Intelligent Control and Asset Management: An Event-based Control Road Map

James H. Taylor
University of New Brunswick
Fredericton, New Brunswick, CANADA, E3B 5A3
Email: jtaylor@unb.ca

***Abstract*—My students and I have participated in a seven-year effort to produce an advanced supervisory control system called ICAM, a system for Intelligent Control and Asset Management, with specific focus on petroleum industry applications. By design, however, ICAM was devised to be widely applicable in a variety of automation and manufacturing arenas.**

We adopted a multi-agent architecture to implement ICAM, and divided ICAM's functionality among the various agents to execute different specific tasks. Most agents were, by nature, computationally intensive; these were implemented as MATLAB® routines. The Supervisor (or "master agent"), on the other hand, was meant to incorporate the "intelligence" of ICAM, so it was created in the G2 expert system shell. We also assumed that a wireless sensor and actuator network (WSAN) would be incorporated in the system under control, including links comprising control loops.

Several important lessons were learned in the course of this project. First, we found it easier and more efficient to distribute much of the intelligence in the agents rather than the Supervisor, and secondly, that implementing the Supervisor as an expert system in G2 made the operation of ICAM very slow and cumbersome, due to the excessive overhead involved. We have concluded that replacing the expert system Supervisor with a top-level event-based controller will make ICAM considerably more effective, efficient, and easier to extend and maintain.

I. INTRODUCTION

Comprehensive asset management and control of a modern process facility can involve many tasks with different time-scales and levels of complexity, including but not limited to: signal processing (gross error detection and correction, plus filtering or data reconciliation); fault detection, isolation, and accommodation; process model identification; and supervisory control. In addition, wireless sensor and actuator network coordination was addressed toward the end of the project, in light of the burgeoning use of wireless links in process control loops. The automation of these complementary and intertwined tasks within an information and control infrastructure promises to reduce maintenance expenses, improve utilization of equipment, enhance safety, and improve production and product quality.

As mentioned, we compartmentalized the functionality of ICAM by adopting a multi-agent architecture. The tasks outlined above were specifically implemented as follows: a fault detection, isolation and accommodation (FDIA) agent [1], a linear model identification (LMId) agent, also [1], an adaptive nonlinear dynamic data reconciliation (ANDDR) agent [2], [3], a wireless network control coordinator (WNCC) [4] and, as lower-level assistants, a steady-state detection (SSD) agent [3] and a rudimentary database manager. A high-level schematic of ICAM is provided in Fig. I; as noted, all agents were implemented in MATLAB [5] except the Supervisor [6], which was built using the expert system shell G2 [7]. The process simulator, also implemented in MATLAB, represents a two-tank, three-phase crude oil separator with five PID control loops built into the process simulator [8]).

