Date of publication xxxx 00, 2019, date of current version xxxx 00, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.DOI*

# Experimental Testing of a Real-Time Implementation of a PMU-based Wide-Area Damping Control System

**ELDRICH REBELLO[1], LUIGI VANFRETTI[2] (SENIOR MEMBER, IEEE), AND MUHAMMAD SHOAIB ALMAS [3] (Member, IEEE)**

[1]KTH Royal Institute of Technology, Stockholm, Sweden SE-100 44 (e-mail: eldrich.rebello@alumni.aalto.fi)
[2]Rensselaer Polytechnic Institute, 110 Eighth Street Troy, NY USA 12180 (e-mail: vanfrl@rpi.edu)
[3]KTH Royal Institute of Technology, Stockholm, Sweden SE-100 44 (email: msalmas@kth.se)

Corresponding author: Eldrich Rebello (e-mail: eldrich.rebello@alumni.aalto.fi).

This work was supported by the Nordic Energy Research's STRON$g^2rid$ project

**ABSTRACT** The modern power grid is increasingly being used under operating conditions of increasing stress, giving rise to grid stability issues. One of these stability issues is the phenomenon of inter-area oscillations. Simulations have demonstrated the advantages of Wide-area Measurement Signals (WAMS)-based Oscillation Damping Controls in achieving improved electromechanical mode damping compared to traditional, local signal-based Power System Stabilizers (PSS). This work takes an existing Phasor-based oscillation damping (POD) algorithm and uses it to implement a proof-of-concept, real-time controller on National Instruments hardware. The developed prototype is tested in a real-time Hardware-in-the-loop setup (RT-HIL) using OPAL-RT's eMEGASIM real-time simulation platform and synchrophasor data from actual Phasor Measurement Units (PMUs). The prototype and experiments provide insight into the feasibility and real-world limitations of wide-area controls. Further, it is demonstrated how the proposed control architecture has applications independent of the controlled device. Challenges faced, the solutions implemented together with the present prototype's limitations are also discussed.

**INDEX TERMS** Power system dynamics, power system stability, power system simulation, power generation control, phasor measurement units

## I. INTRODUCTION

ALTHOUGH the purpose of interconnecting disparate power networks 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. One of these stability issues is the phenomenon of inter- and intra-area electromechanical oscillations [1]. Inter-area oscillations involve groups of generators oscillating against each other and these groups are typically connected by weak tie-lines [1]. Inter-area oscillations are of a lower frequency ($< 1Hz$) relative to intra-area oscillations and are often poorly damped leading to stability problems [1]. Modern solutions to the problems of inter-area and intra-area oscillations use Power System Stabilizers (PSS) [2]. While a PSS provides excellent damping to intra-area modes with good local observability, its performance with inter-area modes may not be satisfactory [3]. The limited observability in a local signal can be complemented by wide-area, synchrophasor data. Wide-area data allows for calculations and observations such as voltage angle differences, which are not possible with solely local measurement data [4]. For a more complete treatment of the phenomenon of power system oscillations, see [5].

### A. LITERATURE REVIEW

#### 1) Analytical Studies

It has been shown in [4] that wide-area signals are preferable to local signals, such as active power and frequency, [3] for the purpose of damping inter-area oscillations. Studies such as [4] have analysed damping performance when using signals such as voltage angle difference, which can easily be computed using data from multiple PMUs.

#### 2) Field Tests

Successful field trials of wide-area oscillation damping controllers are reported in [2] and [6] indicating the potential that

PMU-based wide area controllers have to offer. These results are not without certain limitations. In [2] it is reported that the limited set of tests conducted with the Wide Area Power Oscillation Damper (WAPOD) are insufficient to compare the performance of a WAPOD to a local-measurement based controller. Coordinating field experiments is often difficult and requires the participation of several entities. This work attempts to approximate real-world conditions as closely as possible but avoids several of the limitations of working with an operational power system.

#### 3) Delay Studies

The damping performance of a controller that depends on TCP network-transmitted data for input is constrained by the network transport delay. Beyond a certain value of time delay, damping will cease to be effective. These effects are explored in more detail in [7] and [8].

### B. PAPER CONTRIBUTIONS

The goal of this paper is to imlement a hardware prototype and to demonstrate experimentally the potential and flexibility in oscillation damping controller design that is possible by using synchrophasor data (IEEE C37.118). The Phasor Power Oscillation Damping (Phasor-POD) algorithm originally developed by Ängquist and Gama [10] is implemented and deployed on a National Instruments Compact Reconfigurable Input / Output (cRIO) real-time controller. A modified, Simulink model of the four-machine, two-area network developed by Klein-Rogers and Kundur [1] is executed in real-time on the eMEGASIM [11] platform from OPAL-RT. A Hardware-in-the-loop (HIL) test is set up to verify the performance of the hardware implementation of the Phasor POD algorithm. The flexibility of the developed controller is also demonstrated by extracting various data from the synchrophasor data-set and using each as a damping input to the controller. This paper also illustrates that the controller can have multiple applications by testing it with two different controllable devices; a generator automatic voltage regulator (AVR) system and a Flexible AC Transmission System (FACTS)-device excitation system. A brief analysis of the performance of each synchrophasor input to the hardware POD is also presented.

