A Multi-agent System for Integrated Control and Asset Management of Petroleum Production Facilities - Part 2: Prototype Design Verification

Atalla F. Sayda and James H. Taylor

Abstract— **This three-part paper thoroughly addresses the design and development of multi-agent system for asset management for the petroleum industry, which is crucial for profitable oil and gas facilities operations and maintenance. A research project was initiated to study the feasibility of an intelligent asset management system. Having proposed a conceptual model, architecture, and implementation plan for such a system in previous work [1], [2], [3], defined its autonomy, communications, and artificial intelligence (AI) requirements [4], [5], and initiated the preliminary design of a simple system prototype [?], we are extending the build of a system prototype and simulate it in real-time to validate its logical behavior in normal and abnormal process situations and analyze its performance. The second-part paper addresses the ICAM system prototype design verification and its logical behavior during sensor faults in the plant.**

I. INTRODUCTION

As part of the ICAM system prototype development, the second-part paper addresses the prototype verification aspect in real time. This validates the design decisions and the system requirements upon which the system was designed [1], [9], [2], [10], [11], [3], [12], [13], [14], [15], [4], [16], [5], [17], [**?**]. Real-time simulation experiment was designed to analyze the performance of the ICAM system prototype in terms of its logical behavior and its response to the external environment dynamics. The ICAM system prototype is deployed in a Windows 2003 network, which has two nodes (i.e., workstations). The first node has three running agents, namely the pilot plant agent, the model ID agent, and the supervisory agent. The second node has the remaining agents, namely the statistical preprocessing agent and the FDIA agent. The pilot plant simulation model corresponds to figure 1; it consists of 10 states, 5 manipulated variables, 5 controlled variables, and 17 auxiliary measured inputs and outputs (e.g., disturbances, product quality variables, etc.). Ten sensor/actuator faults are embedded in the pilot plant simulation agent to emulate faulty instrumentation in real-world oil production plants, as indicated in table I.

In the simulation scenario (i.e., indicated by $\boxed{1}$ in figure 1), we will apply a bias fault in the three-phase separator water volume sensor, as described in section 2. We will discuss the behavior of each agent of the system in terms of its results and decisions in sections 3, 4 and 5. Furthermore, we will discuss the decisions made by the supervisory agent in section 6. The network activity will be also discussed in section 7 to see if it is consistent with the decisions made by the supervisory and the reactive agents of the system.

James H. Taylor is with the Department of Electrical & Computer Engineering, University of New Brunswick, PO Box 4400, Fredericton, NB CANADA E3B 5A3 jtaylor@unb.ca

Atalla F. Sayda is a PhD candidate with the Department of Electrical & Computer Engineering, University of New Brunswick, PO Box 4400, Fredericton, NB CANADA E3B 5A3 atalla.sayda@unb.ca

Fault number	Instrumentation name
F1	Faulty two-phase liquid volume sensor
F2	Faulty two-phase pressure sensor
F3	Faulty three-phase water volume sensor
F4	Faulty three-phase oil volume sensor
F5	Faulty three-phase pressure sensor
F6	Faulty two-phase separator liquid outflow valve
F7	Faulty two-phase separator gas outflow valve
F8	Faulty three-phase separator water outflow valve
F9	Faulty three-phase separator oil outflow valve
F10	Faulty three-phase separator gas outflow valve

TABLE I

OIL PRODUCTION FACILITY INSTRUMENTATION FAULTS

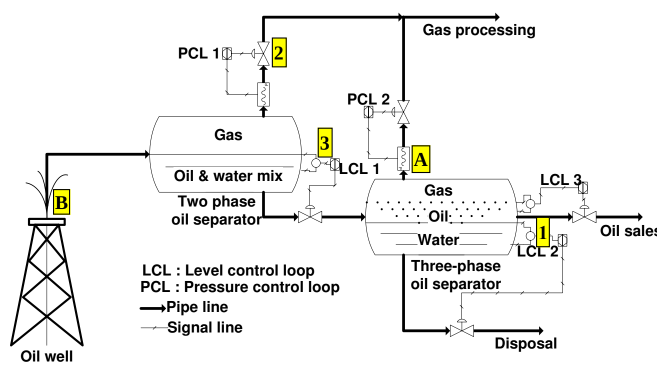

Fig. 1. Oil production facility P&ID

II. SIMULATION SCENARIO: FAULTY WATER VOLUME SENSOR IN THE THREE-PHASE SEPARATOR SUB-PROCESS

The simulation scenario is done by applying a $+15\%$ bias fault in the water volume sensor of the three-phase separator (F3; refer to control loop LCL2 in figure 1). After the ICAM system supervisory agent starts up executing its rule-base, other reactive agents are started and initialized. The pilot plant agent starts its simulation at a nominal value of V $=$ 146 ft^3, P $=$ 625 PSI for the two-phase separation sub-process and V$_{wat}$ $=$ 77.5 ft^3, V$_{oil}$ $=$ 46.5 ft^3, P $=$ 200 PSI for the three-phase separation sub-process. Outliers and missing data are applied to the two-phase separator measurements to emulate real-world data. Since the ICAM system has no knowledge about the pilot plant agent (i.e., no dynamic model), it sends a message to the statistical pre-processor to check if the pilot plant is in steady state.

