DOI: 10.1109/IESES.2018.8349934

Publisher version: https://ieeexplore.ieee.org/document/8349934

Distributed Real-Time Co-Simulation as a Service

Markus Mirz, Steffen Vogel, Bettina Schäfer, Antonello Monti
Institute for Automation of Complex Power Systems
E.ON Energy Research Center
RWTH Aachen University
Aachen, Germany
Email: {mmirz, stvogel, bschaefer, amonti}eonerc.rwth-aachen.de

***Abstract*—The increase of faster dynamics in power systems has led to growing interest in new simulation solutions, especially in the field of hardware-in-the-loop and real-time simulation. Due to the size of power systems, detailed simulation of the faster dynamics is only feasible for a section of the system, whereas the rest is usually modeled as an infinite power bus. The aim of this work is to present a solution which would allow the representation of a significant portion of the dynamics that are usually not captured by the infinite power bus approach and enable the joint simulation among multiple simulation laboratories to share this dynamic model of the network beyond the boundaries of the detailed simulation. Furthermore, the presented architecture should allow the virtualization of each of these laboratories in cases where this coarse model of the neighboring grid sections is sufficient. First distributed simulation examples show the current status of our implementation of the architecture presented here.**

I. INTRODUCTION

The current evolution of the grid characterized by a higher penetration of renewable energy sources is significantly changing the main dynamics of the power system. This change is reflected in two main aspects:

- Higher volatility of the generation input
- Faster dynamics due to reduction of the mechanical inertia connected to the system

This process has attracted a growing interest in the development of new simulation solutions and in particular a significant development of the concepts of real-time simulation and hardware-in-the-loop (HIL). These technologies are widely used both in the analysis and in the development of advanced automation solutions intended to overcome the new challenges that power system dynamics are facing [1]. The lack of knowledge about the behavior of a power grid widely based on power electronics devices is asking for new simulation analysis able to give insight at different levels of details: from the slow dynamics of interaction of the large system scenarios to the complex non-linear interactions of switching devices at local level. On the other hand, it is becoming more and more clear that the computational complexity is such that the problem can not be solved by a single laboratory. Furthermore, availability of data on electrical infrastructure is also sometimes not easy to share among scientists. This scenario is calling for the development of a new architecture of real-time simulation that could support integration of knowledge among different players in the creation of large and meaningful scenarios [2]. The work proposed in this paper fits exactly in this direction of research, providing a platform which facilitates the joint research on large scale power systems.

II. THE VISION

Main goal of the work described in this paper is the creation of a network of laboratories sharing knowledge and creating large scenarios of interaction to tackle the appropriate level of complexity. The idea is that to achieve such goals two components are necessary:

- A programmable system to support data interchange among laboratories set-up in real-time
- A simulation platform that is able to model only the dynamics that are compatible with the bandwidth of the interface.

Experimental results show that, with the current internet infrastructure, it is feasible to reach round-trip times within some milliseconds within distances in the order of hundred kilometers [3]. Such performance is in line with the main dynamics present at transmission level typically in the order of few hertz. Coherently, it would be interesting to have available a real-time simulation with a resolution in the range of milliseconds. It should be clarified that while operating at this time step does not allow the representation of every possible dynamic interaction among different sections of a network, on the other hand it provides a much more significant representation than what is possible within a single lab. In effect, currently, given the computational limits of whatever simulation platform, each laboratory needs to set a boundary condition to the analysis which is typically described in terms of infinite power bus. The architecture and the combination of tools proposed in the paper allows the substitution of such ideal hypothesis with an interaction model able to capture a significant portion of the dynamic interaction.

One step further is the virtualization of the laboratory in the cloud. Given that it is not reasonable that all the laboratories will be permanently connected, it is important to define a process that supports the substitution of the laboratory with its virtual representation. Each institution could run its own real-time simulation either in the cloud or locally in their facilities depending on their current activity while still contributing to the overall network. This approach has two benefits:

- Other participants always have some kind of power system representation available which they can connect to
- Each participant can move his simulation from the cloud to his local simulation environment to connect hardware at any time without coordination of other participants.

