An Experimental Setup for Testing Synchrophasor-based Damping Control Systems

Eldrich Rebello
KTH Royal Institute of Technology
Stockholm, Sweden
Email: rebello@kth.se

Md. Shoaib Almas
KTH Royal Institute of Technology
Stockholm, Sweden
Email: msalmas@kth.se

Luigi Vanfretti
KTH, Stockholm
Statnett SF, Oslo, Norway
Email: luigi.vanfretti@statnett.no

***Abstract*—Prototype Wide-Area Power system Oscillation Damping (WAPOD) controllers using synchrophasor data have been proposed, developed and tested in field demonstration projects and are showing promise. The development process of these control systems has, however been confined to a few specific cases, the details of which have not been thoroughly reported in available literature. This paper details the development, construction and implementation of a real-time, Hardware-in-the-loop (HIL) test set-up for such a controller. A general purpose, phasor-based control algorithm is implemented on a Compact Reconfigurable Input/Output (cRIO) controller from National Instruments. The complete, closed-loop experimental set-up, the hardware used and the rationale behind different design choices are also documented. Because this test-set-up uses IEEE C37.118 data over a TCP/IP network, the complete, real-time data path is examined. An HIL test with the implemented controller is also presented and results are examined.**

I. Introduction

A. Motivation

Instances such as the Northeast blackout of August 2003 and the August 1996 Western North America (WECC) [1] blackout have been significant events on large, interconnected power systems. Both can be attributed, in part, to low frequency, electromechanically induced, inter-area oscillations [1]. These oscillations involve the generators of one synchronous area oscillating against the generators of another area and are typically between 0.1-2Hz in frequency. The fact that these modes are poorly damped [2] presents a danger to power systems with interconnections used for purposes such as power trading. Inter-area oscillations are also a threat to the stability of grids with significant generation coming from renewable energy sources.

B. Previous Experiences

The phenomenon of intra-area oscillations is well documented and has modern solutions such as Power System Stabilisers (PSS) exist and are in use. A PSS uses locally available signals and might not be very effective at damping inter-area modes with poor local observability [4] [3]. Wide-area control systems as implemented in Norway [2] and China [5] have extended the control system of an existing device to receive and use synchrophasor (IEEE C37.118) data.

C. Contributions

The goal of this paper is to document the details of the design and construction of a real-time test set-up for a wide-area control system. Ängquist and Gama's [6] Phasor Power Oscillation (Phasor POD) algorithm was implemented on a Compact Reconfigurable Input-Output controller (cRIO) [12] from National Instruments which is then tested with a inputs from a two-area model [10] running in real-time. The entire physical test set-up is examined together with constraints of the present implementation. The results from one HIL experiment are also presented. This setup can serve as a starting point for future HIL tests that involve real-time, synchrophasor-based controllers.

Note

The term 'real-time' as used in this work is identical to the sense as used in the field of embedded control [8]. Though the power network used here is a simulation, it is run in real-time so any external controller based on inputs from this system would behave in an identical manner when the same inputs are sourced from an actual power network. The clock on both the real-time simulator and on the cRIO controller run as fast as an actual clock. The controller designed is thus able to provide feedback control to the power network so as to affect the network at that point in time. Real-time is used to mean that control output generated on the controller is guaranteed to be produced in a fixed time frame. The timing limitations imposed are strict and if any delays occur, the controller is deemed to have failed.

D. Paper Outline

This paper is organised as follows. Section II outlines the damping algorithm selected, the controller architecture and introduces the two-area model used. Section III presents details of the hardware used in the construction of the HIL test. Section IV presents the design choices behind the selection of the hardware used. The HIL test and one result from it is presented in Section V and conclusions are drawn in Section VI.