In the context of this paper, the term 'real-time' is used to refer to a controller where input data is processed and made available as output within a deterministic time frame. The usage of the term is similar to that in the field of embedded systems in that strict deadlines are imposed and system reaction time is guaranteed and fixed. To this end, the real-time simulator's clock runs as fast as an actual clock and therefore generates and processes data in real-time. This requires that any external, hardware controllers connected to it operate at an identical pace. These constraints imply that the algorithm implementation in this work is tested under conditions that closely approximate a real-world situation. Constraints such as signal magnitude limits, variable network communication delays and noise apply. If the proposed hardware implementation of the POD algorithm fails to respond in a deterministic manner, it is deemed to have failed.

Problems exist with traditional damping controller design that is based around model linearisation and non-linear simulations [6]. This paper circumvents these issues by adopting a phasor-based damping algorithm [10] that can operate independent of the network configuration. In addition, beacuse this is the same algorithm used in proprietary implementations [2], the prototype developed is directly comparable to other real-world implementations. The developed controller is designed with modularity and portability in mind and is demonstrated to be able to provide effective damping in multiple application scenarios. Although [4] presents a scenario where the generator AVR's error signal is modulated using PMU data, this work presents an experimental validation of those results. While multi-mode damping can be attractive [9], the focus of this work is single-mode damping to verify and be comparble to real-world cases [2], [6] .

### C. PAPER ORGANISATION

This paper is organised as follows. Section II presents a brief, theoretical background of the work presented in this study while Section III introduces the software and hardware used. Section IV covers the preparation of the Simulink models for real-time simulation and two real-time test cases, one with the WAPOD input fed to the excitation system of a Static VAR Compensator (SVC) and the other with the WAPOD modulating the input of a generator's Automatic Voltage Regulator (AVR). The results obtained from the two tests are presented and analysed in Section V. Section VI examines some of the major challenges faced in the development, implementation and testing of this WAPOD controller. No experimental implementation is ever ready for the field and there is always room for improvement. This is outlined in Section VII and finally conclusions are drawn in Section VIII.

## II. BACKGROUND

### A. WHY THE PHASOR-BASED ALGORITHM?

Traditional controllers for FACTS devices depend on accurate system models at a specific operating condition. Large systems often experience changes in their topology and data about the present condition may not always be available to apply conventional control design methods. Further, identifying system oscillation modes and designing appropriate dampers is based on small signal and linear analysis techniques applied to power system models. These models are often difficult to derive accurately for large and interconnected power systems [12]. Linearised models are also only valid for small deviations from the linearisation point. One important reason for choosing the Phasor-POD algorithm for real-time implementation in this work was the fact that the algorithm itself uses limited inputs and is independent of network configuration and topology. The only network-dependent algorithm parameter is the oscillation frequency and this is usually known from system studies or can

be determined directly from synchrophasor measurements [13]. Compared to conventional controllers designed using linearisation-based methods, the adoption of phasor-based controllers has not been common mainly due to the fact that such controllers tend to be highly non-linear and thus difficult to, both, model in simulation studies [14] and implement in real-time applications [2].

### B. PHASOR POD ALGORITHM

Some of the problems that emerge from the model linearisation approach are addressed by phasor-based oscillation damping algorithms. The algorithm chosen for implementation here is the Phasor-POD algorithm developed by Ängquist and Gama [10]. The measured signal $s(t)$ can be represented as a space-phasor [14]:

$$s(t) = s_{avg} + \mathrm{Re}\left\{\vec{s}_{ph} \cdot e^{jwt}\right\} \tag{1}$$

where, $\vec{s}_{ph}$ is a complex phasor, rotating at the frequency $\omega$ [10]. This presents an average value and the associated oscillatory part, in a stationary reference frame. The oscillating part can then be used to generate a control signal for the FACTS (or other controllable device) using a control algorithm. This method is independent of the system state or configuration and is not computationally intensive. Controllers based on this approach may also incorporate error checking and phasor estimation algorithms. The real-time hardware implementation of the Phasor-POD algorithm was based on the Simulink implementation in [15] and its goal was to replicate the behaviour of the Simulink implementation as closely as possible. The algorithm accepts three inputs; the search frequency, $\omega_{cs1}$, the sampling time $T_s$, the phase correction $alpha$ in addition to 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 [10]. In simpler terms, the Phasor-POD algorithm 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.

## III. SOFTWARE ARCHITECTURE AND HARDWARE CONFIGURATION

The WAPOD[1] prototype developed here uses commercially available micro-controller hardware and uses PMU measurements received over a TCP/IP network. The Phasor-POD algorithm [10] was executed on the control hardware in real-time. The inputs to the controller come from one or multiple PMU's, each monitoring data at different points of 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 [1]. To prove the real-world applicability of the developed controller, all tests are carried out in real-time, with conditions such as measurement noise and communication network delay present.

### A. HARDWARE CONFIGURATION

The power system model was simulated in real-time on OPAL-RT's eMegasim [11] real-time simulation platform (Figure 1). This model was interfaced with externally generated, analogue signals in a HIL configuration. Current and voltage signals from different points (Buses 5 and 11 in Figure 4) on the simulated network are extracted as analogue signals, amplified and then supplied as inputs to two PMUs. Two cRIO-9076 [16] devices are used as PMUs, although any other commercial device could be used. Each PMU is equipped with three-phase analogue voltage and current input modules in addition to GPS time synchronisation. The PMUs report data every 20 ms. The synchrophasor data stream generated by these PMUs is sent over a TCP/IP network to a Phasor Data Concentrator (PDC) which produces a time-aligned output stream. This stream is then accessed on a PC running LabVIEW over a TCP/IP network. Data is extracted and sent to the FPGA (Field Programmable Gate Array) running the Phasor-POD algorithm. The FPGA on the cRIO-9081 [16] generates a damping signal which is then wired back to the real-time simulator for use in the Simulink model.

