Real-Time Implementation of a Flexible, Synchrophasor-based Wide-Area Damping Control System

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

***Abstract*—The modern power grid is increasingly being used under operating conditions of increasing stress for which it was not designed, giving rise to grid stability issues. One of these stability issues is the phenomenon of low frequency, electro-mechanically induced, inter-area oscillations. Simulations have demonstrated the advantages 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 work takes a Phasor-based oscillation damping algorithm and deploys it on a National Instruments real-time controller. The developed prototype is tested in a real-time Hardware-in-the-loop approach (RT-HIL) using OPAL-RT's eMEGASIM real-time simulation platform and synchrophasor data from real PMU's. It is demonstrated to have applications independent of the controlled device. Challenges faced, the solutions implemented together with the present prototype's limitations are also discussed.**

***Keywords*—WAPOD, WAMPAC, synchrophasor, PMU, damping control, Wide Area measurement and control**

I. INTRODUCTION

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-area oscillations use Power System Stabilizers (PSS) [7]. While a PSS provides excellent damping to intra-area modes with good local observability, its performance with inter-area modes may not be satisfactory [4]. The limited observability in a local signal can be complimented by wide-area, synchrophasor data.

A. Literature Review

Analytical Studies: It has been shown in [13] that wide-area signals are preferable to local signals, such as active power and frequency, [4] for the purpose of damping inter-area oscillations. Studies such as [13] have analysed damping performance when using signals such as voltage angle difference, which can easily be computed using data from multiple PMUs.

Eldrich Rebello has completed his Master's degree the School of Electrical Engineering, Aalto University, Helsinki, Finland e-mail: eldrich.rebello@aalto.fi

M. Shoaib Almas & Luigi Vanfretti are with KTH Royal Institute of Technology, Stockholm, Sweden. Email: msalmas@kth.se lvanfretti@kth.se

Luigi Vanfretti is with Statnett SF email : luigi.vanfretti@statnett.no

Field Tests: Successful field trials of wide-area oscillation damping controllers are reported in [7] and [8] indicating the potential that PMU-based wide area controllers have to offer. These results are not without certain limitations. In [7] 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.

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 [6].

B. Paper Contributions

The goal of this paper is to demonstrate both the potential and flexibility in oscillation damping controller design that is possible by using synchrophasors (IEEE C37.118). The Phasor Power Oscillation Damping (Phasor POD) algorithm originally developed by Ängquist and Gama [1] 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 [2] is executed in real-time on the eMEGASIM [3] 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.

C. Paper Organisation

This paper is organised as follows. Section II presents a brief background of the work presented in this study while Section III presents 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

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 they were 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 be excited.

A. Controller Choice

Traditional controllers for FACTS devices depend on accurate system models at a specific operating condition. Large systems often experience changes in their topology often and data about the present condition may not always be available. POD design 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 inter-connected power systems [5]. 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 uses few inputs and is independent of network configuration and topology. The only algorithm parameter that is network-dependent is the oscillation frequency and this is usually known from system studies or can be determined directly from synchrophasor measurements [14]. Compared to conventional controllers designed using linearisation-based methods, the adoption of phasor-based controllers has not been high mainly due to the fact that such controllers tend to be highly non-linear and thus difficult to, both, model in simulation studies [10] and implement in real-time applications [7].

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 Ångquist and Gama [1]. The measured signal can be represented as a space-phasor [10]:

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

where, $\vec{s}_{ph}$ is a complex phasor, rotating at the frequency ω [1]. 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 by Almas & Vanfretti [9] and its goal was to 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 $alpha$ 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 [1]. 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.

III. Software Architecture and Hardware Configuration

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] was implemented on a general purpose micro-controller and was executed in real-time. The inputs to the controller 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.

A. Hardware Configuration

As illustrated in Figure 1, all real-time simulations were performed on OPAL-RT's eMEGASIM [3] real-time simulation platform. SIMULINK models are executed in real-time and are interfaced with externally generated signals in a HIL configuration. Current and voltage signals from different points (Buses

[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.

5 and 11 in Figure 4) on the simulated network are extracted, amplified and then supplied as the input to two PMUs. Two cRIO-9076 [15] devices are used as PMUs. Each is equipped with three-phase analogue voltage and current input modules. The PMUs report data every 20 ms. The synchrophasor data stream (C37.118) generated by these PMUs is streamed over a TCP/IP network to a Phasor Data Concentrator (PDC) which produces a time-aligned output stream. This stream is then accessed, via a TCP/IP network, on a PC running LabVIEW. Data is extracted and sent to the FPGA running the Phasor-POD algorithm. The FPGA on the cRIO-9081 [15] generates a damping signal which is then wired to the real-time simulator for use in the SIMULINK model.

