No 727481 RESERVE

D4.3 v1.0

Functionality of the Release of the Real-Time Solver

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 727481.

Project Name	RESERVE
Contractual Delivery Date:	30.09.2018
Actual Delivery Date:	29.09.2018
Contributors:	RWTH, POLITO
Workpackage:	WP4
Security:	PU
Nature:	R
Version:	1.0
Total number of pages:	30

Abstract:

This deliverable provides an overview of DPsim and its correspondent library. DPsim is a dynamic phasor solver for real-time simulation. The solver computes the solution of a network consisting of models defined in the dynamic phasor domain for every simulation step. This first version of the solver has been implemented in C++ and will be delivered as an open source solution. This deliverable describes the reference implementation of the solver along with the main algorithms adopted for the simulation and the models implemented in the framework.

Keyword list:

Real-time simulation, dynamic phasors, modified nodal analysis, resistive companion, distributed co-simulation

Executive Summary

The research within RESERVE touches fundamental concepts of today's power systems. Decreasing bulk electricity generation while increasing the share of distributed energy resources, has a major impact on the entire system. The current practice of controlling the grid based on a hierarchical coordination is being steadily replaced by a decentralized approach, which strongly relies on new communication technologies.

Testing new concepts in a large scale power system scenario is a very difficult and a challenging task for two reasons: cost and control of all system parameters. A test grid would have to be set up separately from the grid that provides energy to end-users to ensure quality of service. Moreover, it is not possible to control all test parameters, for example, the weather which has a large impact considering the share of increase in renewable energy sources.

RESERVE proposes a distributed real-time simulation environment which relies on the internet and utilizes the simulation resources available from project partners to test the concepts developed in the project. Real-time simulation allows for the integration of hardware which enables tests that include both hardware and software components.

Given that commercial real-time simulators are usually expensive, RESERVE has progressed in developing of a real-time solver that can be used to connect distributed simulations. Our real-time solver does not work in the electromagnetic transient domain like commercial real-time solvers. Instead it is based on the dynamic phasor concept. The advantage of this approach is that it offers more flexibility in the selection of the simulation time step, which is an important advantage for distributed simulation, as presented in Section 6.1. This report presents the functionalities of this solver called DPsim.

Before explaining the details of the solver, this report provides a technical analysis of the difficulties tied to internet distributed simulation. Then, we propose dynamic phasors as a solution to improve the accuracy of the simulation results for large time steps. The time step is of particular interest since the communication delay in the range of tens of milliseconds between simulators interconnected through wide area networks is much larger than the simulation time step typically used in electromagnetic transient real-time simulations.

The results achieved from simulations in the dynamic phasor and electromagnetic transient domain are compared to quantify the advantage of dynamic phasor simulations in practice. The test platform for this evaluation is the power system simulator, which is currently under development. It is shown that the dynamic phasor results are significantly better than the electromagnetic transient results for time steps of tens of millisecond which is in the range of the expected delays for distributed simulation in Europe.

Compared to the first version of this deliverable, D4.2, the results section was extended by synchronous generator examples and a new section was added that describes the integration of DPsim with the laboratory interface which is presented in deliverable D4.1.

Authors

Partner	Name	e-mail
RWTH		
	Markus Mirz	Phone: +49 241 80 49739 e-mail: mmirz@eonerc.rwth-aachen.de
	Steffen Vogel	Phone: +49 241 80 49577 e-mail: svogel2@eonerc.rwth-aachen.de
	Antonello Monti	Phone: +49 241 80 49700 e-mail: amonti@eonerc.rwth-aachen.de
POLITO		
	Abouzar Estebsari	Phone: +39 011 090 7117 e-mail : abouzar.estebsari@polito.it
	Francesco Arrigo	Phone: +39 011 090 7144 e-mail: francesco.arrigo@polito.it
	Ettore Bompard	Phone: +39 011 090 7154 e-mail: ettore.bompard@polito.it

Table of Contents

1. Introduction ... 5

1.1 Task 4.2 ... 5

1.2 Objectives of the Work Report in this Deliverable ... 5

1.3 Outline of the Deliverable ... 5

1.4 How to Read this Document ... 5

2. Real-Time Simulation in RESERVE ... 7

3. Challenges of Distributed Real-Time Simulation ... 9

3.1 Summary ... 10

4. Basic Structure of the Real-Time Solver ... 11

4.1 Nodal Analysis ... 11

4.2 Resistive Companion Method ... 12

4.3 Dynamic Phasors ... 13

4.4 Summary ... 15

5. Integration of Nonlinear Models ... 16

6. Simulation Results Using the Real-Time Solver ... 17

6.1 Basic Circuit Simulation ... 17

6.2 Simulation of Synchronous Generator ... 19

6.2.1 Load Change ... 19

6.2.2 Three-Phase Fault ... 20

6.2.3 Comparison between Dynamic Phasor and EMT Model ... 21

7. Integration with the Laboratory Interface ... 23

7.1 VILLASframework Laboratory Interface ... 23

7.2 Logging of results to CSV files ... 24

7.3 Python Interface ... 24

8. Conclusion ... 25

9. List of figures ... 26

10. List of Tables ... 27

11. References ... 28

12. List of Abbreviations ... 29

Annex ... 30

A.1 Synchronous Generator Simulation Parameters ... 30

1. Introduction

Renewables in a Stable Electric Grid (RESERVE) is a three-year European Commission funded project within the Work Program H2020-LCE-2016-2017. The project officially started in October 2016.

1.1 Task 4.2

This deliverable is the second major output of Task 4.2 in WP4. This task is about the development of the open source real-time simulator DPsim, which is going to be developed in the frame of RESERVE. Deliverable D4.2 describes the fundamental theory behind DPsim, whereas D4.3, the second version of that deliverable, describes the implementation of more complex models in the solver and the connection of the VILLAS framework, which is described in D4.1. Mainly, the new contributions to this deliverable can be found in Sections 5 to 7.

1.2 Objectives of the Work Report in this Deliverable

While the primary objective of this report is to lay out the concept behind DPsim, the laboratory interface design is touched on as well because of the strong interconnection between the two components.

1.3 Outline of the Deliverable

The first part of this report explains the challenges of distributed real-time simulation and presents previous work in this field. Then, the theory behind the developed real-time simulator is outlined. The main pillars of DPsim: modified nodal analysis, dynamic phasors and the resistive companion method are described. Compared to D4.2 which is the previous deliverable describing solver functionalities, one section on the implementation of more complex power system models and one section on the integration of the laboratory interface have been added. The final section concludes with results of two different simulation scenarios: a basic example, which shows the effect of using dynamic phasors compared to the more traditional EMT approach, and a comparison of different synchronous generator models.

1.4 How to Read this Document

This document can be read on its own, but should the reader want to learn about the components that interconnect real-time simulators such as the simulator described in this deliverable, we suggest reading deliverable D4.1. Overall, this deliverable (D4.3) is related to the following document from the RESERVE project:

- D4.1 – Demonstration of prototype of laboratory infrastructure
- D4.2 – Functionality of the releases of the real-time solver, V1
- D4.4 – First interconnection test of the nodes in pan-European simulation platform

D4.1 offers insight into the laboratory interconnection infrastructure, which is the framework that is connecting instances of DPsim and other real-time simulators. This deliverable (D4.3) is the second iteration of deliverable D4.2 and explains the simulator that is used for first interconnection tests, which are described in deliverable D4.4. The chart below provides a graphical representation of the dependencies between the deliverables in WP4 of RESERVE.

