

A Resilient Real-Time Agent-Based System for a Reconfigurable Power Grid

Hairong Qi, *Senior Member, IEEE*, Wenjuan Zhang, *Student Member, IEEE*,
Leon M. Tolbert, *Senior Member, IEEE*¹

Abstract — Reliable electricity has become an essential underpinning for national security in modern society. The fault-tolerant generic framework proposed herein can prevent potential outages from happening through intelligent agent coordination. Instead of limiting the system to manage existing devices, the developed system is adaptive to the future power grid in years to come. This paper proposes a hardware-in-the-loop multiagent system with embedded models and functions.

Index Terms — multi-agent system, real-time response, reconfigurable power grid, collaboration, fault resiliency

I. INTRODUCTION

The electric power industry in the 21st Century has seen dramatic changes in both its physical infrastructure and its control and communication/information infrastructure. As a result, a shift will take place from a relatively few large, concentrated generation centers and the transmission of electricity over mostly a high voltage ac grid to a more diverse and dispersed generation infrastructure that includes renewable or sustainable energy sources [1][2].

The existing energy management system (EMS) consists of three components, the system control and data acquisition (SCADA) system, the state estimator (SE), and the contingency analysis (CA). SCADA systems serve as both data gathering systems as well as device control systems. Data are collected from generation plants and substations through field remote terminal units (RTUs) and then fed into master stations integrated in the control room of each control area. The SE is used in the control room to improve the accuracy of the raw sampled data by mathematically processing raw data to make it consistent with the electrical system model. The resulting information for equipment voltages and loadings is used in software tools such as CA to simulate various conditions and outages to evaluate the reliability of the power system.

However, there exist serious defects in both system-level management and device-level design and maintenance. The massive power blackout of August 14, 2003 was a wakeup

call for both the industry and academia to re-think techniques that help assure the reliability of the power grid [3].

At the system level, the control area operators lack the capability to 1) obtain *real-time status information* of the equipment, 2) adequately assess the situation and predict trends *before* the fact; 3) respond *rapidly enough* (within milliseconds) once events start to unravel; and 4) perform *coordinated actions* in real time across the region. It is astonishing to realize, after the fact, in what an *information vacuum* and *outdated power delivery grid* the controllers have been operating. At the device level, the traditional hardware lacks the capability to 1) provide *reactive power*, frequency control, and voltage control according to system needs and correspondingly, 2) rapidly *reconfigure* the system to a secure state through disconnect switches, circuit breakers, and power-electronics based devices.

The Electric Power Research Institute (EPRI) estimates that the annual economic losses from outages and other power quality and reliability events to be 1% of GDP, or \$100 billion per year [4]. The vulnerability of the power grid during the August 14 blackout calls for a re-design of the power grid at both the information technology (IT) infrastructure level and the device level. Specifically, the reformed power grid should satisfy the following five requirements:

First, consider the *real time requirement*. Some major factors that affect the implementation of real-time management include the large amount of raw data, the complex mathematical modeling, as well as the time-consuming contingency analysis algorithms. In order to achieve real-time operations, intelligence has to be pushed toward the device level such that only pre-filtered data and local decisions need to be processed at the higher control level. In addition, efficient, lightweight, and on-line data analysis and decision making algorithms are also desired.

Second, consider the *reactivity requirement*. The incorporation of Flexible AC Transmission Systems (FACTS) devices with appropriate sensing and control functions can transform the electric grid into a reconfigurable power system where actions can be taken in *microseconds* to control the flow of power and ensure high levels of reliability and quality in the power system.

Third, consider the *proactivity requirement*. The “proactive requirement” is more difficult to satisfy than the “contingency requirement” in that it requires the system to be able to predict the problem *before* it actually occurs. On-line

¹H. Qi, W. Zhang, and L. M. Tolbert are with the Department of Electrical and Computer Engineering, The University of Tennessee, Knoxville, TN 37996-2100 USA, E-mail: hqi@utk.edu, wzhang5@utk.edu, tolbert@utk.edu.

prediction would pose another challenge to the reformed IT infrastructure.