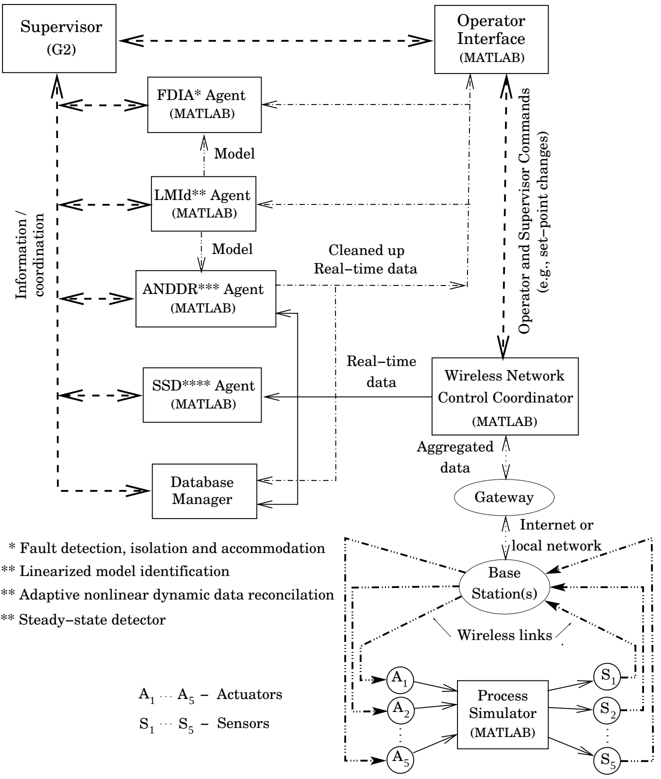

Fig. 1. Schematic showing ICAM's architecture

We anticipated deploying the ICAM system in a real-world application setting, with various components hosted on multiple platforms; at a minimum, the Supervisor on one machine, the MATLAB-based agents on another and the WSAN Gateway on a third host. This necessitated the development of rigorous interprocess communication protocols (see Section VI), which added substantial complexity and overhead to ICAM at run time. The communication between the expert system working in G2 and the MATLAB-based agents was particularly slow and burdensome, and in hindsight unnecessary [9]; this has motivated our recent decision to replace the expert system-based Supervisor with a simpler event-based control solution.

The remainder of this paper will include a brief outline of the primary agent's control-theoretic and logical function, to demonstrate that most of ICAM's knowledge and decision-making was in fact embedded in the MATLAB agents. This exposition is followed by a summary of problems encountered and "lessons learned", and a discussion of plans to reimplement the Supervisor as an event-based master controller.

II. LINEAR MODEL IDENTIFICATION

Industrial processes are generally high-order and nonlinear in nature, making it difficult and impractical to derive or even use an accurate mathematical model for the system if one exists. This has been one of the main reasons that has limited the application of quantitative model-based techniques such as nonlinear dynamic data reconciliation (see Section III) – in fact, this is what prevented us from using our highly realistic nonlinear model for the crude oil separator for NDDR. Despite having the model, it executed too slowly for use in a simulation-based optimization procedure. Also, many such methods are only applicable to linearized models, such as our fault detection, isolation and accommodation (FDIA) technique (see Section IV). Therefore, developing a linear model identification (LMId) agent was essential.

Linear model identification involves exciting the process and collecting input/output signals from it to infer the process dynamics. We excited the process using generalized binary noise (GBN), generated according to a fixed sample interval T_0 and amplitude a; at each sample time t_k the GBN switches from a to $-a$ or *vice versa* with probability p_{sw} [10]. Thus GBN signals are easy to design and generate, and, since they can possess good low-frequency content if T_0 and p_{sw} are suitably chosen, we found that they gave more consistent results overall compared with using pseudo-random binary sequences. Furthermore, being of limited amplitude, we could choose a large enough to achieve adequate excitation yet not drive the process too strongly. We used MATLAB's prediction error/maximum likelihood method which gave excellent results, in terms of fitting percentage and matching output waveforms, as demonstrated in [1], [11].

This agent was designed and implemented so that it could operate completely autonomously, with no need for external direction from the Supervisor or operator. That philosophy was successfully applied to all of ICAM's MATLAB-based agents.

III. SIGNAL PROCESSING

Data from process sensors are typically corrupted by noise (usually high-frequency random signals, e.g., those modeled by first-order Markov processes) and gross errors (e.g., data dropouts, spikes, analog-to-digital or digital-to-analog conversion errors). Low-pass filtering is often used for noise reduction, but this alters the spectrum of the filtered process signals and thus introduces additional dynamics that may negatively impact LMId and other activities. An alternative technique that is often used in the chemical process control context is called nonlinear dynamic data reconciliation (NDDR). This technique, pioneered by Liebman et al. [12], was extended by Laylabadi and Moreno [13], [14] to make NDDR adaptive (hence ANDDR) and more self-sufficient. Also, Laylabadi and Taylor created and demonstrated a new heuristic approach for Gross Error Detection and Correction (GEDC) [15] that is also self-adjusting – it estimates the random noise levels to help in distinguishing gross errors from random variation. ANDDR and GEDC are complicated procedures which were completely automated – space limits do not permit an intelligible summary. Both ANDDR and GEDC proved to be very effective.

IV. FAULT HANDLING

In real-world industrial processes, such as oil and gas facilities, continuous production is required to achieve good asset utilization and profitability. As a result, stopping production suddenly in the middle of operations to fix or replace a sensor or actuator that has failed unexpectedly may result in significant economic losses. To minimize the impact of interruptions in plant operation, sensor fault detection and isolation (FDI) are essential, and accommodation (FDIA), if possible, is even more beneficial, to provide a temporary solution to sustain safe operation while maintenance can be scheduled without significantly upsetting the process or production. FDI for actuator failures is equally important; even though accommodation may not be possible, an immediate alarm for the specific failed actuator is imperative for quick repair or replacement. Both should be integrated as part of an effective asset management strategy.