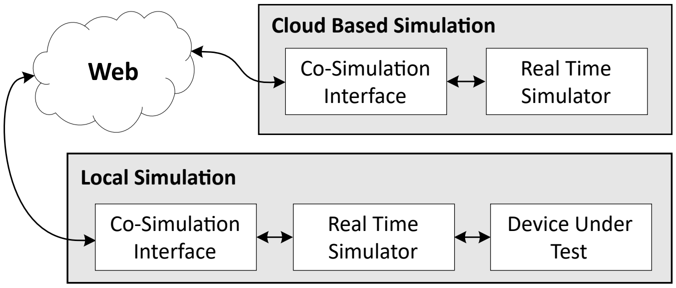

Fig. 1. Co-simulation as a service concept.

Today's availability of cloud computing resources and the free software, which was employed for this study, can form the basis for an always running real-time simulation collaboratively operated by the research community. Open protocols and the free software presented in this paper are key ingredients for this idea.

Other research disciplines already collaboratively operate such test beds. Examples are the Decentralized Network 42 (DN42), which builds a virtual overlay network on top of the real Internet [4]. Operated by individuals and researchers, this network is used for training purposes and experimentation. Annual *Battle the Mesh* events are hosted by a community built around the development of open source mesh network stacks [5].

III. KEY COMPONENTS OF THE ARCHITECTURE

A. Geographically Distributed Real-Time Simulation

Geographically distributed real-time simulation as special case of distributed simulation has the advantage of being able to share computational load among several processing units. However, it is required to synchronize the simulators involved in the simulation. While distributed simulation does not specify the type of network connection between the simulators, geographically distributed simulation excludes high speed network interconnection, such as InfiniBand, since the connection can only be established through the Internet. Section III-B will further elaborate the challenges of this approach.

Apart from increasing the overall computational power of the simulator, there are more advantages of distributed real-time simulation [6], [7]. The following list summarizes the most important ones:

- Hardware and software in different simulation sites can be shared to facilitate remote software-in-the-loop (SIL) and (power-)hardware-in-the-loop (PHIL)
- Knowledge exchange is facilitated and encouraged since several research groups may work on the same case study without the need to move personnel and equipment
- 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
- Several algorithms to control, manage, or regulate systems can be tested in laboratories where no realistic models of the environment (e.g. power grid model) are available

B. Modelling of the Appropriate Resolution

As explained in Section III-A, the simulators are connected through the Internet. This results in a typical round-trip time (RTT) between two simulators in the range of milliseconds. If electromagnetic transient (EMT) values are exchanged among the simulators, a simulation time step of $10\,\mathrm{ms}$ ($50\,\mu\mathrm{Hz}$ AC) or smaller is required according to the sampling theorem. Besides, the typical simulation time step of EMT simulators is $50\,\mu\mathrm{s}$. Therefore, the relation between simulation time step and data exchange time step is very large and a lot of samples have to be sent at once.

The solution proposed [7] overcomes the problem of data size by compressing the information through an EMT-Phasor-Interface. This requires extraction of the phasor information and may alter simulation data. Such a data resolution calls for an appropriate simulation solver able to operate on the same time-scale and working directly in the phasor domain.

An alternative solution to data compression could be the simulation with static phasors. However, this approach would only cover simulations at fixed system frequency.

IV. VILLASNODE GATEWAY

Communication between real-time solver instances is realized by a dedicated gateway called VILLASnode. VILLASnode is part of the VILLASframework, a toolkit for distributed real-time simulation which is released as open source software under the GLPv3 license [8]. The gateway is a software component which runs on a real-time optimized Linux machine which also executes the DPsim solver (see section V).

Responsibilities between the solver and the gateway are clearly separated. This modular approach allows to use the components in combination as well as with other software. The gateway supports a variety of pluggable transports like UDP/IP, ZeroMQ, nanomsg, IEC61850 Sampled Values & GOOSE as well as interfaces to commercial digital real-time simulators, like OPAL-RT and RTDS.

In this paper, the gateway provides two interfaces for exchanging simulation data between the solver and over the Internet as depicted in Figure 2

A. Shared-memory Interface

A shared memory interface is used to exchange simulation data between one or more processes on a single compute node. In this paper, a shared memory region is used to exchange

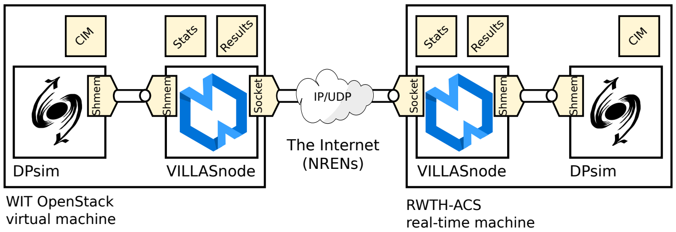

Fig. 2. Co-simulation architecture.

simulation data between the real-time phasor solver DPsim and the gateway.