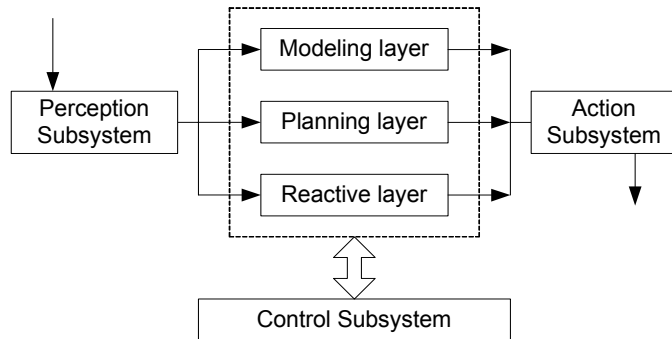
Fourth, consider the *collaborative processing requirement*. Information exchange among hundreds or even thousands of distributed generation centers instead of just among a handful of large utilities is a new challenge brought by deregulation.

Finally, consider the *fault resiliency requirement*. The decision making algorithms need to be able to tolerate the missing, incomplete, or faulty data and still be able to reach a correct decision. We refer to this property as *resiliency*.

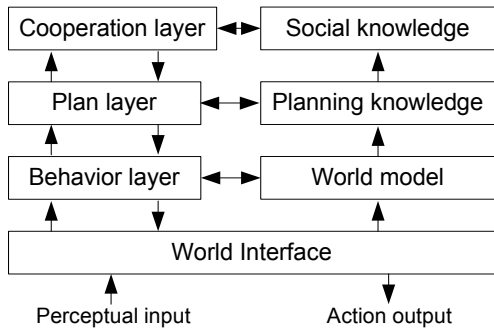
This paper describes the design of a hardware-in-the-loop multi-agent system for resilient, real-time control of the reconfigurable power grid. The approach is divided into three main tasks that are described in the following sections, including the design of multi-agent system architecture, the development of an agent-based reconfigurable power electronic device, and fault-tolerant collaboration.

II. THE MULTIAGENT SYSTEM ARCHITECTURE

There has been an accelerating trend in the integration of computing and communication in critical infrastructure systems as exemplified in the power grid. As these systems become more and more complex, they have to be broken down into ‘weakly coupled’ modules, or independent units with limited interactions, which can be fully controlled [5].



(a) TouringMachines: horizontal layering



(b) InteRRaP: vertical layering

Fig. 1. Layered agent architecture design: (a) horizontal layering vs. (b) vertical layering.

This is the basic philosophy behind the design of object-oriented languages. In multiagent based systems, the emphasis is on the interaction between the agents as well as the flexibility (intelligence) carried by the agents. The metaphor of local agents negotiating their way to reach an equilibrium state is a very natural and attractive one that simplifies the understanding of the principles of the system. Because of the autonomous nature of agents, they enable the utility to do continuous online analysis with every load in the entire system.

A multiagent system is composed of multiple intelligent agents interacting with each other through communication networks in order to satisfy their design objectives. Each intelligent agent is capable of three actions, including autonomously responding to the environment (autonomy and reactivity), taking initiative steps toward goal implementation (proactivity), and interaction with other agents (social ability). The development of a multiagent system includes both system-level designs and individual agent-level designs. In the following two subsections, we will address each in detail.

A. Hybrid Layered Architecture Design for Intelligent Agent

There have been mainly four categories of agent architectures: logic-based, reactive, belief-desire-intention, and layered [6]. Due to the nature of the power grid in which both reactive and proactive behaviors are desired, a layered architecture becomes an appropriate design option. There are two types of control flow within the layered architectures, the horizontal layering (e.g. TouringMachines [7]) and the vertical layering (e.g. InteRRaP [8]), as illustrated in Fig. 1. The horizontal layering architecture needs a mediator (or controller) to decide at each moment which layer has the control of the agent (or which function the agent should perform). The introduction of this central control system inevitably generates a bottleneck to the agent’s decision making process. The vertical layering architecture resembles the way that the organizations work in which information flows up the architecture and control then flows back down.

