Software Architecture Development and Implementation of a Synchrophasor-based Real-Time Oscillation Damping Control System

Eldrich Rebello
Kungliga Tekniska högskolan
Stockholm, Sweden
Email: rebello@kth.se

Luigi Vanfretti
Kungliga Tekniska högskolan
Stockholm, Sweden & Statnett SF, Oslo, Norway
Email: luigi.vanfretti@statnett.no

Md. Shoaib Almas
Kungliga Tekniska högskolan
Stockholm, Sweden
Email: msalmas@kth.se

***Abstract*—Low-frequency, electromechanically induced, inter-area oscillations are of concern in the continued stability of inter-connected power systems. Wide Area Monitoring, Protection and Control (WAMPAC) systems based on wide-area measurements such as synchrophasor (C37.118) data can be exploited to address the inter-area oscillation problem. This work develops a hardware prototype of a synchrophasor-based oscillation damping control system. A Compact Reconfigurable Input Output (cRIO) controller from National Instruments is used to implement the real-time prototype. This paper presents the design process followed for the development of the software architecture. The design method followed a three step process of design proposal, design refinement and finally attempted implementation. The goals of the design, the challenges faced and the refinements necessary are presented. The design implemented is tested and validated on OPAL RT's eMEGASIM real-time simulation platform and a brief discussion of the experimental results is included.**

A. Previous Experiences

Modern solutions to the problem of inter-area oscillations involve using Power System Stabilisers (PSS) with locally available signals. These are effective at damping intra-area modes with good observability but may not be as effective at damping inter-area modes [2] [3]. Although initial field tests in [2] & [5] show promising results, the wide-area control systems tested so far have been implemented by extending the installed control system of an existing device (e.g. an SVC, see [2]) to receive synchrophasor data, process it and to then feed the control algorithm. To the knowledge of the authors, there has not been any reported attempt at designing a general purpose, wire-area control system, starting from specifications and considering different hardware and software constraints. Such an approach is attractive as it can easily be adapted to different controllable elements thereby reducing implementation costs and facilitating straightforward development.

B. Contributions

The goal of this paper is to document the software architecture development process and challenges faced in the real-time implementation of a wide-area control system. Ängquist and Gama's [6] Phasor Power Oscillation (Phasor POD) algorithm was chosen based on the operational requirements and control system design constraints. The developed prototype is termed a Wide Area Power Oscillation Damper (WAPOD). For purposes of comparison, the real-time SIMULINK implementation of the Phasor POD algorithm by Almas and Vanfretti [8] is used as a benchmark. The performance of the hardware prototype developed will be tested and compared to the performance of this implementation of the algorithm. The two-area four-machine model developed by Klein, Rogers and Kundur [9] is used as a test-case and is modified to fit the requirements of experimental testing. A Hardware-in-the-loop experiment is constructed around the eMEGASIM real-time simulator from OPAL-RT [10] with the two-area model running in real-time, physical PMUs and a synchrophasor-based Phasor-POD algorithm running on a cRIO (Figure 1).

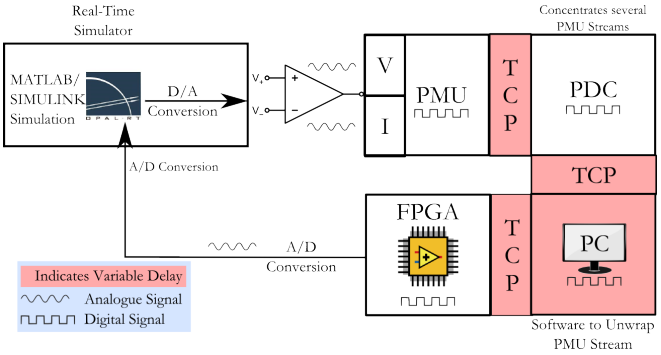

Fig. 1. Data flow with digital and analogue components indicated. Sections in white add deterministic delay to the control loop.

I. Architecture Development Process

The software development approach followed here is based on the Waterfall method detailed in [15]. The linear waterfall method was modified to be iterative to account for various constraints and to also account for revisions in the initial definition caused by these constraints. This iterative process is illustrated in Figure 2, sub-figure 1 and is realised for the specific application here, as illustrated in Figure 2, sub-figure 2.