The interface consists of two simple lock-free single-producer single-consumer (SPSC) queues for bidirectional communication between the processes. Shared-memory has been chosen over other inter-process communcation methods as it requires operating system (OS) support only for the initial setup. During a running simulation the OS remains in the background and does not disturb the execution of the solver. This is a key requirement for executing the solver in real time.

The shared memory interface has also been used for the first simulation case described in section VIII-B.

B. Network Interface

The communication between multiple gateways is realized by UDP/IP packets which are exchanged over the public Internet and the national research and education networks (NREN). Additional meta data such as time stamps and sequence numbers which is being exchanged is used to monitor the current quality of the connection and drop invalid or reordered packets.

By using Linux's NetEm queueing discipline, real network communication characteristics such as latency, packet loss and reordering can be emulated. This feature is utilized to anticipate the results of the distributed co-simulation. As part of the traffic control (TC) subsystem, NetEm is a special queuing discipline which is tightly integrated into Linux networking stack [9], [10] and allows good emulation of real communication characteristics in real-time.

V. DPSIM - DYNAMIC PHASOR REAL-TIME SIMULATOR

As described in Section III-B, the exchange of time-domain or EMT 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 [3]. The difficulty lies in the extraction of the dynamic phasors from the time domain signal for every simulation step. Here, the DPsim simulator offers an alternative. Instead of simulating the entire system in EMT, the simulation operates on dynamic phasors. This method avoids the conversion from the time domain to the frequency domain and improves the possible sampling rate with regard to Shannon's sampling theorem. Simulation results which demonstrate this advantage of dynamic phasors compared to EMT are presented in [11].

The dynamic phasor approach is combined with modified nodal analysis and resistive companion method. Therefore, the majority of network components is represented by the network admittance matrix whereas more complicated models such as synchronous generators are solved separately and interfaced through, for example, a Norton equivalent model [12].

VI. SCHEDULING AND SYNCHRONIZATION

A completely synchronized execution of both solvers over the Internet is not possible for two reasons:

- First, the communication latency of about 15 ms between the two sites exceeds the simulation time step of 1 ms.
- Secondly, the unreliable nature and the inherent jitter makes it impossible to provide any guarantees.

Therefore, all solvers are started synchronously in reference to a global point in time which was agreed upon before. During the execution of the co-simulation, every solver proceeds in real-time with its own time step without synchronization to its peers. The interface quantities described in section VIII are exchanged at periodical intervals which can, but do not have to, be equal to the simulation time step. This scheme allows multiple rates to be used for participating simulators as well as the interface itself.

To guarantee a synchronized start of the simulation as well as to avoid them drifting apart, all simulators must be synchronized to a global clock. This synchronization source is most conveniently provided by the global positioning system (GPS) and distributed by existing time synchronization protocols such as the network time protocol (NTP) or the precision time protocol (PTP/IEEE-1588).

VII. NETWORK MEASUREMENTS

Figure 3 shows the geographical location of routing hops used in the pan-European laboratory infrastructure which has been used to conduct the following simulations. The black line in this map shows the communication link between Waterford Institute of Technology (WIT) in Ireland and RWTH Aachen University (RWTH) in Germany. In total, the connection is comprised of 15 routers/hops. Figure 4 shows the distribution of round-trip time as well as the geographical distance across those hops. The average RTT measured between RWTH and

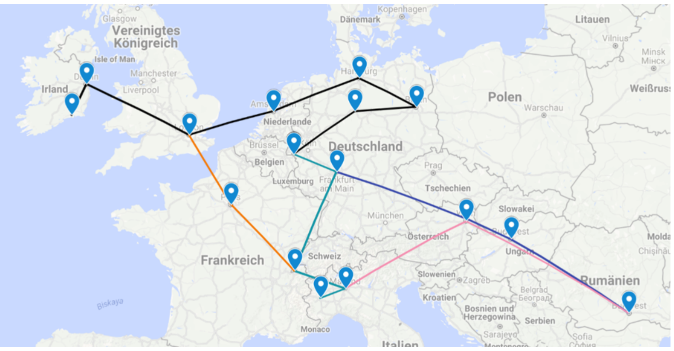

Fig. 3. Network connection of Pan-European Laboratory infrastructure.

