

# 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, electromechanically 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

THE goal of this paper is to demonstrate both the potential and flexibility in oscillation damping controller design that is possible by using synchrophasors (C37.118). The Phasor Power Oscillation Damping (Phasor POD) algorithm originally developed by Ångquist and Gama [1] is implemented and run 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 input and using each as a damping input to the controller. This paper also illustrates that the controller can have multiple applications is demonstrated by testing it with two different controlled devices; a generator

---

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 and L. Vanfretti are with KTH Royal Institute of Technology, Stockholm, Sweden. Email: msalmas@kth.se luigiv@kth.se

This project was supported in part by the Nordic Energy Research through the STRONg<sup>2</sup>rid project.

AVR system and a FACTS-device excitation system. A brief analysis of the performance of each input is also presented.

### A. Motivation

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 inter-area modes may not be satisfactory [5].

### B. Literature Review

Analytical studies and tests. One test from China and one from Norway.

### C. Paper Contributions

### D. Paper Organisation

This paper is organised as follows:

Section II presents a brief background of the work together with the hardware and software architectures implemented. An outline of the two-area model, the hardware used and finally, an outline of the whole HIL test setup is also presented.

Section IV covers the preparation of the SIMULINK models for real-time simulation and the modifications made for their execution on a real-time target. Two test cases are examined, 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 and problems faced in the development, implementation and testing of this WAPOD controller including software

challenges and the difficulties faced when dealing with real-world, analogue signals compared to pure simulations.

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 systems models at the operating condition. Large systems also tend to be dynamic changing 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 of power system models. These models are often difficult to derive accurately for large and inter-connected power systems [6]. 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 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 (**citation** here). 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 [9] and implement in real-time applications (**citation** here).

### B. Phasor POD Algorithm

Some of the problems of 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 [1]. The measured signal can be represented as a space-phasor [9]:

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

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 also incorporate a degree of error checking and phasor estimation.

The real-time implementation of the Phasor-POD algorithm was based on the SIMULINK implementation by Almas & Vanfretti [7] and its goal was the 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  $\alpha$  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].

## III. SOFTWARE ARCHITECTURE AND HARDWARE CONFIGURATION

The Wide Area Power Oscillation Damper (WAPOD)<sup>1</sup> 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.

All real-time simulations here are performed on OPAL RT’s eMEGASIM [8] real-time simulation 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 [14] 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

<sup>1</sup>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.

cRIO-9081 [13] generates a damping signal which is then wired to the real-time simulator for use in the SIMULINK model.

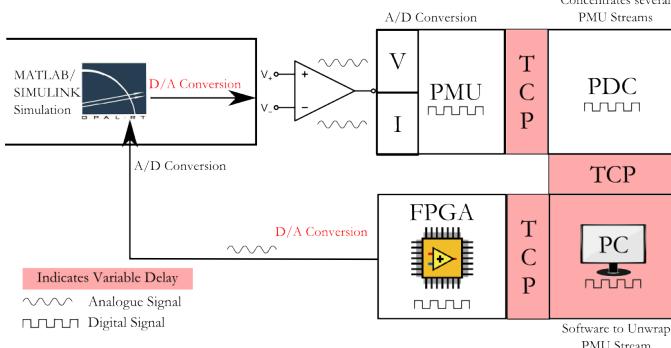


Fig. 1. Harware Outline showing complete data path

Vanfretti et. al. describe the details of the equipment used here in [10]. An image of the physical setup is included for clarity in Appendix B. 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 VI-C.

### A. Software Architecture

The two-area four-machine model used here is available in SIMULINK's SimPowerSystems. The SIMULINK implementation of the Phasor-POD algorithm was developed by Almas and Vanfretti [7]. 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 [16] 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 [16].