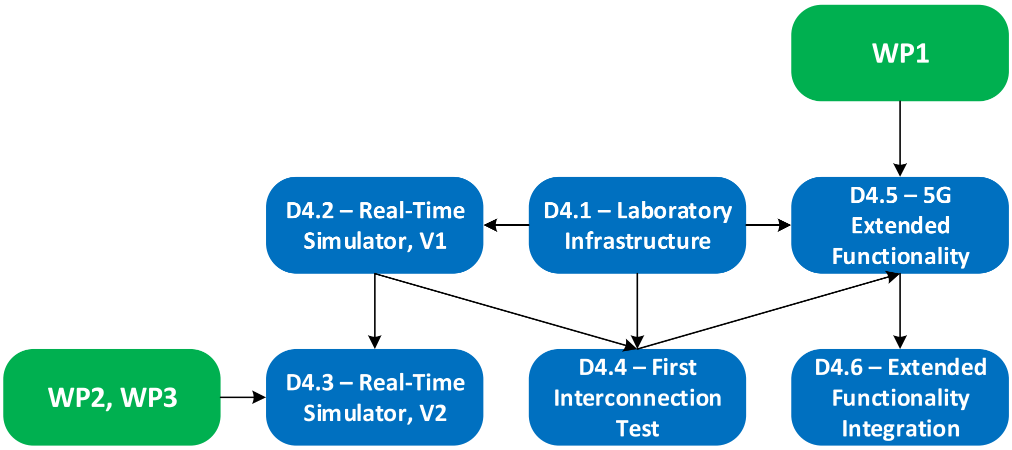


Figure 1.1: Relations between deliverables in WP4 and other work

2. Real-Time Simulation in RESERVE

The transition to power systems, which will rely almost entirely on generation from renewable energy sources (RES), is inevitable and a global priority to tackle climate change. Fossil fuelled bulk generation is gradually replaced by decentralized sources since RES are typically smaller and more geographically distributed. Therefore, new concepts and techniques are required for ancillary services supporting voltage and frequency control. This transformation towards a decentralized system includes ICT to enable near real-time distributed decision-making, placing requirements on ICT which today's systems do not meet. 5G ICT systems offer the prospect of seamless and secure connectivity, data processing with the reliability, availability and resilience required by energy providers and their customers in systems to be on the global market from 2020 onwards. Compared to land lines, 5G technology is more cost efficient in scenarios where a large number of users has to be provided.

Real-time simulation is becoming increasingly popular to test and validate equipment and algorithms in a controlled and realistic environment. The synchronization of simulation time with wall clock time allows the exchange of physical inputs and outputs between externally connected devices and the real-time simulator since a second in real-time takes one second to be simulated. When a device is interfaced with a simulator, in a so-called Hardware-In-the-Loop (HIL), the behavior of that device in a power system can be effectively tested and validated, for both normal and emergency operating conditions, without the need of on-site testing. The applications of real-time simulation in power system analysis are, for example, development and testing of protection and control systems, distributed generation modelling, especially with renewable energy resource integration and micro grid control, etc. [1]. In case real-time simulators and the devices-under-test are geographically distributed or the capabilities of locally available real-time simulators are not sufficient for the given simulation scenario, the model used for simulations can be partitioned for a distributed simulation [2].

One of the main objectives of RESERVE is the development of an innovative pan-European real-time, internet-distributed simulation infrastructure, which is able to support large scale experimental activity in the field of power dynamics and automation. This infrastructure will be a living roadmap supporting the decision-making process in the energy transition including new control algorithms and the increase of communication technology in power systems enabled by the new 5G standard.

The reasons for distributing the simulation are manifold. It allows for the sharing of available hardware and software in different real-time simulation laboratories among participants to enhance computational power and facilitate remote Software-In-the-Loop (SIL) and (Power-)Hardware-In-the-Loop (PHIL/HIL). Then, new devices that are going to be integrated in power systems can be tested even in remote locations where real-time simulation capabilities are not available. Large scale system simulations are facilitated because the real-time capabilities of several simulation facilities could be harnessed. Besides, confidential data does not need to be shared as each laboratory can be responsible for simulating its own part of the model locally, solely exchanging interface variables with other interconnected systems, imitating the real world where regional or national power grids are interconnected through tie-lines. The implementation of this distributed infrastructure can be broken down roughly into three subtasks: Interfacing the real-time simulation laboratories over a communications network, developing a simulation solver and integrating the infrastructure in a cloud-based interface supporting location agnostic access to the simulation.

The exchange of real-time power system simulation data poses a challenge in internet-distributed simulation since the communication delay between simulators may be orders of magnitude larger than the simulation step of commercial electromagnetic transient (EMT) real-time simulators. In this case, it is not possible to exchange data between the simulators in every time step and a compensation of the communication delay is required. Alternatively, the simulation could be performed in the classic phasor domain, which allows larger simulation time steps. However, this would defy the objective of developing new frequency control algorithms since classic phasor domain simulations assume a fixed frequency. In RESERVE, the approach is to engage the problem in two ways. A novel co-simulation interface called VILLASframework enables the real-time data exchange between laboratories. Furthermore, the real-time solver, which is subject to this report, allows an increase of the simulation step size while supporting variable frequency scenarios.

Figure 2.1 shows one of the possible scenarios that could be tested with the available components in WP 4. If the co-simulation model is a part of the Romanian transmission grid, UPB could provide measurement data from the Romanian grid. WIT uses the real-time solver developed in RESERVE to join the simulation while RWTH and POLITO employ a combination of their commercial EMT simulators and the new solver.

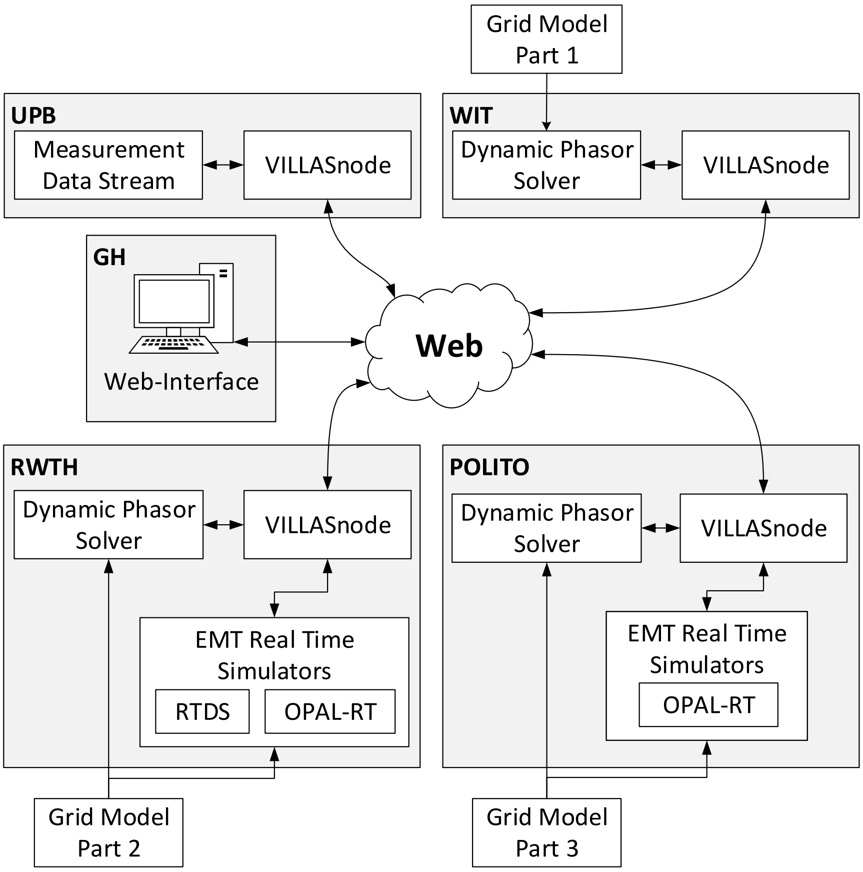


Figure 2.1: Example co-simulation scenario for WP 4

The first version of DPsim described in this report allows the simulation of networks that consist of ideal voltage and current sources, sources with internal impedance and basic passive components like resistors, capacitors and inductors. The code is maintained in a GitLab (https://about.gitlab.com/) instance hosted by RWTH. This platform will also be used to provide the solver to the project partners.

3. Challenges of Distributed Real-Time Simulation

Geographically distributed simulation borrows problems from parallel simulation. In both cases the model must be partitioned and it has to be assured that the parallel simulation results do not differ from the results of a sequential simulation. Furthermore, real-time requirements decrease the number of suitable simulation techniques [3]. Typically, power system real-time simulators follow the fixed time step approach since more sophisticated variable time step integration methods are not suitable for fast calculations as needed for real-time execution and small-time steps [5].