[15] describes the details of the equipment used here. It is important to note that the data flow in Figure 1 involves both D/A and A/D conversions. Also, because 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 prevents data sampling errors. The issue of different loop rates is discussed in Section VI-C.

### B. SOFTWARE ARCHITECTURE

As shown in Figure 2, LabVIEW's Real Time and FPGA software was used to write the code for the respective sections of the cRIO. Because no synchrophasor data-extraction software was available that could run 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 the smart grid Synchrophasor Software Development Kit ($S^3$DK) which runs on a workstation computer, extracts data from the PMU stream and sends this extracted data to a LabVIEW program running on the same computer[2]. Using the $S^3$DK, a LabVIEW application was implemented to select the PMU's from which the data was to be utilized. Once available in LabVIEW, this data was then sent to the

[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.

[2]Available open-source at https://github.com/ALSETLab/S3DK

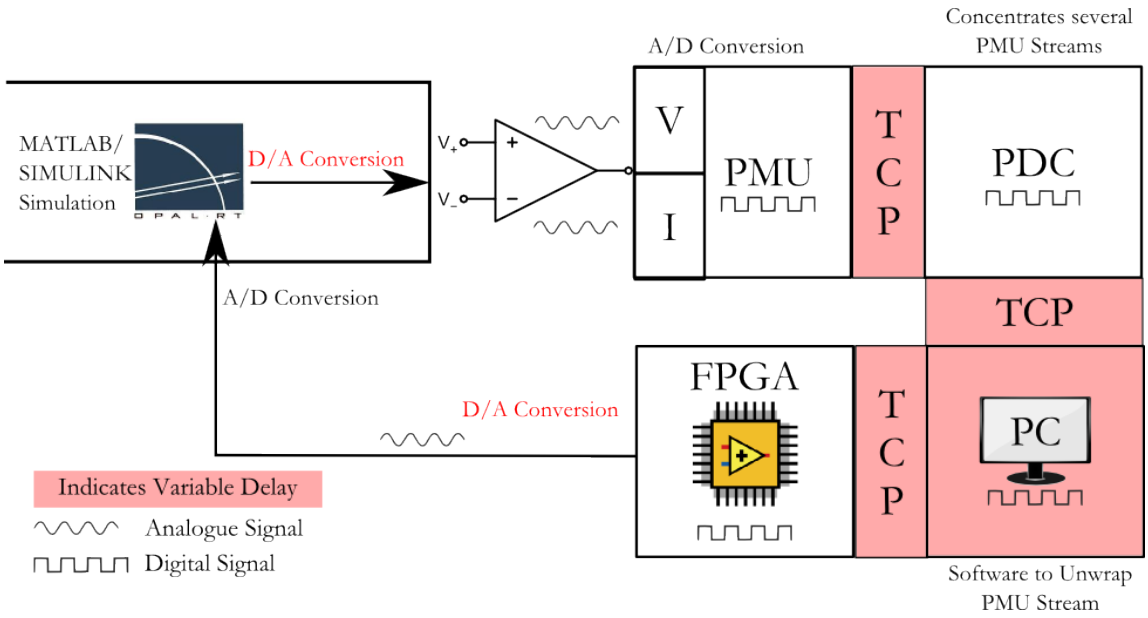


**FIGURE 1.** Hardware loop showing complete data path, signal nature at each point and sources of delay.

RT controller using LabVIEW's Shared Variables [23] over a TCP network.

### C. REAL-TIME IMPLEMENTATION OF PHASOR-POD ALGORITHM

The hardware implementation of the POD was based on the cRIO 9081 [16] from National Instruments. This controller is equipped with an on-board FPGA (maximum clock speed of 80Mhz[3]) in addition to an independent real-time controller (1.06GHz Intel Celeron U3405).

A three-layer, modular code architecture [17], following guidelines from the manufacturer [20], was selected for implementation. An outline of the architecture is shown in Figure 2 and each of the three layers are briefly described below (Corresponding to the numbers in Figure 2).

1) Remote Interface: Runs on a standard, workstation computer. Used to input algorithm parameters and monitor data & performance. Receives the aligned synchrophasor data stream and extracts measured value data. This layer is non deterministic in time.
2) Real Time (RT) Software : Manages network communication to the remote interface and also generates performance monitoring data. The remote interface interacts with this layer over the communication network. Transfers extracted synchrophasor data to the FPGA.
3) Core FPGA Software : Interacts with hardware terminals for I/O and runs the Phasor-POD algorithm. Input data comes through the RT interface running on the cRIO.