### B. Real-Time Implementation of Phasor-POD Algorithm

The hardware implementation of the POD was based on the Compact Reconfigurable Input / Output (cRIO) 9081 [13]

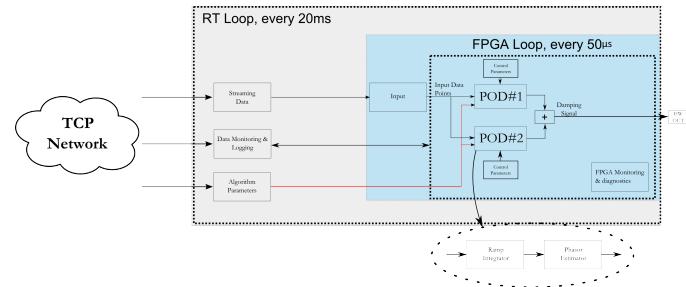


Fig. 2. Software Architecture of RT Controller

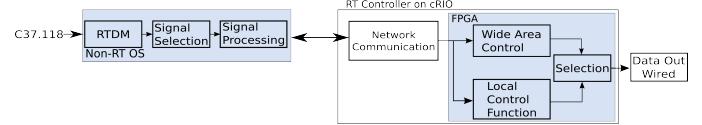


Fig. 3. RT Controller Architecture. The local control function is not implemented in this work

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.

A three-layer, modular code architecture, based on an available template [15] was selected for implementation. An outline of the architecture is shown in Figure 3 and each of the three layers are outlined below.

- 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 50 $\mu$ s 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 [16]. 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.

#### IV. 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 generator AVR modulating input and the other as a modulating input to the excitation system of a FACTS device (here, an SVC<sup>2</sup>). Figure 5 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

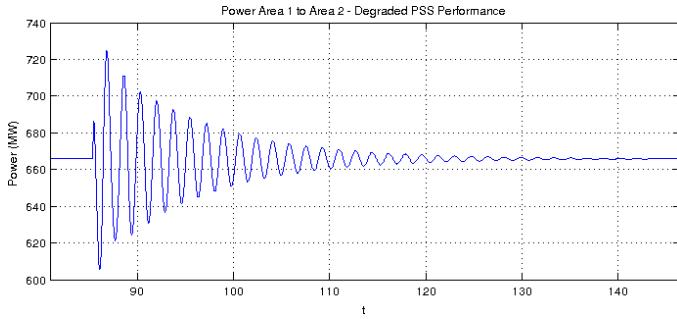


Fig. 4. Degraded performance with PSS

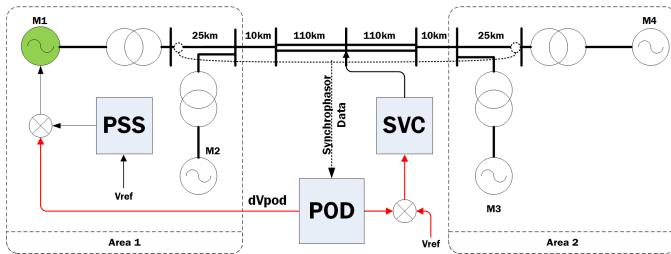


Fig. 5. Modified Two-Area Four-Machine network

<sup>2</sup>Static VAR Compensator

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 5). 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 4). 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.

##### B. Two-Area Model Preparation - FACTS Device

The second application in Figure 5 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 the PSS at each machine was included. The SVC model used was an average valued model, identical to that used in [7]. The SVC was connected at the mid-point of the two area network. As proved by Chow and Larsen in [11], this is the point where voltage swings will be the greatest and also where the SVC can be most effective at damping power swings. As before, 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 algorithm 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 an off-line simulation using the Phasor-POD implemented in SIMULINK. Also, the performance of the PSS's were not degraded 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 in RT-LAB. The simulation time step chosen was  $50\mu\text{s}$ . Outputs and inputs of the

real-time simulator would be updated every  $50\mu\text{s}$ . Current and voltage measurements were taken from the points marked in Figure 5 and were extracted from the analogue outputs of the real-time simulator. The damping signal  $dV_{pod}$  in Figure 5 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

Testing the operation of the hardware prototype involved using the HIL<sup>3</sup> 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 changing 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. 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 prototype is also able to use this same signal as input in each case but can also exploit other data contained in the synchrophasor data stream.