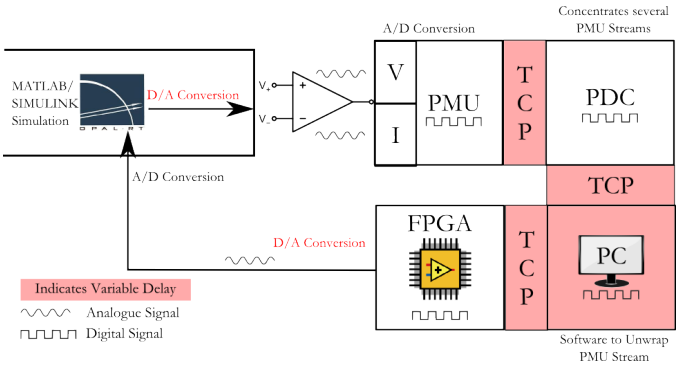

Fig. 1. Hardware Outline showing complete data path

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 discussed in Section VI-C.

B. Software Architecture

As shown in Figure 2, LabVIEW's Real Time and Field Programmable Gate Array (FPGA) modules were used to write the code for the respective sections of the cRIO. Also, since 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 Statnett's Synchrophasor Software Development Kit (S^3DK) [17] 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 [17]. Using the S^3DK 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 RT controller using LabVIEW's Shared Variables [20] over a TCP network.

C. Real-Time Implementation of Phasor-POD Algorithm

The hardware implementation of the POD was based on the Compact Reconfigurable Input / Output (cRIO) 9081 [15] from National Instruments. This controller is equipped with an on-board FPGA (maximum clock speed of 80Mhz[2]) in addition to an independent real-time controller (1.06GHz Intel Celeron U3405).

A three-layer, modular code architecture, following guidelines from the manufacturer [16], 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 update algorithm parameters and monitor data & performance. Receives the 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 terminal and also generates performance monitoring data. The remote interface interacts with this layer over the communication network.
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.

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 computer. The Phasor-POD algorithm can be controlled and monitored from this interface. 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 S^3DK was used to unwrap the PMU streams coming from the PDC and extract phasor measurements [17]. 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 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. If the voltage angle difference is to

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

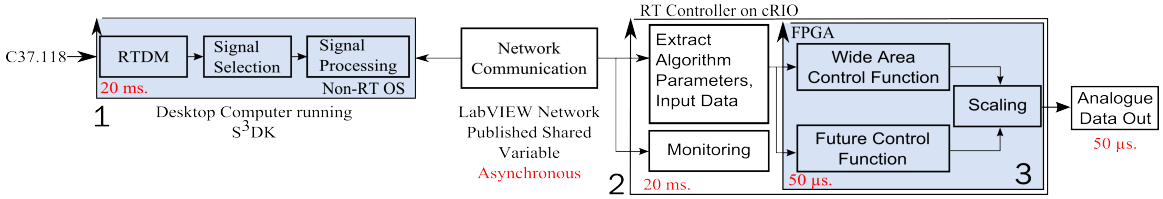

Fig. 2. Three-layer Software and Hardware Architecture of the WAPOD Controller. Loop rates are indicated in red.

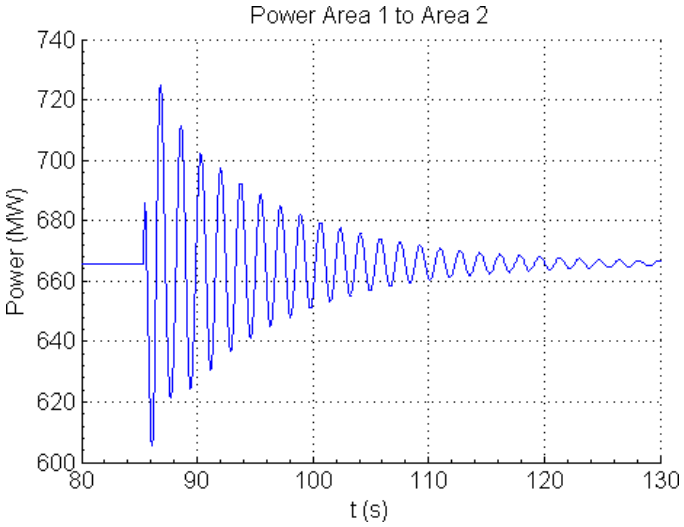

Fig. 3. Damping performance of the base case PSS only at Machine M1

be used as an input, the required calculations are performed in this LabVIEW Virtual Instrument (VI[3]).