[3]http://www.xilinx.com/support/documentation/data_sheets/ds162.pdf

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 each iteration of the algorithm in the required 50$\mu$s response time. The real-time section of the cRIO only 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 computer. The Phasor-POD algorithm can be controlled and monitored from this interface. The operating system here is not real-time but multi-tasking. The execution speed therefore depends on the processor load and is not deterministic. The $S^3DK$ was used to unwrap the PMU streams coming from the PDC and extract phasor measurements. This allowed for data to be extracted and used directly in the LabVIEW environment. The PMU reporting rate was 50 messages a second and new data was available every 20ms. The loop rate used by the $S^3DK$ was therefore 20ms. The selection of an input signal for the controller and any signal processing 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.

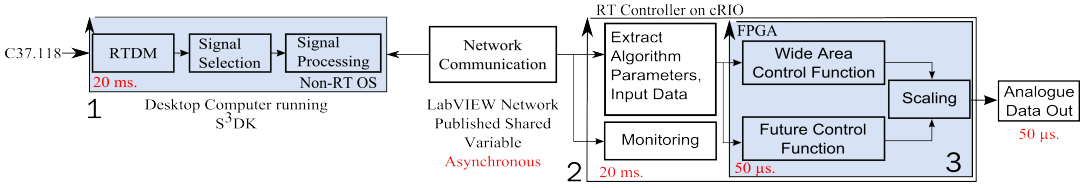


**FIGURE 2.** Three-layer Software and Hardware Architecture of the WAPOD Controller. Loop rates are indicated in red

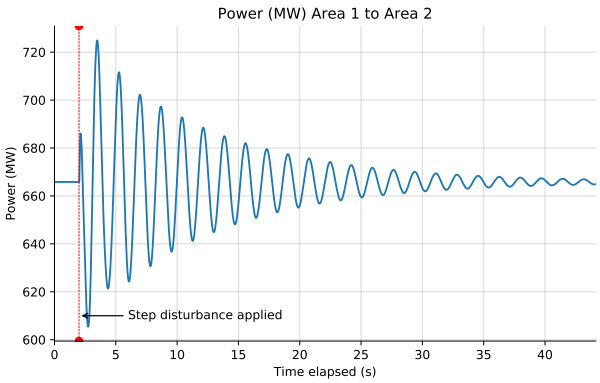


**FIGURE 3.** Damping performance of the base case PSS only at Machine M1. Local active power is used as the PSS input.

If the voltage angle difference is to be used as an input, the required calculations are performed in this LabVIEW Virtual Instrument (VI[4]).

## IV. EXPERIMENTAL SETUP PREPARATION

The nature of the Phasor-POD algorithm is generic enough to allow it to be used as a modulating input to a variety of controlled devices. Two examples are illustrated in this work, one, as a damping controller modulating a generator's AVR system and the other as a modulating input to the excitation system of a FACTS device (here, an SVC). Figure 4 illustrates both these uses along with the two-area network outline. Note that both functions are not implemented simultaneously.

### A. TWO-AREA MODEL PREPARATION - GENERATOR AVR

The original Klein-Rogers-Kundur model in [1] was modified for the studies in this work. In order to assess the performance of the WAPOD, the base case employs a damping control system (PSS) only at Machine M1 in Area-1 (see Figure 4). All other machines do not have a PSS installed. The WAPOD output is used as an additional input to machine M1's AVR system. This configuration was verified to be stable and is able to restore the system to stability after the application (and subsequent clearing) of an 8 cycle, three-phase to ground fault at bus 8 in Figure 4. To demonstrate a potential application scenario for the developed POD prototype, the performance of the PSS was set up so that it is able to provide damping but with a long response and settling time (Figure 3). This figure shows the system response to a 200 ms., 5% 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 with an already installed PSS whose performance has degraded over time due to changing network conditions or poor tuning. Using only one PSS ensures that the inter-area mode is clearly visible. Our intention with this modification is to demonstrate that a hardware implementation of the POD algorithm can improve system damping despite the controller's real-world limitations. It is not our goal to demonstrate that a hardware-based controller is able to improve system damping with a PSS installed at each machine. Though this may be possible, this work only seeks to demonstrate that a hardware implementation of a Phasor-POD based on synchrophasor data is able to function as expected.

### B. TWO-AREA MODEL PREPARATION - FACTS DEVICE

The second application in Figure 4 is to use the WAPOD output as a modulating input to an SVC's excitation system. The two-area model was prepared for simulation in the same way as in the previous case, however, in this case a PSS was included at all four machines. The SVC model implemented was an average-value model, identical to that used in [15]. The SVC was connected at the mid-point of the two area network (Bus 8 in Figure 4). As shown in [19], this is the point where voltage swings will be the greatest and also where the SVC can be most effective at damping power flow swings. For comparison purposes, two parallel and identical implementations of the Phasor-POD algorithm were used, one implemented in Simulink and the other on the cRIO. Either could be switched in at a given time. It is important to note here that when the hardware-POD was switched in, the PSS's at each of the four machines were disconnected, leaving the SVC as the sole control and damping device in the network. The ability of the SVC to keep the network stable and also to restore it to stability was verified with off-line simulations using the Phasor-POD implemented in Simulink.