By studying the current power grid organization, we propose to design three types of intelligent agents, the reliability coordinator agent (RCA), the utility agent (UA), and the reconfigurable device agent (RDA), as illustrated in Fig. 2. The RCAs model the behavior of reliability controllers and the functions performed at each control center. The RDAs are embedded into each FACTS device to facilitate the selection of which control function to execute in real time. The UAs model the behavior of most field RTUs and serve mainly as the interface between the utility (e.g. load, generator, compensator, and transmission line) and RCAs.

Comparing the design architecture between RCA [Fig. 2(a)] and InteRRaP [Fig. 1(b)], we observe some unique characteristics associated with the RCA design: 1) The RCA architecture adds one layer for trend prediction. 2) The RCA design merges the behavior and cooperation layers into a

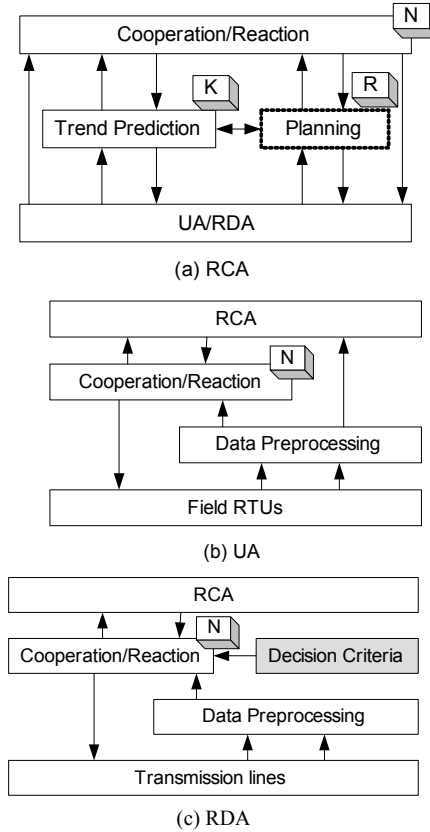


Fig. 2. The hybrid layered architecture design for intelligent agents within the power grid.

single layer, cooperation/reaction layer. An agent can make a decision on its own (reaction); it may also rely on other agents' input to achieve its goal (cooperation), the difference being if the agent has *enough valid* information to make a decision. In Sec. III, we propose a generic interval integration algorithm which can help the agent determine if collaboration with neighboring agents is necessary in order to improve the reliability of the decision. 3) The interconnections between layers are not exactly vertical neither are they horizontal. The inputs reported from UAs and RDAs are fed directly to the three layers presented instead of going through a "situation estimator" for data refinement. Most of the pre-processing job is performed at the UA or RDA level to alleviate RCA's raw data processing burden. Each layer, in turn, exchanges findings between each after processing the input information. We name this kind of connection-rich architecture as a *hybrid-layered agent design*. 4) The planning layer is marked with a dotted boundary to indicate that this paper does not consider the redesign of the planning layer, we leave a flexible interface for ease of future integration of the planning layer.

Similar to the InteRRaP design, each layer of the RCA is associated with a database, including the neighbor information base (N) attached to the coordination layer, the knowledge base (K) attached to the trend prediction layer,

and the rule and regulation base (R) attached to the planning layer.

The structure of UAs mainly consists of two layers, the data preprocessing layer and the coordination/reaction layer. The data preprocessing layer is responsible for data refinement. The coordination layer performs localized collaboration with neighbors (See Sec. II.B). Both the refined data and the coordinated decision are sent to the RCA for further analysis if needed. RCA's feedback is returned to the UA through the coordination layer, which in turn controls the field RTUs.

