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

by

Maira Omana

Master of Science in Electrical Engineering, UNB, 2006
Bachelor in Electrical Engineering, UIS, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

In the Graduate Academic Unit of Electrical Engineering

Supervisor(s):	James Taylor, Ph.D., Electrical and Computer Engineering
Examining Board:	Chris Diduch, Ph.D., Electrical and Computer Engineering Howard Li, Ph.D, Electrical and Computer Engineering Rickey Dubay, Ph.D., Mechanical Engineering
External Examiner:	N. Viswanadham, Ph.D, Executive Director Centre for Global Logistics and Manufacturing Strategies Indian School of Business

This thesis is accepted by the Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

September, 2009

To my love Shafyn, my soul mate and my inspiration.
To my parents, my aunt, my brother and my sister,
for their love and encouragement through the development of this thesis.

Abstract

In real industrial processes 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 produce significant economic losses. Therefore, the current fault management strategy challenge is not only to detect and isolate faults, but also to accommodate them, to keep the safe operation in the plant while maintenance can be scheduled without significantly disturbing the process.

This research extends the generalized parity vector (GPV) approach originally proposed by Viswanadham, Taylor and Luce and continued by Omana and Taylor, to offer a complete sensor fault detection, isolation and accommodation (FDIA) technique viable for implementation in real industrial applications. Fault detection and isolation is also provided for actuators.

A new systematic approach to implement a recursive on-line transformation matrix computation block using optimization is developed. The calculation of this transformation matrix represents an important contribution to the FDI field using directional residuals because it eliminates the restriction on the number of faults that previous researches were able to isolate and significantly increases FDI robustness. The special case for sensor-actuator faults and the hyperplane intersection problem are identified and solved by extending the objective function during the optimization process to compute the transformation matrix. This modification significantly

improves the isolation results by reducing the ambiguous cases produced by these inevitable special geometrical situations given by the system dynamics. This is a major contribution, because it identifies and overcomes these critical limitations of FDI using directional residuals that previous researchers were not aware of. The plant model availability issue is overcome by incorporating an on-line system identification module before executing the FDIA block. This shows that while the GPV approach is a model-based FDI technique, it is still viable for those plants where an *a-priori* mathematical model is not available.

A fault management strategy is implemented using a novel fault-size estimation, classification and accommodation method based on the static GPV magnitude signature. The proposed fault accommodation technique not only preserves closed-loop stability, but also compensates the actual variable affected by the faulty sensor. A initialization section is introduced to make this FDIA technique capable of handling model and operating point changes. The FDI robustness is significantly improved by incorporating an on-line threshold computation block and combining the strengths of the static and dynamic GPV implementations during the decision-making process.

In this work the FDIA technique is successfully analyzed and simulated on a gravity three-phase separation process used in oil production facilities. This model closely simulates a large scale process, which allows the GPV technique to be validated in a higher dimensional space with more complex system dynamics.

Acknowledgments

I would like to especially thank Dr. James H. Taylor without whom this thesis would not exist. From the formative stages of this thesis, to the final draft, I owe him an immense amount of gratitude. His sage advice, insightful criticisms, and patient encouragement aided the writing of this thesis in innumerable ways.

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

I would like to express my sincere appreciation to the faculty and staff of the Department of Electrical and Computer Engineering at University of New Brunswick for their patience and continuous encouragement. For her effort and assistance, a special thanks to Ms. Denise Burke for her kindness and support during my master and Ph.D. programs.

Special thanks to my family, who loved and supported me all the way since the beginning of my studies.

To my love Shafyn, thanks are not enough to express all my appreciation for being always on my side and believing in me even when I doubted myself.

I would also like to thank Pilar Moreno for her friendship and unconditional support during the most challenging situations.

Fredericton, New Brunswick *Maira Omana*

September, 2009

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

Symbols

$a_i(t)$	i^{th} actuator error
A, B, C, E	State space matrices
b_j	Constant bias in the j^{th} actuator
B^j	j^{th} column of matrix B
B_n^2	Heating fluid inflow valve reference direction
B_n^j	j^{th} actuator reference direction
B_n^1	Outflow valve reference direction
$\tilde{B}$	Extended B matrix
c_i	Constant bias in the i^{th} sensor
$c(T_r)$	Nonlinear inequality constraints
$d(t)$	Disturbance vector
$E[T_{sw}]$	Mean switching time
f_a^j	j^{th} actuator fault
$F_{i,j}$	Objective function for T_r calculation
F_{in}	Mixture inflow
F_{Jin}	Jacket heating fluid inflow
f_s^i	i^{th} sensor fault
F_{size}	Fault size
F_{type}	Fault type

G	Disturbance matrix
$\|GPV\|$	Generalized parity vector magnitude
$\|GPV\|_{dyn}$	Dynamic generalized parity vector magnitude
$\|GPV\|_{sig}$	Generalized parity vector signature
$\|GPV\|_{st}$	Static generalized parity vector magnitude
$\|GPV\|_{slope}$	Generalized parity vector magnitude slope
$\angle GPV$	Generalized parity vector angle
$\|GPV\|_{ff}$	Fault free generalized parity vector magnitude
$\angle GPV_k$	Angle between the GPV and the k^{th} reference direction
$\angle GPV^k_{min}$	Minimum GPV angle corresponding to the k^{th} element
$\angle GPV^i_{ss}$	Steady state generalized parity vector for the i^{th} sensor fault
$\angle GPV_2$	Second smallest generalized parity vector angle
I_L	Intersection line
K, F	Stabilizing gain matrices
N_p, D_p	Alternative notation for the right coprime factors
$N(s)$, $D(s)$	Right coprime factors
N^i_{sp}	Normal to the i^{th} Sensor reference hyperplane
$\tilde{N}_p$, $\tilde{D}_p$	Alternative notation for the left coprime factors
$\tilde{N}(s)$, $\tilde{D}(s)$	Left coprime factors
$p_{a,j}(s)$	Parity relation for a fault in the j^{th} actuator
$p^{ss}_{a,j}(s)$	Steady state parity relation for a fault in the j^{th} actuator
$p_{s,i}(s)$	Parity relation for a fault in the i^{th} sensor
$p^{ss}_{s,i}(s)$	Steady state parity relation for a fault in the i^{th} sensor
$p(s)$	Generalized parity vector
$P(s)$	Transfer function matrix
p_{sw}	Switching probability
$s_i(t)$	i^{th} sensor error

SP^i	i^{th} sensor reference hyperplane
SP^2	Temperature sensor reference hyperplane
SP^1	Volume sensor reference hyperplane
t_{fault}	Detection time
t_s	Settling time
t_{ss}	Steady state time
θ_T	Angle threshold
T_{clk}	Clock period in the PRBS generator
T_{error}	Error threshold
M_T	Magnitude threshold
T_{SID}	System identification threshold
S_T	Slope threshold
T_{in}	Mixture inlet temperature
T_{Jin}	Heating fluid inlet temperature
T_{min}	Minimum switching time
T_{sw}	Switching time
T_r	Transformation matrix
T_s	Sampling time
T_w	Time window
$u_d(s)$	Desired control input
$u(t)$	Input vector
$\tilde{u}(t)$	Extended input vector
$x(t)$	State vector
$y_d(s)$	Actual plant output
$y(s)$	Sensor output
Z_i	i^{th} transformed reference direction

$\Delta\|GPV\|_{peak}$	Delta of generalized parity vector magnitude peak
ΔT	Temperature setpoint variation
ΔV	Volume setpoint variation
Θ_i	Angle between the GPV and the i^{th} sensor reference hyperplane
Θ_j	Angle between the GPV and the j^{th} actuator reference direction
σ	Stabilizing pole shift
τ	Time constant
τ_{max}	Maximum time constant
$\Phi_u(\omega)$	Generalized binary noise power spectrum
$\Phi_u^c(\omega)$	Spectrum of the control signal
$\Phi_u^{opt}(\omega)$	Generalized binary noise optimal power spectrum
$\Phi_v(\omega)$	Power spectrum of disturbances
$\Phi_u^{opt}(\omega)$	Generalized binary noise optimal power spectrum

Abbreviations

ARMA	Autoregressive moving average
Cond	Condition number
FDI	Fault detection and isolation
FDIA	Fault detection, isolation and accommodation
FTC	Fault tolerant control
GBN	Generalized binary noise
GPS	Generalized parity space
GPV	Generalized parity vector
ICAM	Intelligent control and asset management
MAS	Multi-agent systems
NDDR	Nonlinear dynamic data reconciliation
pem	Prediction error/maximum likelihood method MATLAB® function
PAWS	Petroleum application of wireless systems
PRBS	pseudo random binary sequences
JCSTR	Jacketed continuously stirred tank reactor
SID	System identification
SNR	Signal to noise ratio
WSSR	Weighted sum of squared residual

Chapter 1

Introduction

1.1 Overview and literature review

The continuous and accelerating advances in process control have resulted in large and complex plants, increasing the need of high performance fault management strategies. As a result, fault detection, isolation and accommodation (FDIA) has become a critical issue for safe and reliable plant operation and reduction of economic losses. Several techniques using quantitative model-based methods [1], qualitative model-based methods [2] and history-based methods [3] have been developed. They have attempted to achieve faster detection times with lower false alarm and misdetection rates during the fault isolation process, in the presence of noise and disturbances [4]. The suitability of each technique depends on the plant characteristics and model availability.

A general description of a fault detection, isolation and accommodation system is illustrated in Fig. 1.1. In general, faults are deviations from the normal behaviour in the plant or its instrumentation. Faults of interest are usually sensor and actuator faults. There may also be disturbances, which are extra inputs acting in the plant and can be measurable or immeasurable. Measurable disturbances can be com-

pensated so FDI performance is not compromised [5], but unmeasured disturbances cannot. Finally, if the size of a sensor fault can be estimated then it can be accommodated by correction [6], e.g. subtracting a bias fault from an erroneous sensor reading. In this research we are considering only measurable disturbances. Once the FDIA block is executed, the resultant diagnosis is sent to the intelligent control and asset management (ICAM) system supervisor to display the corresponding fault alarms through the human interface.

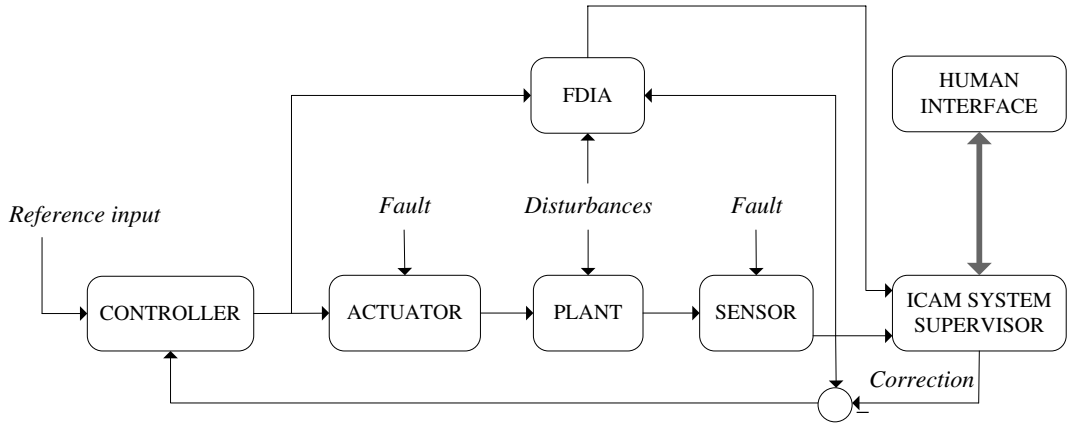

Figure 1.1: Description of an FDIA system

This thesis focuses on the development of a model-based approach using a parity equation implementation of directional residuals for solving the FDIA problem. The research on the fault detection and isolation problem using model-based approaches dates back to the 1970's, when the aerospace fault detection community introduced the concept of analytical redundancy. The basic idea of the analytical redundancy approach is to compare the actual behaviour of the plant with that predicted by a mathematical plant model [7], [8], [9], [10]. The resulting inconsistency is called the residual, which should be close to zero when no fault occurs. However, it shows a significant change when abnormal plant behaviour is detected [11].

In 1976, dynamic parity relations were introduced by Willsky [10], yielding a sys-

tematic development of analytical redundancy provided by the mathematical model of the plant. Later on, Chow and Willsky [7] proposed a procedure to generate parity equations from the state-space representation of a dynamic system. Gertler and Singer [12] extended it to statistical isolability under noisy conditions and generalized the isolability criteria by simultaneously minimizing the sensitivity of residuals to small drifts in cases having only additive plant faults.

While previous approaches were able to detect the faults of a system, there was a need to generate enhanced residuals, which were not only fault sensitive but also fault selective. For the purpose of isolation, Viswanadham, Taylor and Luce [13] introduced the generation of directional residuals using the generalized parity vector (GPV) technique. The idea of this approach is that each failure will result in activity of the parity vector along certain directions or in certain subspaces [13], [14], [15] and [16]. Therefore, fault isolation amounts to determining which predefined fault direction the parity vector is most nearly aligned with.

As another strategy for solving the isolation issue, a structural residual approach was proposed by Gertler and Singer [12]. This approach is characterized by selective fault responses; any residual responds only to a specific subset of faults, and to any fault only a specific subset of the residuals responds. Structured residuals are tested individually, in parallel, against pre-defined thresholds. The outcome of each test is *fired/not fired*, usually represented by the binary values "1" and "0". The set or vector of simultaneous test bits is the observed *binary fault signature* or *fault code* [11].

In parallel, several researches in the area of model-based FDI looked into the possibility of applying Kalman filters to solve fault detection and isolation problems. The Kalman filter idea can be traced to R.K. Mehra and J. Peschon (1971) of Scientific Systems Inc [17]. Other early contributors include Bernard Friedland (1979),

then with the Singer Company [18], and Garru Leininger (1981), then at Purdue [19], followed by Michelle Basseville (1985) of IRISA, France [20], and by Ramine Nikoukhah (1994) of INRIA, France [21], among others [11]. The idea of this model-based FDI technique is that a bank of Kalman filters designed on the basis of all the available possible system models under all possible changes can be used for isolation purpose [1], [22], [23]. This approach uses multiple estimators, each of which is designed for detecting a specific fault. Since each estimator is designed based on a specific hypothesis such as the failure of a single sensor or actuator, all the estimators except the one using the correct hypothesis will produce large estimation errors when a fault occurs. By monitoring the residual of each estimator, the specific fault that has occurred can be detected and isolated.

Although quantitative model-based approaches have made an impact in mechanical and aeronautical engineering applications, they have had little impact in process industries [3]. One of the major advantages of using quantitative model-based approaches is that we will have some control over the behaviour of the residuals. However, several factors such as system complexity, high dimensionality, process nonlinearity and/or lack of good data often render it very difficult or even impractical to develop an accurate mathematical model for the system. This has limited the usefulness of this approach in real industrial processes [1]. In the present work, an on-line system identification (SID) agent has been proposed to overcome the model availability issue and evaluate the GPV performance in the absence of an *a-priori* mathematical model.

Fault accommodation is a crucial part of the fault management strategy as it minimizes the economic losses due to sudden interruptions in the plant operation caused by faulty sensors. Several techniques have been proposed during the past decades for fault tolerant control (FTC) systems design [24], [25], [26], [27]. In [24] process fault accommodation for nonlinear continuous-time systems is proposed by imple-

menting a new fault estimation module based on an adaptive estimator. Then a fault tolerant controller is constructed to compensate for the effect of the faults by stabilizing the closed-loop system. For LTI systems a fault accommodation strategy is proposed in [25]. The accommodation scheme is based on the generalized internal model control architecture for fault tolerant control. In order to improve the performance after a fault, the compensation is considered in two steps according with a fault detection and isolation algorithm. After a fault scenario is detected a general fault compensator is activated. However, once the fault is isolated, a specific compensator is introduced. As another approach, an analytical model-based control redesign for fault accommodation is proposed in [26]. The complete fault accommodation system includes a dynamic process model, fault detection and isolation, fault identification and state estimation, and a dynamic optimization module for control redesign. State estimation is implemented using a pseudomeasurement strategy based extended kalman filter and the dynamic optimization problem is solved through the control parametrization or the simultaneous approach.

Another approach for fault accommodation is sensor measurement replacement based on analytical redundancy using observers and neural networks [28], [29], [30]. In [28], neural networks are used to provide analytical redundancy, from which residuals are generated, enabling the detection of failures on sensor measurements. Upon detection of a failure, the faulty signal is replaced by the neural network based estimate, commonly known as soft sensor. Similar results are presented in [29], where the actual size of each fault is estimated by a multilayer perceptron neural network used as a nonlinear function approximator. A method of sensor validation using nonlinear minor-component analysis is proposed in [30] as a theoretical basis for a sensor FDI module in a fault-tolerant control system. Once a fault in a sensor or any other component is detected and isolated, the fault-tolerant control system would automatically reconfigure itself to compensate for the effect of that fault to maintain

acceptable control performance of the plant even in the presence of faults. In the case of a faulty sensor, the system would utilize any available analytical redundancy among all sensor signals to estimate the value of the physical quantity desired to be measured by that sensor, and that value would then be used for feedback control.

Although satisfactory fault accommodation results have been shown for some applications using the proposed methods, some of them are highly dependant on model accuracy or failed to accommodate faults that are quickly rejected by the controller. For this type of faults, close loop stability is not an issue because the controller is able to quickly drive back all the measured variables to their corresponding setpoint values. However, the actual value of the variable corresponding to the faulty sensor is not at its setpoint, affecting the actual system's output. To improve some of the limitations of these methods, this research has developed two new approaches for fault accommodation using the GPV technique.

1.2 Comparison between different FDI methodologies

Quantitative model-based methods, such as parity space and observer-based approaches, have several desirable characteristics. If complete knowledge of all inputs and outputs of the system, including all forms of interactions with the environment are available, fault diagnosis using quantitative methods would be a well defined problem, regardless of the number of components or faults present. However, some of the limitations of model-based approaches are the difficulties related to modeling and the fact that they do not support an explanation facility owing to their procedural nature.

The type of models the analytical approaches can handle are limited to linear and

some very specific nonlinear models. For some nonlinear models, linear approximations can be poor and hence the effectiveness of these methods might be greatly reduced. For methods like the Kalman filter, when a large-scale process is considered the size of the bank of filters can be very large, increasing the computational complexity. This suggests that it would be efficient to use qualitative abstraction hierarchies to quickly reduce the focus of diagnosis and then use filter banks in the region of focus.

Rule-based expert systems can be used where fundamental principles are lacking and there is an abundance of experience but not enough detail is available to develop accurate quantitative models. Causal models are a very good alternative when the quantitative models are not available but the functional dependencies are understood. Abstraction hierarchies help to focus the attention of the diagnostic system quickly to problem areas. One of the advantages of qualitative methods based on deep knowledge is that they can provide an explanation of the path of propagation of a fault. This is very important when it comes to decision-support for operators. They can also guarantee completeness in that the actual fault will not be missed in the final set of faults identified. However, they suffer from the resolution problems resulting from the ambiguity in qualitative reasoning.

Pattern recognition approaches or classifiers are constructed generally from historic process data. Neural network architectures, such as radial basis function networks and ellipsoidal unit networks, have been demonstrated to perform well in terms of robustness to noise and isolability requirements. However, the limitation of methods which are based solely on historic process data is their generalization capability outside of the training data. Besides its lack of ability to generalize to unfamiliar regions of measurements space, networks also have a difficulty with multiple faults.

This brings out a crucial point of distinction between model-based approaches and

classifiers based on historic process data. In the case of qualitative model-based approaches, the combinatorial complexity is unavoidable and can only be partly alleviated with efficient search. Because of the combinatorially many multiple fault combinations, the search for multiple faults by specifying them explicitly as different classes and obtaining training patterns for them is not feasible. The general limitation of process history based methods is not in the classifiers that are available to them, but in the availability of only a finite sampling of the distribution of the class data in the measurement space.

From an industrial point of view, two of the most important considerations for FDI are the adaptability of the systems and ease of deployment. Process plants rarely remain invariant with periodic minor changes in operating policy, retrofit design and so on. Once a diagnostic system is deployed, it should be able to adapt with minimal effort as new situations are encountered and the scope of the system is expanded. One of the important points that the comparison in this section has tried to emphasize is that no single method is adequate to handle all the requirements for a diagnostic system. Though all the methods are restricted, in the sense that they are only as good as the quality of information provided, it was shown that some methods might better suit the knowledge available than others. Thus, some of these methods can complement one another resulting in better diagnostic systems. Integrating these complementary features is one way to develop hybrid methods that could overcome the limitations of individual solution strategies. As an example, fault explanation through a causal chain is best done through the use of digraphs, whereas, fault isolation might be very difficult using digraphs due to the qualitative ambiguity and so analytical model-based methods might be superior. Hence, hybrid methods might provide a general, powerful problem-solving platform [3].

Based on this idea of hybrid methods, three different approaches were considered for FDI implementation in the initial design stage of the intelligent control and

asset management of oil and gas production facilities research proposed in [31]. One of them was the generalized parity vector approach, which is the model-based technique presented in this thesis. The other two approaches provided fault diagnosis using digraphs [32] and Neuro-fuzzy [33] based methods respectively. These three techniques were successfully implemented and tested using the jacketed continuously stirred tank reactor (JCSTR) model presented in appendix B. However, from the FDI design experience with each of them, it was clear than only the GPV approach was viable for implementation in larger and more complex plants.

The design stage using digraphs and neuro-fuzzy logic was very complex and time consuming even for the simple JCSTR three-input/three output model used as a case study. Thus, it was evident that as the number of faults to identify in the system increases, the practical implementation of these two methods was significantly reduced. Taking into consideration the scope of the petroleum applications of wireless systems (PAWS) project and the strengths and limitations of these methods, it was decided that the GPV technique was the most viable approach for FDI in the final ICAM system supervisor prototype proposed in [34]. The overview for the PAWS project and the ICAM system supervisor is presented in chapter 4.

1.3 Contributions of this thesis

This thesis extends our previous work presented in [5], [6], [35], [36] and [37] to provide a robust FDIA technique based on the GPV approach, capable of accommodating sensor faults and viable for industrial implementation despite the usual lack of model availability. This is achieved by significantly extending and developing the proposed FDIA software to allow on-line initialization having as available information only input-output data and the system time constants. The initialization section makes this FDIA technique feasible for implementation in systems where new iden-

tified models need to be computed on-line for different operating points. To take into account this situation, we also propose a system identification agent with FDIA self-diagnosis based on the identified model's quality. This module plays a critical role because it allows the implementation of the GPV technique in the absence of an *a-priori* mathematical model, which is often the case in real industrial processes. This represents a significant advance in the scope of application of quantitative model-based approaches in process industries because it overcomes real industrial limitations, such as such as using identified models with limited percentage of fitting due to the complexity of the system.

In the present work we introduce a new systematic approach to implement a recursive on-line transformation matrix computation block using optimization. The calculation of this transformation matrix represents a major contribution to the FDI field using directional residuals because it eliminates the restriction in the number of faults that previous researches in this area were able to isolate. It also improves significantly the FDI robustness and makes the proposed method a viable implementation for systems where the identified model requires on-line changes to accommodate nonlinear behaviour.

Through the course of this research we also uncovered the special case for sensor-actuator faults and the hyperplane intersection problem, which are inevitable special geometrical situations given by the system dynamics, that result in ambiguous isolation when ignored. However, in this thesis we solve these issues by extending the objective function during the optimization routine to compute the transformation matrix. This modification significantly improves the isolation results by reducing the ambiguous cases produced by these special geometrical situations. This is a major contribution because it identifies and overcomes these critical limitations of FDI using directional residuals that previous researchers were not aware of.

The use of the static GPV is proposed to implement two novel fault-size estimation and classification algorithms and improve the decision-making logic. By combining the complementary strengths of the static and dynamic GPV implementations, the detection time and ambiguous isolation cases are significantly reduced. The main contribution of our fault accommodation method over the traditional measurement replacement using observers is that it is less sensitive to modeling errors. Despite being a model-based approach, our technique for fault-size estimation does not calculate the fault size (F_{size}) or the correct sensor measurement directly from the model. Instead, F_{size} is computed based on the change in the delta of static $|GPV|$ signature, which is proportional to the size of the fault.

The proposed GPV technique provides excellent FDIA results, satisfactory robustness with respect to noise for scenarios with reasonably large signal to noise ratio (SNR) and significantly shorter computation time than similar methods like Kalman filters. The shorter computation time is crucial for real-time FDI implementation in fast industrial processes, making the GPV technique a more viable option for these situations. Also, the fact that our FDIA technique can be fully designed online makes it easy to deploy and adapt to changes in real industrial processes. In this section the major contributions of this thesis have been highlighted. However, chapter 10 presents a more detailed description of this research contributions.

1.4 Thesis organization

This thesis is outlined as follows: first, a general overview of stable factorization and its application to implement the static and dynamic GPV is presented in chapter 2. Next, in chapter 3, fault isolation using directional residuals for sensor and actuator faults is presented, together with the implementation of disturbance decoupling. The system overview and the interaction between the proposed FDIA strategy and

the ICAM system supervisor are summarized in chapter 4, followed by the SID agent design guidelines in chapter 5. Chapter 6 proposes an on-line calculation of an optimal transformation matrix to enhance the FDI properties.

In chapter 7, an improved decision-making logic is described and the GPV signature concept is introduced. Chapter 8 defines the fault size estimation and accommodation methods using the GPV approach and shows the simulation results illustrating the capabilities of the complete FDIA strategy. Finally, chapter 9 evaluates the FDIA robustness with respect to noise and propose some modifications to improve the tradeoff between detection time and sensitivity to high frequency influences. Some relevant simulation results are also included in each chapter to clarify and demonstrate the different aspects discussed in the corresponding section.

Chapter 2

Residual Generation Using the Generalized Parity Vector Technique

2.1 Introduction

Quantitative model-based FDI methods rely on the comparison of a system's available measurements, with *a priori* information represented by the system's mathematical model. There are two main trends of this approach: analytical redundancy or residual generation methods and parameter estimation approaches [16]. In the present work, the FDI methodology is developed based on residual generation.

Since quantitative model-based methods for FDI are developed based on some fundamental understanding of the physics of the process, FDI using analytical redundancy methods is a viable implementation for systems where *a priori* knowledge is available in terms of mathematical functional relationships between the inputs and outputs of the system [1]. However, in chapter 5, it will be shown that quantitative model-based techniques using residual generators are feasible even for those systems where

the only available information is input-output data.

While there are several methods for residual generation, this work is focused on residual generation using the generalized parity vector (GPV) technique, which is developed in the stable factorization framework. Before introducing the GPV concept, some of the fundamental mathematics of stable factorization are outlined in this chapter.

2.2 Stable factorization

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 defined in terms of its right and left stable coprime factors as follows [38]:

$$P(s) = N(s)D(s)^{-1} \tag{2.1}$$

$$P(s) = \tilde{D}(s)^{-1}\tilde{N}(s) \tag{2.2}$$

where $N(s)$ and $D(s)$ are said to be right coprime factors, and $\tilde{N}(s)$ and $\tilde{D}(s)$ are called the left coprime factors. All factors belong to the set of stable transfer function matrices. For both cases, this implies that the matrix extension of the Bezout identity holds [39]:

$$X(s)N(s) + Y(s)D(s) = I \tag{2.3}$$

$$\tilde{X}(s)\tilde{N}(s) + \tilde{Y}(s)\tilde{D}(s) = I \tag{2.4}$$

where $X(s)$, $Y(s)$, $\tilde{X}(s)$ and $\tilde{Y}(s)$ are also in the set of stable transfer function matrices; refer to [39] for further information. Equating the left and right descriptions of $P(s)$ given in equations (2.1) and (2.2), the following identity holds:

$$\tilde{D}(s)N(s) - \tilde{N}(s)D(s) = 0 \tag{2.5}$$

Combining the identities in equations (2.3), (2.4) and (2.5), the double coprime factorization of $P(s)$ is defined as follows:

$$\begin{bmatrix} Y(s) & X(s) \\ -\tilde{N}(s) & \tilde{D}(s) \end{bmatrix} \begin{bmatrix} D(s) & -\tilde{X}(s) \\ N(s) & \tilde{Y}(s) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{2.6}$$

The two block matrices in the left-hand side of (2.6) are unimodular, and each one is the inverse of the other. A unimodular matrix is a real square matrix with determinant equal to one. More generally, a matrix with elements in the polynomial domain $F(s)$ is called unimodular if it has an inverse whose elements are also in $F(s)$. A matrix is therefore unimodular if its determinant is a unit of $F(s)$. As a result, the matrix inverse of a unimodular real matrix is another unimodular matrix [40]. Equation (2.6) is called the generalized Bezout identity [13].

2.2.1 Stable factors from state-space representations

The GPV technique is based on the stable factorization of the system transfer function matrix in terms of its state-space representation. The following is an abstract from Vidyasagar, [39]: Let the system be described by the set of equations:

$$\dot{x}(t) = Ax(t) + Bu(t) + Gd(t) \tag{2.7}$$
$$y(t) = Cx(t) + Eu(t) \tag{2.8}$$

where x, u, d, and y represent the state variables, inputs, disturbances and outputs of the system, respectively, and A, B, G, C, E are matrices of compatible dimensions. The following derivations in this section are obtained assuming $d = 0$. The effect of disturbances will be analyzed in section 3.5 and the corresponding modifications to the left coprime factors computation will be presented there. The input/output transfer matrix of this system is thus:

$$P(s) = C(sI - A)^{-1}B + E \tag{2.9}$$

The objective is to derive a doubly coprime factorization of P, as it is given in Theorem (2.2.1) by Vidyasagar, [39]. Before introducing this theorem, we define some terminology involved in it.

Definition 2.2.1. *Given the system (2.7)-(2.8), the matrix pair (A,B) is said to be stabilizable if for every initial condition it is possible to find an input that asymptotically steers the state x(t) to the origin. Therefore, there always exists a matrix K such as $A - BK$ is asymptotically stable [41].*

Definition 2.2.2. *Given the system (2.7)-(2.8), the matrix pair (A,C) is said to be detectable if all the unobservable modes are stable. Therefore, there always exists a matrix F such as $A - FC$ is asymptotically stable [41].*

Theorem 2.2.1. *Given the system (2.7)-(2.8), suppose the pairs (A, B) and (A, C) are stabilizable and detectable, respectively. Select constant matrices K and F, such that the matrices $A_o : A - BK$, $\tilde{A}_o : A - FC$ are both Hurwitz. Then $P = N_p D_p^{-1} = \tilde{D}_p^{-1}\tilde{N}_p$ and*

$$\begin{bmatrix} Y & X \\ -\tilde{N}_p & \tilde{D}_p \end{bmatrix} \begin{bmatrix} D_p & -\tilde{X} \\ N_p & \tilde{Y} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{2.10}$$

where the various matrices are defined as follows [39]:

$$\begin{aligned}
\tilde{N}_p &= E + C(sI - \tilde{A}_o)^{-1}(B - FE) && (2.11)\\
\tilde{D}_p &= I - C(sI - \tilde{A}_o)^{-1}F && (2.12)\\
N_p &= E + (C - EK)(sI - A_o)^{-1}B && (2.13)\\
D_p &= I - K(sI - A_o)^{-1}B && (2.14)\\
X &= K(sI - \tilde{A}_o)^{-1}F && (2.15)\\
Y &= I + K(sI - \tilde{A}_o)^{-1}(B - FE) && (2.16)\\
\tilde{X} &= K(sI - A_o)^{-1}F && (2.17)\\
\tilde{Y} &= I + (C - EK)(sI - A_o)^{-1}F \quad \blacksquare && (2.18)
\end{aligned}$$

Using the definition given in equations (2.11) and (2.12), the left coprime factors $\tilde{N}_p$ and $\tilde{D}_p$ are renamed for simplicity hereafter as $\tilde{N}$ and $\tilde{D}$. Also, to ease the mathematical derivations, it is assumed that the state space matrix E in equation (2.8) is equal to zero, implying that the plant input/output transfer matrix is strictly proper. Based on these assumptions the left coprime factors are rewritten as follows:

$$\tilde{N} = C(sI - \tilde{A}_o)^{-1}B \tag{2.19}$$
$$\tilde{D} = I - C(sI - \tilde{A}_o)^{-1}F \tag{2.20}$$

2.3 Generalized parity vector technique

Consider the linear time invariant plant depicted in Fig. 2.1, which is described by the $n \times m$ transfer function matrix $P(s)$. Let u_d be the desired or correct control input and u be the actual plant input (output of the actuator). The relation between u and u_d is given by equation (2.21):

$$u(t) = u_d(t) + a(t) \tag{2.21}$$

where $a(t)$ is a time-varying vector with elements $a_i(t)$ representing various failure modes of the i^{th} actuator. Similarly, let y_d be the actual output of the plant (desired or correct sensor output) and y to be the actual output of the sensor. The relation between these variables is expressed in equation (2.22):

$$y(t) = y_d(t) + s(t) \tag{2.22}$$

where $s(t)$ is a time-varying vector with elements $s_i(t)$ representing sensor failures. As noted, the development here assumes noise-free signals; the effects of noise are considered in chapter 9. The variables u_d, y and d are "external" or available for FDI, while u and y_d are "internal" or inaccessible. In this research we have considered only measurable disturbances, which is often the case for processes that are affected

by external inputs that are not controlled by the system. Clearly, the presence of unmeasured disturbances may defeat any FDI method that monitors u_d and y. The relationship among these signals is depicted in Fig. 2.1.

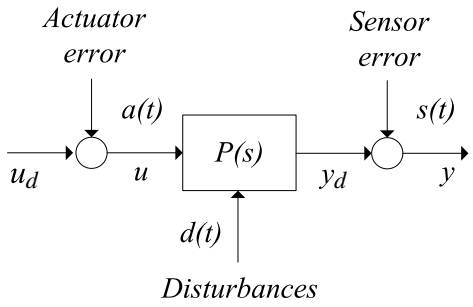

Figure 2.1: LTI system with disturbances, sensor and actuator failure models, noise-free case

Based on the system in Fig. 2.1, the definition of the transfer function matrix $P(s)$ given in equation (2.2) and taking the relationship among the desired control input, u_d, and the actual output of the sensors, y, the following fault-free relations are obtained:

$$P(s) = \tilde{D}(s)^{-1}\tilde{N}(s) = Y(s)U_d(s)^{-1} \tag{2.23}$$

$$\tilde{D}(s)Y(s) - \tilde{N}(s)U_d(s) = 0 \tag{2.24}$$

Under ideal conditions, when the plant is linear, noise and fault free, equation (2.24) 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:

$$p(s) = T_r[\tilde{D}(s)Y(s) - \tilde{N}(s)U_d(s)] \tag{2.25}$$

where the transformation matrix T_r is discussed below. Using the definitions in Theorem (2.2.1) and equations (2.19) and (2.20) the GPV can be rewritten in terms of the state space matrices as follows:

$$p(s) = T_r\big\{[I - C(sI - A + FC)^{-1}F]Y(s) - [C(sI - A + FC)^{-1}B]U_d(s)]\big\} \tag{2.26}$$

The GPV $p(t) = \mathcal{L}^{-1}[p(s)]$ is of small magnitude under normal operating conditions, if modeling errors and noise effects are small enough. However, when a failure occurs, the GPV exhibits a significant magnitude change, which depends on the fault size. Each distinct fault produces a parity vector with different characteristics, allowing the use of the GPV for isolation purposes. The effect of modelling error is important; we address it by obtaining the best possible identified model (see chapter 5) and by adapting decision thresholds accordingly. The effect of noise in the GPV behaviour and how this affect the FDI results are discussed in detail in chapter 9.

A transformation matrix T_r is introduced in equation (2.25) to improve fault isolability. The present work is focused on fault diagnosis using the direction of the parity vector under various failure conditions. We assume hereafter that T_r is constant, and that F in equations (2.19) and (2.20) is chosen such that $\tilde{A}_o = -\sigma I$ where possible (which can always be done if (A, C) is observable and C is a square matrix); this simplifies our discussion of GPV behaviour [13]. However, the static and dynamic GPV implementations are also possible for rectangular C matrixes by choosing an F matrix with proper dimensions that satisfies Definition (2.2.1) and Theorem (2.2.1); $\tilde{A}_o = -\sigma I$ is most convenient as it yields just one tuning parameter, σ. This is illustrated in section 5.4 where the GPV technique was successfully implemented using a rectangular identified model of a separation process with 5 inputs/outputs and 10 states and also in the paper by Viswanadham and Taylor [13], where the GPV approach was effectively applied to the GE-21 turbine engine using a rectangular model with 3 inputs/outputs and 2 states.

2.4 Time-domain implementation

In the time-domain the GPV can be implemented using either a static or dynamic approach as presented in sections 2.4.1 and 2.4.2, each one exhibiting different fea-

tures. These differences allow us to combine both implementations to enhance the detection, isolation and accommodation capabilities of the parity vector technique. These aspects will be discussed in more detail in chapter 7.

2.4.1 Dynamic parity vector

The GPV can be implemented as a detection filter whose numerator $\tilde{N}$ and denominator $\tilde{D}$ in equation (2.23) are represented by the following general state space models respectively:

$$\dot{x}_N(t) = A_N x_N(t) + B_N u_N(t) \tag{2.27}$$

$$y_N(t) = C_N x_N(t) + E_N u_N(t) \tag{2.28}$$

$$\dot{x}_D(t) = A_D x_D(t) + B_D u_D(t) \tag{2.29}$$

$$y_D(t) = C_D x_D(t) + E_D u_D(t) \tag{2.30}$$

Taking the Laplace transform of equations (2.27) to (2.30) and solving for $X_D(s) = \mathcal{L}(x_D(t))$ and $X_N(s) = \mathcal{L}(x_N(t))$, the following relations are obtained:

$$sX_N(s) = A_N X_N(s) + B_N U_N(s) \Rightarrow X_N(s) = (sI - A_N)^{-1} B_N U_N(s) \tag{2.31}$$

$$Y_N(s) = C_N X_N(s) + E_N U_N(s) \tag{2.32}$$

$$sX_D(s) = A_D X_D(s) + B_D U_D(s) \Rightarrow X_D(s) = (sI - A_D)^{-1} B_D U_D(s) \tag{2.33}$$

$$Y_D(s) = C_D X_D(s) + E_D U_D(s) \tag{2.34}$$

Given that the residual vector output from the failure detection filter is given by $p = Y_D - Y_N$, $p(s)$ can be rewritten by substituting equations (2.31) to (2.34) and properly grouping terms as follows:

$$p(s) = [E_D + C_D(sI - A_D)^{-1}B_D]U_D(s) - [E_N + C_N(sI - A_N)^{-1}B_N]U_N(s) \tag{2.35}$$

Comparing equation (2.35) with the definition of the parity vector given in terms of the left coprime factors in equation (2.25), the following relations are established:

$$
\begin{aligned}
T_r(s)\tilde{N}(s)u_d(s) &= [E_N + C_N(sI - A_N)^{-1}B_N]U_N(s) \\
&= T_r[C(sI - \tilde{A}_o)^{-1}B]u_d(s) \qquad (2.36)
\end{aligned}
$$

$$
\begin{aligned}
T_r(s)\tilde{D}(s)y(s) &= [E_D + C_D(sI - A_D)^{-1}B_D]U_D(s) \\
&= T_r[I - C(sI - \tilde{A}_o)^{-1}F]y(s) \qquad (2.37)
\end{aligned}
$$

Equating terms in equations (2.36) and (2.37), the state space matrices for the numerator dynamics given by equations (2.27) and (2.28) and the denominator dynamics given by equations (2.29) and (2.30) can be defined as:

$$
\begin{aligned}
A_N &= \tilde{A}_o \\
B_N &= B \\
C_N &= T_rC \\
E_N &= [0] \\
U_N &= u_d \qquad (2.38)
\end{aligned}
\qquad\qquad
\begin{aligned}
A_D &= \tilde{A}_o \\
B_D &= -F \\
C_D &= T_rC \\
E_D &= T_rI \\
U_D &= y \qquad (2.39)
\end{aligned}
$$

In general $\tilde{A}_o$ is defined as $\tilde{A}_o = A - FC$, where F must be chosen such as $\tilde{A}_o$ is stable. However, if C is the identity matrix (which is often the case), $F = \sigma I + A$ is a convenient choice, where σ is a real tuning parameter to be chosen based on the process poles and it determines the speed of the detection filter. For an arbitrary C matrix, F is chosen such as Definition (2.2.1) and Theorem (2.2.1) are satisfied.

Using the residual vector output definition $p = Y_D - Y_N$ from the failure detection filter and the state space matrices obtained in equations (2.38) and (2.39), the GPV can be implemented by solving the set of differential equations for $y_D(t) = \mathcal{L}^{-1}(Y_D(s))$ and $y_N(t) = \mathcal{L}^{-1}(Y_N(s))$ defined by (2.27) to (2.30). In this research the dynamic GPV has been implemented by discretizing equations (2.27) to (2.30) using the zero order hold method [42]; this is simpler, faster and more practical compared with a continuous-time implementation.

2.4.2 Static parity vector

The static parity vector provides a steady state implementation for the GPV. This approach has some advantages with respect to the dynamic formulation in terms of detection speed and fault size estimation and classification. Also, the static implementation is computationally simpler, since it does not involve solving differential equations or discretization. These aspects will be discussed and illustrated in more detail in chapter 7.

In order to derive the static GPV implementation, it is assumed an additive actuator fault $a(t)$ of the form described in equation (2.21). Substituting equation (2.21) into (2.25) and noting that $\tilde{D}y - \tilde{N}u_d = 0$, the following relation is obtained:

$$p_a(s) = -(T_r\tilde{N})a(s) \tag{2.40}$$

To demonstrate the directional behaviour of the residual, consider a step or actuator bias failure. Since $\tilde{N}(s)$ and $\tilde{D}(s)$ are stable matrices, the steady-state parity vector, p^{ss}, can be computed using the final value theorem [13] according to equation (2.41).

$$p^{ss} = \lim_{s\to 0} sp(s) = \lim_{t\to\infty} p(t) \tag{2.41}$$

Thus, for a constant actuator bias fault of magnitude b, the p_a^{ss} is given by:

$$\begin{aligned} p_a^{ss} &= \lim_{s\to 0}\left\{ -T_r\tilde{N}\begin{bmatrix}\frac{b}{s}\end{bmatrix} s\right\} = \lim_{t\to\infty} p(t) && (2.42)\\ &= -T_r\tilde{N}(0)b && (2.43)\end{aligned}$$

Similarly, for the sensor case, it is assumed an additive bias fault $s(t)$, of magnitude c and the form described in equation (2.22). Based on the same assumptions used for the actuator case, the steady-state GPV is computed as follows:

$$\begin{aligned} p_s(s) &= (T_r\tilde{D})s(s) && (2.44)\\ p_s^{ss} &= \lim_{s\to 0}\left\{T_r\tilde{D}\begin{bmatrix}\frac{c}{s}\end{bmatrix} s\right\} = \lim_{t\to\infty} p(t) && (2.45)\\ &= T_r\tilde{D}(0)c && (2.46)\end{aligned}$$

2.5 GPV relation with Kalman filter

The Kalman filter is a recursive algorithm for state estimation that has found wide applications in chemical as well as other industrial processes. The Kalman filter in state-space model is equivalent to an optimal predictor for a linear stochastic system in the input-output model. The essential Kalman filter theory can be summarized briefly as follows.

Consider the linear finite dimensional stochastic system governed by the following state space model [1]:

$$x(t+1) = Ax(t) + Bu(t) + w(t) \tag{2.47}$$

$$y(t) = Cx(t) + v(t), t \geq 0 \tag{2.48}$$

where $x(t)$ is n-dimensional vector, A, B and C are matrices with suitable dimensions, x_0 has mean $\bar{x_0}$ and covariance Σ_0. In order for the Kalman filter to converge, the matrix pair (A, C) must be observable. The random variables $w(t)$ and $v(t)$ are independent gaussian white noise sequences representing the process and measurement noise, with expected values (means) $E\{w(t)\} = 0$, $E\{v(t)\} = 0$ and the covariance matrix:

$$E\left\{\begin{pmatrix} w(t) \\ v(t) \end{pmatrix}(w^T(\tau), v^T(\tau))\right\} = \begin{pmatrix} Q & S \\ S' & R \end{pmatrix}\delta_{t-\tau} \tag{2.49}$$

where $\delta_{t-\tau}$ is Kronecker's delta function. In the actual implementation of the filter, the measurement noise covariance R is usually measured prior to operation of the filter; this is generally practical (possible) because we need to be able to measure the process output anyway (while operating the filter) so we should generally be able to take some off-line measurements in order to determine the variance of the measurement noise. The determination of the process noise covariance Q is generally more difficult, as we typically do not have the ability to directly observe the process we are estimating. In practice, Q and R might change with each time step or

measurement, however here we assume they are constant [43]. The optimal Kalman filter estimates the state $x(t+1)$ based on the observed data $y(t)$ and $u(t)$ by minimizing the function:

$$J = \lim_{t\to\infty} E(e^T(t)e(t)) \tag{2.50}$$

where $e(t)$ is the estimation error and it is defined as:

$$e(t) = y(t) - C\hat{x}(t) \tag{2.51}$$

Assume the initial state and noise sequences are jointly Gaussian. Consider the estimator $\hat{x}(t+1) = E\{x(t+1)|y(t), ..., y(0), u(t), ..., u(0)\}$. Then, the filtered state $\hat{x}(t+1)$ is given by:

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + K(t)[y(t) - C\hat{x}(t)]; \quad \hat{x}_0 = \bar{x}_0 \tag{2.52}$$

The Kalman filter gain $K(t)$ is given by:

$$K(t) = [A\Sigma(t)C^T + S][C\Sigma(t)C^T + R]^{-1} \tag{2.53}$$

where $\Sigma(t)$ is a $n \times n$ state error covariance matrix. Under conditions where Q and R are in fact constant, both the state error covariance and the Kalman gain will stabilize quickly and then remain constant. If this is the case, these parameters can be pre-computed by either running the filter off-line, or, for example, by determining the steady-state value of $\Sigma(t)$ as described in [44]. For this scenario $\Sigma(t)$ and $K(t)$ would be constant matrices which would give the Kalman filter the maximum computation efficiency. Since this section is presenting the basis to compare the FDI performance in terms of computation time for the GPV and Kalman filter methods, we assume from now on that $K(t)$ and $\Sigma(t)$ in equations (2.52)and (2.53) are constant. This assumption will provide a more fair comparison between the GPV and Kalman filter methods.

Now that the basis for the Kalman filter theory has been presented, we proceed to show that the Kalman filter estimation error is equivalent to the GPV definition

given in equation (2.25) [13]. First, let us rewrite equation (2.52) in a differential form assuming a constant Kalman gain:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K[y(t) - C\hat{x}(t)] \tag{2.54}$$

Taking the Laplace transform of equations (2.51) and (2.54) and solving for $\hat{x}(s)$ the following relations are obtained:

$$\begin{aligned} e(s) &= y(s) - C\hat{x}(s) && (2.55)\\ s\hat{x}(s) &= A\hat{x}(s) + Bu(s) + K[y(s) - C\hat{x}(s)] \\ \Rightarrow \hat{x}(s) &= (sI - A + KC)^{-1}Bu(s) + (sI - A + KC)^{-1}Ky(s) && (2.56) \end{aligned}$$

Substituting equation (2.56) into (2.55) and rearranging terms, the estimation error can be rewritten as:

$$\begin{aligned} e(s) &= y - C[(sI - A + KC)^{-1}Bu(s) + (sI - A + KC)^{-1}Ky(s)] \\ e(s) &= [I - (sI - A + KC)^{-1}K]y(s) - C[(sI - A + KC)^{-1}B]u(s) && (2.57) \end{aligned}$$

The Kalman gain matrix K in equation (2.57) is chosen such that the *a posteriori* error covariance is minimized and the filter is stable. Similarly, matrix F in equation (2.26) is chosen such as $\tilde{A}_o : A - FC$ in theorem (2.2.1) is stable. Then, by comparing equations (2.57) and (2.26) for the untransformed case ($T_r = I$), it can be established that the Kalman filter estimation error is equivalent in form to the GPV definition. As a result, the computation time required to run one Kalman filter or the GPV for a given system is the same, assuming that K has been previously computed off-line. If K has to be computed on-line, the computation time for the Kalman filter is longer. This is an important result which will be used in section 3.7 to compare the FDI performance for the GPV and Kalman filter methods based on computation time.

Chapter 3

Fault Isolation Using Directional Residuals

3.1 Introduction

The basic idea of FDI using failure directions is that each fault will result in activity of the parity vector along certain axes or in certain subspaces. These reference axes or subspaces are determined by the state space matrices. This information can be used to isolate the fault with fewer parity variables than required using voting based on parity variable magnitude alone, as will be shown in this chapter. Therefore, in many cases this approach is much simpler to implement than a voting scheme.

Note that the GPV activity will only be restricted to specific FDI axes or subspaces if there is only one fault. Multiple simultaneous faults can only be detected, not isolated. However, it will be shown that a second fault can be detected and isolated if an earlier fault has been successfully detected, isolated and accommodated as it is presented in section 8.5.1.

Depending on the dynamics of the system, some of these reference directions may be similar 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 is proposed in chapter 6.

3.2 Actuator faults

In order to analyze actuator FDI, an additive fault $a_j(t)$ of the form described in equation (2.21) is assumed in the j^{th} actuator. Substituting equation (2.21) into (2.25) and noting that $\tilde{D}y - \tilde{N}u_d = 0$, the following relation is obtained:

$$p_{a,j}(s) = -(T_r\tilde{N})^j a_j(s) \triangleq (T_r B_n)^j a_j(s) \tag{3.1}$$

Equation (3.1) shows that $p_{a,j}(s)$ is restricted to exhibit activity along the direction defined by the j^{th} column of $T_r\tilde{N}$. For a system with output $y{=}x$, or state space matrices $C = I_{n\times n}$ and $E = 0_{n\times m}$, as in the JCSTR and separator models used through out this research, equation (2.19) can be rewritten more simply as:

$$\tilde{N} = (sI - \tilde{A}_o)^{-1}B = [diag(s+\sigma)]^{-1}B \tag{3.2}$$

Using equation (3.2), equation (3.1) can be rewritten as follows:

$$p_{a,j}(s) \triangleq (T_r B)^j \frac{a_j(s)}{s+\sigma} \tag{3.3}$$

Comparing equations (3.1) and (3.3), the j^{th} actuator reference direction, B_n^j, is defined as $B_n^j = B^j$ in this special case. Therefore, equation (3.3) can be written in terms of the the j^{th} actuator reference direction as:

$$p_{a,j}(s) \triangleq (T_r B_n)^j \frac{a_j(s)}{s+\sigma} \tag{3.4}$$

Assuming a step or bias failure in the j^{th} actuator, and since $\tilde{N}(s)$ and $\tilde{D}(s)$ are stable matrices, the steady-state parity vector, p^{ss}, can be computed using the final value theorem [13] according to equation (3.5).

$$\lim_{s\to 0} sp(s) = \lim_{t\to\infty} p(t) = p^{ss} \tag{3.5}$$

Thus, for a constant bias of magnitude b_j on the j^{th} actuator, the $p^{ss}_{a,j}$ is given by:

$$p^{ss}_{a,j} = \lim_{s\to 0}\left\{ -(T_r B_n)^j \left[\frac{b_j/s}{s+\sigma}\right] s \right\} = -(T_r B_n)^j \frac{b_j}{\sigma} \tag{3.6}$$

For $T_r = I$, the actuator fault isolation is based on the angle Θ_j between the GPV and B^j_n as illustrated in Fig. 3.1. If the j^{th} actuator is faulty, this angle should be zero in the ideal case or very small, to account for model uncertainty, noise and/or unknown disturbances. It should be noticed that to make the illustration easier, the GPV in Figs. 3.1, 3.2 and 3.3 is plotted in a 3-dimensional space. However, for the separator model described in section 4.3, FDI is performed in a 5-dimensional space, which corresponds to the number of inputs and outputs of the system.