Once it is in steady state, the supervisor then commands the control system of the pilot plant to apply a sufficiently exciting pseudo random binary (PRBS) signal with an amplitude of 2% about the nominal operating point. This allows the model ID agent to collect enough data to identify the pilot plant model, after which the FDIA agent designs its FDI filter. Having gained new knowledge about the current dynamic behavior of the pilot plant, the ICAM system now can start monitoring the pilot plant for any instrumentation failure. If a sensor/actuator fault occurs the FDIA agent reports its decision to the supervisory agent. The supervisory agent in turn commands the FDIA agent to start the fault accommodation task, if applicable. The fault accommodation task is stopped if the sensor is fixed. The behavior of each agent during this scenario is discussed in the following sections.

III. The Pilot Plant Agent Behavior

The process variables are logged at the pilot plant agent during the first simulation scenario as indicated by figures 2, 3, 4, 5, and 6. Positive fixed-size outliers and missing data are applied to the two-phase separator measurements at random time instants (i.e., the liquid volume and the pressure measurement as shown by the top plots of figures 2, and 3). The pilot plant at first runs at its nominal operating point. Independent PRBS signals are applied to all the plant inputs to identify its model. Subsequently, a $+15\%$ bias fault is applied to the three-phase separator water volume sensor at time T_{fault} = 9:47:32, and the accommodation task starts at time T_{accom} = 9:48:44, as shown in figure 4. The PI controller in loop LCL2 (refer to figure 1) rejects the fault, as it is considered as a constant disturbance applied to the water volume sensor. However, the sensor measurement does not reflect the actual state of the water volume, as shown in the FDIA agent results. The effect of the faulty volume sensor on the oil volume and the gas pressure in the three-phase separator is also shown in figures 5, and 6.

IV. Behaviors of the Statistical Preprocessing and Model ID Agents

Raw data is received by the statistical pre-processing agent, which removes any outliers and corrects missing data by replacing it with the previous data value, as demonstrated by the clean two-phase separator liquid volume and pressure data record in figures 7 and 8. The statistical agent first checks if the pilot plant is in steady state to prevent applying the PRBS signal in a transient state. Apparently the pilot plant takes a time period of $T_{SS} = 37.204\,s$ to reach steady state due to the plant small initial conditions, as shown in figures 7 and 8. Processed data is sent to the model ID agent during the PRBS signal application, after which a new process model can be estimated.

Figure 9 shows measured plant outputs along with their simulated counterparts using the newly identified plant model. Each process variable data record has a length of 300

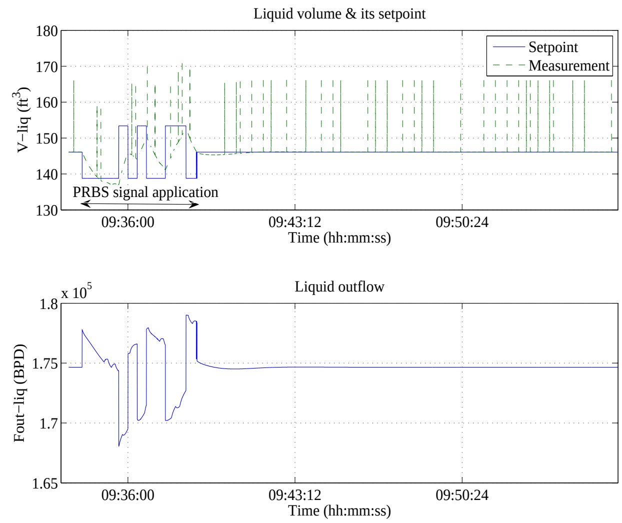

Fig. 2. Scenario 1: Two-phase separator liquid volume logged by the pilot plant agent

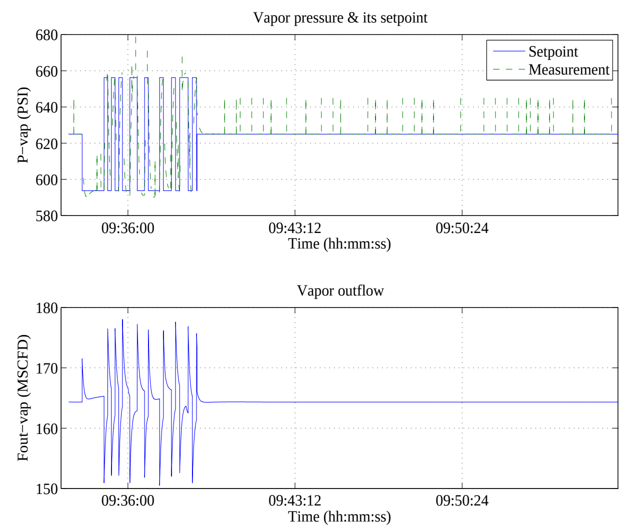

Fig. 3. Scenario 1: Two-phase separator pressure logged by the pilot plant agent

seconds, which was the pre-specified PRBS signal application time. It is interesting to notice that although missing data has been corrected, yet they still affect the identified model quality, as indicated by the two-phase separator pressure data record (refer to the second plot in figure 9 with a model fit of 66%). Figure 10 shows the plant inputs during the PRBS signal application task.

