Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Fault Detection, Isolation and Accommodation Using the Generalized Parity Vector Technique ⋆

James H. Taylor * **Maira Omana** **

* *University of New Brunswick, Fredericton, NB E3B 5A3 Canada (e-mail: jtaylor@unb.ca)*
** *University of New Brunswick, Fredericton, NB E3B 5A3 Canada (e-mail: maira.omana@unb.ca).*

Abstract: This paper extends the generalized parity vector approach for fault detection and isolation presented in Omana and Taylor [2], [3], [4], to achieve sensor accommodation. In this study, this fault detection, isolation and accommodation technique is applied to a two-phase separator followed by a three-phase gravity separator model used in oil production facilities. This model simulates a large scale process, which allows the technique to be tested on a high dimensional space with more complex system dynamics. The fault management strategy is significantly improved by implementing a fault-size estimation and classification technique using the GPV magnitude signature. This fault characterization is refined by incorporating a recursive fault size recalculation algorithm based on the sensor accommodation error. Two different methods for sensor accommodation and fault size recalculation are proposed to take into account the software and hardware configuration in the plant.

1. INTRODUCTION

In real world processes, such as oil and gas facilities, continuous production is required to achieve productivity and profitability requirements. As a result, stopping a production line suddenly in the middle of a process, to fix or replace a faulty sensor, may result in significant economic losses. To avoid these unexpected interruptions in the plant operation, sensor accommodation must be integrated as part of the fault management strategy. This provides a temporary solution to keep the safe operation in the system, while maintenance can be scheduled without significantly disturbing the process.

So far, the generalized parity vector (GPV) technique has been successfully tested for fault detection and isolation (FDI) using a second-order aircraft engine model [14], a third-order nonlinear model for a jacketed continuously stirred tank reactor (JCSTR) [2], [3] and a fifth-order identified state-space model for a gravity three-phase separation process [15], [4]. In this paper, the performance of new fault-size estimation, classification and sensor accommodation methods is evaluated using the same identified separator model. This allows us to introduce a complete fault detection, isolation and accommodation (FDIA) technique with only input-output data as available information.

This paper is outlined as follows: First, a brief overview of stable factorization and its application to implement the generalized parity vector technique is given in section 2. Next, in section 3, sensor and actuator FDI using directional residuals are defined [9], [10], [11]. Section 4 presents the gravity three-phase separation process [15]. The $|GPV|$ signature is defined in section 5 for further use in fault-size estimation and classification in section 5.1. Two different methods for sensor accommodation are presented in sections 5.2.1 and 5.2.2 to fit the hardware configuration of the plant. Correspondingly, two techniques for accommodation error computation and recursive fault-size recalculation are presented in sections 5.3.1 and 5.3.2. Finally, section 6 presents the FDIA results obtained using the separator model described in [4] for two consecutive faults.

⋆ This project is supported by Atlantic Canada Opportunities Agency (ACOA) under the Atlantic Innovation Fund (AIF) program. The authors gratefully acknowledge the support and the collaboration of the Cape Breton University (CBU) and the College of the North Atlantic (CNA).

2. RESIDUAL GENERATION USING THE GENERALIZED PARITY VECTOR TECHNIQUE

The residual generator is implemented using the generalized parity vector technique, which is developed in the stable factorization framework. The significance of using the stable coprime factorization approach is that the parity relations obtained involve stable, proper and rational transfer functions even for unstable plants. Therefore the realizability and stability of the residual generator is guaranteed. Given any $n \times m$ proper rational transfer function matrix $P(s)$, it can be expressed in terms of its left coprime factors as follows [12]:

$$P(s) = \tilde{D}(s)^{-1}\tilde{N}(s) \tag{1}$$

where $\tilde{N}(s)$ and $\tilde{D}(s)$ are called the left coprime factors and belong to the set of stable transfer function matrices. The GPV technique is based on the stable factorization of the system transfer function matrix in terms of its state-space representation. Let the system be described by the set of equations:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2)$$

$$y(t) = Cx(t) + Eu(t) \quad (3)$$

where x, u, and y represent the state variables, inputs, and outputs of the system, respectively. Assuming that the pairs (A, B) and (A, C) are stabilizable and detectable, it is possible to select a constant matrix F with appropriate dimensions, such that the matrix $A_o \triangleq A - FC$ is stable. If C is the identity matrix, $F = \sigma I + A$ is a convenient choice, where σ is a tuning parameter to be chosen based on the process poles. Using the definition of the coprime factorization of P(s) in [13], the left coprime factors are given by:

$$\tilde{N} = C(sI - A_o)^{-1}(B - FE) + E \quad (4)$$

$$\tilde{D} = I - C(sI - A_o)^{-1}F \quad (5)$$

Based on the definition of the transfer function matrix *P(s)* given in (1) and taking the relationship among the desired control input, u_d, and the actual output of the sensors, y, the following relations are obtained:

$$P(s) = \tilde{D}(s)^{-1}\tilde{N}(s) = y(s)u_d(s)^{-1} \quad (6)$$

$$\tilde{D}(s)y(s) - \tilde{N}(s)u_d(s) = 0 \quad (7)$$

Under ideal conditions, when the plant is linear, noise and fault free, equation (7) holds. However, when a fault happens, this relation is violated showing the inconsistency between the actuator inputs and sensor outputs with respect to the unfailed model. Using this fact, the generalized parity vector, *p(s)*, is defined as:

$$p(s) = T_r\,[\,\tilde{D}(s)y(s) - \tilde{N}(s)u_d(s)\,] \quad (8)$$

The GPV $p(s)$ is the Laplace transformed version of a time varying function of small magnitude under normal operating conditions, due to the presence of noise and modeling errors arising from linearization and order reduction. However, it exhibits a significant magnitude change when a fault occurs. Each distinct failure produces a parity vector with different characteristics, allowing the use of the GPV for isolation purposes. A static transformation matrix T_r is introduced to facilitate FDI [4], [14].