A common approach to partition power systems for parallel simulation is the use of travelling wave transmission line models [6]. However, it should be noted that electromagnetic waves travel about 15 km in 50 µs, which is a typical step time in real-time power system simulation. The expected delay in internet-distributed simulation is tens of milliseconds. In case of geographically large distances between the simulators, the time needed for information exchange can reach tens of milliseconds. Therefore, the insertion of a line with the required length into the model would have a severe impact on the behaviour of the system.

Without compensation, the communication delay might cause large errors and even instability as shown in [1] for a delay of more than 10 ms. One cause for this is the sampling requirement imposed if an AC 50 Hz or 60 Hz system is simulated in EMT. According to the sampling theorem, the minimum sampling frequency is twice the maximum frequency expected in the system. This combined with the large RTT expected in geographically distributed simulations, complicates the synchronization among simulators.

Simulations using traditional static phasors do not impose the strong sampling requirement since the system frequency is implicitly included but this frequency is fixed. Therefore, this approach does not support frequency control or stability studies, for example, on transmission level. The general theory of dynamic phasors is covered in Section 4.3.

The authors of [7] have developed an integrated real-time co-simulation laboratory by applying a communication platform as a simulator-to-simulator interface proposed in [1] in order to enable remote and online monitoring of an interconnected transmission-distribution system. Based on that novel approach, each simulator carried out simulations in time domain, while the time-varying Fourier coefficients of the quantities in the interconnection node, i.e. decoupling point, are exchanged. In the model, the interface was represented as ideal transformer model (ITM) [8].

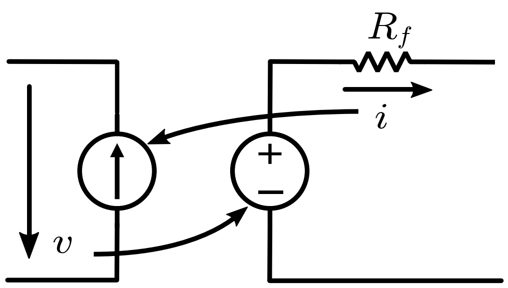


Figure 3.1: ITM represented by current and voltage source

EMT values could not be exchanged for every simulation step due to the communication RRT. Apart from that, the local EMT simulation is computationally less efficient compared to phasor simulations. Instead, the following transformed quantities could be exchanged: static phasors or dynamic phasors of the fundamental and harmonic components. The delay is compensated by shifting the phasors in time at the receiving end.

The problem with the former solution is the following. For transient analysis, frequency deviation in one side cannot be captured on the other side to perform a distributed real-time simulation with the same results as a local real-time simulation. With the latter solution, dynamic phasor exchange, simulations do not imply fixed system frequency.

The approach explained in [7] requires the extraction of phasor information from the EMT signals. Therefore, transparency of the interface is not given since the interface algorithm may alter the exchanged signals. So far, the interface algorithm can extract magnitude and phase for several harmonic components. The frequency is assumed to be the nominal system frequency with a DC link connecting the two systems.

3.1 Summary

The following list summarizes the problems explained above:

- In internet distributed simulation the RRT between simulators is several tens of ms
- If EMT values are exchanged among simulators, a time step of 10 ms (50 Hz AC) or smaller is required according to the sampling theorem
- Variable system frequency is not supported by classic static phasors
- EMT-Phasor-Interface requires extraction of the phasor information and may alter the simulation data

Section 4 describes the real-time solver under development which is supposed to avoid / cope with these limitations.

4. Basic Structure of the Real-Time Solver

Typically, geographically distributed simulators are connected via Wide Area Networks (WAN) based on VPNs. In addition to the challenge of partitioning the model and creating interfaces for parallel execution, the communication delay can be even larger than the simulation time step which is typically used in power system EMT simulations. The real-time solver described in this section aims at increasing the simulation time step to cancel or at least minimize this large difference between the step size and the communication delay and avoid the need for an extraction of the phasor information before sending them to other simulators.

The base of the solver is the nodal analysis as it is the case for many commercial EMT solvers. In addition to this, resistive companion models are introduced to increase the number of component types that can be directly solved with this method. The novel part about the solver is the use of dynamic phasors for the representation of state variables instead of classic phasors or electromagnetic transient variables.

The following three subsections provide an overview of these three concepts and how they are employed in the solver.

4.1 Nodal Analysis

The nodal analysis method provides a schematic way to find a set of equations that fully represent an electric circuit consisting of current sources and resistances. Therefore, it is a common algorithm used in circuit simulation software. This section describes the general procedure.

A circuit with b branches has $2b$ unknowns since there are b voltages and b currents. Hence, $2b$ linear independent equations are required to solve the circuit. If the circuit has n nodes and b branches, it has

- n Kirchoff's current law (KCL) equations
- b Kirchoff's voltage law (KVL) equations
- b characteristic equations (Ohm's law)

The nodal analysis method reduces the number of equations that need to be solved simultaneously. $n-1$ voltage variables are defined and solved, writing $n-1$ KCL based equations. A circuit can be solved using nodal analysis, by performing the following steps:

- Step 1: Select a reference node (mathematical ground) and number the remaining $n - 1$ nodes, that are the independent voltage variables.
- Step 2: Represent every branch current i as a function of node voltage variables v with the general expression $i_k = g(v)$.
- Step 3: Write $n-1$ KCL based equations in terms of node voltage variables. The resulting equations can be written in matrix form and must be solved for v.

In matrix form, the equation to be solved can be written as

$$G \cdot x = A \tag{1}$$

where x is the vector of unknown variables, G is the system matrix and A is the vector of known sources. The advantage of writing the equations in matrix form is that the matrix can be populated in a very systematic way. For example, a resistance that is connected between nodes h and j can be considered by stamping the matrix of the complete system in rows and columns h and j as follows:

$$G = \begin{array}{c} \vdots \\ h \\ j \\ \vdots \end{array} \overset{\begin{array}{cccc} \cdots & h & j & \cdots \end{array}}{\begin{bmatrix} & & & \\ & \frac{1}{R} & -\frac{1}{R} & \\ & -\frac{1}{R} & \frac{1}{R} & \\ & & & \end{bmatrix}} \quad (2)$$

An ideal current source on the other hand does not alter the system matrix but the right-hand side vector of the equation:

$$A = \begin{array}{c} \vdots \\ h \\ j \\ \vdots \end{array} \begin{bmatrix} \vdots \\ I \\ -I \\ \vdots \end{bmatrix} \quad (3)$$

This method is limited to resistances and cannot represent other passive components like capacitances and inductances.

4.2 Resistive Companion Method

The nodal analysis method is further extended by resistive companion models. Components that store energy, such as inductors and capacitors require some kind of “memory” of simulation steps in the past. Applying the trapezoidal integration rule to the following equation which describes an inductor

$$\frac{d}{dt} i(t) = \frac{1}{L} \cdot v(t) \quad (4)$$

results in

$$i(k+1) = i(k) + \frac{\Delta t}{2L} (v(k) + v(k+1)) \quad (5)$$

where Δt is the simulation time step. Equation (5) can be visualized as in Figure 4.1 with the equivalent resistance R_L and the history current A_L:

$$R_L = \frac{\Delta t}{2L} \quad (6)$$

$$A_L(k) = i(k) + \frac{\Delta t}{2L} v(k) \quad (7)$$

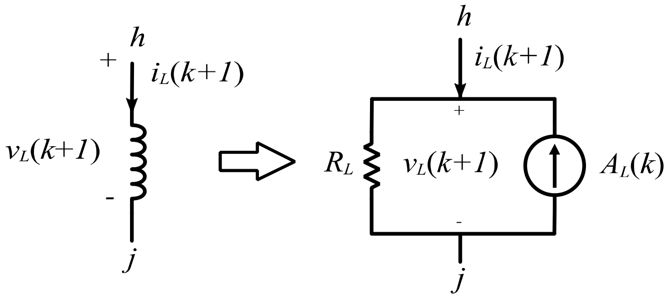


Figure 4.1: Resistive companion transformation of inductor

At this point, the matrix stamp presented in Section 4.1 can be utilized again since the resulting model consists only of a current source and a resistance. The same approach can be applied to capacitances as shown in the appendix.