V. The FDIA Agent Behavior

Once the new process model is received by the FDIA agent, then it can design its FDI filter and deploy it to diagnose faulty plant instrumentation. Figure 11 shows the three-phase water volume data record collected after the FDI filter is deployed. When the water volume sensor fault occurs, its effect can be noticed not only in the local control

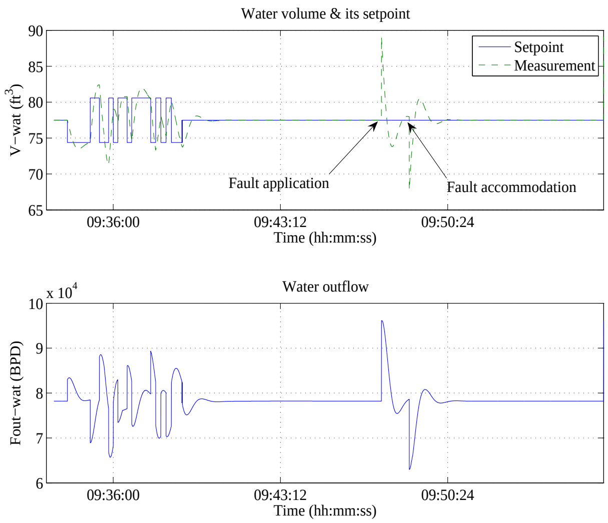

Fig. 4. Scenario 1: Three-phase separator water volume logged by the pilot plant agent

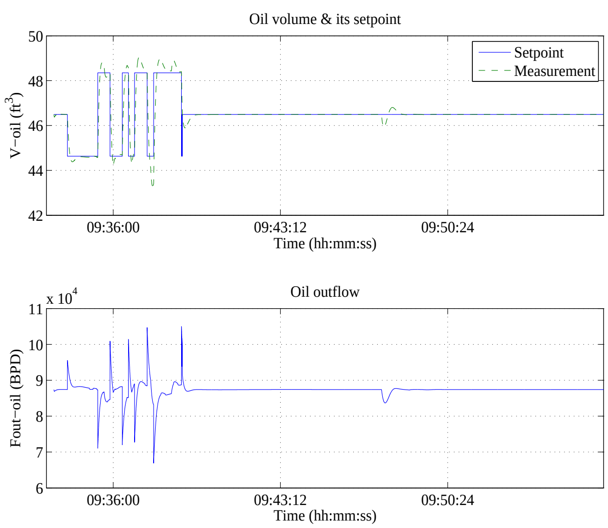

Fig. 5. Scenario 1: Three-phase separator oil volume logged by the pilot plant agent

loop of the faulty instrumentation but also downstream, which is seen as disturbance in the three-phase oil volume control loop, as shown in figure 5. Figure 11 shows that the actual process variable has a different response from its corresponding measurement, i.e., the $+15\%$ error starts to drive the actual water volume to a lower setpoint in an effort to make the sensed setpoint approach the desired value; once the fault is accommodated the actual water volume returns to the correct setpoint.

The FDIA agent generates a general parity vector whose abnormal magnitude can detect faulty instrumentation, and generates the angles between the parity vector and the reference directions of the process variables. When there is a fault, then the smallest angle indicates the approximate alignment of the parity vector with the reference direction

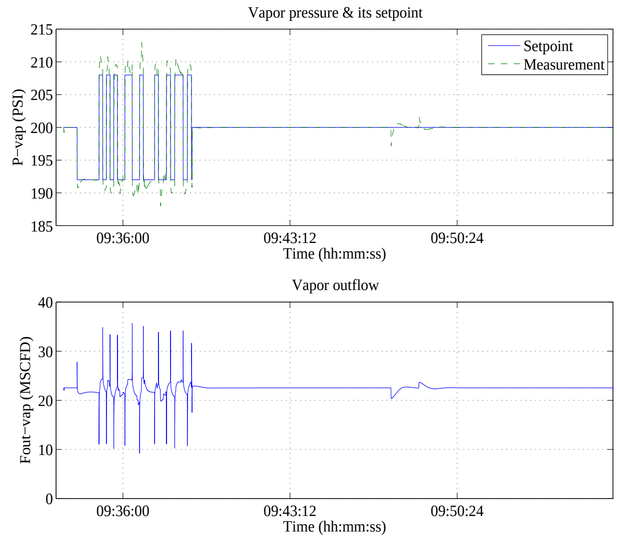

Fig. 6. Scenario 1: Three-phase separator pressure logged by the pilot plant agent