### C. REAL-TIME SIMULATION

The modified network in each case was grouped into subsystems and prepared for simulation using RT-LAB [11]. The

[4]A LabVIEW program. See http://www.ni.com/white-paper/7001/en/

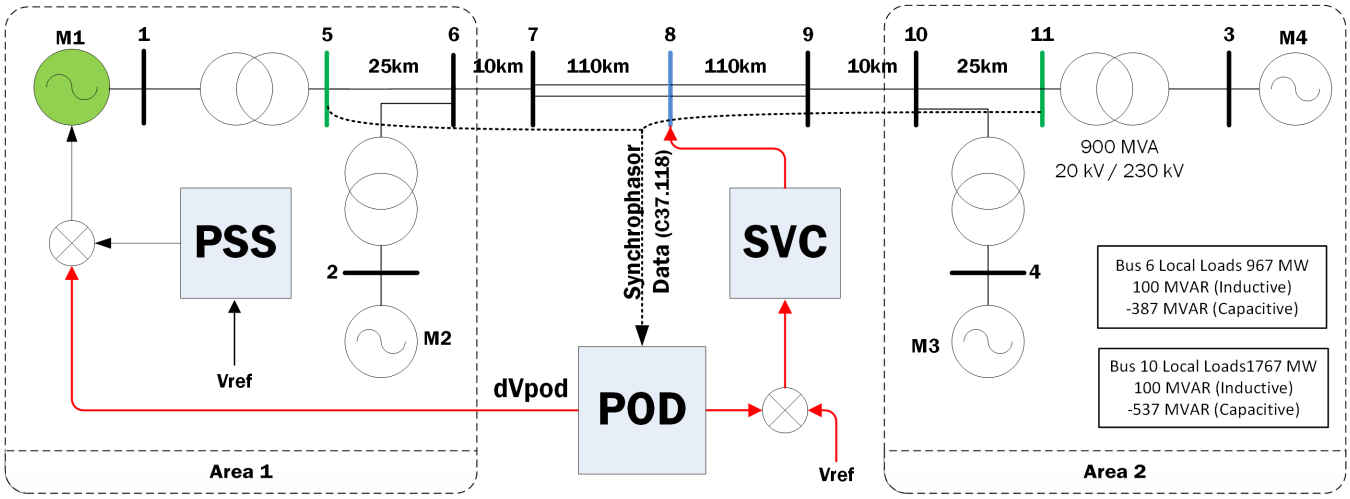


**FIGURE 4.** Modified Two-Area Four-Machine network outline showing PMU data sources and both uses of the POD damping output. Note that only one scenario is implemente

simulation time step chosen was 50$\mu$s meaning that outputs and inputs of the real-time simulator were updated every 50$\mu$s. Current and voltage measurements were taken from the buses marked (in green) in Figure 4 and were extracted via 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. Both signals were generated simultaneously and either one could be switched in for use in the simulation.

## V. TESTING & RESULTS

The POD algorithm implemented in Simulink uses locally available active power measurements as input. In the case of the Generator PSS, this is the active power measured at the terminals of the generator. In the case of the SVC, the active power at the mid-point of the interconnecting line (also the point of connection of the SVC) is used as an input. The WAPOD hardware prototype can use these same signals as inputs in each case but can also exploit other data from the synchrophasor data stream. Testing the operation of the hardware prototype involved using the HIL setup outlined in Figure 1 and verifying whether steady-state stability could be maintained in the simulated two-area network. This test proved that the hardware prototype was able to prevent the existing inter-area oscillation mode from increasing in magnitude. Once this was demonstrated, the inter-area mode was excited by perturbing the voltage reference $V_{ref}$ of Machine M1. The oscillations caused by this disturbance would then be damped out by the proposed controller. Testing was carried out in several phases, one with the Simulink POD operational and the other with the cRIO-based WAPOD feeding in to the real-time simulation. In the latter case, three different damping inputs were tested: active power, positive sequence current magnitude and the voltage angle difference between buses 5 and 11 in Figure 4.

### A. SVC EXCITATION SUPPLEMENTARY INPUT

Figure 5 illustrates the response of the hardware controller to a small disturbance at machine M1. This disturbance is a 5% change for 200 ms. in the reference voltage of the AVR. This is sufficient to excite the inter-area mode. The best damping performance is achieved using voltage angle difference as a damping input to the Phasor-POD algorithm. This supports the theoretical results discussed in [4]. Response parameters such as settling times and overshoot were calculated based on the response in Figure 5 and are presented in Table 1. The response of the simulated POD (in Simulink) was used as a baseline to calculate the overshoot. The settling time is calculated as the time required for the response of the controller (in volts) to be constrained to +/- 1V.

**TABLE 1.** Response Parameters : SVC Excitation Supplementary Input

| Input Parameter | Percentage Overshoot | Settling Time (s) |
|---|---|---|
| Simulated POD (Active Power) | N/A | 3.875 |
| Active Power | 290.05 | 13.94 |
| Pos. Seq. Current | 142.67 | 16.95 |
| Voltage Angle Diff. | -11.26 | 9.66 |

All data presented in Figure 5 was captured in the real-time simulator. Data was also recorded at other locations in the simulation such as at the PDC but these have a lower time resolution and are thus not presented here. As further proof of the results presented in this work, the output of the hardware controller is captured directly using an oscilloscope (Figure 6).