4.3 Dynamic Phasors

In commercial real-time solvers, the nodal analysis method is employed with time domain variables or classic phasors as shown in Section 4.1 and 4.2. As mentioned before, the aim of the dynamic phasor solver is to take advantage of the phasor representation while maintaining the variable frequency feature of EMT simulations. Furthermore, an increased step size is not only desirable for distributed simulation; it also facilitates the calculation of larger grids on one computation node.

Dynamic phasors were initially developed for power electronics analysis [9]. Later, the concept was extended to power systems analysis [10]. Subsequently, the authors of [11] describe the use of dynamic phasors for power system simulation. Besides, they are still used to construct efficient models for the dynamics of switching gate phenomena with a high level of detail to simulate the integration of new DERs and HVDC converter technologies [12] [13]. Fault analysis and unbalanced conditions are other important research topics in which dynamic phasors allow larger models and simulations that are more efficient. Asymmetrical faults are studied in [14] [15] [16] [17]. In the last two articles, the behaviour of AC machines like Doubly-Fed Induction Generators (DFIG) or synchronous generators are evaluated using dynamic phasors. In [18] an effort to generalize the dynamic study with dynamic phasors is made by modelling and validating a multiple synchronous generator test grid. The goal was twofold: application of dynamic phasors for multi-source; multi-frequency systems and modelling of systems with time-varying frequencies.

Although dynamic phasors allow a significant saving in terms of computational cost, the idea to apply the concept to real-time simulation is still fairly novel. Commercial simulators like RTDS and OPAL-RT offer analytic modelling tools being able to perform EMT simulations, e.g. eMEGAsim developed by OPAL-RT. OPAL-RT introduced also simulation tools in the traditional complex phasor domain called ePHASORsim, which is limited to system fundamental frequency. Dynamic phasors might allow larger systems to be simulated in real-time with larger time-steps (i.e. milliseconds instead of microseconds) while catching the dynamic behaviour of a system with frequency deviation.

As mentioned in Section 3, the exchange of time-domain values among simulators imposes strong requirements on the sampling rate. Therefore, previous work already introduced dynamic phasors as a means of exchanging data in the frequency domain rather than the time domain [19]. However, this requires the extraction of the dynamic phasors from the time domain signal or every simulation step. Instead, we propose to simulate the entire system in dynamic phasors to be able to increase the simulation time step and to avoid the conversion from the time domain to the frequency domain.

In the following, the general approach of dynamic phasors for power system simulation is explained while pointing out the main features that are of interest for the real-time solver and distributed simulation. Using dynamic phasors, it is possible to treat an AC signal as a DC signal without losing its dynamic properties as it is the case when using static phasors in power system analysis. Instead of fixing the frequency, the signal is shifted by the system frequency, e.g. 50 Hz. Besides, one time domain variable can be approximated by several dynamic phasors of different harmonics, each of these shifted by their center frequency. However, this shift only decreases the maximum frequency of the simulated signals if all frequencies of interest lie in a small band around these center frequencies. The fundamental frequency of power systems is normally varying in a region close to the nominal system frequency. Hence, the bandpass limitation is fulfilled. The shift in the frequency domain for a bandpass signal represented in grey is visualized in Figure 4.2. The signal is shifted from the center frequency ω_c to $\omega = 0$.

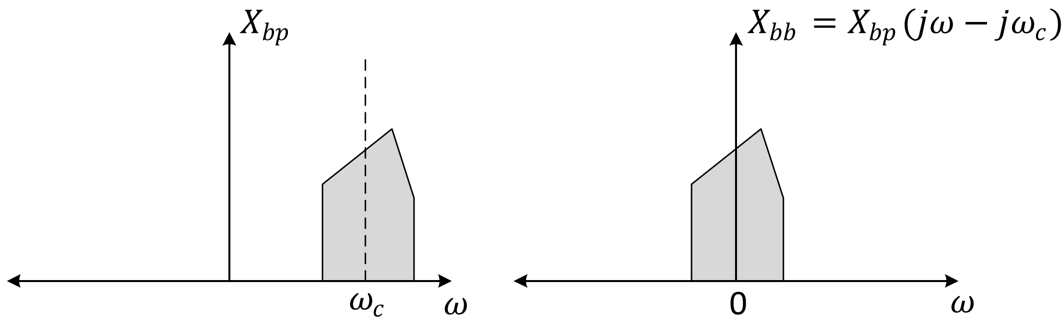


Figure 4.2: Frequency shift of dynamic phasors

The bandpass signal X_{bp} centered around ω_c is real valued and can be represented in the right half plane of the frequency spectrum. The shifted signal, which is also denoted baseband signal, X_{bb} features a smaller maximum frequency. According to the sampling theorem, the baseband signal requires a smaller sampling rate to be represented correctly. This property is very important in the application of real-time simulation since the RTT between two simulators in different locations can be very significant. In case of pan-European simulations, the RTT has been found to be several tens of ms [20], whereas links between Europe and the US can exhibit an RTT well over 100 ms [1]. Therefore, the default time step of 50 μs, used by many commercial real-time simulators, does not allow a data exchange between the simulators for every simulation step without compensation for the communication delay.

In the following, the general dynamic phasor approach is explained which is the basis of the simulation example in the next section. First, the time domain signal x is approximated with a Fourier series representation:

$$x(\tau) = \sum_k X_k(t) e^{jk\omega_s(\tau)} \quad (8)$$

where $\tau \in (t - T, t]$. The k^{th} coefficient is determined by

$$X_k(t) = \langle x \rangle_k(t) = \frac{1}{T} \int_{tT}^{T} x(\tau) e^{-jk\omega_s(\tau)} d\tau \quad (9)$$

where ω_s is the fundamental system frequency and $k\omega_s$ are its harmonics. Deriving equation (9) leads to

$$\frac{d}{dt} \langle x \rangle_k(t) = \left\langle \frac{d}{dt} x \right\rangle_k (t) - jk\omega_s \langle x \rangle_k(t) \quad (10)$$

Accordingly, a state space model of the general form

$$\frac{d}{dt} x(t) = f(x(t), u(t)) \quad (11)$$

would be transformed to the equation given in (12).

$$\frac{d}{dt}\langle x\rangle_k(t) = \langle f(x(t), u(t))\rangle_k - jk\omega_s\langle x\rangle_k(t) \tag{12}$$

Applying (12) to the equation of an inductance

$$\frac{d}{dt}i(t) = \frac{1}{L}\cdot v(t) \tag{13}$$

results in the following equation for the fundamental dynamic phasor:

$$\frac{d}{dt}\langle i\rangle_1(t) = \frac{1}{L}\cdot\langle v\rangle_1(t) - j\omega_s\langle i\rangle_1(t) \tag{14}$$

4.4 Summary

This section describes the concepts which are combined in the development of the real-time solver, DPsim: Nodal analysis, resistive companion method and dynamic phasors. The base of the solver is the representation of the circuit according to the nodal analysis which is extended by the resistive companion method. Instead of using the real voltage and current variables, DPsim calculates in the dynamic phasor domain.

In Section 5, a simple circuit, that includes an inductance modelled according to the approach described previously, is simulated using the EMT and dynamic phasor approach for different time steps.

5. Integration of Nonlinear Models

Nonlinear models such as the synchronous machine can be included into the MNA system solution as Norton or Thevenin equivalent. This results again in an equivalent conductance and voltage or current source that are stamped into the system matrix and the source vector. In contrast to linear models, nonlinear models may require an update every time step not just of the source but also the conductance in the system matrix.

The following, explains the necessary steps to solve the system with a general nonlinear model:

During the initialization, the equivalent conductance and source, which are calculated from a previous load flow solution, are stamped into the system as presented in Figure 5.1. Up to this point, the nonlinear component does not behave different from the linear components.

Then for every time step, the voltage or current solution of the system, depending on the type of equivalent model, is sent to the nonlinear model which calculates new equivalent values that need to be stamped into the system. Figure 5.1 shows the procedure for a Norton equivalent.