II. Architecture Implementation and Refinement

Figure 2, sub-figure 1 shows the three-stage design process with design proposal, implementation and revision. To begin with, the controller specifications were used to draft a design proposal. An implementation attempt was made using this draft proposal. When the additional limits imposed by the software and hardware platforms were included, some goals of the original implementation needed to be revised. With these limitations, the original design was modified to generate a revised design. An attempt was then made to implement this revised design. As further limitations were encountered, the design was further modified. This iterative process was repeated till a working implementation was reached.

1) Initial Design Proposal: Initially, an autonomous, independent controller was envisaged. Such a controller would be able to receive IEEE C37.118 synchrophasor data directly over the communication network and extract measurement data (Figure 2, sub-figure 2). This design also incorporated two control functions, a wide area control function and a local control function. The wide area control function was the Phasor POD [6]. The local control function would use locally available data in a manner similar to a Power System Stabiliser (PSS). Automatic switching between the two functions and automated input signal selection was also considered. If the signal to noise ratio in the wide area signal deteriorated or if the signal delay became prohibitively high, the controller would automatically switch to the local signal or another wide-area signal. The principal consideration in this design was the limited resources available on the FPGA and the complexity of the algorithms to be implemented. This required implementation of all algorithms and computation sections on the RT section of the cRIO with the FPGA being used for interpolation only.

2) Development and Revision: With the limitations of the initial architecture in mind, the architecture was refined. The process of extracting data from the synchrophasor stream was shifted to a workstation computer along with the processes of signal selection and processing. Once data was available, it was sent to the RT controller, over the network. The damping control was kept on the RT controller with the FPGA performing the interpolation required to match the read-interval of the real-time simulator. Besides the damping control algorithm, a local control function was included on the RT controller. This revision is shown in Figure 2, sub-figure 3. Here, the complexity of implementing an interpolation algorithm on the FPGA was examined in detail. An implementation was also attempted. It was determined that a simple linear interpolation algorithm would not be sufficiently accurate. An implementation of the damping control algorithm on the RT controller was also tested. The fastest loop rate that could be achieved was always in excess of 25ms. This was slower than the reporting rate of the PMUs. Communication between the cRIO and the RT simulator using TCP/IP was also abandoned. Analogue output signals of the FPGA were instead hard-wired to the real-time simulator's analogue inputs.

3) Final Implementation: At this stage, the damping control algorithm was moved to the FPGA with the RT controller being used only for network communication and data monitoring. The Phasor POD algorithm parameters and monitoring data would be sent over the network to the RT section of the controller. The local control function was also discarded. The desired 50μs. data output rate could also be maintained as the FPGA was capable of response times of this order. The problem of differing data and loop rates was solved by implementing a basic sample-and-hold algorithm on the FPGA. With the Phasor POD algorithm implemented on the FPGA, resource utilization stood at 78%. As with the previous design, the process of extracting data from the synchrophasor stream was done on a workstation computer. This (Figure 2) was the final architecture as implemented.

III. Hardware-in-the-Loop Test

A setup identical to that outlined in Figure 1 was constructed and used to verify the working of the hardware WAPOD.

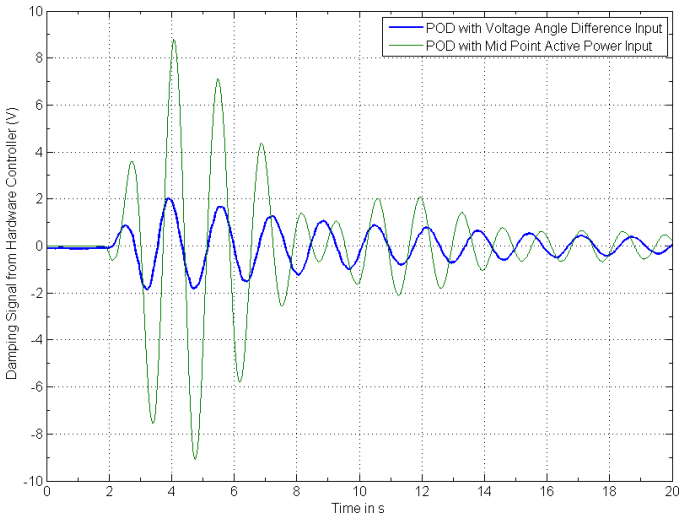

Fig. 3. Performance of Hardware controller with small disturbance. Note that the decreasing magnitude of the damping signal indicates that the oscillation magnitude is decreasing in correspondence.