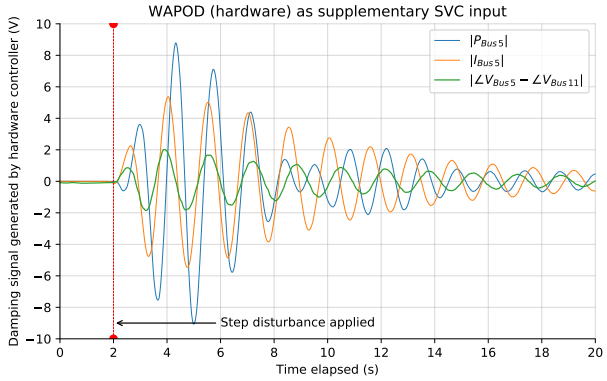


**FIGURE 5.** Controller Response Comparison : Supplementary SVC Excitation Input.

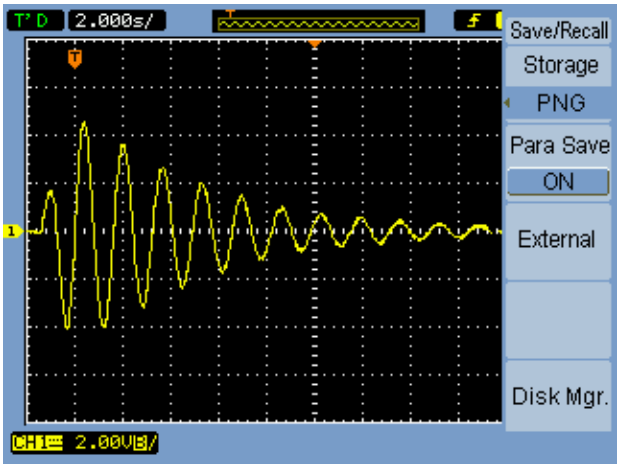


**FIGURE 6.** Controller Response with Voltage Angle Difference input Captured using an Oscilloscope.

### B. GENERATOR EXCITATION SUPPLEMENTARY INPUT

The controller response to a small disturbance when operating in tandem with a degraded PSS is shown in Figure 7. It can also be noted that the response of the damping signal is significantly different from that in Figure 5, where the dominant 0.64Hz mode is clearly visible. Also evident from Figure 7 is the fact that the damping performance of the WAPOD changes significantly as its input is changed. Using the voltage angle difference as input provides the best performance. The performance of the WAPOD with the voltage angle difference input is very close to the performance of the simulated POD algorithm that uses local active power as input. This performance is achieved despite the presence of a stochastic time delay and noise in the input measurements of the WAPOD. As in the case with the SVC, the theoretical results in [4] agree with the experimental results presented here.

As in the previous case, response parameters such as settling times and overshoot were calculated based on the response in Figure 7 and are presented in Table 2. Definitions are identical to the previous case.

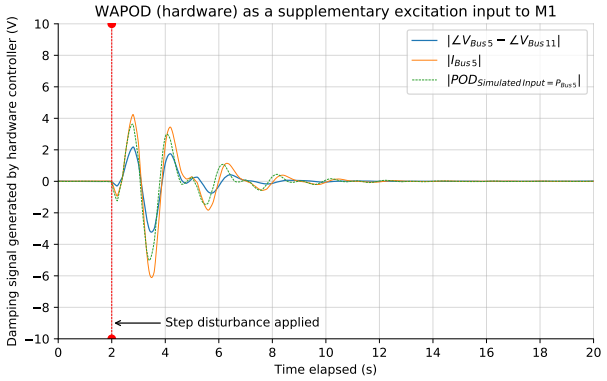


**FIGURE 7.** Controller Response Comparison : Supplementary Generator Excitation Input.

**TABLE 2.** Response Parameters : Supplementary Generator Excitation Input

| Input Parameter | Percentage Overshoot | Settling Time (s) |
|---|---|---|
| Simulated POD (Active Power) | N/A | 6.26 |
| Active Power | 0.10 | 6.41 |
| Pos. Sequence Current | 23.46 | 6.53 |
| Voltage Angle Diff | -34.72 | 5.20 |

## VI. CHALLENGES OF A REAL-WORLD IMPLEMENTATION

### A. TIME DELAYS

The Phasor POD algorithm implemented in Simulink has close to zero time delay between the power system changing and the controller responding. The same algorithm, when run on the cRIO, receives input data with a stochastic delay. It is evident from Figure 1 that several sources of stochastic time delay exist. Sections such as the analogue amplifiers, the FPGA (50$\mu$s) and the real-time section of the cRIO all represent fixed, non-zero time delays. Elements such as the D/A and A/D conversion in the real-time simulator also add a deterministic time delay. However, elements in the data path such as TCP/IP network communication and the PC used to extract raw measurement data from the PDC synchrophasor stream all represent variable delays. The total delay allows for a network disturbance to grow slightly before the controller begins to respond. This delay also changes the phase compensation required in the Phasor-POD algorithm. The phase compensation needed can be changed from the remote interface, depending on the measured delay. The experimentally measured end-to-end delay over the complete data path in Figure 1 averaged 344 ms. This was calculated by logging data in the real-time simulator and measuring the delay between the power system response and the controller response (Fig. 8). Presently, the phase compensation required is determined iteratively and off-line, however, this can be automated using time-stamped data together with an adaptive controller.