The reconfigurable device agent (RDA), as shown in Fig. 2(c), is an agent embedded in the device (or hardware-in-the-loop agent design) to help implement a reconfigurable-grid concept and power electronic compensators that can provide an array of services such as reactive power generation, power flow control [9], harmonic compensation, voltage regulation, or dynamic control over the frequency and voltage [10][11]. The incorporation of FACTS devices with appropriate sensing and control functions can transform the electric grid into a reconfigurable power system. In addition, DG equipment that can implement any FACTS functions or ancillary services such as reactive power supply, frequency control, power flow control, voltage control, etc., provide a great opportunity to reconfigure the system.

However, the traditional FACTS devices and DG equipment are function dependent and locally controlled. The reason is two-fold: one is isolated installation cases because of limited penetration and the other is lack of means for coordination and management. As far as this reason is concerned, we propose that converters be integrated with generation resources, loads, or act as a stand-alone system. We will develop a large set of *decision criteria* for the agents such that they can make control decisions that will ultimately improve the power quality and reliability of the electric grid according to system needs for survivability, security, damage control, and optimization. The difference between the UAs and RDAs mainly comes from the addition of the "decision criteria" database. Because of the higher-level intelligence of the RDA, only the actions taken by the RDA are fed into the RCA to ensure real-time control and situation awareness.

B. Location-Centric Hybrid Multiagent Architecture

Wesson *et al.* were among the first to propose the design of network structures for distributed computing. Two structures were analyzed in their initial work [12]: the anarchic committee (AC) structure (Fig. 3a) and the dynamic hierarchical cone (DHC) structure (Fig. 3b). AC can be viewed as a fully interconnected network without hierarchy, where each node can communicate with any other node, thus coordination between nodes is straightforward. Although easy for communication, AC structure is expensive to implement and also hard to extend. On the other hand, DHC provides a hierarchical structure, also called a tree structure.

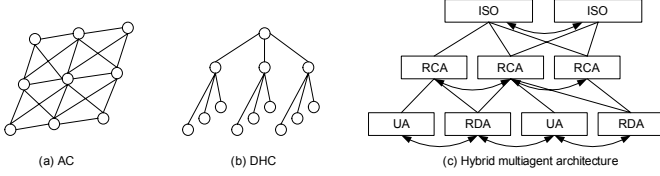


Fig. 3. Location-centric hybrid multiagent architecture.

It only allows communications between nodes in adjacent layers, but not within the same layer. Compared to AC, DHC is easier to extend, but is more vulnerable since a faulty node can disconnect an entire subtree.

Considering the nature of the power grid, we propose a hybrid multiagent structure with three layers, as shown in Fig. 3(c). At the bottom layer, UAs and RDAs are responsible for collecting and preprocessing data from substations and generation plants. At this layer, only neighboring agents need to communicate with each other such that simple collaborative work can be achieved without going through the upper layer. We use the “one-hop” rule to define neighbors, that is, neighbors should be directly connected by transmission line or power electronics interface. The middle layer is where the RCAs reside. In general, each control room (or each control area) should have one RCA. At this layer, neighboring RCAs communicate with each other if they are geographically adjacent. The top layer is managed by the Independent System Operator (ISO).

This hybrid multiagent system design, as shown in Fig. 3(c), includes both centralized control and peer-to-peer location-centric communications. This architecture facilitates the system development in satisfaction of the five design requirements stated in Sec. I. More data processing and analysis tasks are pushed to the bottom layer of the architecture such that real-time response can be achieved in an efficient manner and large amount of raw data transmission is avoided over the communication link, which in turn reduces the burden of RCA. If the bottom layer coordination cannot reach a reliable solution (See Sec. I), decisions relayed from the upper layer will be expected and performed.

III. FAULT RESILIENCY THROUGH ROBUST COLLABORATIVE INTERVAL INTEGRATION

Here, we develop a generic algorithm to achieve fault resiliency based on multiagent coordination even through potentially faulty, incomplete, and missing information.