To improve the convergence, it is possible to conduct several Newton-Raphson iterations over the voltage for every time step. However, we are trying to avoid this technique whenever possible in order to guarantee real-time performance.

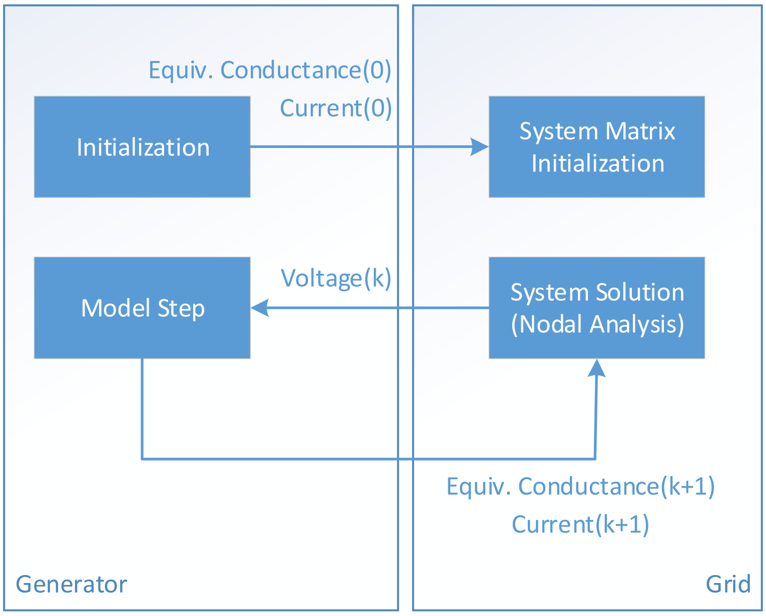


Figure 5.1: Integration of complex models with MNA as Norton equivalent

In the case of the synchronous generator in the dq reference frame, the stator values of the generator are exchanged between the system and the generator model. Therefore, it is required to transform the values from the abc reference frame to per unit values and the dq reference frame and vice versa:

$$v_{dq0}(k) = K_{sr}\big(\theta(k)\big)\, v_{abc}(k) \frac{1}{v_{base}} \qquad (15)$$

6. Simulation Results Using the Real-Time Solver

6.1 Basic Circuit Simulation

To support the theoretical advantage of dynamic phasor over EMT simulations, we present the simulation results for a simple circuit as depicted in Figure 6.1. The circuit consists of an AC voltage source of $V_{Source} = 1\,kV$ peak voltage with an internal resistance of $R_{Source} = 1\,\Omega$, a RX-series element of $R_{Line} = 1\,\Omega$ and $L_{Line} = 100\,mH$, and a load resistance of $R_{Load} = 100\,\Omega$. Internally, the voltage source is transformed to its Norton equivalent.

The simulation scenario is as follows. At 0.2 s, the load resistance is decreased to 50 Ω, and at 0.4 s the frequency of the AC voltage source is decreased from 50 Hz to 45 Hz. This scenario is simulated for different time steps between 50 µs and 40 ms using the real-time solver described before. The EMT simulations are based on the traditional approach using physical variables instead of dynamic phasors. In the following, we compare the voltage V_{Load} across the load resistance.

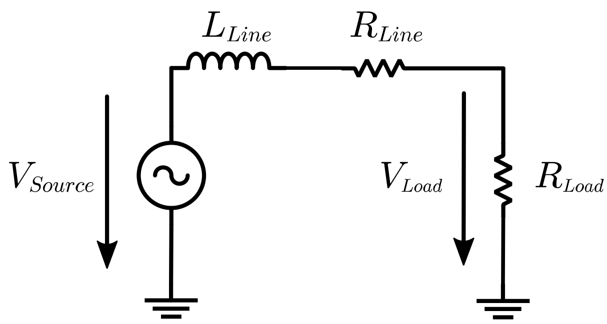


Figure 6.1: Example circuit for the comparison of dynamic phasor and EMT simulations

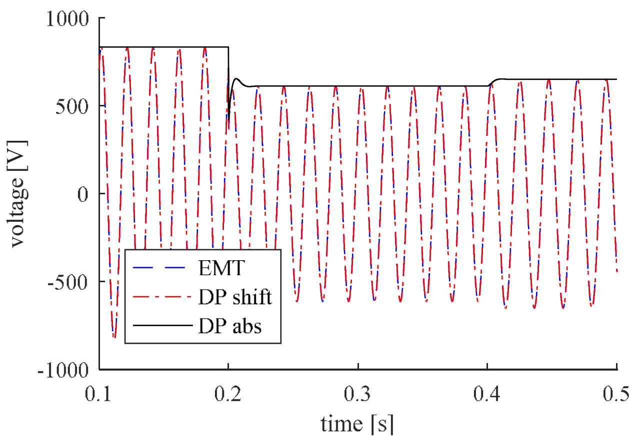


Figure 6.2: Comparison of dynamic phasors and EMT simulation for time steps of 50 µs

As can be seen in Figure 6.2 and Table 1, the results are almost identical for time steps of 50 µs. Figure 6.2 shows the EMT results, the absolute value of the fundamental dynamic phasor and the time domain signal of the fundamental dynamic phasors after it is shifted back by 50 Hz in the frequency domain.

Table 1: Simulation results for different time steps

Timestep [ms]	EMT	EMT interp.	DP	DP interp.
0.05	0	-	3.97E-05	-

1	105.61	138.92	97.004	110.75
5	7912.6	21133	2086.7	588.39
10	2.53E+05	2.24E+05	5764.8	914.98
15	3.84E+05	4.58E+05	12080	2050.3
20	2.48E+05	1.02E+06	16898	3187.9
25	1.58E+05	5.70E+05	17379	3937.1
30	1.43E+05	3.19E+05	26352	3271.9
35	3.67E+05	4.77E+05	29534	3456.7
40	2.70E+05	9.91E+05	28822	3643.7

The shift and transformation into the time domain is accomplished by taking the real part of the signal after applying equation (8).

$$x_{re}(\tau) = Re\left\{\sum_k X_k(t)e^{jk\omega_s(\tau)}\right\} \tag{16}$$

Furthermore, Table 1 depicts the mean squared error for the signals after linear interpolation. It is important to point out that the interpolation of the dynamic phasors is applied for real and imaginary part separately and before shifting the signal back to 50 Hz. Comparing the 20 ms time step results presented Figure 6.3 and Figure 6.4, it can be seen that the dynamic phasor simulation is very accurate even for large time steps. Without interpolation, the fundamental sinusoidal is not represented correctly by the dynamic phasor values since the number of data points is too small.

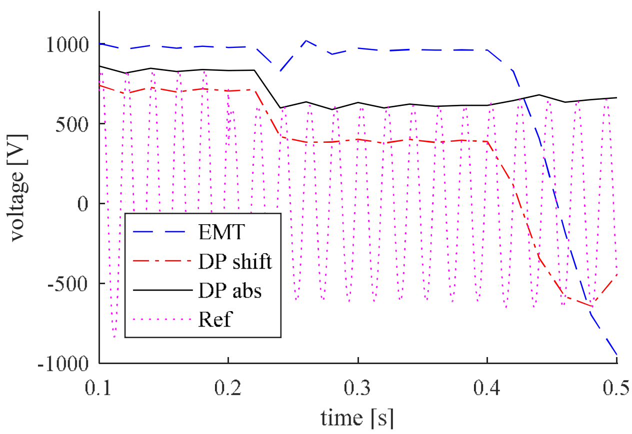


Figure 6.3: Comparison of dynamic phasors and EMT simulation for time steps of 20 ms

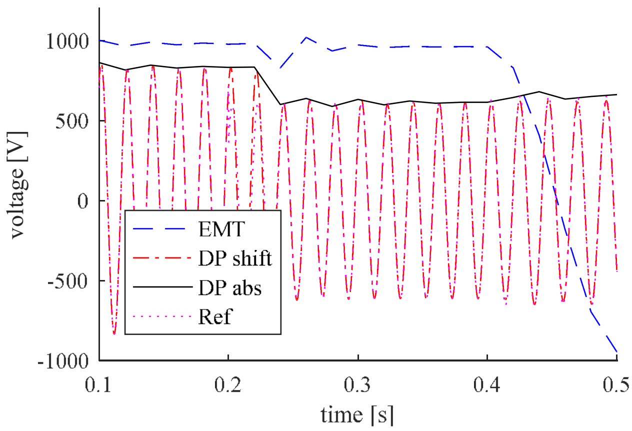


Figure 6.4: Comparison of dynamic phasors and EMT simulation for time steps of 20 ms with interpolation