3. FAULT DETECTION AND ISOLATION USING DIRECTIONAL RESIDUALS

The basic idea of FDI using failure directions is that each failure will result in activity of the parity vector along a certain axis or in a certain subspace. Depending on the dynamics of the system, some of these reference directions may be close or identical, making the isolation for some faults difficult or unachievable. To overcome the angle separation problem between the reference directions, the calculation of an optimal transformation matrix T_r was introduced in [2]. The transformation matrix T_r in equation (8) plays an important role in FDI using directional residuals. It is desirable to choose T_r to increase the separation angle between the original set of reference directions and reference hyperplanes as much as possible, to enhance robustness and maximize the number of faults that can be isolated and the number of disturbances that can be decoupled, beyond the number of outputs of the system [10]. The optimization of T_r to maximize the isolability of this approach is described fully in [2], [3], [4].

3.1 Actuator Faults

Assuming an additive fault $a_j(t)$ in the j^{th} actuator and using the definitions and choice of A_o in section 2, the GPV becomes:

$$p_{a,j}(s) = -(T_r\tilde{N})^j a_j(s) \triangleq T_r B_n^j \frac{a_j(s)}{s+\sigma} \quad (9)$$

Equation (9) shows that $p_{a,j}(s)$ is restricted to exhibit activity along the direction defined by the j^{th} column of $\tilde{N}$, which we denote B_n^j [14]. Actuator fault isolation is thus based on the angle Θ_j between the GPV and the direction B_n^j in the generalized parity space. If the j^{th} actuator is faulty, this angle should be zero in the ideal case or close to zero if we take into account model uncertainty, noise and/or unknown disturbances. Thus isolation is based on finding the k that minimizes (Θ_j - B_n^k) for the j^{th} fault.

3.2 Sensor faults

Similarly, for an additive fault $s_i(t)$ in the i^{th} sensor the parity vector in (8) reduces to:

$$p_{s,i}(s) = (T_r\tilde{D})^i s_i(s) \triangleq T_r\left[E_d^i + \frac{B_d^i}{s+\sigma}\right] s_i(s) \quad (10)$$

Thus, for the sensor failure case, it is not possible to confine $p_{s,i}(s)$ to lie in a fixed direction. Only for fortuitous cases, depending on the dynamics of the system, can this be achieved. However, for any system, the GPV always lies on a hyperplane in the generalized parity space, defined by the vectors E_d^i and B_d^i [14]. Thus isolation is based on finding the k that minimizes (Θ_i - SP^k) for the i^{th} fault. Note that a static GPV algorithm may be obtained by setting s=0 in this development. Results in this presentation are obtained using the static GPV approach; the use of a dynamic technique is under study.

4. GRAVITY THREE-PHASE SEPARATION PROCESS DESCRIPTION

Three-phase separators are designed to separate and remove the free water from the mixture of crude oil and water. The fluid enters the separator and hits an inlet diverter. This sudden change in momentum produces the initial gross separation of liquid and vapor. In most designs, the inlet diverter contains a downcomer that directs the liquid flow below the oil/water interface. This forces the inlet mixture of oil and water to mix with the water continuous phase (i.e., aqueous phase) in the bottom of the vessel and rise to the oil/water interface. This process is called water-washing, and it promotes the coalescence of water droplets which are entrained in the oil continuous phase. The inlet diverter assures that little gas is carried with the liquid. The water wash assures that the liquid does not fall on top of the gas/oil or oil/water interface, mixing the liquid retained in the vessel and making control of the oil/water interface difficult.

The gas flows over the inlet diverter and then horizontally through the gravity settling section above the liquid. As the gas flows through this section, small drops of liquid that were entrained in the gas and not separated by the inlet diverter are separated out by gravity and fall to the gas-liquid interface. Some of the drops are of such a small

diameter that they are not easily separated in the gravity settling section. Before the gas leaves the vessel it passes through a coalescing section or mist extractor to coalesce and remove them before the gas leaves the vessel.

The simulation model basically consists of two processes, as illustrated in figure 1. The first is a two-phase separator in which hydrocarbon fluids from oil wells are separated into two phases to remove as much light hydrocarbon gases as possible. The produced liquid is then pumped to the three-phase separator (i.e., the second process), where water and solids are separated from oil. The produced oil is then pumped out and sold to refineries and petrochemical plants if it meets the required specifications.

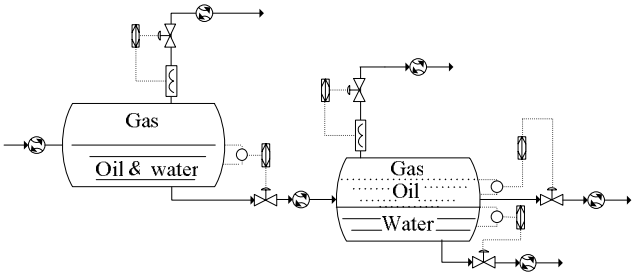

Fig. 1. Gravity three-phase separator process

The two separation processes of the simulation model are controlled to maintain the operating point at its nominal value, and to minimize the effect of disturbances on the produced oil's quality. As shown in figure 1, the first separation process is controlled by two PI controller loops. In the first loop, the liquid level is maintained by manipulating the liquid outflow valve. The second loop is to control the pressure inside the two-phase separator by manipulating the amount of the gas discharge. The second separation process has three PI controller loops. An interface level PI controller maintains the height of the oil/water interface by manipulating the water dump valve. While the oil level is controlled by the second PI controller through the oil discharge valve, the vessel pressure is maintained constant by the third PI loop [5].

