Real-Time Implementation of Synchrophasor-based Wide-Area Power Oscillation Damping Control System

Eldrich Rebello, *Non-Member, IEEE,* Muhammad Shoaib Almas, *Fellow, OSA,* and Luigi Vanfretti, *Senior Member, IEEE*

***Abstract*—The modern power grid is increasingly being used under operating conditions of increasing stress for which it was not designed. The increasing penetration levels of variable energy sources such as wind present significant power grid stability issues. One of the these stability issues is the phenomenon of low frequency, electro-mechanically induced, inter-area oscillations. Simulations have demonstrated the potential of Wide Area Measurement Signals (WAMS) -based Power Oscillation Damping (POD) in achieving improved electromechanical mode damping compared to traditional, local signal based, Power System Stabilizers (PSS). This paper takes an established Phasor-based oscillation damping method and combines it with modern PMU technology to implement a hardware prototype of a real-time oscillation damping control system using remote PMU signals sent over a communications network. The developed prototype is tested in a real-time Hardware-in-the-loop approach in conjunction with the Klein-Rogers-Kundur two-area four-machine test system.**

***Keywords*—IEEEtran, journal, synchrophasor, PMU, damping control, Wide Area measurement and control**

I. Introduction

THE goal of this paper is to demonstrate both the potential and flexibility in oscillation damping controller design that is provided by using synchrophasors. The Phasor Power Oscillation Damping algorithm originally developed by Ängquist and Gama [1] is run on a National Instruments Compact Reconfigurable Input / Output (cRIO) real-time controller. A slightly modified, SIMULINK model of the four-machine, two-area network developed by Klein-Rogers and Kundur [2] is run on the eMEGASIM [3] platform from OPAL RT. This allows interfacing an externally generated control signal with the simulated model. The flexibility of the developed controller is also demonstrated by extracting various data from the synchrpohasor input and using each as a damping input to the controller. A brief analysis of the performance of each input is also presented.

A. Background

As modern power systems grow in size, both in terms of power transfer capacity and geographic spread, they are increasingly being used for purposes that the power system was not designed for. Examples of these 'new' uses include conditions of increasing stress such as power trading between countries. These interconnections, which link synchronous generators, often separated by vast physical distances, create conditions where small disturbances can excite oscillations that may or may not settle. When the generators of one area oscillate at a low frequency (typically 0.2–2.5Hz) against the generators of another interconnected, but distinct area, 'inter-area' oscillations may start.

Although the purpose of system interconnection was to increase stability, the present situation of the power system incorporates renewable energy sources and power trading corridors, both of which impact system stability. More modern solutions to the problems of inter area and intra oscillations use Power System Stabilizers (PSS) [4]. While a PSS provides excellent damping to intra area modes with good local observability, its performance with intra-area modes may not be satisfactory [5].

1) Outline: The Wide Area Power Oscillation Damper (WAPOD)[1] prototype developed here uses commercially available micro-controller hardware and is based entirely on PMU measurements received over a TCP/IP network. The Phasor-POD algorithm [1] will be implemented on a general purpose micro-controller and will be run in real-time. The inputs to the controller will come from one or multiple PMU's, each monitoring data at different points in the power system. The power system model used in this paper is the two-area four-machine model, originally proposed by Klein, Rogers and Kundur [2]. To prove the real-world applicability of the developed controller, all tests are carried out in real-time, with conditions such as noise and network transport delay present.

2) Hardware Outline: All real-time simulations here are performed on OPAL RT's eMEGASIM [7] real-time simulation

Eldrich Rebello is with the School of Electrical Engineering, Aalto University, Helsinki, Finland e-mail: eldrich.rebello@aalto.fi

M. Shoaib Almas and L. Vanfretti are with KTH Royal Institute of Technology, Stockholm, Sweden.

This project was supported by the Nordic Energy Research through the STRONg^2rid project.