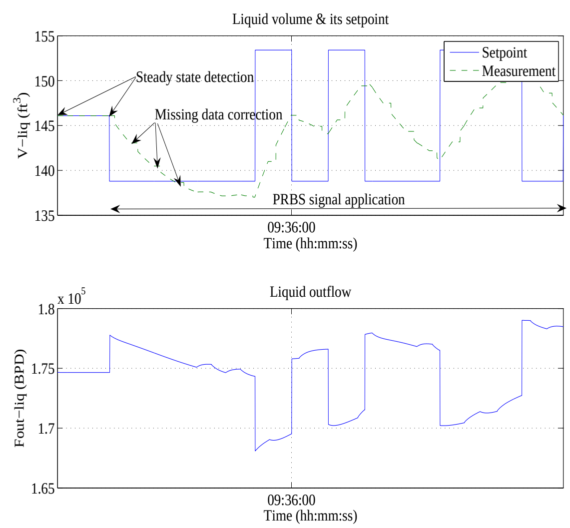

Fig. 7. Scenario 1: Two-phase separator liquid volume logged by the statistical pre-processing agent

of a specific instrumentation fault. Hence the fault can be isolated based on the smallest angle after the fault detection. It is clear from the top plot in figure 12 (produced by FDI routines documented in [18], [9], [13], [19], [16], [17]) that the general parity vector (GPV) magnitude increased significantly, which indicates that a fault occurred. Furthermore, the smallest angle after the fault detection instant is the one that corresponds to the water volume sensor in the three-phase separator, as indicated by the dash-dotted trace in the middle plot of figure 12. The other GPV angles are higher than the faulty volume sensor GPV angles, as indicated by the other traces in the middle and bottom plots of figure 12.

The parity vector-based FDI angles are highly sensitive to process variable changes when there is no fault. This is because of the small size of the GPV vector in no-fault

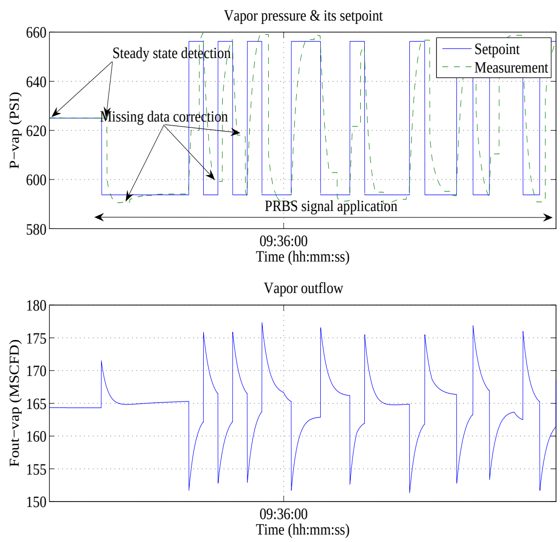

Fig. 8. Scenario 1: Two-phase separator pressure logged by the statistical pre-processing agent

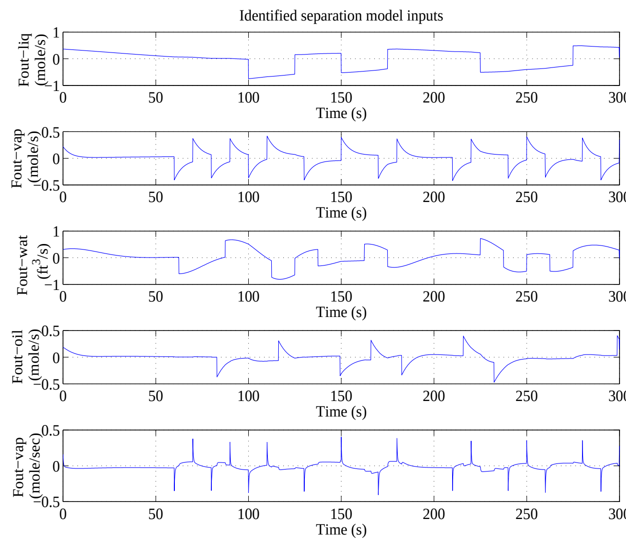

Fig. 10. Scenario 1: Plant inputs logged at the model ID agent

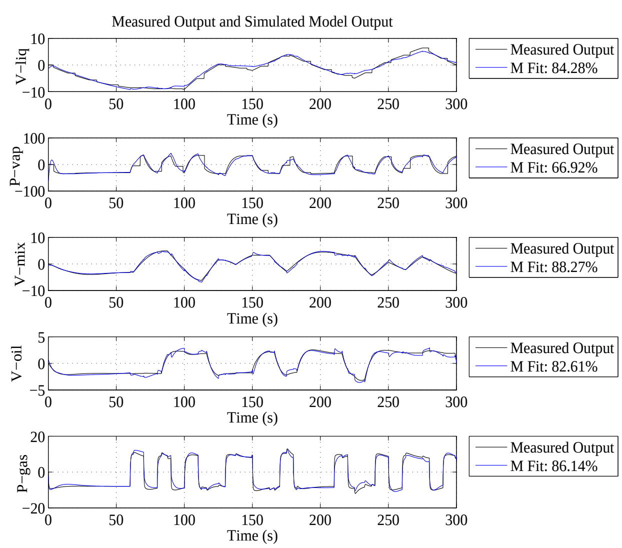

Fig. 9. Scenario 1: Measured plant outputs and simulated model outputs logged by the model ID agent