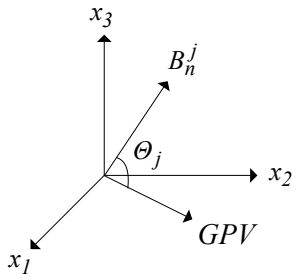

Figure 3.1: Actuator FDI

3.3 Sensor faults

Similarly, to explain sensor FDI an additive fault $s_i(t)$, of the form described in equation (2.22) is applied to the i^{th} sensor. Based on this assumption, the parity vector in equation (2.3) reduces to:

$$p_{s,i}(s) = (T_r \tilde{D})^i s_i(s) \tag{3.7}$$

Assuming the same special case state space representation used in section 3.2, $C = I$ and $(sI - \tilde{A}_o) = [diag(s+\sigma)]$, equation (2.20) can be simplified as:

$$\tilde{D} = I - (sI - \tilde{A}_o)^{-1} F = I - [diag(s+\sigma)^{-1}]F \tag{3.8}$$

Recalling that we chose $\tilde{A}_o = A - FC = [diag\,(-\sigma)] = -\sigma I$ for the special case $C = I$, we have $F = A + \sigma I$, so equation (2.20) can be rewritten as follows:

$$\tilde{D} = I - [diag(s+\sigma)^{-1}][A + \sigma I] \tag{3.9}$$

To simplify equation (3.9) the vector B_d is defined as:

$$B_d = -F = (-A - \sigma I) \tag{3.10}$$

Using the previous definitions, equation (3.7) can be rewritten in terms of the vectors which define the sensor reference hyperplane as:

$$p_{s,i}(s) = \left[T_r^i + \frac{(T_r B_d)^i}{s+\sigma} \right] s_i(s) \tag{3.11}$$

Thus, for the sensor failure case, it is not possible to confine $p_{s,i}(s)$ to lie along a fixed axis. Only for fortuitous cases, depending on the dynamics of the system, can this be achieved. However, for any system, the GPV always lies in a hyperplane of the generalized parity space, defined by the column vectors T_r^i and $(T_r B_d)^i$.

As in actuator FDI, to demonstrate the directional behaviour of the residual, consider a step or bias failure in the i^{th} sensor. Then, the steady-state parity vector, p^{ss}, can be computed using the final value theorem [13] according to equation (3.5). For a constant bias of magnitude c_i on the i^{th} sensor, the $p_{s,i}^{ss}$ is given by:

$$\lim_{s\to 0} s \left[T_r^i + \frac{(T_r B_d)^i}{s+\sigma} \right] \frac{c_i}{s} = \lim_{t\to\infty} p(t) = p_{s,i}^{ss} \tag{3.12}$$

$$p_{s,i}^{ss} = (T_r B_s)^i \frac{c_i}{\sigma} \tag{3.13}$$

where

$$B_s = \sigma I + B_d \tag{3.14}$$

The sensor fault isolation is based on the angle Θ_i, between the GPV and the i^{th} sensor reference hyperplane, SP^i, as illustrated in Fig. 3.2. If the i^{th} sensor is faulty,

this angle should be zero or very small.

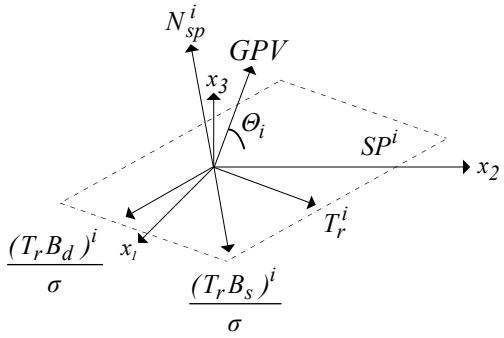

Figure 3.2: Sensor FDI

3.4 Special case for sensor-actuator faults

We consider a special case in terms of the actuator direction B_n^j and the SP^i normal, N_{sp}^i shown in Fig. 3.2 and defined by $N_{sp}^i = T_r^i \otimes (T_r B_d)^i$ as:

$$B_n^j \cdot N_{sp}^i = 0 \tag{3.15}$$

If the dot product of B_n^j and the normal to the i^{th} sensor reference hyperplane is zero, then the j^{th} actuator axis lies on the i^{th} sensor reference hyperplane [45] as is illustrated in Fig. 3.3. This condition would be a result of the system state space

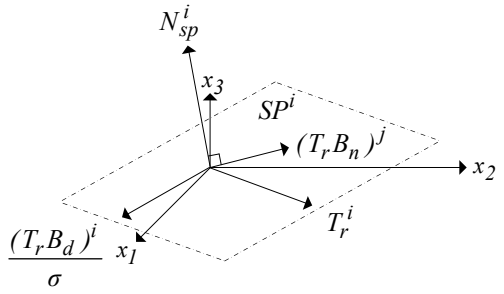

Figure 3.3: Special case for Sensor-Actuator FDI

structure. For this case it is not possible to calculate a transformation matrix T_r such as the actuator reference direction can be taken out of the sensor reference hyperplane. This can be demonstrated mathematically by showing that equation (3.16)

is satisfied for arbitrary T_r, which was done by symbolic manipulation in MATLAB®. From this proof it is concluded that equation (3.16) holds for any transformation matrix T_r, which means that the j^{th} actuator reference direction will always lie on the i^{th} sensor reference hyperplane, regardless of T_r.

$$(T_r B_n)^j \cdot (T_r^i \otimes (T_r B_d)^i) = 0 \tag{3.16}$$

Under this circumstance we may still be able to distinguish between these faults by taking a more detailed look at the parity vector relation in equation (3.11): Let us assume that $s_i(s) = c_i/s$ (a bias fault) [13]; we can apply the initial value theorem to show that the initial GPV activity is in the direction T_r^i, as follows:

$$\lim_{s\to\infty} s\left[T_r^i + \frac{(T_r B_d)^i}{s+\sigma}\right]\frac{c_i}{s} = \lim_{t\to 0} p(t) = p_{s,i}^o \tag{3.17}$$

$$p_{s,i}^o = T_r^i\, c_i \tag{3.18}$$

and invoke the results obtained in equations (3.12), (3.13) and (3.14) using the final value theorem, to demonstrate that the steady-state GPV activity is in the direction $\left[T_r^i + \frac{(T_r B_d)^i}{\sigma}\right] \triangleq (T_r B_s)^i$. Thus $p_{s,i}^{ss}$ and $p_{s,i}^o$ define a sector in the hyperplane SP^i that encompasses the dynamic behaviour of $p_{s,i}$.

This was demonstrated for the JCSTR model described in appendix B by applying volume (S_1) and temperature in the tank (S_2) sensor faults at *t=0.5 hours*. Figures 3.4 and 3.5 show that after the fault is applied, the angle between the GPV and B_s approaches zero, while the angle between the GPV and T_r grows. Thus the GPV swinging from T_r to B_s for the faulty case is shown. However, this is only valid for the pure linear case.

Nevertheless, we can still clearly isolate the i^{th} sensor fault from the j^{th} actuator fault unambiguously as long as B_n^j is not in or near the cone angle (sector) between T_r^i and $(B_s T_r)^i$ [5], using the following logic:

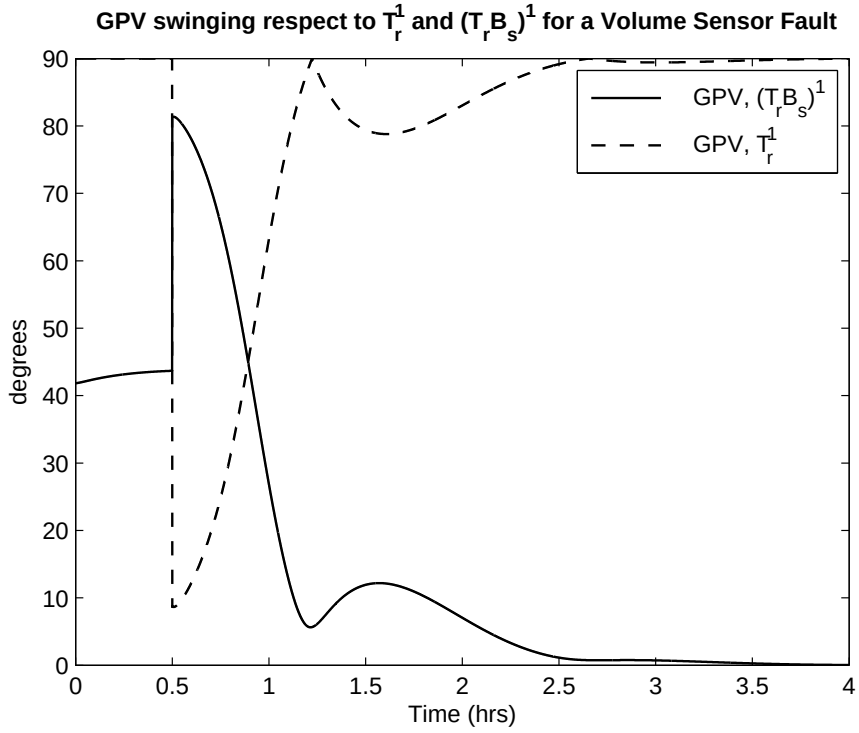

Figure 3.4: $\measuredangle\,(GPV, T_r^1)$ and $\measuredangle\,(GPV, (T_rB_s)^1)$ for a Volume Sensor Fault

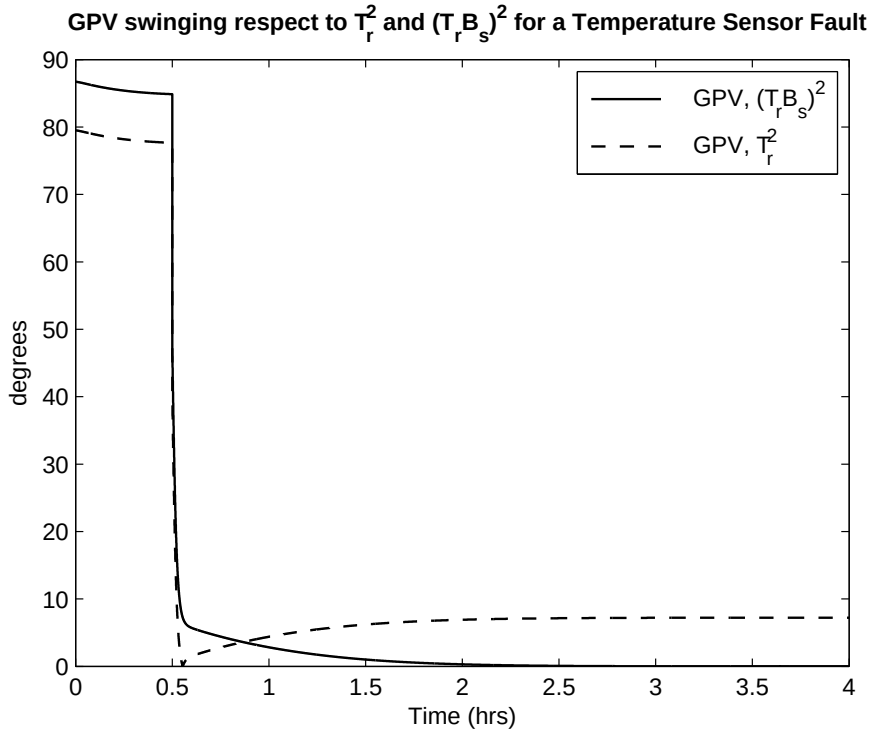

Figure 3.5: $\measuredangle\,(GPV, T_r^2)$ and $\measuredangle\,(GPV, (T_rB_s)^2)$ for a Temperature Sensor Fault

$$\left.\begin{array}{l} if\ \measuredangle(GPV, SP^i) \leq \theta_T\ then \\ \quad if\ \measuredangle(GPV, B_n^j) \leq \theta_T\ then\ f_a^j \\ \quad else\ f_s^i \end{array}\right\} \tag{3.19}$$

where θ_T is a small angle threshold and f_s^i and f_a^j denote the i^{th} sensor and j^{th} actuator faults respectively. Based on equation (3.19), a sensor fault is declared if just the angle between the GPV and SP^i is smaller than a threshold value. Conversely, an actuator fault is announced if both the angles between the GPV and SP^i and the GPV and B_n^j are smaller than the angle threshold θ_T [35].

For some cases when B_n^j is inside the cone sector, depending on the dynamics of the system, a sensor fault may produce a GPV aligned with the j^{th} actuator reference direction. This condition makes the sensor fault isolation incorrect, since the j^{th} actuator will be the one declared faulty, according with the logic in equation (3.19). Although it was already proved in equation (3.16) that the the actuator reference direction can not be taken out of sensor reference hyperplane, it is still possible to ensure that the j^{th} actuator reference direction is not aligned with the sensor fault steady-state GPV. This is achieved by extending the objective function or equivalently adding an optimization constraint during the T_r calculation, as presented in section 6.3.

3.5 Disturbance decoupling

In real world processes there are many disturbances acting on the plant. It would be desirable that an FDI technique should be unaffected by such disturbances, to the extent possible. Many disturbances in real plants are accessible/measurable, and it is possible to compensate for their effect in this FDI approach. This is implemented by introducing them as extra inputs to the FDI algorithm (parity vector filter), by

extending the filter inputs, the B matrix and the left coprime factor $\tilde{N}$ accordingly.

Note that we are not assuming that such disturbances exhibit any particular temporal behaviour [11]; it suffices that they are not involved in the system's control loops as sensor or actuator signals. Faults, on the other hand, are modelled as additive inputs at particular sensors and actuators, and may have specific temporal behaviour. For instance, sensor measurements can be affected by a $\pm$ bias level, making the readings off by a constant value. Also, some actuators (e.g., valves) can be stuck at some fixed value and are thus unable to perform the required control action. Any other extra inputs that are not sensor or actuator signals and have an effect on the process are categorized as disturbances. It is desirable that the FDI approach not be affected by such extra inputs.

In order to demonstrate that residual directionality can be unaffected by extra inputs whose measurements are available, equation (2.7) is rewritten as follows:

$$\dot{x}(t) = Ax(t) + \tilde{B}\tilde{u}(t) \tag{3.20}$$

where $\tilde{B} = [\, B \quad G \,]$, $\tilde{u} = [\, u \quad d \,]^T$, and G and d represent the disturbance allocation matrix and inputs respectively. Using equation (3.20), the coprime factorization definition given by equation (2.19) can be rephrased by replacing B with $\tilde{B}$ as:

$$\tilde{N} = C(sI - \tilde{A}_o)^{-1}\tilde{B} \tag{3.21}$$

Using the modified definition of $\tilde{N}$ given by equation (3.21) and the extended input $\tilde{u}$, disturbance decoupling is implemented in the stable factorization framework to make the GPV immune to measurable disturbance effects.

To illustrate the effect of implementing disturbance decoupling in the FDI algorithm, a -50% temperature sensor fault is applied at *t=2.5 hours* to the JCSTR, followed by a 50% low inlet flow disturbance [1], F_{in}, at *t= 6 hours*, for setpoints that differ from

[1]Note that inlet flow is neither an actuated nor sensed variable; it is exogenous.

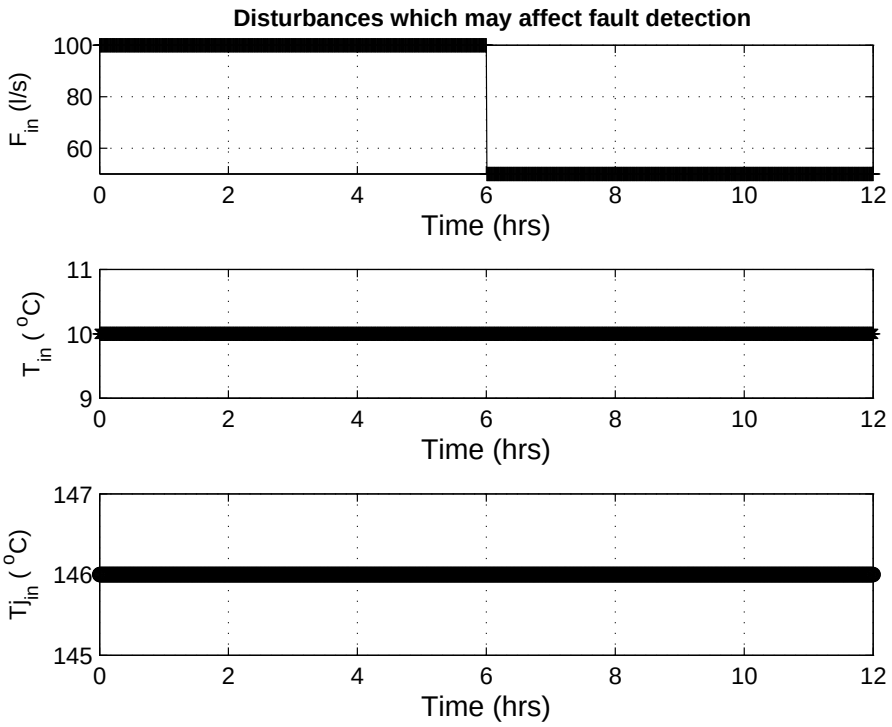

Figure 3.6: Disturbance time-histories

nominal, ΔV=40% and ΔT=10%. The corresponding time histories are shown in Figs. 3.6, 3.7 and 3.8. Since we are considering only measurable disturbances, they are already known and therefore there is no need for isolation; our aim is to decouple FDI from the effect of disturbances.

In Fig. 3.9 (FDI without disturbance decoupling) it is observed that the $|GPV|$ is significantly decreased after the disturbance is applied. Since for small fault sizes the magnitude increment after a fault is not very large, this reduction may be enough to cause an indication that the fault disappeared. However, when disturbance decoupling is incorporated into the FDI algorithm, this situation is overcome as illustrated in Fig. 3.10, where the disturbance is rejected quickly, driving the $|GPV|_{faulty}$ back to its undisturbed value.

The effect of disturbances is even more significant in the $\angle GPV$ behaviour, as illustrated in figure 3.11. After the disturbance is applied, the temperature sensor

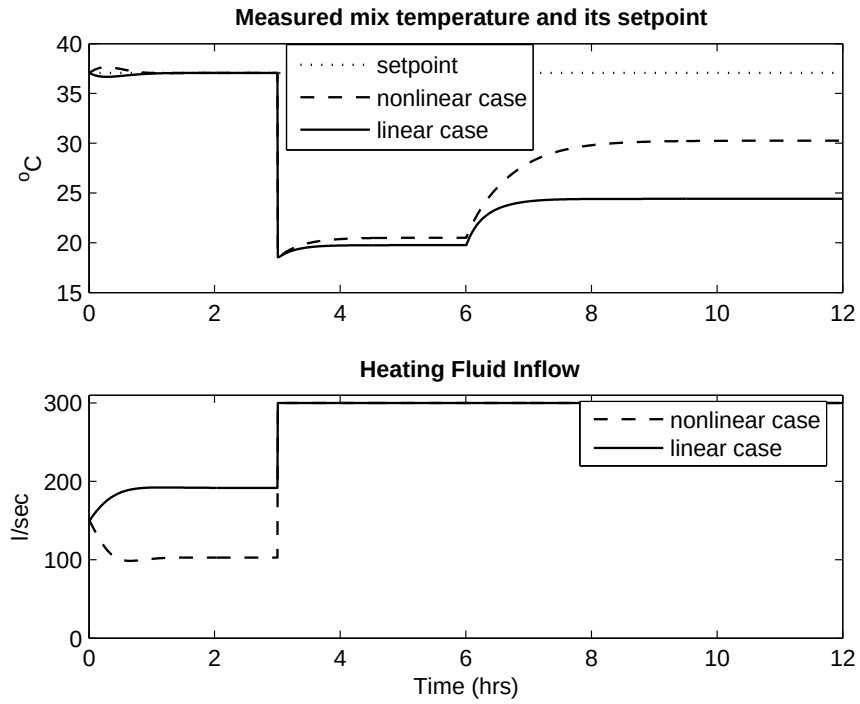

Figure 3.7: Time-histories for a -50% temperature sensor fault + 50% low inlet flow disturbance

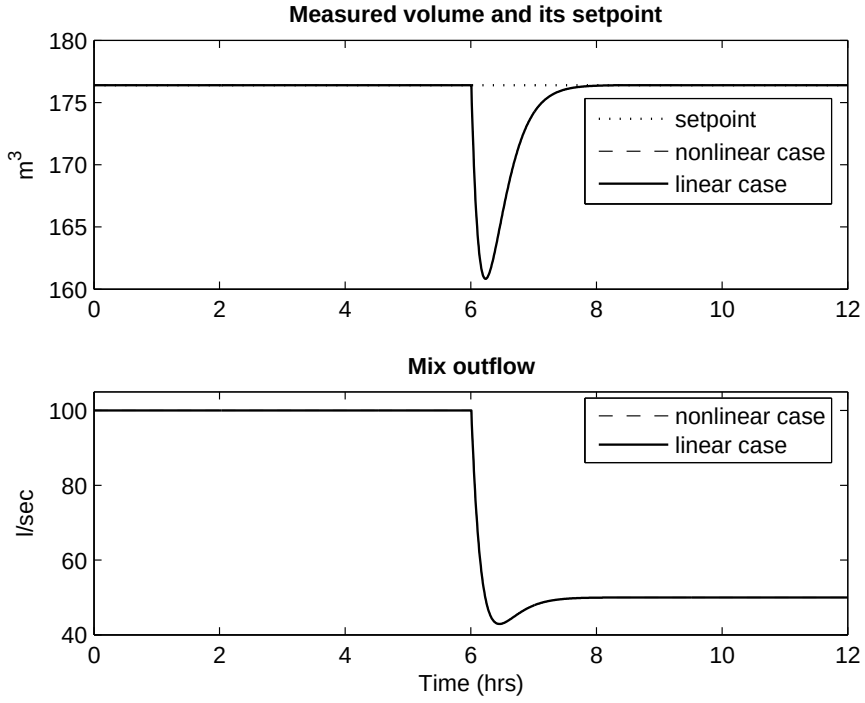

Figure 3.8: Time-histories for a -50% temperature sensor fault +50% low inlet flow disturbance (Linear and nonlinear cases are indistinguishable)

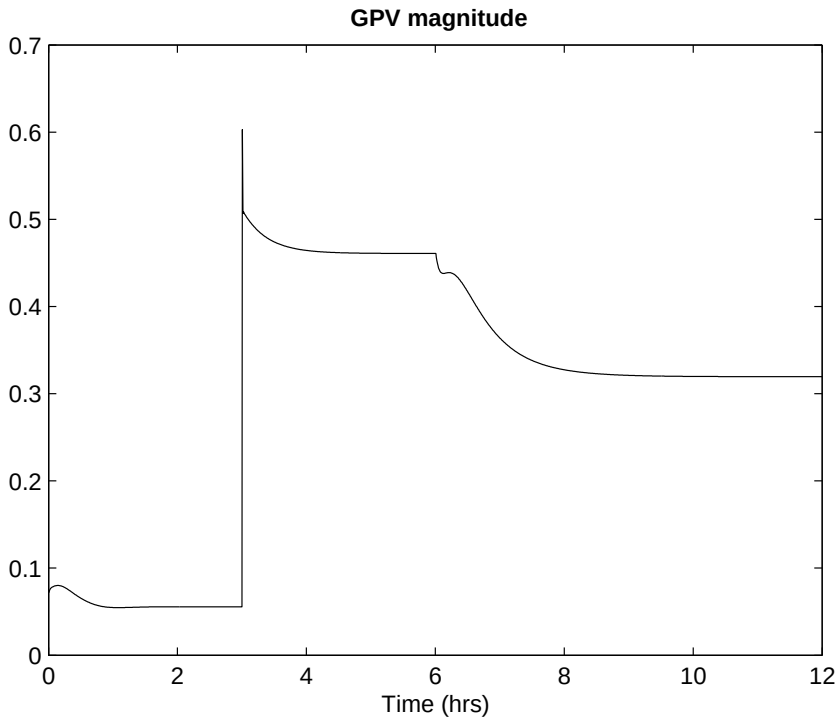

Figure 3.9: $|GPV|$ without disturbance decoupling

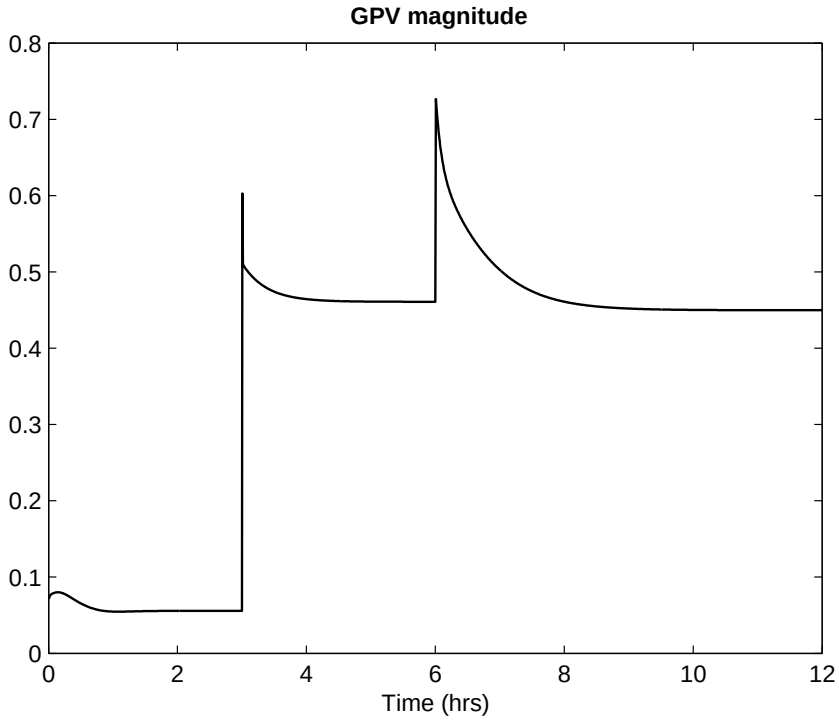

Figure 3.10: $|GPV|$ with disturbance decoupling

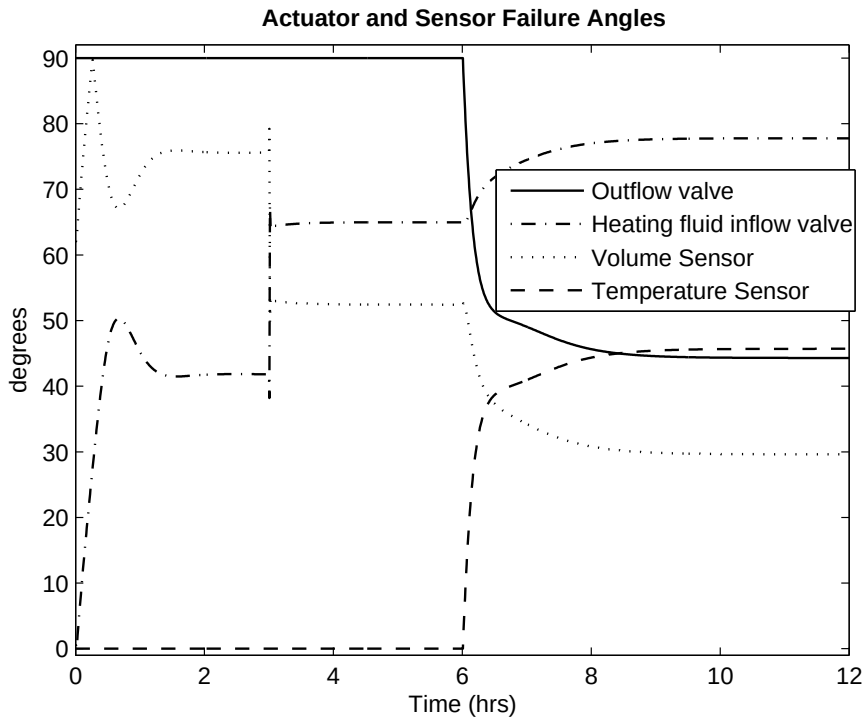

Figure 3.11: $\angle GPV$ without disturbance decoupling

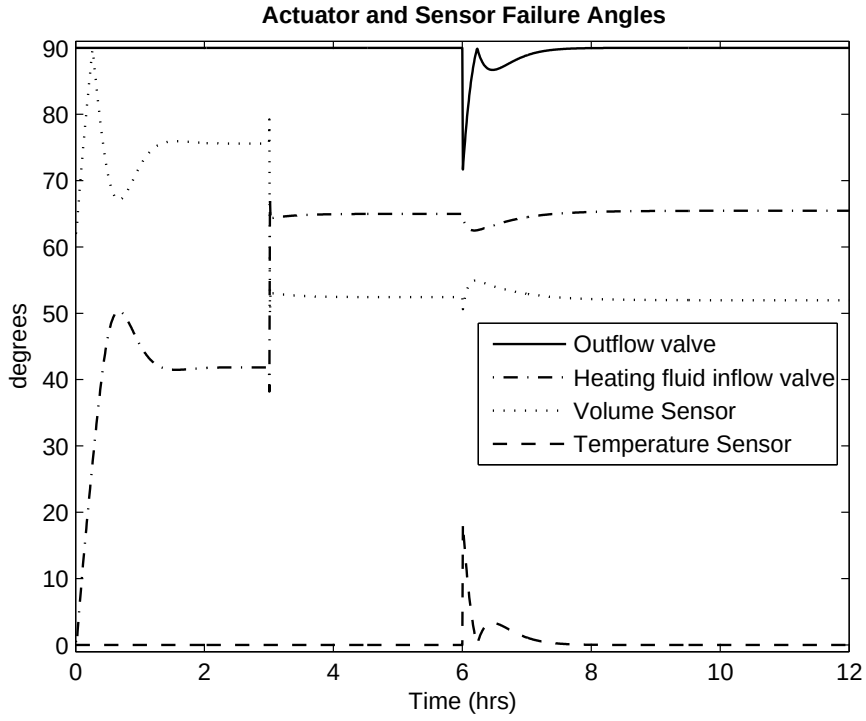

Figure 3.12: $\angle GPV$ with disturbance decoupling

$\angle GPV$ increases substantially, becoming larger than the volume sensor and outflow valve $\angle GPV$. As a result, the FDI algorithm will declare a volume sensor fault instead, yielding an incorrect isolation. To avoid these false alarms, disturbance decoupling is implemented in the FDI algorithm to maintain the GPV directionality in the presence of disturbances, as portrayed in Fig. 3.12. It is observed that after the disturbance is applied at *t= 6 hours*, the temperature $\angle GPV$ is only slightly affected for a short period of time while the disturbance is completely rejected. Similarly, the other $\angle GPV$ are just trivially perturbed and keep their undisturbed directionality, allowing a clear isolation of the temperature sensor fault. Therefore, by incorporating disturbance decoupling, the FDI algorithm becomes robust to disturbance effects [37].

3.6 Selection of σ

The real tuning parameter σ is chosen based on the real part of the eigenvalues of the state space model and determines the speed of the detection filter. Note that $\sigma > 0$ is required to ensure the stability of the parity vector filter. To illustrate the effect of σ, FDI is performed using the separator model described in section 4.3 for +1% bias fault applied to sensor S_4, the treator oil volume sensor, at t=160 sec. The FDI results were obtained using the 5^{th} order identified state space model shown later in Fig. 5.2 and calculated with GBN as excitation signals. Its eigenvalues are: $\lambda_1 = 0.0000155$, $\lambda_2 = 0.0000323$, $\lambda_3 = -0.00134$, $\lambda_4 = -0.0275$ and $\lambda_5 = -0.0695$.

The corresponding time histories for the treator and the separator are included in appendix A, Figs. A.3 and A.4. Figures 3.13 to 3.16 show the GPV magnitude and angles using the dynamic implementation for different values of σ. For very small values of sigma, the parity vector becomes too slow and takes longer to reach steady state. As a consequence, if a small fault happens before $|GPV|$ has reached steady

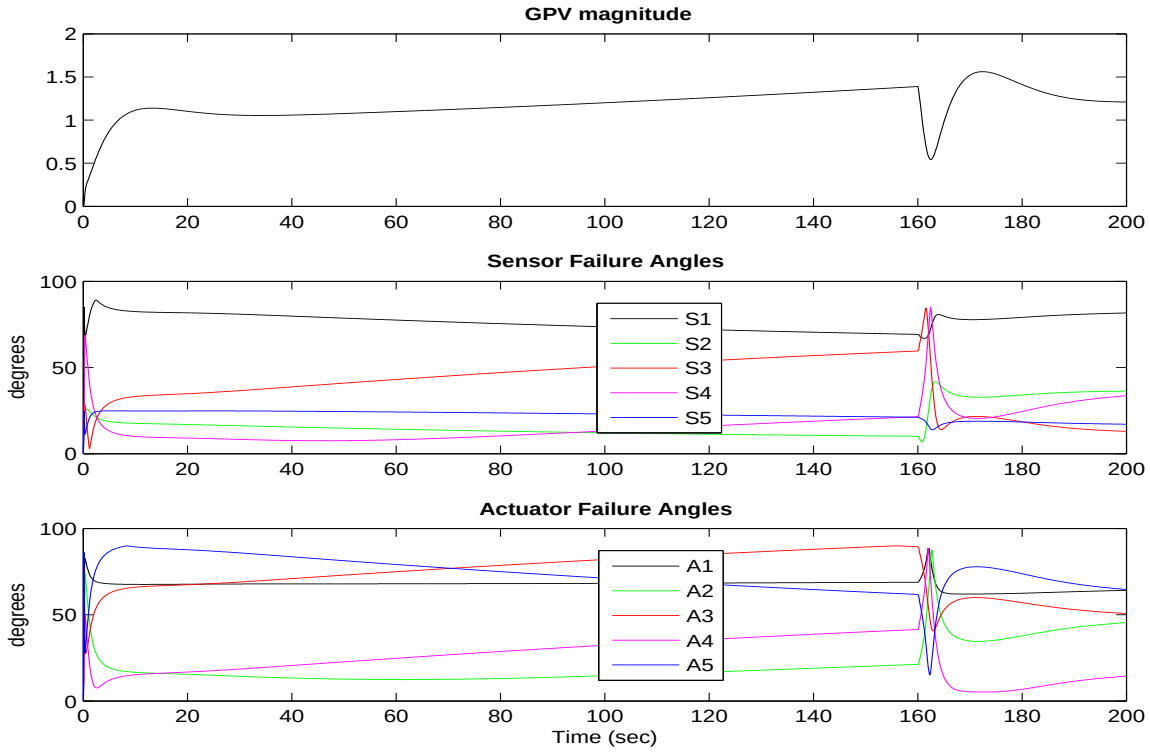

Figure 3.13: Dynamic $|GPV|$ and $\angle GPV$ using $\sigma = \lambda_1$, S_4 faulty

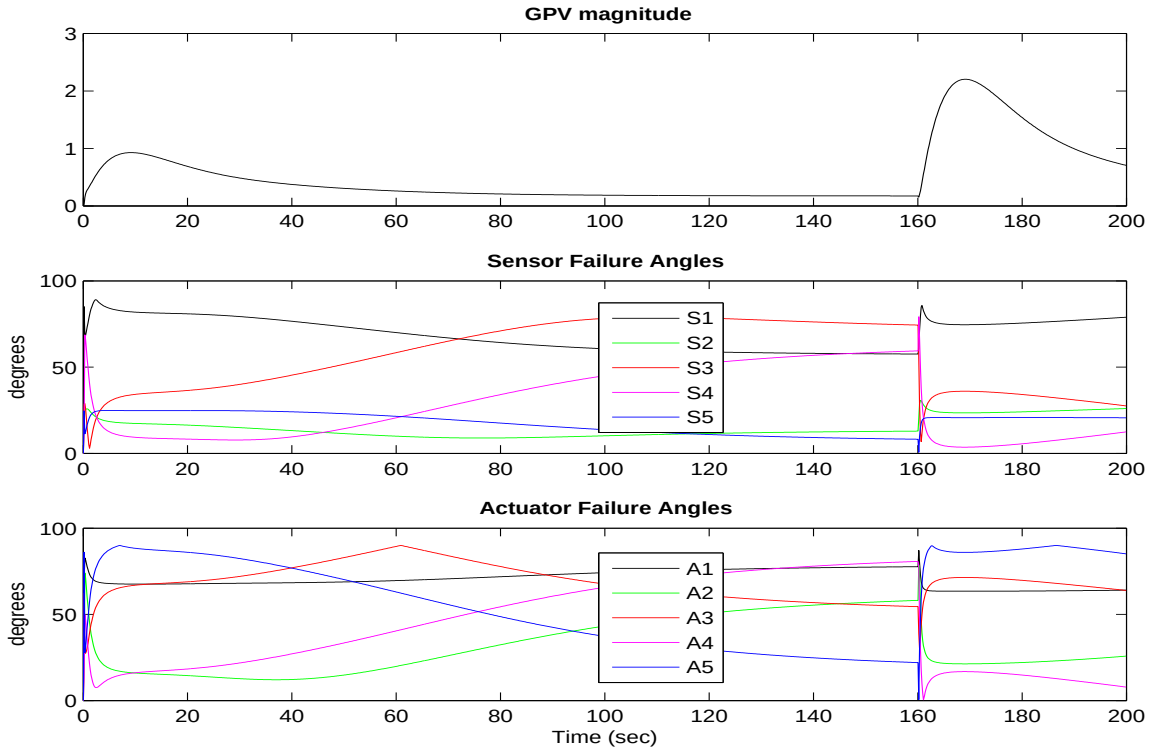

Figure 3.14: Dynamic $|GPV|$ and $\angle GPV$ using $\sigma = \lambda_4$, S_4 faulty

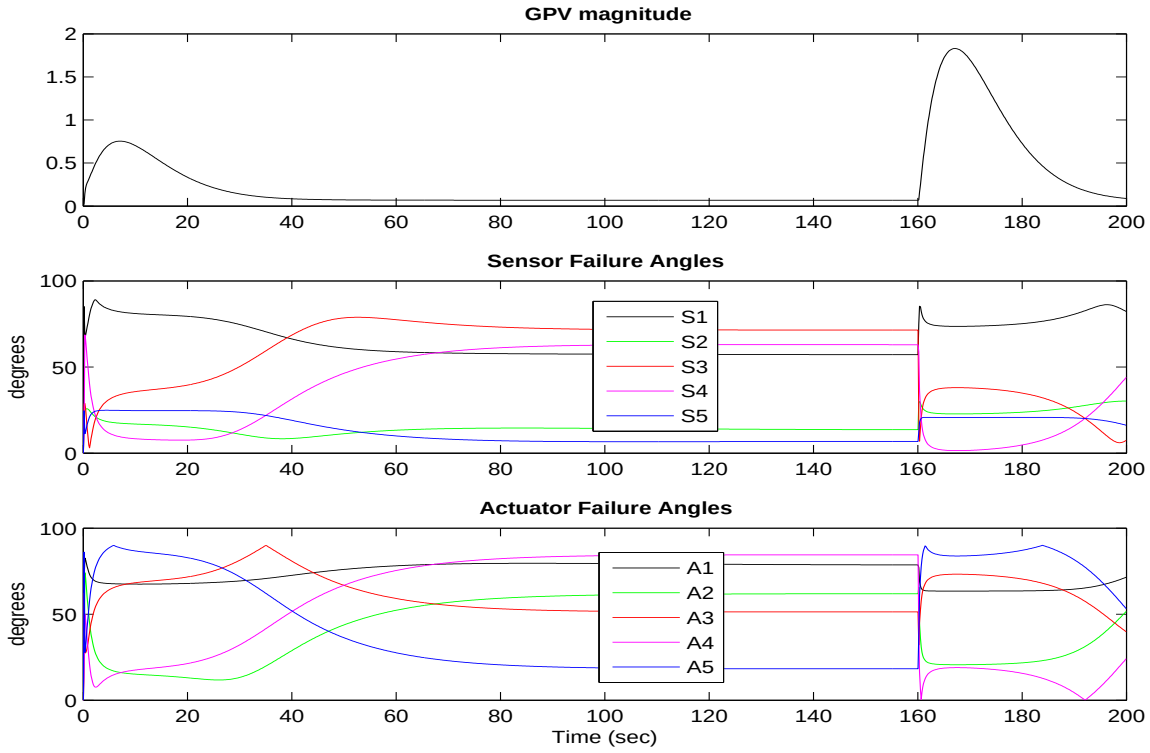

Figure 3.15: Dynamic $|GPV|$ and $\angle GPV$ using $\sigma = \lambda_5$, S_4 faulty

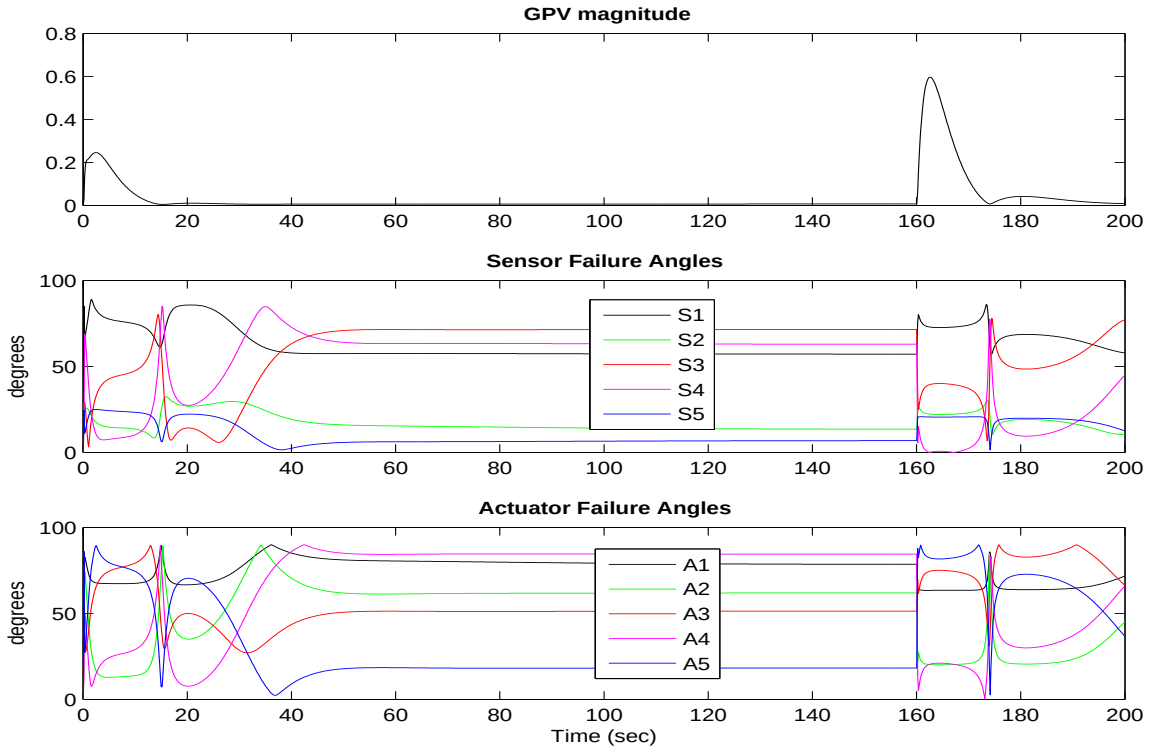

Figure 3.16: Dynamic $|GPV|$ and $\angle GPV$ using $\sigma = 10 \times \lambda_5$, S_4 faulty

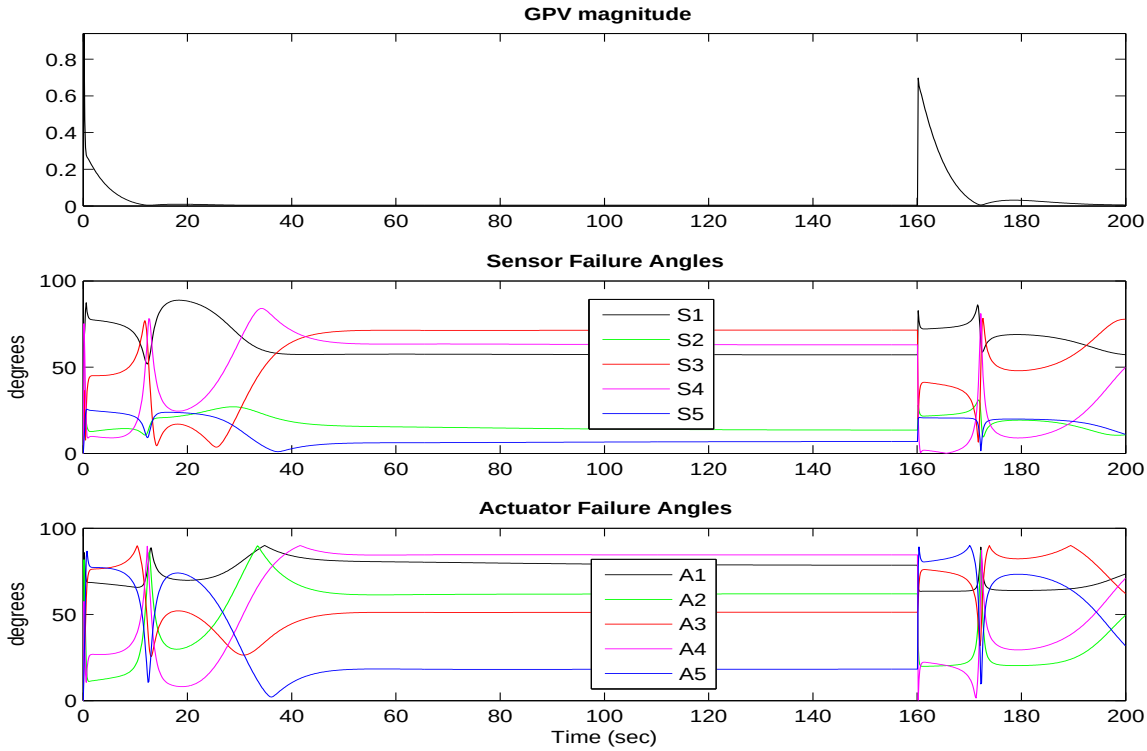

Figure 3.17: Static $|GPV|$ and $\angle GPV$ using $\sigma = \lambda_4$, S_4 faulty

state, $|GPV|$ may even decrease after the fault is applied and take very long for $\angle GPV$ to reach steady state. This results in a delayed fault detection and also unclear isolation for a long period of time, while $\angle GPV$ stabilizes. This situation is illustrated in Fig. 3.13, where even fault detection is impossible assuming a low magnitude threshold, for $\sigma = \lambda_1$, the smallest process eigenvalue. As a result, the FDIA block is unable to run for a longer period of time since it is required to wait until the dynamic GPV reaches steady state to have a proper initialization for correct FDI.

It is observed that as σ increases, the $|GPV|$ change after the fault is applied is sharper, giving a faster detection. But at the same time, if σ is too large with respect to the system eigenvalues, the $|GPV|$ becomes small rapidly, as the control loop starts rejecting the fault. This results in a wrong "fault free" diagnosis after

a short period of time, since the $|GPV|$ returns to its fault free GPV magnitude ($|GPV|_{ff}$) value although the system is still faulty. For these cases where the fault is quickly rejected by the control loop, it is desirable to have a dynamic GPV that can provide clear isolation for a period of time that is long enough to satisfy the decision-maker requirements presented later in chapter 7. Large values of σ also give short periods of ambiguous isolation due to the rougher GPV angles behaviour during the $|GPV|$ minimums as illustrated in Figs. 3.15 and 3.16.

From Figs. 3.16 and 3.17 it is seen that as σ gets bigger, the dynamic parity vector behaviour approaches the static one. It should be noticed that for the static case, the tuning parameter σ only affects the $|GPV|$ scale and not its dynamics. This was expected, since this approach gives a steady state implementation of the parity vector. From the previous results it is established that the selection of σ is a tradeoff between several factors that have different priority depending on the system specifications. For this particular state space model obtained for the three-phase separator, $\sigma = \lambda_4$ (Fig. 3.14) provided a good tradeoff, giving excellent FDIA results overall.

3.7 FDI comparison between the GPV and bank of Kalman filter methods

Several researchers have shown that a bank of Kalman filters designed on the basis of system models incorporating all possible faults can be used for isolation purpose [1], [22], [23]. This FDI approach uses multiple estimators, each of which is designed for detecting a specific fault. Since each estimator is designed based on a specific hypothesis such as the failure of a single sensor or actuator, all the estimators except the one using the correct hypothesis will produce large estimation errors when a fault occurs. By monitoring the residual of each estimator, the specific fault that

has occurred can be detected and isolated.

For the sensor fault isolation case, the bank of Kalman filters contains m Kalman filters where m is the number of sensors being monitored. Each filter estimates the state vector using $m-1$ sensors. The sensor which is not used by a particular filter is the one being monitored by that filter for fault detection. For instance, the i^{th} filter uses the sensor subset y^i that excludes the i^{th} sensor, where i is an integer from 1 to m. In the event that sensor i is faulty, all filters will use a corrupted measurement, except for filter i. Consequently, filter i is able to estimate the state vector from fault-free sensor measurements using equation (2.52), whereas the estimates of the remaining filters are distorted by the fault in sensor i.

In order to evaluate the accuracy of state estimation, the residual vector is generated for each filter using the Kalman estimation error or residual in equation (2.51). From this residual, a weighted sum of squared residual (WSSR) is computed. The fault indicator signals (i.e., $WSSR_i$ for the i^{th} filter) will be compared against the detection threshold θ in order to detect a sensor fault; if $WSSR_i < \theta$ then sensor i is faulty.

Actuator FDI using the bank of Kalman filters method is more challenging than sensor FDI. In the general Kalman filter approach, it is assumed that the actuators are properly set to the values that a control system demands. However, if a large discrepancy between commanded and true actuator positions does exist due to an actuator fault, it can result in significant state estimation errors. In this method, an actuator fault is modeled as a bias, which results in an inconsistency between an actuator command used as a Kalman filter input and a true actuator position under which the plant is operating. To account for a potential bias, the actuator bias vector b is added to the actuator command inputs.

A bank of Kalman filters is also applied for actuator FDI, although their structures

are different from those for sensor FDI. The filter for actuator j FDI will use all m sensors and estimate an augmented state vector, which includes an actuator j bias. Using the augmented state space structure, a Kalman filter is designed for each of the k actuators using equation (2.52). After the estimation of the augmented state variables and sensor measurements, a fault indicator signal is generated for each filter similar to the sensor FDI approach using equation (2.51). The fault indicator signal ($WSSR_j$) will be compared against a given detection threshold in order to detect an actuator fault. When an actuator is biased, all filters except filter j use corrupted information, however, the j^{th} filter with the correct hypothesis is able to accommodate it. Therefore, this particular filter will maintain a low residual value and consequently can be isolated from the rest of the filters. For more details on sensor and actuator FDI using a bank of Kalman filters see [22], [23].

From this overview of FDI using a bank of Kalman filters it is clear than this method requires the implementation of $m + k$ filters to be able to generate the number of residuals needed to provide isolation for m sensors and k actuators. Therefore, when a large-scale process is considered, the size of the bank of filters can be very large, thus increasing significantly the computational complexity. Conversely, FDI using the GPV methodology proposed in sections 3.2, 3.3 and 3.4 is capable of isolating $m + k$ sensor and actuator faults by exploiting the direction of only one residual. Recall that it was already established in section 2.5 that the definition for the GPV and Kalman filter error are equivalent in form and consequently it takes the same computation time to run either of them. Therefore, since FDI using the Kalman filter method requires $m + k$ filters, it takes $m + k$ times longer to run than FDI using only one parity vector. This longer computation time becomes a crucial limitation when it comes to real-time FDI implementation in fast industrial processes. For these type of scenarios, FDI using the GPV technique will provide a more feasible implementation.

Chapter 4

System Overview

4.1 Introduction

The purpose of this research is to develop an FDIA technique and implement it as an agent to be an important actor, as part of the ICAM supervisory system intended to be implemented on a pilot plant of the PAWS project, which emulates an offshore oil production facility. An SID module has also been built to overcome the model availability issue and also, to improve the FDIA accuracy at different operating points. So far the mathematical bases for FDI using a parity equation implementation of directional residuals have been presented. In this chapter, FDIA using the GPV technique and its relation with the SID and data reconciliation agents and the ICAM system supervisor are discussed, to illustrate the significance that the work presented in this thesis has in the performance of the entire system.

The FDIA and SID agents, which are the main contributions of the present work, are presented in detail in the following chapters. Here we overview the PAWS project, the ICAM system supervisor proposed and implemented by Taylor and Sayda [34], [46], [47], [48] and the dynamic data reconciliation (DDR) agent developed by Laylabadi [49], [50] and being extended and implemented by Moreno [51], to provide

the context and framework for presenting the components contributed by this research and development effort. A more detailed discussion of contributions is given in chapter 10, after all the technical details are presented.