[1] Historically, damping stabilizers have been termed WAPOD where the P represents a measurement of active power through the line. Active power here would be used as a controller input signal. Although this term is not accurate when other quantities are used as control inputs or feedback signals, the term is used here to maintain consistency with existing literature.

platform. SIMULINK models are executed in real-time and are interfaced with externally generated signals. Current and voltage signals from different points on the simulated network are extracted, amplified and then supplied as the input to two PMU's. Two cRIO-9076 [11] devices are used as PMU's. The synchrophasor data stream from these devices is streamed over a TCP/IP network to a Phasor Data Concentrator which produces a time-aligned output stream. This stream is then accessed, via a TCP/IP network, on a PC running LabVIEW which extracts raw measurement data. This extracted data is streamed over the network to a cRIO-9081 which runs the Phasors-POD algorithm in real-time. The FPGA on the cRIO-9081 [10] generates a damping signal which is then wired to the real-time simulator for use in the SIMULINK model.

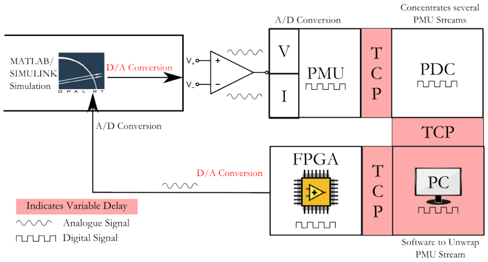

Fig. 1. Harware Outline

Vanfretti et. al. describe the details of the equipment used here in [9]. It is important to note that the data flow in Figure 1 involves both D/A and A/D conversions. Also, since no synchronisation is used for these conversions, it is important that the sample rates or loop rates of each section be integer multiples of each other. This will prevent data value errors. The issue of different loop rates is covered in Section V-C.

3) Software Outline: The two-area four-machine model used here is available in SIMULINK's SimPowerSystems and can be accessed by typing `power_PSS` as a MATLAB command. The SIMULINK implementation of the Phasor-POD algorithm was developed by Almas and Vanfretti [6]. The Phasor-POD algorithm essentially separates an input signal into an average valued and an oscillating component. The oscillating component when suitably phase-shifted can be used as a supplementary damping input to a generator's AVR or the excitation system of a FACTS device.

LabVIEW's Real Time and FPGA modules were used to write the code for the respective sections of the cRIO. Also, since no synchrophasor data-extraction software was capable of running independently on the RT controller, the process of extracting raw measurement data from the synchrophasor data stream was performed on a desktop computer. The software used for this was identical to that in [12] and would run on a workstation computer, extract data from the PMU stream and send this extracted data to a LabVIEW program running on the same computer [12].

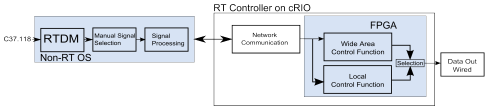

Fig. 2. RT Controller Architecture

4) Real-Time Implementation of Phasor-POD Algorithm: The hardware implementation of the POD was based on the Compact Reconfigurable Input / Output (cRIO) 9081 [10] from National Instruments. This controller is equipped with an on-board Field Programmable Gate Array (FPGA) running at 400MHz in addition to an independent real-time controller. The real-time implementation of the Phasor-POD algorithm was based on the SIMULINK implementation by Almas & Vanfretti [6] and its goal was the replicate the behaviour of the SIMULINK implementation as closely as possible. The algorithm accepts three inputs; the search frequency, ω_{cs1}, the sampling time T_s, the phase correction $alpha$ and a signal scaling factor. It takes advantage of the fact that the oscillation frequency for a given network configuration is usually known, which in this case is the 0.64Hz inter-area mode. Using this known frequency value, a co-ordinate system, rotating at this known frequency, is set up where the oscillating component is continuously extracted as a phasor [1].

A three-layer, modular code architecture was selected for implementation, as shown in Figure 2.