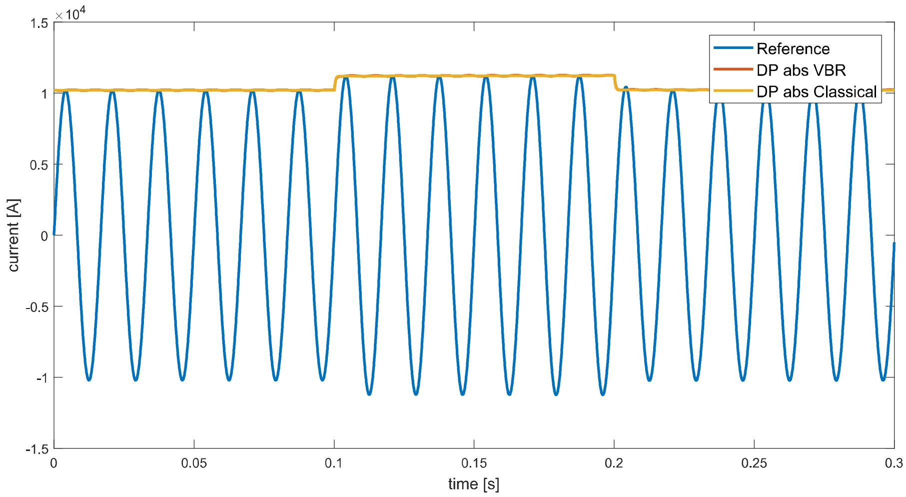
From Table 1, it can be concluded that the error is growing much slower for dynamic phasor simulations. Tens of ms can be a feasible range of time steps for dynamic phasor simulations if the system transients are not too fast. Therefore, the dynamic phasor approach could enable distributed simulations without having to compensate for the communication delay in some cases, for example, distributed simulation among participants in Europe.

6.2 Simulation of Synchronous Generator

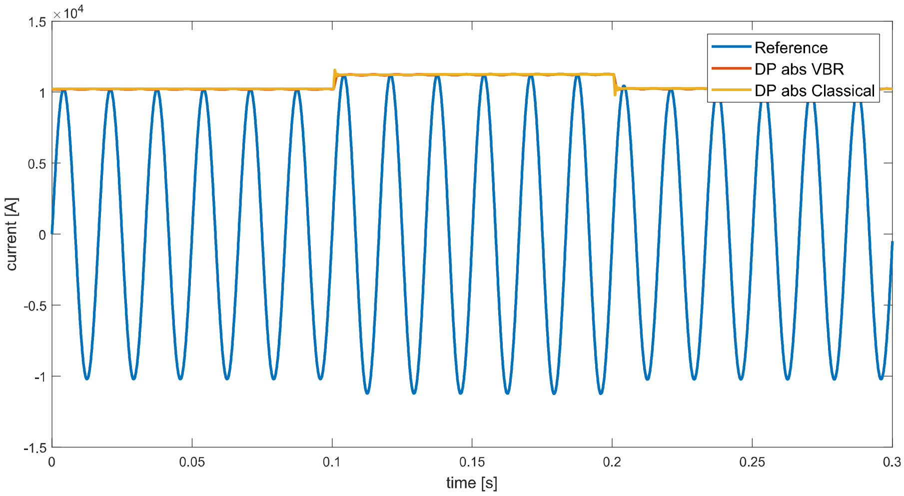
This section presents simulation results that we have obtained in the process of developing several synchronous generator models and validating them against a commercial solution, Simulink. The synchronous generator has been implemented both in the EMT and dynamic phasor domain using two approaches: the classic dq model and the voltage-behind-reactance (VBR) model. The following results demonstrate a load change and a three phase fault at the terminals of the synchronous generator. The model parameters are given in the Annex.

6.2.1 Load Change

A 10% load change is simulated to validate the generator models for small disturbance. After 0.1 seconds, the load is increased from $300\ \mathrm{MW}$ to $330\ \mathrm{MW}$ then 0.1 seconds later it is decreased to $300\ \mathrm{MW}$.



(a)



(b)

Figure 6.5: Magnitude of DP stator current for load change simulations for $50\mu s$ (a) and $500\mu s$ (b)

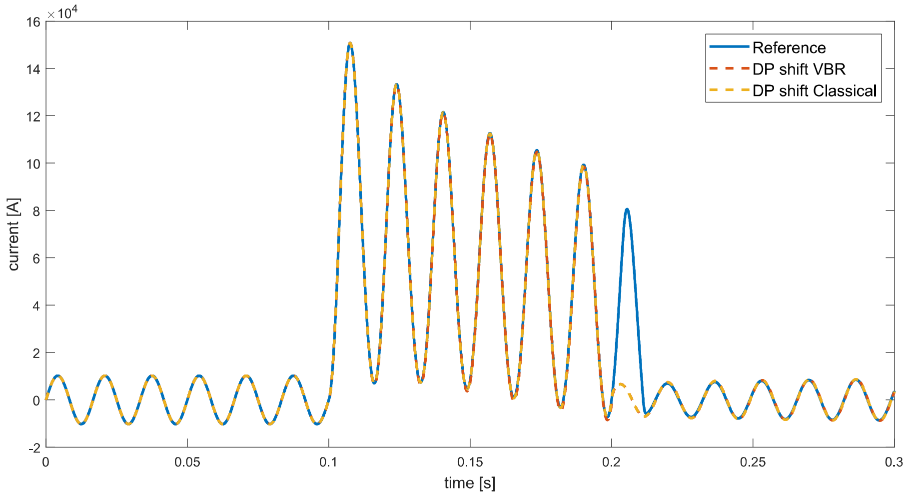
Figure 6.5 shows the stator currents for the load change of the reference, which is a Simulink EMT simulation, and the two dynamic phasor models for $50\ \mu s$ and $500\ \mu s$ simulation steps. It is important to note that in Figure 6.5, the dynamic phasor results are given in terms of the phasor magnitude value. Therefore, the phasor results envelope the EMT values. It can be seen that the DP results are satisfactory when compared to the reference results for both models. A quantitative analysis of the results can be found in Section 6.2.3.

6.2.2 Three-Phase Fault

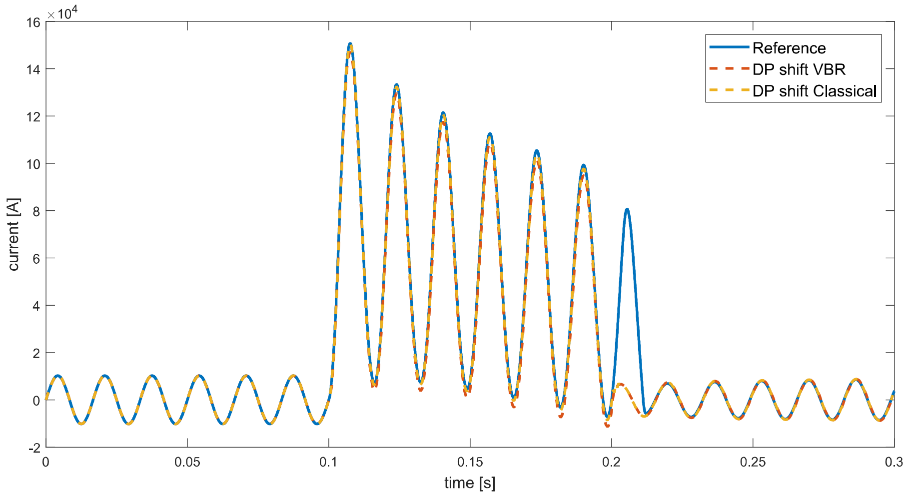
The three-phase fault simulation scenario is defined as follows: the fault is applied to the generator terminals after 0.1 seconds and cleared at 0.2 seconds simulation time. The simulation is stopped after 0.3 seconds.