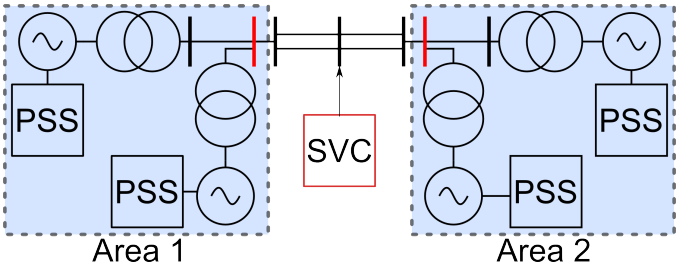

Fig. 1. Two-area model including SVC

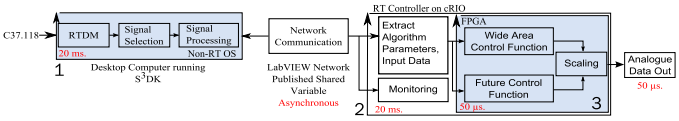

Fig. 2. Three layer controller architecture as implemented

II. Background

A. Two-Area Model & Phasor POD

The Phasor-POD algorithm essentially separates an input signal into average-valued and oscillating components [6]. The oscillating component, when suitably phase-shifted can act as a damping input to a controllable device. This algorithm [6] was selected due to its wide applicability and because it does not depend on network topology which can change in several situations. Typical model-linearisation based damping algorithms rely on computer-intensive calculations and are often valid for a particular network operating point and are hence not suitable for real-time implementation. In contrast, the Phasor-POD algorithm only requires knowledge of the inter-area oscillation frequency, which is generally known from system studies or can be determined from synchrophasor measurements [7]. This algorithm has been demonstrated to work with a two-area model (see Figure 1) where it acts as a modulating input to a flexible AC transmission system (FACTS) device [9]. The SIMULINK implementation of the Phasor-POD algorithm as presented in [9] was re-written in LabView and deployed on a cRIO9081.

B. Controller Architecture

Figure 2 details the three-layer controller architecture as implemented. The two-area model runs in real-time on the OPAL-RT simulator. Phasor Measurement Units (PMUs) are connected to the analogue outputs of the real-time simulator. The synchrophasor data stream that these PMUs generate is parsed on a desktop computer (Label 1 in Figure 2). This data is then sent over a TCP/IP network to the cRIO running the control algorithm. The Phasor-POD algorithm is implemented on the cRIO's FPGA which generates an analogue damping signal.

C. SmarTS Lab

All the hardware and software detailed in this work is available in the SmarTS Lab at KTH, Stockholm. For full details of the lab, all its equipment and capabilities, the reader is referred to [16]. This work only presents the hardware that form part of the HIL test described here and only touches upon their capabilities.

III. Hardware & Test Setup

Figure 4 presents the data path and a simplified outline of the entire HIL test conducted. This setup was arrived at after several changes to the design necessitated by limitations of the hardware and software. These are covered in the next section.

A. OPAL-RT Simulator

The eMEGASIM [15] real-time simulator from OPAL-RT [11] was the core of the HIL test. The simulator ran the two-area SIMULINK model in real-time and allowed for interfacing hardware with the model through its analogue input and output terminals. Details of these terminals are below.

- **Analogue Outputs** : Number: 32 (+/-16V and +/-10mA)
- **Analogue Inputs** : Number: 128 (+/-100V and +/-10mA)

The simulator would calculate and update values for all variables in the two-area model every 50μs. This time step was selected to allow for the assumption of linearity of power system parameters. Updated values would be written to the analogue outputs and input values read at the analogue inputs every 50μs. The SIMULINK model was developed and edited on a workstation computer and the same computer was used to monitor output from the simulator while the model was running.

B. cRIO Real-Time Controllers

Fig. 3. cRIO9081 showing NI9225, three-channel, analogue voltage input module and seven empty add-on module slots

Two different cRIO models were used in this particular HIL test. One model, the cRIO9081, was used to run the Phasor-POD algorithm in real-time. Two cRIO9076s were deployed as PMUs. Figure 3 shows the cRIO9081 with an analogue

voltage input add-on module connected. The data generated by the Phasor-POD algorithm was sent as an analogue signal back to the real-time simulator.