WIT is $47.7\,\mathrm{ms}$ and was between $50\,\mathrm{ms}$ and $60\,\mathrm{ms}$ during the execution of the test simulations described in Section VIII.

These results show that the geographical distance between the sites is the main contributor of the RTT. The RTT is only one, but important, metric for evaluating the quality of a connection for real-time simulation. Another metric of interest is the maximum packet rate, which we can use to exchange simulation samples while keeping the packet loss below a certain threshold. For this test, VILLASnode is used to generate sample streams at different packet rates and number of values.

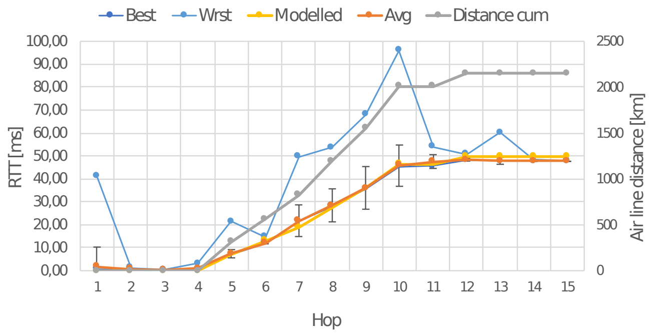

Fig. 4. RTT over communication hops.

Figure 5 shows the cumulative distribution of the RTT between RWTH and WIT in dependency to the sending rate. Not shown in this figure is the loss and reordering of packets, which gets significant with higher rates. At higher rates, the connection is susceptible to third party network traffic which causes loss and reordering.

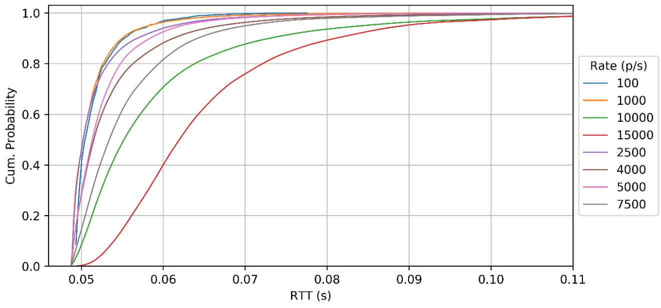

Fig. 5. Cumulative probability of RTT.

VIII. Co-simulation Example

The RTT delay measurements described before, allow an estimation of the dynamics that are able to propagate through the laboratory interface. However, real-time simulation test cases provide us with more details on the expected dynamic interaction between the distributed simulation laboratories since also the simulators themselves introduce latency.

A. Circuit Model

To validate the estimation of the delay and demonstrate the functionality of the interface between VILLASnode and DPsim, we present the simulation results for a simple circuit as depicted in Figure 6. The circuit consists of an AC voltage source of $V_{Source} = 10\,\mathrm{kV}$ peak voltage with a resistance of $R_{Source} = 1\,\Omega$, an RX-series element of $R_{Line} = 1\,\Omega$ and $L_{Line} = 1\,\mathrm{H}$ and a load resistance of $R_{Source} = 10\,\Omega$. Internally, the voltage source is transformed to its Norton equivalent.

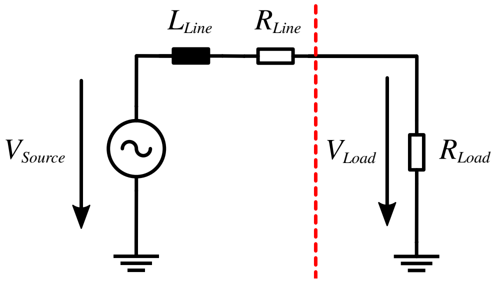

Fig. 6. Circuit model for distributed simulation.

For a first test simulation, the circuit depicted in Figure 6 is split into two parts, indicated by the red line, and each part is simulated in a different location. The two locations that we consider are our institute, RWTH, and WIT. The source and line model are simulated in RWTH on a dedicated Linux workstation while the load resistance is simulated in the WIT OpenStack installation.