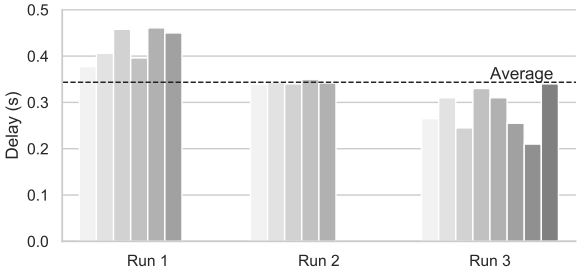


**FIGURE 8.** Time delay measurements for 19 HIL experiments.

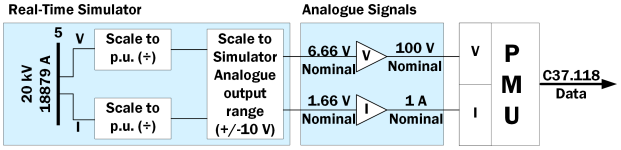


**FIGURE 9.** Signal Scaling at each step of simulation.

### B. ANALOGUE LIMITS AND NOISE

The original POD algorithm [10] was developed and simulated in an ideal, noise-free environment with zero delay (Simulink). More importantly, no limits are imposed on the magnitude of either the controller's inputs or outputs. In contrast, a hardware-based implementation as in this paper, usies analogue signals and thus is constrained by analogue signal limits (see Figure 9). Consider the analogue outputs of OPAL-RT's eMEGASIM simulator which are rated for $\pm 16V$ or $\pm 10mA$ [11]. In contrast, the standard input modules of typical PMU's are rated for 0-300V. A 0-16V analogue signal will not occupy a significant dynamic range of the PMU's inputs. Additionally, a signal of such a small magnitude will be contaminated by noise and will consequently have a poor Signal-to-Noise ratio. A similar argument can be made for the analogue current outputs. Our solution involved amplifying the analogue signals from the real-time simulator to levels that used more of the dynamic range of the PMU input modules. This simultaneously improves both accuracy and the signal-to-noise ratio.

The output of the simulated POD can vary over several orders of magnitude, ranging from $10^5$ p.u.. at times of peak damping to as small as $10^{-3}$ p.u. once the oscillation magnitude becomes small. It is difficult to accurately recreate this output as an analogue signal with such a vast dynamic range. The voltage output module used here had a 24-bit resolution and was limited to $\pm 10V$ 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. This was because a hard-wired POD output was used in our experiment. A solution would be to transmit the POD output digitally.

### 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 loop ran at different loop rates or step sizes (see Figure 2). Added to this was the fact that the real-time simulator generated data every 50 $\mu$s (and thus expected data from the HIL set-up at the same rate) while the rest of the HIL set-up did not support such a high data rate. The PMUs used supported a maximum reporting rate of 50 samples/second. Table 3 lists the different components of the HIL set-up together with their respective loop rates.

**TABLE 3.** Comparison of Loop Rates of Different Components

| Element | Loop Rate | Mode |
|---|---|---|
| OPAL-RT Simulator | 50$\mu$s | Real Time |
| PMU (cRIO) | 20ms | Real Time |
| Workstation Computer | 20ms | Not Real Time |
| PDC | 20ms | Not Real-Time |
| POD - RT Section | 20ms | Real Time |
| POD - FPGA | 50$\mu$s | Real Time |

50 $\mu$s. was chosen for the FPGA loop rate to match the loop rate of the real-time simulator. This ensures that data was always available at the analogue input of the real-time simulator and that no erroneous data points were read. The 50 $\mu$s. loop rate of the FPGA meant that new input data was expected every 50 $\mu$s., corresponding to a 20000 samples/s PMU sampling rate. The first limitation was the execution speed of the real-time section of the cRIO, which was limited to 1ms, significantly slower than the FPGA's speed. Further, new synchrophasor data was available from the PMUs only every 20 ms. One solution to this problem was up-sampling synchrophasor data. The major drawback of this method was that the up-sampling process on the RT controller is computationally intensive and would not run at the required 20 ms. loop rate. An alternative solution was a sample-and-hold algorithm implemented on the FPGA. The FPGA simply holds any data received by it till new data arrives. This was implemented and found to work satisfactorily.

### D. FPGA NUMERIC DATA FORMATS AND ACCURACY

While the FPGA is a fast, deterministic and reliable computational device, it brings with it some limitations. Most of these arise from the fact that an FPGA has no operating system and all circuit logic is directly implemented in hardware. All computations are performed at the bit level, limiting the amount and complexity of computations that can be performed. Functions such as division or multiplication consume significant space on the FPGA [23]. The FPGA on the cRIO9081 implements a numeric representation called Fixed Point [23]. Here, the number of bits assigned to represent the integer and fractional part of a number is fixed before code execution [23]. This assignment must be done in advance of code execution and the FPGA as a whole uses one fixed-point accuracy.

Fixing the number of bits used for fractions in the fixed-point representation is constrained by several factors such as the FPGA's analogue outputs, which have finite limits. This limits the maximum integer part of the numbers used in the fixed point representation. Problems arise, however, when handling input to the POD algorithm as its magnitude it not constrained by analogue limits. Observe from Figures1 and 2 that input data to the POD algorithm is digital and is transmitted over a TCP/IP network. This is the challenge with determining the fixed point representation used on the FPGA. Using too many bits for the integer part of the number results in poor fractional accuracy which is important as the analogue outputs have finite limits. Using too few bits for the integer part of the number results in possible rounding and overflow conditions when working with large input data values.