### A. SVC Excitation Supplementary Input

Figure 6 illustrates the response of the hardware controller to a small disturbance at machine M1. This disturbance is a 5% change for 200ms 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 [12]. A frequency analysis of each of these responses shows that the dominant mode in each response is the 0.64Hz inter-area mode.

Response parameters such as settling times and overshoot were calculated based on the response in Figure 6 and are presented in Table I.

TABLE I. RESPONSE PARAMETERS : SVC EXCITATION SUPPLEMENTARY INPUT

Element	Loop Rate	Mode
OPAL RT Simulator	$50\mu\text{s}$	Real Time
PMU (cRIO)	20msec	Real Time
Workstation Computer	20msec	Not Real Time
POD - RT Section	20msec	Real Time
POD - FPGA	$50\mu\text{s}$	Real Time

All data presented in Figure 6 was captured in the real-time simulator. Data was also recorded at other points such as at the

<sup>3</sup>Hardware in the Loop

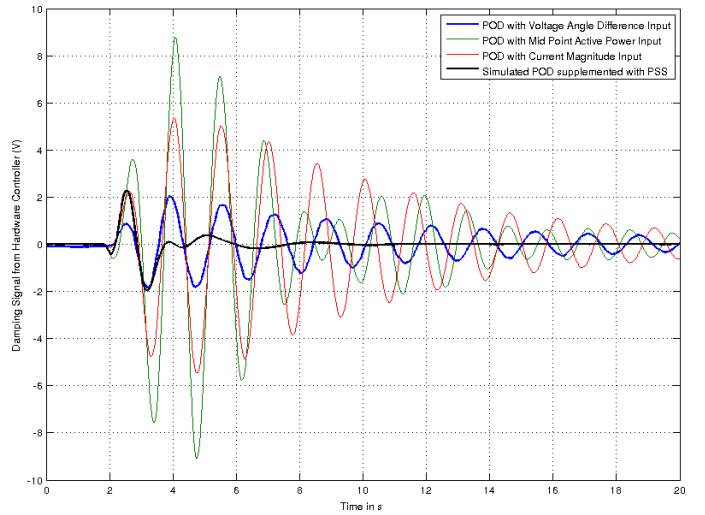


Fig. 6. Controller Response Comparison : Supplementary SVC Excitation Input

PDC but these have a lower resolution and are not presented here. As further proof of the results presented in this work, the output of the hardware controller is captured directly and presented on an oscilloscope (Figure 7).

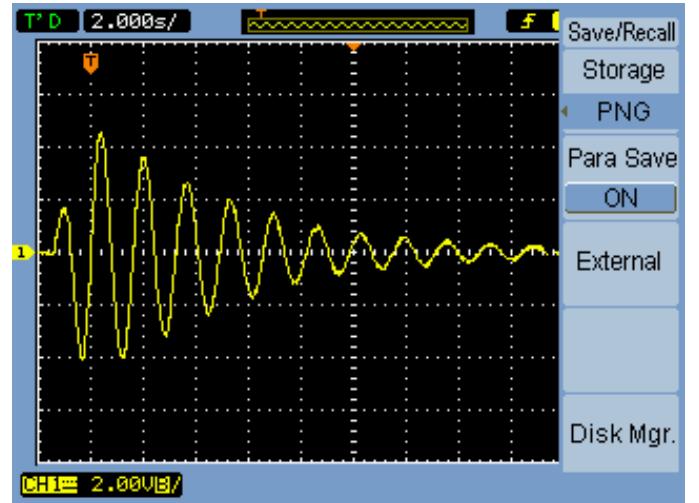


Fig. 7. Controller Response with Voltage Angle Difference input Captured on Oscilloscope

### B. Generator Excitation Supplementary Input

The controller response to a small disturbance when operating in tandem with a degraded PSS is shown in Figure 8. It 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 shape of the response shows a significant deviation from that in Figure 6, where the dominant 0.64Hz mode is clearly visible. Also evident from Figure 8 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 8 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 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 [12] agree with the experimental results.