The depicted circuit is simulated for four different cases. First, both parts are simulated within one instance of DPsim. This case serves as reference for the three co-simulation cases. Secondly, the circuit is simulated using two separate instances of DPsim running on one computer that are interconnected via VILLASnode in the most optimal way. Then, the co-simulation is executed including a network latency emulation between two DPsim instances running on the same machine. Finally, one DPsim instance is running on a machine in RWTH while the other DPsim instance is executed in WIT.

The simulation time step is $1\,\mathrm{ms}$ in all cases. In comparison to EMT based simulation, this time step is relatively large and allows us to exchange interface quantities without down sampling. EMT simulations with $50\,\mu\mathrm{s}$ time step would require a decimation to reduce the amount of data. At $t = 1\,\mathrm{s}$, the load resistance is changed from $10\,\Omega$ to $8\,\Omega$. This change is an ideal step, which would not appear like this in a real setup. Still, it is an interesting test case because the step introduces very high frequencies and, therefore, presents the worst-case scenario for the distributed co-simulation and dynamic phasor approach since only the fundamental phasor is used.

The ideal transformer model (ITM) [13] which is shown in Figure 7 interconnects the two network solutions where t_D describes the latency introduced by the interface. The ITM is introduced as follows: The external voltage source is used in the left part of the circuit whereas the external current source is integrated into the right side. The voltage and current values which are exchanged between the two simulations are the complex voltage and current values split into real and imaginary part.

It should be noted that the bandwidth of the signal passing through the interface can be increased by considering also the phasors of higher harmonics. Therefore, fast electromagnetic phenomena in one section of the network could be seen in the remote section of the network. Still, the bandwidth of a control loop which is across the co-simulation interface is restricted by the delay t_D of the interface.

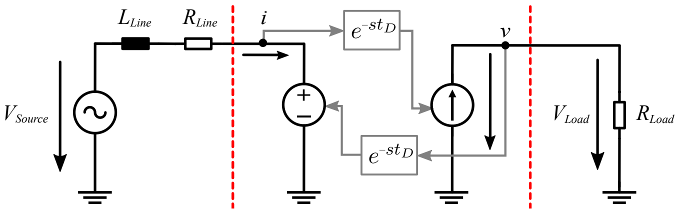

Fig. 7. Separated circuit with ITM and communication delay.

B. Local Co-Simulation Separated

For the co-simulation case where both DPsim instances are coupled in an optimal way, the communication delay is rather small but already visible. The delay between the monolithic reference simulation and the co-simulation is about $3\,\mathrm{ms}$ as can be seen in Figure 8. Since there is no communication network in between the two DPsim instances, this delay is only introduced by the software interface. DPsim is interfaced to VILLASnode through a shared memory segment and the two VILLASnodes instances, one for each DPsim instance, are exchanging data through a UDP connection.

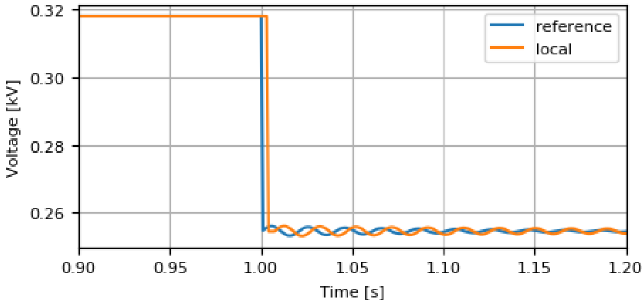

Fig. 8. Comparison of integrated reference simulation and local co-simulation.

C. Local Co-Simulation with Emulated Network Latency

In this case, the communication delay is emulated using the NetEm feature of VILLASnode. The properties of the artificial delay for sending data are set to $30\,\mathrm{ms} \pm 1\,\mathrm{ms}$, normal distributed for each VILLASnode. This results in an emulated RTT of about $60\,\mathrm{ms}$ which is close to the measured RTT in Section IV-B. Therefore, the delay in simulation should be comparable to the distributed case. As can be observed in Figure 9, the resulting delay in the simulation is about $32\,\mathrm{ms}$.