- Core FPGA Software : Interacts with hardware terminals for I/O and runs Phasor-POD algorithm
- Real Time (RT) Software : Manages network communication to remote terminal and also generates performance monitoring data
- Remote Interface: Runs on workstation computer: Used to update algorithm parameters and monitor data & performance. This layer is non deterministic.

The Phasor-POD algorithm could be implemented on either the real-time section of the cRIO or the FPGA but was implemented on the FPGA. This decision was made keeping in mind the computational resources and response speed needed to match the step size of the real-time simulator. The complexity of the code meant that the cRIO's real-time controller would not be able to complete an iteration of the algorithm in the required 50μs response time. The real-time section of the cRIO handles network communication. Its primary purpose is to receive measurement data that the workstation computer extracts and to stream this data to the Phasor-POD algorithm running on the FPGA. The real-time controller also handles commands coming from the user interface running on the workstation computer. It also monitors the output of the FPGA, sends data to the user interface for monitoring, periodically logs input and output data and handles error conditions.

The remote interface runs in LabVIEW on a conventional Windows® based computer. The Phasor-POD algorithm can be controlled from this interface. Also, the algorithm parameters can be set and modified. Since the operating system here is not real-time but is multi-tasking, the execution speed depends on the processor load and is not deterministic. Statnett's Synchrophasor Software Development Kit (S^3DK) was used to unwrap the PMU streams coming from the PDC and extract phasor measurements [12]. This allowed for data to be extracted and used directly in the LabVIEW environment. Since the PMU reporting rate was 50 messages a second, new data was available every 20ms. The loop rate used by the S^3DK was hence 20ms. The selection of a input for the controller and the computations required are performed here. For example, if active power is to be used as a POD input, voltage and current values must be multiplied to obtain the active power. If the voltage angle difference is to be used as an input, the required calculations are performed in this VI.

II. Setup Preparation

A. Two-Area Model Preparation

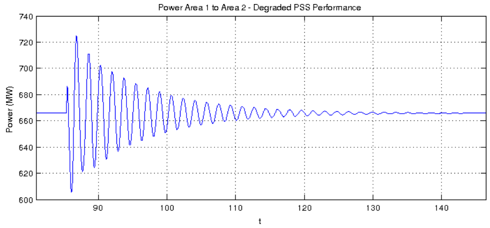

Fig. 3. Degraded performance with PSS

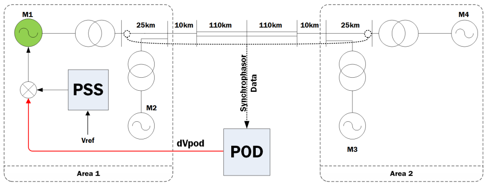

Fig. 4. Modified Two-Area Four-Machine network

The two-area network [2] is known to be unstable without external damping control. The difference between the rotational inertias of two of the machines, in an otherwise symmetric network gives rise to the 0.64Hz inter-area mode. The SIMULINK model used here implements damping control using a supplementary signal generated from a PSS and applied to the excitation system of each of the machines to achieve steady-state stability and also to damp out the inter-area mode. This model was modified to have a damping control system only at Machine M1 in Area-1 (see Figure 4). All other machines have their PSS's disconnected. This configuration was verified to be able to both achieve stability and to be able to restore the system to stability after the application (and subsequent clearing) of an 8 cycle, three-phase to ground fault. The original model uses a PSS tuned to achieve maximum damping, given the operating conditions. To demonstrate a potential application scenario for the developed POD prototype, the performance of the PSS is degraded so that it is still able to provide damping but takes substantially longer (Figure 3) . This figure shows the system response to a 200ms perturbation in the voltage reference of machine M1. Note that under the action of the sole PSS at machine M1, the inter-area mode is damped and the system is restored to stability. This mimics a real-world situation where an already installed PSS whose performance has degraded over time due to changing network conditions or poor tuning.