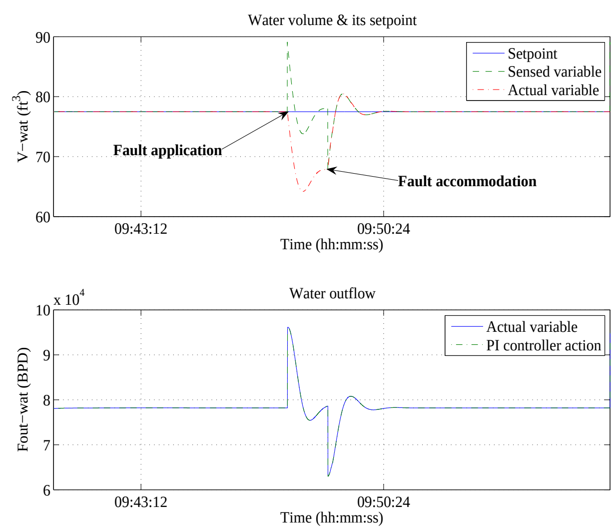

Fig. 11. Scenario 1: Three-phase separator water volume logged by the FDIA agent

situation, which can change its angle widely even in case of very small process variable changes, as indicated by the large variation of the GPV angles before fault occurrence in figure 12. The local decision making logic of the FDIA agent ignores the angles until a large GPV magnitude signals fault detection, then it isolates the fault after its occurrence as demonstrated in the FDI GUI [17], as shown in figure 13 . It interesting to notice a fault # of -1 occurred at the beginning of fault isolation task (-1 indicates an unknown fault). The FDIA agent isolates faults when the process variables have reached an acceptable steady state level, so isolation is ineffective during the transient part of the fault dynamics. As soon as the supervisory agent receives the fault information from the FDIA agent, including the fact that it is a sensor fault, it alerts the FDIA agent to start the fault accommodation task. The FDIA agent then estimates the fault size, which is used to accommodate the fault (correct the sensor reading).

Once the fault has been accommodated the actual water volume process variable returns to its nominal setpoint, which matches its corresponding corrected measurement, as indicated by figures 11 and 12. The FDIA agent logic then indicates a no-fault situation during the fault accommodation task, as indicated by figure 13. Figure 14 shows the accommodation parameters in terms of the estimated fault size and the recursive fault size estimation error, which is only effective during faults of ramp type. The estimated fault size is $+15\%$, which matches the original fault size value.