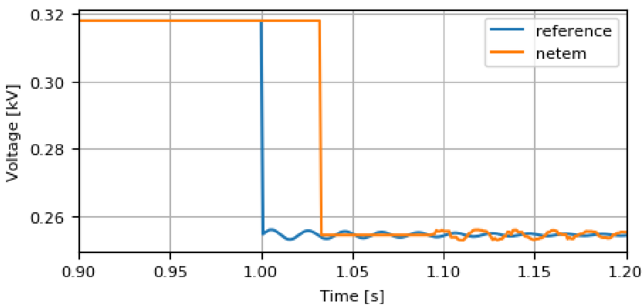

Fig. 9. Comparison of integrated reference simulation and local co-simulation with emulated network latency.

D. Distributed

As depicted in Figure 10, the distributed simulation between RWTH and WIT features a delay of about $33\,\mathrm{ms}$. Hence, the measurement of the delay as in Section IV-B and the simulation including emulated delay as presented in Section VIII-C allow a very good prediction of the behavior of the distributed simulation.

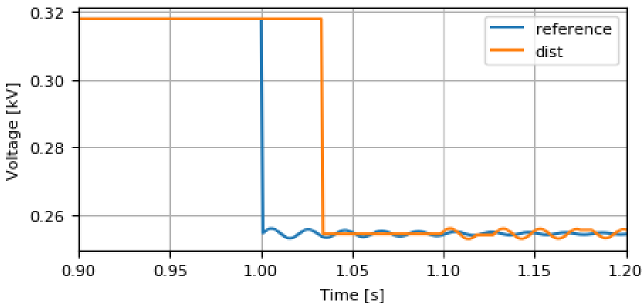

Fig. 10. Comparison of integrated reference simulation and distributed co-simulation among RWTH and WIT.

IX. SIMULATION OF SLOW DYNAMICS

The previous simulations are intended to show how the communication latency affects the distributed simulation in the extreme case. The main purpose of the proposed simulation solution is the investigation of slower dynamics. Slow dynamics can be represented by dynamic phasors very accurately as shown in Figure 11. Here, the voltage source is changing its frequency by $-1\,\mathrm{Hz}$ over $200\,\mathrm{ms}$. Figure 11 shows the EMT reference simulation which is simulated with a simulation time step of $50\,\mu\mathrm{s}$ and the absolute value as well as the shifted version of the dynamic phasor values are simulated with a time step of $1\,\mathrm{ms}$. The shifted version is the result of the complex phasor signal shifted by $50\,\mathrm{Hz}$ in the frequency domain. It can be seen that the dynamic phasor values are following the transient very well even though the simulation time step is much larger. In this case, the simulation is integrated and not interfaced through VILLASnode. This is an example of the dynamics that we want to be able to propagate between distributed simulators in the future.

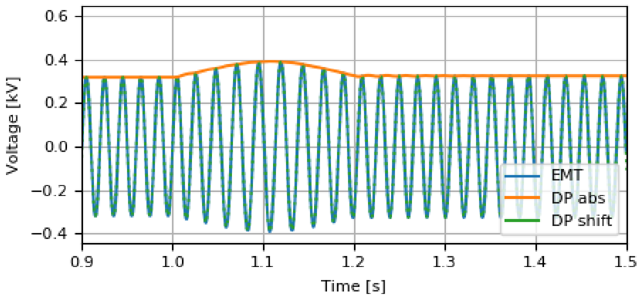

Fig. 11. Frequency change simulated for dynamic phasors and EMT.

X. CONCLUSION

Initial test cases have shown that DPsim and the VILLASnode gateway can be used to conduct simple geographically distributed simulations in real time over the Internet. Besides, it can be seen that the latency caused by VILLASnode is very small compared to the latency expected from the network connection with the co-simulation partner WIT.

It is presented that the communication network measurements and the simulation test cases are coherent. The comparison with the network connectivity measurement in Section IV-B shows that predictions of the delay in the simulation can be very accurate based on the network measurements.

Furthermore, the simulation highlights an important condition for the validity of co-simulation results. The dynamics of variables which are propagated between the simulators should not be faster than the communication delay or otherwise the results will differ from the integrated simulation results. This is exactly why dynamic phasors can be an important tool to support distributed real-time simulation. They allow the mapping of high frequencies to lower frequencies which decreases the impact of the communication delay. Hence, the dynamic behavior of the grid model beyond the detailed simulation in one laboratory can be improved as stated in Section I.