The hardware implementation of the POD algorithm was verified to be able to keep the system stable in steady state. It was also tested with small perturbations applied at one generator to excite the inter-area mode and was verified to work satisfactorily. Figure 3 shows the performance of the controller with two different inputs,active power and voltage angle difference. It is immediately evident that using the voltage angle difference provides better performance.

References

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

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

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

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

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

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

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

[8] 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.

[9] 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.

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

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

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

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

[14] 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

[15] James Taylor *Managing Information Technology Projects : Applying Project Management Strategies to Software, Hardware, and Integration Initiatives*, AMACOM Books, September 2003 pp 237-239

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

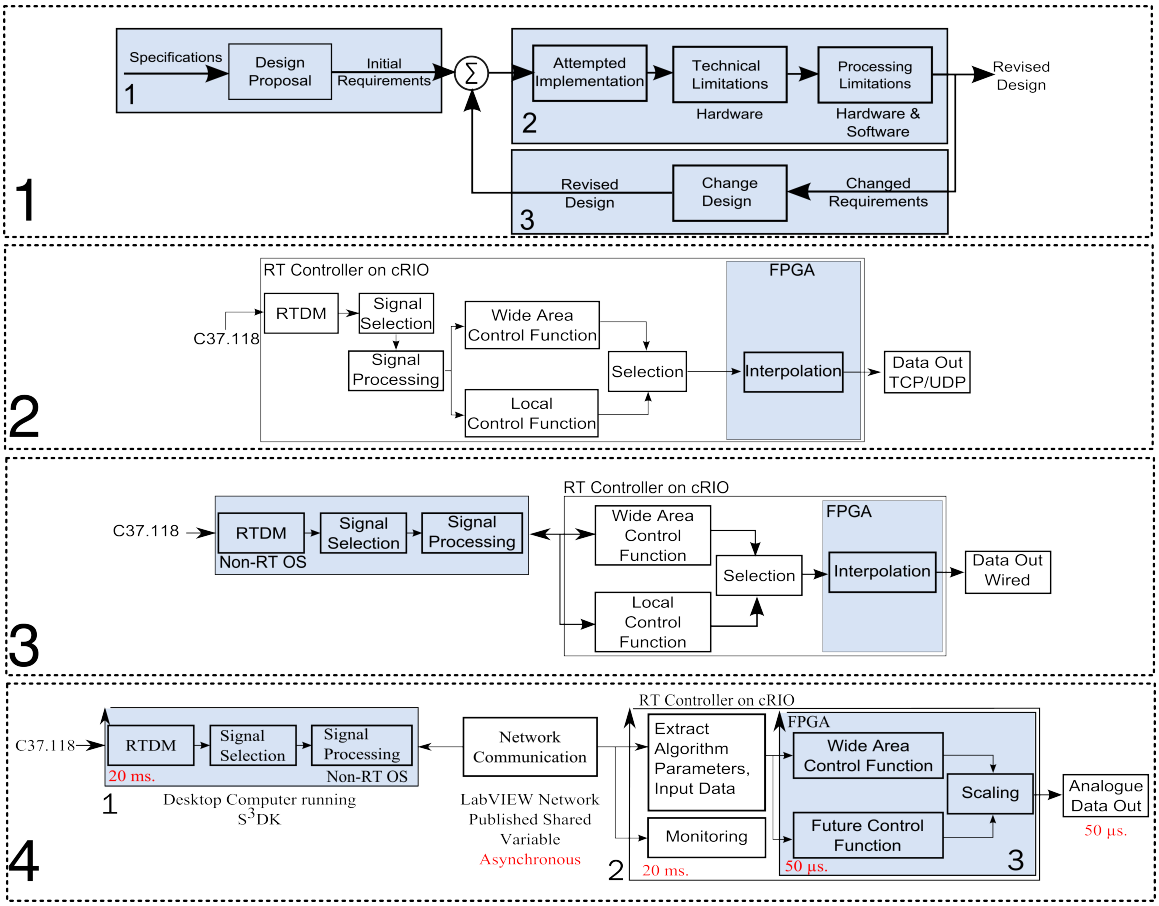

Fig. 2. 1. Three-stage architecture refinement process. 2. Initial architecture realised 3. First architecture refinement 4. Final architecture as implemented