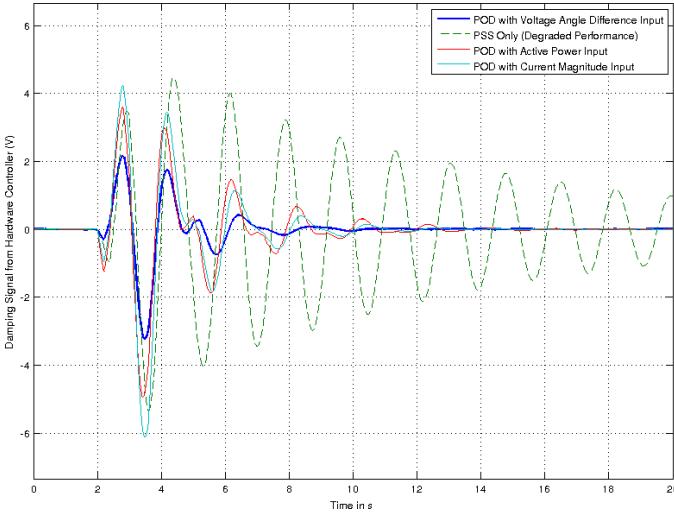


Fig. 8. Controller Response Comparison : Supplementary Generator Excitation Input

As in the previous case, response parameters such as settling times and overshoot were calculated based on the response in Figure 8 and are presented in Table ??.

TABLE II. RESPONSE PARAMETERS : SUPPLEMENTARY GENERATOR EXCITATION INPUT

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

## VI. 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 50 $\mu$ s), 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 is constrained by the analogue signal magnitude limits.

Consider the analogue outputs of OPAL RT's eMEGASIM simulator [8].

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

These are low level outputs and can be directly connected to the PMU inputs. However, the inputs 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 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 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 than 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  $50\mu s$  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 II lists the different components of the HIL setup together with their respective loop rates.

TABLE III. COMPARISON OF LOOP RATES OF DIFFERENT COMPONENTS

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

$50\mu s$  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  $50\mu s$  loop rate of the FPGA meant that new data was expected every  $50\mu s$ , 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<sup>4</sup> 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  $50\mu s$  while the cRIO-based POD received data every 20ms.

#### 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 as such and all circuit logic is directly implemented in hardware. All computation is performed at the bit level and can hence become very complex. This limits the amount and complexity of computations that can be performed with the FPGA. Functions such as division or multiplication consume

significant space on the FPGA [20]. The FPGA used on the cRIO9081 implements a unique numeric representation called Fixed Point [19]. Here, the number of bits assigned to represent the integer and fractional part of a number is fixed before code execution [20]. For example, if four bits are used to represent the integer part of a number, then the maximum binary number that can be represented is [1111] or 15 in decimal. 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 thesis 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, in particular the 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. The most common result is that a conversion from a floating-point NaN results in a fixed-point number where all the bits are 1. This produces a finite number and is incorrect.

## VII. FURTHER WORK

The WAPOD prototype as developed in this work demonstrates the possibilities of using wide area synchrophasor data for damping control. 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. The algorithm parameters 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

<sup>4</sup>First In First Out

the synchrophasor data stream on a desktop computer can be addressed by performing this function on the WAPOD controller (here, the cRIO) itself.

## VIII. CONCLUSION

This work has demonstrated the feasibility of using wide-area power system data in the design of 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 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 a proof-of-concept to what?

## APPENDIX A MODIFICATIONS MADE TO PSS MODEL

The default simulation model supplied with the `power_PSS` example in MATLAB has three types of PSS models [17] : an MB-PSS, a  $\Delta\omega$ -PSS and a  $\Delta P_a$  PSS. The comparison of the three as presented in [17] concludes that the performance of the MB-PSS is the best. This model is thus, not used when modulating the PSS output with the Phasor-POD algorithm. Instead, the  $\delta P_a$  PSS is used. The PSS model here is the generic PSS model available with SimPowerSystems [18].