5. GPV MAGNITUDE SIGNATURE

So far, the Generalized Parity Vector magnitude, $|GPV|$, has been used only for fault detection purposes. However, it is possible to characterize a $|GPV|$ signature for fault-size estimation and classification for further use in sensor accommodation. The $|GPV|$ signature is defined by its slope change right after the fault is applied, its peak value, $|GPV|_{peak}$ and its steady state value, $|GPV|_{ss}$.

5.1 Fault-size estimation and classification

The block diagram shown in Fig. 2 summarizes the fault detection, isolation and accommodation (FDIA) process using the Generalized Parity Vector technique. If there is no model available or there has been a setpoint change, the identification module is executed using the reconciled inputs-outputs measurements sent by the data reconciliation agent [6], [1] [1].

[1] The use of a data reconciliation agent as a preprocessor justifies the assumption that the data for FDIA is noise free.

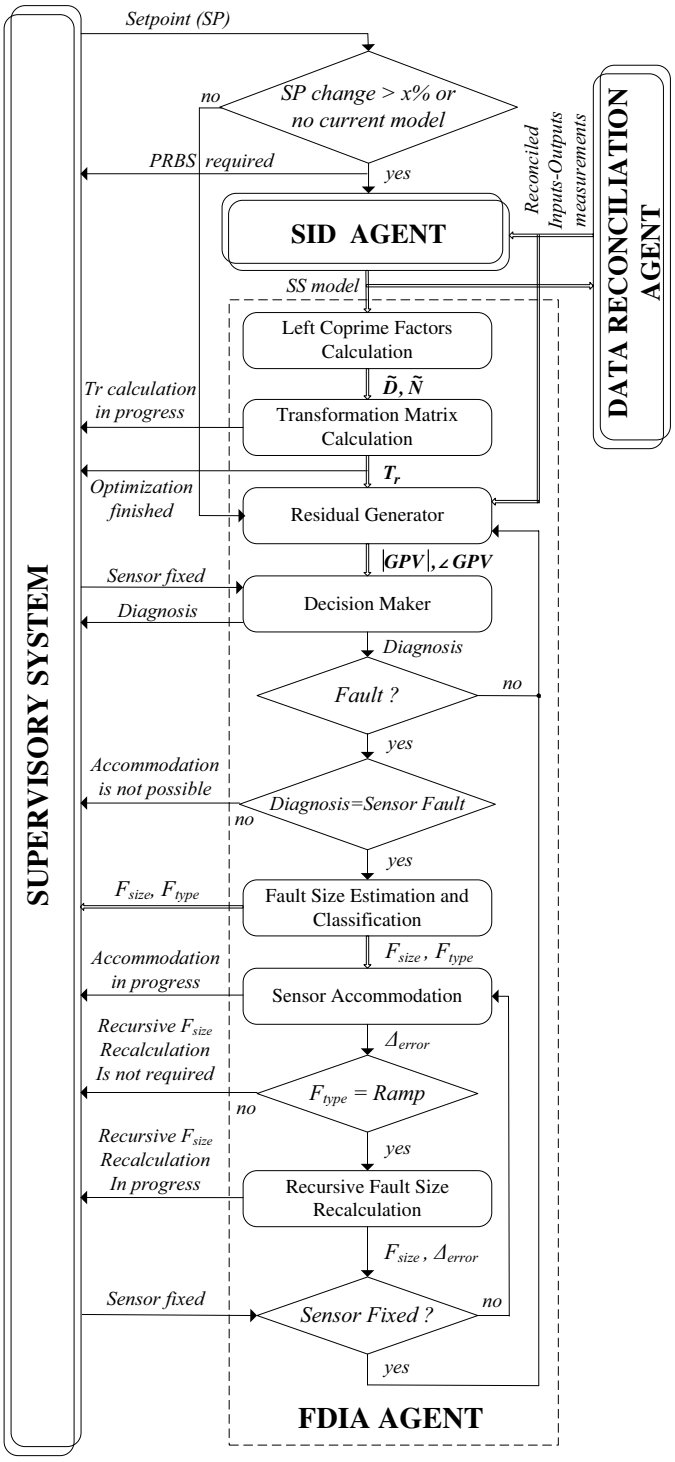

Fig. 2. FDIA block diagram

Then, the corresponding left coprime factors are calculated using the identified state-space (SS) representation. Once the left coprime factors are calculated, the GPV magnitude and angles are computed by the residual generator for each input-output set at every sample. Then, the decision maker block provides a detection and isolation decision based on the $|GPV|$ and the $\angle GPV_{min}$. If the magnitude threshold is exceeded, a fault is detected at $t{=}t_{fault}$ and isolated based on $\angle GPV_{min}$. Also, if the $|GPV|$ slope change at $t{=}t_{fault}$ is larger than a predetermined slope threshold, the FDI algorithm classifies the fault as bias type, otherwise a

ramp type is declared. Once the detection, isolation and classification steps are performed, the algorithm proceeds to estimate the fault size, F_{size}, if the isolated fault corresponds to one of the sensors. For the actuator fault case, there is no purpose in calculating F_{size}, since it is not possible to perform accommodation. If a valve is stuck, it cannot be compensated and must be repaired as soon as possible to avoid further damage in the plant.