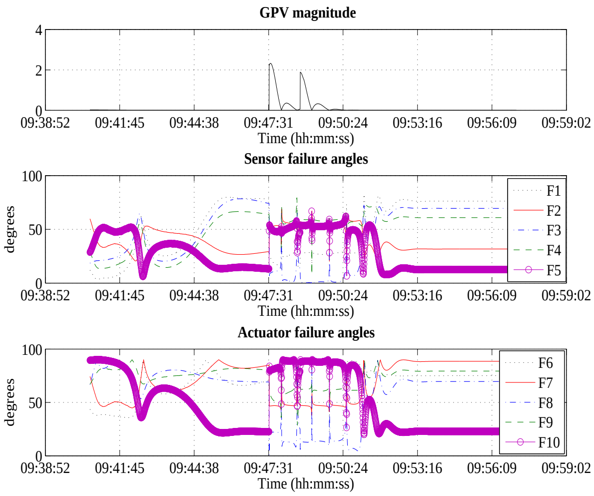

Fig. 12. Scenario 1: FDIA agent diagnostic signals

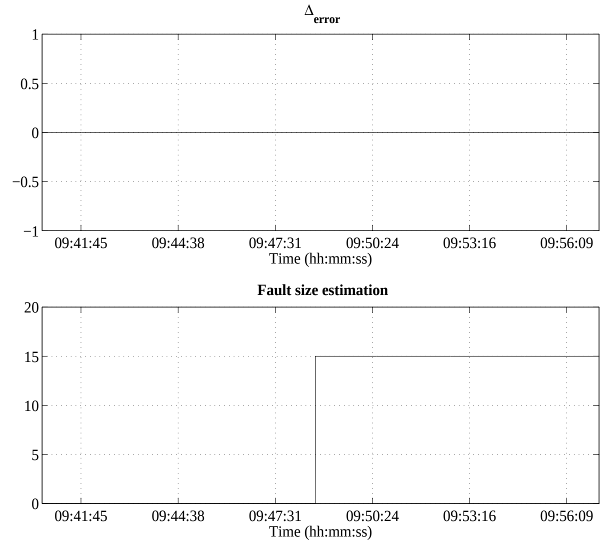

Fig. 14. Scenario 1: FDIA agent fault accommodation parameters

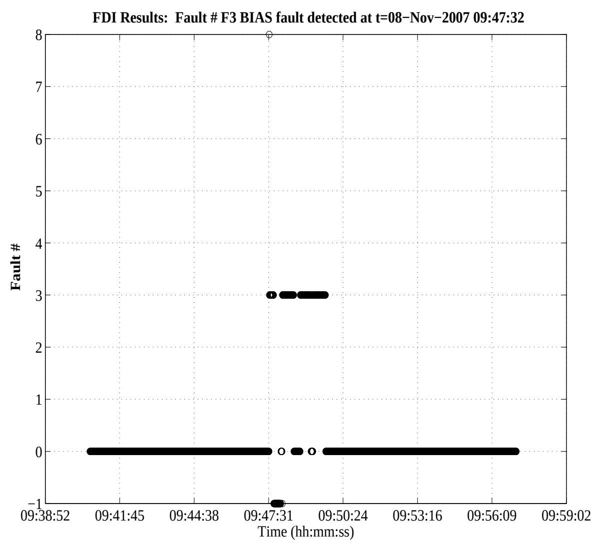

Fig. 13. Scenario 1: FDIA agent fault display

VI. The Supervisory Agent Behavior

The supervisory agent monitors the state of the reactive agents and reasons about their current state according to its knowledge base. Each reactive agent is represented by an object with a set of attributes that represents its own state. Table II demonstrates the pilot plant supervisory frame during the fault accommodation task. The pilot plant agent frame shares some common attributes with the other reactive agents, which represent the agent's internal state, MPI, and G2 communication channels' states. For example, the pilot plant frame is in the simulation state and executing its functionality as indicated by the simulation status attribute. Its MPI and G2 links are connected, and the pilot plant agent has a rank (i.e., the software process number) of 0 in the MPI environment defined by the MPI communicator attribute. The agent's decision attribute indicates that a fault simulation scenario is applied. The decision attribute is the decision made by the supervisory agent depending on the current state of the agent. The MPI channel decision attribute indicates that the accommodated data MPI channel is opened, through which the pilot plant agent receives the accommodation parameters from the FDIA agent.

Names	ENV-OBJECT
Rank	0
State	simulate
Mpi comm	comm-world
Mpi comm size	4
G2 link status	connected
Mpi link status	connected
Decision	fault
Simulation status	on
Mpi channel decision	open-accom-channel
Prbs status	none

TABLE II
Scenario 1: Pilot plant agent supervisory frame

Tables III and IV show the statistical preprocessing and model ID supervisory frames, which have the same ICAM system common attributes (i.e., rank, G2 link, MPI link status, decision etc.). Both agents have a common model status attribute, which indicate that the two agents have updated their knowledge about the current dynamics of the pilot plant. The statistical preprocessing agent has a steady state detection attribute to indicate if the pilot plant is in a steady or transient state. The decision attributes in these agents' supervisory frames have the value no-decision, which indicates that the supervisory agent does not require these agents to do any task. Likewise, the MPI channel decision attributes of the statistical preprocessing and model ID supervisory frames have a no-decision value, which indicates that the supervisory agent does not require these agents to close any of their MPI data channels.

Names	STAT-OBJECT
Rank	1
State	simulate
Mpi comm	comm-world
Mpi comm size	4
G2 link status	connected
Mpi link status	connected
Decision	no-decision
Simulation status	on
Mpi channel decision	no-decision
Model status	model-is-identified
Steady state	none

TABLE III
SCENARIO 1: STATISTICAL PREPROCESSING AGENT SUPERVISORY FRAME

Names	MODELID-OBJECT
Rank	2
State	simulate
Mpi comm	comm-world
Mpi comm size	4
G2 link status	connected
Mpi link status	connected
Decision	no-decision
Simulation status	on
Mpi channel decision	no-decision
Model status	model-is-identified

TABLE IV
SCENARIO 1: MODEL ID AGENT SUPERVISORY FRAME

The FDIA agent supervisory frame has the same common attributes, which indicate that the agent is in the simulation state and is executing the fault diagnosis task, as shown in table V. It also has attributes about the fault information such as the fault size, sign, type, time, and location. The model status and the FDI design status attributes indicate that the FDIA agent has received the process model and has deployed the designed FDI filter. The FDIA agent supervisory frame has also attributes to represent the accommodation task status and the recursive fault estimation in case of ramp faults. For example, the FDIA agent has reported the fault information back to the supervisor for further processing and actions. In this case the FDIA agent successfully detected, isolated, and identified the faulty instrumentation, which is the three-phase separator water volume sensor (F3; refer to table I). The fault has occurred at time T_{fault} = 9:47:32, which is nearly the exact fault application time. The fault has a type bias with an estimated size of $+15\%$. The fault accommodation task is in progress and the accommodation parameters are sent to the pilot plant agent, as indicated by the MPI channel decision attribute. Since the fault type is bias and not of a ramp type, then recursive fault size estimation is not required as indicated by the corresponding attribute of the table.