Figure 6.6 shows the stator phase currents for the three-phase fault simulations, using instantaneous physical variables with time-steps of 50 μs and 500 μs respectively. The reference values are again obtained from a small time step Simulink simulation.

In this case, the dynamic phasor results have been post-processed by modulating them with a $60\ \mathrm{Hz}$ sinusoidal or even harmonics of the fundamental system frequency if more than the fundamental dynamic phasor would have been considered. This way, the instantaneous values are retrieved. The same procedure could have been applied to the previously presented results by applying equation (16). As in the previous section, the results fit very well with the reference.



(a)



(b)

Figure 6.6: DP Stator Current for three-phase fault simulations for $50\mu s$ (a) and $500\mu s$ (b)

6.2.3 Comparison between Dynamic Phasor and EMT Model

From literature it is known and our tests have shown that the VBR model is stable for much larger simulation steps compared to the classic model. Here, we want to show that using dynamic phasors this property of the VBR model can still be improved.

To evaluate the accuracy of the models, the maximum error and the root-mean-squared error (RMSE) is calculated for the three-phase and load change simulations. The error is calculated for two situations: in steady-state, between 0.0 seconds and 0.1 seconds, and during the simulated event, between 0.1 seconds and 0.2 seconds. The steady-state error is expressed in percent relative to the peak reference current in steady-state, and the error during the simulated event is in relation to the peak of the reference peak value during the load change.

As the reference results are given in instantaneous physical variables, the error is calculated based on the shifted dynamic phasor results.

Figure 6.7 presents the errors in function of the time step for the EMT VBR and DP VBR models. It is visible that the EMT VBR and the DP VBR models have similar accuracy for time-steps up to $4\ ms$. From this point on, the error of the EMT VBR model is growing faster than the error of the DP VBR model. This is another indicator that dynamic phasor simulation can be advantageous especially for large time step simulations including dynamics.

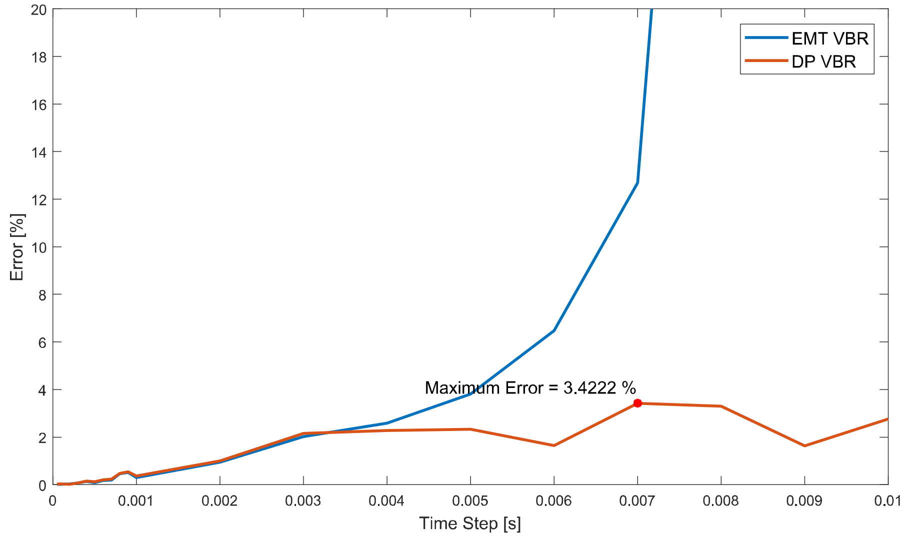


Figure 6.7: Root-mean-squared error after load change for different time steps for VBR EMT and DP

7. Integration with the Laboratory Interface

Like most simulation tools, the DPsim provides communication and data interfaces for several purposes:

- Real-time exchange of simulation signals
  - for co-simulation with other simulators and solvers
  - for hardware-in-the-loop testing
- User interfaces
  - for online monitoring of the simulation state
  - for online adjustments of parameters
- Offline storage and
  - export of simulation results (node voltages, branch currents, …)
  - import of timeseries profile data (load, production, …)

Currently, DPsim provides three solutions for realizing these interfaces:

- VILLASframework laboratory interface
- Logging of results to CSV files
- Python interface

7.1 VILLASframework Laboratory Interface

Deliverable D4.1 describes the VILLASframework laboratory interface. DPsim uses shared-memory to exchange real-time simulation data between the simulation kernel and the VILLASnode gateway. Both actors are executed as independent processes on a Linux operating system.

Interfaces to external systems, databases, files or the web interface is then handled by the wide range of supported interfaces of the VILLASnode gateway which in this case acts as a proxy to DPsim.

Responsibilities are clearly separated. VILLASnode handles input/output as well as translation between different protocols. DPsim focuses on the simulation and provides only a single type of interface to VILLASnode.

Shared-memory is a common method for inter-process communication (IPC) used on symmetric multi-processing (SMP) machines which enables user processes to exchange data without involving the operating system kernel. The absence of the operating system in the communication is crucial, as any involvement of the kernel would result in costly context switches which is undesired in a real-time context. Shared-memory IPC allows both processes (the solver and the gateway) to be executed in parallel while streaming their data with minimal latency over a shared queue.

The import and export of signals to and from the DPsim simulation kernel is handled via attributes. Each component defines a set of named attributes which represent static or tuneable parameters and states.

In the simulation scenario the user can specify which of the attributes shall be exported and which of the attributes shall be adjusted with data exchanged over the shared memory interface. With this simple and flexible mechanism almost all simulation states as well parameters can be exported or adjusted during a running real-time simulation.

DPsim provides several models of voltage and current sources which can be controlled via a tuneable parameter. With these controllable sources a simple ideal transformer model (ITM) interface can be realized as described in deliverable D4.4.

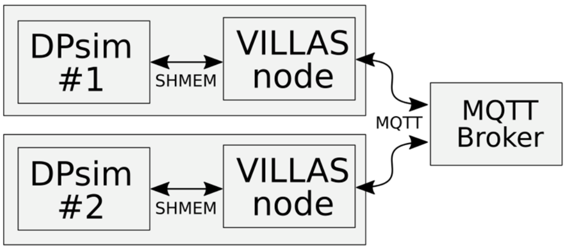


Figure 8: Example setup of two distributed DPsim instances interfaced via VILLASnode to an MQTT broker.

7.2 Logging of results to CSV files

The initial version of the DPsim solver used simple CSV files to log results for archival and post processing. This method is easy to setup and convenient for small simulations where post processing and analysis of simulation results is done in MATLAB or Python.

In the future, we envision to replace the CSV logger by integrating the VILLASnode interfaces directly into the solver core and thereby provide a wider range of supported file formats such as HDF5.

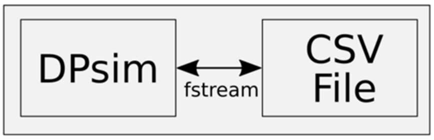


Figure 9: Simple logging of simulation results to CSV files.

7.3 Python Interface

The DPsim simulation kernel and its component models are implemented in the C++ programming language. The availability of good compilers and highly optimized software libraries like Eigen or Sundials for C++ were key factors for this decision. Only a compiled language like C++ with minimal runtime overhead where suitable for the implementation of a real-time solver.

Originally, network models have been directly defined in the C++ code. Later on, this technique has been complemented by a CIM importer which allows the user to directly load network models from CIM-XML files. This proved to be an adequate form to describe network topology and component parameters which are required by the solver. It relieves the user from defining the model in plain C++ code. However, CIM-XML is not suitable for the definition of more complex simulation scenarios with time varying parameters or topology changes caused by contingencies in the system (e.g. breaker events or faults).

In such cases, a scripting language like Python can be used to define scenarios by leveraging the flexibility of a general purpose imperative language.