For the bias case, the fault size is calculated based on the $|GPV|_{peak}$. Conversely, for the ramp case, F_{size} is computed using the $|GPV|_{ss}$ value. In both cases, different fault-size scenarios are simulated to obtain the corresponding $|GPV|_{peak}$ vs. F_{size} or $|GPV|_{ss}$ vs. F_{size} pairs. Using these sets of data, the best fitting is calculated for each case, providing an equation for F_{size} as a function of $|GPV|_{peak}$ or $|GPV|_{ss}$, depending on the fault type declared previously. While for the bias case the fault-size estimation and classification problem may seem trivial due to the ability to acquire and manipulate simulation data, that is not the case when we are dealing with an actual process. In a real plant, there are limitations on how small the sampling time can be and also, on the amount of data that the supervisory system can send to the FDI agent without overloading the network. For our specific application, a wireless sensor network agent manages real time communications between the control room and the offshore oil facility. For further information on the Petroleum Application of Wireless Systems (PAWS) project see [5]. Therefore, the frequency of the data set received in the control room is also restricted by the wireless network specifications.

Figures 3 and 4 show the simulation results using the identified separator model described in [4] for a +15% bias fault applied to the treator vapor pressure sensor (F5). We observe in Fig. 3 that the GPV angle corresponding to fault 5 is the smallest, giving a clear isolation. We also note that this fault is not accommodated in this scenario. To illustrate the infeasibility of using the faulty sensor measurement for fault size estimation and classification, we set the fault to happen at t=100 sec, which is between two sampling intervals. Since the sampling period in this simulation is 0.75 sec, the last pressure reading available before the fault occurs is at t=99.75 sec. In Fig. 3, it is observed that the pressure measurements change rapidly during the first 0.9 sec after the fault happened. This behavior was expected, due to the fast dynamic nature of the pressure and the appropriate controller action. From previous simulations results it was established that it is only possible to accurately estimate the fault size for a sampling time of 0.3 sec, since the first reading after the fault occurs is 230 PSI at t=100.2 sec. Conversely, if the sampling period is 0.75 sec, the first pressure reading available after the fault was applied is 202.8 PSI at t=100.5 sec, giving a wrong fault-size estimation of 1.4% which has a 90.67% error. This also affects the fault classification, since the slope change may not be large enough to exceed the specified threshold to declare a bias fault.

Let us assume that the same sampling time of 0.75 sec is used to estimate the fault size and type using the $|GPV|$ signature method proposed above. From Fig. 4 it is observed that the $|GPV|$ at t=100.5 sec is still 994.6 times larger than the fault free $|GPV|_{ff}$ at t=99.75 sec.

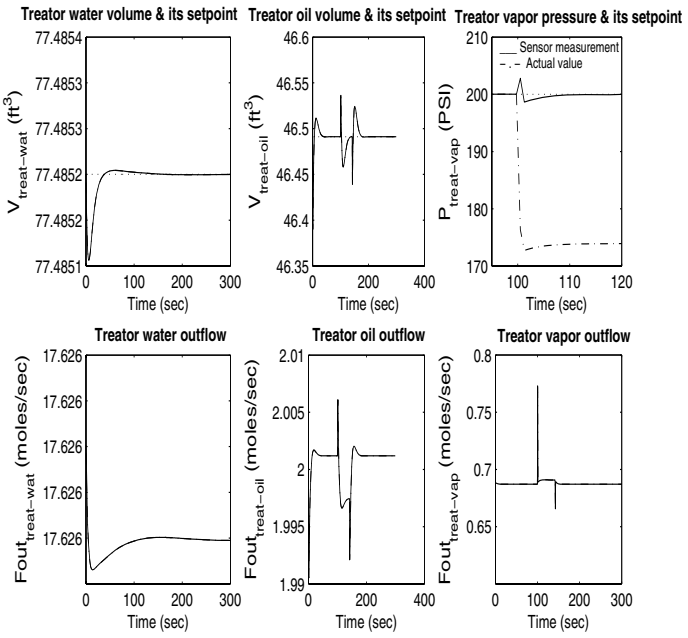

Fig. 3. Treator time histories

This sharp slope change allows the classification of this fault as bias type and makes it possible to use the $|GPV|$ value at t=100.5 sec as $|GPV|_{peak}$ for $F_{size} = 15\%$, during the offline curve fitting procedure to obtain the F_{size} vs. $|GPV|_{peak}$ function. After the fault size and type are defined, the accommodation block is implemented as described in section 5.2.

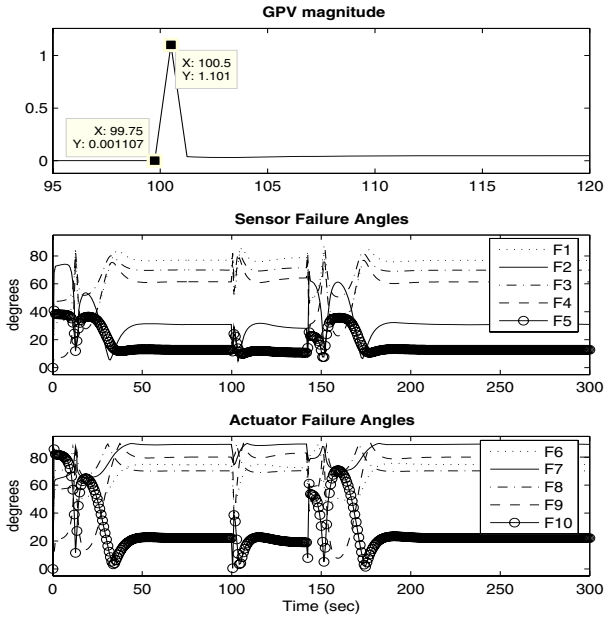

Fig. 4. $|GPV|$ and $\angle GPV$, fault F5

5.2 Sensor accommodation

To avoid significant economic losses due to sudden interruptions in the plant operation, sensor accommodation is integrated as part of the fault management strategy.