PMUs: Two cRIO9076s were used as PMUs, each with a four-channel, analogue voltage input module and a three-channel, analogue current input module. The inputs to the PMUs were three-phase currents and voltages and a GPS signal for time-sync. PMU software from National Instruments was run on both. The synchrophasor stream from each PMU was sent to a Phasor Data Concentrator (PDC). Each generated a C37.118 synchrophasor data stream which was sent over a TCP/IP network. The reporting rate of the PMUs was 20ms meaning that new measurements were available every 20ms. Though it was possible to generate synchrophasor data from the real-time simulator, the option of using analogue signals was chosen as it mimicked a real-world scenario. The module ratings were 0-300V and 0-5A with 24 bit resolution each [12] for the NI9225 and NI9227 respectively. To reduce subsequent computation, the PMUs were configured to compute and report values of active power.

Real-Time Phasor-POD Algorithm: The FPGA of a cRIO9081 was used to run the Phasor-POD algorithm in real-time. Though the cRIO9081 has a real-time controller in addition to the FPGA, the latter was chosen to run the Phasor-POD algorithm. This was because the FPGA runs at 400Mhz and is thus capable of a deterministic response time in the order of nanoseconds. The loop rate chosen was 50μs so that the data output rate of the FPGA was identical to the read rate on the real-time simulator's input.

C. Analogue Signal Amplifiers

The SMRT1 Single Phase Relay Tester [17] from Megger was used as an analogue signal amplifier. Each individual amplifier had a single current and single voltage input. To drive the inputs of two PMUs, six amplifier units were required. The inputs to the amplifiers were the low-level analogue signals extracted from the real-time simulator. The outputs of the amplifiers were wired to the analogue input modules of the PMUs. Table I lists the ratios used. To avoid saturation in the amplifiers, the outputs from the real-time simulator were limited to $\pm$10V.

TABLE I
AMPLIFIER INPUTS AND OUTPUTS

	Input, from simulator	**Amplified Output**
Voltage	$\pm 10V$	$\pm 100V$
Current	$\pm 20mA$	$\pm 1A$

D. Phasor Data Concentrator

The Phasor Data Concentrator (PDC) consisted of a network connected desktop computer running PDC software from SEL. The PDC was used to allow for data from multiple synchrophasor streams to be manipulated and used. It also had data logging functionality which was used to analyse the system performance.

IV. HARDWARE DESIGN CHOICES

Figure 4 shows the final HIL test setup which was used to evaluate the performance of the real-time Phasor-POD algorithm. This design was arrived at after numerous changes to the setup. This section outlines the most significant changes made together with their reasons.

A. Analogue Signal Amplifiers

The initial setup envisaged connecting the analogue outputs of the real-time simulator directly to the inputs of the PMUs. This approach simplified wiring but had to be abandoned due to a poor signal-to-noise ratio. The main reason for this poor signal-to-noise ratio was the fact that a very small portion of the dynamic range of the PMU's input modules was being used. A 0-10V signal was being read by a voltage module rated for 0-300V and a 0-20mA signal was being read by a module rated for 0-5A. This resulted in the POD algorithm producing a large error signal even at steady state. Using an amplified signal as a PMU input greatly improved the signal-to-noise ratio and thus the performance of the Phasor-POD algorithm. Though the amplifiers greatly improved the signal-to-noise ratio, additional scaling factors were introduced in the simulation and in the PMUs. For instance, the analogue output of the real-time simulator had to be limited in magnitude to avoid problems such as amplifier saturation. Also, CT and PT ratios in the PMUs were used to account for scaling factors in other sections of the HIL test.