This network was grouped into sub-systems and prepared for simulation on in RT-LAB. The time step chosen was 50μs. Current and voltage measurements were taken from the points marked in Figure 4 and were extracted from the analogue outputs of the real-time simulator. The damping signal dVpod in Figure 4 was generated at two points; one in the real-time simulation itself using a Simulink implementation of the Phasor POD algorithm and one externally on the cRIO. Either one of these signal could be switched in for use in the simulation.

III. Testing & Results

A. Local Signals

B. Wide Area Signals

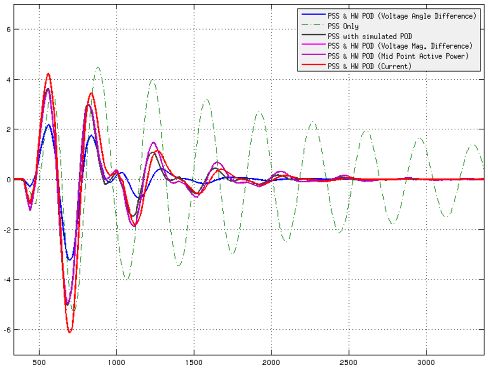

Fig. 5. Response Comparison Using Wide Area Signals

IV. Additional Inputs

V. Challenges

A. Time Delays

The Phasor POD algorithm running in SIMULINK has close to zero time delay between network behaviour changing and the controller responding. The same algorithm, when run on the cRIO, receives data with a stochastic delay. Figure 1 illustrates the complete data path and from this, it is evident that several sources of time delay exist. Sections such as the analogue amplifiers, the FPGA execution speed (a constant 50μs), the real-time section of the cRIO and the PDC all represent fixed, non-zero time delays. Sections such as the

D/A and A/D conversion in the real-time simulator also add a deterministic time delay. However, sections of the data path such as communication over a TCP/IP network and the PC used to extract raw measurement data from the synchrophasor stream all represent variable delays. These delays all contribute to a delayed response from the cRIO. This delay allows for a network disturbance to grow slightly before the cRIO starts responding. It also has the effect of changing the phase compensation required in the Phasor-POD algorithm. Though the delay cannot be compensated for, the phase compensation needed can be changed depending on the measured delay. Presently, the phase compensation required is determined iteratively though this can be automated using time-stamped data.

B. Analogue Limits and Noise

The original POD (Phasor Oscillation Damper) algorithm [1] was developed and simulated in an ideal, noise-free environment with zero delay. More importantly, no limits are imposed on the magnitude of either the controller's inputs or outputs when it is simulated. In contrast, a hardware-based implementation that uses analogue signals faces several challenges, one of which is the analogue signal magnitude limits.

Consider the analogue outputs of the OPAL RT simulator listed on Page **??**. These are low level outputs and can be directly connected to the PMU inputs. However, the inputs modules of the PMU's are rated for 0-300V and a 0-16V analogue signal will not use a significant portion of this range. Additionally, a signal of such a small magnitude will also be contaminated by noise and will consequently have a poor Signal to Noise ratio. A similar argument can be made for the current outputs.

The output of the simulated POD can vary over several orders of magnitude, ranging from 10^5 at times of peak damping to as small as 10^{-3} once the oscillation magnitude has become small. It is not possible to recreate analogue signals with such vast ranges. The voltage output module used with the cRIO here had a 24-bit resolution and was limited to 10V in magnitude. Any values generated by the POD algorithm that were greater in magnitude that 10V would cause output saturation. All these issues meant that signal magnitudes had to be amplified in certain cases to use the full measurement ranges or had to be limited in other cases, so as to capture variations without saturation.

C. Loop Rates and FPGA Resources

The most significant challenge in the real-time implementation of the Phasor-POD algorithm was the fact that different sections of the hardware control loop run at different loop rates or step sizes. Added to this was the fact that the real-time simulator generated data every 50μs and thus expected data from the HIL setup at the same rate while the rest of the HIL setup did not support such a high data rate. The PMU's used supported a maximum data reporting rate of 50 samples/second. Table I lists the different components of the HIL setup together with their respective loop rates.