We implemented FDIA using a powerful technique based on generating and interpreting directional residuals (generalized parity vectors) [1], [11], [16]. The residuals are very small when all sensors and actuators are working normally, but become large in magnitude in the presence of a fault. Moreover, the residuals "point" in different directions in the parity vector space, depending on which component has failed, and, in the case of sensor faults, the magnitude of the parity vector can be used to infer fault size, e.g., "temperature sensor 3 is reading 15 C low", so the control system can add that offset to the measured signal and continue normal operation. This directional residual method was demonstrated to be very effective.

The procedure for disambiguating the various residual directions for each possible sensor and actuator fault required the development of an optimization method to calculate a transformation matrix that can separate them as much as possible. This must be done every time the LMId agent has to be invoked in response to a significant change in set point. In addition, the threshold for declaring a fault also involves on-line calculation. The FDIA agent performs these tasks and all others autonomously.

V. WIRELESS NETWORK CONTROL COORDINATOR

Finally, we devised and implemented a wireless network control coordinator (WNCC) to solve a serious problem: if wireless paths are to be incorporated safely in process control loops one must not allow control signals to suffer slow data rates and variable time delays, both of which may degrade the performance of the control loop or even lead to instability. Those time delays, inherent to WSANs, stem from network delay and data latency, and depend on the network configuration and loading. In addition, the sampling rate also has a great impact on the performance and stability of the closed-loop control systems, as mentioned, yet the data rate should be minimized to conserve WSAN node battery life and accommodate other wireless communications traffic. These conflicting requirements must be reconciled as much as possible; this was the goal of creating the WNCC [4], [17], [18].

The WSAN Gateway (see Figure I) has the responsibility of generating proposed network configurations, when the system is started and after every relevant event, such as a node outage or unbalanced network traffic. The WNCC then (1) checks the proposed configuration to determine the time delay which the control data packets will encounter, and rejects any proposed configuration that would lead to poor closed-loop system performance; and (2) determines the minimum acceptable sampled data rate that does not degrade control loop performance excessively. This latter task is done by comparing the performance of the control system with wireless links over the proposed configuration with that of an ideal hard-wired control system and insisting that the behavior of the former is not excessively degraded (according to a specified metric, e.g., percent overshoot). This minimum acceptable data rate is mandatory whenever the control loops are closed; they may be open if the process is in steady state, in which case the WSAN can work at a slower rate, as long as the process can be monitored to detect disturbances or significant process drift so loops can be closed and an appropriate sampling rate reinstated. This involves carefully coordinated interchanges among the the WNCC, the WSAN Gateway and the Supervisor (which controls the opening and closing of control loops); this must be done at start-up and every time the WSAN reconfigures or its performance changes, e.g., due to interference or a node outage.

We demonstrated and validated the WNCC by applying it to a jacketed continuous stirred-tank reactor model, a significant third-order nonlinear model which has two control loops, in a number of realistic scenarios, including data drop-outs, node outages, heavy WSAN traffic, opening and closing control loops as needed, and accommodating unusual procedures such as LMId. This is not the same as field testing, but it demonstrated excellent coordination in all cases.

VI. SUPERVISORY CONTROL

A Supervisor prototype was developed to monitor and control the activities of the other agents overviewed previously and run the process effectively [19], [20]. It was designed to interact with the agents realistically, taking into account that the Supervisor is implemented in G2 [7], and it and the agents would be distributed over a number of platforms. This required a three-layered architecture, with (1) reactive agents (e.g., FDIA, LMId, etc.); (2) middleware using the remote memory access (RMA) communication approach, which is part of the message passing interface (MPI) communication library, to address data communications between itself and other layers; and (3) the artificial intelligence layer (i.e., the expert system-based Supervisor). The active target RMA communication type was chosen to achieve high reliability, i.e., data are moved rigorously from the memory of one process to the memory of another, and both are explicitly involved in the communication [21]. Further detail may be found in [6].