B. FPGA

The Phasor-POD algorithm was implemented on the FPGA of the cRIO9081. This was despite the challenges of writing FPGA code and despite the fact that the algorithm could have been run on the real-time section of the controller. Two reasons were behind this choice. One was that the real-time section of the cRIO9081 handles asynchronous TCP/IP communication. This had to be incorporated in to a deterministic control loop which would also run the control algorithm. Any delays in the network would result in a delay in output from the control loop which would affect performance. The second problem was that the fastest loop rate that the real-time controller was capable of was 1ms. This was achievable only in ideal conditions. The real-time simulator expected input from the controller every 50μs, a rate that the real-time controller could not match. By leaving the task of network communication to the real-time controller, the FPGA was used to run the control algorithm together with a sample-and-hold algorithm and produce output every 50μs.

C. Desktop Computer

An ideal controller would be able to parse the synchrophasor stream, extract measurement data and perform control action autonomously. However, no software was available that could run on the cRIO and parse the synchrophasor stream. Software [18] was, however, available that could perform this function on a desktop computer. The solution implemented here thus

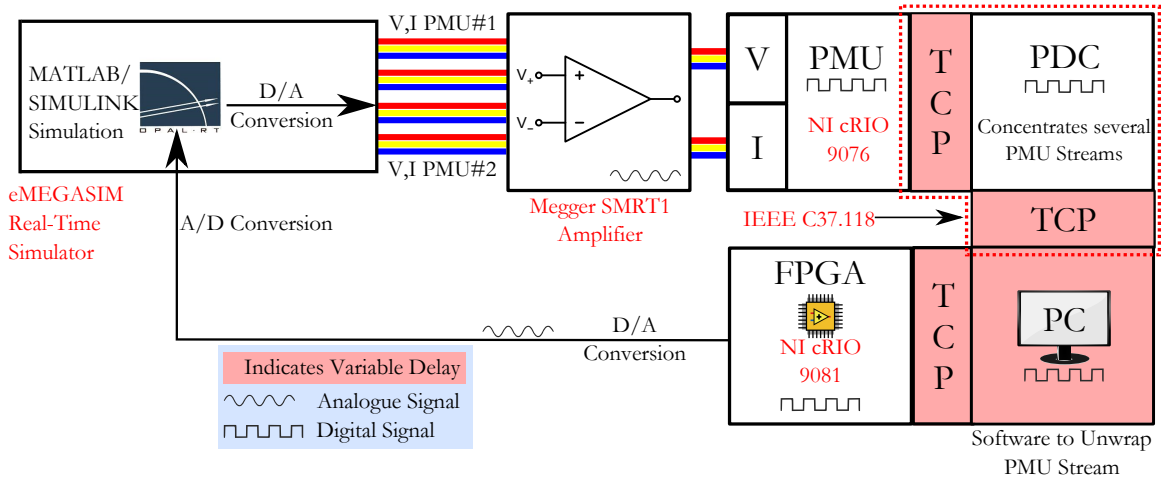

Fig. 4. Data path and experiment outling of HIL test. Hardware used is indicated in red text. Note that two PMUs are use though one is indicated. Section where synchrophasor data is used is indicated with dotted outline.

has a desktop computer running a multi-tasking operating system to extract data from the synchrophasor stream. This data is then sent to the cRIO over the TCP network. In the scenario where data from multiple PMUs is used, computations such as the voltage phase angle difference are performed on this computer.

V. Hardware-in-the-Loop Test

A setup identical to that outlined in Figure 4 was constructed and used to verify the working of the hardware WAPOD. As opposed to a simulation, a HIL test has sources of delay and noise. Sources of stochastic delay are indicated in red in Figure 4. Signals are extracted from the edges of Area 1 and Area 2 as indicated in Figure 1 and are sent to two PMUs via analogue amplifiers. The PMUs generate a sychophasor data stream which is sent to a PDC over a TCP network. The PDC time aligns both streams and retransmits them over the same TCP network. The streams are received on a desktop computer running LabView. Here, measurement data is extracted from the streams and is sent to the control algorithm running on the cRIO. The cRIO recieves this data and passes it to the built-in FPGA which runs the Phasor-POD algorithm. The FPGA uses an analogue voltage output module to generate a control signal which is wired back to the real-time simulator. After scaling, this signal is reintroduced in the Simulink model running in real-time.