4.2 The PAWS project

Driven by the technical and economic demands of the offshore oil and gas industry in Atlantic Canada, a joint venture between several Atlantic Canadian universities, the National Research Council of Canada, and international companies was established in order to advance wireless systems technology in the oil and gas industries and to assess the feasibility of an ICAM system built on a wireless sensor network. The PAWS project scope is to develop a control and information management system which consists of two subsystems. The first subsystem is a wireless sensor network which will alleviate the need for data cables in offshore oil rigs and improve flexibility for adding and reconfiguring sensors [47].

The second subsystem intelligently manages the massive data flow from oil rigs and interprets it so as to help operators take more appropriate decisions during normal operation and abnormal events and, through intelligent control, improve process economics. As part of the PAWS project, a practical intelligent multi-agent system has been developed by Taylor and Sayda [46], [47], [48], [34] to manage the massive information flow from offshore oil rigs. Multi-agent systems (MAS), which can be considered as an instantiation of distributed artificial intelligence, is another conceptual framework for modeling and implementing complex systems. A MAS is defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities [52]. The MAS platform emphasizes distribution, autonomy, interaction (i.e., communication), coordination, and organization of individual agents. Agents in a MAS can be defined as con-

ceptual entities that perceive and act in a proactive or reactive manner within an environment where other agents exist and interact with each other based on shared knowledge of communication and representation [53]. Each agent contains processes for behaviour generation, world modeling, sensory processing, and value judgment together with a knowledge database. The final developed system will be deployed and validated on a pilot plant which emulates an offshore oil production facility [46].

4.3 PAWS application: a gravity three-phase separator

Three-phase separators, as depicted in Fig. 4.1, are designed to separate and remove the gas and free water from the mixture of crude oil, water and natural gas. The College of the North Atlantic (CNA) has a three-phase separator pilot plant that is fully instrumented and under five loops of automatic control. This facility has been selected as the primary test-bed for the PAWS project. In order to facilitate research and testing at UNB, a detailed nonlinear model of a three-phase gravity separator was developed by Sayda and Taylor [47] for use as a simulation test-bed under MATLAB®. The process description in this section is a synopsis taken from [47] and it is presented here to establish another aspect of the context for this research.

The simulation model corresponds to these two process steps shown in Fig. 4.1. The first process represents the two-phase separator (first vessel), in which hydrocarbon fluids from oil wells are separated into two phases to remove as much light hydrocarbon gases as possible. The produced oil and water mixture is then pumped to the second process which corresponds to the three-phase separator (second vessel), which models how water and solids are separated from oil. The gas that is separated is compressed and treated for sales and the produced oil is then pumped out and sold to refineries and petrochemical plants if it meets the required specifications.

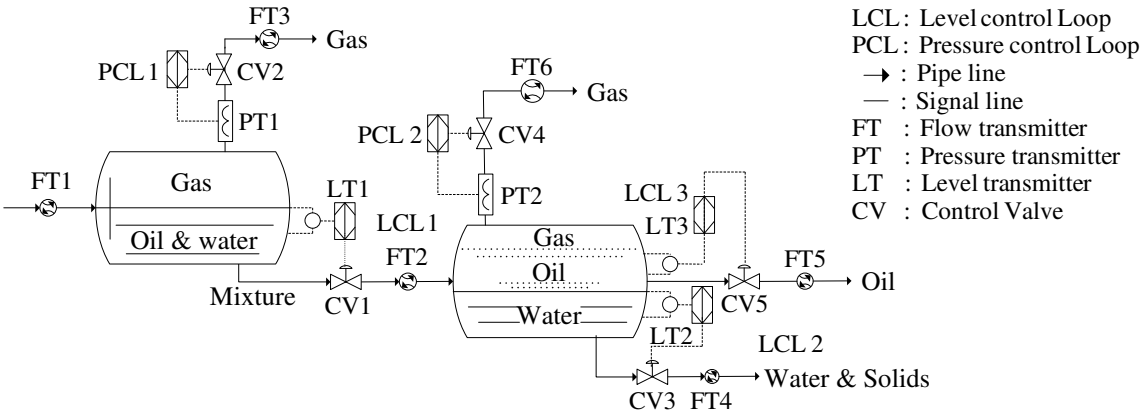

Figure 4.1: Gravity three-phase separator process

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 Fig. 4.1, 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. The oil level is controlled by the second PI controller through the oil discharge valve, and the second vessel pressure is maintained constant by the third PI loop [47].

4.4 The ICAM system

In order to have the ICAM system requirements deployed in a real-world application, a prototype has been developed by Taylor and Sayda [34]. In this prototype, data from the external plant (the pilot plant at CNA) or the simulation model of the separation process described in section 4.3 is received by the DDR agent, which pre-processes the data by removing undesired discrepancies such as outliers and missing

data and reconciles the measurements to reduce noise and enforce dynamic constraints [49], [50]. Processed data is stored in a real-time database for logging and other purposes, and is then sent to the SID and FDIA agents for further processing.

When a change in the operating point is detected, the SID and FDIA agents are alerted to further identify the nature of the data change. If a significant change in the process operating point occurs, the ICAM system supervisor asks the SID agent to update the process model parameters. The SID agent also notifies the supervisor of the quality of fit of the identified model, so other agents (FDIA, DDR) can asses their own capabilities. If the change is a process fault (i.e., a sensor or actuator failure), the FDIA agent detects, isolates and, if possible, accommodates the fault and notifies the ICAM system supervisor for further processing. For every event that occurs, the supervisor is notified, which in turn monitors, directs, and assesses the logical behaviour of the system. Processed data from every agent is assessed by the supervisor and sent to an operator interface, which allows operators make the appropriate decision depending on the plant situation.

The FDIA agent exploits the GPV technique to generate a set of directional residuals, from which process faults can be determined. This agent plays a crucial role in the ICAM process since it diagnoses the system health and provides accommodation to keep the safe operation in the plant under failure situations. The data reconciliation agent filters the measurements and removes missing data and outliers by exploiting the median absolute deviation algorithm [54]. The SID agent estimates the multivariable linearized state space plant model by using the standard prediction error/maximum likelihood method *(pem)* implemented in MATLAB®. The task of this agent is very important since it provides the plant model which is the basis for FDIA and DDR, and also generates the quality-of-fit metrics for diagnostic purposes. The ICAM system supervisor is a G2 real-time expert system [52], which codifies the ICAM system internal and external behaviour in its knowledge base.

4.5 Data reconciliation agent

In general, measured data in chemical processes are subject to be corrupted by noise. In addition, sensor data may not obey the laws of conservation of mass and energy of the system, due usually to instrumentation errors or process disturbances. The procedure of adjusting the measurements to meet the laws of conservation and dynamic constraints is known as data reconciliation [51]. The general nonlinear dynamic data reconciliation (NDDR) formulation can be expressed as follows [55]:

$$\min_{\hat{y}(t)} \Phi[\tilde{y}, \hat{y}(t); \sigma], \tag{4.1}$$

Subject to

$$f[\frac{d\hat{y}(t)}{d(t)}] = 0, \tag{4.2}$$

$$h[\hat{y}(t)] = 0, \tag{4.3}$$

$$g[\hat{y}(t)] \geq 0, \tag{4.4}$$

where

$$\begin{array}{rcl} \tilde{y} & = & \text{corrupted measurements,} \\ \sigma & = & \text{measurement noise standard deviations,} \\ f & = & \text{process dynamic constraints,} \\ h & = & \text{energy and/or material balance constraints,} \\ g & = & \text{process variable limits,} \\ \hat{y}(t) & = & \text{reconciled measurements.} \end{array}$$

The lengths of $\hat{y}(t)$, $\tilde{y}$ and σ are equal to the total number of variables (states and inputs). Operators f, h and g depend on the specific problem. Most of the applications use a simple weighted least-squares as the objective function. Solving the general NDDR problem requires optimizing an objective function by adjusting the estimated measurements, which are constrained by differential and algebraic equalities and inequalities [55].

The DDR agent proposed in [51] is also in charge of detecting steady state conditions in the process. The method presented in [51] for steady state detection performs

a linear least square regression over a moving data window, whose length depends on the time constants of the variables. Once the parameters of this regression are obtained, the attention is focused on the behaviour of the slope of this line. If the slope is smaller than a threshold, steady state is declared. This threshold is chosen based upon the setpoint variation and the standard deviation of the noise.

4.6 FDIA agent

FDIA using the generalized parity vector technique and its relation with the SID and data reconciliation agents and the ICAM system supervisor are presented in this section, based on the block diagram shown in Fig. 4.2. The details for the different blocks in the FDIA agent are discussed in the following chapters; here we overview the entire system to provide the framework for presenting the components.

If there is no model available or there has been a setpoint change that is large enough to make the previous model invalid, the system identification module presented in section 5.2 and the initialization section are performed using the reconciled inputs-outputs measurements sent by the data reconciliation agent [49], [50] [1]. The validity of the model is determined based upon the severity of nonlinearities in the system; this is learned from simulation experience. The resulting model is the basis for FDIA.

The initialization section starts with the calculation of the corresponding coprime factors, reference directions and hyperplanes based on the identified state-space model. Using these directions, the transformation matrix calculation block described in section 6.4 is executed to obtain an optimal T_r that maximizes the minimum separation angle between all the reference directions and hyperplanes. This block is recursively computed until the minimum GPV angle ($\angle GPV_{min}$) obtained during the

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

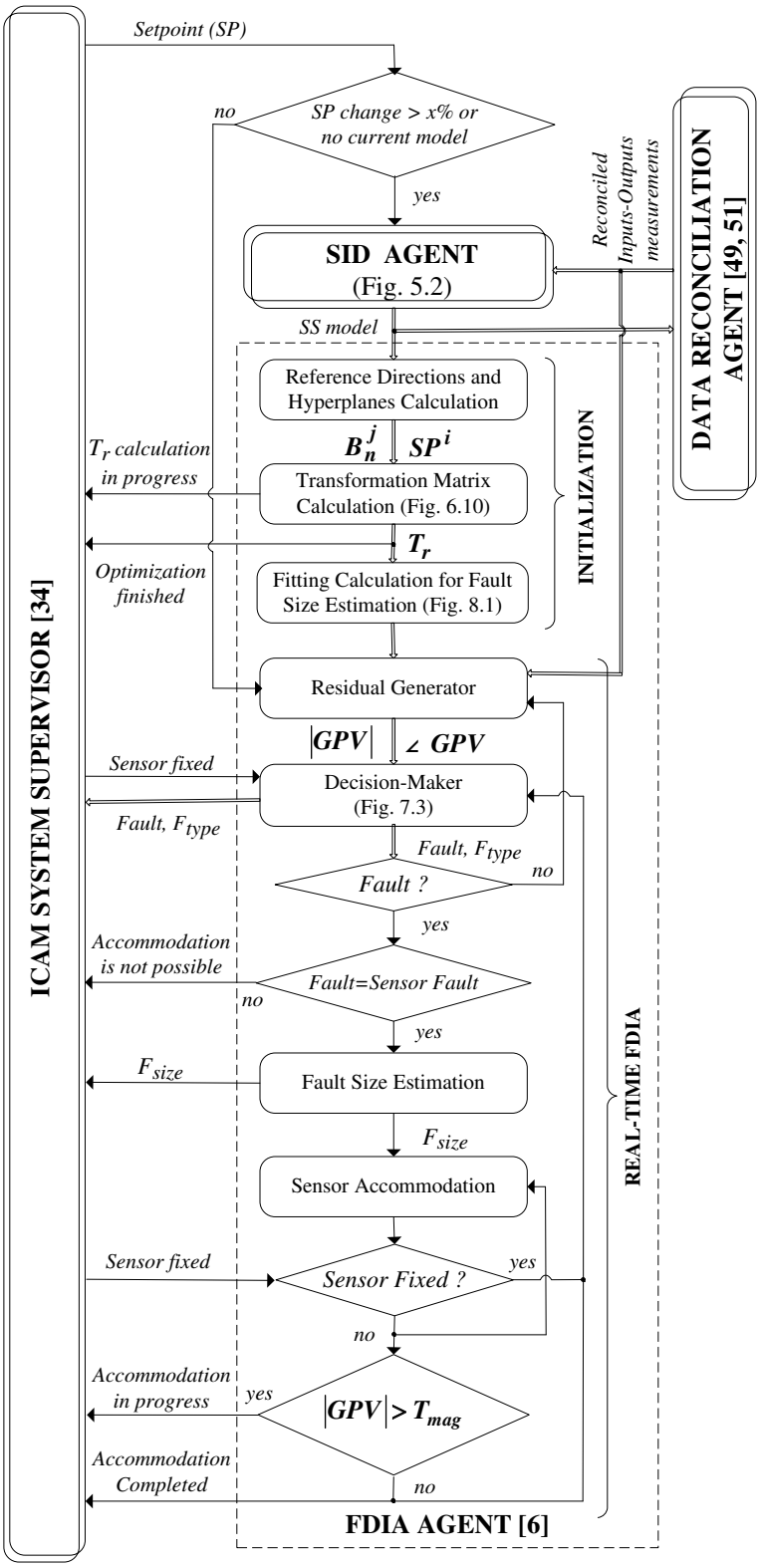

Figure 4.2: FDIA block diagram

optimization process is greater than a prespecified isolation angle threshold (Θ_T) or the maximum number of trials is exceeded. The ICAM system supervisor is notified while the T_r is being calculated and when it has been successfully computed.

With the previous information, the fitting calculation block illustrated later in chapter 8 (Fig. 8.1) is performed. For the bias case, the fault size (F_{size}) is calculated based on the change in static GPV magnitude peak ($\Delta|GPV|_{peak}$). Conversely, for the ramp case, F_{size} is computed using the static GPV magnitude slope ($|GPV|_{slope}$) after the fault is detected. In both cases, different fault-size scenarios are simulated to obtain the corresponding $\Delta|GPV|_{peak}$ vs. F_{size} or $|GPV|_{slope}$ vs. F_{size} pairs. Using these sets of data, the best low-degree polynomial fitting is calculated for each case, providing an equation for F_{size} as a function of $\Delta|GPV|_{peak}$ or $|GPV|_{slope}$, depending on the fault type declared previously. The complete theory for the proposed fault size estimation method is presented in sections 8.2 and 8.3. This is the end of the initialization section, which is executed only after the state space model changes. The fact that the initialization section is performed only once and the total computation time in a Pentium 4 PC is 10.48 minutes (including the SID block execution), makes the proposed on-line implementation viable for operation in real industrial processes.

The initialization section adds a lot of flexibility to the GPV technique in terms of fault size estimation and sensor accommodation. It allows the efficient on-line computation of the transformation matrix for different state space models, and also the calculation of the corresponding set of curve fitting equations for fault size estimation. These blocks are discussed in detail in chapters 6 and 8 respectively.

Once the initialization section finishes, the GPV magnitude and angles are computed by the static residual generator for each input-output set at every sample. Then, the decision-maker block described in chapter 7 detects, isolates and classifies the fault

based on the static GPV magnitude ($|GPV|_{st}$), the GPV direction match ($\angle GPV_{min}$) and the $|GPV|_{slope}$ and sends the diagnosis to the ICAM system supervisor. 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. The fault size is computed using the polynomial equations obtained previously by the fitting calculation block. 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 loses in production. The accommodation block is continuously performed until the "sensor fixed" message is received from the ICAM system supervisor.

Chapter 5

System Identification Agent

5.1 Introduction

Chemical processes are generally higher-order and nonlinear in nature, making it very difficult and impractical to derive an accurate mathematical model for the system. This has been one of the main reasons that has limited the application of quantitative model-based FDI techniques in real industrial processes [3], while they have been widely used in aerospace applications. So far, the GPV technique had been successfully implemented only using well defined mathematical models, for the GE-21 jet engine [13] and jacketed continuously-stirred tank reactor [5], [37].

In order to address the model availability issue for the practical application of the GPV technique to industrial processes, a system identification module was implemented in this research for the gravity three-phase separator described in section 4.3, which closely emulates the CNA pilot plant of the PAWS project. The fact that an identified model can be used for FDIA using the GPV approach, as shown in this research, represents a significant extension in the scope of application of this technique. It should be noticed that this model took six months of research and development effort to create; this underscores the importance and challenge of working

with identified models in practical applications.

In this chapter some identification results are presented with the general guidelines to implement an on-line SID module (agent) for any process. The SID approach is established in the literature [56]; the contribution here is demonstrating how to "package" such techonlogy to make it useable in an automated agent-based system.

5.2 System identification

The purpose of identification tests for control is to excite and to collect control relevant information about the process dynamics and its environment (disturbances). Often several different types of tests need to be performed, each of them for collecting specific information about the process. The model is estimated from the final tests where the process inputs are perturbed by some test signals. These excitation signals must be carefully designed so the normal operation in the plant is preserved. Three aspects are important in selecting test signals for SID, the first is the shape or waveform, the second is its power spectrum or frequency content and the third is amplitude.

In order to have data with high signal to noise ratios, one needs to have as much excitation signal power as possible. In practice, on the other hand, the test signal amplitudes are constrained because they should not disturb the normal process operation and should not excite too much the nonlinearities for linear model identification. Therefore, for a given signal power, a small amplitude is desirable. There are several types of test signals commonly used, such as pseudo random binary sequences (PRBS), filtered white noise or autoregressive moving average (ARMA) processes, sums of sinusoids, or generalized binary noise (GBN) among others [56]. In this research we were able to identify adequate models using both PRBS and GBN as excitation signals. However, the GBN signals were easier to design and gave more

consistent results overall. This was expected, given the advantages of the GBN signals with respect to the PRBS which are discussed in more detail in section 5.2.1. On the other hand, the PRBS signal design was more empirical, and it sometimes provided poor quality models just with small variations in the PRBS signal parameters.

5.2.1 Generalized binary noise signals

This signal was proposed by Tullenken (1990) [57]. The motivation was to generate a test signal that is suitable for control relevant identification of industrial processes. A GBN signal *u(t)* takes two values, -*a* and *a*. At each candidate switching time *t*, it switches according to the following probabilistic rule:

$$\begin{aligned} P[u(t_k) = -u(t_{k-1})] &= p_{sw} \\ P[u(t_k) = u(t_{k-1})] &= 1 - p_{sw} \end{aligned} \tag{5.1}$$

where p_{sw} is switching probability. The distribution for the event at each switching time is an independent uniform distribution with parameter p_{sw}. Because of this, the GBN has zero mean. Define the minimum switching time T_{min} as the time (in samples) to keep the signal constant and switching time T_{sw} as the elapsed time (in samples) between two switches, then the mean switching time is determined by:

$$E[T_{sw}] = \frac{T_{min}}{p_{sw}} \tag{5.2}$$

and the GBN power spectrum is given as:

$$\Phi_u(\omega) = \frac{(1-q^2)T_{min}}{1 - 2q\cos(T_{min}\omega) + q^2} \tag{5.3}$$

where $q = 1 - 2p_{sw}$. For more details on the GBN test signals formulation see [57].

In general, white noise signals are not a good choice of test signals as they do not have adequate low frequency content. However, low-pass GBN signals are generated by reducing the switching probability p_{sw}, or, equivalently, increasing the mean

switching time $E[T_{sw}]$. Although low-pass PRBS and GBN signals look similar in the time domain, the spectrum of a low-pass GBN signal does not have "dips" at frequencies $2\pi/T_{clk}$, $4\pi/T_{clk}$, ..., where T_{clk} denotes the clock period in the PRBS generator; this spectral behaviour is advantageous. Another advantage with GBN is that the signal length is flexible. GBN signals can be generated using realizations of independent uniformly distributed random variables in the set [0,1] [56].

After the type and the amplitudes of the test signals are determined, it is then important to chose the proper spectral distributions. The signal spectra should be designed in such a way that the identified model is best for the intended use of the model. If only the error caused by disturbance (variance error) is considered, then it is possible to derive an optimal spectrum $\Phi_u^{opt}(\omega)$ as follows [56]:

$$\Phi_u^{opt}(\omega) \approx \mu\sqrt{\Phi_u^c(\omega)\Phi_v(\omega)} \tag{5.4}$$

where $\Phi_v(\omega)$ is the spectrum of disturbance, $\Phi_u^c(\omega)$ is the spectrum of the control signal and μ is a constant adjusted so that the input power is constrained. This formula tells us that the power of the test signal should be high at frequencies where the power of the control signal is high, in order to mimic the situation where the model is used. The power should also be high when the disturbance power is high, in order to counteract it. Guided by this formula, the following design rule for the mean switching time of the GBN signal is derived [56]:

$$E[T_{sw}] = \frac{98\%\ settling\ time}{3} = \frac{4\tau}{3} \tag{5.5}$$

where τ is the corresponding variable's time constant. The reasoning behind this formula is that the spectra of both the disturbance and the control signal are determined by the bandwidth of the process. The factor 1/3 is obtained from simulation exercises and project experience. Using this formula the mean switching time is a simple function of process settling time or time constant (which are assumed to

be known from pre-test data or from operational experience) corresponding to the operating point chosen for linearization [56].

5.2.2 Identification test design

In practice, it is often desirable or even necessary to carry out identification tests in closed loop operation. For most industrial processes where the controllers are already in place and operational, it will cause production loss or even safety problems if the loops are opened. In such cases the only option is to perform the identification tests in closed loop. Moreover, even when open loop testing is viable, there are many advantages to closed loop testing [56] for example, when a multivariable open loop test is implemented, some of the outputs may drift and the operator may need to intervene in order to prevent product quality deterioration, while in a closed loop test, the amplitude of the setpoint movement can be specified and the controller should keep the outputs within their operation limits. Closed loop tests are also easier to carry out, considering the need for disturbance reduction: In an open loop test, operator intervention may be necessary in order to keep the process outputs in range, though manual control can be a difficult task when many inputs are excited. For these reasons, closed loop tests are more acceptable to operators.

In this research system identification is performed for the gravity three-phase separation process described in section 4.3 with the intent of demonstrating that the proposed FDIA approach is viable for industrial implementation in gas and oil facilities. Given the advantages presented above, the system identification results presented in section 5.4 were obtained using a closed loop identification test which makes this approach more feasible for real industrial implementation.

The following aspects should be taking into account in designing the closed loop identification test:

- It is assumed that the range of normal operation (or the high/low limits of process outputs) and time constants are known *a-priori* from operator knowledge or pre-test.

- The test time can be set in the range of 6 to 18 times the process maximum settling time (t_s). A minimum length of test time is necessary to make use of available theoretical results of identification which are often asymptotic in the number of data points N (assuming $N \rightarrow \infty$). However, it should be kept in mind that long test times are not desirable because the plant is being disrupted and also, the model computation time is longer, delaying the FDIA agent initialization. The test time can be decreased to 5 to 8 times t_s, if the number of inputs is small and if the signal-to-noise ratio is high, or even to 1 to 2 times t_s, if the process is linear and almost noise free. Conversely, the test time should be longer, 14 to 18 times t_s, if there are many inputs and the signal to noise ratio is low.

- When it comes to the signal shape selection, the GBN signals are highly recommended because of the features presented in section 5.2.1 and also, the fact that operators are familiar with binary test signals. The amplitudes of the test signals should be chosen such that the signal to noise ratio is reasonably high and, at the same time, the process operation is not disturbed too much and it does not drive the process out of a nearly linear operating range [56]. This requires judgment and some knowledge of the process dynamics.

5.3 On-line identification of systems

Figure 5.1 illustrates the on-line implementation for the SID agent introduced previously in Fig. 4.2. The SID agent is executed only if there is no model for the process or there is a significant setpoint change that makes the previously identified

model invalid. First, the excitation signals are generated based on the GBN theory presented in section 5.2.1. Since it is assumed that the process time constants are known by the ICAM system supervisor it is possible to use equation (5.5) as an on-line design rule to chose a suitable mean switching time $E[T_{sw}]$. For the separator model described in section 4.3 the time constants were obtained from simulation results.

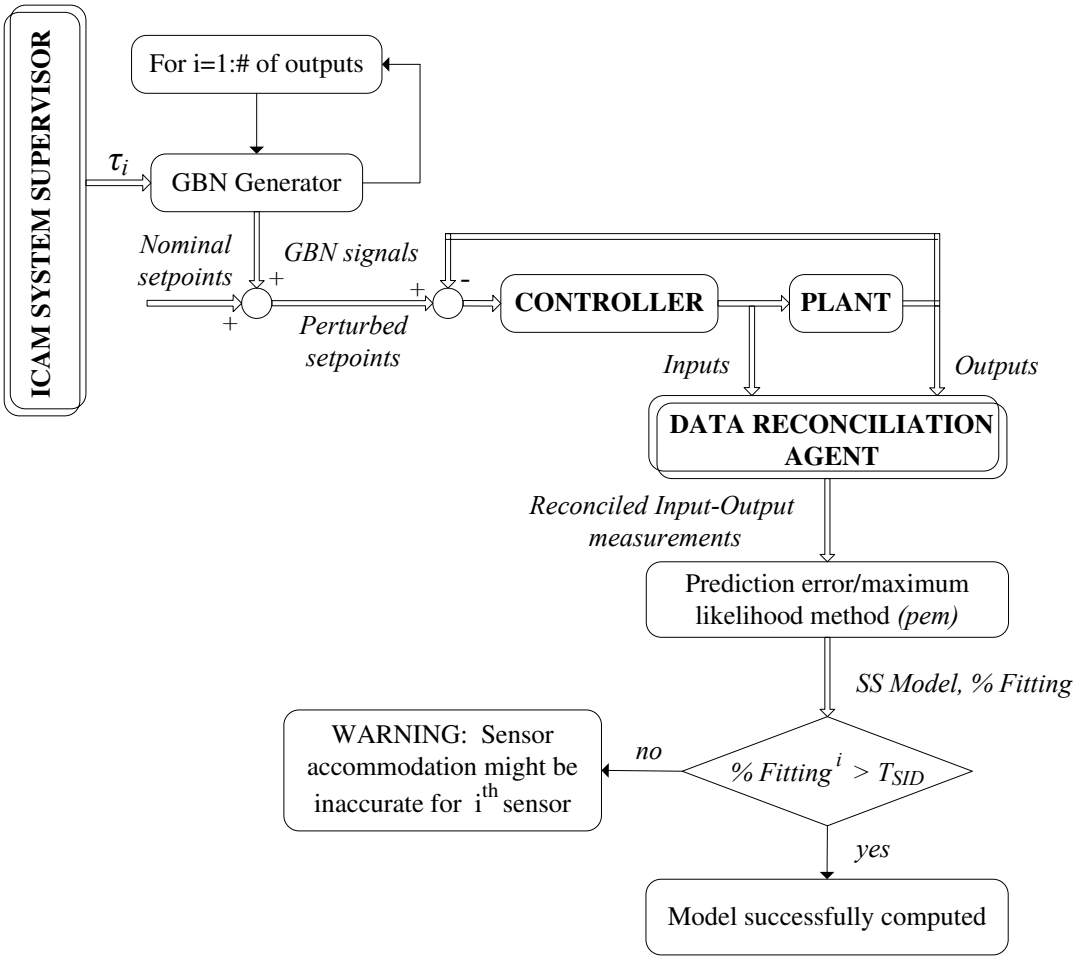

Figure 5.1: SID agent block diagram

We have chosen the GBN excitation signals to design the identification test because the FDI results using this identified model were slightly better overall than the results obtained using the model identified with PRBS as excitation signals. This will be discussed in more detail in section 5.4. Once the GBN signals are generated, they are

applied as setpoint variations to excite each control loop and generate the controller output (plant inputs) and the plant outputs. These input/output measurements are preprocessed by the data reconciliation agent before system identification is performed, to reduce noise.

Then, the linearized state space model and the corresponding % of fitting for each output are calculated using the prediction error/maximum likelihood method *(pem)* in MATLAB® for the reconciled input and output measurements. If the % of fitting in any of the outputs is less than a prespecified threshold (T_{SID}), a warning stating that the "Sensor accommodation might be inaccurate for the i^{th} sensor" is displayed. For instance, for the GBN model illustrated in Fig. 5.3 the following warning message is displayed for sensor 4:

"System identification in output 4 has a low % of fitting: 76.98%. As a result sensor accommodation might be inaccurate for sensor 4".

From numerous simulation results using the separator model it was established that fault detection and isolation are quite tolerant to modeling errors (imperfect % of fitting). However, fault size estimation, and therefore sensor accommodation, are more sensitive to it. From experimental results, a threshold $T_{SID} = 85\%$ was found to give good sensor accommodation results overall. If the % of fitting in all the outputs is greater than T_{SID}, the model computation is considered successful.

5.4 System identification results

The system identification results presented in this section were obtained using the gravity three-phase separator nonlinear model described in section 4.3 to simulate the plant and generate the input/output measurements. However, it is assumed that the actual nonlinear model for the plant is not available and also the system order is unknown, so an analytical linearized state-space representation cannot be

obtained. The only available information is input-output data corresponding to PRBS or GBN signals applied as setpoint variations to excite each of the system reference inputs. For both cases, the amplitude of the excitation signals is +/-1% of the nominal value, so the normal operation in the plant is only slightly disturbed and the process is not driven out of the linear range. On-line system identification is performed based on the logic depicted in Fig. 5.1 and using the standard prediction error/maximum likelihood method *(pem)* implemented in MATLAB®, which provides a linearized state space model. Since the model order is not specified, this is calculated automatically using the subspace-based method in MATLAB®*(n4sid)*, to provide the best fitting.

The first 600 sec. of the input/output measurements are used to identify the model, while the last 300 sec. are used to validate it. The test time of 600 sec. was chosen based on the guidelines given in section 5.2.2 and simulation experiments which provided good identification results for several different trials. This is approximately 6.4 times the longest process settling time, which fits into the 5 to 8 times range given the assumption that the signals are essentially noise free (because they are received from the data reconciliation agent) and the number of inputs is not very large. Figures 5.2 and 5.3 show the system identification validation results for each output variable using PRBS and GBN as excitation signals respectively.

The "best model order" obtained during the identification process using PRBS signals is 10, which is considerably higher than the actual order of five. Nevertheless, this order discrepancy allows us to test the GPV technique using a state space model whose state variables do not correspond to physical parameters as in the original nonlinear model. After many trials to identify the model using different PRBS signals it was not possible to find a 5^{th} order model that provides satisfactory FDI performance. Even using the model shown in Fig. 5.2 which has very high % of fitting, the FDI results were slightly better overall using the identified model obtain with

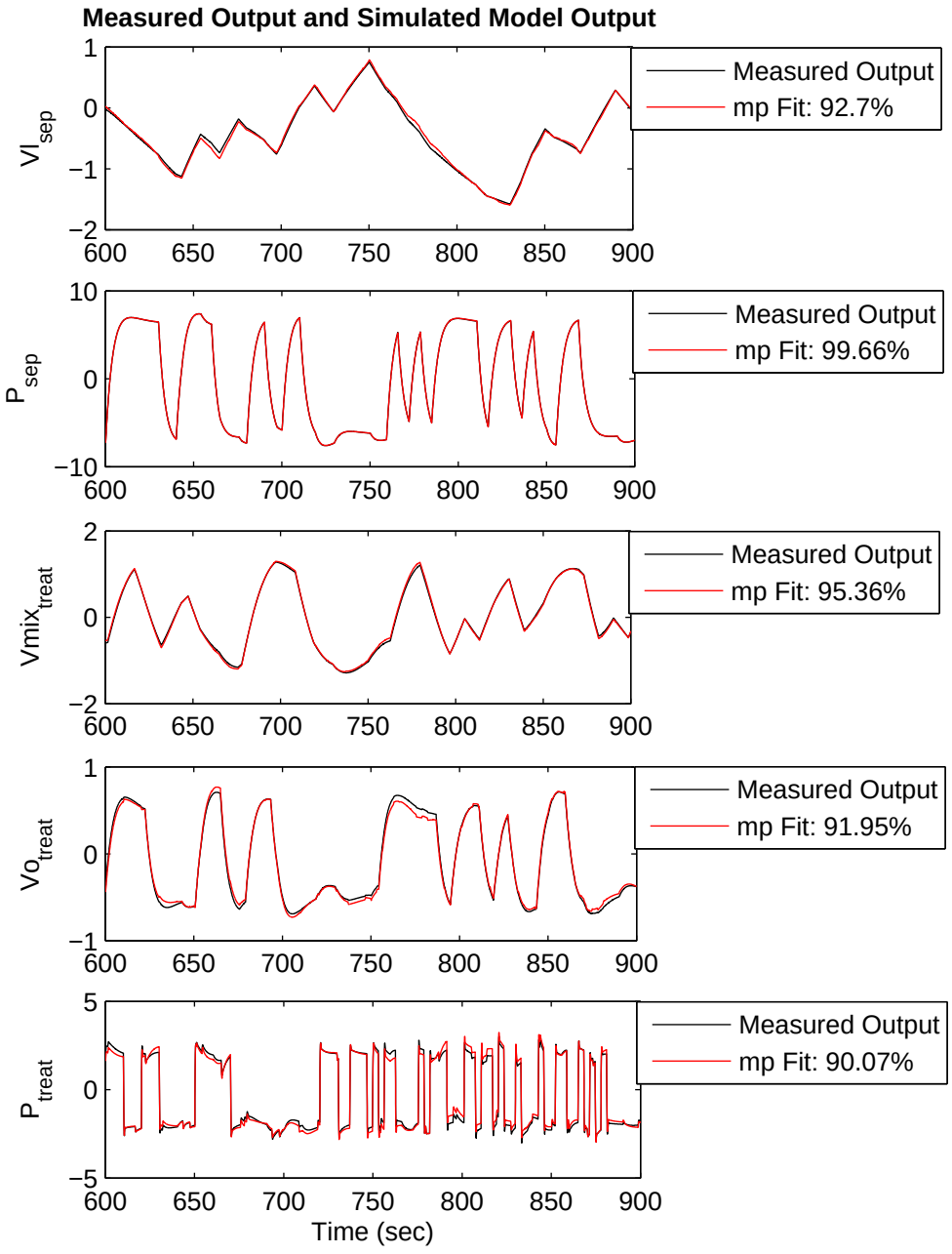

Figure 5.2: System identification results using PRBS signals

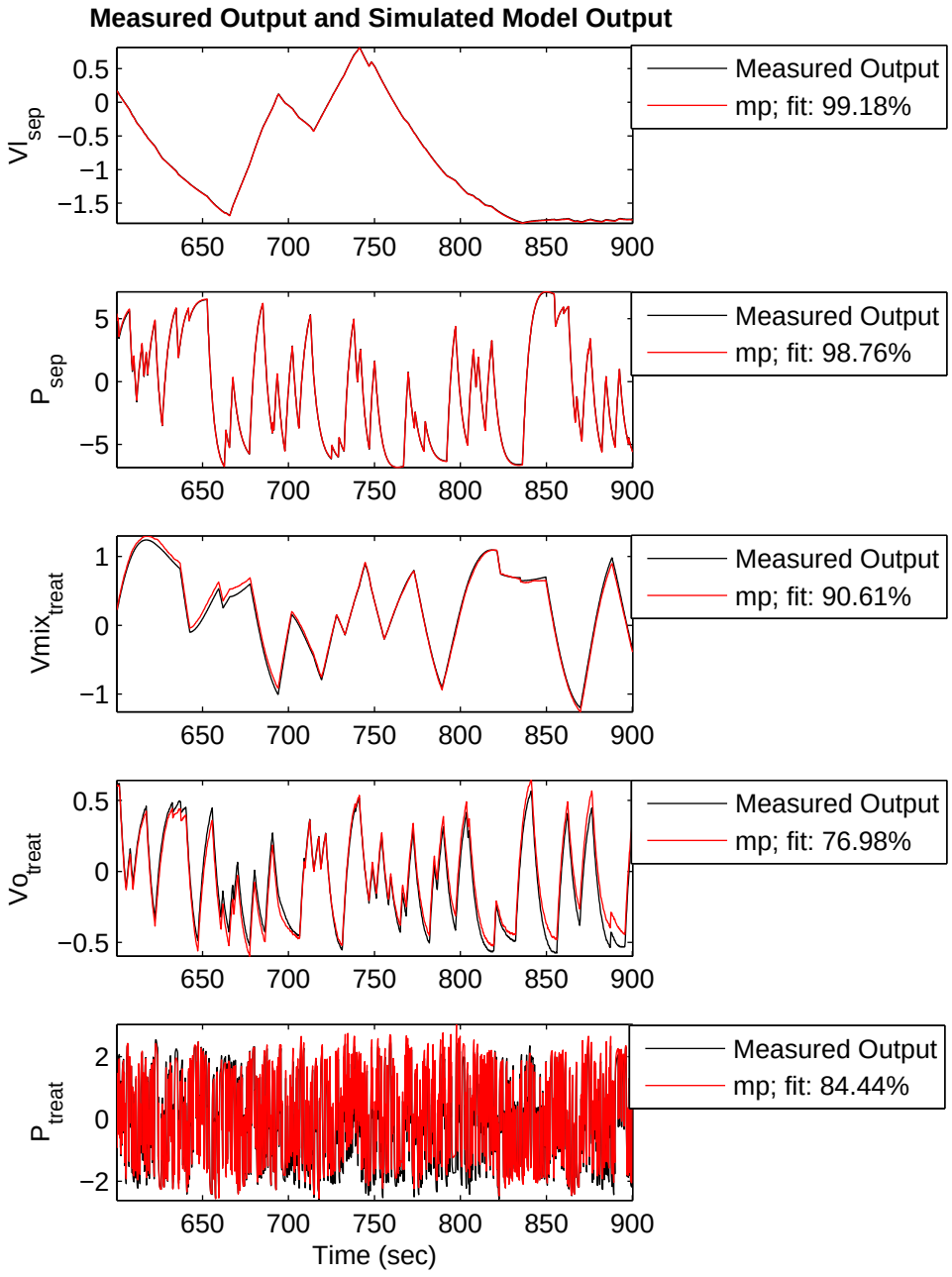

Figure 5.3: System identification results using GBN signals

GBN signals. After studying different options for system identification and based on the overall FDI performance, we decided to use GBN as the best option for excitation signals.

Although the average % of fitting for the identified model using PRBS in Fig. 5.2 is higher than the one using GBN signals in Fig. 5.3, this does not mean that the identified PRBS model is actually better. The higher % of fitting in Fig. 5.2 is due to the fact that the excitation signals do not have sufficient high-frequency content to excite their dynamics (see outputs 2, 4 and 5), which have smaller time constants. As a result, the % of fitting given by the *compare* function in MATLAB® and calculated using equation (5.6) does not give a real estimate of this model's quality, given that the excitation signals are not persistent enough for all the outputs:

$$Fit\,\% \;= 100\left\{1 - \frac{norm(Y-\hat{Y})}{norm(Y-mean(Y))}\right\} \tag{5.6}$$

where Y is the output of the validation data and $\hat{Y}$ is the model output.

Even though this set of PRBS signals does not properly excite the system, we chose to use this model to show some aspects of the GPV technique using a higher order identified model. Given the highly nonlinear nature and complexity of the process, the % of fittings obtained in Fig. 5.3 are considered acceptable. Also, since the excitation signals are properly designed taking into account the time constants for each output variable, the % of fitting actually gives a good estimate of the model's quality.

In order to evaluate the performance of the identified models obtained using PRBS and GBN signals the following fault scenario is simulated: +10% bias fault is applied to the separator vapor outflow valve (A_2) at t=60 sec. The corresponding time histories for the treator and the separator are attached in appendix A, Figs. A.9 and A.10. For both models, an optimal σ was chosen according to the guidelines given

in section 3.6. It is observed in Fig. 5.4 that the model obtained using PRBS signals and depicted in Fig. 5.2 produces S_2 as the minimum angle, giving an erroneous faulty sensor 2 isolation. On the other hand, using the GBN identified model, we see in Fig. 5.5 that the S_2 and A_2 angles are both almost zero, giving a clear A_2 isolation by applying the logic giving in equation (3.19). It should be noticed that for this GBN model, loops 2 and 5 belong to the special case for sensor-actuator FDI as it will be discussed in more detail in section 6.2. This is the reason that the logic in equation (3.19) must be used.

The FDI results illustrated in Fig. 5.4 correspond to one of the few cases where the identified 10^{th} order model obtained using PRBS as excitation signal does not provide good identification results. This scenario was carefully chosen to show a situation where the FDI performance was significantly better using the GBN model. However, the PRBS model also provides satisfactory FDI results in general and could be safely used to implement the proposed FDI method if a better model is not available.

To illustrate a situation where the FDI results given by the identified PRBS model are more representative of its overall performance, the following fault scenario is simulated: -5%/min ramp fault is applied to the treator water volume sensor (S_3) at t=60 seconds, for a setpoint variation of 5% for all the loops. The corresponding time histories for the separator and the treator are attached in appendix A, Figs. A.11 and A.12. It is observed in Fig. 5.6 that the $|GPV|$ increases more than 100% just 0.25 sec. after the fault was applied, providing fast detection, even for faults slowly increasing with time. Also, the S_3 angle moves towards zero rapidly while keeping a minimum separation angle of approximately 16 degrees with the other reference directions, giving a clear isolation. Thus, the FDI performance is excellent using the 10^{th} order model even for a setpoint variation of 5%. These results demonstrate that the state-space representation does not require the actual system order and/or physical states to provide suitable left coprime factors for the

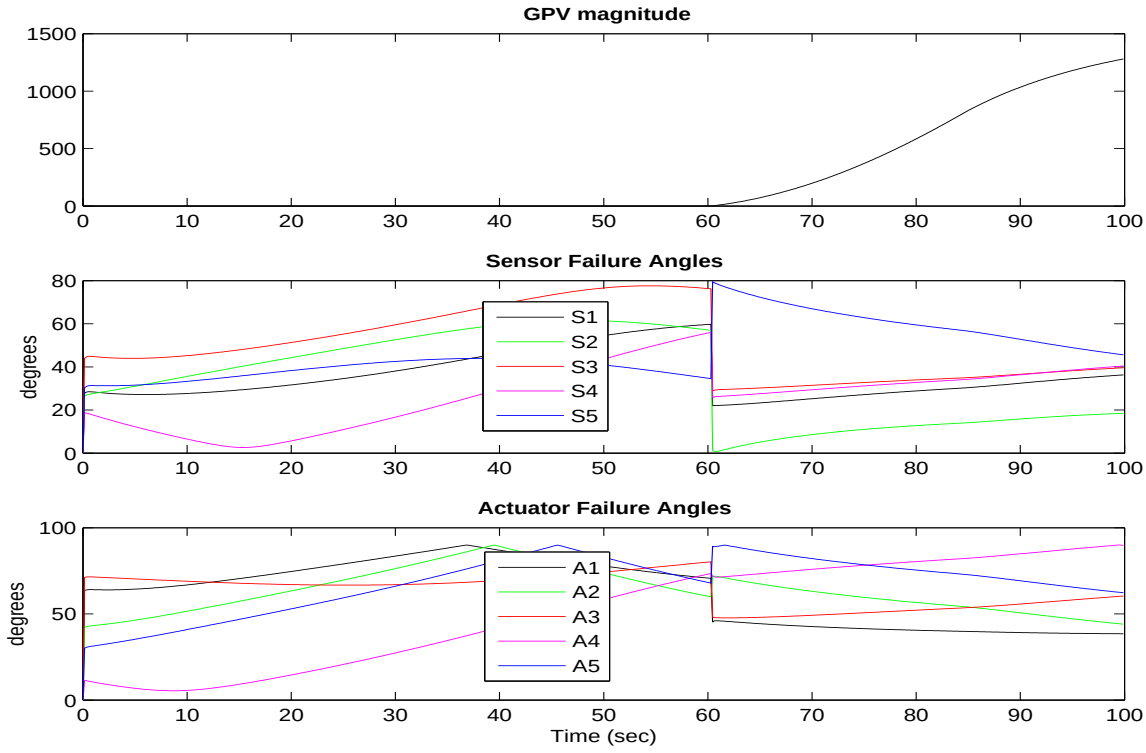

Figure 5.4: Dynamic $|GPV|$ and $\angle GPV$ using the PRBS model

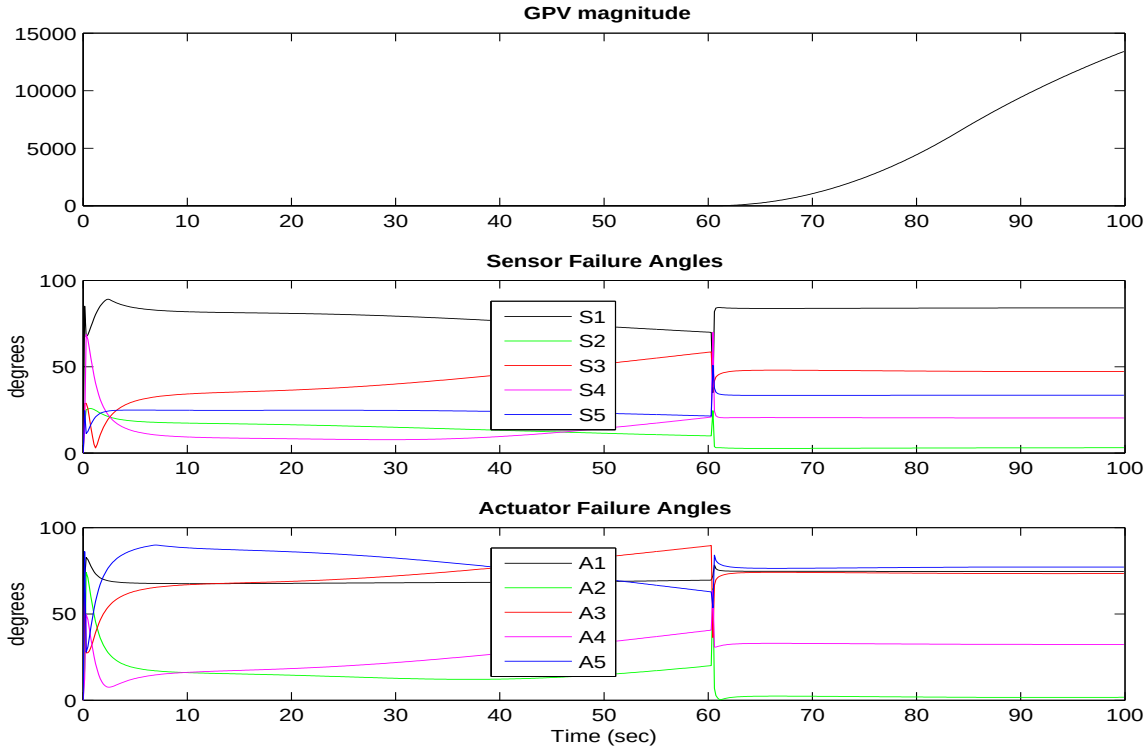

Figure 5.5: Dynamic $|GPV|$ and $\angle GPV$ using the GBN model

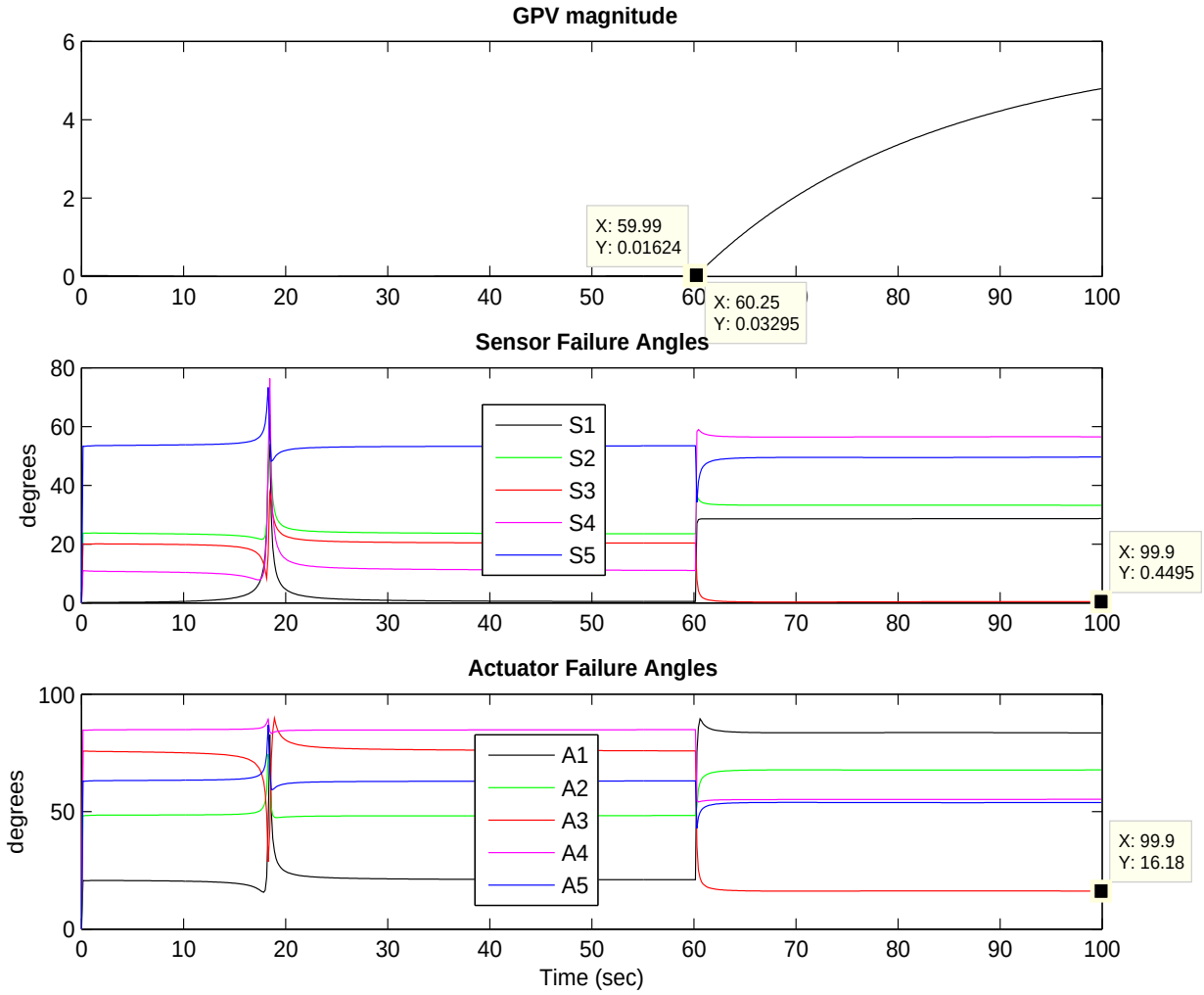

Figure 5.6: Dynamic $|GPV|$ and $\angle GPV$ using the PRBS model

GPV computation. As a result, FDI is possible even for those systems where input-output data is the only accessible information.

The above overview of the SID agent demonstrates the procedure developed in this thesis for taking solid theoretical results, such as the *(pem)* and *(n4sid)* model identification algorithms, and constructing software agents that use these methods in an automated and "self diagnosing" manner to produce useful and reliable results viable for industrial implementation.

Chapter 6

Transformation Matrix Optimization

6.1 Introduction

Fault detection and isolation based on the stable coprime factorization approach assures 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 using the GPV approach is guaranteed. However, FDI techniques based on parity equation implementation of directional residuals are highly dependent on the dynamics of the system. This means that for some cases, the reference directions for some faults might be close or identical, making their isolation difficult or unachievable.

Most previous research in this area had the restriction that the number of faults for which independent directional responses could be determined was limited to the number of plant outputs (number of sensors). As a result, the maximum number of faults that could be isolated with those techniques were limited to m, where m is the number of plant outputs [11], [14].

As an important contribution of this research, the calculation of an optimal transformation matrix T_r is described in this chapter to overcome various directionality problems and the limit in the maximum number of faults that can be isolated. It will be shown in the different simulation results using the separator model described in section 4.3 that the GPV technique is capable of clearly isolating 10 different faults for a system with five outputs. An additional contribution is the implementation of diagnostic logic in the T_r optimization process, to warn the operators if isolation ambiguities may occur.

6.2 Problem definition

The transformation matrix T_r in equation (2.25) plays an important role in FDI using directional residuals. Its significance is based on its ability to change the original set of reference directions to a new one, with better features to improve FDI discrimination and robustness. We state our objectives as follows:

- The calculated T_r should transform the original system in such a way that the GPV magnitude ($|\,GPV|$) after a fault exhibits a significant magnitude change. This requirement assures a clear fault detection.

- T_r should provide enough separation between the reference directions and hyperplanes, guaranteeing unambiguous fault isolation despite nonlinear effects and modelling uncertainty. The ability to separate depends upon the dynamics, the degree of nonlinearity and the presence of noise.