This increases the system reliability and safety, extends useful life, minimizes maintenance and maximizes performance. Sensor fault accommodation is implemented after the fault-size estimation and classification block is executed, as depicted in Fig. 2. Depending on the control system hardware and software configuration, two different methods can be implemented to compensate a sensor fault:

Method 1: Sensor reading correction The sensor reading correction method can be implemented on any plant with software controllers and some with hardware ones, if the sensor outputs can be manipulated before they are sent to the controller. Although it takes some time to estimate the fault type and size before starting the accommodation, this method is still capable of driving the system back to normal operation. The advantage of this method is that once the accommodation starts, it directly corrects the measurement before it is sent to the controller, providing a faster accommodation than method 2.

The basic idea is to correct the measured variable Y_{meas} at every time sample t_k, by the relative fault size estimated for the bias case $F_{size.bias}$ (in %/100) and/or the estimated relative value for the ramp case, $F_{size.ramp}$ (in %/100 sec), using the corresponding mathematical relations given in (11); this takes into account the effect of both type of faults in the event that a ramp fault happens after the earlier bias fault is isolated and accommodated. The fault sign is obtained from the faulty variable measurement change around its setpoint, right after the fault is detected, and it is included in F_{size}.

$$Y_{meas.corr} = \begin{cases} \dfrac{Y_{meas} \times u(t_k - t_{f1})}{F_{size.bias} + 1} \\ \\ F_{size.ramp} \times (t_k - t_{f2}) \end{cases} \quad (11)$$

where u(t) is the unit step function and t_{f1} and t_{f2} are the times at which faults 1 and 2 were detected.

Method 2: Setpoint manipulation Sensor accommodation using setpoint manipulation is proposed as an alternative for those installations where the sensor outputs are directly wired to a physical controller. Given that the sensor output cannot be accessed for correction, the fault is accommodated by manipulating the original variable's setpoint, $Y_{sp.orig}$. At every sample time t_k, a relative delta setpoint ΔY_{sp} is calculated with the fault size estimated for the bias case $F_{size.bias}$ (in %/100) and/or estimated for the ramp case $F_{size.ramp}$ (in %/100 sec), using the mathematical relation in (12), taking into account the effect of both types of faults in a situation where a second fault happens after the first one is isolated and accommodated. The accommodated setpoint value $Y_{sp.acc}$ is calculated using (13).

$$\Delta Y_{sp} = \begin{cases} F_{size.ramp} \times (t_k - t_{f2}) \\ \\ F_{size.bias} \times u(t_k - t_{f1}) \end{cases} \quad (12)$$

$$Y_{sp.acc} = (1 + \Delta Y_{sp}) \times Y_{sp.orig} \quad (13)$$

Since this accommodation technique manipulates the setpoint, its performance depends on how fast the system responds to the control actions. Thus, for variables with slow dynamics, the accommodation process takes longer to completely accommodate the fault due to the large settling time. Although the sensor output cannot be manipulated before being sent to the controller, it is still possible to correct this measurement only for GPV calculation purposes, using (11). By using $Y_{meas.corr}$ instead of y in (8), the GPV is compensated as well, which permits showing the accommodation behavior in the FDI operator panel. Thus, when the system is completely accommodated, the $|GPV|$ returns to its fault free value, showing that the fault has vanished (Fault # 0), refer to Fig. 8.

5.3 Recursive Fault-Size Recalculation

The fault-size estimation methods presented in section 5.1 for bias and ramp faults produce very accurate results for both cases. However, since the ramp fault increases its magnitude as time progresses, it requires an estimation error less than 0.001% to provide a correct accommodation during a long period of time. Otherwise, as the time passes, the estimation error is amplified by the factor $(t_k - t_{fault})$, making the accommodation incorrect. To overcome this limitation, a recursive fault-size estimation is implemented for each of the methods presented in section 5.2, by calculating the delta of fault-size estimation error, Δ_{error}. It should be noticed that the recursive fault size recalculation cannot start until the accommodation applied using the first $F_{size.ramp}$ reaches its steady state value at $t = t_{acc.ss}$. This allows obtaining a Δ_{error} caused by the fault-size estimation error, and not by the accommodation transient.

Equations (14) and (16) show the Δ_{error} calculation using only available information, such as the fault time t_{fault}, the variable's setpoint, Y_{sp}, the faulty measurement, Y_{meas}, the sampling time, h, and the fault-size estimate, $F_{size.ramp}$ at the current sample time t_k and the previous sample time t_{k-1}. Using the calculated value for Δ_{error}, the new $F_{size.ramp}$ at time t_k can be recursively calculated using (15) and (17), for methods 1 and 2 respectively. Note that the recursive fault-size estimation is updated only at long time intervals; the control loop operates normally between those times.