Results from the real-time HIL test are presented in Figure 5. The Phasor-POD algorithm running on the cRIO uses active power (outflow from Area 1) as a control input. The inter-area mode is excited at t=2s by momentarily changing the voltage reference of one generator in Area 1 before returning it to normal. As evident in Figure 5, the Phasor-POD algorithm on the cRIO is able to restore the system to stability. As a comparison, the response of the system to the same disturbance but damped by a combination of the generator PSS and the POD (running in Simulink) is shown with the black plot. It is evident that the hardware implementation

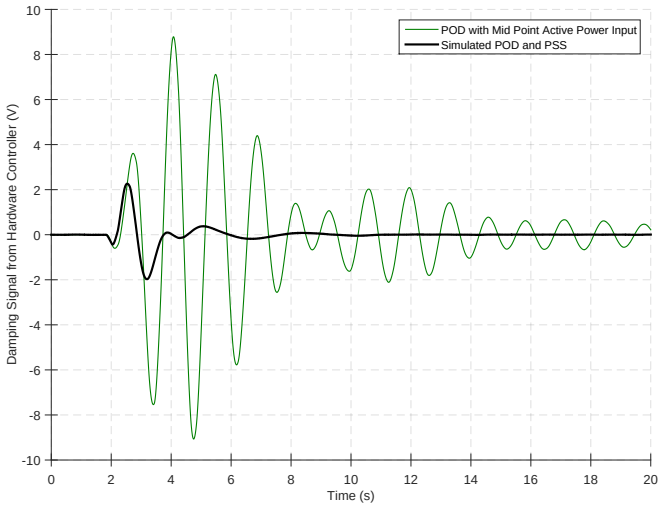

Fig. 5. Output (in V) from cRIO running Phasor-POD algorithm, shown in green. Only SVC provides damping action in this case. Black plot indicates ideal response with damping from SVC at mid-point and PSS's at each mahine

of the Phasor-POD algorithm is able to damp the inter-area mode and restore the system to stability. It is important to note that the black plot represents system performance with a PSS at each machine in addition to the SCV at the mid-point of the two-area network. In comparison, the green plot represents system performance with the SVC acting as the only damping element. Figure 5 is not meant to serve as a comparison but rather, confirmation, that real-time damping can be achieved with commercially available, general purpose controllers.

The experimental setup detailed here was designed for testing one particuar algorithm. The setup, however, is modular enough that sections of it can be replaced or removed entirely. The controller used here is based on proprietary hardware

and software from National Instruments. The support, documentation and software from National Instruments greatly accelerated the development and testing of this prototype. The authors, however, would like to see similar algorithms run on open hardware platforms such as the Raspberry pi or Arduino.

VI. Conclusion

An outline of a real-time HIL test setup used to test the Phasor-POD algorithm is presented. The hardware used in the construction of this test together with the design decisions that influenced these choices are presented. The results from one HIL experiment are presented and serve to verify the working of the real-time, hardware implementation of the Phasor-POD algorithm.

Acknowledgment

The work of E. Rebello & M.S. Almas was supported by the STRONg^2rid project, funded by Nordic Energy Research. L. Vanfretti is supported by the STRONg^2rid project, funded by Nordic Energy Research, and by the STandUP for Energy collaboration initiative and by Statnett SF, the Norwegian transmission system operator.