The Supervisor was implemented in G2, wherein the ICAM system internal and external behavior was to be codified in its knowledge base [7]. The rule base was meant to capture the desired system behavior in response to external events and environment dynamic changes and to process operator interactions. Therefore, we felt that it was crucial to carefully design the rule-base of the supervisory agent to achieve overall robust system performance.

However, as the MATLAB-based agents were developed with sophisticated logic to deal with most anticipated ICAM tasks there came to be little left for the Supervisor to do other than run test scenarios [9]. Therefore, the Supervisor design and implementation process was suspended in its preliminary stage after developing the extensive infrastructure mentioned; it was, in fact, quite rudimentary, only running simple scenarios to test interactions between it and various ICAM agents and among the reactive agents themselves. Given that the MATLAB-based agents incorporate substantial local intelligence including inter-agent communication and coordination; that the interaction between the G2-based Supervisor and MATLAB-based agents is so slow and cumbersome; and that the rule-based capabilities of G2 are so under-utilized; it makes sense to cease its development at this time.

VII. AN EVENT-BASED SOLUTION

Based on our critique of the present ICAM implementation, it seems wise to seek an alternative to the G2-based Supervisor. Considering the obvious importance of “events” in the operation of ICAM, e.g., occasionally needing to perform linear model identification after the operating set-point has changed significantly, dealing with sporadic gross errors in data transmission, performing FDIA upon occurrence of a failure, and handling all the random events in the WSAN (interference, congestion, variable time delays, node outages and WSAN reconfiguration) it is compelling to seek an event-based solution. In fact, the WNCC (see Section V) was implemented with an event-based control idea, namely, when the process was in steady state the WNCC informed the Supervisor that it could open the loops to ease congestion in the WSAN, and when the process started to deviate from steady state the WNCC told the Supervisor that it must close the loops. We are particularly motivated to adopt this strategy for the entire ICAM system by the land-mark publication by Miśkowicz [22], which presents many approaches and successful applications.

We plan to retain the two-level hierarchical control scheme used before: An Event-based Supervisor (EBS) will send set points wirelessly to local PID control loops implemented in their own processors located at the process convenient to the corresponding sensor-actuator pair. One innovation is that the EBS will generate a progression of several future PID set points, using Model Predictive Control (MPC) [23] to define the short-term strategy for maneuvering the process controlled variables; this should give the local PID controllers a degree of robustness should the WSAN delay or drop the next PID set point. (We assume (reasonably, given the capabilities of modern microcontrollers, digital signal-processing chips and field programmable gate arrays) that the local PID controller modules will have this capability.)

Furthermore, the contention between the WSAN requirements and limitations (minimizing data rates, random delays and drop-outs) and the control loops (maintain performance and avoid instability with regular and sufficiently fast sampling) will be addressed by adopting the event-based control strategy described by Årzén [24] and, in more generality, in Heemels *et al* [25]. In our context, event-based sampling is commonly used in the process industry (e.g., when statistical process control is used); a new control action is only calculated when a predefined deviation has occurred. The resulting two-level control system with MPC and event-based PID control should be more agile, and the overall system much less cumbersome.

Finally, to test our plan and approach we will implement a prototype event-based ICAM system solely in MATLAB, as a proof of concept. If, as we anticipate, the prototype successfully overcomes the difficulties we encountered in the first implementation, these ideas may be useful as a road-map for other similar industrial automation and control projects.

VIII. SUMMARY AND CONCLUSIONS

The ICAM system developed in the first phase of our effort was very comprehensive, and was designed to automate all the functionality required for asset management in a broad context. One general principle emerged rather early in our work: knowledge about algorithms and numerical methods should be incapsulated in the MATLAB agents, while knowledge about situation assessment and asset management could be embedded in the Supervisor. Although the use an expert system Supervisor made good sense methodologically we found that it was impractical when implementing the system in a real-world context, with various ICAM components hosted on different machines, due to the excessive overhead involved.

Very briefly, much was accomplished in the first phase of the ICAM project, and much learned. We came to realize that developing "smart agents" in MATLAB is effective in many ways: testing, debugging, extension and refinement were easy, and execution was much faster than could be accomplished with the Supervisor managing the process. We now believe that the ICAM architecture is sound, and that reimplementing the Supervisor as an event-based controller is the best way forward.

REFERENCES

[1] M. Omana, "Fault detection, isolation and accommodation using the generalized parity vector technique," Ph.D. dissertation, University of New Brunswick, 2009.