In all the studies performed during this research, it was always possible to compute a suitable T_r. In most cases it is necessary to compute a suitable transformation matrix ($T_r \neq I_{n\times n}$), as is evidenced in the following results obtained for the separator model described in section 4.3.

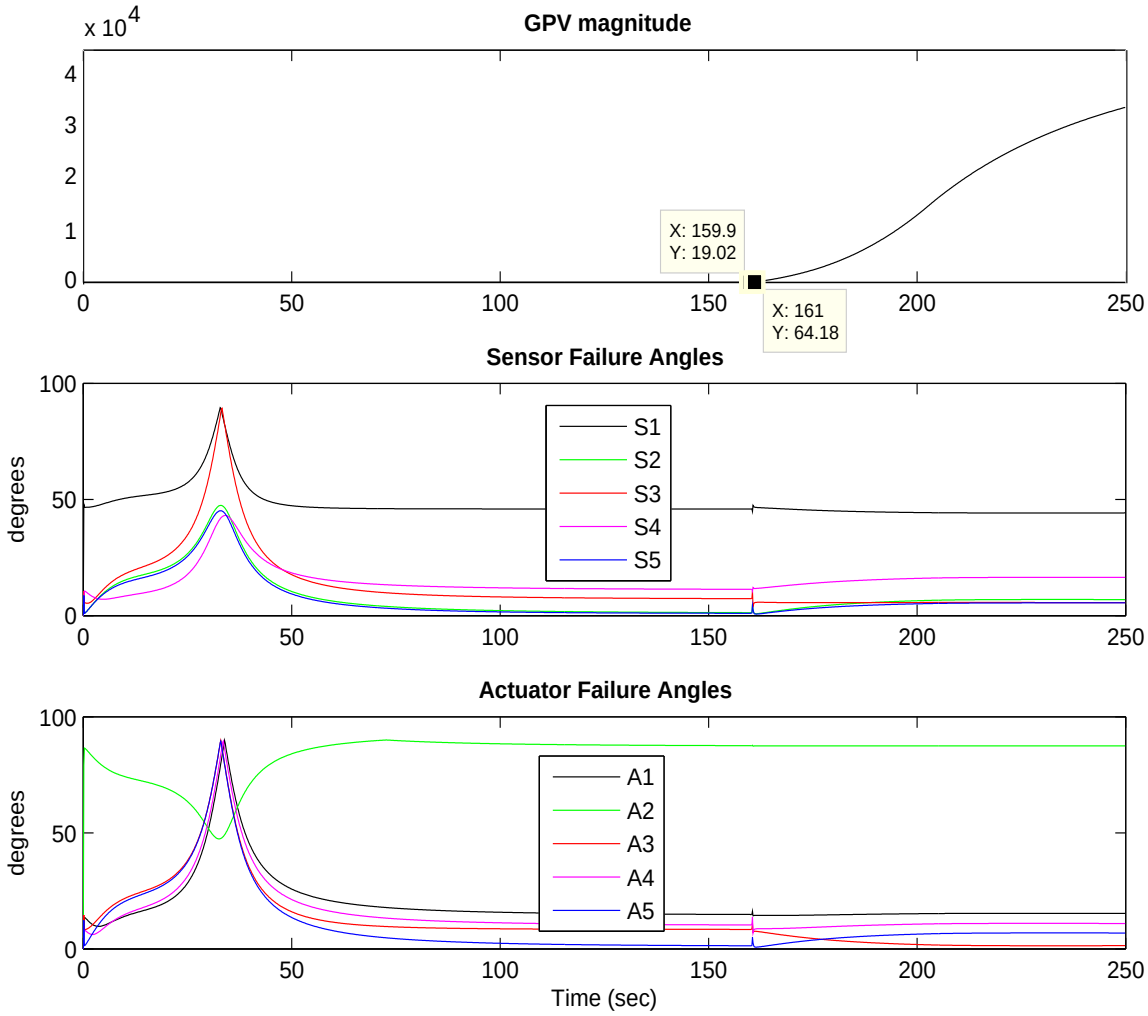

Figure 6.1: Dynamic $|GPV|$ and untransformed $\angle GPV$, A_3 faulty

Figure 6.1 shows the $|GPV|$ and angles using the 5^{th} order identified model illustrated in Fig. 5.3 for a +9% bias fault applied to the treator water outflow valve (A_3) at t=160 sec. The corresponding time histories for the separator and treator are attached in appendix A, Figs. A.1 and A.2. In Fig. 6.1 it is observed that there is a significant increment (3.4 times increase) in the GPV magnitude in one second, giving a fast detection. However, isolation would be ambiguous since the angles between the GPV and A3 and A5 reference directions, and the GPV and S2, S3 and S5 reference hyperplanes, are all close to zero.

The isolation problem is due to the dynamics of the system which do not provide a well separated set of reference hyperplanes and directions. This fact is better demonstrated in tables 6.1, 6.2 and 6.3 which include the angles between the different actuator reference directions (A_i) and sensor reference hyperplanes (S_i) for the separator model in section 4.3.

$\measuredangle$ (°)	A_1	A_2	A_3	A_4	A_5
A_1	-	73.46	15.84	16.10	14.52
A_2	73.46	-	87.41	87.42	87.46
A_3	15.84	87.41	-	11.99	<u>8.20</u>
A_4	16.10	87.42	11.99	-	<u>8.96</u>
A_5	14.52	87.46	<u>8.20</u>	<u>8.96</u>	-

Table 6.1: Actuator/actuator untransformed reference directions angles

$\measuredangle$ (°)	S_1	S_2	S_3	S_4	S_5
S_1	-	89.93	89.95	89.97	89.84
S_2	89.93	-	90	89.96	75.57
S_3	89.95	90	-	89.30	50.69
S_4	89.97	89.96	89.30	-	43.97
S_5	89.84	75.57	50.69	43.97	-

Table 6.2: Sensor/sensor untransformed reference hyperplanes angles

From these tables it is observed that the separation angles between some reference directions are large enough, which is desirable to provide clear isolation. However, the underlined angles might not be large enough, making the isolation difficult because the GPV will be close to or lie on more than one reference hyperplane or direction. This situation can be addressed by transforming the original system using a suitable T_r that improves its FDI capabilities. The transformation matrix computation is presented in the following section.

$\measuredangle$ (°)	A_1	A_2	A_3	A_4	A_5
S_1	38.98	59.54	43.64	52.50	46.65
S_2	3.98	0.25	8.27	9.03	1.16
S_3	16.59	89.20	5.79	3.66	6.03
S_4	10.53	78.77	17.62	11.09	11.29
S_5	14.10	75.42	6.42	5.99	0.10

Table 6.3: Sensor/actuator untransformed reference directions and hyperplanes angles

It should be noticed in table 6.3, that although the separation angle between S_2 and A_2, and that between S_5 and A_5 are not exactly zero, we can treat loop 2 and loop 5 as belonging to the special case for sensor-actuator FDI illustrated previously in Fig. 3.3 and defined by equation (3.15), and thus these faults can be isolated. Based on this observation, these angles will be excluded from the objective function for T_r calculation in section 6.3. Given that these angles are almost zero, it is very difficult to compute a T_r that can separate them without affecting the other angles. If these angles are included in the objective function, the optimization problem becomes ill-conditioned, increasing the computation time and decreasing significantly the other transformed angles.

6.3 Transformation matrix computation

It is desirable to choose T_r to increase the separation angle between the original set of reference directions 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 [14].

This can be formulated as a constrained optimization problem, whose objective is to maximize the angles between the transformed reference directions, to the extent

possible. These angles were calculated using the *subspace* function in MATLAB® which computes the angle between two hyperplanes embedded in a higher dimensional space. The optimization routine maximizes the minimum of $F_{i,j}(T_r)$, where $F_{i,j}(T_r)$ is the objective function containing the angles between the reference directions that are separable [5]. This is implemented by minimizing $-F_{i,j}(T_r)$, using the *fminimax* function in the MATLAB® optimization toolbox, which uses a sequential quadratic programming (SQP) method. According with the description presented in the MATLAB® optimization toolbox user guide [58], this method solves the constrained optimization problem in a similar way as Newton's method for unconstrained optimization. An approximation of the Hessian of the Lagrangian function is made at each major iteration, using a quasi-Newton updating method. This result is used to generate a QP subproblem whose solution is used to form a search direction for a line search procedure [58].

In this context, the term "separable" refers to those directions which do not satisfy equation (3.16). The angle between those actuator reference directions which are lying on one of the sensor reference planes, should be excluded from the $F_{i,j}(T_r)$ function, since it was already demonstrated in equation (3.16) that it is not possible to calculate a T_r to separate them. By eliminating this special case, the optimization performance is improved. The mathematical formulation is given by:

$$F_{i,j}(T_r) = \measuredangle(Z_i, Z_j) \tag{6.1}$$

$$\max_{T_r} \min_{\{F_{i,j}\}} \{F_{i,j}(T_r)\} \tag{6.2}$$

$$\textit{such that } c(T_r) \leq 0$$

where $c(T_r) \leq 0$ represents nonlinear inequality constraints; and Z_i and Z_j are transformed reference directions. These directions are given by transforming B_n^j, B_d^i and I^i, where by convention I^i is the i^{th} column of $I_{n\times n}$. The nonlinear inequality constraints can be used to set a desired minimum separation angle between the

transformed reference directions as presented in equations (6.4) and (6.6).

The T_r calculation approach proposed using optimization is highly flexible because it allows extending the objective function or adding different nonlinear constraints to take into account the dynamics of the system and solve some geometrical restrictions, such as handling the special case for sensor-actuator FDI discussed in section 3.4 and dealing with the hyperplane intersection situation presented in section 6.3.1. In order to improve the optimization results towards the FDI application proposed in this research, the objective function is extended or nonlinear constraints of the form $c_{eq}(T_r)$ are implemented as presented in section 6.3.1.

As the number of inputs and outputs in the system increases, the optimization problem becomes more challenging, since there are more reference directions and reference hyperplanes to separate. For the separator model described in section 4.3, the system has five inputs and five outputs, which produces 45 different combinations of reference directions, reference hyperplanes and combination of both to be separated in the objective function (consider five sensor reference directions, each to be distinguished from 4 others and from 5 actuator reference directions, yielding 5×9 combinations). Despite the large number of separation angles to maximize, the original objective function in equation (6.1) was able to provide a transformation matrix that increases the angles in the original set of reference directions and reference hyperplanes well enough to provide clear isolation.

To illustrate the T_r effect, FDI is performed for the same fault scenario used to generate Fig. 6.1, but this time using the T_r obtained by optimization to transform the original set of reference directions. It is observed in Fig. 6.2 that the transformed GPV rapidly moves towards the A_3 reference direction, while keeping a minimum separation angle of 21.2 degrees with the other reference directions. Conversely, if the untransformed set of angles in Fig. 6.1 is used for isolation, it is not possible

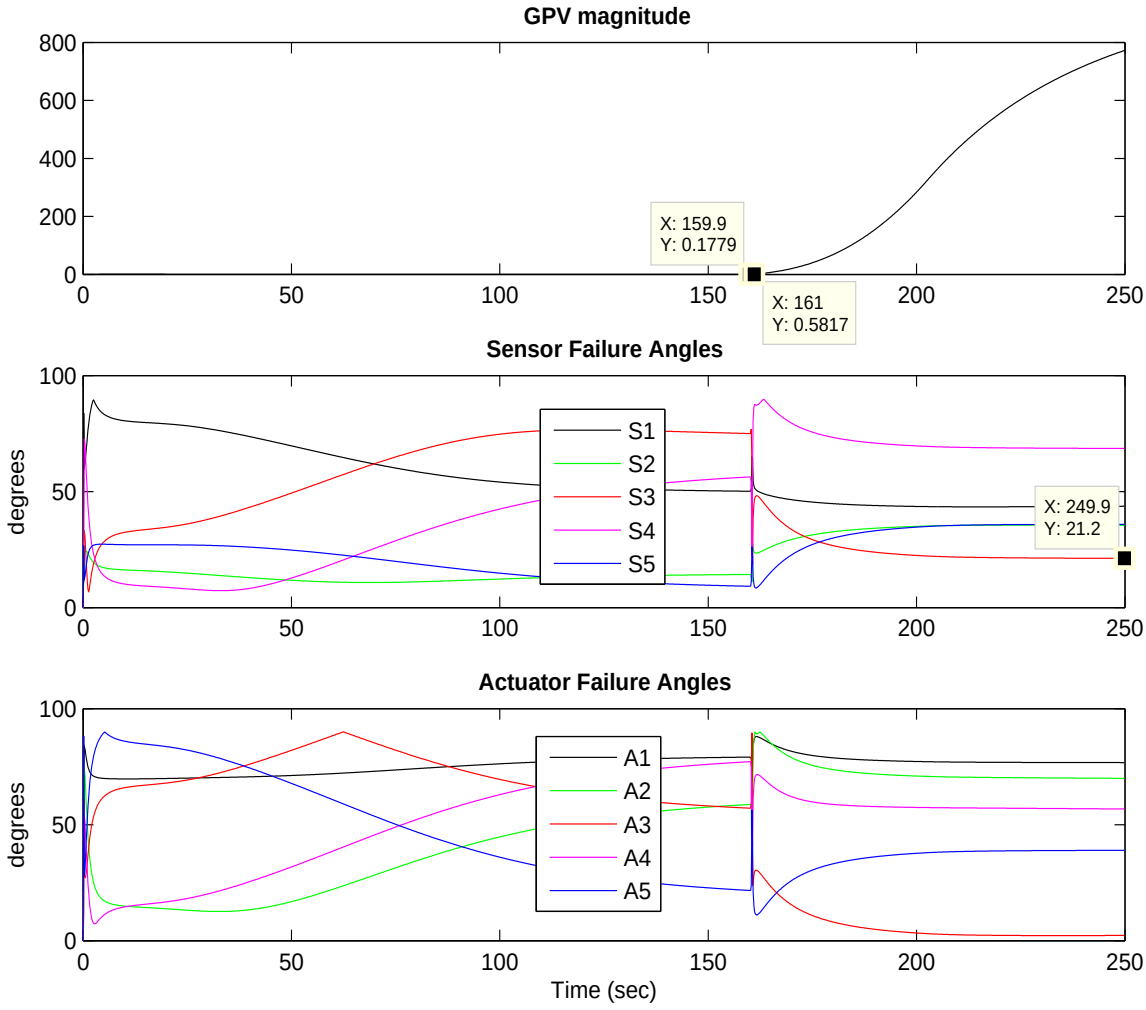

Figure 6.2: Dynamic $|GPV|$ and transformed $\angle GPV$, A_3 faulty

to distinguish which one is the faulty element. This shows the significance of the transformation matrix for FDI using directional residuals.

The transformed set of reference directions and hyperplanes is given in tables 6.4, 6.5 and 6.6. Although the transformed set of reference directions and hyperplanes has some smaller angles, the objective of maximizing the smallest angles in the original set was achieved. The transformed set provides better separation angles overall, with a minimum angle of 19.79 degrees between A_3 and S_3, which is considered large enough to provide clear isolation based on a 10 degrees threshold assumption.

$\measuredangle\,(^\circ)$	A_1	A_2	A_3	A_4	A_5
A_1	-	74.73	76.33	68.29	87.10
A_2	74.73	-	69.56	28.22	78.99
A_3	76.33	69.56	-	57.23	40.61
A_4	68.29	28.22	57.23	-	81.22
A_5	87.10	78.99	40.61	81.22	-

Table 6.4: Actuator/actuator transformed reference directions angles

$\measuredangle\,(^\circ)$	S_1	S_2	S_3	S_4	S_5
S_1	-	80.99	82.26	71.94	81.79
S_2	80.99	-	85.39	69.68	36.61
S_3	82.26	85.39	-	51.22	65.99
S_4	71.94	69.68	51.22	-	80.34
S_5	81.79	36.61	65.99	80.34	-

Table 6.5: Sensor/sensor transformed reference hyperplanes angles

$\measuredangle\,(^\circ)$	A_1	A_2	A_3	A_4	A_5
S_1	19.79	83.60	42.43	83.69	51.95
S_2	66.19	NT	36.86	25.61	19.79
S_3	69.75	42.63	19.79	19.79	59.24
S_4	28.23	19.82	67.41	20.19	76.97
S_5	79.44	35.55	37.97	23.50	NT

Table 6.6: Sensor/actuator transformed reference directions and hyperplanes angles

NOTE: The abbreviation NT in table 6.6 refers to the angles that were not optimized, due to their belonging to the special case for sensor-actuator FDI discussed previously in section 6.2.

6.3.1 Objective function extensions

The reference hyperplanes are generally not parallel, so they intersect with each other in a line. This condition may adversely affect the FDI performance if a sensor failure produces a faulty steady-state GPV that lies on or close to the hyperplane intersection line, in which case it appears to lie nearly on both hyperplanes. For simplicity, although the separator FDI is performed in a 5-dimensional space, the hyperplane intersection situation illustrated in Fig. 6.3 is plotted in a 3-dimensional space.

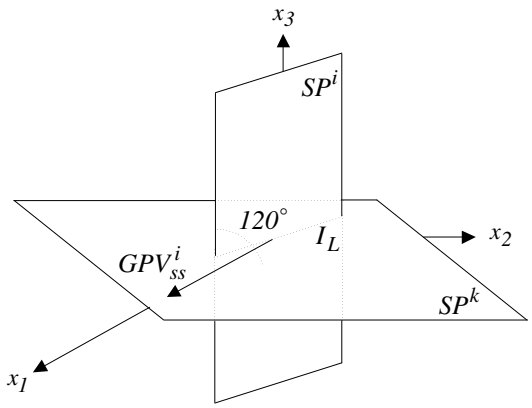

Figure 6.3: Planes intersection

Fig. 6.3 shows that if the GPV^i_{ss} lies on or close to the hyperplane intersection line (I_L), its angle with respect to the i^{th} and k^{th} reference hyperplanes are both zero or close to zero, giving an ambiguous isolation. Nevertheless, this ambiguity can be avoided by adding equation (6.3) to extend the objective function proposed in (6.1). This assures that the separation angle between the faulty steady-state GPV for the i^{th} sensor and the k^{th} sensor reference hyperplane is maximized to provide an unambiguous isolation. This can be expressed mathematically as:

$$F_{i,j}(T_r) = \measuredangle(T_r\, GPV^i_{ss}, Z_k) \tag{6.3}$$

However, if a specific minimum separation angle between $\measuredangle(T_r\, GPV^i_{ss}, Z_k)$ is required, equation (6.3) can be reformulated in a nonlinear constraint as follows:

$$\measuredangle(T_r\, GPV_{ss}^{i}\,, Z_k) \geq \Theta_{min} \tag{6.4}$$

where Θ_{min} is the desired minimum separation angle. If the objective function is not extended as proposed in equation (6.3) or the nonlinear constraint in equation (6.4) is omitted during solution of the optimization problem, then there is no guarantee that the resulting T_r will not produce a GPV_{ss}^{i} aligned or close to the i^{th}-k^{th} hyperplanes intersection line [36]. This modification is required only for those cases where optimization using the original objective function in equation (6.1) returns a transformation matrix that pushes a faulty steady-state GPV close to a hyperplane intersection line.

To illustrate the hyperplane intersection issue and the effect of equation (6.3) during the T_r calculation, the FDI algorithm is tested using the 10^{th} order identified state space model illustrated in Fig. 5.2 for the separation process described in section 4.3. The scenario is described as follows: At t=300 sec. a 16.43%/min ramp fault representing an inaccurate sensor reading increasing with time is applied to the separator liquid volume sensor (S_1) at the nominal operating point. The FDI results illustrated in Fig. 6.4 were obtained using a T_r calculated by extending the objective function as proposed in equation (6.3).

It is observed that even for faults slowly increasing with time, the GPV magnitude increases around 15 times, only 0.9 sec. after the fault is applied, providing immediate detection even though the deviation at that time is only 0.25%. Also, the GPV angle corresponding to sensor 1 (S_1) moves toward zero very rapidly, reaching a minimum separation of 11.29 degrees with respect to the other reference directions within the first 20 sec. These detection and isolation times are acceptable, since the fault was applied to the liquid phase which has slower dynamics than the gas phase. The time histories for the separator and treator are shown in appendix A, Figs. A.5 and A.6. However, if the modification to the objective function given by equation

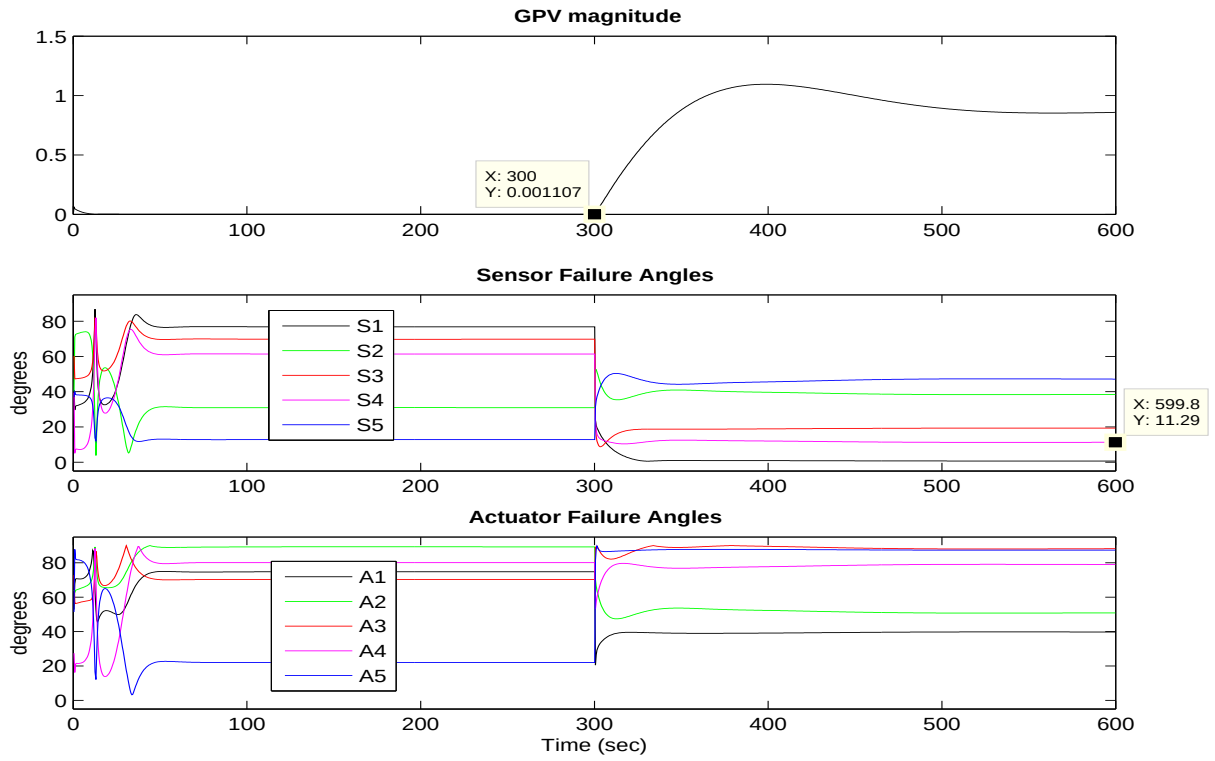

Figure 6.4: Static $|GPV|$ and $\angle GPV$, S_1 faulty - Modified

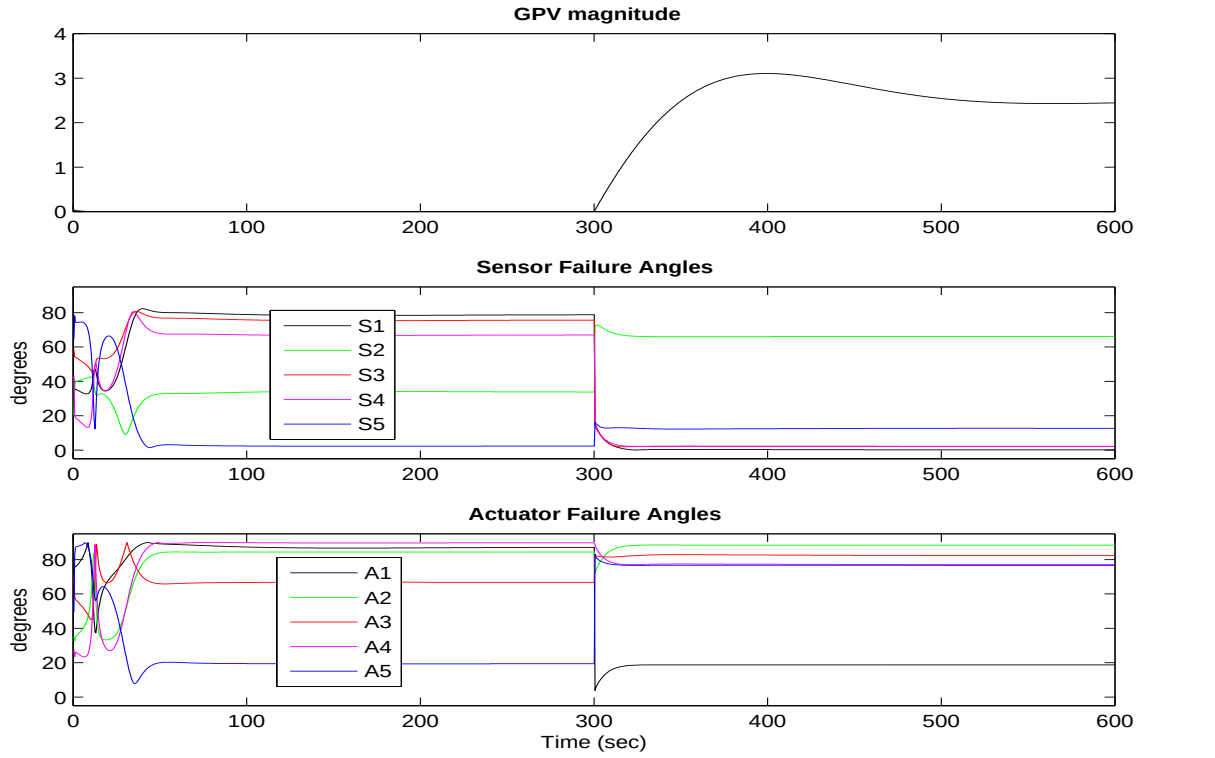

Figure 6.5: Static $|GPV|$ and $\angle GPV$, S_1 faulty - Unmodified

(6.3) is omitted for this case, the GPV_{ss} lies very close to the intersection line for hyperplanes 1, 3 and 4, giving an isolation angle close to zero for all of them. This is illustrated in Fig. 6.5, where the angles corresponding to S_1, S_3 and S_4 are all close to zero, making the isolation ambiguous.

Similarly, if the system dynamics lead to the special case defined by equation (3.15) the objective function must also be extended. The actuator FDI issue is overcome by implementing the logic presented in equation (3.19), which is only effective if the j^{th} actuator direction vector which lies on the plane SP^i is outside the cone defined by vectors B_s^i and I^i. Figures 6.6 and 6.7 illustrate how these sensor fault sectors are defined for the JCSTR model described in appendix B and their relations with B_n^j. In Fig. 6.6 it is observed that the outflow valve (A_1) reference direction (B_n^1) lies on the boundary of the volume sensor (S_1) fault sector, making isolation difficult. Ideally, isolation based on the steady-state GPV activity along B_s^1 should still be feasible. However, it has been observed that this does not work in the nonlinear case.

Conversely, in Fig. 6.7 the heating inflow valve (A_2) reference direction (B_n^2) is well outside of the temperature sensor fault sector, which is desirable for the sensor fault isolation. In the situation depicted in Fig. 6.6 we can maximize the angle between B_n^1 and B_s^1 if the steady state GPV strategy is used.

To overcome the FDI ambiguity arising from the sensor-actuator special case defined in section 3.4 when B_n^j is inside the cone sector, the objective function needs to be extended. This assures a minimum separation angle between the steady-state GPV for the i^{th} sensor and the j^{th} actuator reference direction to be large enough to provide an unambiguous isolation. This can be expressed mathematically as:

$$F_{i,\,j}\left(T_r\right) = \measuredangle(T_r\, GPV_{ss}^{i}\,, Z_j) \tag{6.5}$$

However, if a specific minimum separation angle between $\measuredangle(T_r\, GPV_{ss}^{i}\,, Z_j)$ is re-

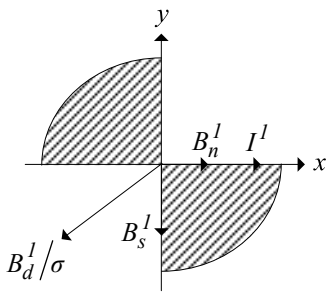

Figure 6.6: Volume sensor fault sector ($T_r = I$)

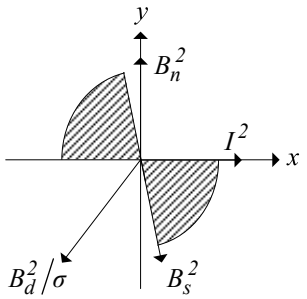

Figure 6.7: Temperature sensor fault sector ($T_r = I$)

quired, equation (6.5) can be reformulated in a nonlinear constraint as follows:

$$\measuredangle(T_r\, GPV_{ss}^{i}\,, Z_j) \geq \Theta_{min} \tag{6.6}$$

where Θ_{min} is the desired minimum separation angle. If the objective function is not extended as proposed in equation (6.5) or the nonlinear constraint in equation (6.6) is omitted during the optimization routine, there is no guarantee that the resulting T_r will provide enough separation to distinguish the i^{th} sensor fault from the j^{th} actuator fault, for systems satisfying equation (3.15).

The effect of equation (6.5) is reflected in Figs. 6.8 and 6.9 which correspond to the $|GPV|$ and angles for a -10% bias fault applied to the separator vapor pressure sensor (S_2) at t=60 sec. These results were obtain using the 5^{th} order identified model illustrated in Fig. 5.3 for the separation process described in section 4.3. We chose to apply a fault in this sensor since S_2 and A_2 were considered belonging to the special case for sensor-actuator FDI in section 6.2. The corresponding time

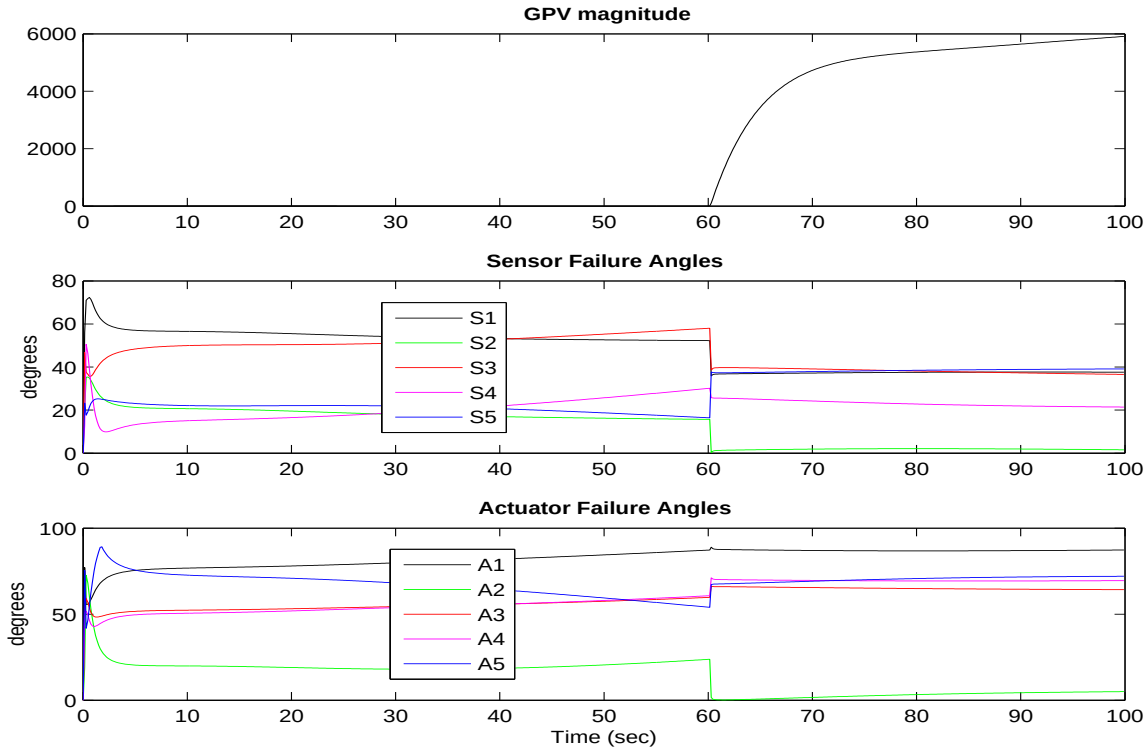

Figure 6.8: Dynamic $|GPV|$ and $\angle GPV$, S_2 faulty - Unmodified

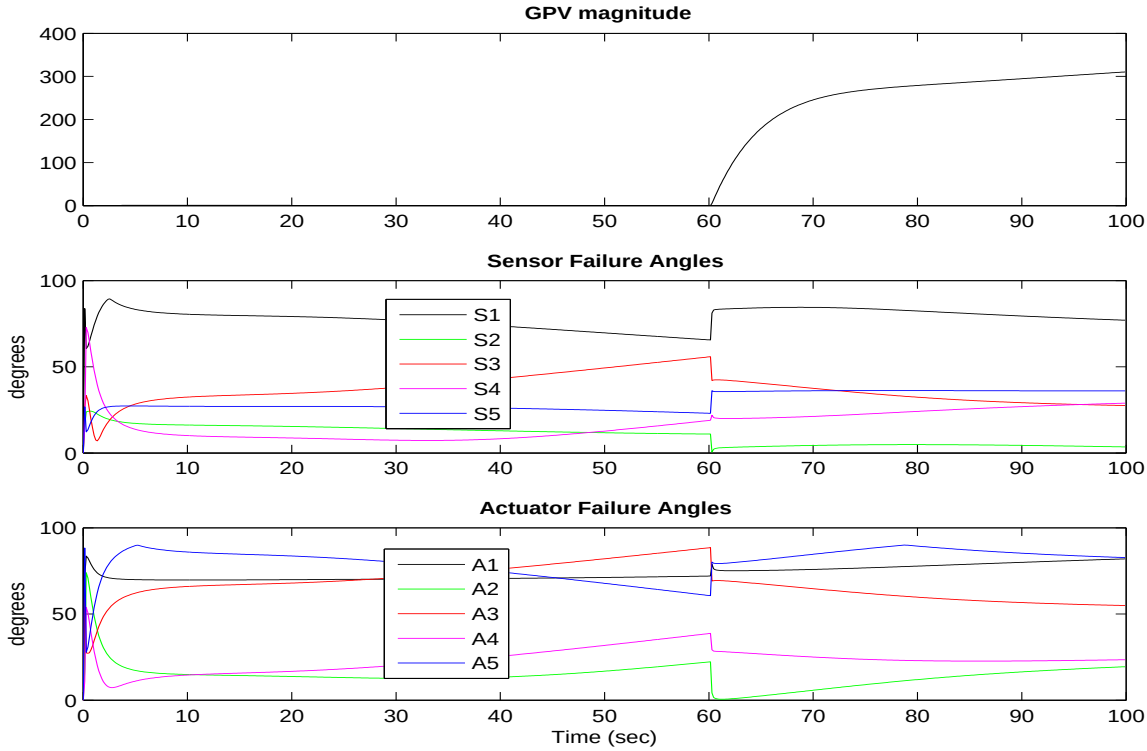

Figure 6.9: Dynamic $|GPV|$ and $\angle GPV$, S_2 faulty - Modified

histories are shown in appendix A, Figs. A.7 and A.8. Figure 6.8 shows that if T_r is computed with the original objective function proposed in equation (6.1), the GPV for a S_2 fault lies not only in sensor 2 reference hyperplane, but almost in the A_2 reference direction during the next 40 sec. after the fault was applied. This would result in a wrong A_2 isolation instead of S_2, according to the logic in equation (3.19). However, if the objective function is extended as presented in equation (6.5), the A_2 GPV angle starts moving away from zero right after the fault is applied, rapidly reaching the required angle threshold in equation (3.19) to clearly declare a fault in sensor 2.

6.4 On-line implementation

Figure 6.10 shows the detailed transformation matrix evaluation logic block diagram corresponding to the "Transformation matrix calculation block" shown in Fig. 4.2. The logic in Fig. 6.10 allows the on-line T_r computation for any new identified state space model. It also implements automatically the required objective function modifications proposed in section 6.3.1 to solve the hyperplanes intersection issue and the ambiguity arising from the sensor-actuator special case defined in section 3.4 when B_n^j is inside the cone sector.

The transformation matrix calculation starts whenever a new state space model is available and its corresponding actuator reference directions and sensor reference hyperplanes are computed. If any sensor reference hyperplane and the corresponding actuator reference direction satisfies equation (3.15), they are excluded from the objective function since it was already shown in equation (3.16) that they are not separable, so there is no point trying to improve the situation. The other reference directions and hyperplanes are included in the objective function. Once the objective function is assembled, the optimization algorithm is executed until the computed T_r

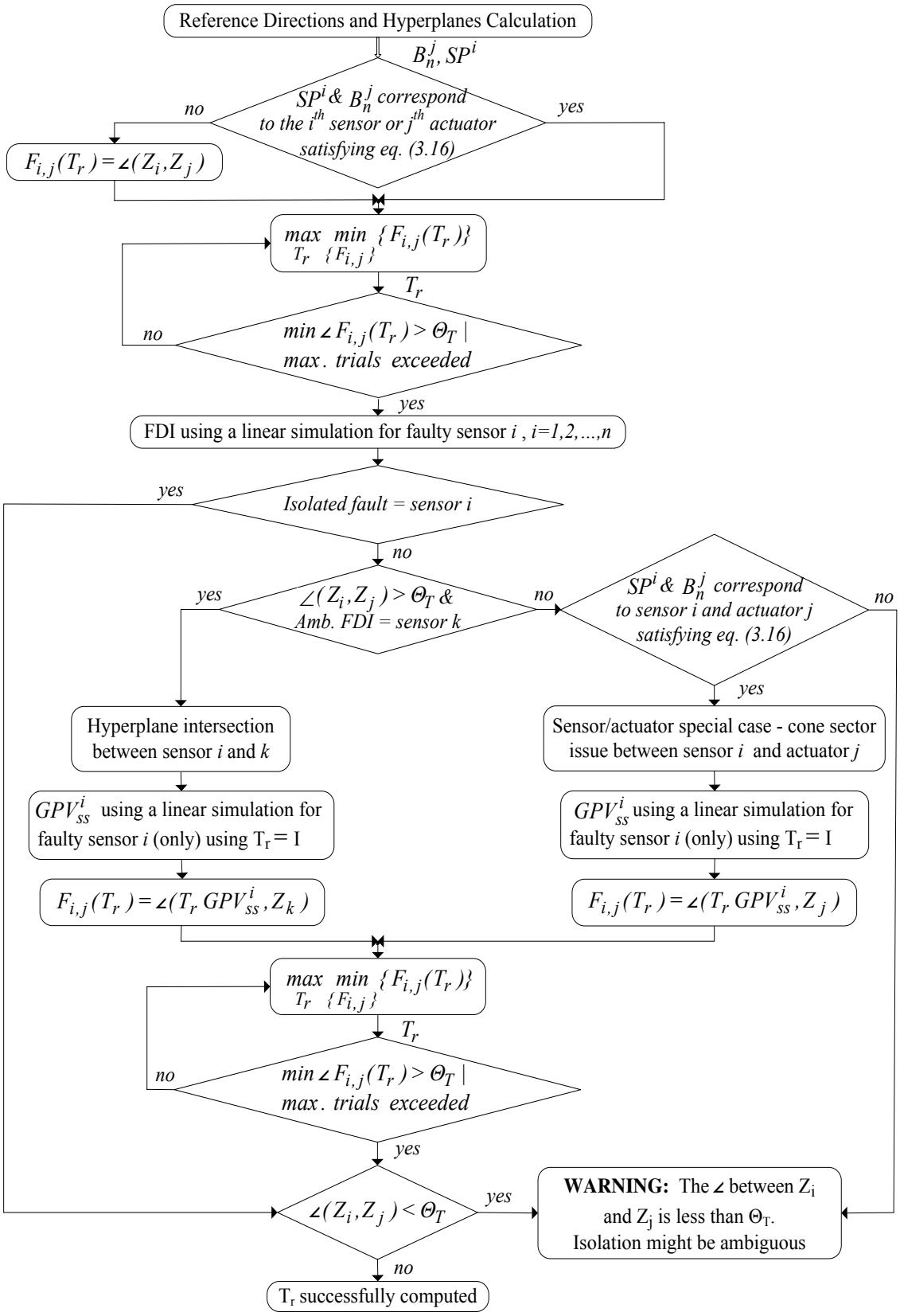

Figure 6.10: Transformation matrix block diagram

provides a transformed set of reference directions whose minimum separation angle is larger than a prespecified angle threshold Θ_T or the maximum number of trials allowed is exceeded. Θ_T defines the limit between clear and ambiguous isolation during the decision-maker process, as it will be explained in more detail in chapter 7.

Once the T_r is calculated the first time, the algorithm proceeds to self-diagnose its performance before it is used for FDIA in the actual plant. To evaluate the T_r performance, FDI tests are executed using a linear simulator of the plant built with the identified model. A medium size fault is applied to each sensor, S_i, to determine if it is required to extend the objective function and recalculate T_r. The medium size fault assumption is valid to generalize the FDI performance for an arbitrary sensor fault, given that the GPV direction for different fault sizes is almost the same and there is only a magnitude increase when the fault size F_{size} increases. If the isolation result for each case corresponds to the faulty sensor i in the linear simulation, it means that the computed T_r does not need to be recalculated.

On the other hand, if the isolation results obtained from the linear simulation do not correspond to the sensor fault that was applied, the algorithm proceeds to identify the causes of this wrong diagnosis. If the minimum separation angle between the transformed reference hyperplane for the faulty sensor i using this T_r is greater than Θ_T and the isolation is ambiguous with respect to the k^{th} sensor, it means that we have the hyperplane intersection issue between sensor i and k. Then, the algorithm proceeds to estimate the steady state GPV for a fault in sensor i (GPV_{ss}^{i}) using the linear simulator with T_r equal the identity matrix. Using GPV_{ss}^{i}, equation (6.3) can be implemented to extend the objective function to maximize its separation angle with respect to the k^{th} reference hyperplane which was the one intersecting hyperplane i using the T_r computed with the original objective function.

Conversely, if the minimum separation angle between the transformed reference hyperplane for the faulty sensor i using the T_r computed with the original objective function is not greater than Θ_T or the isolation is not ambiguous with respect to a sensor, the logic proceeds to check for the sensor/actuator special case. If the isolated fault was an actuator fault and it satisfies equation (3.15) with respect to the faulty sensor i, this means that we have the FDI ambiguity arising from the sensor-actuator special case defined in section 3.4 when B_n^j is inside the cone sector. Then, the algorithm proceeds to estimate the steady state GPV for a fault in sensor i (GPV_{ss}^i) using the linear simulator with T_r equal the identity matrix. Using GPV_{ss}^i, equation (6.5) can be implemented to extend the objective function to maximize its separation angle with respect to the j^{th} actuator reference direction that lies on the i^{th} sensor reference hyperplane.

Once the objective function is extended for the different cases as required, the T_r is recalculated until the minimum separation angle given by Θ_T is achieved or the maximum number of trials is exceeded. If after the final T_r calculation the minimum separation angle of the transformed set of reference directions is greater than Θ_T, the T_r computation is considered successfully. Otherwise, the following warning is displayed:

"The $\angle$ between Z_i and Z_j is less than Θ_T. Isolation might be ambiguous."

This means that the isolation between element i and j might be ambiguous since the separation angle between their reference directions is not greater than the angle threshold. It should be noticed that the recalculation process is intended to identify and solve the special cases described by equations (6.3) and (6.5). Thus, the previous warning might be still generated since the recursive T_r calculation cannot guarantee that the minimum separation angle will be greater than Θ_T. However, it is expected that if the maximum number of a trials is selected properly, the optimization results will give a T_r satisfying the angle requirements. For the 5^{th} order identified model

in Fig. 5.5, T_r was successfully computed with only one trial giving a minimum separation angle of 19.79 degrees as shown previously in table 6.6. Nevertheless, depending on the complexity of the system dynamics and the number of reference directions to separate, the maximum number of trials required may be higher.

Chapter 7

Decision Maker

7.1 Introduction

The decision-maker block plays a crucial role in FDI using the GPV technique, since it translates the detection, isolation and classification diagnosis made in terms of the GPV magnitude, angles and slopes, to a simple operator display that can be easily understood without knowing the mathematical background behind it. It also has the capability to decide whether the system is facing the special case for sensor-actuator faults presented in section 3.4 or an ambiguous isolation situation between two different elements. Robustness is also increased by implementing the logic in equations (7.1), (7.2) and (7.3), which ensures that there is enough information about the faulty element, before a final isolation statement is made to start fault accommodation. The decision-maker also exploits the advantages of the dynamic and static GPV implementations in its logic to provide fault classification, faster detection and clearer isolation. The new contribution described here is the demonstration that a complicated decision process can be automated, based on human logic, although an FDI expert has to tune the logic.

In most real world processes the operating point is continuously changing, and

this becomes an issue for model-based FDI techniques applied to nonlinear plants [59]. To overcome this situation, an on-line threshold computation block has been implemented as part of the decision-making strategy to take into account modeling errors [60]. The proposed decision-maker logic is presented in this chapter based on the block diagram in Fig. 7.3. First, however, the concept of $|GPV|$ signature is introduced in section 7.2.

7.2 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 using the static GPV for fault-size estimation and classification, for further use in sensor accommodation. The $|GPV|$ signature is defined in terms of its slope after the fault occurs ($|GPV|_{slope}$) and the peak change of GPV magnitude, ($\Delta|GPV|_{peak}$) [6]. This signature is best exhibited by the static GPV implementation, as it will be shown in the following simulation results.

In order to illustrate the GPV signature concept, the following fault scenarios are simulated: a -3% bias fault is applied to the separator liquid volume sensor (S_1) and a -10%/min ramp fault is applied to the treator oil volume sensor (S_4), both at t=60 sec. The separator and treator time histories for each fault are shown in appendix A, Figs. A.13 to A.16 respectively. It is observed in Fig. 7.1 that for the bias case, the static GPV magnitude exhibits an abrupt change right after the fault is applied, while for the ramp fault case $|GPV|$ initially shows a ramping behaviour, as illustrated in Fig. 7.2. As a result, a fault can be clearly classified as bias or ramp type, using the static GPV magnitude behaviour at the time that the fault is detected. In contrast, for the dynamic GPV implementation, the GPV magnitude change does not exhibit a clearly different behaviour for the bias and ramp fault

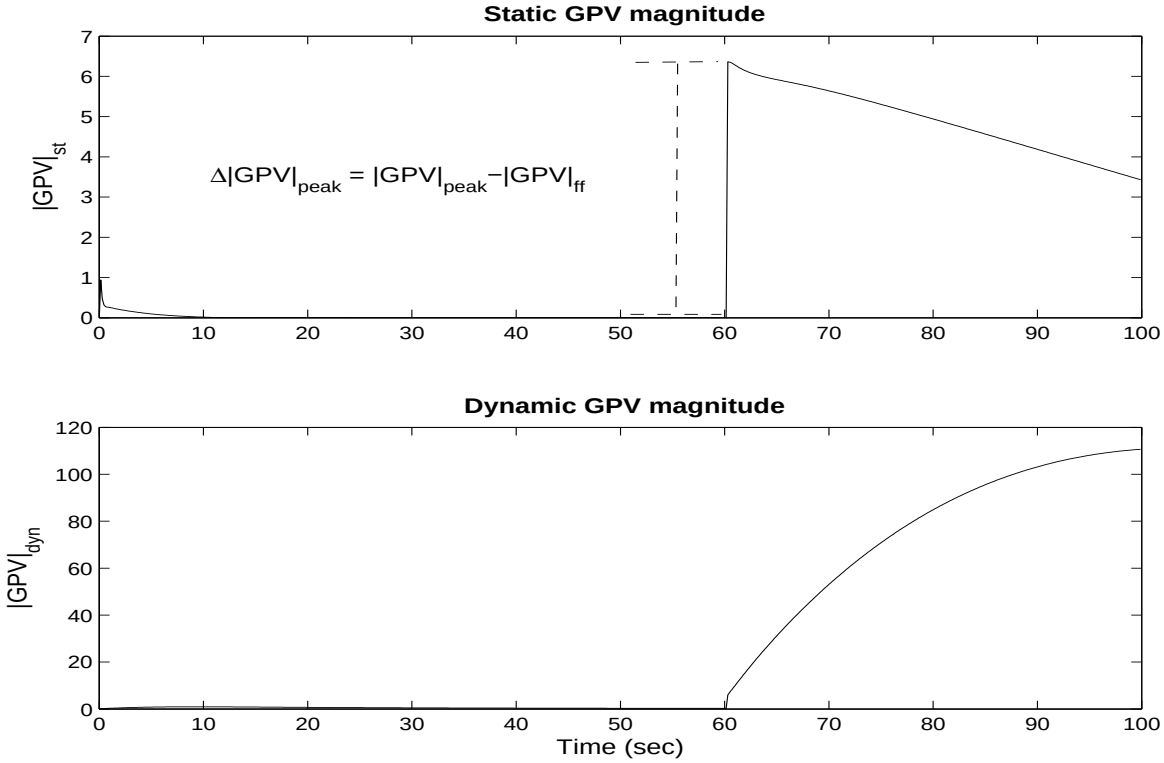

Figure 7.1: Static and Dynamic $|GPV|$ for -3% bias S_1 applied at t=60 sec.

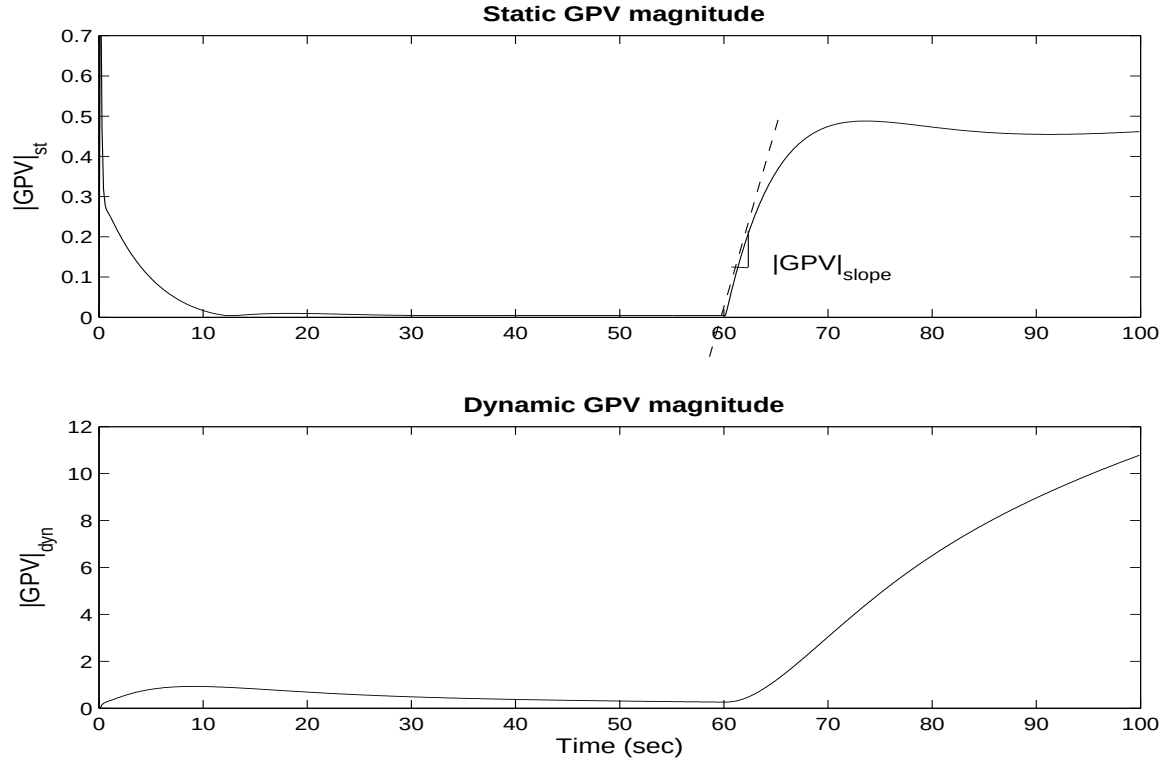

Figure 7.2: Static and Dynamic $|GPV|$ for -10%/min ramp S_4 applied at t=60 sec.

cases. It should be noticed that for the dynamic implementation a suitable σ was chosen, following the guidelines in section 3.6.