## VII. FURTHER WORK

This prototype is dependent on manual signal selection and algorithm parameter value calculation. These two processes can be automated by having the WAPOD itself monitor the various input signals and intelligently select the one having the highest observability of a particular mode [4]. The algorithm parameters, including the required phase compensation can also be determined adaptively on the WAPOD itself. On the same lines, this can further be extended to include selection among available measurement locations on the network. The fact that a stochastic time delay is introduced by unwrapping the synchrophasor data stream on a desktop computer can be addressed by performing this function on the WAPOD controller (here, the cRIO) itself thus making the whole control loop more deterministic which is part of on-going work [24].

## VIII. CONCLUSION

This work has provided experimental results showing the feasibility and benefits of using wide-area power system measurements as an input to an oscillation damping controller. The generic nature of the developed controller was demonstrated by using an identical implementation with mere changes in parameters to suit the controlled device's input requirements and capabilities. The hardware prototype developed on the NI cRIO was tested in a real-time HIL setup and was demonstrated to work satisfactorily. Real-world constraints such as time delays and signal conditioning were replicated as closely as possible. The flexibility of synchrophasor data was demonstrated and the wide range of inputs possible from this were also tested. The results from this work serve as experimental proof-of-concept to the theory presented in [4] and pave the way for the development of other PMU-based real-time control systems.

## REFERENCES


[1] M. Klein, J. G. Rogers and P. Kundur "A fundamental study of inter-area oscillations in Power Systems" IEEE Transactions on Power Systems, Volume. 6, Issue 3, pp. 914-921, 1991.

[2] K. Uhlen, L.Vanfretti, MM De Oliveira, AB Leirbukt, Vemund Halmo Aarstrand, Jan 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, 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] G. Rogers "Power system oscillations", Springer Science & Business Media, 2012.

[6] L. Peng, W. Xiaochen, L. Chao, Shi Jinghai, H. Jiong, He Jingbo, Zhao Yong, 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

[7] Zhang, F., Sun, Y., Cheng, L., Li, X., Chow, J. H., & Zhao, W. (2015) "Measurement and modeling of delays in wide-area closed-loop control systems" IEEE Transactions on Power Systems, 30(5), 2426-2433.

[8] Pal, S., Sikdar, B. and Chow, J.H., 2018. "An online mechanism for detection of gray-hole attacks on PMU data" IEEE Transactions on Smart Grid, 9(4), pp.2498-2507.

[9] R. Grondin, I. Kamwa, G. Trudel, L. Gerin-Lajoie, and J. Taborda. "Modeling and closed-loop validation of a new PSS concept, the multi-band PSS." In Power Engineering Society General Meeting, 2003, IEEE, vol. 3, pp. 1804-1809. IEEE, 2003.

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

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

[12] J.C Agee , S Patterson, R Beaulieu, Coultes M.,Grondin R. , Kamwa I., Trudel G., Godhwani A. , BÃl'rubÃl' R. , Hajagos L. , Malik O. , Murdoch A.,Boukarim G. , Taborda J. , and Thornton-Jones R. "IEEE tutorial course in Power System stabilization via excitation control" Technical report, June 2007

[13] 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, June 2012.

[14] 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

[15] 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.

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

[17] E. Rebello, L Vanfretti, and Md Shoaib Almas. "Software architecture development and implementation of a synchrophasor-based real-time oscillation damping control system" In PowerTech, 2015 IEEE Eindhoven, pp. 1-6. IEEE, 2015.

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

[19] E.V Larsen and E.H Chow, General Electric Company, NY, "Application of Static VAR Systems for System Dynamic Performance", 1987 IEEE pp. 43-46

[20] "FPGA Control on Compact RIO Sample Project Documentation", National Instruments, Available Online at http://www.ni.com/white-paper/14137/en/

[21] Grondin, R., Kamwa, I., Trudel, G., Gerin-Lajoie, L. and Taborda, J., 2003, July. "Modeling and closed-loop validation of a new PSS concept, the multi-band PSS" In Power Engineering Society General Meeting, 2003, IEEE (Vol. 3, pp. 1804-1809). IEEE.

[22] "Generic Power System Stabilizer" Documentation distributed with Matlab R2014a Available online at https://www.mathworks.com/

[23] "IP Corner: The LabVIEW Fixed-Point Data Type Part–1 Fixed–Point 101" Available Online at http://www.ni.com/newsletter/50303/en/, Dec 20, 2011

[24] L Vanfretti, GM Jónsdóttir, MS Almas, E Rebello, SR Firouzi, M Baudette 11Audur–A platform for synchrophasor-based power system wide-area control system implementation", SoftwareX Volume 7, January–June 2018, Pages 294-301 https://doi.org/10.1016/j.softx.2018.08.003

ELDRICH REBELLO (M'76–SM'81–F'87) and all authors may include biographies. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year the degree was earned. The author's major field of study should be lower-cased.

LUIGI VANFRETTI was born in Greenwich Village, New York, NY, USA in 1977. He received the B.S. and M.S. degrees in aerospace engineering from the University of Virginia, Charlottesville, in 2001 and the Ph.D. degree in mechanical engineering from Drexel University, Philadelphia, PA, in 2008.

MUHAMMAD SHOAIB ALMAS (M'87) received the B.S. degree in mechanical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2004 and the M.S. degree in mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2006. He is currently pursuing the Ph.D. degree in mechanical engineering at Texas A&M University, College Station, TX, USA.