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 and the other as a modulating input to the excitation system of a FACTS device (here, an SVC[4]). Figure 4 illustrates both these uses along with the two-area network outline. Note that both possibilities are not implemented simultaneously.

A. Two-Area Model Preparation - Generator AVR

The original Klein-Rogers-Kundur model in [2] was modified for the studies in this paper. 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. 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 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 where an already installed PSS whose performance has degraded over time due to changing network conditions or poor tuning. The next section illustrates that the addition of the WAPOD as an additional damping control aids damping performance when the conventional PSS cannot be recalibrated immediately.

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 except that a PSS was included at all four machines. The SVC model implemented was an average-value model (Figure **??**), identical to that used in [9]. The SVC was connected at the mid-point of the two area network (Bus 8 in Figure 4). As shown by Chow & Larsen in [12], this is the point where voltage swings will be the greatest and also where the SVC can be most effective at damping power swings. 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. In addition, the performance of the PSS's were not modified as the Phasor-POD algorithm would not be running in parallel with them.

C. Real-Time Simulation

The modified network in each case was grouped into subsystems and prepared for simulation using RT-LAB [3]. The simulation time step chosen was 50μs. The outputs and inputs of the real-time simulator were updated every 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

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

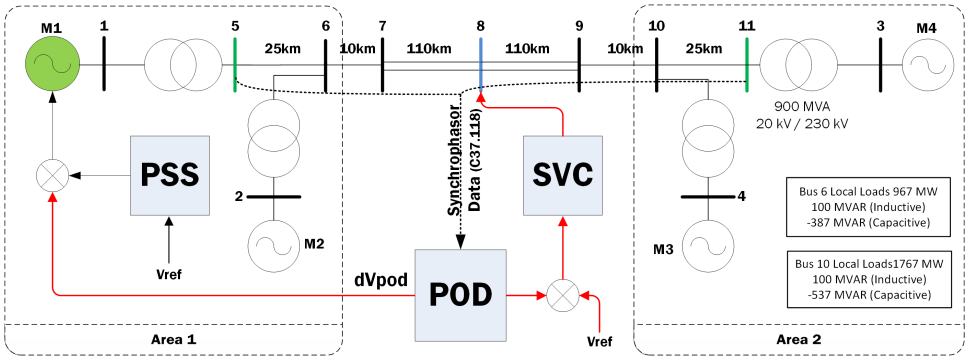

Fig. 4. Modified Two-Area Four-Machine network outline showing PMU data sources and scenarios with SVC and generator excitation supplementary input. Note that only one scenario is implemented at a time i.e. either generator or SVC modulation. Network details are left incomplete for illustration purposes.

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 the 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 was able to use the same signal as an input in each case but can also exploit other data contained in 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. 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 PSS in tandem with the Phasor POD algorithm. Testing was carried out in several phases, one with the SIMULINK POD connected and the other with the cRIO-based WAPOD connected 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 predicted by Chompoobutrgool and Vanfretti in [13]. Response parameters such as settling times and overshoot were calculated based on the response in Figure 5 and are presented in Table I. 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 I. 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.	-11.264	9.66

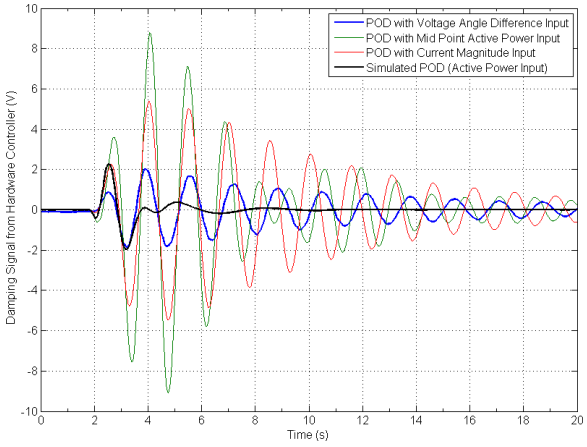

Fig. 5. Controller Response Comparison : Supplementary SVC Excitation Input

All data presented in Figure 5 was captured in the real-time simulator. Data was also recorded at other points such as at the PDC but these have a lower 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).