Names	FDIA-OBJECT
Rank	3
State	simulate
Mpi comm	comm-world
Mpi comm size	4
G2 link status	connected
Mpi link status	connected
Decision	diagnose-faults
Simulation status	on
Mpi channel decision	open-accom-channel
Model status	model-is-identified
Fdi design status	fdi-filter-is-designed
Fault	f3
Fault sign	plus
Fault size	15.0
Fault type	bias
Fault time	"08-Nov-2007 09:47:32"
Accommodation status	acc-in-progress
Recursive fault size estimation status	none
Acknowledge fault	off

TABLE V
SCENARIO 1: FDIA AGENT SUPERVISORY FRAME

VII. NETWORK ACTIVITY

The ICAM system prototype is deployed in a Windows 2003 network, which has two nodes (i.e., workstations). The first node has the statistical preprocessing agent and the FDIA agent running. The second node has three running agents, namely, the pilot plant agent, the model ID agent, and the supervisory agent, as shown in figure 15. The total communication throughput between the two nodes (indicated by green solid arrows in figure 15) is composed of five channels; one asynchronous supervisory channel (indicated by black dashed arrows in figure 15), and four synchronous MPI data channels. The first MPI data channel is the raw data channel which connects the pilot plant agent with the statistical preprocessing agent (indicated by a green solid arrow). The statistical preprocessing agent transfers the processed data on the second MPI data channel (indicated by dark-blue solid arrows) to the model ID and FDIA agents. Once the plant model is identified, it is transferred to the statistical preprocessing and FDIA agents through the model MPI channel (indicated by magenta dash-double-dotted arrows). Finally the accommodation parameters are transferred from the FDIA agent to the pilot plant agent through the accommodated data MPI channel (indicated by a purple dash-dotted arrow).

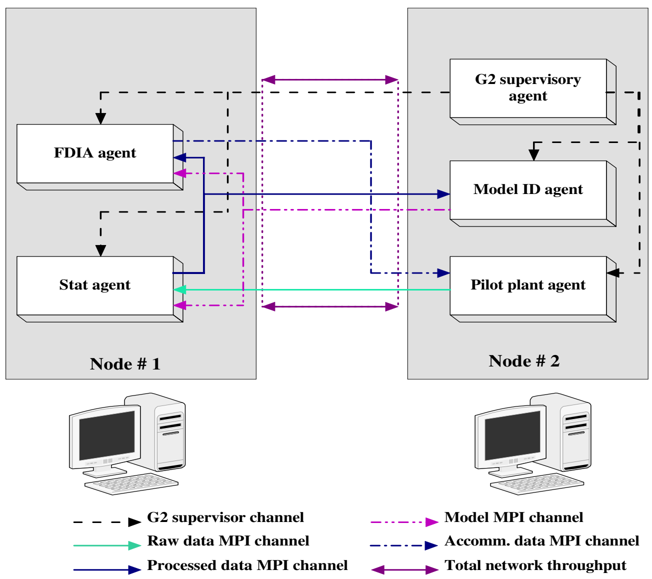

Fig. 15. ICAM system prototype network architecture

Figure 16 depicts the ICAM system prototype network activity during the first simulation scenario. After the ICAM system prototype starts up (event $\boxed{1}$), the raw data and processed data MPI channels start to transfer data at a rate of 330 Kbps (i.e., 0.33% of the 100 Mbps network transfer rate) for each channel, as indicated by the event $\boxed{1}$; note that the green and dark-blue traces have nearly the same rate. The total network throughput is represented by the purple trace. The transfer rates of the MPI channels dip prior to event $\boxed{2}$ because of increasing memory consumption and computations resulted from increasing data storage in some agents (refer to the third part of this paper for more details). The processed data channel (i.e., the dark-blue trace) is closed during the plant model identification task, as indicated by event $\boxed{2}$. Once the plant model is transferred to the corresponding agents, the processed data channel is opened again and the fault diagnosis task is started, as indicated by event $\boxed{3}$. When the three-phase water volume sensor fault is detected and the fault accommodation task is started, the accommodated data channel is opened, as indicated by event $\boxed{4}$; the dark-blue trace is stepped up to its twice rate (i.e., 580 kbps), and the total network transfer rate is at 870 kbps. When the system shuts down at the end of the first scenario, the MPI channels are closed sequentially starting with accommodated data channel, followed by the processed data channel, and finally the raw data channel, as indicated by event $\boxed{5}$.