References

[1] North American Electric Reliability Council *Review of selected 1996 Electric System Disturbances in North America,* August 2002. Available Online: http://www.nerc.com/pa/rrm/ea/System\%20Disturbance\%20Reports20DL/1996SystemDisturbance.pdf

[2] K. Uhlen, L. Vanfretti, M.M. De Oliveira, A.B. Leirbukt,V. H. Aarstrand and J. O. Gjerde *Wide-Area Power Oscillation Damper implementation and testing in the Norwegian transmission network* in Power and Energy Society General Meeting, 2012 IEEE pp. 1-7

[3] M. E. Aboul-Ela, A. A. Sallam, J. D. McCalley and A. A. Fouad, *Damping Controller Design for Power System Oscillations Using Global Signals*, IEEE trans. on Power Systems, Vol. 11, No. 2, May 1996, pp. 767-773

[4] L. Vanfretti, Y. Chompoobutrgool, and J.H. Chow, *Chapter 10: Inter-Area Mode Analysis for Large Power Systems using Synchrophasor Data*, Book Chapter, in Coherency and Model Reduction of Large Power Systems, Joe H. Chow (Ed.), Springer, 2013.

[5] Li Peng and Wu Xiaochen and Lu Chao and Shi Jinghai and Hu Jiong and He Jingbo and Zhao Yong and Aidong Xu *Implementation of CSG's Wide-Area Damping Control System: Overview and experience* in Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES pp. 1-9

[6] L. Ängquist and C. Gama *Damping Algorithm based on Phasor Estimation* in Power Engineering Society Winter Meeting, 2001. IEEE, Volume 3, pp. 1160 - 1165

[7] M.Crow, M. Gibbard, A. Messina, J. Pierre, J. Sanchez-Gasca, D. Trudnowski, D. Vowles *Identification of Electromechanical Modes in Power Systems* IEEE Task Force Report, Special Publication TP462, June2012.

[8] Ben-Ari, M., "Principles of Concurrent and Distributed Programming", Prentice Hall, 1990. ISBN 0-13-711821-X. Ch16, Page 164

[9] M. Shoaib Almas and L. Vanfretti, *Implementation of Conventional PSS and Phasor Based POD for Power Stabilizing Controls for Real-Time Simulation*, IEEE IES IECON14, 29 Oct-1 Nov, 2014, Dallas, USA.

[10] M. Klein, J. G. Rogers and P. Kundur *A fundamental study of inter-area oscillations in Power Systems* IEEE Trans, PWRS, no. 6, pp. 914-921, 1991.

[11] eMEGAsim PowerGrid Real-Time Digital Hardware in the Loop Simulator Opal RT, [Online]. Available: http://www.opal-rt.com/

[12] *Operating Instructions and Specifications Compact RIO NI cRIO-9075/9076 & NI cRIO-9081/9082*, National Instruments, Available Online at http://www.ni.com/

[13] M. Paolone, A. Borghetti, C. A. Nucci, *A Synchrophasor Estimation Algorithm for the Monitoring of Active Distribution Networks in Steady State and Transient Conditions*, Proc. of the 17th Power Systems Computation Conference (PSCC 2011), Stockholm, Sweden, Aug. 22-26, 2011

[14] Kamwa I. (Hydro-Quebec) "Performance of Three PSS for Interarea Oscillations" Available Online at : http://www.mathworks.se/help/physmod/sps/examples_v2/performance-of-three-pss-for-interarea-oscillations.html

[15] *eMEGASIM Power Grid Real-Time Digital HArdware in the Loop Simulator* Available Online at : http://www.opal-rt.com/

[16] M.S. Almas, M. Baudette, L. Vanfretti, S. Lovlund and J.O. Gjerde "*Synchrophasor network, laboratory and software applications developed in the STRONg^2rid project*," PES General Meeting, Conference Exposition, July 2014 IEEE pp 1 - 5

[17] SMRT1 Single Phase Relay Test System Data Sheet, Available Online http://www.megger.com/common/documents/SMRT1_DS_en_V10.pdf Retreived 22.12.2014, 2014 © Megger

[18] L. Vanfretti, V.H. Aarstrand, M. Shoaib Almas, V. S. Perić and J.O Gjerde *A Software Development Toolkit for Real-Time Synchrophasor Applications*, IEEE PES Grenoble PowerTech, 2013