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

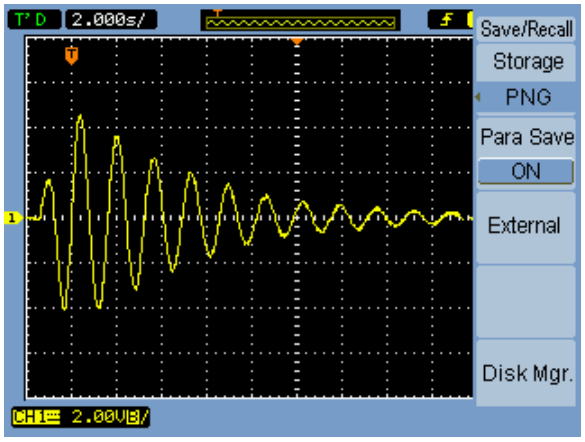

Fig. 6. Controller Response with Voltage Angle Difference input Captured using an Oscilloscope

is evident that the combination of the WAPOD and the PSS is significantly better than the PSS alone in every case. 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 with the simulated POD algorithm is not shown in Figure 7 for clarity. 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 [13] agree with the experimental results presented here.

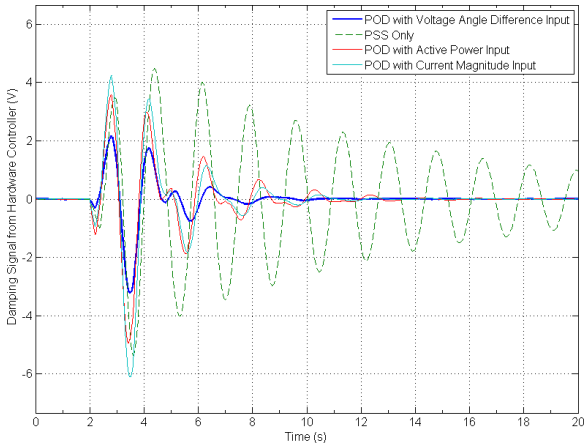

Fig. 7. Controller Response Comparison : Supplementary Generator Excitation Input

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 II. Definitions are identical to the previous case.

TABLE II. RESPONSE PARAMETERS : SUPPLEMENTARY GENERATOR EXCITATION INPUT

Input Parameter	Percentage Overshoot	Settling Time (s)
Simulated POD (Active Power)	N/A	6.255
Active Power	0.1	6.41
Pos. Sequence Current	23.46	6.525
Voltage Angle Diff	-34.72	5.2

VI. CHALLENGES

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 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 (50μs) 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 cRIO starts responding. It also has the effect of changing the phase compensation required in the Phasor-POD algorithm. Observe that 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 283.1 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. Data for this is presented in Table III. Presently, the phase compensation required is determined iteratively however, this can be automated using time-stamped data together with an adaptive controller.

TABLE III. SIGNAL PROPAGATION DELAY CALCULATION

Pulse Number	Simulated POD Response Start (s)	cRIO POD Response Start (s)	Time Delay (ms)
1	60.01	60.275	265
2	70.04	70.35	310
3	80.02	80.265	245
4	90.04	90.37	330
5	100.04	100.35	310
6	110.03	110.285	255
7	120.05	120.26	**210**
8	130.02	130.36	**340**

B. Analogue Limits and Noise

The original POD 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 in a simulation. In contrast, a hardware-based implementation using analogue signals is constrained by the analogue signal limits (see Figure 8). Consider the analogue outputs of OPAL-RT's eMEGASIM simulator which are rated for $\pm 16V$ or $\pm 10mA$ [3]. These are low level outputs and can be directly connected to the low-voltage level PMU inputs. However, the standard input modules of typical 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 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 solution involved amplifying the analogue signals from the real-time simulator to levels that used more of the dynamic change of the PMU input modules. This simultaneously improves both accuracy and the signal-to-noise ratio.

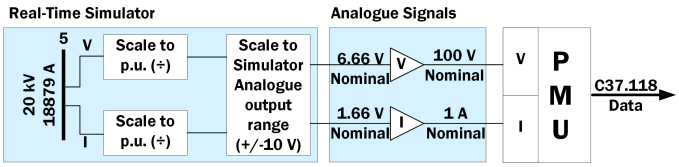

Fig. 8. Signal Scaling at each step of simulation

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 becomes small. It is difficult to accurately recreate analogue signals with such vast dynamic ranges. 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.

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 (see Figure 2). Added to this was the fact that the real-time simulator generated data every 50 μs (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 IV lists the different components of the HIL set-up together with their respective loop rates.