The input to the  $\Delta\omega$ -PSS is the rotor speed deviation  $\Delta\omega$ . For an accurate comparison, this was changed to the measured active power measured at the generator terminals,  $P_a$ . The lead-lag block parameters were also changed from  $\frac{1+50x10^{-3}s}{1+1s}$  to  $\frac{1+50x10^{-3}s}{1+6s}$ . Parameters here were determined iteratively. The change to the lead-lag block produces the effect in Figure 4 where the PSS is still able to restore the system to stability but takes almost a minute to damp out the oscillations versus approximately three seconds in the ideal case.

## APPENDIX B SMARTS LAB OUTLINE & SETUP IMAGES

The SmarTS Lab at KTH was set up with the aim of developing wide area monitoring, protection and control (WAMPAC) schemes for the power grid. Much of the infrastructure and activities involve PMU data and the associated communication and computer systems [10]. The lab is equipped with facilities for real-time (RT) simulations and also RT Hardware-in-the-loop (HIL) tests. A reduced schematic is shown in Figure 10

The core of the setup is the eMEGASIM Real-time simulator from OPAL RT [3]. Two 'targets' are available, each running a 12-core 3.3Ghz Intel i7 processor [10]. This allows the running of models created in MATLAB/Simulink

in real-time. These simulations can interact with external devices through the simulator's low-power analogue outputs and inputs or with data streamed over TCP/IP, UDP etc [10]. In this thesis, the power system under test is simulated on this platform.

The analogue outputs and their ratings are listed below:

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

The full listing of the simulator's capabilities and interfaces is covered in [10]. Only necessary and relevant details are covered here.

The WAMPAC platform includes a Phasor Data Concentrator (PDC) and its associated software from Schweitzer Engineering Laboratories (SEL). Other devices are interfaced with the PDC such as protection relays with embedded PMU functionality, line differential protection relays (ABB), Compact RIO micro-controllers (National Instruments) and analogue signal amplifiers (Megger) [10]. The hardware list here is incomplete and other devices are also used such as a GPS receiver, a relay current and voltage injection kit etc.

## 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] 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 power system stabilization via excitation control*. Technical report, June 2007.
- [7] 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.
- [8] eMEGAsim PowerGrid Real-Time Digital Hardware in the Loop Simulator Opal RT, [Online]. Available: <http://www.opal-rt.com/>
- [9] 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

- [10] 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
- [11] 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
- [12] 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. Available Online at <http://www.springer.com/energy/systems%2C+storage+and+harvesting/book/978-1-4614-1802-3>
- [13] *Operating Instructions and Specifications Compact RIO NI cRIO-9081/9082*, National Instruments, Available Online at <http://www.ni.com/pdf/manuals/375714a.pdf>
- [14] *Operating Instructions and Specifications Compact RIO NI cRIO-9075/9076*, National Instruments, Available Online at <http://www.ni.com/pdf/manuals/375650b.pdf>
- [15] *FPGA Control on Compact RIO Sample Project Documentation*, National Instruments, Available Online at <http://www.ni.com/white-paper/14137/en/>
- [16] 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
- [17] I Kamwa *Performance of Three PSS for Interarea Oscillations Simulations Examples*, MATLAB R2011a
- [18] *Generic Power System Stabilizer Documentation* distributed with MATLAB R2014a Available online at <http://www.mathworks.se/help/physmod/sps/powersys/ref/genericpowersystemstabilizer.html>
- [19] *LabVIEW Online Help System Available Online* at <http://sine.ni.com/psp/app/doc/p/id/psp-357>
- [20] *LabVIEW Course Manuals - Core 1, Real Time, FPGA, August 2012 Editions*, Copyright 2012 National Instruments

**M S Almas** Biography text here.



**Eldrich Rebello** Biography text here.

**Luigi Vanfretti** Biography text here.





Fig. 9. Outline of the SmarTS Lab. Corresponding to Numbers: 1: Real Time Simulator, 2: PDC Interface, 3: cRIO Tray, 4: Oscilloscope, 5: Analogue Signal Amplifiers, 6: SEL Protection Relays