DPsim provides Python bindings to most parts of the C++ programming interface. This allows the user to write a single Python script to:

1. Describe the network topology and parameters,
2. Define a simulation scenario with events or parameter changes
3. Execute the simulation
4. And analyze or plot the simulation results

The following listing shows an example Python script which implements the previously describes steps:

8. Conclusion

The first part of task 4.2 is the identification of challenges posed by our use case, the pan-European real-time simulation, and research on how these challenges were tackled in previous work. After presenting the results of this investigation, dynamic phasors are proposed as solution to solve one of the main problems of distributed real-time simulation, the large round trip time between the simulators which determines the minimum time step if the simulators are to exchange data for every step. Furthermore, a larger time step is beneficial for large system simulations.

The three main methods used to build the real-time solver are explained and we present a study that shows the advantage of using dynamic phasors for distributed real-time simulation. The focus of this study is on the simulation time step. The larger the time step, the lesser the impact of the communication delay between two geographically distributed simulators on the real-time data exchange.

As presented in Section 7, the integration with the co-simulation interface described in deliverable D4.1 is now completed and its usage is demonstrated in deliverable D4.4 where two small system are simulated in parallel and connected over the internet.

9. List of figures

Figure 1.1: Relations between deliverables in WP4 and other work ... 6

Figure 2.1: Example co-simulation scenario for WP 4 ... 8

Figure 3.1: ITM represented by current and voltage source ... 9

Figure 4.1: Resistive companion transformation of inductor ... 12

Figure 4.2: Frequency shift of dynamic phasors ... 14

Figure 5.1: Integration of complex models with MNA as Norton equivalent ... 16

Figure 6.1: Example circuit for the comparison of dynamic phasor and EMT simulations ... 17

Figure 6.2: Comparison of dynamic phasors and EMT simulation for time steps of 50 μs ... 17

Figure 6.3: Comparison of dynamic phasors and EMT simulation for time steps of 20 ms ... 18

Figure 6.4: Comparison of dynamic phasors and EMT simulation for time steps of 20 ms with interpolation ... 19

Figure 6.5: Magnitude of DP stator current for load change simulations for $\mathbf{50\mu s}$ (a) and $\mathbf{500\mu s}$ (b) ... 20

Figure 6.6: DP Stator Current for three-phase fault simulations for $\mathbf{50\mu s}$ (a) and $\mathbf{500\mu s}$ (b) ... 21

Figure 6.7: Root-mean-squared error after load change for different time steps for VBR EMT and DP ... 22

Figure 14: Example setup of two distributed DPsim instances interfaced via VILLASnode to an MQTT broker. ... 24

Figure 15: Simple logging of simulation results to CSV files. ... 24

10. List of Tables

Table 1: Simulation results for different time steps 17

11. References

[1] X. Guillaud, M. O. Faruque et al., “Applications of real-time simulation technologies in power and energy systems,” IEEE Power and Energy Technology Systems Journal, vol. 2, no. 3, pp. 103–115, 2015.

[2] E. Bompard, A. Monti et al., “A multi-site real-time co-simulation platform for the testing of control strategies of distributed storage and V2G in distribution networks,” in Power Electronics and Applications (EPE’16 ECCE Europe), 2016 18th European Conference on. IEEE, 2016, pp. 1–9.

[3] A. Benigni and A. Monti, “A parallel approach to real-time simulation of power electronics systems,” IEEE Transactions on Power Electronics, vol. 30, no. 9, pp. 5192–5206, 2015.

[4] M. Stevic, A. Monti, and A. Benigni, “Development of a simulator-to-simulator interface for geographically distributed simulation of power systems in real time,” in IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, Nov 2015, pp. 005 020–005 025.

[5] J. Belanger, P. Venne, and J. Paquin, “The what, where and why of real-time simulation,” Planet RT, vol. 1, no. 1, pp. 25–29, 2010.

[6] J. E. Schutt-Aine, "Latency insertion method (LIM) for the fast transient simulation of large networks", IEEE Trans. Circuits Syst. I Fundam. Theory Appl., vol. 48, no. 1, pp. 81-89, Jan. 2001.

[7] M. Stevic, S. Vogel et al., “Virtual integration of laboratories over long distance for real-time co-simulation of power systems,” in IECON 2016 42nd Annual Conference of the IEEE Industrial Electronics Society, Oct 2016, pp. 6717–6721.

[8] Wei Ren, Steurer, M., Baldwin and T.L., “Improve the Stability and the Accuracy of Power Hardware-in-the-Loop Simulation by Selecting Appropriate Interface Algorithms”, vol. 44, no. 4, pp. 1286-1294

[9] S. R. Sanders, J. M. Noworolski et al., “Generalized averaging method for power conversion circuits,” IEEE Transactions on Power Electronics, vol. 6, no. 2, pp. 251–259, 1991.

[10] V. Venkatasubramanian, H. Schattler, and J. Zaborszky, “Fast timevarying phasor analysis in the balanced three-phase large electric power system,” IEEE Transactions on Automatic Control, vol. 40, no. 11, pp. 1975–1982, 1995.

[11] T. Demiray, G. Andersson, and L. Busarello, “Evaluation study for the simulation of power system transients using dynamic phasor models,” in Transmission and Distribution Conference and Exposition: Latin America, 2008 IEEE/PES. IEEE, 2008, pp. 1–6.

[12] A. Coronado-Mendoza, J. L. Bernal-Agustín, and J. A. Domínguez-Navarro, “Photovoltaic boost converter system with dynamic phasors modelling,” Electric Power Systems Research, vol. 81, no. 9, pp. 1840–1848, 2011.

[13] H. Zhu, Z. Cai et al., “Hybrid-model transient stability simulation using dynamic phasors based hvdc system model,” Electric power systems research, vol. 76, no. 6, pp. 582–591, 2006.[16]

[14] A. M. Stankovic and T. Aydin, “Analysis of asymmetrical faults in power systems using dynamic phasors,” IEEE Transactions on Power Systems, vol. 15, no. 3, pp. 1062–1068, 2000.

[15] R. H. Salim and R. A. Ramos, “A model-based approach for small-signal stability assessment of unbalanced power systems,” IEEE Transactions on Power Systems, vol. 27, no. 4, pp. 2006–2014, 2012.

[16] A. M. Stankovic, S. R. Sanders, and T. Aydin, “Dynamic phasors in modeling and analysis of unbalanced polyphase ac machines,” IEEE Transactions on Energy Conversion, vol. 17, no. 1, pp. 107–113, 2002.

[17] T. Demiray, F. Milano, and G. Andersson, “Dynamic phasor modeling of the doubly-fed induction generator under unbalanced conditions,” in Power Tech, 2007 IEEE Lausanne. IEEE, 2007, pp. 1049–1054.

[18] T. Yang, S. Bozhko et al., “Dynamic phasor modeling of multigenerator variable frequency electrical power systems,” IEEE Transactions on Power Systems, vol. 31, no. 1, pp. 563–571, 2016.

[19] M. Stevic, S. Vogel et al., “Feasibility of geographically distributed real-time simulation of HVDC system interconnected with AC networks,” in 2015 IEEE Eindhoven PowerTech, June 2015, pp. 1–5.

[20] Kundur, Prabha; Balu, Neal J.; Lauby, Mark G. Power system stability and control. New York: McGraw-hill, 1994.

12. List of Abbreviations

EMT	Electromagnetic Transient
DFIG	Doubly-Fed Induction Generator
ITM	Ideal Transformer Model
HIL	Hardware-In-the-Loop
RES	Renewable Energy Sources
RTT	round-trip time
SIL	Software-In-the-Loop
PHIL	Power-Hardware-In-the-Loop
RTT	Round-Trip Time
WAN	Wide Area Networks

Annex

A.1 Synchronous Generator Simulation Parameters

According to the example presented in [20]:

Nominal Power	555 MW
Phase-to-phase RMS Voltage	24 KV
Nominal frequency	60 Hz
Number of poles	2
Inertia coefficient H	3.7 s
Rs	0.003 pu
Ll	0.15 pu
Lmd	1.6599 pu
Lmq	1.61 pu
Rfd	0.0006 pu
$Llfd$	0.1648 pu
Rkd	0.0284 pu
$Llkd$	0.1713 pu
$Rkq1$	0.0062 pu
$Llkq1$	0.7252 pu
$Rkq2$	0.0237 pu
$Llkq2$	0.125 pu