50 μs. 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 50 μs. loop rate of the FPGA meant that new data was expected every 50 μs., corresponding to a 20000 samples per second PMU sampling rate. The fastest execution speed of the real-time

TABLE IV. 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
PDC	20msec	Not Real-Time
POD - RT Section	20msec	Real Time
POD - FPGA	50μs	Real Time

section of the cRIO was 1ms, significantly slower than the FPGA's speed. New synchrophasor data was available from the PMUs only every 20 ms. One solution to this problem was to up-sample the synchrophasor data, interpolate between consecutive data points and then stream this data to the Phasor-POD algorithm on 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 20 ms. loop rate. An alternative solution was a sample and hold algorithm implemented on the FPGA. As data was extracted every 20 ms., 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 arrived. 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 [20]. The FPGA on the cRIO9081 implements a unique numeric representation called Fixed Point [20]. Here, the number of bits assigned to represent the integer and fractional part of a number is fixed before code execution [20]. This representation is similar to binary in the sense that increasing and decreasing powers of 2 are used to represent the integer and fractional parts of a number, respectively.

Floating point calculations, although possible, consume significant space on the FPGA and are typically slower than corresponding fixed-point calculations [20]. Trigonometric functions such as a sine or cosine can be implemented using specifically designed code that takes several clock cycles to execute. A trade-off has to be made between code execution speed and accuracy. The Phasor POD algorithm implemented in this work uses floating point calculations, multiplication, division operations and trigonometric operations.

Due to the FPGA design, not all these calculations can be performed in the same data format. The FPGA-specific data format, the Fixed Point representation, is used to optimise complex computations such as those required in trigonometric functions. The drawback of this representation is that a trade-off must be made between the accuracy achieved and the range of values that can be represented. Keeping in mind the fact that the input values to the POD algorithm are not

necessarily limited in magnitude, the POD algorithm is implemented using Floating-point numbers. Certain functions used in the algorithm, such as trigonometric functions, use FPGA-optimised code and require input and output in the fixed-point representation. Conversion between these two formats (Fixed and Floating point) sometimes results in errors. The floating-point format includes a representation for calculations that result in infinite or complex values, called `NaN` (Not a Number) [20]. This representation is not available in the fixed-point format and conversion results in errors.

VII. Further Work

The WAPOD prototype as developed in this work together with the experimental tests give strong evidence of the possibilities and benefits of using wide area synchrophasor data for damping control, as theorised in [13]. The prototype here is, however, dependent on manual input for signal selection and algorithm parameter values. 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 [13]. The algorithm parameters, including the required phase compensation due to a stochastically varying delay, can also be determined adaptively on the WAPOD itself. On the same lines, this can further be extended to include selection among several 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. The HIL system at present includes several sources of stochastic delay besides the delay caused by TCP communication. The HIL set-up can also be expanded to use actual generator excitation control hardware such as ABB's ECS system.

VIII. Conclusion

This work has provided experimental proof of 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. The performance of the WAPOD can be improved by exploiting the full range of data available in a synchrophasor data stream. 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 [13] and pave the way for the development of other PMU-based real-time control systems.

Appendix A
SmarTS Lab Outline

The full listing of the SmarTS Lab, the simulator's capabilities and its interfaces is covered in [9] and [11]. The reader is referred to these sources for a better understanding of the hardware described here.

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] 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

[5] J.C Agee , S Patterson, R Beaulieu, Coultes M.,Grondin R. , Kamwa I., Trudel G., Godhwani A. , Brub 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

[6] C.F.M. Danielson and L. Vanfretti and , M.S. Almas and , Y. Choompoobutrgool and , J.O. Gjerde *Analysis of communication network challenges for synchrophasor-based wide-area applications* Bulk Power System Dynamics and Control - IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium

[7] Uhlen, K. and Vanfretti, L. and De Oliveira, M. M. and Leirbukt, AB. and Aarstrand, V. H. and Gjerde, J. O. *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

[8] 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

[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] 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

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

[12] 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

[13] 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.

[14] 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.

[15] *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/

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

[17] 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

[18] I Kamwa *Performance of Three PSS for Interarea Oscillations* SimPowerSystems Examples, MATLAB R2011a

[19] *Generic Power System Stabilizer* Documentation distributed with MATLAB R2014a Available online at http://www.mathworks.se/help/physmod/sps/powersys/ref/genericpowersystemstabilizer.html

[20] *LabVIEW Online Help System Available Online at* http://sine.ni.com/psp/app/doc/p/id/psp-357