eisen MB, Spellman PT, Brown PO, et al. (1998) Cluster analysis and display of genome-wide expression patterns. *P Natl Acad Sci USA* 95: 14863–14868.

88. Theodoridis S, Koutroumbas K (2008) Pattern Recognition. *IEEE T Neural Network* 19: 376.
89. KLUSolve. Available from: <https://sourceforge.net/projects/klusolve>.
90. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. *Commun Stat* 3: 1–27.
91. Montenegro D, Dugan RC, Reno MJ (2017) Open source tools for high performance quasi-static time-series simulation using parallel processing. In IEEE Photovoltaic Specialists Conference (PVSC).
92. OpenDSS-G. Available from: <https://sourceforge.net/projects/dssimpc>.



**AIMS Press**

© 2018 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)