A. Interval-based Integration

An important concept in the derivation of the fault-resilient algorithm is the *interval-based* integration as compared to the *value-based* integration. The value-based integration performs the coordination algorithm based on inputs from devices which, without loss of generality, we assume to be concrete numbers (be integers or floating point numbers).

The interval-based integration performs the coordination algorithm based on an interval clustered around the physical readout (or the number). We design the preprocessing layer in the RCAs, RDAs, and UAs to behave as an *abstract device*. We define an *abstract device* as a device that reads a physical parameter and gives out an abstract interval estimate which is a bounded and connected subset of the real number.

Based on this definition, a *correct device* is an abstract device whose interval estimate contains the actual value of the parameter being measured. Otherwise, it is a faulty device. A faulty device is *tamely faulty* if its interval estimate overlaps with the actual value being measured, and is *wildly faulty* if there is no overlap between its estimate and the actual measurement.

For example, UA might receive a reading from a power line, which is 25% overloaded. The preprocessing layer of the UA can then output an interval of [20%, 30%] modeled by a Gaussian, in which we interpret the interval as, “The power line is 20% to 30% overloaded.” The *estimate* itself can be modeled by different stochastic distributions, the simplest of which would be a uniform distribution, where equal weight has been put on each value within the estimate range. Other appropriate distributions could be a Gaussian (more weight on the central value within the estimate range) or a Rayleigh (more weight on the lower values within the estimate range).

B. Distributed Interval Integration Algorithm

Let ζ_i represent an estimate distribution provided from UA i over an estimate range $[a_i, b_i]$. In order to integrate the estimate range distribution ζ_i from different agents, we develop a distributed interval integration algorithm [13][14]. The original centralized algorithm was proposed by Prasad, Iyengar, and Rao in 1994 [15], in which a control center collects the outputs of the devices and constructs an *overlap* function $\Omega(x) = \sum_{i=1}^n \zeta_i(x)$, where n is the number of hardware devices. Fig. 4 illustrates the construction of an overlap function for a set of 6 devices. If we assume that the correct

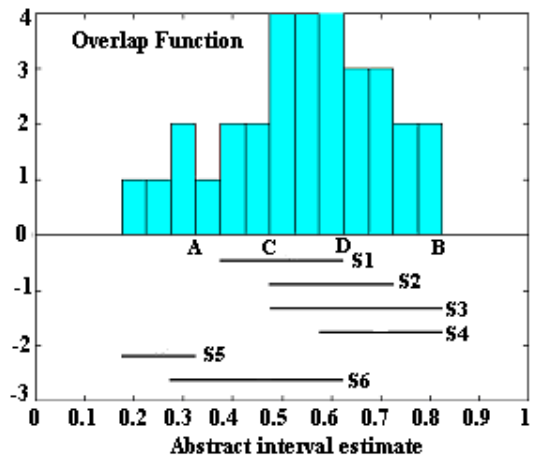


Fig. 4. The overlap function constructed from six devices.

interval estimate is $[0.5, 0.6]$, we then observe that: the tamely faulty devices (s_4) cluster around correct devices (s_1, s_2, s_3, s_6) and create high and wide (maximal) peaks in the profile of $\Omega(x)$ while wildly faulty devices (s_5) have little or no overlap with correct devices, and therefore contribute to smaller and narrower peaks. A one-dimensional array serves as an appropriate data structure to store this result. The size of the array depends upon the resolution used. In the example, the resolution is 0.05.

The original algorithm picks a “crest” from the overlap function and resolves only the crest in the next finer resolution level. The process will continue until the finest resolution is reached. Taking the overlap function in Fig. 4 as an example, the crest picked at the current resolution (0.05) would be $[A, B]$. In [15], the authors show that the algorithm is robust and satisfies a Lipschitz condition [16], which ensures that minor changes in the input intervals cause only minor changes in the integrated result.