From Figs. 7.1 and 7.2 it is also observed that the $|GPV|$ change after the fault is applied is larger using the static GPV implementation, which is highly desirable to provide fast detection. Also, Fig. 7.2 shows that the static GPV magnitude ($|GPV|_{st}$) reaches steady state faster than the dynamic one during normal operation. This is advantageous to reduce the calculation time for the on-line threshold computation block presented in section 7.3.1. This qualitative behaviour was found to be a pattern after extensive simulations were performed using different sizes of bias and ramp faults, for all ten faults. Based on these results, we have chosen the static GPV as the best option to implement the on-line threshold computation, fault detection, fault classification, fitting calculation and fault size estimation blocks.

7.3 Decision-maker logic

The decision-maker logic is composed of four modules: the on-line threshold computation, fault detection, fault classification and fault isolation blocks as shown in Fig. 7.3. These blocks are discussed in detail in the following subsections.

7.3.1 On-line threshold computation block

Once the residuals are generated, the on-line threshold computation block is executed to estimate the magnitude threshold (M_T) at the corresponding operating point. The on-line threshold computation is executed only once, after the identified model has changed, but before the fault free GPV magnitude ($|GPV|_{ff}$) is known. It is assumed that the system is in normal operation for a period of time that is long enough for the static GPV magnitude $|GPV|_{st}$ to reach steady state ($|GPV|_{st} = SS$). However, since the static GPV follows the dynamics of the system, it will stabilize as

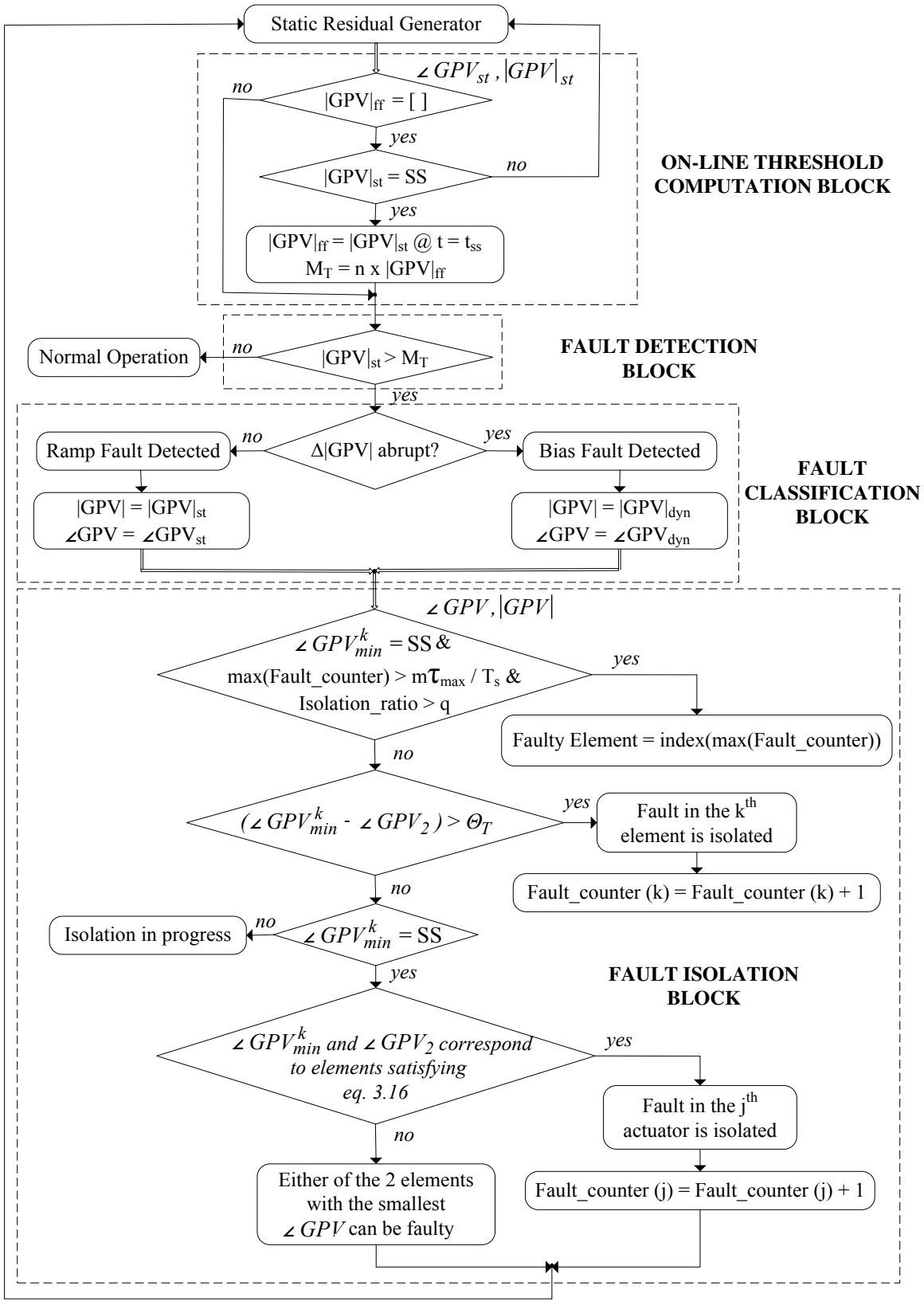

Figure 7.3: Decision-maker block diagram

soon as the system reaches steady state. When the $|GPV|_{st}$ reaches steady state, the $|GPV|_{ff}$ is defined as the static GPV magnitude value at the steady state time (t_{ss}). Thus, the threshold magnitude can then be computed on-line as $M_T = n \times |GPV|_{ff}$, where $n > 1$ regulates the tradeoff between detection speed and the occurrence of false alarms. If n is too small, it may generate false alarms due to noise and modeling errors. Conversely, if n is very large, the detection may be very slow or impossible for small bias or ramp faults. Ideally, the identified model and the on-line estimated $|GPV|_{ff}$, should be still valid for small setpoint variations. However, there will be an increase in the fault free $|GPV|$ during transients, due to modeling errors. In order to avoid false fault alarms, but still be able to identify possible faults happening through these transients, the value of n is increased during such periods. Thus, the magnitude threshold, $M_T = n \times |GPV|_{ff}$ has a different value depending on the state of the process. The corresponding values of n are empirically chosen to provide a good tradeoff between false alarms and misdetection rates. The complete FDIA performance during the transient is evaluated in section 8.5.3 using the separator model. This aspect is also clarified in appendix C using a simple linear model.

In our previous research, see [37], four different techniques were proposed to implement an adaptive threshold: Empirically scheduled, piece-wise linear, quadratic and cubic threshold logics. Although these techniques perform very well overall, they require a lot of previous testing to obtained the necessary data to implement them. Conversely, the on-line threshold computation logic proposed in this section, gives a more practical and accurate way to compute a suitable magnitude threshold.

7.3.2 Fault detection and classification blocks

The detection block is executed following the magnitude threshold computation. If the static GPV magnitude is greater than the magnitude threshold, a fault is detected at $t{=}t_{fault}$ and the classification and isolation blocks are then executed.

Otherwise, normal operation is declared. The fault classification block exploits the GPV magnitude signature concept presented in section 7.2 to identify the fault type (F_{type}). If the static GPV magnitude changes abruptly at t=t_{fault} then the fault is classified as bias, otherwise, a ramp fault is declared.

If a bias fault is detected, the isolation is performed based on the dynamic GPV angles. Conversely, for the ramp case, the isolation is based on the static GPV angles. Previous research using the GPV technique, see [13], only used the dynamic GPV implementation for fault detection and isolation. However, given the faster behaviour of the static GPV, its use is not only convenient for detection, classification and fault size estimation, but also for isolation of slowly varying (ramp) faults. This is illustrated in Figs. 7.4 and 7.5 for +1%/min ramp fault applied to the treator water volume sensor (S_3) at t=200 seconds; the corresponding time histories for the separator and treator area attached in appendix A, Figs. A.17 and A.18.

In Fig. 7.4 it is observed that the static $|GPV|$ increases significantly right after the fault is applied, giving a detection time of 1.7 sec. (using n=2 during the M_T computation). At the same, the S_3 GPV angle moves rapidly towards zero reaching a minimum separation angle of approximately 5 degrees within the first 8.095 sec. These detection and isolation times are excellent taking into account that the fault varies very slowly with time as shown in Fig. A.18. Conversely, it is seen from Fig. 7.5 that the dynamic GPV takes 12.4 sec. to detect the fault and 40 sec. to isolate it using the same threshold criterions; these detection and isolation times are approximately 7 and 5 times longer than the ones obtained using the static GPV. Depending on the application, these delays to perform the diagnosis might cause safety issues or product quality degradation.

To illustrate the convenience of using the dynamic GPV implementation for bias fault isolation, a -15% bias fault is applied to the treator oil volume sensor (S_4) at t=60

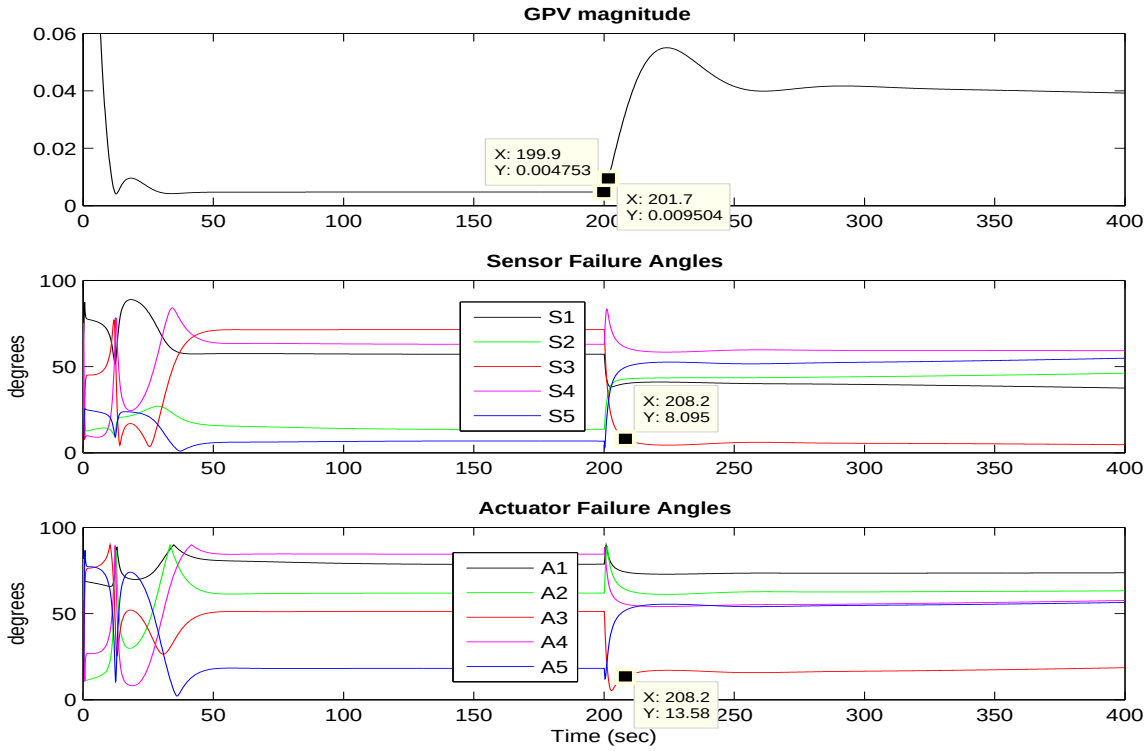

Figure 7.4: Static $|GPV|$ and $\angle GPV$, S_3 faulty

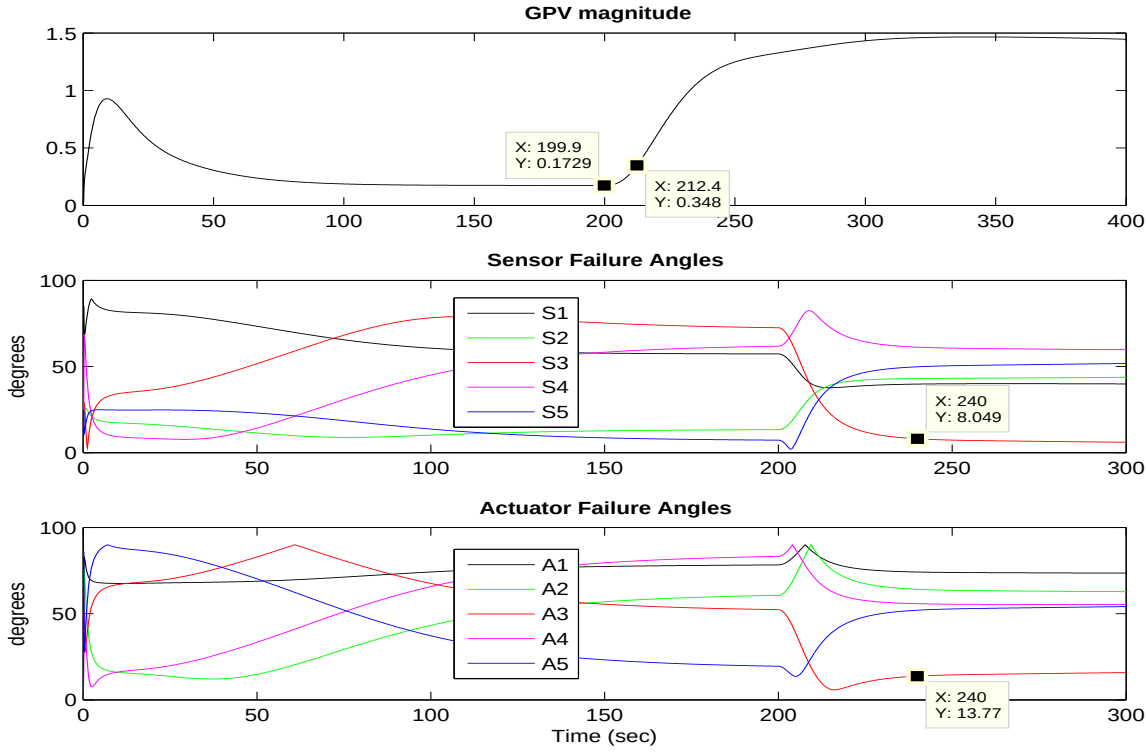

Figure 7.5: Dynamic $|GPV|$ and $\angle GPV$, S_3 faulty

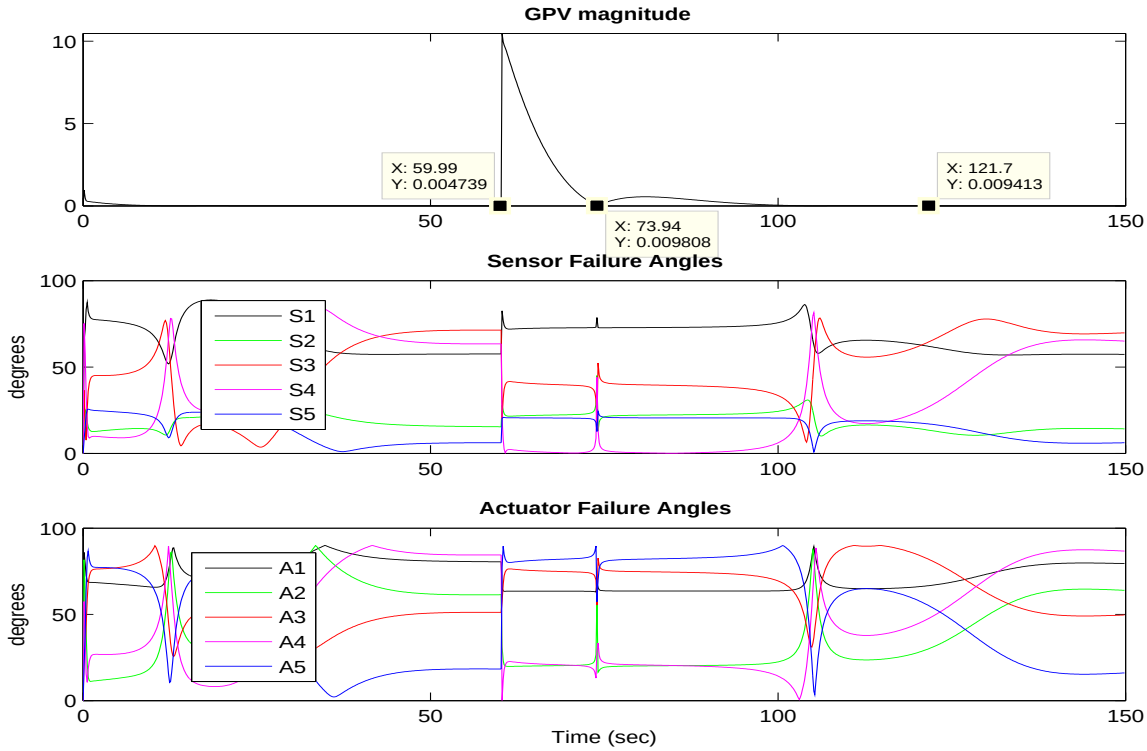

Figure 7.6: Static $|GPV|$ and $\angle GPV$, S_4 faulty

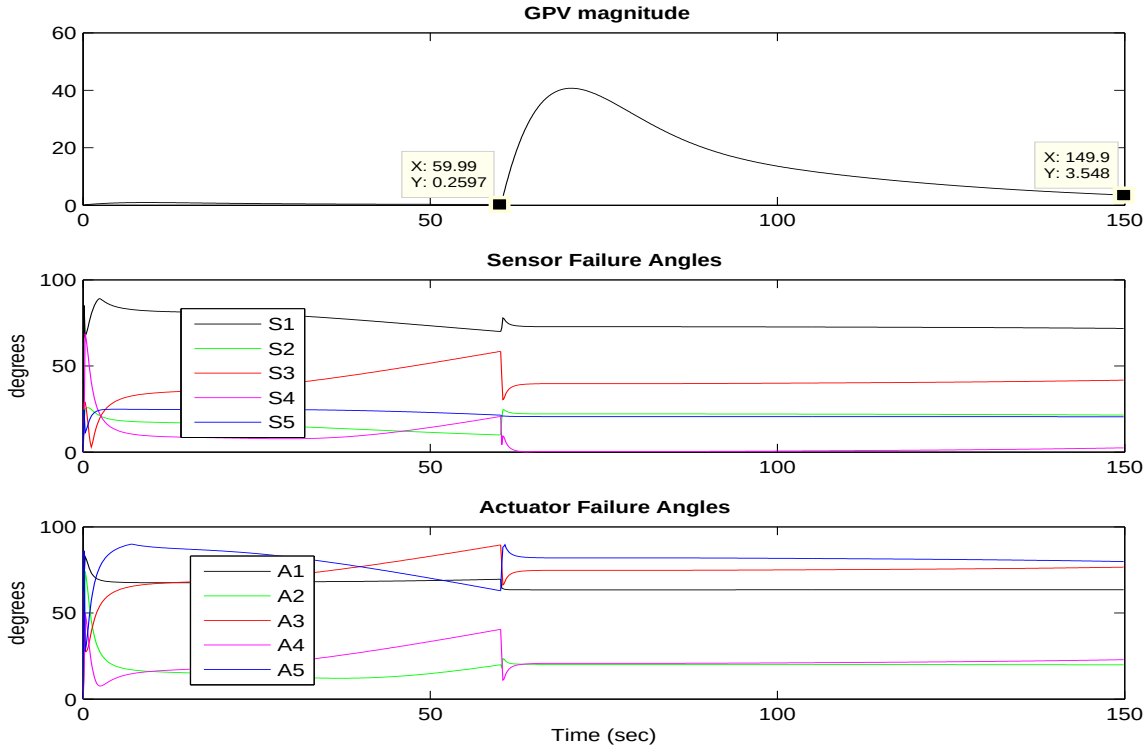

Figure 7.7: Dynamic $|GPV|$ and $\angle GPV$, S_4 faulty

sec. The corresponding time histories for the separator and treator are attached in appendix A, Figs. A.23 and A.24. Since a fault in sensor 4 only affects loops 4 and 5, the fault is quickly rejected by the controller and all the input-output measurements return to their fault free values as it is seen in Figs. A.23 and A.24. However, S_4 is still faulty, making the actual oil volume in the treator significantly higher than the sensor reading. Faults that are quickly rejected by the controller are particularly challenging to identify given the short detectability period. In Fig. 7.6 it is observed that the static GPV magnitude decreases rapidly, reaching the fault free threshold conditions 61.7 sec. after the fault was detected. Also, when the GPV magnitude becomes very small (t=73.94 sec.), the directional behaviour is lost, making the isolation ambiguous. In contrast, the dynamic GPV magnitude in Fig. 7.7 decreases slowly and exhibits smooth angular behaviour. As a result, S_4 remains close to zero during the entire simulation time, providing clear isolation. These simulation results show that the combined static-dynamic GPV implementation proposed in the decision-maker logic improves significantly the FDI performance.

7.3.3 Fault isolation block

It should be noticed that the fault isolation methodology presented in this section is developed assuming a single fault occurrence. This assumption is based on the observation that in most real industrial processes is very unlikely to have simultaneous faults in sensors or actuators happening at exactly the same time. It is usually more likely to have consecutive faults, as a consequence of the effect of the first fault in the rest of the system. For this type of scenarios, the GPV technique has shown satisfactory results, as presented in section 8.5.1. In that case study it is shown to be possible to perform FDIA for a different consecutive fault, as long as the first one has been already isolated and accommodated.

After the fault has been classified and the suitable GPV implementation has been

decided, the fault isolation block in Fig. 7.3 is executed. Although the GPV is computed at every sample and detection and isolation are provided accordingly, eventually the fault will be rejected by the control loop making it undetectable. The impact of this situation in the FDI performance was analyzed in the case study shown in Figs. 7.6 and 7.7 in the previous section. The rejection period varies depending on the system time constants and the controller performance. However, given the features of the combined static and dynamic GPV logic, it is possible to detect and isolate the faults for a time window (T_w) that is long enough to make a definite isolation statement, even if the faults are quickly rejected by the controller.

The time window should be long enough to make sure that the final isolation statement corresponds to the actual faulty element, but at the same time, it should not be excessively long because this will delay fault accommodation. Also, if T_w is too long, the fault might be totally rejected and undetectable at the end of this window even using the dynamic GPV implementation. As a result, if the time window conditions in equations (7.1) to (7.3) are not satisfied, a final isolation statement is not achieved and sensor accommodation cannot be implemented.

The time window length depends on the system time constants and also varies according to the GPV behaviour for each fault scenario. T_w starts at the time that the fault is detected and finishes when the following conditions from Fig. 7.3 are satisfied:

$$\angle GPV^k_{min} = SS \tag{7.1}$$

$$max(\textit{Fault-counter}) > m\frac{\tau_{max}}{T_s} \tag{7.2}$$

$$\textit{Isolation-ratio} = \frac{\textit{Fault-counter(k)}}{\textit{Fault-counter(i)}} > q \tag{7.3}$$

The condition in equation (7.1) checks if the GPV has already reached its steady state direction, so the minimum GPV angle corresponding to the k^{th} element ($\angle GPV^k_{min}$) is pointing in a meaningful direction. The steady state condition is detected by the

data reconciliation agent as described in section 4.5. We define *"Fault-counter"* in equation (7.2) as an array that stores in the corresponding index the number of times that each element has had the minimum GPV angle and has been tentatively isolated. Thus, the condition in equation (7.2) guarantees that there is an element that has been declared faulty at least for a minimum number of samples given by the maximum system time constant (τ_{max}), the sampling time (T_s) and the parameter m, chosen such that the system outputs are settled before the final isolation statement is made. The condition in equation (7.3) is implemented to handle cases where there are two GPV angles close to zero at different times within the time window, making the isolation ambiguous; fault k is declared if its angle was small more often than the others, as defined by the safety factor $q > 1$. If the mathematical model used for the GPV implementation closely describes the system dynamics and the value of q for the isolation ratio condition is large enough, the isolated fault should correspond to the actual faulty element as was verified after extensive simulation using the separator model. The parameters m and q regulate the tradeoff between reliability and speed to make a final isolation statement. For the separator model in section 4.3, m=2 and q=2 provided a good tradeoff.

Following the fault isolation logic in Fig. 7.3, a final isolation statement is made if the time window defined by equations (7.1) to (7.3) has been completed. Otherwise, the isolation process based on the GPV angles continues. If the difference between the smallest GPV angle corresponding to element k ($\angle GPV^k_{min}$) and the second smallest GPV angle ($\angle GPV_2$) is greater than the threshold angle (Θ_T), a fault in the k^{th} element is isolated and its corresponding fault counter is increased. This fault counter is used to check the time window condition in equation (7.2). If the Θ_T is not met and the $\angle GPV^k_{min}$ has not reached steady state, the isolation is considered in progress and the "isolating fault" option is shown in the operator display. Conversely, if the $\angle GPV^k_{min}$ is already in steady state, but the minimum

separation angle is not large enough, it is possible to have the special case for sensor-actuator FDI discussed in section 3.4. If the pair of elements corresponding to $\angle GPV^k_{min}$ and $\angle GPV_2$ are a sensor/actuator pair satisfying equation (3.15), this means that the actuator that lies on the hyperplane is faulty. This diagnosis is made using the logic in equation (3.19). On the other hand, if $\angle GPV^k_{min}$ and $\angle GPV_2$ do not satisfy equation (3.15), both elements are displayed as possibly faulty in the operator display until T_w has been completed.

When the time window conditions are met, the isolation block is able to make a final isolation statement that is sent to the supervisor and also used for fault accommodation. The final isolation results will be continuously displayed in the operator panel until the faulty element is fixed. It should be remarked that for the sensor case is crucial to have correct isolation since accommodation will be performed accordingly. This validates the need to implement the logic in equations (7.1) to (7.3) using a conservative choice for the parameters m and q, instead of having a fixed time window.

7.4 Simulation results

In this section two simulation results are presented to show the operator display obtained using the decision-making logic proposed in section 7.3. We chose to apply a fault in sensor 5 and then in actuator 5, since this loop corresponds to the special case for sensor-actuator FDI illustrated previously in Fig. 3.3 and discussed in section 6.2. These two fault scenarios allow us to evaluate the decision-maker performance for challenging isolation situations.

The first scenario corresponds to a -25% bias fault applied to the treator vapor pressure sensor (S_5) at t=60 sec. The corresponding time histories for the separator and treator are shown in appendix A, Figs. A.19 and A.20 respectively. Following

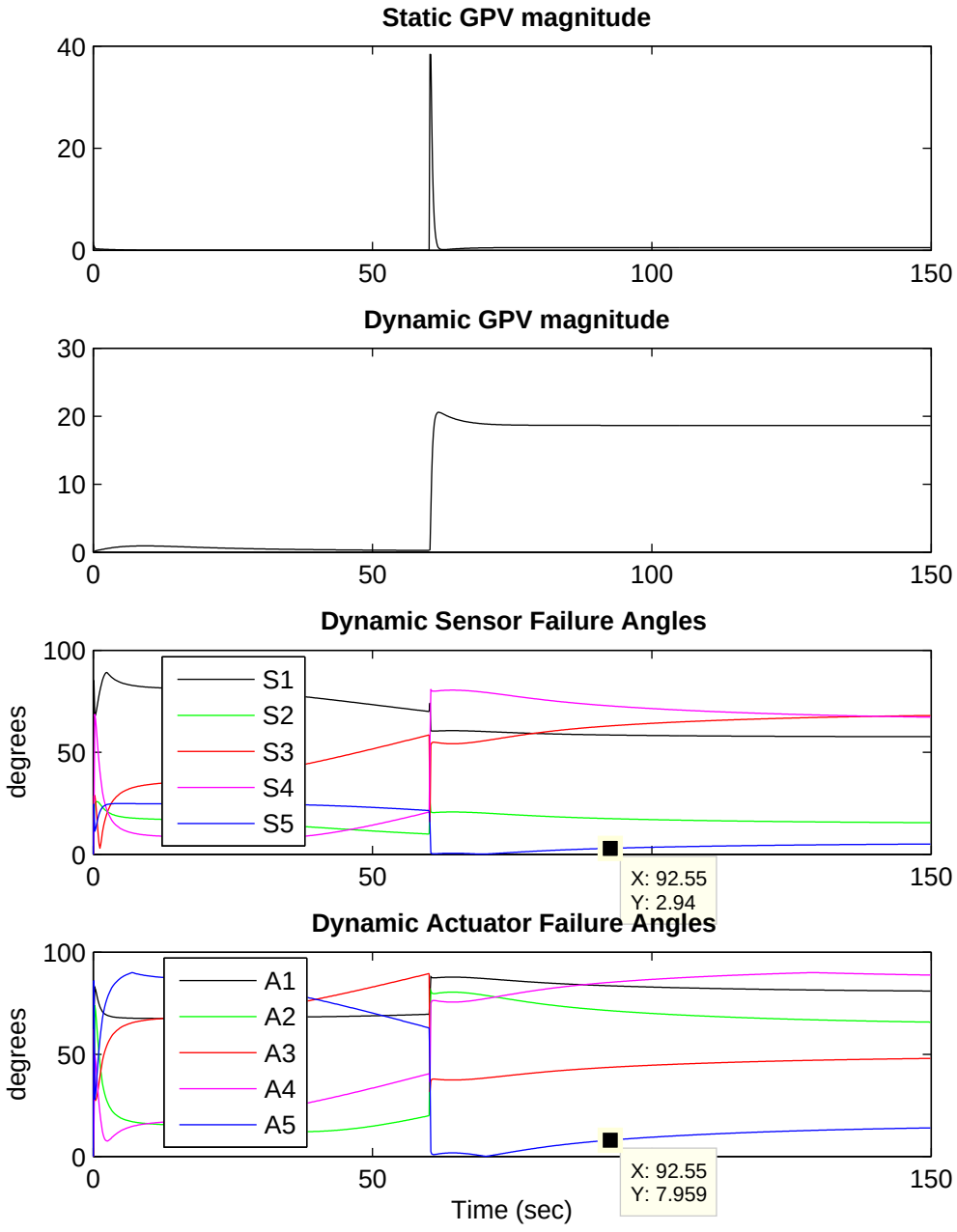

Figure 7.8: $|GPV|$ and $\angle GPV$, S_5 faulty

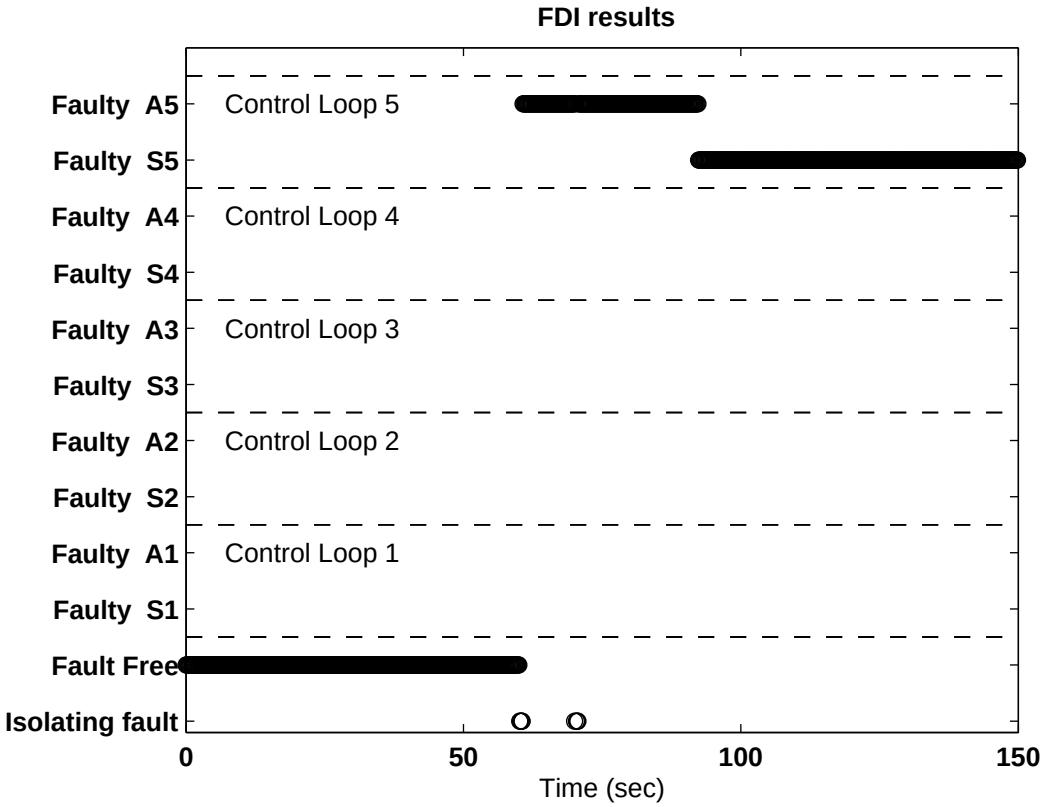

Figure 7.9: Operator display using the dynamic GPV, S_5 faulty

the decision-making logic described in section 7.3, the fault is quickly detected and classified as a bias fault using the static $|GPV|$ in Fig. 7.8. Since $F_{type} = bias$, the dynamic GPV is used to execute the isolation block. The operator display in Fig. 7.9 shows a fault in actuator 5 during the first 32.4 sec. after the fault was detected. This is a result of the A_5 angle slowly moving away from zero in Fig. 7.8. This result validates the use of equation (7.2), which states that the final isolation statement cannot be made before $2\tau_{max}$ seconds, where $\tau_{max} = 23.44$ sec. for this system. If the final isolation statement was done before $2\tau_{max} = 46.88$ sec. (without using equation (7.2)), A_5 would be wrongly declared the faulty element according to the results shown in fig. 7.8.

However, once the angle threshold of 5 degrees is achieved between S_5 and A_5 in Fig. 7.8, the operator display correctly isolates S_5 at t=92.55 sec. This isolation time is considered satisfactory, given that FDI for S_5 and A_5 is very challenging because

this pair correspond to the special case for sensor-actuator FDI, as mentioned. At t=150 sec. S_5 has been isolated for more than $2\tau_{max}$ sec. and also has doubled the isolation period for A_5. S_5 can also be considered steady enough, so the conditions in equations (7.1) to (7.3) are satisfied and the final isolation statement is made 90 sec. after the fault was applied. If faster accommodation is desired, the value of the parameter q in equation (7.3) can be reduced accordingly to the minimum value that meets the reliability standards for the given plant. At t=150 the FDI results are sent to the sensor accommodation block and the ICAM system supervisor. This diagnosis time provides a good tradeoff between reliability and isolation speed for the separator model. It should be noticed that a few samples showing the "isolating fault" option in the operator display are because the angle threshold is not satisfied and the minimum GPV angle is not in steady state.

The second scenario corresponds to a -25% bias fault applied to the treator vapor outflow valve (A_5) at t=60 sec. The corresponding time histories for the separator and treator are shown in appendix A, Figs. A.21 and A.22 respectively. Similarly, the fault is quickly detected and classified as a bias fault using the static GPV in Fig. 7.11, so the dynamic GPV implementation is chosen for isolation. Given that sensor and actuator 5 satisfy equation (3.15), the isolation is made based on the logic in equation (3.19). As a result, A_5 is clearly isolated almost immediately in Fig. 7.11, since the GPV lies not only in the S_5 reference hyperplane but also in the A_5 actuator reference direction as seen in Fig. 7.10. For this case the conditions in equations (7.1) and (7.3) are satisfied almost immediately after the fault was detected since the corresponding GPV angles reach steady state rapidly. However, the final isolation statement is not made until the condition in equation (7.2) is satisfied $2\tau_{max} = 46.88$ sec. after the fault is detected. For this particular scenario the final isolation statement could have been done faster by choosing a smaller value for the parameter m in equation (7.2). However, it was already shown in the previous

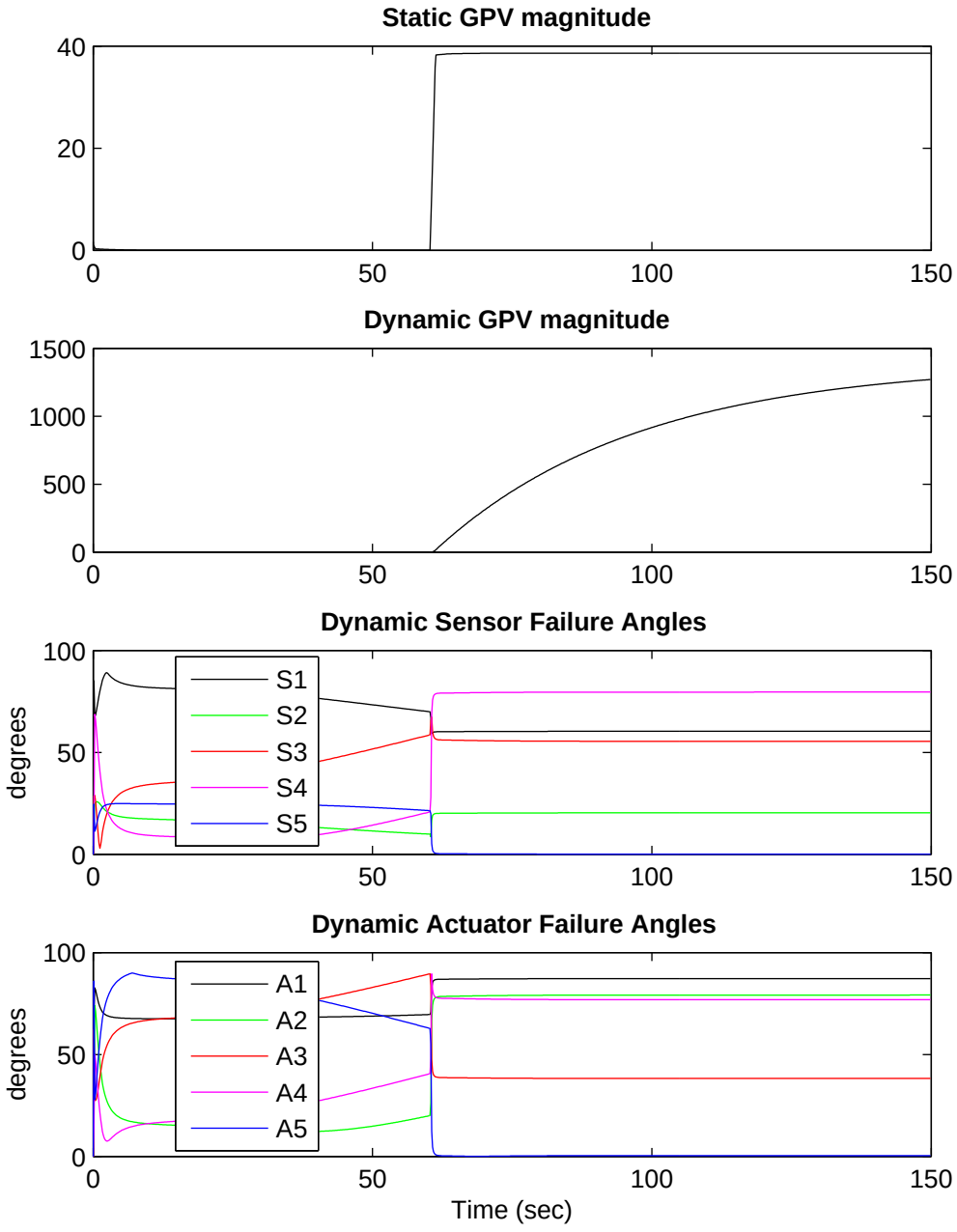

Figure 7.10: $|GPV|$ and $\angle GPV$, A_5 faulty

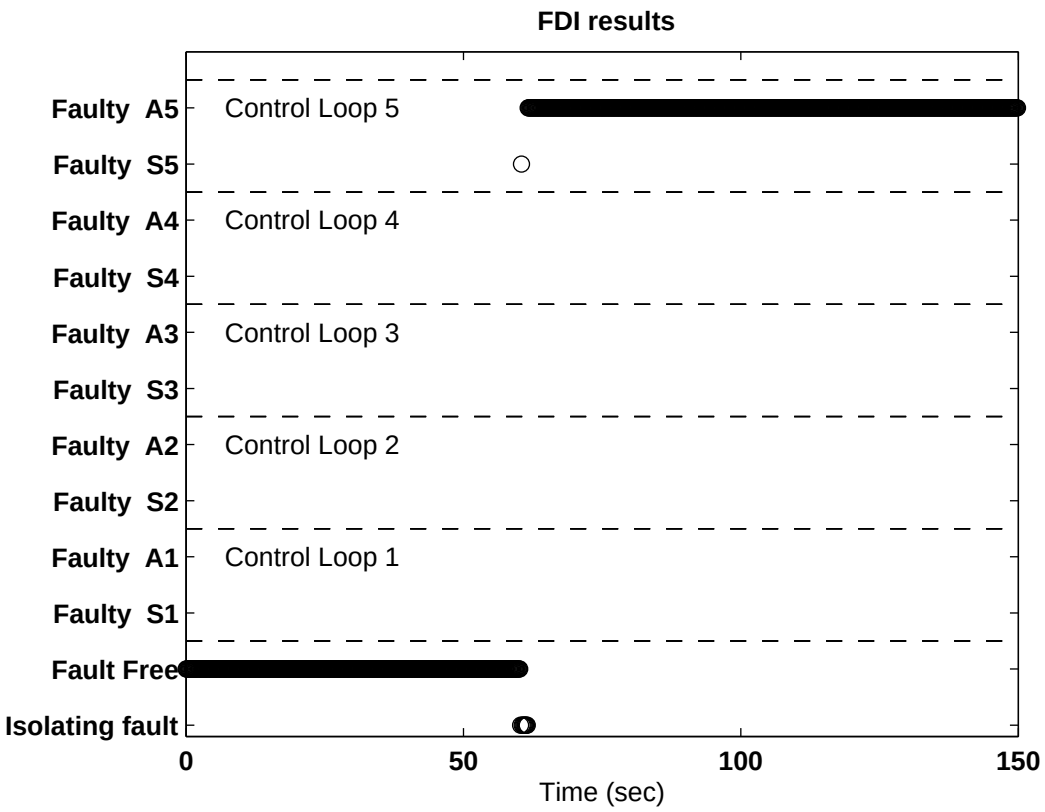

Figure 7.11: Operator display using the dynamic GPV, A_5 faulty

example that a value of $m = 2$ is required to avoid a wrong final isolation statement for that scenario. Therefore, a conservative value of $m = 2$ was necessary to provide a good tradeoff between reliability and speed to make the final isolation statement for all possible scenarios for the ten different faults.

Chapter 8

Fault Accommodation

8.1 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 maintain normal operation in the system, while maintenance can be scheduled without significantly disturbing the process.

So far, the GPV technique has been successfully tested for FDI using a second-order aircraft engine model [13], a third-order nonlinear model for a jacketed continuously stirred tank reactor (JCSTR) [5], [35], [37] and the fifth-order identified state-space model for the gravity three-phase separation process illustrated previously in Fig 5.3 [36]. In this chapter a new fault size estimation and sensor accommodation method is proposed and evaluated using the most challenging separation process. This allows us to introduce a complete fault detection, isolation and accommodation

(FDIA) technique using only input-output data as available information.

8.2 Fitting calculation block

The fitting calculation block illustrated in Fig. 8.1 is part of the initialization section described previously in section 4.6. The fitting calculation block is executed after the model is identified, the corresponding reference directions and hyperplanes are calculated, and a suitable transformation matrix is computed. At this point, we have all the information required to implement a linear simulator for the plant based on the identified model.

Using the linear simulator, three different fault sizes are applied to each sensor, i=1,2,...n for the bias and ramp cases. To generalize the logic in Fig. 8.1, the number of fault sizes is denoted by the variable N_{size}; since the relation between the GPV signature presented in section 7.2 and the fault size (F_{size}) was usually linear or at most quadratic, three F_{size} values were considered, enough to obtain an adequate fit in this study. The purpose of these linear simulations is to characterize the $|GPV|$ signature ($|GPV|_{sig}$) for different fault sizes and types, so a curve fitting equation for F_{size} estimation can be computed for each case.

For the bias case, the GPV magnitude signature is given by the delta of static GPV magnitude peak ($\Delta|GPV|_{peak}$). Conversely, for the ramp case, F_{size} is determined by the static GPV magnitude slope ($|GPV|_{slope}$) at the time that the fault is detected. In either case the corresponding $\Delta|GPV|_{peak}$ vs. F_{size} or $|GPV|_{slope}$ vs. F_{size} pairs are obtained based on the linear simulator results. Using these sets of data, the best fitting is calculated for each case, providing an equation for F_{size} as a function of $|GPV|_{sig}$, i.e., $\Delta|GPV|_{peak}$ or $|GPV|_{slope}$, depending on the declared fault type (F_{type}). The curve fitting computation starts assuming a linear approximation, as illustrated in Fig. 8.1. However, if the error % in the fitting results is not small

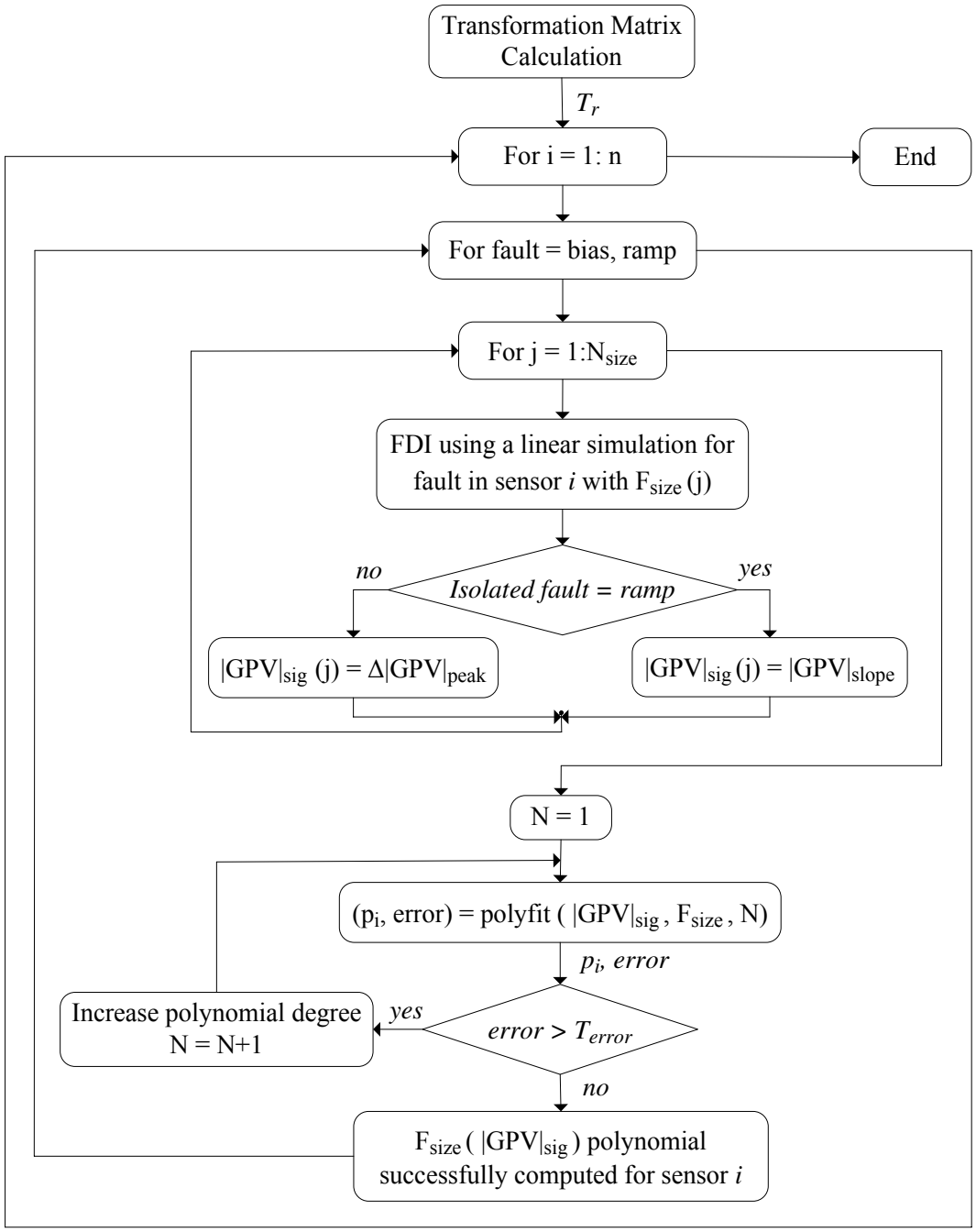

Figure 8.1: Fitting calculation block diagram

enough, the polynomial degree N is increased until the error % is acceptable. The curve fitting results were obtained using the *polyfit* function in MATLAB®. This function finds the coefficients p_i of a polynomial of degree N that fits the data best in a least-squares sense and it also provides the error estimates for predictions.

The total computation time for the fitting calculation block using the separator model was 4.2 minutes using a Pentium 4 PC. This execution time can be considered fast enough to make it viable for on-line implementation in most real industrial processes, since this block is performed only infrequently, when the identified model changes. Tables 8.1 and 8.2 summarize the fitting results obtained for the 5^{th} order identified model in Fig. 5.3 during the on-line initialization process. It should be noticed that the degree N in tables 8.1 and 8.2 refers to the degree of the polynomial that describes how the GPV signature changes for different fault sizes; it is not related to the order of the identified process model.

Fault	Coefficients p_i	Degree N	Fitting error
S_1	[0.47 0]	1	0.21×10^{-13}
S_2	[0.14 0]	1	0
S_3	[2.90 0]	1	0.04×10^{-13}
S_4	[1.43 0]	1	0.10×10^{-13}
S_5	[0.03 0.15 0]	2	0.01×10^{-13}

Table 8.1: Curve fitting results for bias faults

The polynomial for fault size estimation computed by the fitting calculation block is defined as follows:

$$F_{size} = p_1 \times |GPV|_{sig}^{N} + p_2 \times |GPV|_{sig}^{N-1} + ... + p_N \times |GPV|_{sig} + p_{N+1} \qquad (8.1)$$

Fault	Coefficients p_i	Degree N	Fitting error
S_1	[0.27 31.10 0]	2	0.002×10^{-13}
S_2	[2.37 26.71 0]	2	0.02×10^{-13}
S_3	[94.39 293.77 0]	2	0.02×10^{-13}
S_4	[424.88 318.40 0]	2	0.01×10^{-13}
S_5	[33966.85 1969.74 0]	2	0.001×10^{-13}

Table 8.2: Curve fitting results for ramp faults

where the coefficients p_i for each type of fault are given in the corresponding table in descending powers. It is observed that the maximum fitting degree $N = 2$ provided a fitting error less than 1×10^{-13} overall. The small values for the fitting error using low-degree polynomials were expected, since the change in the $|GPV|_{sig}$ with respect to the fault size was usually linear or quadratic at the most.

8.3 Fault-size estimation

After the fitting calculation block is executed, the curve fitting equations for each bias or ramp sensor fault are available to estimate F_{size}. The fault size is computed by evaluating equation (8.1) with the corresponding GPV magnitude signature. The GPV magnitude signature is given by $\Delta|GPV|_{peak}$ or $|GPV|_{slope}$, depending on the F_{type} diagnosis made by the decision-maker block.

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 ICAM system supervisor can send to the FDIA agent without overloading the network. For our specific application, the PAWS project, a wireless sensor network agent manages real-time communications between the control room and the offshore

oil facility as presented previously in sections 4.2 and 4.4. For further information on the PAWS project see [47]. Therefore, the frequency of the data set received in the control room is also restricted by the wireless network specifications.