XI. OUTLOOK

Upcoming tests must validate these results using larger networks including multiple sources and detailed models, such as synchronous generators.

The transition of frequency changes as shown in [11] should be investigated. If fast events are of interest, the simulation could be extended to include dynamic phasors of higher harmonics as well. However, for system frequency control investigations, as conducted in the Horizon 2020 project RESERVE the fundamental phasor is the most important one [14]. In the frame of RESERVE, the presented components are developed as part of a pan-European real-time simulation infrastructure for the validation of innovative approaches to system level automation based on innovative ancillary service provision.

In the simulation case described here, no data processing is applied to the exchanged values. In the future, current simulation values at the interface could be extrapolated from previous values. This could further improve the correct propagation of fast dynamics through the co-simulation interface.

The achievable data exchange rates discussed in section IV-B are dependent on the current conditions of the communication medium. Future work could investigate adaptive methods to determine the optimal sending rate and its effects on the simulation fidelity. The real-time protocol (RTP) commonly used for streaming of audio / video content is a good candidate and can be used as the basis to develop a new variant of that protocol for exchanging simulation data.

XII. ACKNOWLEDGMENT

This project has received funding from the European Unions Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 727481.

REFERENCES

[1] M. D. O. Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, C. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, K. Strunz, M. Saeedifard, X. Wang, D. Shearer, and M. Paolone, "Real-time simulation technologies for power systems design, testing, and analysis," *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 2, pp. 63–73, June 2015.

[2] C. F. Covrig, G. D. Santi, G. Fulli, M. Masera, M. Olariaga, E. F. Bompard, G. Chicco, A. Estebsari, T. Huang, E. Pons *et al.*, "A european platform for distributed real time modelling & simulation of emerging electricity systems," 2016.

[3] M. Stevic, S. Vogel, A. Monti, and S. D'Arco, "Feasibility of geographically distributed real-time simulation of hvdc system interconnected with ac networks," in *2015 IEEE Eindhoven PowerTech*, June 2015, pp. 1–5.

[4] (Accessed 2017-09-27) Decentralized network 42. [Online]. Available: http://www.dn42.net

[5] (Accessed 2017-09-27) Wireless Battle Mesh. [Online]. Available: http://battlemesh.org

[6] E. Bompard, A. Monti, A. Tenconi, A. Estebsari, T. Huang, E. Pons, M. Stevic, S. Vaschetto, and S. Vogel, "A multi-site real-time co-simulation platform for the testing of control strategies of distributed storage and v2g in distribution networks," in *2016 18th European Conference on Power Electronics and Applications (EPE'16 ECCE Europe)*, Sept 2016, pp. 1–9.

[7] M. Stevic, S. Vogel, M. Grigull, A. Monti, A. Estebsari, E. Pons, T. Huang, and E. Bompard, "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] FEIN Aachen e. V. (Accessed 2017-09-27) VILLAS Framework. [Online]. Available: http://www.fein-aachen.org/projects/villas-framework

[9] S. Hemminger and others, "Network emulation with NetEm," in *Linux conf au*, 2005, pp. 18–23. [Online]. Available: https://www.rationali.st/blog/files/20151126-jittertrap/netem-shemminger.pdf

[10] (Accessed 2017-09-27) netem. [Online]. Available: https://wiki.linuxfoundation.org/networking/netem

[11] M. Mirz, A. Estebsari, F. Arrigo, E. Bompard, and A. Monti, "Dynamic phasors to enable distributed real-time simulation," in *2017 6th International Conference on Clean Electrical Power (ICCEP)*, June 2017, pp. 139–144.

[12] L. Wang, J. Jatskevich, V. Dinavahi, H. W. Dommel, J. A. Martinez, K. Strunz, M. Rioual, G. W. Chang, and R. Iravani, "Methods of interfacing rotating machine models in transient simulation programs," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 891–903, April 2010.

[13] W. Ren, M. Steurer, and T. L. Baldwin, "Improve the stability and the accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms," *IEEE Transactions on Industry Applications*, vol. 44, no. 4, pp. 1286–1294, July 2008.

[14] (Accessed 2017-09-27) RESERVE. [Online]. Available: http://www.re-serve.eu/