Cho *et al.* [17] improves the original algorithm to only return the interval with the overlap function ranges $[n-f, n]$ where f is the number of faulty device inputs. Once again, take Fig. 4 as an example, where among the $n = 6$ devices, there are $f = 2$ faulty devices. Thus the final integration result using Cho’s approach will be $[C, D]$ where the overlap function ranges $[4, 6]$. This algorithm also satisfies Lipschitz condition, and its main advantage is that it is able to reduce the width of the output interval in most cases and produce a narrower output interval when the number of devices involved is large.

However, the problem with Cho’s approach is that the value of f is normally unknown. We make the following modifications in order to improve the reliability of the system to faulty inputs. We use $c = h \times w \times acc$ to pick the “crest”, where h is the height of the highest peak in the overlap function, w is the width of the peak, and acc is the estimate at the center of the peak. The peak with the largest c is selected as the crest. For example, in Fig. 4, the crest selected is the rectangle between C and D . The height of the crest is $h = 4$, the width is $w = 3$, and the central value is $acc = 0.55$, therefore, $c = 6.6$. Based on the change of parameter c obtained at different stages of integration, the agent can determine if the integration process can be terminated.

We design a *protocol* for decision making which concerns both the degree of fault tolerance and the accuracy achieved. According to the Byzantine generals problem, the maximum number of faults (f) that a certain amount of devices (n) can tolerate is

$$f = \left\lfloor \frac{n-1}{3} \right\rfloor \quad (1)$$

We define the degree of fault tolerance df as the ratio between f and $f+n$, that is, $df = f/(f+n)$. As the agent interacts with other peer agents, the maximum number of faulty inputs that can be tolerated will change, so does the integration

result $[n-f, n]$, but the degree of fault tolerance maintained is the same as that of the Byzantine generals problem. The protocol says that if and only if the following three criteria are satisfied then a decision can be rendered; otherwise, the agent has to continue coordination with its neighbors to obtain more information, that is, the agent’s decision is not reliable enough to be trusted:

1. The overlap function has its highest peaks ranging from $[n-f, n]$, where f is calculated from Eq. (1).
2. The center value of the pick (acc) has to be equal to or larger than the median of the estimated interval. For example, if the estimated interval is $[0, 1]$, then the acc cannot be less than 0.5.
3. Both 1 and 2 have to be satisfied in two adjacent integrations excluding the first agent in order to add stability to the decision.

C. Case Study

We provide an application example to show how this protocol is applied [18]. Assume a neighborhood of four UAs participate in a collaborative decision making process in order to determine which power line needs Var compensation the most. The three intervals associated with each agent in Table 1 indicate the estimate range of that agent *thinking* how much Var that power line needs. For example, UA1 thinks power line #1 needs 1000-2900 kVar. In this example, UA1 provides a tamely faulty result. Fig. 5 illustrates how agent collaboration generates the partially integrated estimate range when the integration is progressively performed from UA1 to UA4. We assume the resolution requirement is 500 kVar. Table 2 summarizes the agents’ decision making procedure after integration with each of the four neighbors. The highlighted number in Table 2 indicates the largest c , which can show us the integrated decision of several agents. We observe that the integration result at UA1 and UA1+UA2 shows that “line #2” needs Var the most, but changes to “line #3” when integrating with inputs from UA3 and UA4.

How does an agent determine when it can stop the integration process? When three agents coordinate with each other (Fig. 5(c)), i.e., device number $n = 3$, we get $f = 0$ from Eq. (1) and the highest peak range is $[3,3]$. In the three graphs of Fig. 5(c), we can see the highest peak exists in the leftmost graph, which is 3, so the first criterion is satisfied; the “acc” of Line#3 in Fig. 5(c) is 0.525, which is larger than 0.5, hence the second criterion is also satisfied. With device number $n = 4$, i.e., there are four agents coordinating with each other, we get $f = 1$ and the highest peak range is $[3,4]$. We observe that the highest peak in the three graphs of Fig. 5 (d) is 4; and the “acc” of Line#3 is 0.55, so both criterion 1 and 2 are satisfied. Because the first two criteria are satisfied in two adjacent integrations of “UA1+2+3” and “UA1+2+3+4” (criterion 3), we can conclude the agent’s decision is reliable enough, we can stop the integration even though more agents’ decision are available.