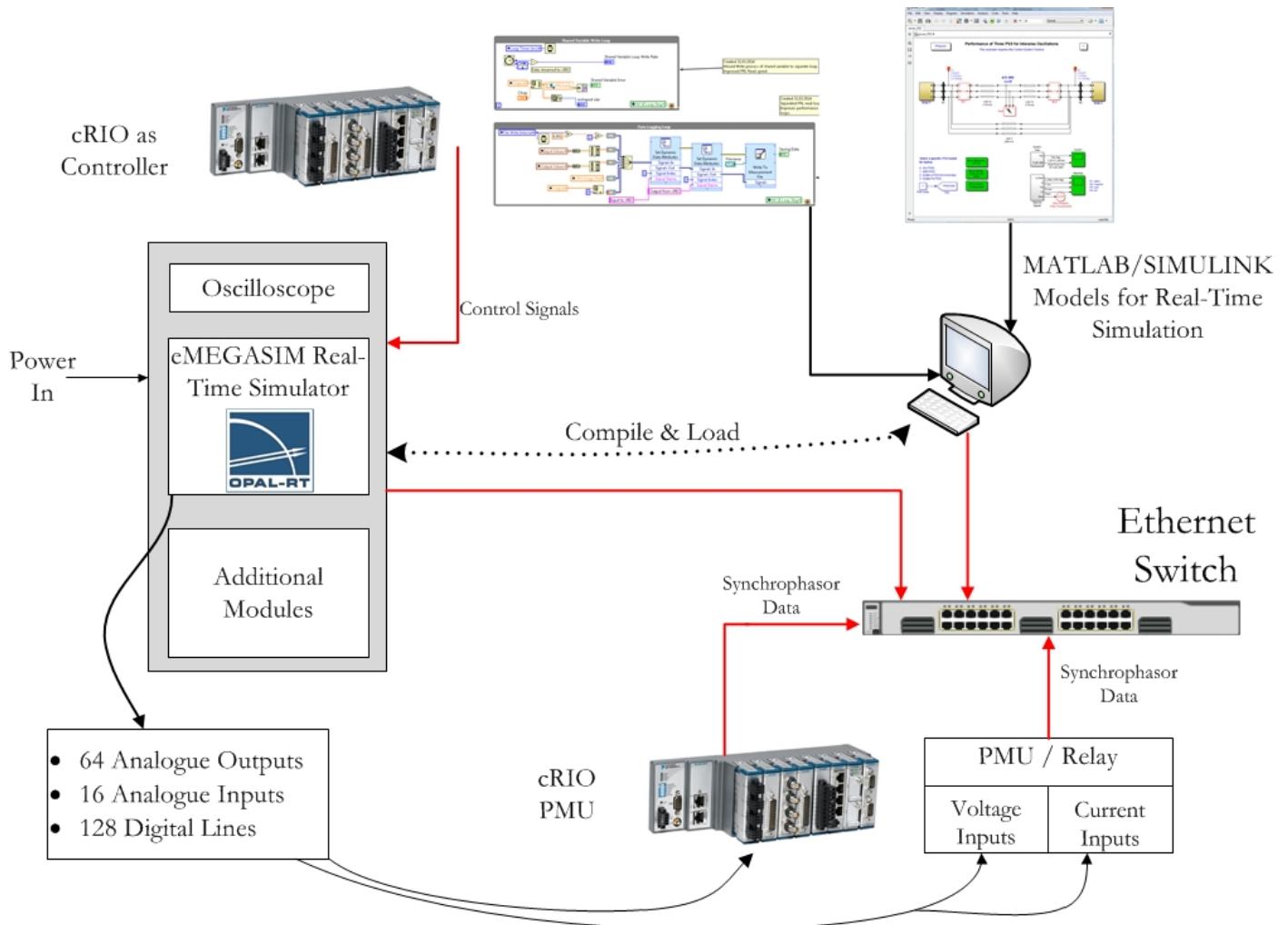


Fig. 10. Outline of SmarTS Lab at KTH