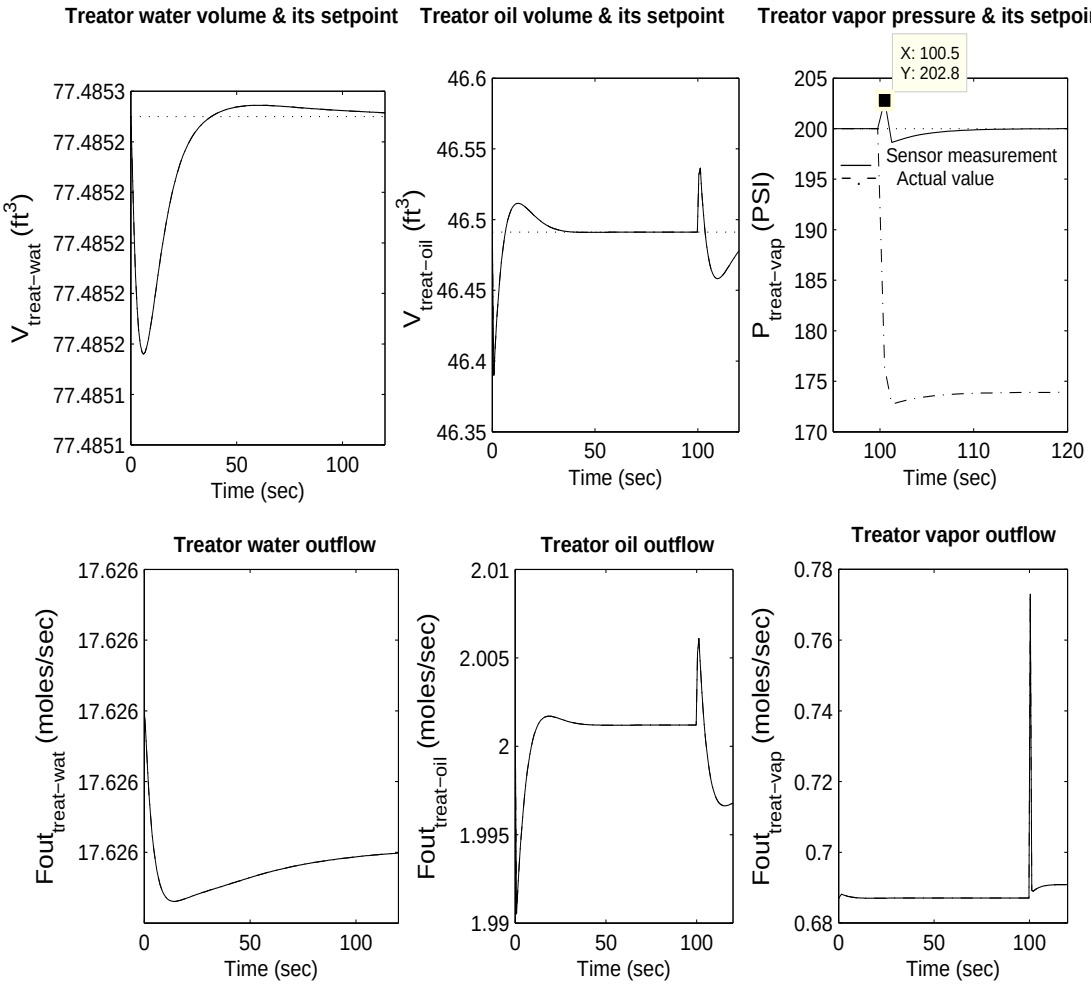

Figure 8.2: Treator time histories

Figures 8.2 and 8.3 show the simulation results using the identified separator model described in Fig. 5.3 for a +15% bias fault applied to the treator vapor pressure sensor (S5). We observe in Fig. 8.3 that the GPV angle corresponding to fault 5 is the smallest, giving a clear isolation. 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 seconds, which is between two sampling intervals. Since the

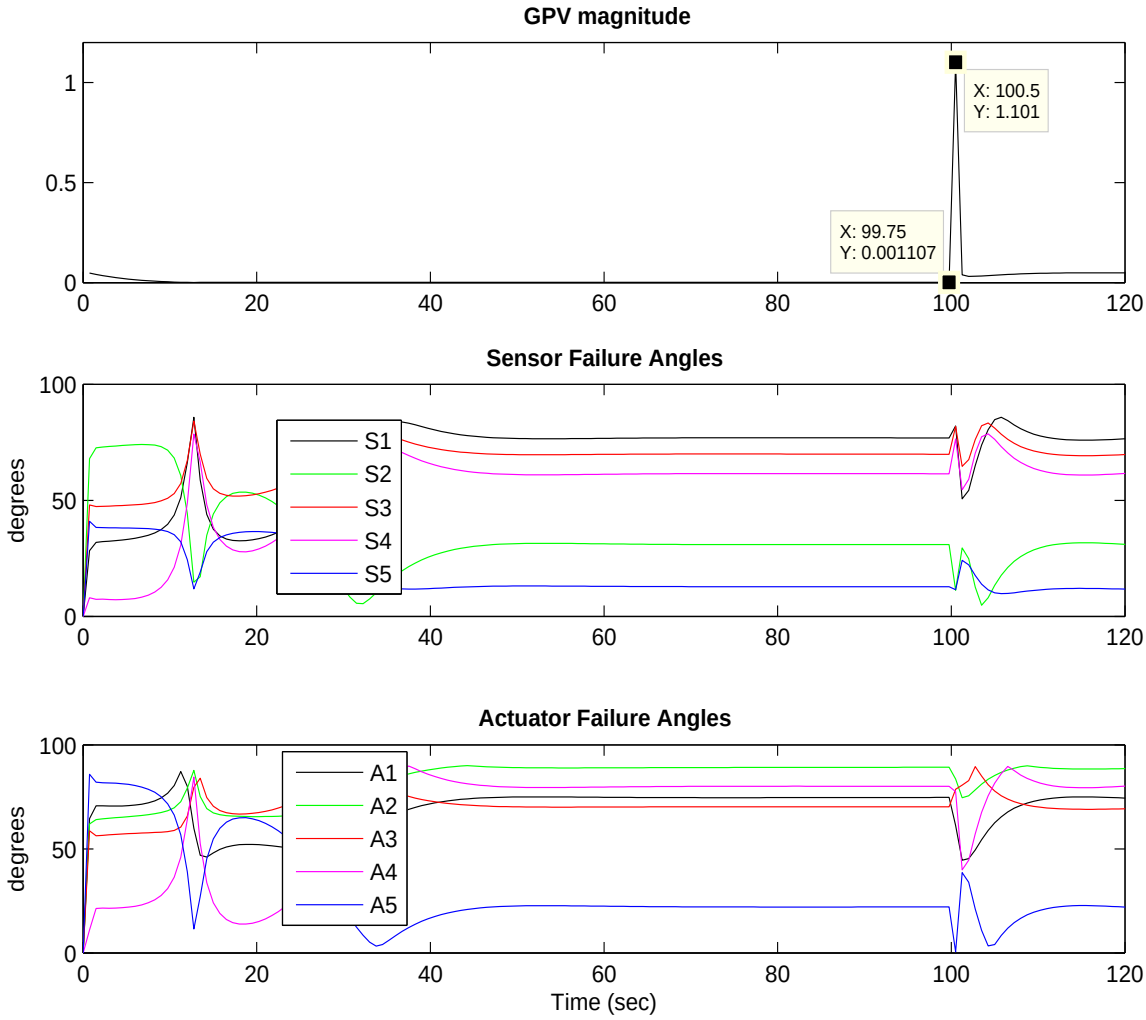

Figure 8.3: Static $|GPV|$ and $\angle GPV$, S5 faulty

sampling period in this simulation is 0.75 seconds, the last pressure reading available before the fault occurs is at t=99.75 sec. In Fig. 8.2, it is observed that the pressure measurements change rapidly during the first 0.9 sec. after the fault happened. This behaviour was expected, due to the fast dynamic nature of the pressure and the appropriate controller compensatory action. From previous simulation results it was established that it is possible to accurately estimate the fault size using the sensor measurement only for a sampling time of T_s=0.3 sec. or less. For this sampling period the first reading after the fault occurs is 230 PSI at t=100.2 seconds, giving

the correct fault size of +15%. More realistically, if the sampling period is 0.75 seconds, the first pressure reading available after the fault was applied is 202.8 PSI at t=100.5 sec. as illustrated in Fig. 8.2, giving a wrong fault size of +1.4% which has a 90.67% estimation error. Similarly the sampling time would also affect the fault classification if it was made based on the sensor measurement and not the GPV signature proposed in section 7.3.2. If F_{type} is defined from the sensor measurement, the slope change might not be large enough to declare a bias fault, depending on the sampling period.

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 previously. From Fig. 8.3 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. This abrupt change allows the classification of this fault as bias type as proposed in section 7.3.2. This makes it possible to use the $|GPV|$ value at t=100.5 sec. to calculate $\Delta|GPV|_{peak}$ for $F_{size} = 15\%$ during the on-line curve fitting procedure to obtain the function in equation 8.1 [35]. After the fault type and size are defined, the accommodation block is implemented as described in section 8.4.

8.4 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 system reliability and safety, extends useful life, minimizes maintenance and maximizes performance. In general, sensor accommodation can be accomplished using system reconfiguration or by modifying or replacing the faulty signal.

For the first method, several techniques have been proposed during the past decades

for fault tolerant control (FTC) systems design to reduce the effect of faults and perturbations simultaneously in the closed loop system [24], [25], [26], [27]. The aim is to find the appropriate control law that preserves stability, where "appropriate" is meant with respect to given objectives, which depend on the application. Although these FTC techniques are very important in many applications, their significance is reduced for scenarios where faults are rapidly rejected. As was discussed previously in section 7.3.2, faults in sensor 4 are quickly rejected by the controller, making them untraceable, while the sensor is still faulty and the actual oil volume in the treator is significantly higher, as illustrated in Figs. A.23 and A.24. This situation is even worse for the ramp fault cases in Figs. A.15 and A.16, where the oil volume keeps increasing while the control loops seem to have returned the system to normal operation. For the separator process described in section 4.3, the quick fault rejection is also present in sensors 3 and 5, which makes this issue one of our challenges to solve with the proposed FDIA technique. For the second method, different approaches using observers and neural networks have been used to exploit analytical redundancy and replace the faulty measurements with synthetic ones [28], [29], [30]. However, their performance is highly dependent on model accuracy.

To overcome the limitations of these methods, we have developed two new approaches for fault accommodation using the fault size estimation results obtained in section 8.3. Sensor fault accommodation is implemented after the fault-size estimation block is executed, as depicted in Fig. 4.2. It should be noticed that 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. Depending on the controls systems hardware and software configuration, the following two methods can be implemented to compensate for a sensor fault.

8.4.1 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 faulty 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 $F_{size.bias}$ and/or the estimated relative rate for the ramp case, $F_{size.ramp}$, using the corresponding mathematical relations given in (8.2). The fault sign, denoted as $\pm$ in equation (8.2), is obtained from the faulty measurement slope change around its setpoint, right after the fault is detected. The corrected measurement is then given by:

$$Y_{meas.corr} = \begin{cases} \frac{Y_{meas}}{\frac{\pm F_{size.bias(\%)}}{100}+1} \times u(t_k - t_{f1}), & F_{type} = bias \\ \frac{Y_{meas}}{\frac{\pm F_{size.ramp(\%/min)}}{100} \times \frac{t_k - t_{f2}}{60}+1} \times u(t_k - t_{f2}), & F_{type} = ramp \end{cases} \tag{8.2}$$

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

8.4.2 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

t_k, a relative delta setpoint ΔY_{sp} is calculated with the fault size estimated for the bias case $F_{size.bias}$ and/or estimated for the ramp case $F_{size.ramp}$, using the mathematical relation in (8.3). The accommodated setpoint value $Y_{sp.acc}$ is calculated using equation (8.4),

$$\Delta Y_{sp} = \begin{cases} \frac{F_{size.bias(\%)}}{100} \times u(t_k - t_{f1}), & F_{type} = bias \\ \frac{F_{size.ramp(\%/min)}}{100} \times \frac{t_k - t_{f2}}{60} \times u(t_k - t_{f2}), & F_{type} = ramp \end{cases} \tag{8.3}$$

$$Y_{sp.acc} = (1 + \Delta Y_{sp}) \times Y_{sp.orig} \tag{8.4}$$

It should be noticed that accommodation has been proposed to provide a temporary solution to maintain normal operation in the system, while maintenance can be scheduled without significantly disturbing the process. However, the plant is not supposed to run indefinitely under faulty circumstances because accommodation is not 100% accurate and it is also limited by sensor and actuator saturations. In particular, for the ramp case, a given setpoint can only be increased or decreased for a period of time while it reaches the maximum or minimum values specified for the corresponding variable. As a result, appropriate ramp fault accommodation using method 2 can only be provided for a limited period of time. Nevertheless, this period where accommodation can be performed provides valuable time to arrange maintenance and replace the faulty sensor.

Since this accommodation technique manipulates the setpoint, its performance depends on how fast the system responds to apparent setpoint change. Thus, for variables with slow dynamics, the accommodation process using this method takes longer to completely accommodate the fault due to the large settling time. Conversely, accommodation using method 1 is faster because the faulty measurement is directly corrected before it is sent to the controller. Although for the scenario considered in method 2 the sensor output cannot be manipulated before it is sent

to the controller, it is still possible to correct this measurement for GPV calculation purposes, using equation (8.2). By using $Y_{meas.corr}$ instead of y to generate the residual using equation (2.25), the GPV is also compensated. Thus, when the system is completely accommodated, the $|GPV|$ returns to its fault free value, showing the "fault free" and the corresponding faulty element options in the operator display.

Extensive simulations results have shown excellent F_{size} estimation and accommodation results, although our identified model for the separator does not have very high % of fitting for sensors 4 and 5 due to the complexity of the process. Only for these variables whose % of fitting was less than 85%, the F_{size} estimation accuracy was lower for some fault scenarios. While the accommodation was not perfect for these cases, it still improved the performance of the faulty system significantly.

The main contribution of our method over the traditional measurement replacement using observers is that it is less sensitive to modeling errors. This is due to the fact that F_{size} is estimated using $\Delta|GPV|_{peak}$ or $|GPV|_{slope}$ instead of absolute $|GPV|$ values, which reduces significantly the impact of modeling errors in the F_{size} estimation. In other words, using deltas compensates for the fact that the $|GPV_{ff}|$ obtained using the linear model during the curve fitting calculation would be zero, while the $|GPV_{ff}|$ using the actual plant is not, so the inherent plant nonlinear behaviour is retained. Since the two methods presented above are implemented based on the computed F_{size}, the effect of modeling errors in sensor accommodation is reduced. Conversely, if the identified model is used to implement an observer to replace the faulty sensor measurement, the computed sensor output would be directly affected by the model's quality and there would not be a way to compensate for it.

8.5 FDIA Simulation results

This section presents various simulation results to show the capabilities of the complete FDIA technique using the methods described in section 8.4 for sensor accommodation. For all the scenarios, the GPV is implemented using the 5^{th} order identified model illustrated in Fig. 5.3. The results are obtained using the initialization section proposed in section 4.6, so the FDIA logic is implemented using just input-output measurements as the only available information for FDIA design. In fact, although we have the separator nonlinear model, it is not possible to derive the linearized state space representation analytically due to the complexity of the process [47]; this is generally the case in real industrial applications. Therefore, the only practical way to implement the GPV technique is by using an identified model, as described in chapter 5.

8.5.1 FDIA results using method 1

In this section two fault scenarios are simulated to evaluate the FDIA performance using method 1. The first scenario is described as follows: a -7% bias fault is applied to the treator water volume sensor (S_3) at t=60 sec. After this fault has been completely accommodated, a +1%/min ramp fault is applied to the separator liquid volume sensor (S_1) at t=500 sec. The first fault affects only the treator inputs and outputs (Fig. 8.5), and the second fault primarily affects the separator variables (Fig. 8.4). From Fig. 8.6 it is observed that the static $|GPV|$ increases significantly right after the fault is applied, giving a fast detection time of 0.15 sec. Also, the $|GPV|$ change is very abrupt, so the fault is classified as a bias and the dynamic GPV is calculated to implement the isolation block. Figure 8.7 shows that the corresponding GPV angle S_3 moves towards zero rapidly, providing a quick and clear isolation in the operator display in Fig. 8.8. At t=108 sec. the decision-making

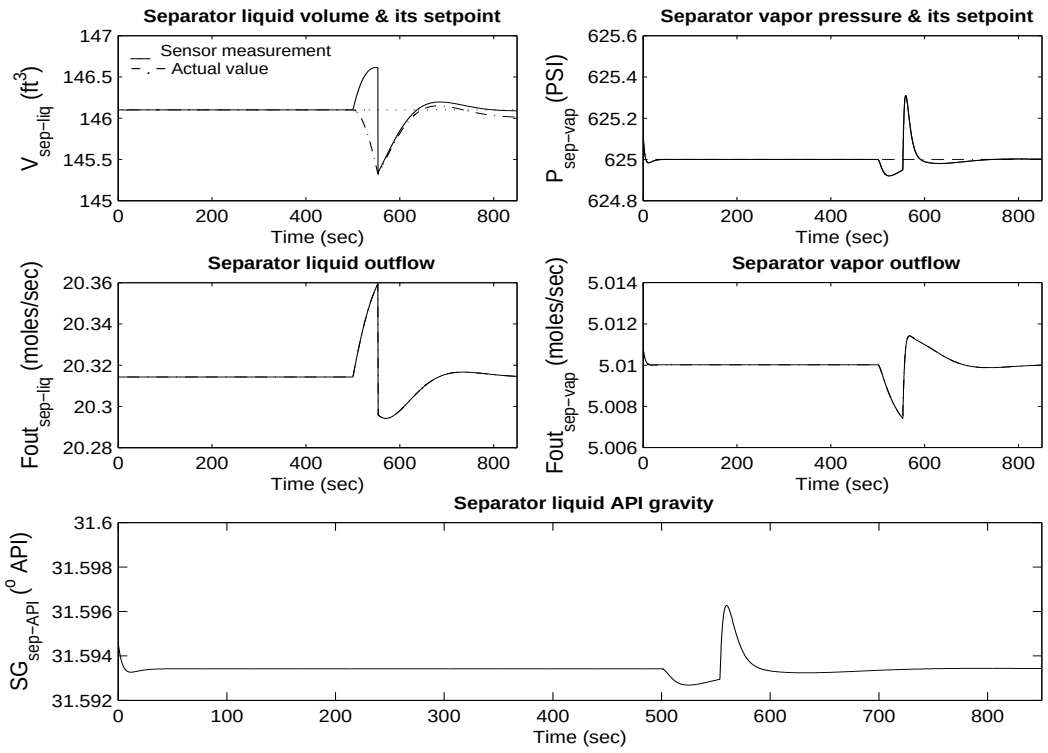

Figure 8.4: +1%/min ramp S_1 applied at t=500 sec. - Nominal Operating Point

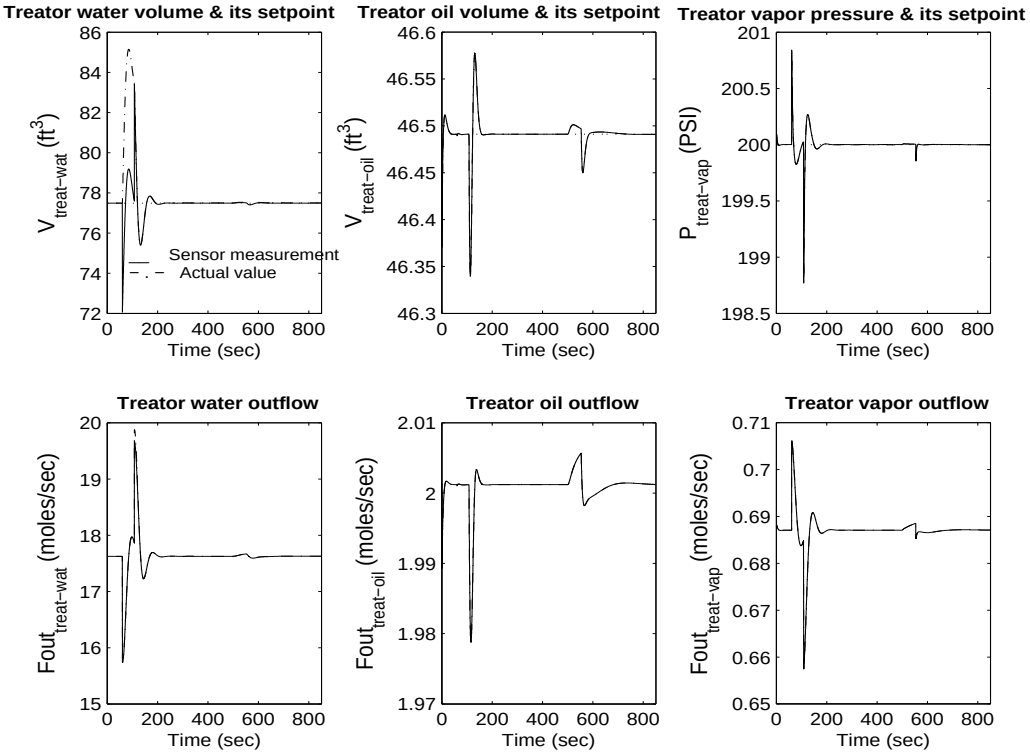

Figure 8.5: -7% bias S_3 applied at t=60 sec. - Nominal Operating Point

logic gives the final isolation statement for S_3 that $F_{size.bias} = -6.98\%$, thus sensor fault accommodation starts at that time. The estimated F_{size} provided by the FDIA software is shown at the top of the operator display in Fig. 8.8.

In Fig. 8.5 it is seen that when the faulty measurement is replaced at t=108 seconds, the actual water volume in the treator is quickly driven back to its setpoint value and the static $|GPV|$ returns to its $|GPV|_{ff}$. There is a short transient when the accommodation starts, while the controller is driving the compensated measurement back to its setpoint value. This transient is also shown as a temporary $|GPV|$ increase at t=108 sec. in Figs. 8.6 and 8.7. As a result, the operator display in Fig. 8.8 shows the "Fault Free" and "Faulty S_3" options, which means that the fault has been successfully accommodated, but S_3 is still faulty and maintenance should be scheduled. It should be noticed that once the final isolation statement has been made and the accommodation has started, the GPV angles are not taken into account anymore since the $|GPV|$ is returning to its $|GPV|_{ff}$ value. After the sensor accommodation starts, the decision-making logic switches back to the static $|GPV|$ to check the accommodation status. This decision is due to the faster response of the dynamic GPV implementation as it was discussed in detail in section 7.3.

Later at t=500.25 seconds, the static $|GPV|$ in Fig. 8.6 increases, detecting a second fault. For this case the $|GPV|$ change at the time that the fault is detected (t_{fault}) is not abrupt, thus the fault is classified as a ramp type. It is observed in Fig. 8.6 that the $|GPV|$ signature at the time that the faults were applied, t=60 sec. and t=500 seconds, is significantly different for the bias and ramp fault cases. It should also be noticed that if a very small bias fault is applied, i.e. $\pm 1\%$, most likely the FDI algorithm will not be able to detect the fault because its effect in the system variables is very small, thus the static $|GPV|$ increase will not be enough to exceed the magnitude threshold for detection. However, a ramp fault, even if it is a very

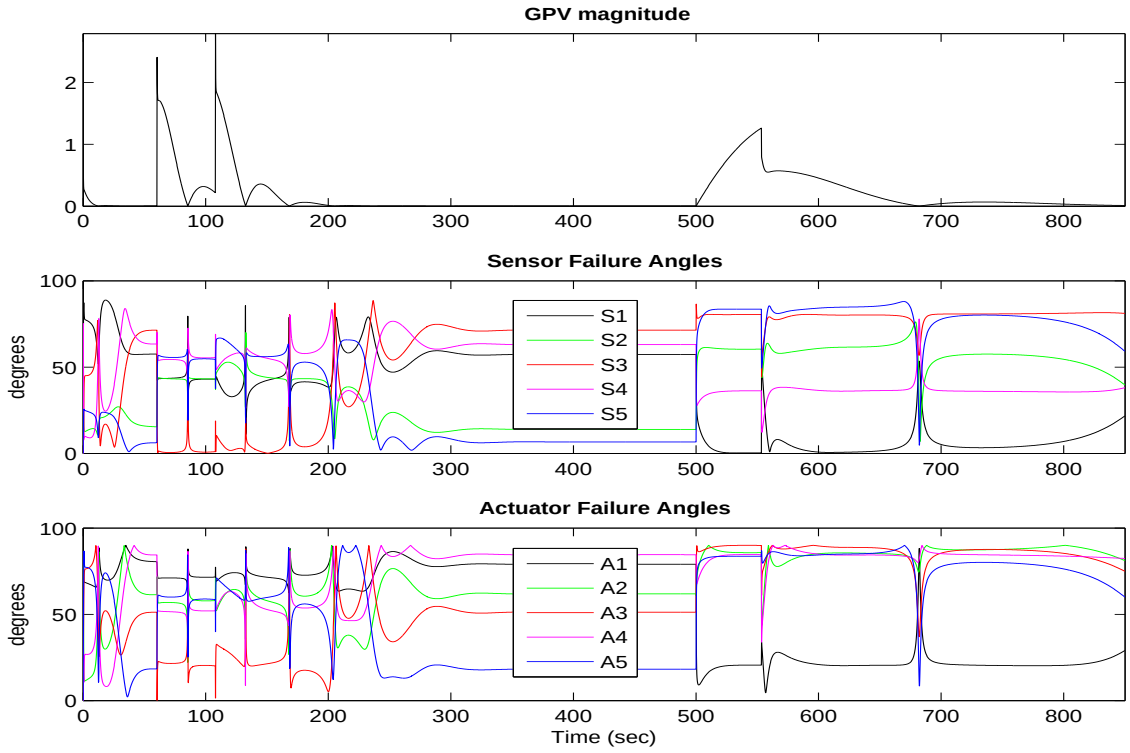

Figure 8.6: Static $|GPV|$ and $\angle GPV$, S_3 and S_1 faulty

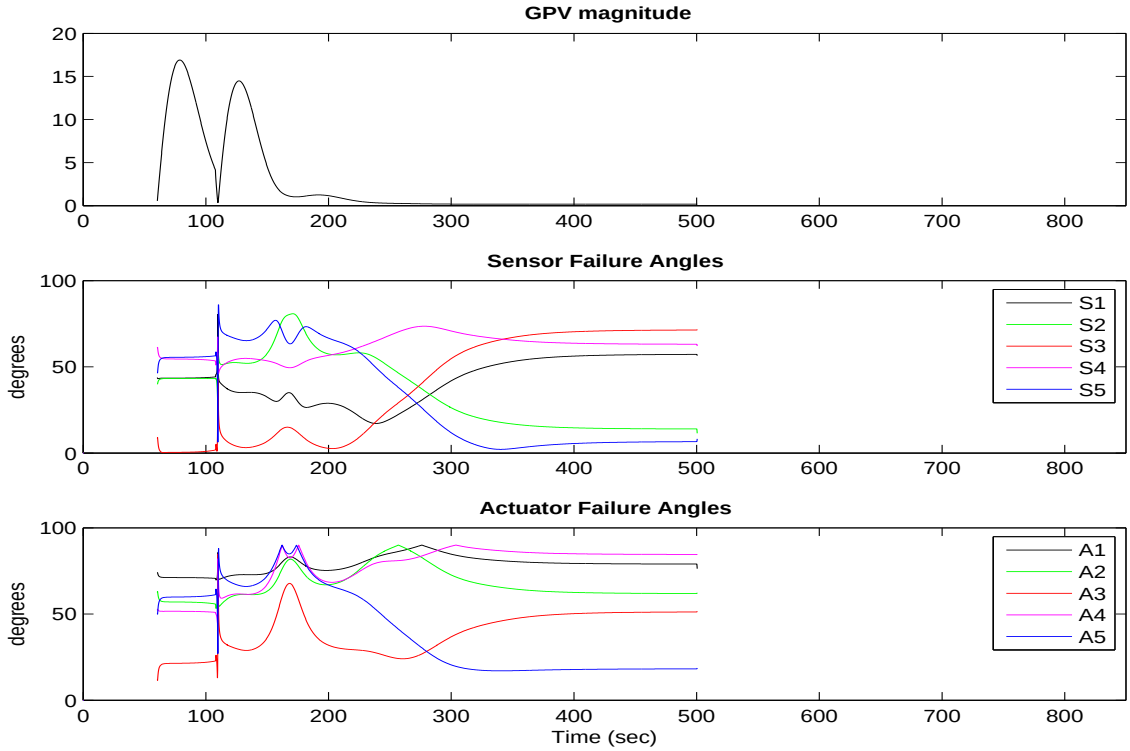

Figure 8.7: Dynamic $|GPV|$ and $\angle GPV$, S_3 and S_1 faulty

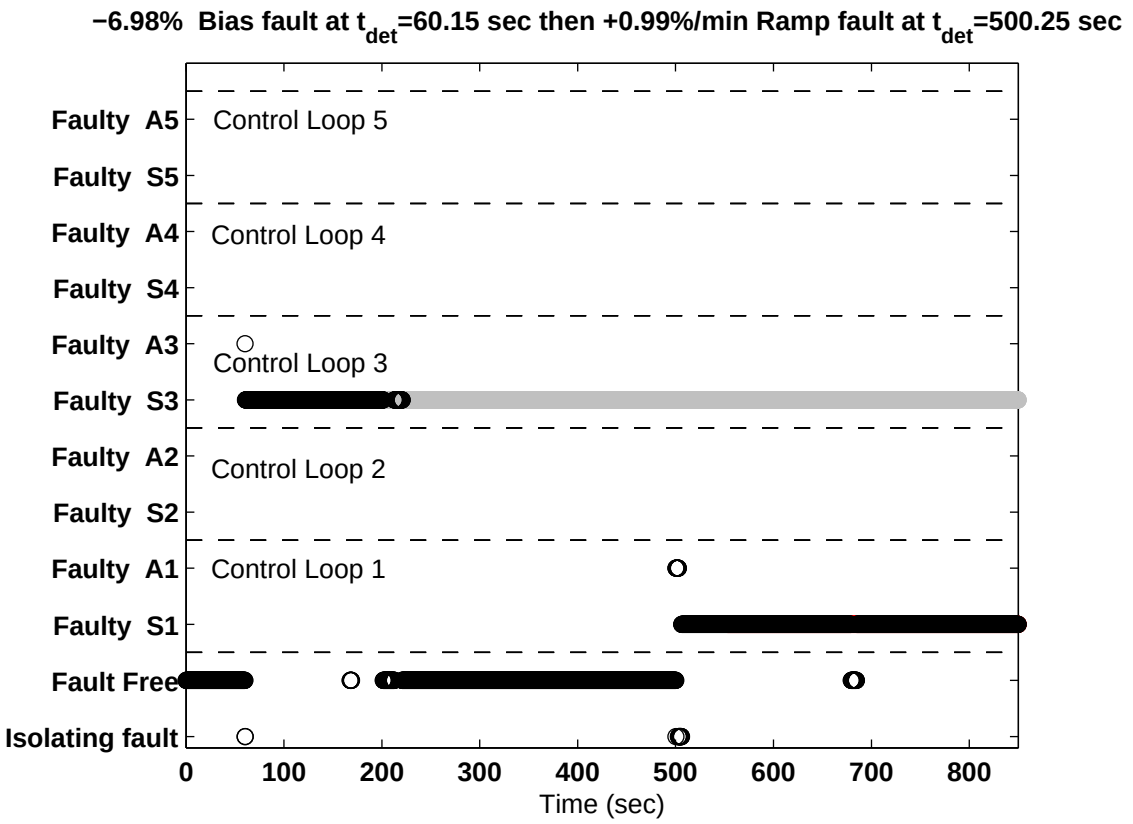

Figure 8.8: Operator display, S_3 and S_1 faulty

slow one, will cause a continuous increase in the static $|GPV|$ which will make it eventually exceed the detection threshold and show a $|GPV|$ signature with a very slow slope. Based on the decision-making logic presented in section 7.3 for the ramp case, fault isolation is performed using the static GPV angles, so the calculation of the dynamic GPV is stopped to decrease the computation time (Fig. 8.7). The minimum GPV angle in Fig. 8.6 corresponds to S_1, showing a correct isolation result in the operator display. The few samples showing the "Isolating fault" and "Faulty A_1" options in Fig. 8.8 happen only during the first 6 seconds after the fault was detected, while the GPV reaches its steady state direction.

At t=553 sec. the decision-making logics reaches the final isolation statement for S_1, i.e., $F_{size.ramp} = 0.99\%$ and the accommodation process starts. Figure 8.4 shows that once the accommodation starts, the actual liquid volume in the separator increases

towards its setpoint value. For the second fault the static $|GPV|$ does not reach its fault free value during the simulation time, so that is why the fault free option is not displayed. However, the operator panel is still showing the "Faulty S_3" option in gray, given that the ICAM system supervisor has not acknowledged that the sensor is fixed. The estimated fault sizes using the method described in section 8.3 are -6.98% and 0.99%/min, giving an estimation error of only 0.29% and 1% respectively.

The second simulation scenario is intended to show the FDIA robustness with respect to modeling errors; it is described as follows: a +15% bias treator oil volume sensor (S_4) fault applied at t=250 sec. for an initial setpoint variation (Δ_{SP}) of 5% for all the variables. In other words, all process setpoints were changed by 5% and process variables were settled to their corresponding steady state values at t=250 sec. Thus the nominal operating point around which the linear model was identified does not correspond to the current setpoint. All process variables are depicted in Figs. 8.9 and 8.10.

It is observed in Fig. 8.11 that the static $|GPV|$ increases abruptly at t=250.15 sec. Thus, the fault is quickly detected and classified as a bias type, 0.15 sec. after it happens, for a sampling period of 0.15 sec. As a result, the isolation is made based on the dynamic GPV shown in Fig. 8.12. Sensor 4 is clearly isolated in the operator display in Fig. 8.13 since its GPV angle in Fig. 8.12 is almost zero after the fault is detected. The time window criteria discussed in section 7.3.3 is satisfied at t=300 seconds, so the decision-maker logic makes the final isolation statement and accommodation starts then. Figure 8.10 shows that the actual treator oil volume is rapidly compensated and its setpoint is almost reached. Also, the static $|GPV|$ in Fig. 8.11 moves back towards its fault free value, showing that the fault has been successfully accommodated by displaying both the "Fault Free" and "Faulty S_4" options in the operator display in Fig. 8.13.

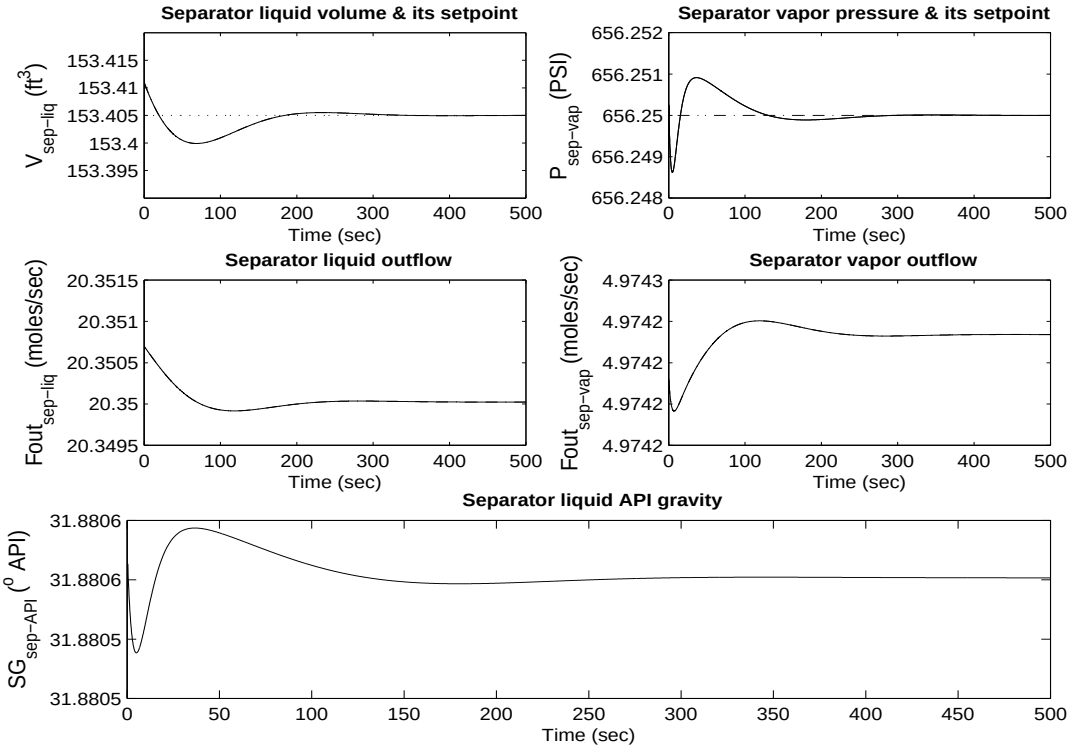

Figure 8.9: +15% bias S_4 applied at t=250 sec. - ΔSP=5%

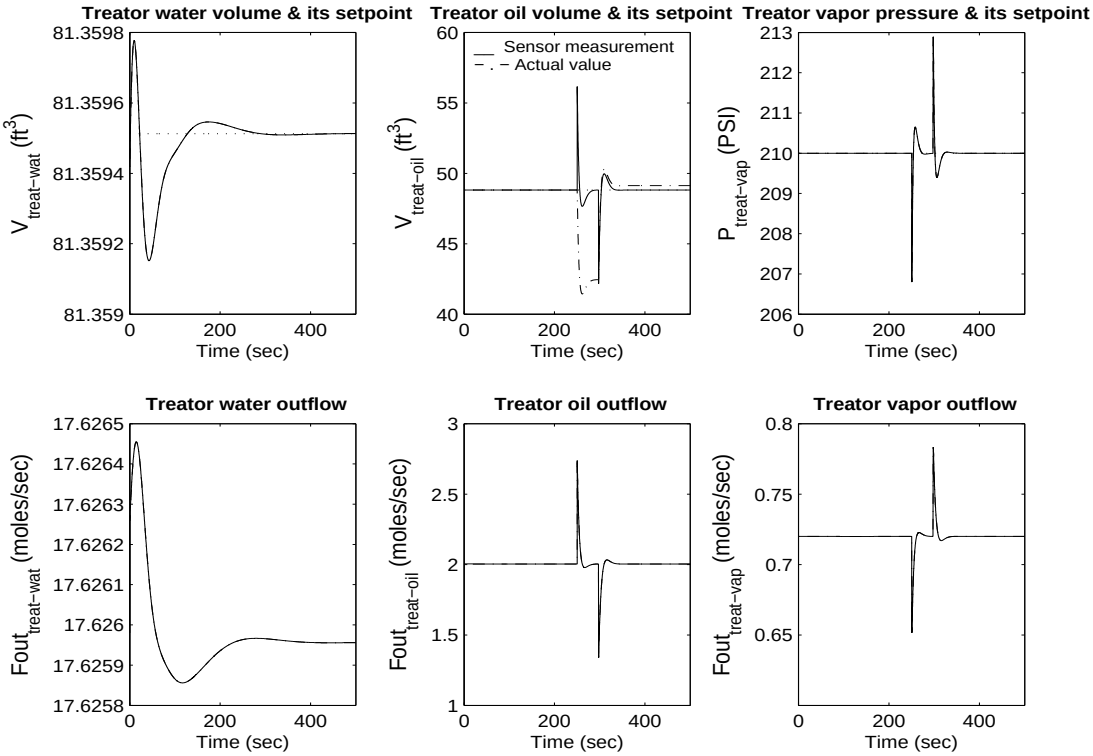

Figure 8.10: +15% bias S_4 applied at t=250 sec. - ΔSP=5%

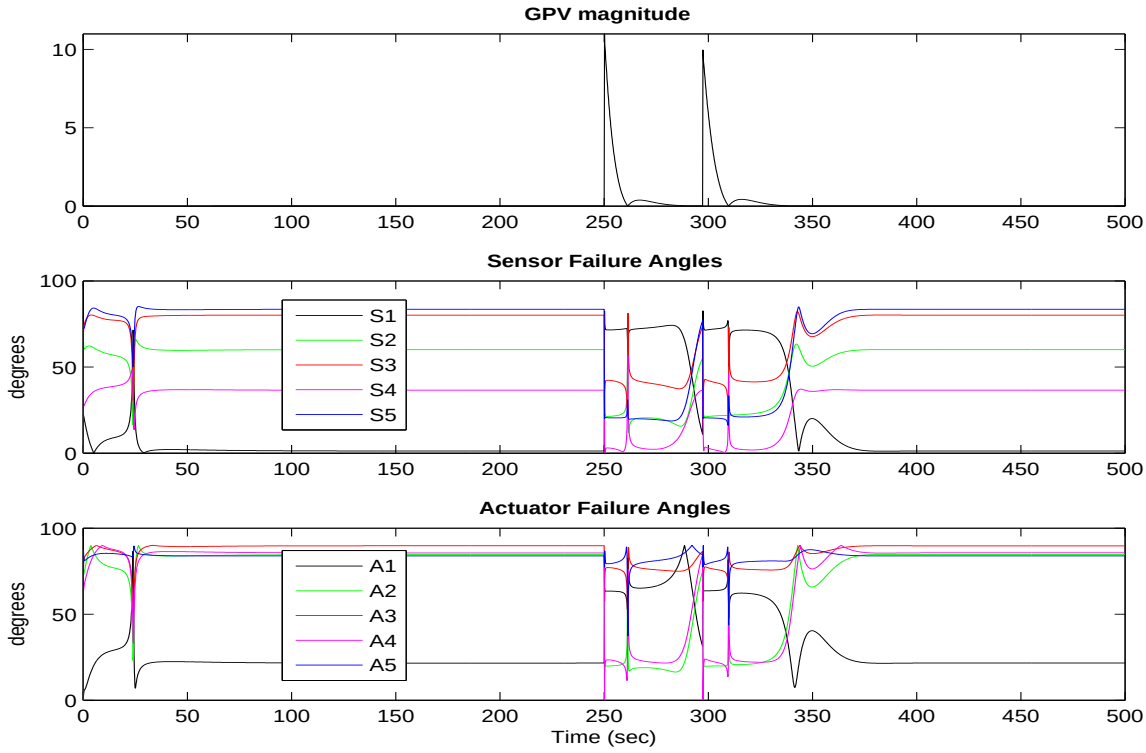

Figure 8.11: Static $|GPV|$ and $\angle GPV$, S_4 faulty

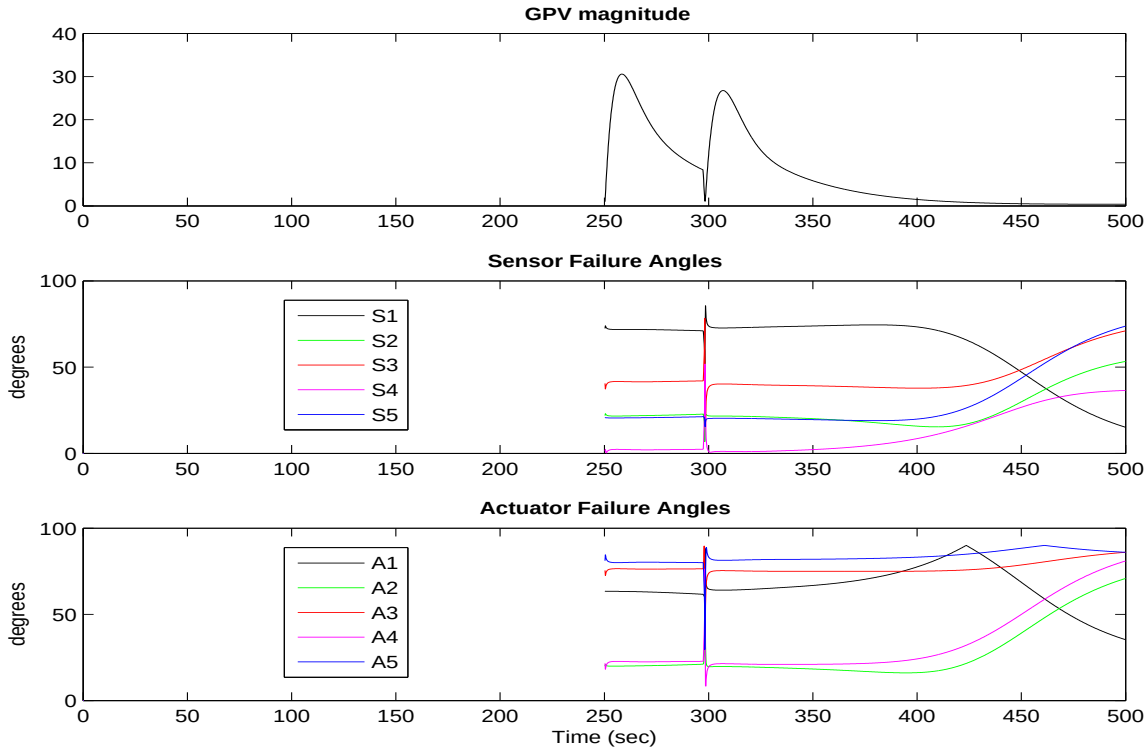

Figure 8.12: Dynamic $|GPV|$ and $\angle GPV$, S_4 faulty

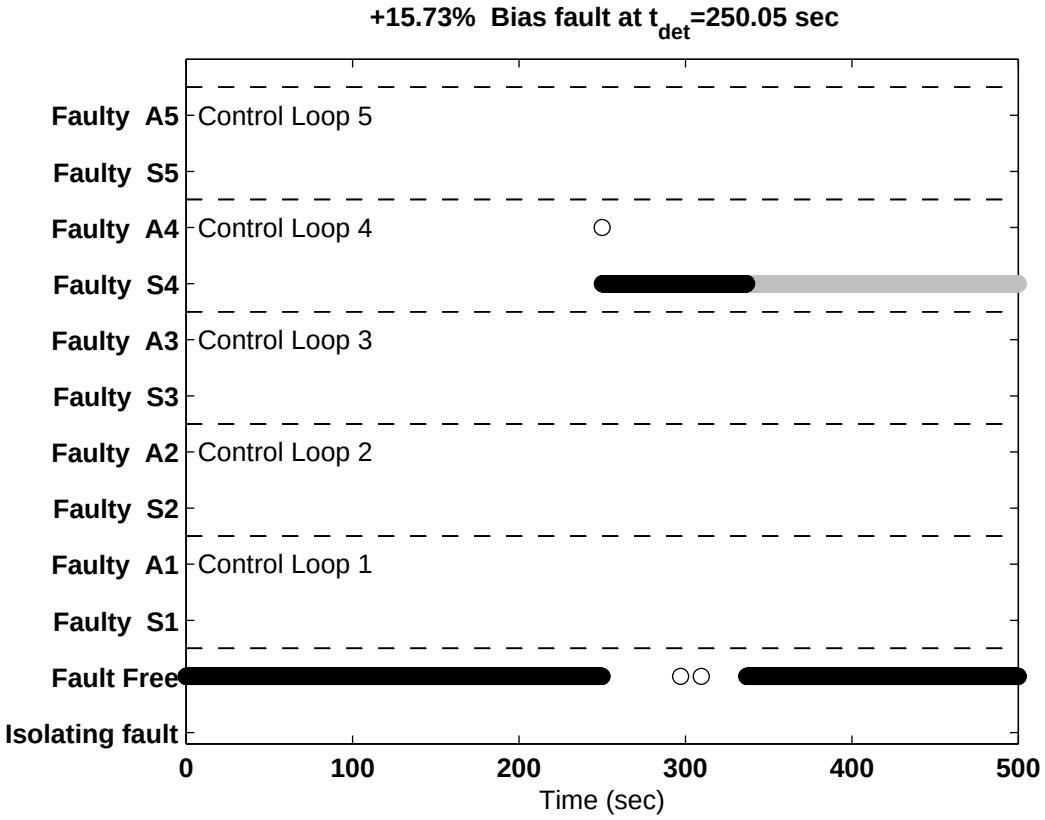

Figure 8.13: Operator display, S_4 faulty

The estimated fault size is 15.73%, which has an estimation error of 4.87%. This % of error is acceptable, taking into account that the SID % of fitting for output 4 was not very high (76.98%) and the fact that the fault was applied for Δ_{SP}=5% so the modeling errors became significant. These two facts made this FDIA scenario very challenging. Despite the 4.87% error in the fault size estimation, it is observed from Fig. 8.10 that the accommodation significantly reduced the measurement error in the treator oil volume sensor (S_4). After the accommodation started at t=300 seconds, there is a short transient while the controller compensates the corrected measurement. However, the controller rapidly drives the oil volume sensor measurement to its setpoint value. It is also seen in Fig. 8.10 that after the fault is accommodated at t=300 seconds, the oil volume sensor measurement (solid line) and its actual value (dash-dot line) become very similar, reducing significantly the error in the produced oil volume estimation. Thus method 1 for fault accommoda-

tion preserves closed-loop stability and also drives the actual volume level for the faulty sensor very close to its desired value. These results also show that the fault size estimation and accommodation methods proposed in sections 8.3 and 8.4.1 do not require unrealistic accuracy in the linear model to provide satisfactory results.

8.5.2 FDIA results using method 2

This section presents FDIA results for two case studies where the accommodation was performed using method 2, i.e., we modify the setpoint instead of correcting the faulty measurement. The first scenario illustrates a +10% bias treator vapor pressure sensor S_5 applied at t=60 seconds, followed by a -5%/min ramp separator liquid volume sensor S_1 fault applied at t=400 seconds, after the first fault was completely accommodated. All process variables are plotted in Figs. 8.14 and 8.15.

At t=60.15 seconds, the static $|GPV|$ in Fig. 8.16 increases significantly and exhibits a sharp magnitude change, thus a bias fault is rapidly detected. Since a bias fault is detected, the isolation is performed using the dynamic GPV. Fig. 8.17 shows that A_5 GPV angle is initially small but starts moving away from zero while, S_5 remains very small. As a result, the operator display in Fig. 8.18 shows the "Isolating fault" and "Faulty A_5" options for a few samples, while the GPV is swinging to reach its steady state direction. At t=125 sec. the time window condition defined in section 7.3.3 is satisfied, allowing the decision-making logic to provide the final isolation statement for sensor 5.