TABLE 1. Initial estimate range distribution made by four neighboring agents.

UAs	Line#1(*10 ⁴ kVar)	Line#2(*10 ⁴ kVar)	Line#3(*10 ⁴ kVar)
1	[0.10,0.29]	[0.45,0.65]	[0.10,0.21]
2	[0.05,0.14]	[0.05,0.41]	[0.22,0.58]
3	[0.05,0.15]	[0.05,0.15]	[0.49,0.59]
4	[0.08,0.16]	[0.08,0.16]	[0.51,0.60]

TABLE 2. Collaborative decision between agents (unit: 10⁴kVar).

	UA1		UA1+2		UA1+2+3		UA1+2+3+4	
	<i>c</i>	<i>acc</i>	<i>c</i>	<i>acc</i>	<i>c</i>	<i>acc</i>	<i>c</i>	<i>acc</i>
Line#1	0.8	0.2	0.5	0.125	0.375	0.125	0.5	0.125
Line#2	2.2	0.55	4.2	0.35	0.4	0.1	0.6	0.1
Line#3	0.525	0.175	0.45	0.225	3.15	0.525	3.3	0.55

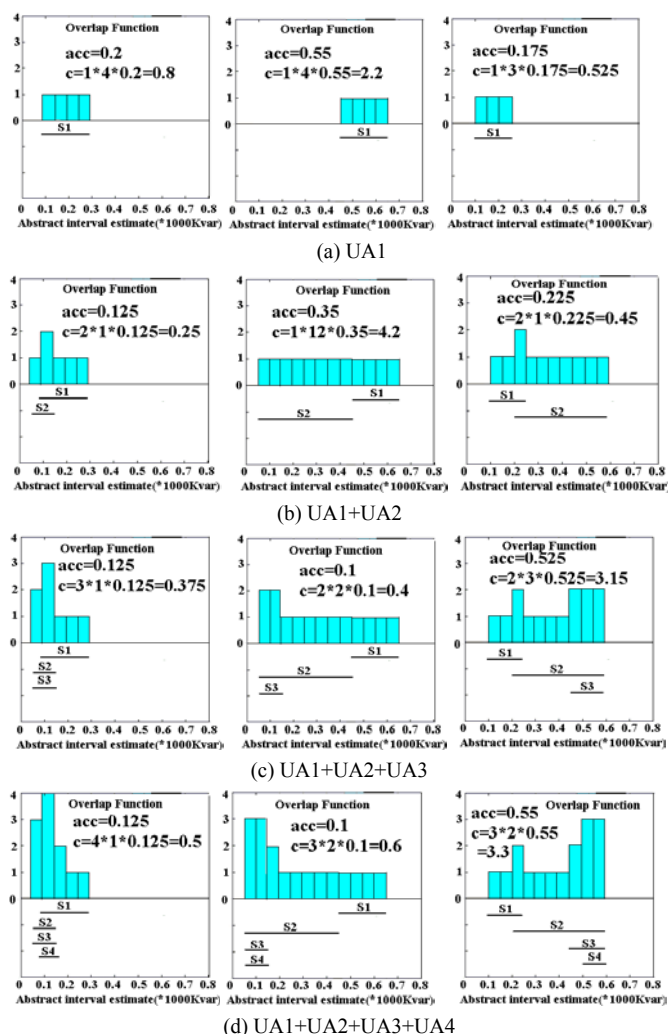


Fig. 5. Multiagent interval integration results with a progressively improving reliability (From left to right: Line#1, Line#2, Line#3).

IV. CONCLUSIONS

Power electronics based grid interface systems have the ability for reactive power generation/compensation, power

flow control, harmonic compensation, voltage regulation, dynamic control over the frequency and voltage output, and real-time control of distributed resources.