TABLE I. COMPARISON OF LOOP RATES OF DIFFERENT COMPONENTS

Element	Loop Rate	Mode
OPAL RT Simulator	50μs	Real Time
PMU (cRIO)	20msec	Real Time
Workstation Computer	20msec	Not Real Time
POD - RT Section	20msec	Real Time
POD - FPGA	50μs	Real Time

50μs was chosen for the FPGA loop rate in order to match the loop rate of the real-time simulator. This was to ensure that data was always available at the analogue input of the simulator and no erroneous data points were read. The 50μs loop rate of the FPGA meant that new data was expected every 50μs, corresponding to a 2000s/s sampling rate. The fastest execution speed of the real-time section of the cRIO was 1ms, significantly slower than the FPGA's speed. New synchrophasor data was available from the PMU's only every 20ms. One solution to this problem was to upsample the synchrophasor data, interpolate between consecutive data points and then stream this data to the Phasor-POD algorithm on the FPGA. A FIFO[2] buffer would be used to buffer the data generated by the up-sampling process as it was gradually consumed by the FPGA. The major problem with this method was that the up-sampling process on the RT controller is computationally intensive and would not run at the required 20ms loop rate.

An alternative solution would be to implement a **sample and hold** algorithm on the FPGA. As data was extracted every 20ms, the RT controller would receive this data and send it to the FPGA. This value would then be held constant till the next data point arrives. This was implemented and found to work satisfactorily. In essence, the only difference between the Phasor-POD algorithm running the SIMULINK simulation and that on the cRIO was the data rates of their input data. The simulated POD received new data every 50μs while the cRIO-based POD received data every 20ms.

VI. CONCLUSION

This work has demonstrated the feasibility of using wide-area power system data in the design of an oscillation damping controller. The real-time simulation results prove that the design goals of the controller were met. The flexibility of synchrophasor data was demonstrated and the wide range of inputs possible from this were also tested.

APPENDIX A
PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

[2] First In First Out

APPENDIX B

Appendix two text goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

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

[2] 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.

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

[4] F. P. Dmello, and. C. Concordia *Concepts of Synchronous Machine Stability as Affected by Excitation Control* IEEE TRANSACTIONS ON POWER APPARATUS AND SYSTEMS, vol. PAS 88, no. 4, 1969.

[5] 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

[6] 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.

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

[8] N.S. Chaudhuri, R. Majumder and B Chaudhuri *Interaction between conventional and adaptive phasor power oscillation damping controllers* in Power and Energy Society General Meeting, 2010 IEEE Minneapolis, MN, 2012 pp. 1-7

[9] L. Vanfretti.; M. Chenine; M.S. Almas; R. Leelaruji; L. Angquist; L. Nordstrom, "*SmarTS Lab A laboratory for developing applications for WAMPAC Systems*," Power and Energy Society General Meeting, 2012 IEEE , vol., no., pp.1,8, 22-26 July 2012

[10] *Operating Instructions and Specifications Compact RIO NI cRIO-9081/9082*, National Instruments, Available Online at http://www.ni.com/pdf/manuals/375714a.pdf

[11] *Operating Instructions and Specifications Compact RIO NI cRIO-9075/9076*, National Instruments, Available Online at http://www.ni.com/pdf/manuals/375650b.pdf

[12] Vanfretti, Luigi and Aarstrand, Vemund H. and Almas, M. Shoaib and Perić, Vedran S. and Gjerde, Jan O. , *A Software Development Toolkit for Real-Time Synchrophasor Applications*, IEEE PES Grenoble PowerTech, 2013

Luigi Vanfretti Biography text here.

M S Almas Biography text here.

Eldrich Rebello Biography text here.