Δ_{error} calculation for Method 1:

$$\Delta_{error}(t_k) = \frac{Y_{sp} - Y_{meas}(t_k)}{t_k - t_{fault}} \times \; u(t_k - t_{acc.ss}) \quad (14)$$

$$F_{size.ramp}(t_k) = [F_{size.ramp}(t_{k-1}) + \Delta_{error}(t_k)] \times u(t_k - t_{acc.ss}) \quad (15)$$

Δ_{error} calculation for Method 2:

$$\Delta_{error}(t_k) = [Y_{sp.orig} - Y_{meas}(t_k) + F_{size.ramp}(t_{k-1}) \times (t_k - h - t_{fault})] \times u(t_k - t_{acc.ss}) \quad (16)$$

$$F_{size.ramp}(t_k) = \pm \left| \frac{Y_{meas}(t_k) - Y_{sp.orig} - \Delta_{error}(t_{k-1})}{t_k - t_{fault}} \right| \times u(t_k - t_{acc.ss} + h) \quad (17)$$

As illustrated in Fig. 2, the iterative fault-size recalculation is performed while the sensor is not fixed.

6. FAULT DETECTION, ISOLATION AND ACCOMMODATION RESULTS

Using the identified linearized 10^{th} order state-space model discussed in [4], fault detection, isolation and accommodation is performed using the GPV technique and the sensor reading correction method presented in section 5.2.1. The simulation scenario illustrated in Figs. 5 to 10 is described as follows: At t=100 sec, a +14.37%/minute ramp fault is applied to the separator liquid volume sensor (F1) which is repaired or replaced later, at about t=2300 sec. This was simulated as a reset signal sent by the supervisory system at that time; this behavior is shown on the FDIA graphical user interface (GUI) in Fig. 8. This is followed by a -11.62%/minute ramp fault applied to the treator water volume sensor (F3) at t=3000 sec, which is accommodated but not fixed during the simulation time; resulting signals are portrayed in Fig. 6, and again, the operator observes this on the FDIA GUI.

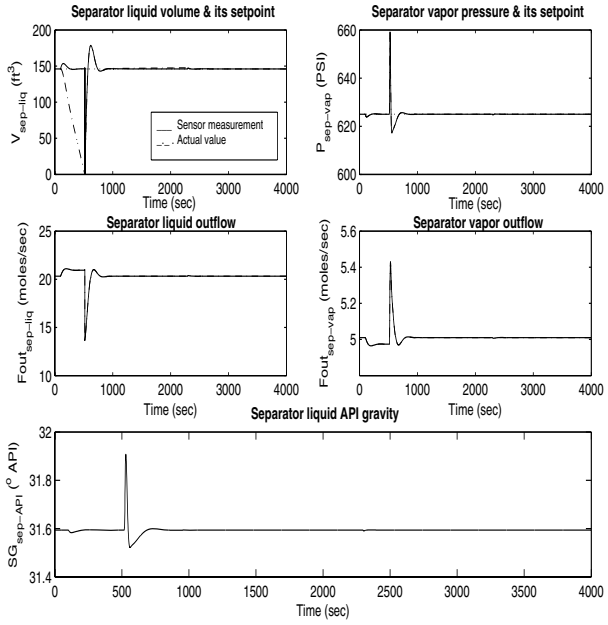

Fig. 5. Separator time histories

In Fig. 5 it is observed that the separator liquid volume measurement completed its accommodation approximately 900 sec after the fault was applied. This was expected from its slow dynamics, which is reflected in the GPV behavior. Figure 7 shows that the $|GPV|_{ss}$ is not reached until approximately t=520 sec. Therefore, $F_{size.ramp}$ is estimated for the first time at t=521.2 sec, since the $|GPV|_{ss}$ is required to evaluate the fitting equations to compute $F_{size.ramp}$, as presented in section 5.1. Once the accommodation started at t=521.2 sec, we wait until the process variables and the $|GPV|$ reach steady state around t=950 sec to start calculating Δ_{error}. If the computed Δ_{error} is becoming excessive, $F_{size.ramp}$ will be recalculated once in a while to give a better estimate. After the accommodation has been performed for around 500 sec, the $|GPV|$ decreases rapidly towards its $|GPV|_{ff}$ value, which is the criteria to declare a total fault compensation. This is illustrated in the operator panel in Fig. 8:

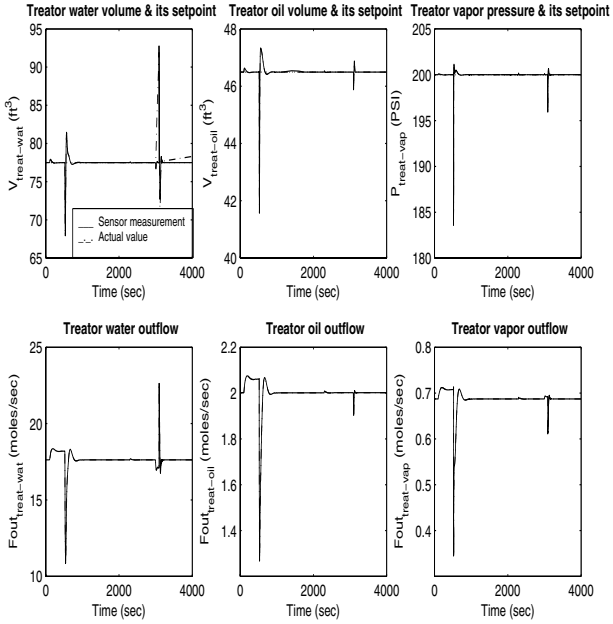

Fig. 6. Treator time histories

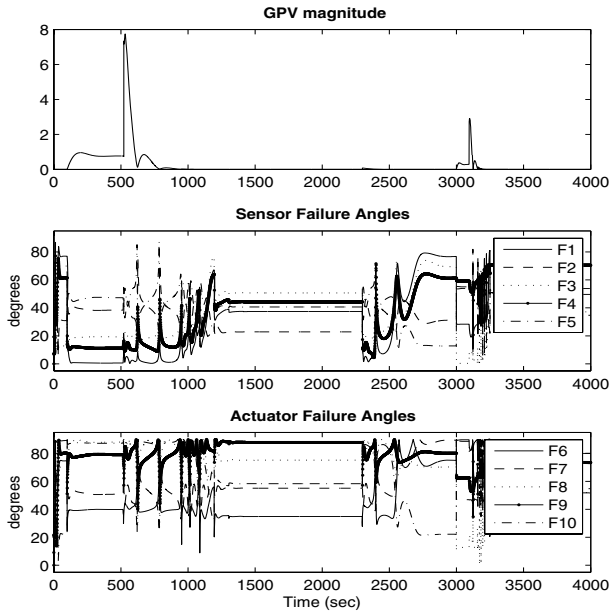

Fig. 7. $|GPV|$ and $\angle GPV$

From 0 sec to 100 sec, fault #0 is displayed, which corresponds to the fault free case. After the fault happened at t=100 sec, there were 20 sec of unknown abnormal behavior represented by fault #-1 in the operator panel. During this period isolation was not possible due to the transient affecting the GPV angles. However, once the transient is finished, the identified fault is correctly displayed in Fig. 8 after t=120 sec. When the $|GPV| < T_h$ around t=1000 sec, faults #0 and #1 are displayed (primarily fault #0), indicating that the fault was successfully accommodated, but still, the sensor needs to be fixed or replaced. At about t=2300 sec, the reset signal notifying that the sensor

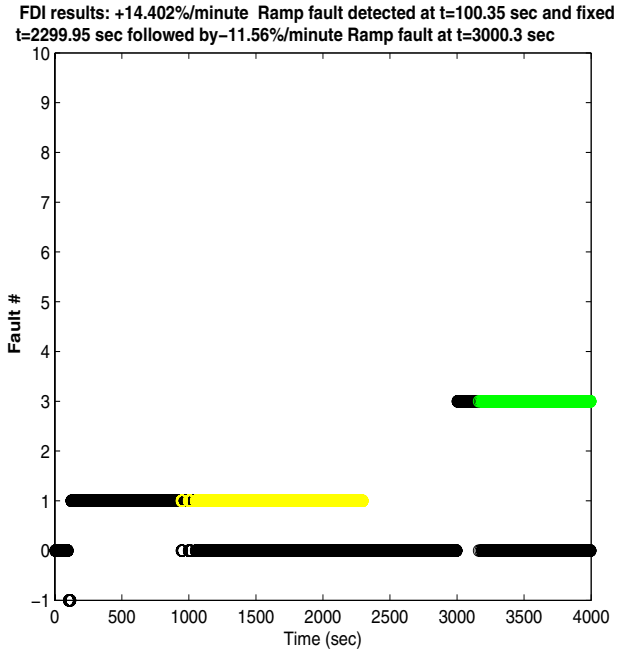

Fig. 8. FDI results

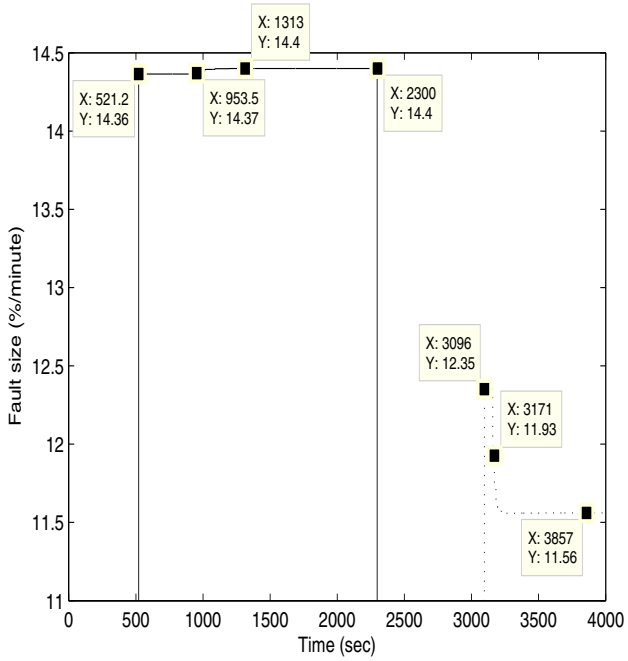

Fig. 10. Recursive fault size recalculation using method 1

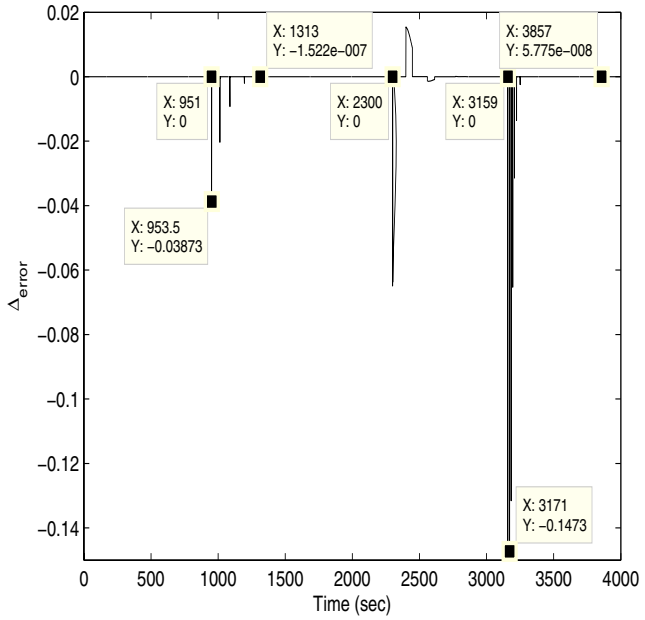

Fig. 9. Delta of fault size estimation error, Δ_{error}, using method 1

problem has been solved is received from the supervisory system. As a result, the operator panel stops displaying fault #1 and remains showing fault #0 until the second fault is detected. This reset signal also produces a transient of approximately 200 sec in Δ_{error} starting at t=2300 sec, when the accommodation stops compensating the wrong measurement and the fixed sensor takes place instead.