Multi-agent systems provide a mechanism to facilitate collaboration among individuals over the grid. We propose a hardware-in-the-loop agent system that can provide the power grid with new and improved capabilities and fill in the gap from previous research on agent-based systems, and solve several unanswered fundamental questions, including the challenge of real-time response (within milliseconds), the integration of future reconfigurable devices, and decision making based on incomplete, missing, or even faulty information.

REFERENCES

- [1] J. Contreras, F. F. Wu, "Coalition formation in transmission expansion planning," *IEEE Trans on Power Systems*, vol. 14, no. 3, pp. 1144-1152, August 1999.
- [2] M. Meinhardt, G. Cramer, "Past, present, and future of grid-connected photovoltaic- and hybrid-power-systems," *Conference Record – IEEE Power Engineering Society Summer Meeting*, July 16-20, 2000, Seattle, Washington, pp. 1283-1288.
- [3] Symmetricom, *White Paper: How time finally caught up with the power grid*, IT Research, <http://www.bitpipe.com>, January 1, 2004.
- [4] EPRI, "Estimating the costs of power disturbances," *EPRI Journal Online*, July 15, 2003.
- [5] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999.
- [6] M. Wooldridge, "Chapter 1: Intelligent Agents," from *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*. Editor: G. Weiss. The MIT Press, 1999.
- [7] I. A. Ferguson, *Turing Machines: An Architecture for Dynamic, Rational, Mobile Agents*. Ph.D. Thesis, Clare Hall, University of Cambridge, UK, November 1992.
- [8] J. Muller, "A cooperation model for autonomous agents," In *Intelligent Agents III*, LNAI vol. 1193, pages 245-260. Editors, J. Muller, M. Wooldridge, N. R. Jennings. Springer-Verlag: Berlin, Germany, 1997.
- [9] J. Wang, F. Z. Peng, "A novel configuration of unified power flow controller," *IEEE APEC*, pp. 919-924, February 2003.
- [10] F. Z. Peng, J. S. Lai, "Dynamic performance and control of a static var generator using multilevel inverters," *IEEE Trans. Industry Applications*, 33(3): 748-755, May/June 1997.
- [11] L. M. Tolbert, F. Z. Peng, "Multilevel converters as a utility interface for renewable energy Systems," *IEEE Power Engineering Society Summer Meeting*, 2000, Seattle, Washington, pp. 1271-1274.
- [12] R. Wesson, F. Hayes-Roth, J. W. Burge, C. Stasz, C. A. Sunshine, "Network structures for distributed situation assessment," *IEEE Trans. Syst., Man, Cybern*, SMC-11(1): 5-23, January 1981.
- [13] H. Qi, Y. Xu, X. Wang, "Mobile-agent-based collaborative signal and information processing in sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1172-1183, August 2003.
- [14] H. Qi, S. S. Iyengar, K. Chakrabarty, "Multi-resolution data integration using mobile agents in distributed sensor networks," *IEEE Trans. Syst, Man and Cybern (Part C)*, vol. 31, pp. 383-391, August 2001.
- [15] L. Prasad, S. S. Iyengar, R. L. Rao, "Fault-tolerant sensor integration using multiresolution decomposition," *Physical Review E*, 49(4): 3452-3461, April 1994.
- [16] L. Lamport, "Synchronizing time servers," *Digital System Research Center*, Technical Report 18, 1987.
- [17] E. Cho, S. S. Iyengar, K. Chakrabarty, H. Qi, "A new fault tolerant sensor integration function satisfying local Lipschitz condition," Submitted to *IEEE Trans. Aerosp. Electron. Syst.*, 2000
- [18] W. Zhang, L. M. Tolbert, "Survey of reactive power planning methods," *IEEE Power Engineering Society General Meeting*, June 12-16, 2005, San Francisco, California, pp. 1580-1590.