[2] M. Laylabadi, "Adaptive nonlinear dynamic data reconciliation and gross error detection," Master's thesis, University of New Brunswick, 2006.

[3] R. P. Moreno, "Steady state detection, data reconciliation, and gross error detection," Master's thesis, University of New Brunswick, 2010.

[4] H. M. S. Ibrahim, "Wireless sensor network management agent," Master's thesis, University of New Brunswick, 2009.

[5] The MathWorks, "User's guide for MATLAB," 1990-2004, Natick, MA 01760.

[6] A. F. Sayda, "Intelligent control and asset management of oil and gas production facilities," Ph.D. dissertation, University of New Brunswick, 2008.

[7] GenSym Corporation, "G2 for application developers reference manual, 8th ed." 2005, Burlington, MA 01760.

[8] A. F. Sayda and J. H. Taylor, "Modeling and control of three-phase gravity separators in oil production facilities," in *Proc. American Control Conference*, New York, NY, 2007.

[9] J. H. Taylor, "Intelligent control and asset management: A retrospective," in *Proc. The 25th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Wolverhampton, West Midlands, England, June 2015.

[10] J. A. F. Tulleken, "Generalized binary noise test-signal concept for improved identification-experiment design," *Automatica*, vol. 26, no. 1, 1990.

[11] J. H. Taylor and M. Omana, "Fault detection, isolation and accommodation using the generalized parity vector technique," in *Proc. IFAC World Congress*, Seoul, Korea, July 2008.

[12] M. J. Liebman, T. F. Edgar, and L. S. Lasdon, "Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques," *Computers & Chemical Engineering*, vol. 16, no. 10/11, 1992.

[13] J. H. Taylor and M. Laylabadi, "A novel adaptive nonlinear dynamic data reconciliation and gross error detection method," in *Proc. IEEE Conference on Control Applications*, Munich, Germany, October 2006.

[14] J. H. Taylor and R. P. Moreno, "Nonlinear dynamic data reconciliation: In-depth case study," in *Proc. IEEE Multiconference on Systems and Control*, Hyderabad, India, August 2013.

[15] M. Laylabadi and J. H. Taylor, "ANDDR with novel gross error detection and smart tracking system," in *Proc. 12th IFAC Symposium on Information Control Problems in Manufacturing*, Saint-Etienne, France, May 2006.

[16] N. Viswanadham, J. H. Taylor, and E. C. Luce, "A frequency domain approach to failure detection and isolation with application to GE21 turbine engine control system," *Control Theory and Advanced Technology*, vol. 3, no. 1, pp. 45–72, 1987.

[17] J. H. Taylor and H. S. Ibrahim, "A new, practical approach to maintaining an efficient yet acceptably-performing wireless networked control system," in *Proc. IEEE International Conference of System Science and Engineering*, Taipei, Taiwan, July 2010.

[18] J. H. Taylor, J. Åkerberg, H. M. S. Ibrahim, and M. Gidlund, "Safe and secure wireless networked control systems," in *Proc. IEEE Multiconference on Systems and Control*, Dubrovnik, Croatia, October 2012.

[19] J. H. Taylor and A. F. Sayda, "Prototype design of a multi-agent system for integrated control and asset management of petroleum production facilities," in *Proc. American Control Conference*, Seattle, Washington, June 2008.

[20] A. F. Sayda and J. H. Taylor, "An implementation plan for integrated control and asset management of petroleum production facilities," in *Proc. IEEE Conference on Control Applications*, Munich, Germany, 2006.

[21] ——, "Toward a practical multi-agent system for integrated control and asset management of petroleum production facilities," in *Proc. IEEE International Symposium on Intelligent Control (ISIC)*, Singapore, October 2007.

[22] M. Miśkowicz, *Event-based Control and Signal Processing*. CRC Press / Taylor & Francis Group, 2016.

[23] E. F. Camacho and C. Bordons, *Model Predictive Control*. London, UK: Springer-Verlag London, Ltd, 2004.

[24] K.-E. Årzén, "A simple event-based PID controller," in *Proc. 14th IFAC World Congress*, Beijing, P. R. China, July 1999.

[25] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. 51st IEEE Conference on Decision and Control*, Maui, Hawaii, USA, December 2012.