Figure 8.15 illustrates how sensor accommodation is performed using method 2, by manipulating the setpoint of the corresponding faulty loop. When the accommodation starts at t=125 seconds, the setpoint and the sensor measurement for output 5 change according to the estimated fault size. Although the sensor reading is showing that the pressure is 220 PSI, the actual value inside the treator is around 200 PSI

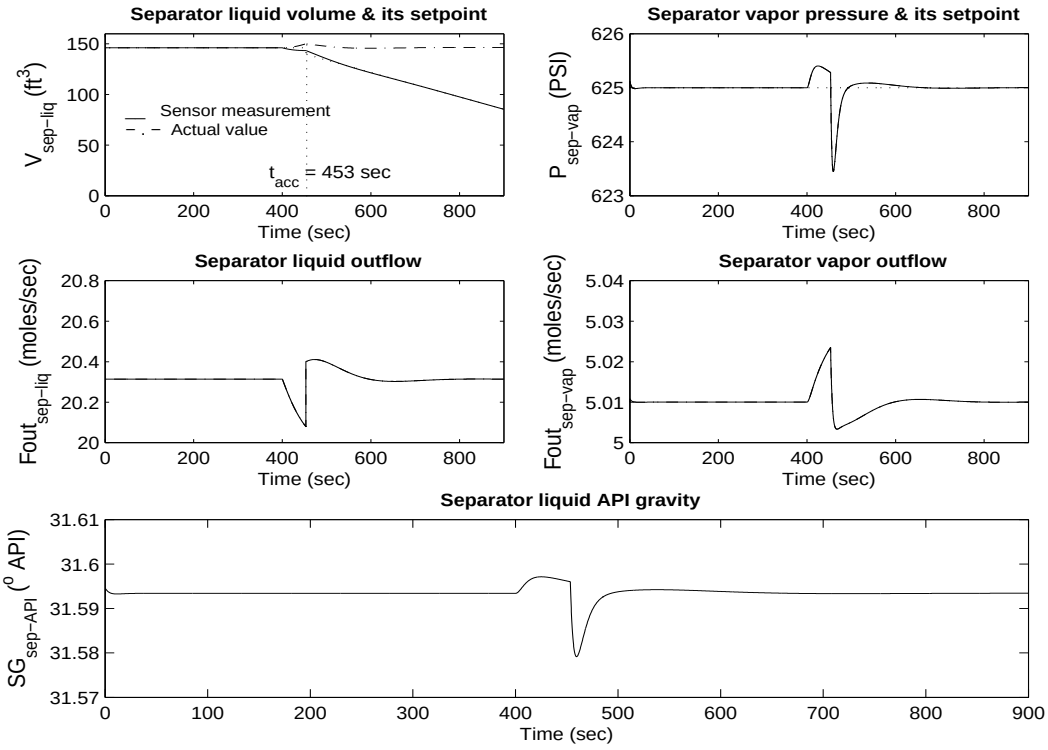

Figure 8.14: -5%/min ramp S_1 applied at t=400 sec. - Nominal Operating Point

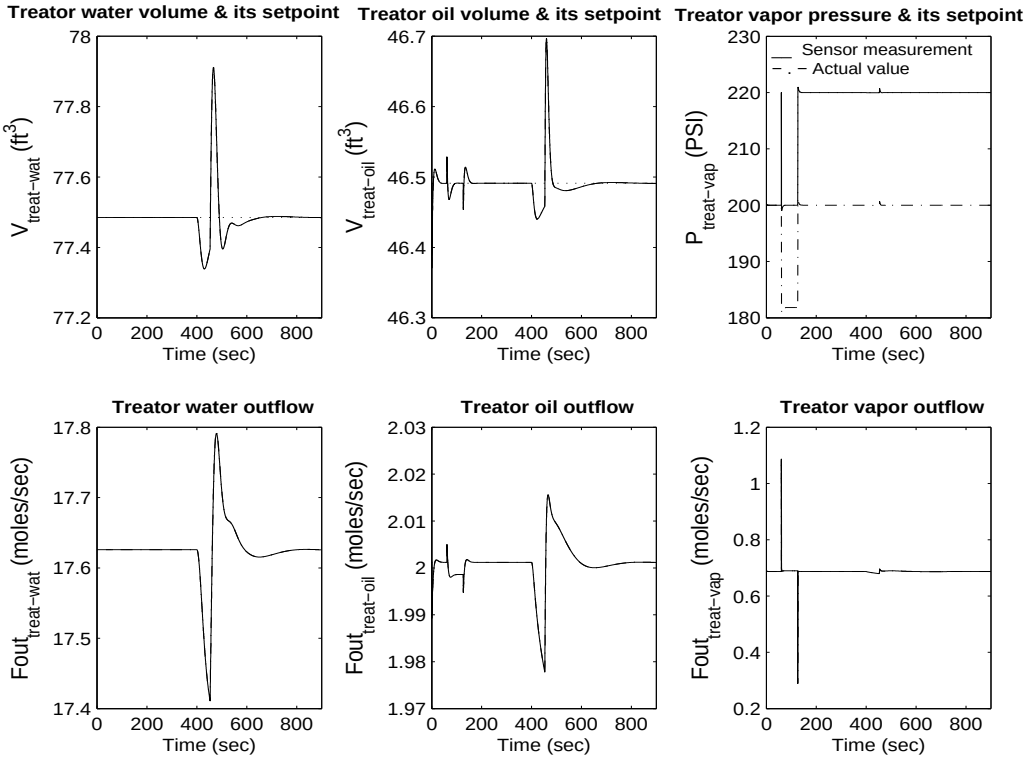

Figure 8.15: +10% bias S_5 applied at t=60 sec. - Nominal Operating Point

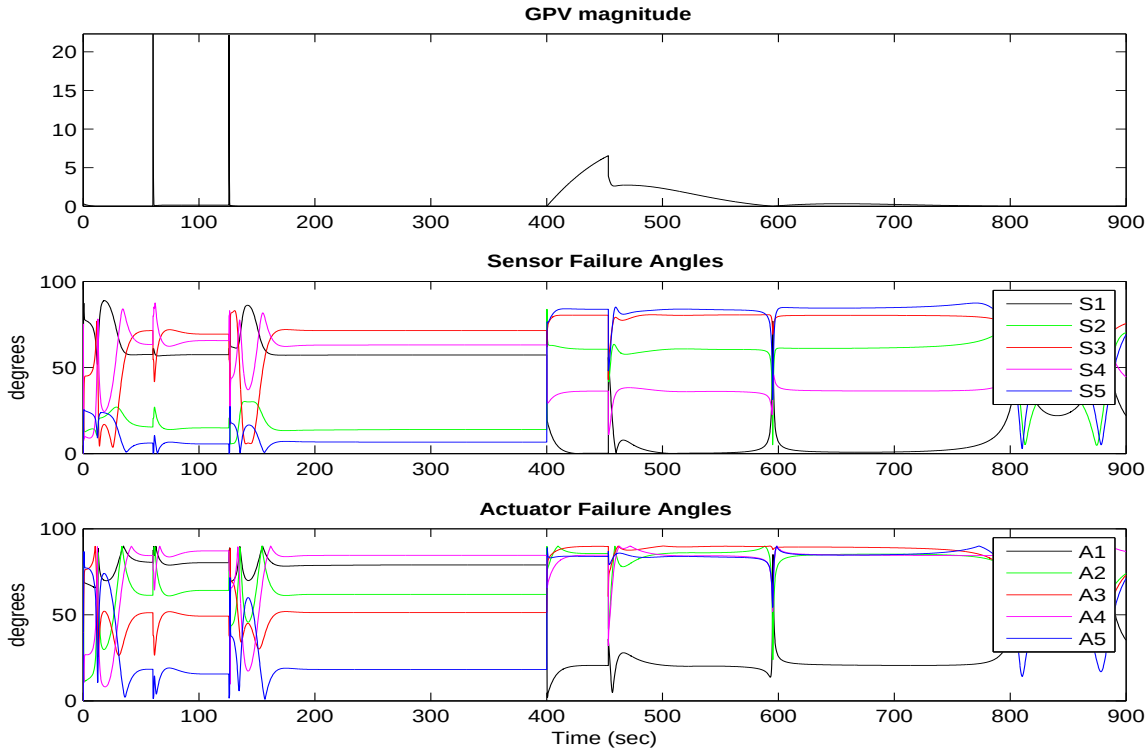

Figure 8.16: Static $|GPV|$ and $\angle GPV$, S_1 and S_5 faulty

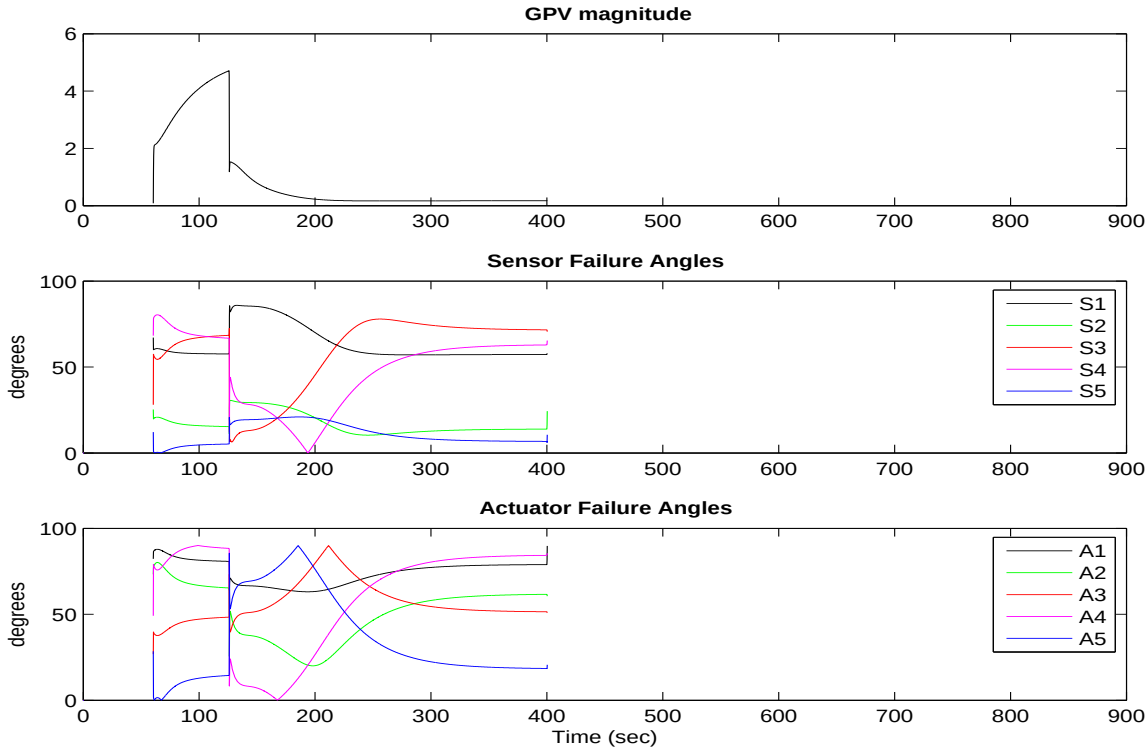

Figure 8.17: Dynamic $|GPV|$ and $\angle GPV$, S_1 and S_5 faulty

(due to the fault in sensor 5) which corresponds to the actual desired value. It is also observed that although the beginning of the accommodation slightly disturbs loop 4, it rapidly stabilizes within the next 30 sec. Since the accommodation successfully compensates the actual treator pressure value, the $|GPV|$ returns to its fault free value and the "Fault Free" and "Faulty S_5" options are displayed in the operator panel in Fig. 8.18.

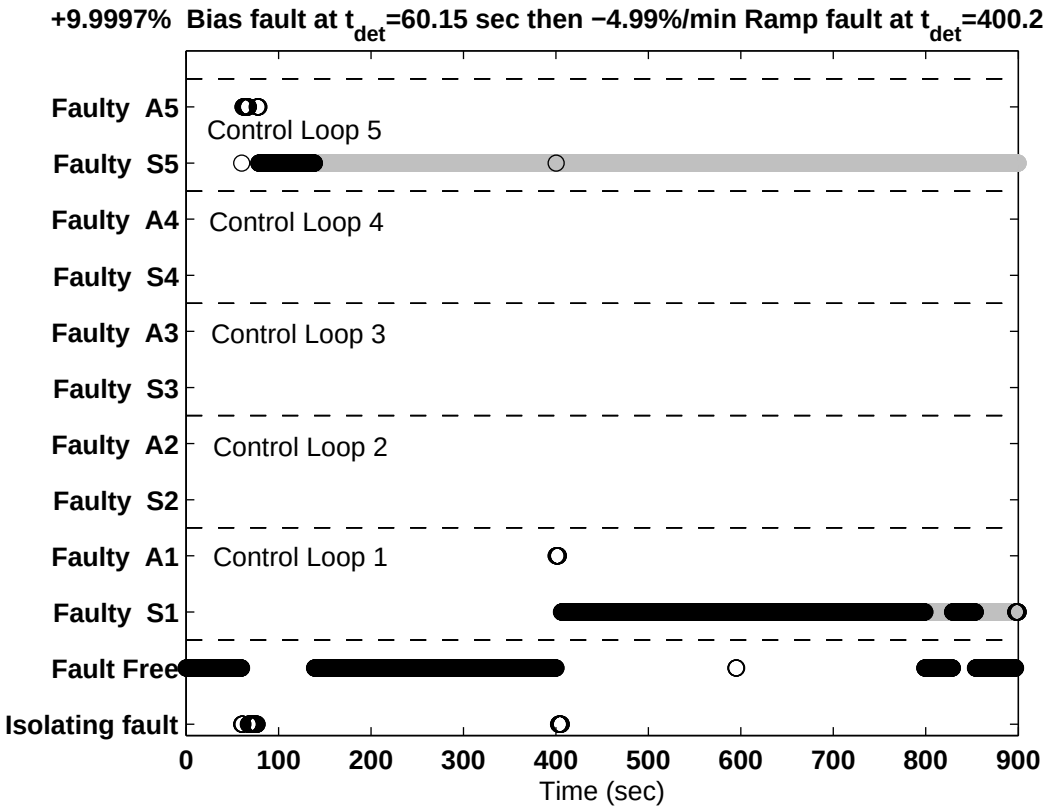

Figure 8.18: Operator display, S_1 and S_5 faulty

After S_5 is completely accommodated, the $|GPV|$ increases significantly again, detecting a second fault at t=400.2 sec. For this case the $|GPV|_{slope}$ change at t_{fault} is not very large, so the fault is classified as ramp and the isolation is made based on the static GPV. It is observed in Fig. 8.16 that S_1 moves speedily towards zero, providing a clear isolation in the operator display in Fig. 8.18, 6 sec. after the fault happens. There are a few samples showing the "Isolating fault" and "Faulty

A_1" options while the GPV reaches its steady state direction. At t=453 seconds, the decision-maker logic makes a final isolation statement for S_1 and the accommodation starts.

The successful accommodation is clearly shown in Fig. 8.14, where the setpoint and the sensor measurement in loop 1 decrease due to the accommodation but the actual separator liquid volume returns to its desired value. Although the beginning of the accommodation process slightly perturbs the other loops, they stabilize rapidly and the complete system is soon back to the fault free operation conditions. At the end of the simulation the operator panel displays the "Fault Free", "Faulty S_1" and "Faulty "S_5" options, which means that sensors 1 and 5 are both successfully accommodated, but they are still faulty and maintenance should be scheduled. The estimated fault sizes are 9.9997% and -4.99%/min giving an estimation error of 0.003% and 0.2% respectively. These small % of error confirms that even for cases where the % of SID fitting is not very high, as it is the case for sensor 5, (84.44%), the F_{size} estimation method using the $|GPV|$ signature provides excellent results.

The aim of this third final case study is to evaluate the FDIA performance using accommodation method 2 with respect to modeling errors. In this scenario a -5%/min ramp separator vapor pressure sensor (S_2) fault is applied at t=150 sec. for a setpoint variation, Δ_{SP}=5%. In this case, all process variables are settled to steady state at t=0 seconds; they are depicted in Figs. 8.19 and 8.20.

Figure 8.21 shows that the $|GPV|$ increases significantly at t=150.15 seconds, giving a fast detection time of 0.15 sec. Also, the fault is classified as a ramp given the small $|GPV|_{slope}$ change at t_{fault}, so the static GPV is used for isolation. During the first 16 seconds, actuator 2 is declared faulty in the operator panel while the GPV is swinging to reach its steady state direction. This isolation time is acceptable, given that sensor and actuator 2 belong to the special case for sensor-actuator FDI, as

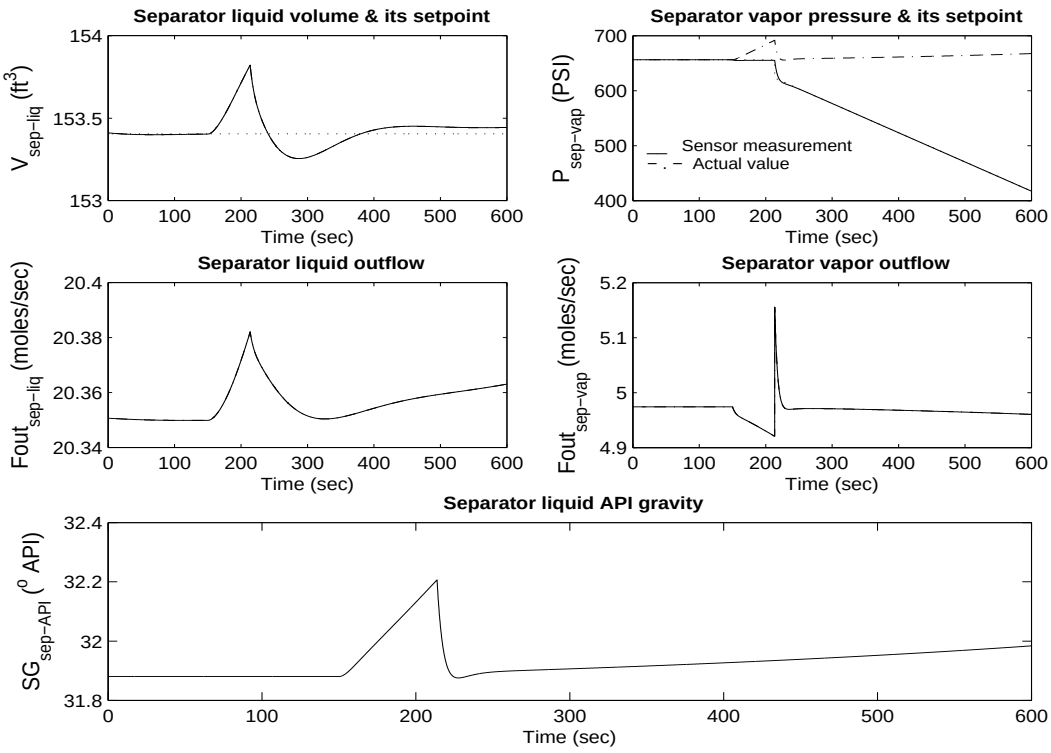

Figure 8.19: -5%/min ramp S_2 applied at t=150 sec. - ΔSP=5%

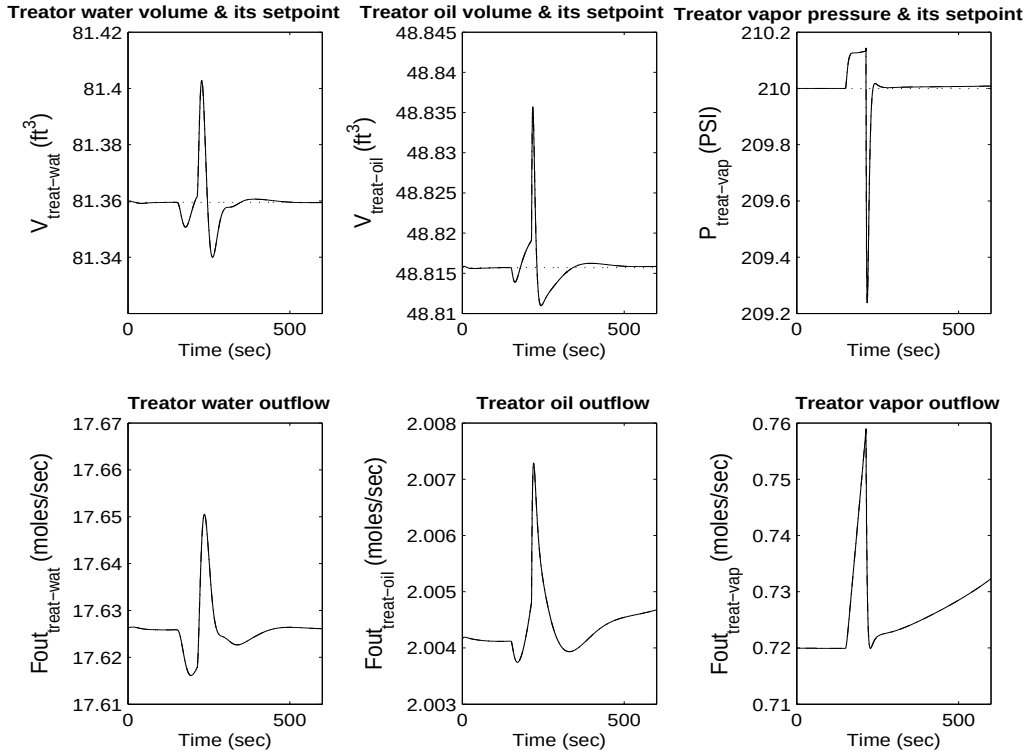

Figure 8.20: -5%/min ramp S_2 applied at t=150 sec. - ΔSP=5%

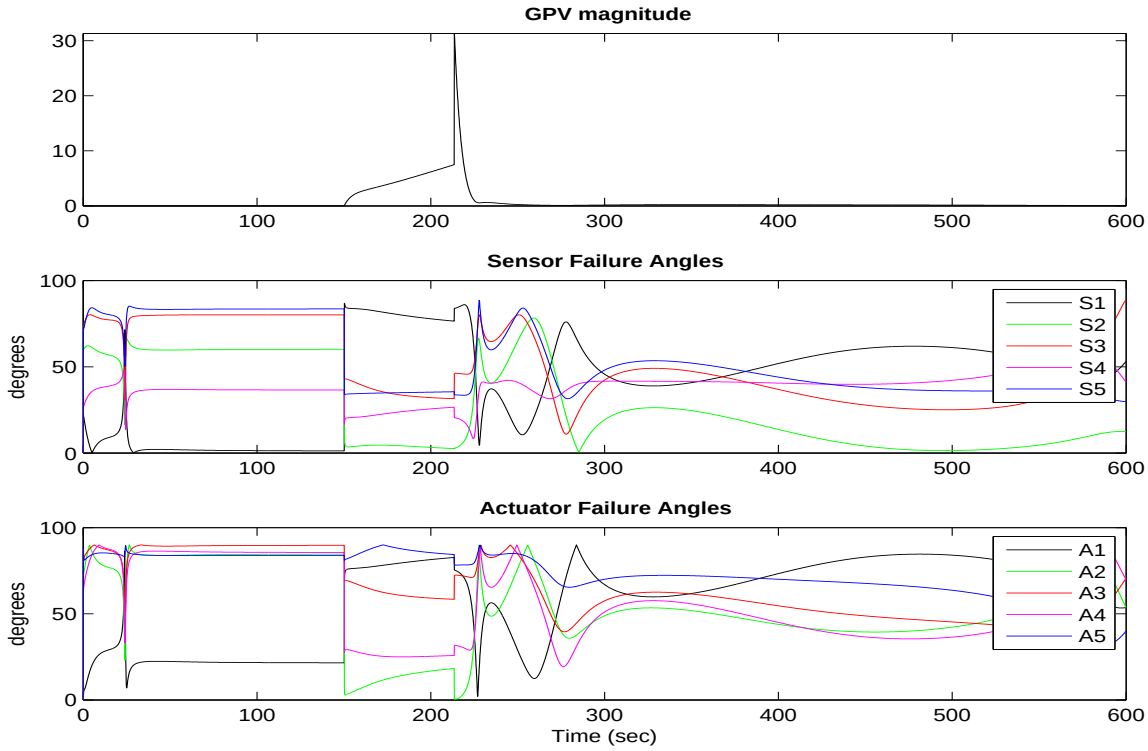

Figure 8.21: Static $|GPV|$ and $\angle GPV$, S_2 faulty

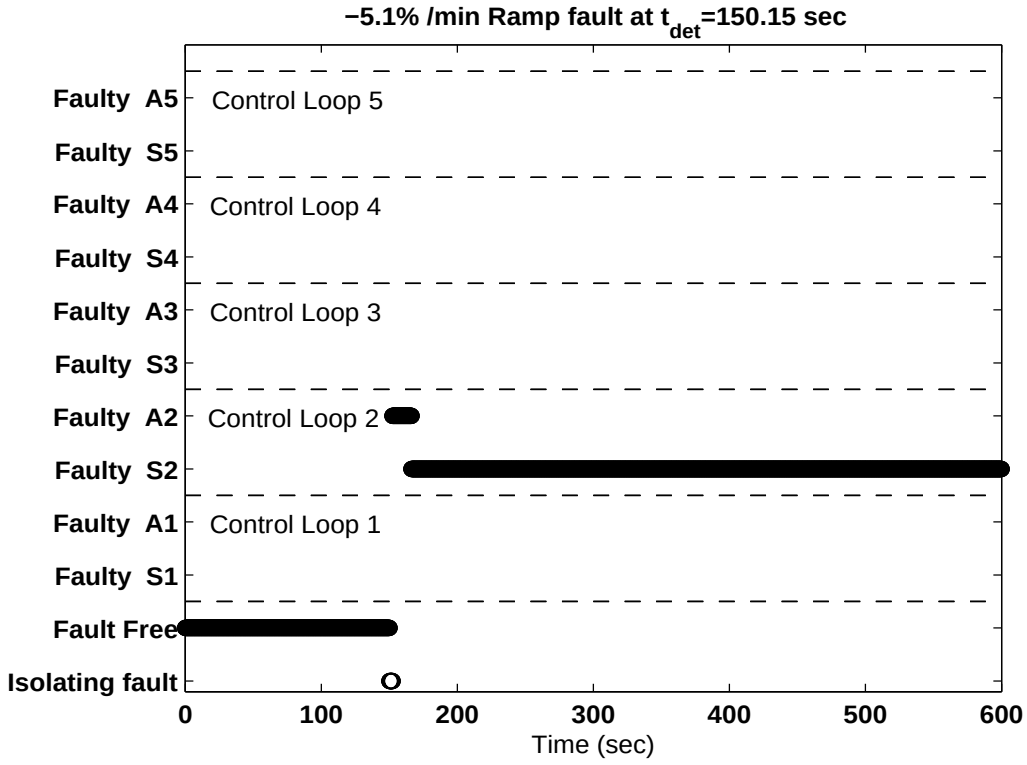

Figure 8.22: Operator display, S_2 faulty

discussed in section 6.2. After t=166.4 seconds, sensor 2 is correctly isolated in Fig. 8.22 and a final isolation decision is reached by the decision-making logic at t=214 sec.

The resulting accommodation is shown in Fig. 8.19 where the setpoint and sensor measurement for output 2 decrease with time, according to the calculated ramp F_{size}. The estimated fault size is -5.1%, which gives an isolation error of 2.03%. This % of error is very satisfactory, given that the fault was applied for a 5% setpoint variation. However, it is observed from Figs. 8.19 and 8.20 that although the beginning of the accommodation slightly disrupted the other loops, they start to stabilize as the time progresses and the overall performance of the system is significantly improved.

From the case studies presented in this section it is established that the proposed FDIA strategy provides fast detection and isolation times even when the SID % of fitting is not very high and the fault scenario is applied at a different operating point. Also, the fault size estimation and accommodation methods proved to perform very well on faults that are quickly rejected and using an identified model with a reasonable % of fitting.

8.5.3 FDIA results during transients

The simulation results presented in this section are intended to evaluate the FDIA performance for an scenario where a fault is applied during a process transient, right after a small setpoint variation. The case study is described as follows: +45% bias separator liquid volume sensor (S_1) fault happens at t=65 seconds, after a setpoint change (Δ_{SP}) of 3% is applied at t=60 sec. in all the loops. It is observed in Fig. 8.25 that although the $|GPV|$ abruptly increases at t=60.3 sec. due to the setpoint variation, it starts to decrease immediately. The $|GPV|$ change right after the fault happens at t=65.25 seconds, is substantially larger. Thus, it is possible to

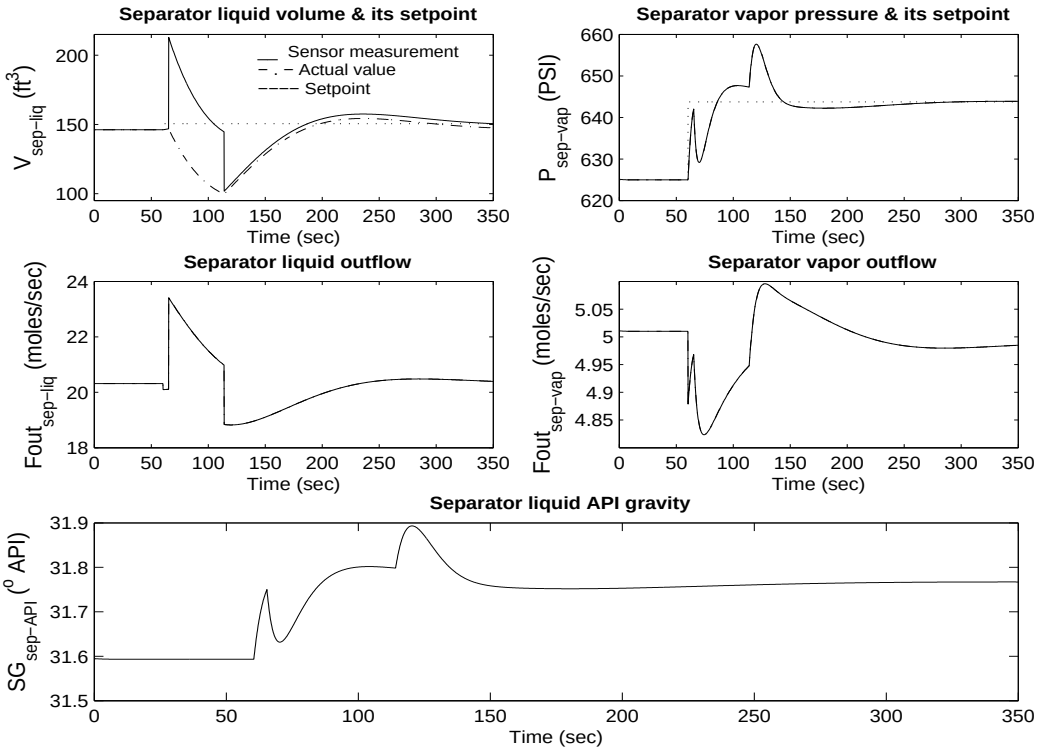

Figure 8.23: +45% Bias S_1 applied at t=65 sec. - ΔSP=3% at t=60 sec.

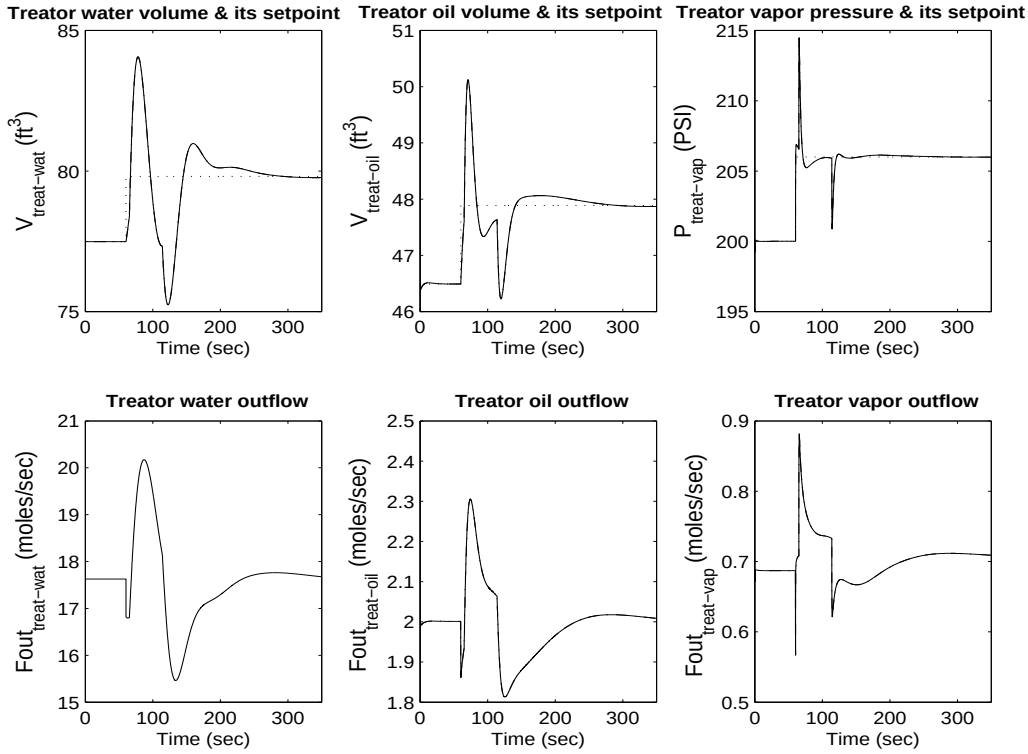

Figure 8.24: +45% Bias S_1 applied at t=65 sec. - ΔSP=3% at t=60 sec.

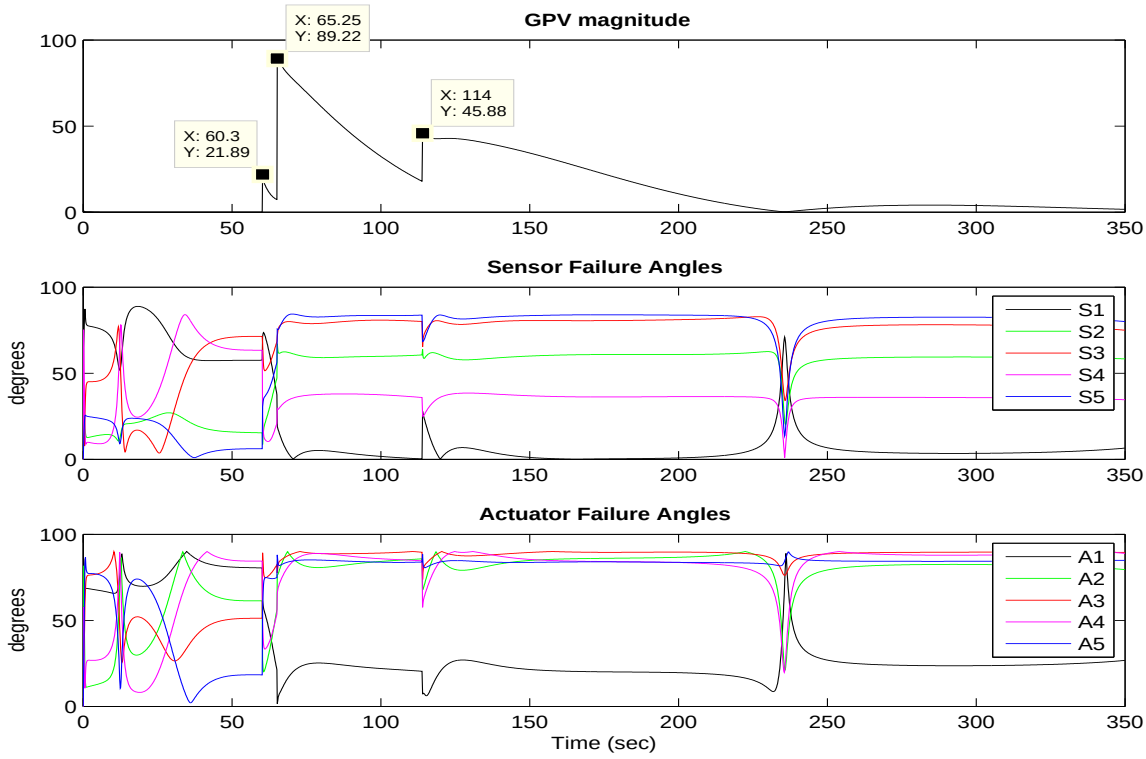

Figure 8.25: Static $|GPV|$ and $\angle GPV$, S_1 faulty

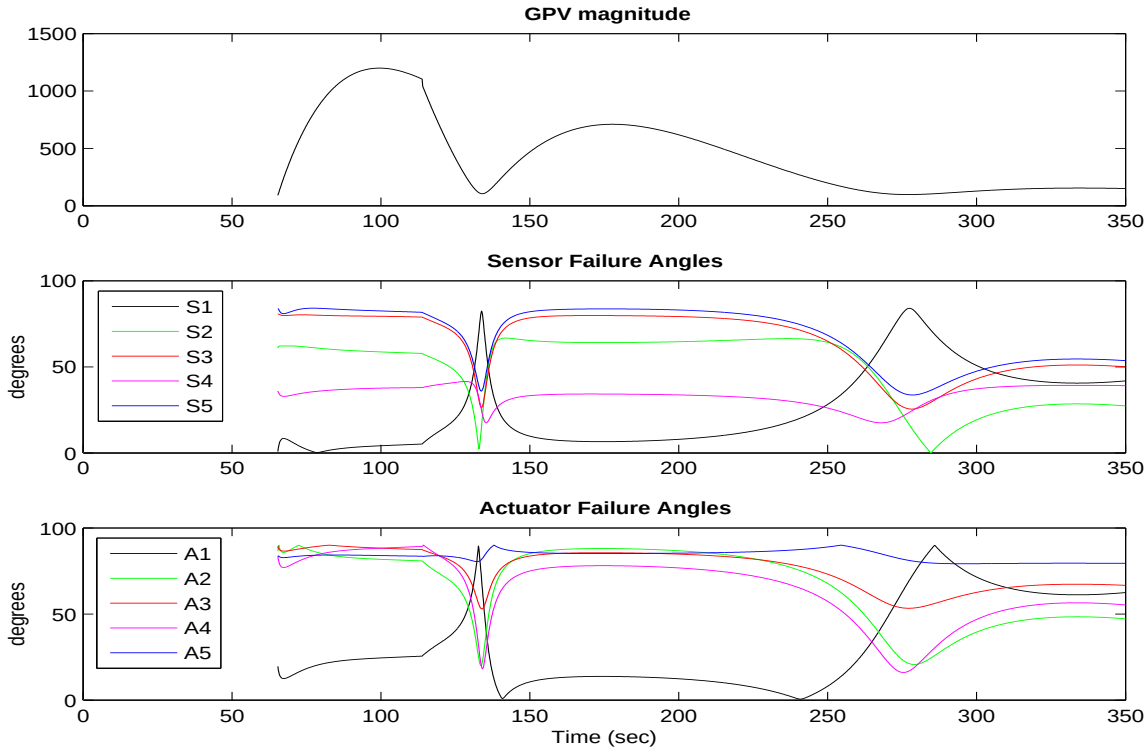

Figure 8.26: Dynamic $|GPV|$ and $\angle GPV$, S_1 faulty

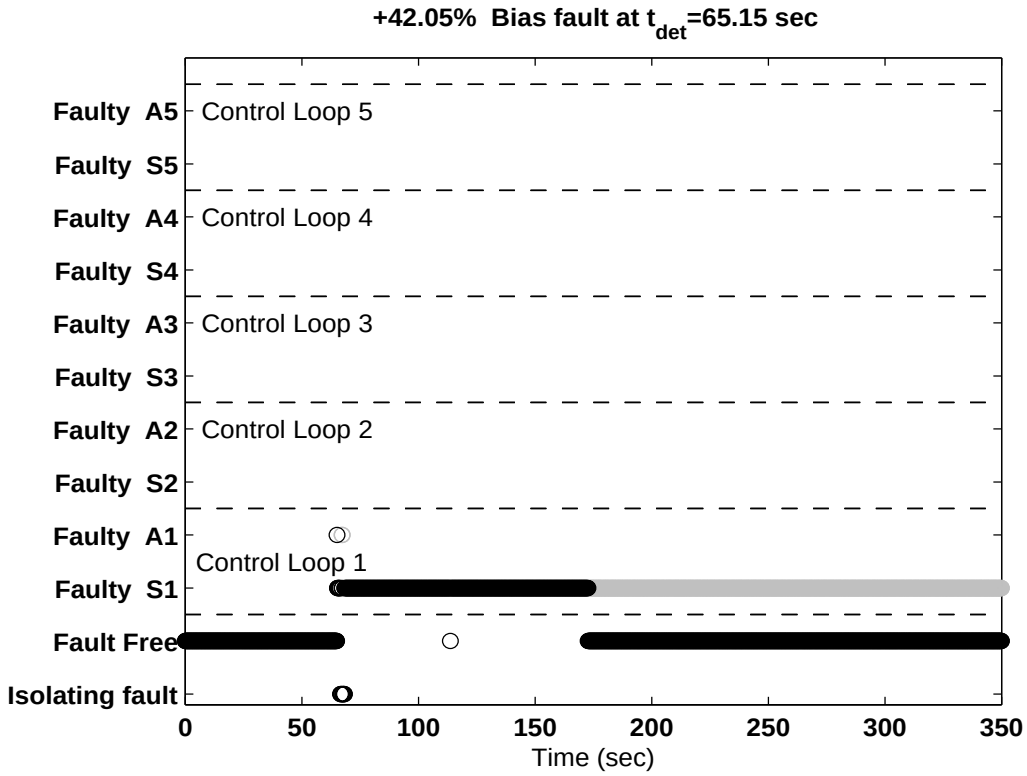

Figure 8.27: Operator display, S_1 faulty

avoid a false fault alarm and still be able to detect the fault during the transient, by properly increasing the value of the $|GPV|_{ff}$ multiplier n through this period, using the magnitude threshold calculation proposed in section 7.3.1. By adapting the magnitude threshold at the known times when setpoint changes are issued the fault is correctly detected and classified as bias in Fig. 8.27, just 0.15 sec. after it happens. As a result, the isolation is made based on the dynamic GPV angles in Fig. 8.26. In this figure the minimum GPV angle corresponds to S_1, giving the correct "Faulty S_1" option in Fig. 8.27.

At t=114 sec. the time window conditions for isolation introduced in section 7.3.3 are satisfied, thus the final isolation statement for S_1 is made and the accommodation starts. It is seen in Figs. 8.23 to 8.26 that the other loops and the $|GPV|$ are slightly perturbed by the beginning of the accommodation. However, all the outputs move back to their setpoint values rapidly, showing a significant improvement in the

system's performance after the accommodation started. For this case study the fault accommodation is performed using method 1. The estimated fault size is 42.05% which has an estimation error of approximately 6.55%. This estimation error is satisfactory, taking into account that the fault was applied during the transient of a 3% setpoint variation, which makes the modeling errors more significant. At the end of the simulation the operator panel displays the "Fault Free" and "Faulty S_1" options, to acknowledge that sensor 1 has been successfully accommodated (according to the adapted magnitude threshold) but sensor 1 is still faulty and maintenance should be scheduled.

It should be noticed that this case study is intended to evaluate the GPV technique performance during transients due to small setpoint variations taking place after the system has already reached steady state. If the setpoint variations are larger than $\pm 10\%$ (approximately), the linearized model is not longer valid and will have to be identified again. However, when the ICAM system supervisor orders the SID agent to identify a new model, the system must be in steady state in order to start the identification process [34]. Once the new model has been identified, the on-line threshold computation block will have to be executed again to estimate the magnitude threshold (M_T) at the new operating point, which also requires the system to be in steady state as described previously in section 7.3.1. Therefore, the only scenarios that can be considered for FDIA during transients for the GPV technique are those where the setpoint variations are inside the range of validity of the previously identified linear model. This range is defined based upon the qualitative degree of nonlinearities in the system, which is application dependent. Thus, for this type of scenario, the simulations results using the GPV technique have shown satisfactory FDIA performance during transients if the magnitude threshold is properly adapted and the modeling errors are not very large.

Chapter 9

FDIA for Noisy Scenarios

9.1 Introduction

Ideally, an FDIA technique should provide fast detection time with low false alarm and misdetection rates while being robust to disturbances, modeling errors and noise. In section 3.5 disturbance decoupling was proposed for the GPV technique and its robustness was successfully tested using the JCSTR model in appendix B. Robustness with respect to modeling errors was shown in chapters 5 and 8, where excellent FDIA results were obtained using an identified model with a reasonable percentage of fitting, even when the model is used at operating points different from the nominal point used for model identification.

The simulation results presented in the previous chapters were obtained using noise-free sensor measurements, based on the assumption that the Data Reconciliation Agent was capable of reducing the noise to a negligible level. However, if the sensor measurements are quite noisy, it is possible that the signals may contain a significant amount of noise even after been preprocessed by the Data Reconciliation Agent. In this chapter the FDIA robustness with respect to noise is evaluated and modifications are proposed to improve the tradeoff between detection time and sensitivity to high

frequency influences.

9.2 Modifications to the FDIA methodolody

The diagnostic system should respond quickly in detecting and isolating faults. However, quick failure diagnosis and robustness to noise are two conflicting goals. An FDIA algorithm that is designed to detect a fault quickly, becomes more susceptible to high frequency influences. This makes the system sensitive to noise and can lead to frequent false alarms during normal operation, which is certainly undesirable. Thus, robustness precludes deterministic isolability tests where the thresholds are placed close to zero. In the presence of noise, these thresholds may have to be chosen conservatively. Thus, robustness needs are to be balanced with those of performance.

In order to improve the robustness with respect to noise, some modifications are proposed to the FDIA agent presented in chapter 4 and the decision-maker logic in section 7.3. First, the plant's input and output measurements are collected for a moving time window of 10 samples to average out and reduce the effect of noise; this process approximately mimics the reconciliation process.

Next, Fig. 7.3 is modified as highlighted in the fault classification block in Fig. 9.1 by hatching. This change means that the dynamic GPV will be the only one used for isolation, for both cases, bias and ramp faults. For systems with negligible noise levels as considered in section 7.3.3, we took advantage of the faster behaviour of the static GPV to reduce the isolation time of slowly varying (ramp) faults. However, for scenarios where measurement noise is significant, the dynamic GPV with a proper choice of σ, as presented in section 3.6, becomes very powerful as it acts as a filter during the residual computation. This results in a smoother behaviour of the parity vector, and therefore a clearer and cleaner fault isolation. This is illustrated and

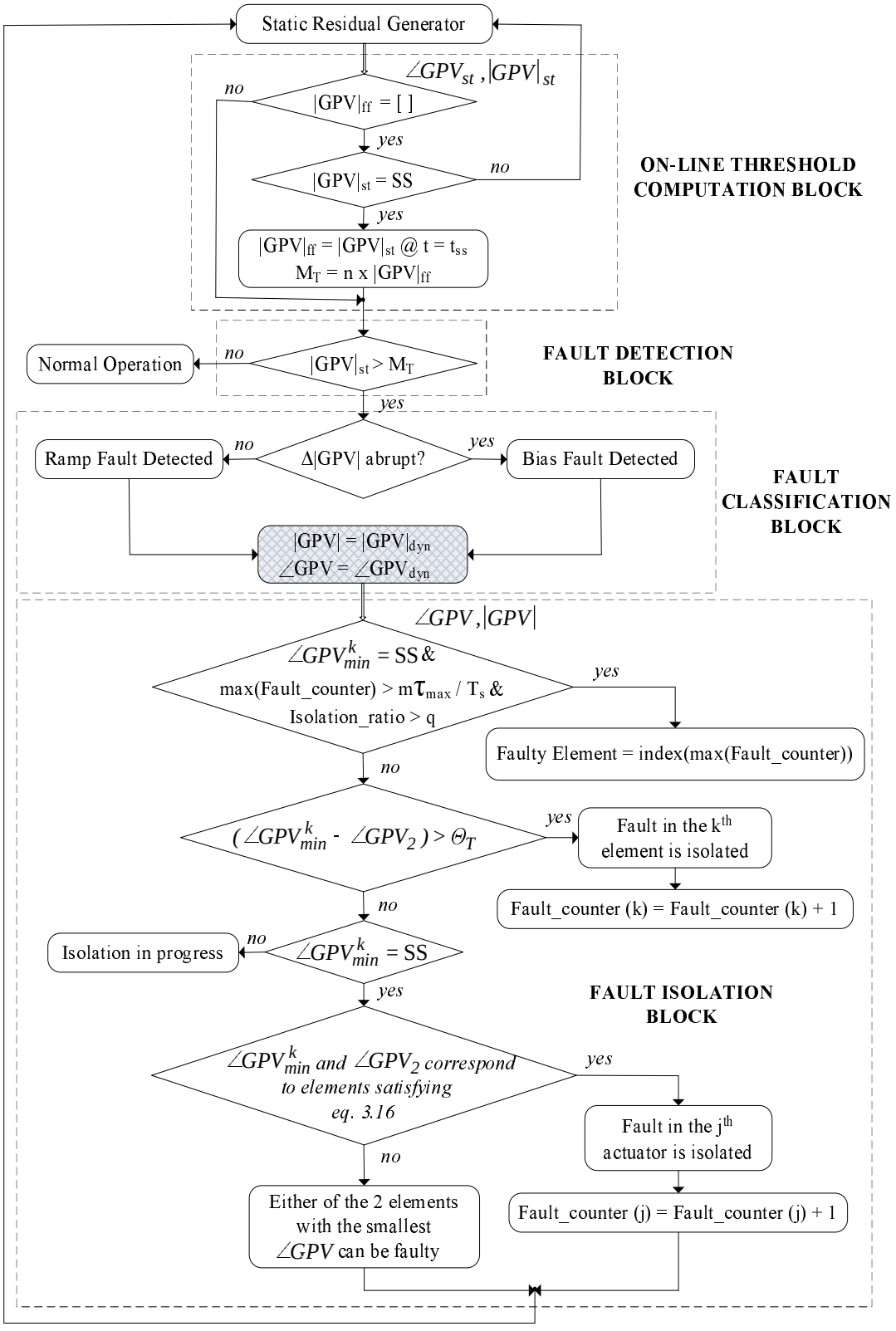

Figure 9.1: Decision-maker block diagram for noisy scenarios

discussed in more detail in section 9.3.

Also, the value of the constant n in the on-line threshold computation block in Fig. 9.1 was increased from 2 (noise-free case) to 4 (noisy case), to avoid false alarms due to the effect of noise. Although these modifications increase the detection and isolation times, as it will be observed in the following simulations results, they provide a good tradeoff between FDIA performance and robustness with respect to noise. The other aspects of the FDIA logic presented for the noise-free conditions in previous chapters remain unmodified and are valid for the noisy case.

9.3 FDIA Simulation results for noisy scenarios

The following FDIA results for the separator model were obtained applying the modifications described in section 9.2. For each scenario the sensor measurements were contaminated by noise with a Gaussian amplitude distribution whose standard deviation is specified as a percentage of the actual signal value. The noisy measurements are preprocessed by a procedure that mimics data reconciliation to reduce the level of noise [1].

The first scenario corresponds to a -20% bias fault applied to the separator liquid volume sensor (S_1) at t=150 sec. For this case all the sensor measurements are corrupted by gaussian noise with a standard deviation of 5% their actual values. Figures 9.2 and 9.3 illustrate the effect of the fault in the separator and treator variables.

In Fig. 9.4 it is seen that the static GPV preserves its signature even for noisy conditions, allowing fault size estimation and classification using the logic proposed in section 7.3. However, once the fault is detected, it is observed from Fig. 9.5 that

[1] It should be noticed that the data reconciliation agent was not fully developed at this point, thus these results do not illustrate its complete capabilities.