At t=3000 sec, the $|GPV|$ increases significantly again, due to the treator water volume sensor fault F3 applied at that time. This is shown in the operator panel as fault #3.

It is observed in Fig. 7 that the $|GPV|_{ss}$ is reached faster than for the separator liquid volume sensor case, due to the treator's faster dynamics. Therefore, the first fault size is estimated at t=3096 sec and accommodation is performed since then. When the process variables and $|GPV|$ reach steady state around t=3160 sec, Δ_{error} calculation starts and the fault size is recursively recalculated if Δ_{error} becomes excessive. Total accommodation is achieve only 210 sec after the fault happens. This is shown as fault #0 and #3 in Fig. 8, from t=3210 sec until the end of the simulation. This indicates that sensor 3 is still faulty but the fault has been successfully accommodated.

Figures 9 and 10 show the Δ_{error} and recursive fault-size recalculation using equations (14) and (15) respectively. It is observed that Δ_{error} decreases with time as the fault size is recalculated. Since it is not possible to obtain 100% accuracy on F_{size} estimation, the recalculated fault size exhibits small oscillations around the actual value in accordance with the variations in Δ_{error}.

7. CONCLUSION

A complete fault detection, isolation and accommodation technique using the GPV approach has been successfully tested on a large-scale industrial process model in the absence of an *a-priori* mathematical model. The $|GPV|$ signature concept has been defined and exploited to implement a fault-size estimation and classification algorithm. Sensor bias faults are directly accommodated; sensor ramp-fault accommodation has been accomplished by integrating a recursive fault-size recalculation algorithm based on the accommodation error. Two different algorithms taking into account the software and hardware implementations in the plant have been effectively developed for sensor-ramp faults and bias compensation.

REFERENCES

[1] M. Laylabadi and J. H. Taylor. ANDDR with novel gross error detection and smart tracking system. *Proc. 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM2006)*. Saint-Etienne, France, 2006.

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

[3] M. Omana and J. H. Taylor. Enhanced sensor/actuator resolution and robustness analysis for FDI using the extended generalized parity vector technique. *Proc. American Control Conference*. Minneapolis, MN, 2006.

[4] M. Omana and J. H. Taylor. Fault detection and isolation using the generalized parity vector technique in the absence of an a priori mathematical model. *IEEE Conference on Control Applications*. Singapore, 2007.

[5] A. Sayda and J. H. Taylor. An implementation plan for integrated control and asset management of petroleum production facilities. *Proc. IEEE International Symposium on Intelligent Control*. Munich, Germany, 2006.

[6] J. H. Taylor and M. Laylabadi. A novel adaptive nonlinear dynamic data reconciliation and gross error detection method. *Proc. IEEE Conference on Control Applications (CCA2006)*. Munich, Germany, 2006.

[7] V. Venkatasubramanian and R. Rengaswamy and K. Yin and S. N. Kavuri. A review of process fault detection and diagnosis, part III: Process history based methods. *Computers and chemical engineering*, volume 27, pages 327–346, 2003.

[8] V. Venkatasubramanian and R. Rengaswamy and K. Yin and S. N. Kavuri. A review of process fault detection and diagnosis, part I: Quantitative model-based methods. *Computers and chemical engineering*, volume 27, pages 293–311, 2003.

[9] F. Hamelin and D. Sauter and M. Aubrun. Fault diagnosis in systems using directional residuals. *Proceedings on the 33rd IEEE Conference on Decision and Control*, 1994.

[10] J. J. Gertler and R. Monajemy. Generating directional residuals with dynamic parity relations. *Automatica*, volumen 31, no. 4, pages 627–635, 1995.

[11] J. J. Gertler. Fault detection and diagnosis in engineering systems. *Marcel Dekker*, Inc., 1998.

[12] P. J. Antsaklis. Proper stable transfer matrix factorization and internal system descriptions. *IEEE Transactions on automatic control*, volume AC-31, no. 7, pages 634–638, 1986.

[13] M. Vidyasagar. Control system synthesis: A factorization approach. *The MIT press*, 1985.

[14] N. Viswanadham and J. H. Taylor and E. C. Luce. A frequency domain approach to failure detection and isolation with application to GE-21 turbine engine control systems. *Control-Theory and advanced technology*, volume 3, no. 1, pages 45–72, 1987.

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

[16] Y. Zhu. Multivariable system identification for process control, *Pergamon*, 2001.

[17] E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on automatic control*, volume AC-29, no. 7, pages 603–614, 1984.

[18] J. J. Gertler. Fault detection and isolation using parity relations. *Control Eng. Practice*, volume 5, no. 5, pages 653–661, 1997.

[19] J. J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine*, volume 8, no. 4, pages 3–11, 1988.

[20] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, volume 12, no. 6, pages 601–611, 1984.

[21] J. J. Gertler and D. Singer. A new structural framework for parity equation-based failure detection and isolation. *Automatica*, volume 26, no. 2, pages 381–388, 1990.