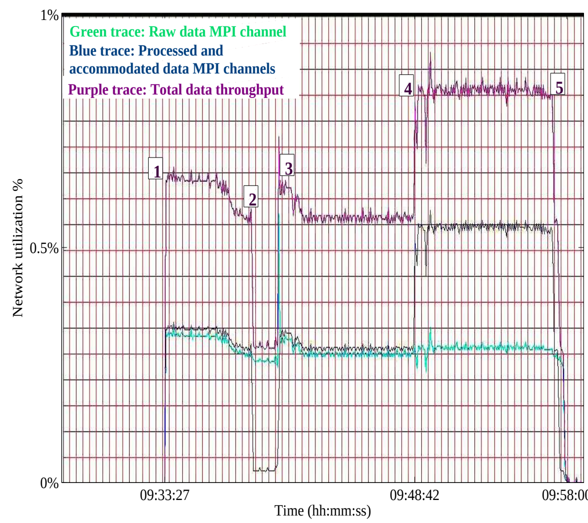

Fig. 16. Scenario 1: ICAM system prototype network activity

VIII. Conclusions

A real-time simulation experiment was conducted to verify the ICAM system prototype design decisions, which were discussed in the first part of this paper. The plant data were collected at each reactive agent, and were examined to analyze the behavior of each reactive agent during the simulation experiment. Furthermore, the decisions taken by the supervisory agent were consistent with system design requirements. A network analysis was also conducted to verify the logical behavior of the middleware of the system. Simulation results revealed that the ICAM system prototype behaved according to the design requirements specified in the first part of this paper. However, a more detailed performance analysis must be conducted to detect the limitations of the designed prototype, which will be discussed in the third part of this paper.

IX. Acknowledgement

This project is supported by Atlantic Canada Opportunities Agency (ACOA) under the Atlantic Innovation Fund (AIF) program. The authors gratefully acknowledge that support and the collaboration of Cape Breton University (CBU), and the College of the North Atlantic (CNA). The authors also acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding the second author's research.

References

[1] J. H. Taylor and A. F. Sayda, "Intelligent information, monitoring, and control technology for industrial process applications," in *The 15th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Bilbao, Spain, July 2005.

[2] ——, "An intelligent architecture for integrated control and asset management for industrial processes," in *Proc. IEEE International Symposium on Intelligent Control (ISIC05)*, Limassol, Cyprus, June 2005, pp. 1397–1404.

[3] A. F. Sayda and J. H. Taylor, "An implementation plan for integrated control and asset management of petroleum production facilities," in *IEEE International Symposium on Intelligent Control ISIC06*. Munich, Germany: IEEE, October 4-6 2006, pp. 1212–1219.

[4] ——, "An intelligent multi agent system for integrated control and asset management of petroleum production facilities," in *In Proc. of The 17th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Philadelphia, USA, 18-20 June 2007, pp. 851–858.

[5] ——, "Toward a practical multi-agent system for integrated control and asset management of petroleum production facilities," in *IEEE International Symposium on Intelligent Control (ISIC)*, Singapore, 1–3 October 2007.

[6] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis part 1, 2, 3," *Computer & Chemical Engineering*, vol. 27, no. 3, pp. 293–346, 2003.

[7] R. J. Patton, "Fault-tolerant control systems: The 1997 situation," in *IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes*, R. J. Patton and J. Chen, Eds., vol. 3. Kingston Upon Hull, UK: IFAC, August 1997, pp. 1033–1054.

[8] P. M. Frank and B. Köppen-Seliger, "New developments using AI in fault diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 10, no. 1, pp. 3–14, 1997.

[9] M. Omana and J. H. Taylor, "Robust fault detection and isolation using a parity equation implementation of directional residuals," in *IEEE Advanced Process Control Applications for Industry Workshop (APC2005)*, Vancouver, Canada, May 2005.

[10] W. Larimore, in *Multivariable System Identification Workshop*. Fredericton, New Brunswick: University of New Brunswick, 31 October – 2 November 2005.

[11] C. Smith, C. Gauthier, and J. H. Taylor, in *Petroluem Applications of Wireless Sensors* (PAWS) *Workshop*. Sydney, Nova Scotia: Cape Breton University, 22–23 August 2005.

[12] E. Durfee, V. R. Lesser, and D. D. Corkill, "Trends in cooperative distributed problem solving," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, no. 1, pp. 63–83, 1989.

[13] M. Omana and J. H. Taylor, "Enhanced sensor/actuator resolution and robustness analysis for FDI using the extended generalized parity vector technique," in *Proc. of American Control Conference*. Minneapolis, Minn.: IEEE, 14-16 June 2006, pp. 2560–2566.

[14] J. H. Taylor and M. Laylabadi, "A novel adaptive nonlinear dynamic data reconciliation and gross error detection method," in *Proc. of IEEE Conference on Control Applications*. Munich, Germany: IEEE, October 4-6 2006, pp. 1783–1788.

[15] M. Laylabadi and J. H. Taylor, "ANDDR with novel gross error detection and smart tracking system," in *12th IFAC Symposium on Information Control Problems in Manufacturing*. Saint-Etienne, France: IFAC, May 17-19 2006.

[16] M. Omana and J. H. Taylor, "Fault detection and isolation using the generalized parity vector technique in the absence of a mathematical model," in *IEEE Conference on Control Applications (CCA)*, Singapore, 1-3 October 2007.

[17] J. H. Taylor and M. Omana, "Fault detection, isolation and accommodation using the generalized parity vector technique," in *submitted to the IFAC World Congress*, Seoul, Korea, July 6–11 2008.

[18] 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.

[19] M. Omana, "Robust fault detection and isolation using a parity equation implementation of directional residuals," Master's thesis, University of New Brunswick, 2005.