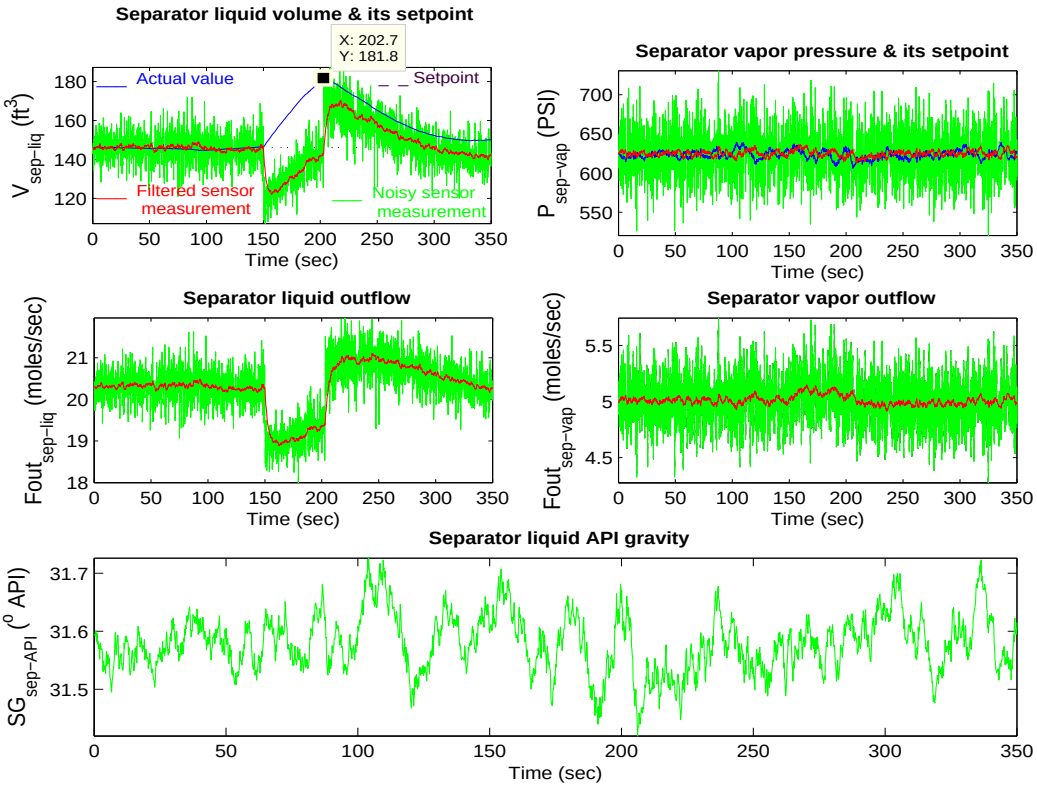

Figure 9.2: -20% bias S_1 applied at t=150 sec. - Nominal Operating Point

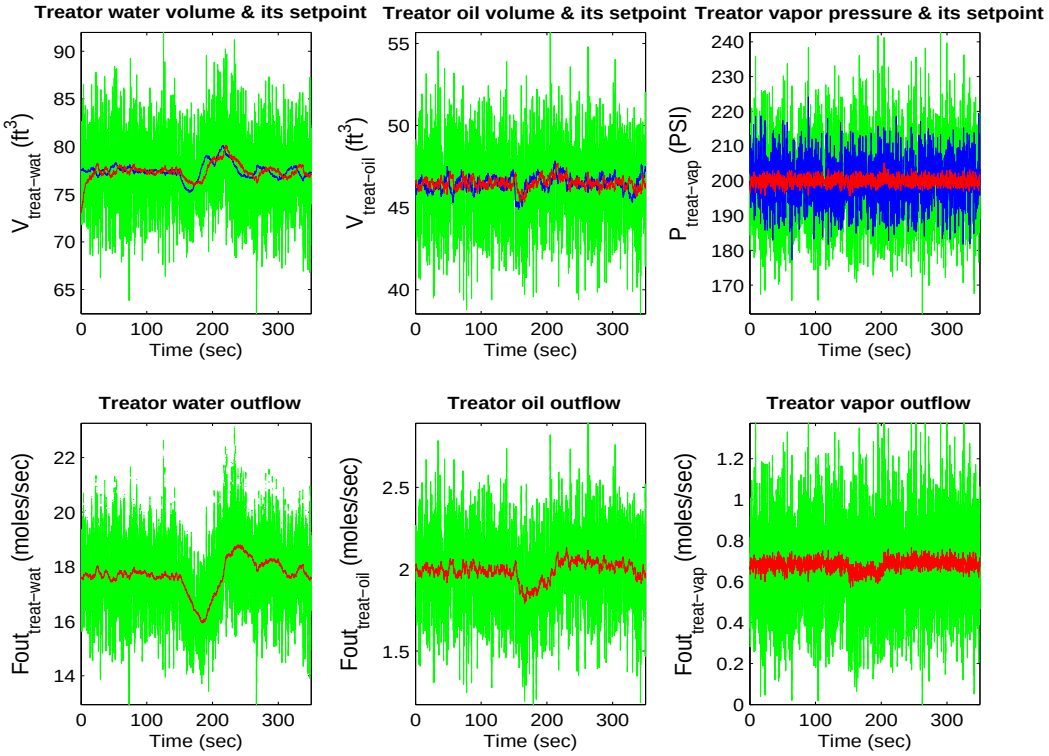

Figure 9.3: -20% bias S_1 applied at t=150 sec. - Nominal Operating Point

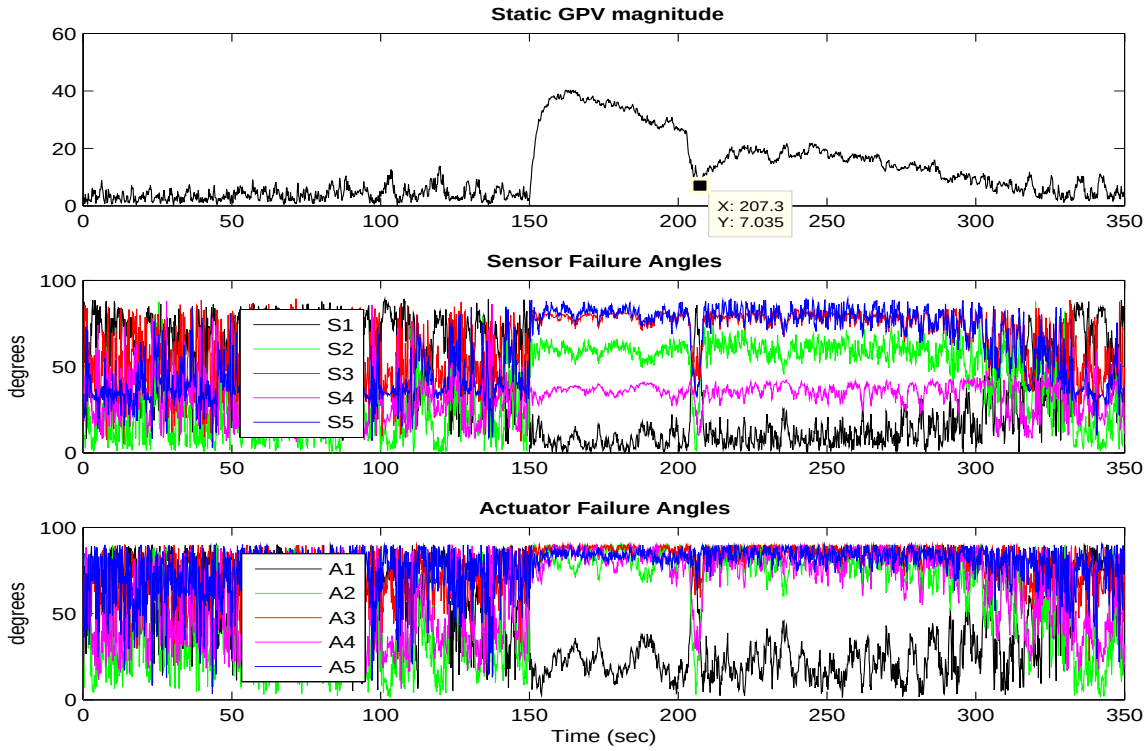

Figure 9.4: Static $|GPV|$ and $\angle GPV$, S_1 faulty

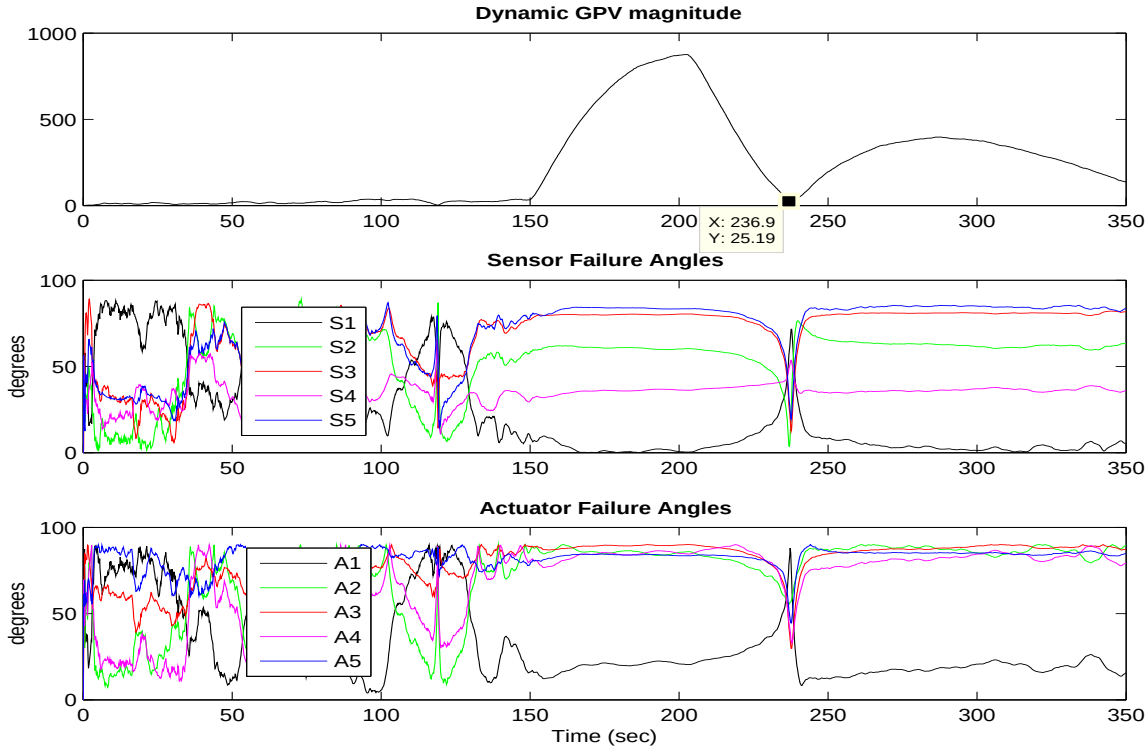

Figure 9.5: Dynamic $|GPV|$ and $\angle GPV$, S_1 faulty

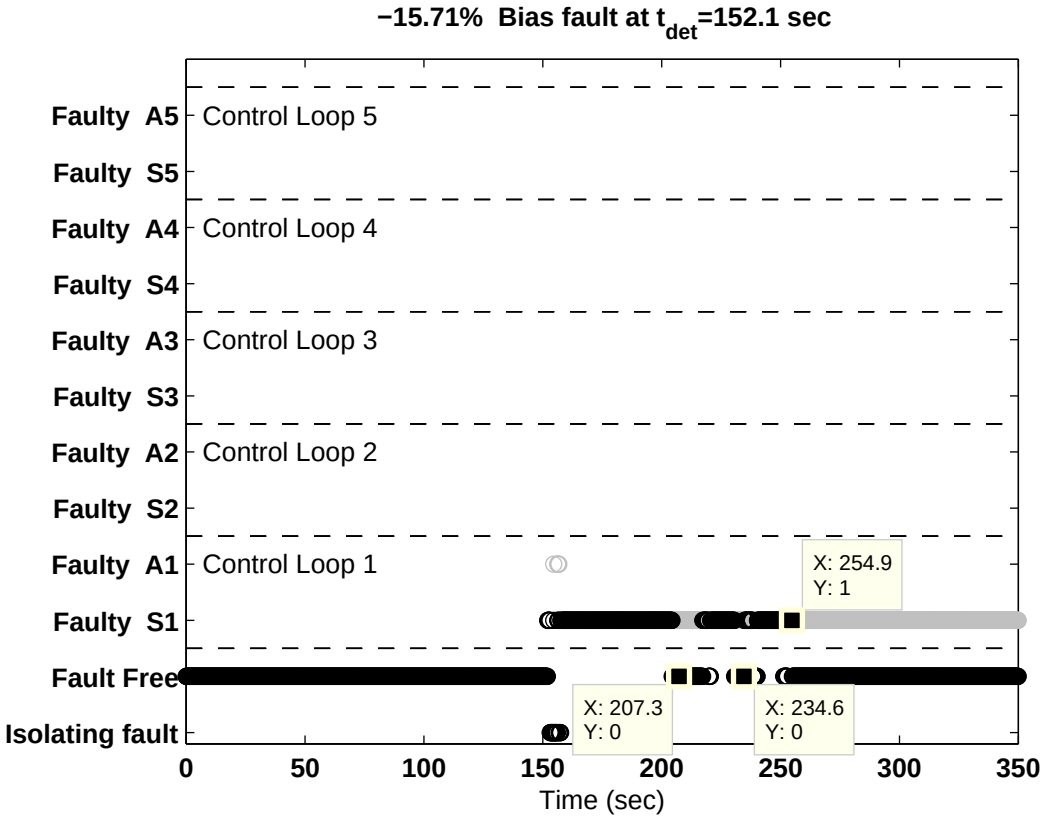

Figure 9.6: Operator display, S_1 faulty

the filter properties of the dynamic GPV provides a smoother set of angles for fault isolation, clearly showing S_1 as the angle closest to zero.

Figure 9.6 shows that the fault was detected 2.1 sec. after it was applied, which is a very fast response for such a noisy scenario. The calculated fault size is -15.71% which has an estimation error of 21.4%. Although the fault size estimation accuracy has decreased due to the large amount of noise affecting the measurements, it is clear from Fig. 9.2 that the actual value of the liquid inside the tank and the corresponding sensor measurement became very similar after the accommodation started, at t=202.7 sec. Also, the other process variables returned close to their pre-fault values, which is reflected in a rapidly reduction of the $|GPV|$. Since the threshold selection for the detection block has been more conservative in order to achieve a good tradeoff between false alarms and misdetection rates, the decision-

maker logic declares complete accommodation with a larger value of $|GPV|$ than the one required for the noise free case. This is illustrated in the operator panel in Fig. 9.6 which shows that total accommodation has been achieved, approximately 104.9 sec. after the fault, by displaying the "Faulty S1" and "Fault Free" options during the intervening time. It should be noticed in Fig. 9.4 that around 207.3 sec. the $|GPV|_{st}$ becomes very small due to the transient after the accommodation started. A few values around this minimum are smaller than the detection threshold resulting on a few "Fault Free" options in Fig. 9.6 around 207.3 sec. and 234.6 sec. However, the isolation is not affected by the minimum on the $|GPV|_{dyn}$ around t=236.9 sec. because once the final isolation statement is made (at t=202.7 sec. when the accommodation started), the corresponding fault is continuously displayed until the faulty element is fixed and the GPV angles are not longer used.

The second scenario illustrates the case when a -10% ramp fault is applied to the separator vapor pressure sensor (S_2) at t=150 sec. while the measurement readings are affected by gaussian noise with a standard deviation of 1% their actual values. Figure 9.7 shows a significant increase in the actual pressure (blue line) while the corresponding sensor measurement is displaying an incorrect value close to the set-point. In Fig. 9.9 it is observed that the $|GPV|$ rapidly increases after the fault was applied at t=150 sec. However, due to the conservative threshold choice made to minimize false alarms and misdetection rates for noisy conditions, the fault is not detected until t=160.95 seconds, as stated in Fig. 9.11. Considering the slowly varying nature of this fault and the significant amount of measurement noise, a detection time of 10.95 sec. is fairly satisfactory. The static and dynamic $|GPV|$ also show a different signature than the one exhibited for the bias case presented previously, which allows the classification of this fault as ramp type.

It is observed in Fig. 9.10 that despite the noisy behaviour in the system, the filtering nature of the dynamic GPV provides a smoother set of angles than the one provided

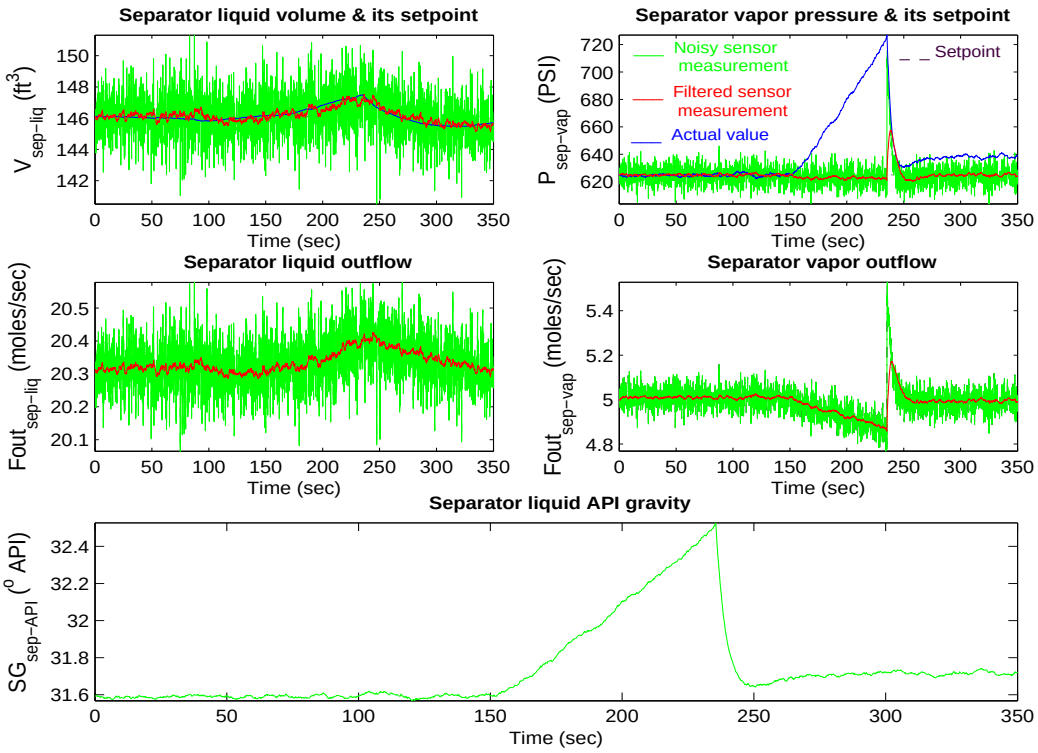

Figure 9.7: -10% ramp S_2 applied at t=150 sec. - Nominal Operating Point

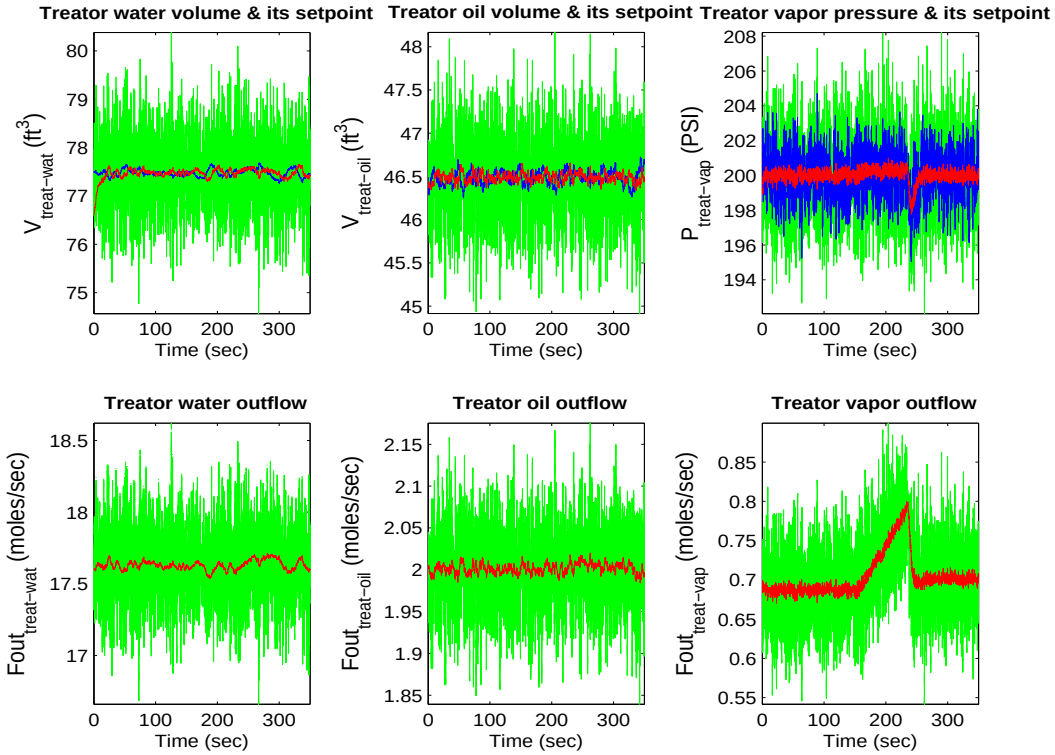

Figure 9.8: -10% ramp S_2 applied at t=150 sec. - Nominal Operating Point

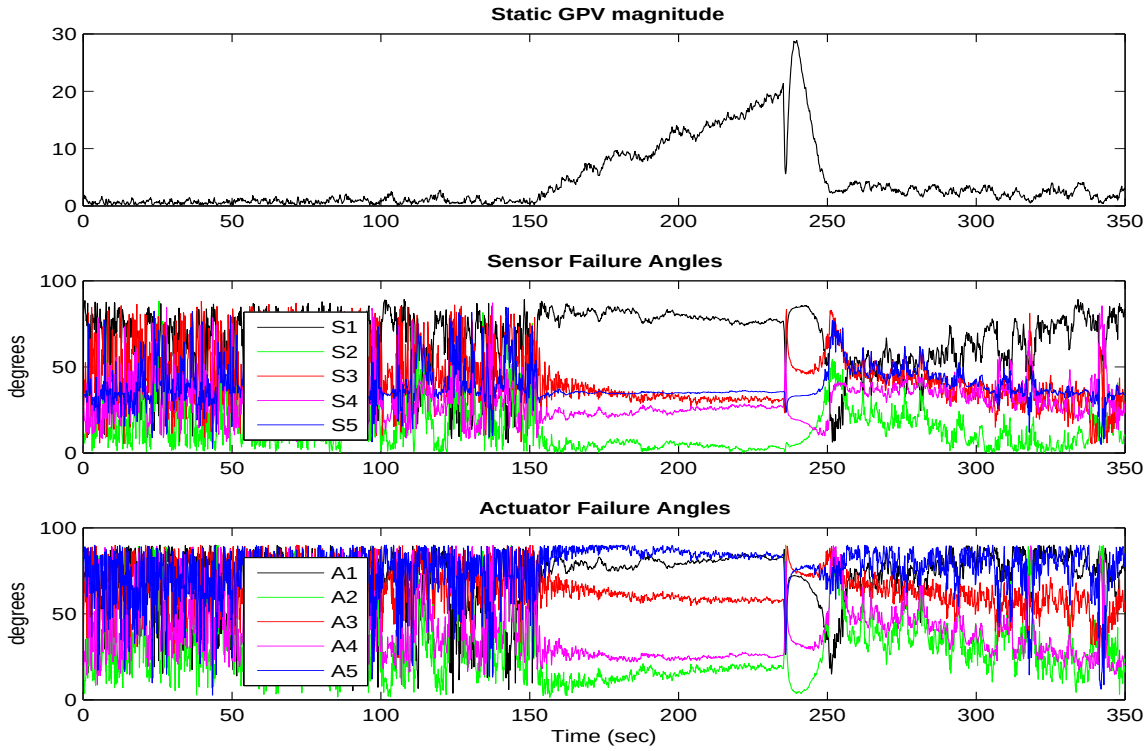

Figure 9.9: Static $|GPV|$ and $\angle GPV$, S_2 faulty

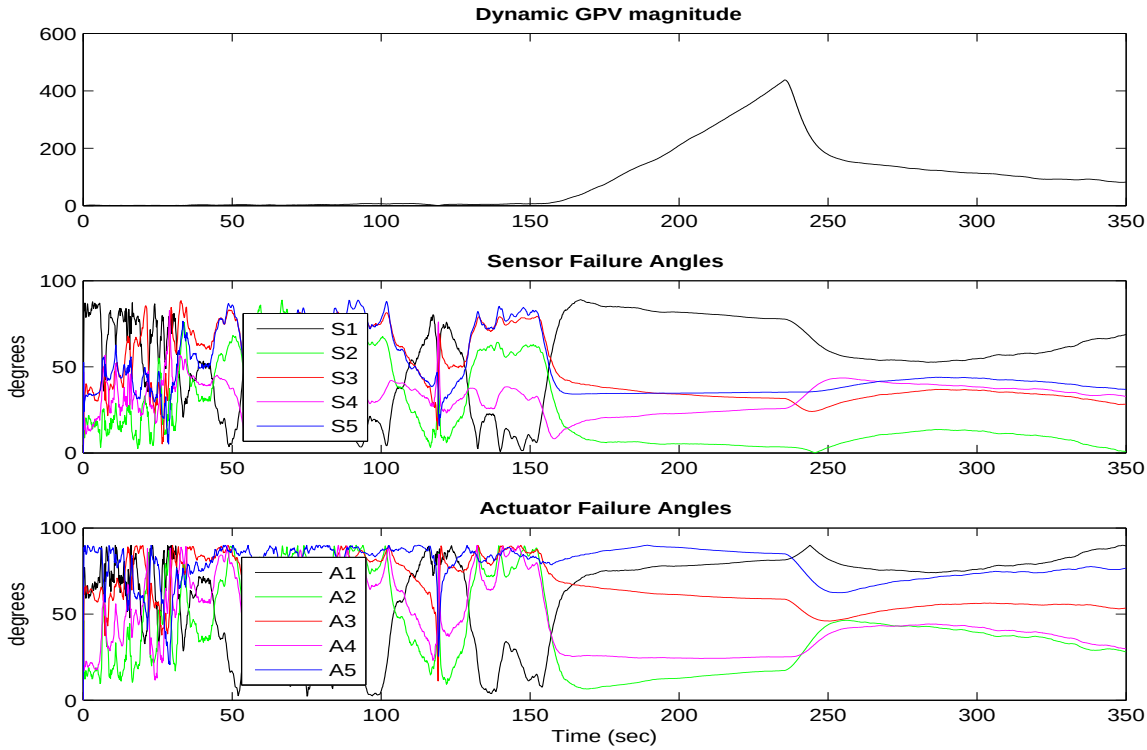

Figure 9.10: Dynamic $|GPV|$ and $\angle GPV$, S_2 faulty

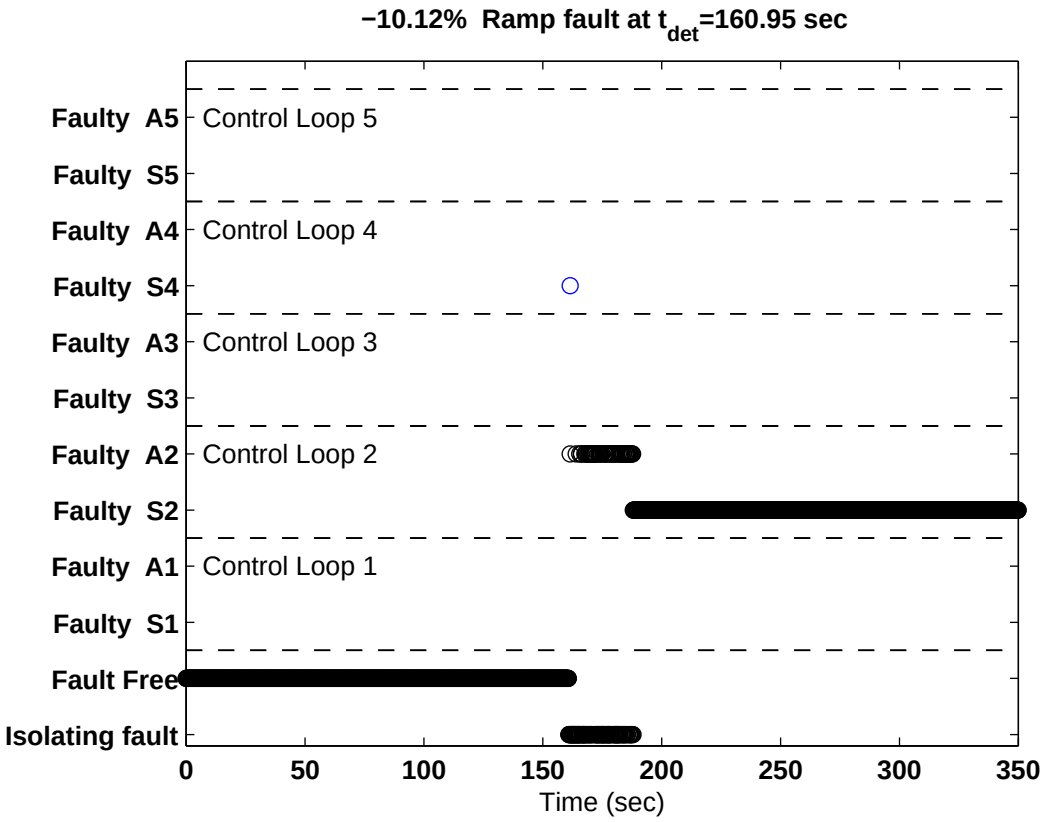

Figure 9.11: Operator display, S_2 faulty

by the static GPV. Based on these results, the isolation logic for ramp faults was changed to be based on the dynamic GPV angles as shown in Fig. 9.1. Although this modification slightly increases the isolation time, it provides a clearer isolation overall. During the first 27 sec. after the fault was detected, the FDI operator display in Fig. 9.11 shows the options "Isolating fault" and "Faulty A2" due to the GPV transient after the fault happens; this indicates that "Faulty A2" is uncertain. However, the GPV rapidly swings to point towards the sensor 2 reference direction, which is the faulty element, at t=188.1 sec. The fault size was accurately computed to be -10.12% providing an estimation error of only 1.2%. Figure 9.7 shows that after the accommodation started at t=235 sec. the corrected sensor measurement and the actual pressure value are very similar and the different process variables are driven close to their pre-fault values.

The final scenario simulates a situation where the treator water outflow valve (A_3) is stuck at +4% of its nominal value after t=150 seconds, so the water level slowly decreases. For this case the measurements are contaminated by gaussian noise with a standard deviation of 3% their actual values. It is observed in Fig. 9.13 that when the treator water outflow valve gets stuck, the desired control action attempts to decrease the water outflow to try to keep the water level at its setpoint. However, since the outflow rate is stuck at 18.33 moles/sec, the water volume keeps decreasing from its setpoint value even though the desired controller action would be to completely close the valve. It should be noticed that the oil volume and vapor pressure inside the treator are decreasing slowly after the fault happens. However, this behaviour is not very clear in the figure due to the axes scales and large amount of noise affecting the variables.

Figures 9.14 and 9.15 show a significant GPV magnitude increase about 10-20 sec. after the fault happens at t=150 sec. However, given the conservative threshold choice made to improve robustness with respect to noise and the slow effect of this fault in the other process variables, the fault is not detected until t=191.55 sec. as shown in Fig. 9.16. The isolation at t=191.55 is performed using the dynamic GPV angles in Fig. 9.15, which clearly show that the angle closest to zero corresponds to actuator 3 (A_3), immediately at the time of detection.

The previous simulation results have shown the FDIA capabilities of the GPV technique for noisy scenarios. It has been demonstrated that for a reasonably large signal-to-noise ratio (SNR) the GPV methodology is able to detect and isolate different faults within an acceptable period of time. Although the accuracy of the fault size estimation varied depending on the SNR, the simulation results showed that for every scenario the behaviour of the process improved significantly overall after the accommodation started.

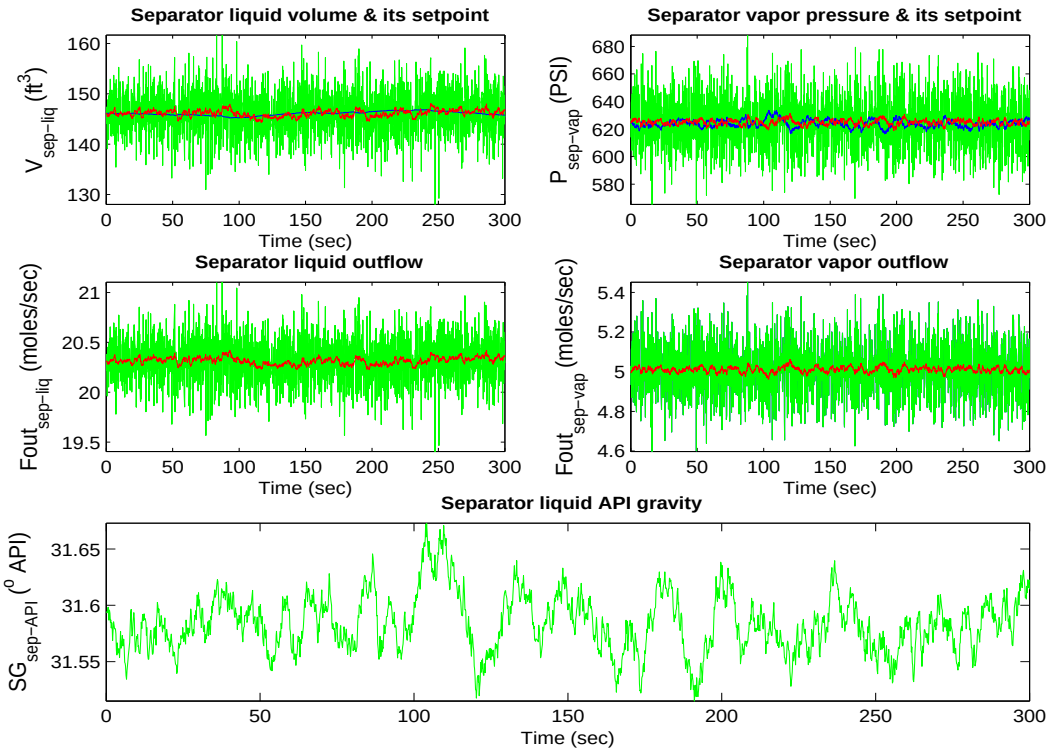

Figure 9.12: +4% A_3 applied at t=150 sec. - Nominal Operating Point

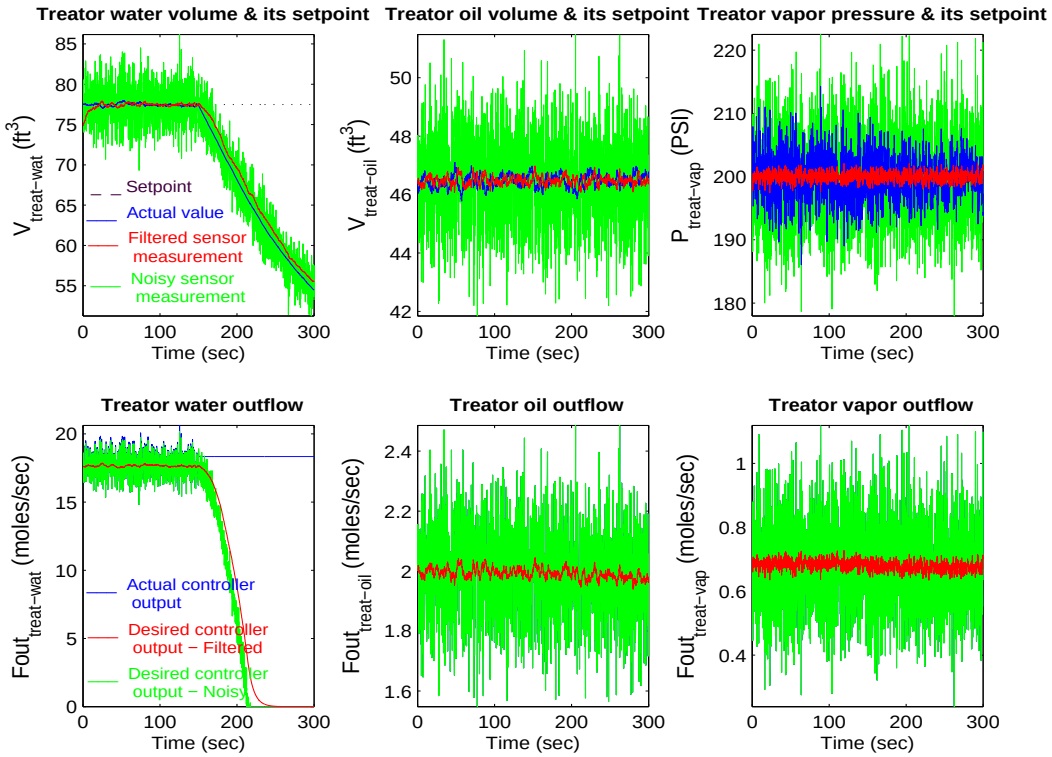

Figure 9.13: +4% A_3 applied at t=150 sec. - Nominal Operating Point

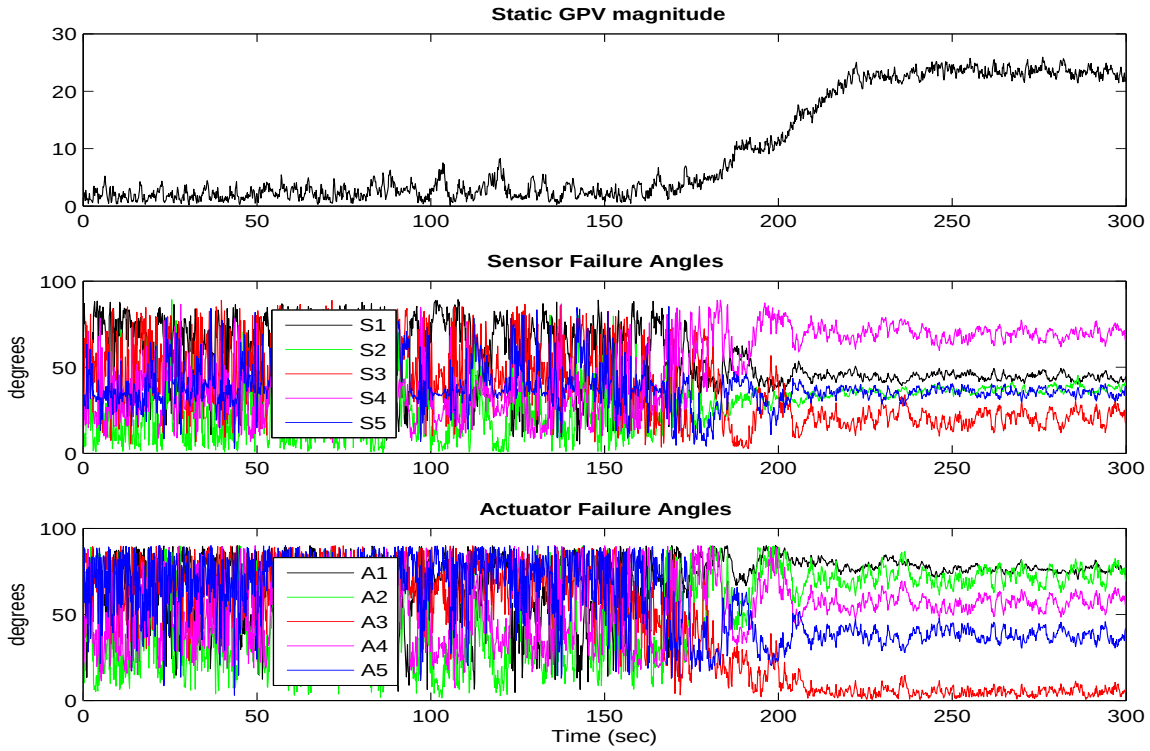

Figure 9.14: Static $|GPV|$ and $\angle GPV$, A_3 faulty

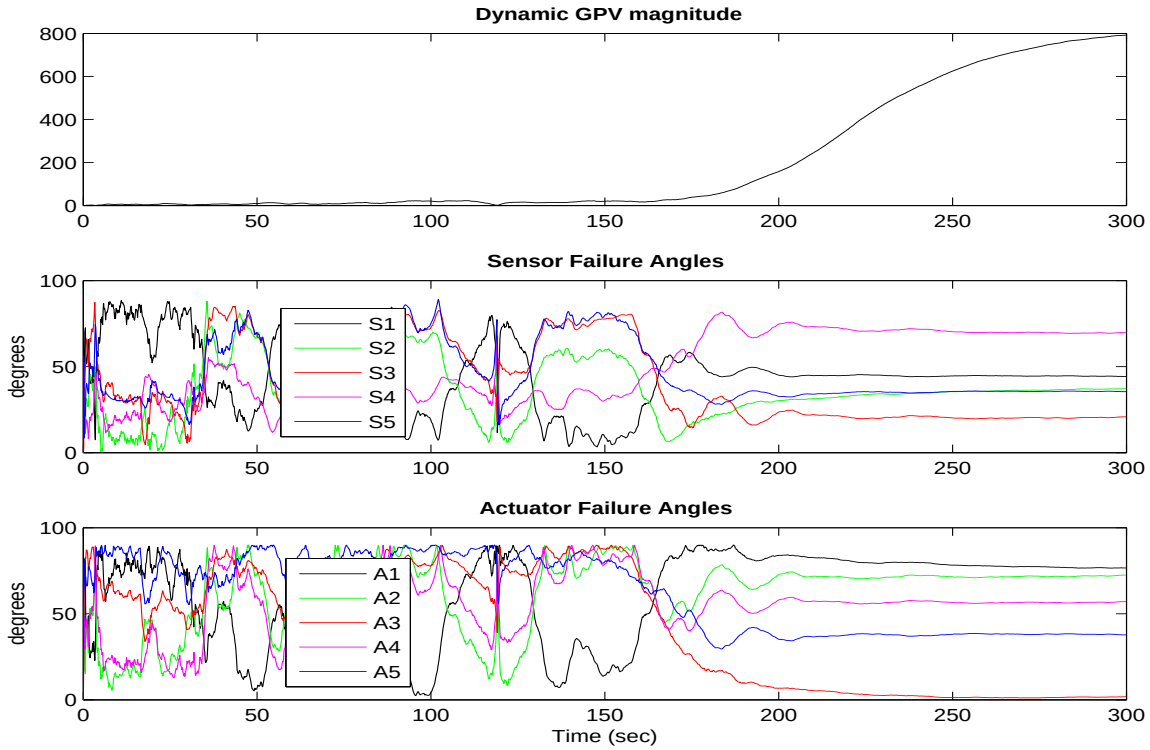

Figure 9.15: Dynamic $|GPV|$ and $\angle GPV$, A_3 faulty

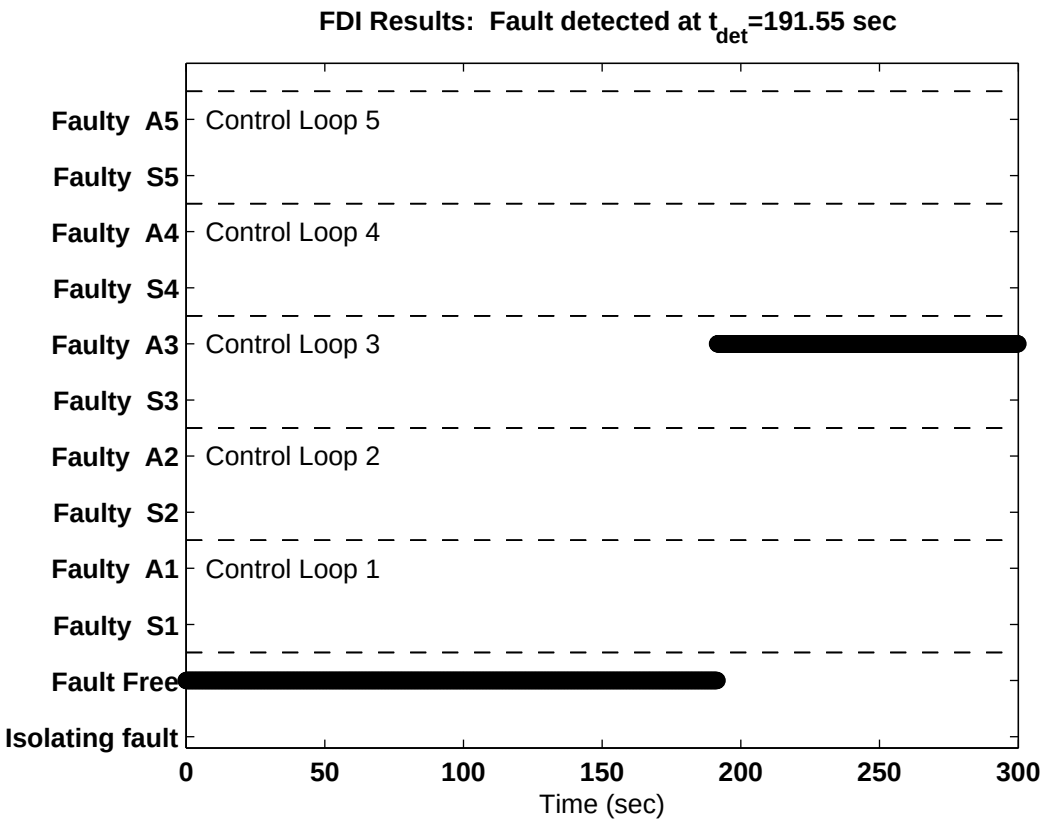

Figure 9.16: Operator display, A_3 faulty

Chapter 10

Contributions, Conclusions and Future Work

In my master's research the GPV technique was implemented and tested using a third-order linearized state space model obtained analytically for a jacketed continuously stirred tank reactor. Despite the relative simplicity of this model, it allowed us to identify different issues of the GPV technique and validate the proposed approach. In that work a new off-line systematic approach for calculating an optimal transformation matrix was developed to enhance the FDI performance and its scope in terms of the number of faults that can be isolated and disturbances that can be decoupled. Also, the special case for sensor-actuator fault isolation when an actuator reference direction lies on a sensor reference plane was identified and solved by extending the off-line transformation approach.

The FDI robustness was significantly improved by incorporating disturbance decoupling and different threshold logics. Although the results of that study were very encouraging in terms of FDI robustness, they were obtained based on the assumption that a perfect linearized model for the plant was available. It also required some off-line initialization and, most importantly, did not provide sensor fault size esti-

mation and accommodation. These were crucial limitations for the implementation of the GPV technique in real industrial applications.

The focus of my Ph.D. research was to extend the previous work substantially, to provide sensor accommodation and make the GPV approach viable for industrial implementation despite the usual lack of model availability. Thus, the proposed FDIA software was significantly extended and developed, to permit on-line initialization where the only available information is input-output data and the approximate system time constants.

Its performance was tested using a three-phase separation process with much more complex system dynamics that closely matches actual industrial applications. The proposed methodology has overcome real industrial limitations, such as using identified models with limited percentage of fitting due to the complexity of the system. A new approach for sensor accommodation has also been successfully developed, and robustness has been significantly improved by implementing a smarter decision-maker block. The GPV technique offers a systematic and flexible approach for fault detection, isolation and accommodation since it can be implemented and modified on-line for different identified models. The specific contributions of this research can be summarized as follows:

1. An initialization section was developed and integrated, making the proposed FDIA technique viable for implementation in systems where new identified models need to be computed on-line for different operating points.

2. Systematic guidelines have been established for on-line design of a system identification agent with FDIA self-diagnosis based on the identified model's quality of fit. This module plays a critical role because it allows the implementation of the GPV technique in the absence of an *a-priori* mathematical model, which

is often the case in real industrial processes. This represents a significant advance in the scope of application of quantitative model-based approaches in process industries. It also demonstrates that the identified state-space representation does not require an unrealistic percentage of fitting and that the actual system order and/or physical states do not need to be known in order to provide excellent FDI results using the directional GPV implementation.

3. A new systematic approach to implement a recursive on-line transformation matrix computation block using optimization has been effectively developed. The calculation of this transformation matrix represents a major contribution to the FDI field using directional residuals because it eliminates the restriction on the number of faults that previous researches were able to isolate. It also improves significantly FDI robustness and makes the proposed method a viable implementation for systems where the identified model requires on-line changes to accommodate nonlinear behaviour.

4. The special case for sensor-actuator faults and the hyperplane intersection problem have been identified and solved by extending the objective function during the optimization process to compute the transformation matrix. This modification has significantly improved the isolation results by reducing the ambiguous cases produced by these inevitable special geometrical situations given by the system dynamics. This is a major contribution, because it identifies and overcomes these critical limitations of FDI using directional residuals that previous researchers were not aware of.

5. An on-line threshold computation block has been proposed to allow the GPV technique implementation for new on-line identified models for different operating points. This block significantly reduces the false alarm and misdetection rates, increasing the FDI robustness with respect to modeling errors and noise.

6. The effect of the GPV filter tuning parameter σ in the GPV performance has been analyzed and guidelines to estimate an optimal value to achieve the best isolation results has been established. The proper choice of σ significantly improves the FDI results for faults that are quickly rejected by the control system and minimizes the effect of noise in the GPV computation.

7. It has been established that the GPV technique is capable of identifying faults during transients after small setpoint variations, if the magnitude threshold is properly adapted and the modeling errors are not significant.

8. The decision-making logic has been significantly improved by combining the complementary strengths of the static and dynamic GPV implementations, to reduce the detection time and decrease the ambiguous isolation cases.

9. The static $|GPV|$ signature concept proposed in this research is an important contribution, because it allows the implementation of a novel fault-size estimation and classification algorithm.

10. Two different approaches taking into account the software and hardware implementations in the plant have been effectively developed for sensor-ramp and sensor-bias fault accommodation. The proposed fault accommodation strategies proved to work effectively, even for faults that were almost untraceable due to quick rejection by the controller, providing not only closed-loop stability, but also compensating the actual value of the variable affected by the faulty sensor. The main contribution of our fault accommodation method over the traditional measurement replacement using observers is that it is less sensitive to modeling errors. Despite being a model-based approach, our technique for fault-size estimation does not calculate F_{size} or the correct sensor measurement directly from the model. Instead, F_{size} is computed based on the change in the delta of static $|GPV|$ signature, which is proportional to the size of the

fault. In other words, using deltas compensates for the fact that the fault-free residual using the linear model during the curve fitting calculation would be zero, while the fault-free residual using the actual plant is not, so the inherent plant nonlinear behaviour is retained.

11. An on-line fitting calculation block has been implemented to obtain the $|GPV|$ signatures for different identified models and compute the corresponding curve fitting equations for fault size estimation. This block allows performing fault size estimation and accommodation in systems where the identified model requires on-line changes to accommodate nonlinear behaviour.

12. The FDIA robustness with respect to noise has been successfully evaluated for scenarios with reasonably large signal to noise ratio. The proposed FDIA algorithm was able to detect and isolate different faults within an acceptable period of time, and it also improved significantly the system performance once the accommodation started.

13. It has been demonstrated that the computation time required to perform FDI using the GPV approach is significantly shorter than the required by the bank of Kalman filters method. The shorter computation time makes the GPV technique a more viable option for real-time FDI implementation in fast industrial processes.

14. A complete on-line fault detection, isolation and accommodation technique has been successfully developed using input-output measurements from the separation process as the only available information. The proposed FDIA technique has achieved excellent isolation and accommodation results even for ramp sensor faults slowly varying with time under noisy conditions. The fact that this FDIA technique can be fully designed on-line makes it easy to deploy and adapt to changes in real industrial processes. Also, since the complexity

of the separator closely simulates large-scale process, these results are very promising for a future industrial implementation.

Future research will implement the proposed FDIA technique using the actual data originating from one of the pilot plant facilities involved in the Petroleum Applications of Wireless Systems (PAWS) project. This will allow the GPV technique performance to be tested in an actual industrial process.

Bibliography

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

[2] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri. A review of process fault detection and diagnosis, part ii: Qualitative models and search strategies. *Computers and chemical engineering*, 27:313–326, 2003. Elsevier Science Ltd.

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

[4] F. Gustafsson. Stochastic fault diagnosability in parity spaces. *Proc. IFAC World Congress*, 2002.

[5] 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, Vancouver, BC, Canada*, 2005.

[6] M. Omana and J. H. Taylor. Fault detection, isolation and accommodation using the generalized parity vector technique. 17^{th} *IFAC World Congress, Seoul, Korea*, pages 1914–1921, 2008.

[7] E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on automatic control*, 29(7):603–614, 1984.

[8] J. J. Gertler. Fault detection and isolation using parity relations. *Control Eng. Practice*, 5(5):653–661, 1997.

[9] J. J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control systems magazine*, 8(6):3–11, 1988.

[10] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6):601– 611, 1984.

[11] J. J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, Inc., 1998.

[12] J. J. Gertler and D. Singer. A new structural framework for parity equation-based failure detection and isolation. *Automatica*, 26(2):381–388, 1990.

[13] N. Viswanadham, 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*, 3(1):45–72, MITA press, 1987.

[14] J. J. Gertler and R. Monajemy. Generating directional residuals with dynamic parity relations. *Automatica*, 31(4):627–635, 1995.

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

[16] A.D. Pouliezos and G. S. Stavrakakis. *Real Time Fault Monitoring of Industrial Processes*. Kluwer Academic Publishers., 1994.

[17] R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637–640, 1971.

[18] B. Friedland. Maximum-likelihood estimation of a process with random transitions (failures). *IEEE Trans. on Automatic Control*, AC-24(6):932–937, December 1979.

[19] G. G. Leininger. Model degradation effects on sensor failure detection. *Proc. American Control Conference, Blacksburg, VA*, (FP-3a), 1981.

[20] M. Basseville and A. Benveniste. *Detection of Abrupt Changes in Signals and Dynamical Systems*. Springer-Verlag, Berlin, DE, 1985.

[21] R. Nikoukhah. Innovations generation in the presence of unknown inputs: Aplication to robust failure detection. *Automatica*, 30:1851–1867, 1994.

[22] T. Kobayashi and D. L. Simon. Application of a bank of kalman filters for aircraft engine fault diagnostics. *Nasa and U.S. Army Research Laboratory*, 2003.

[23] P. D. Hanlon and P. S. Maybeck. Characterization of kalman filter residuals in the presence of mismodeling. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1), January 2000.

[24] B. Jiang, M. Staroswiecki, and V Cocquempot. Fault accommodation for nonlinear dynamic systems. *IEEE transactions on automatic control*, 51(9), 2006.

[25] D. U. Campos-Delgado, E. Palacios, and D. R. Espinoza-Trejo. Fault detection, isolation, and accommodation for LTI systems based on GIMC structure. *Journal of control science and engineering*, (853275), 2006.

[26] Y. Huang, G. V. Reklaitis, and V. Venkatasubramanian. A model-based fault accommodation system. *Industrial and Engineering Chemistry Research*, 41(16):3806–3821, 2002.

[27] G. Tortora, B. Kouvaritakis, and D. Clarke. Fault-accommodation with intelligent sensors. *Automatica*, 39(7):1227–1233, 2003.

[28] L. R. Cork, R. Walker, and S. Dunn. Fault detection, identification and accommodation techniques for unmanned airborne vehicle. *Cooperative Research Centre for Satellite Systems and Queensland University of Technology, Australia*.

[29] S. Simani. Identification and fault diagnosis of a simulated model of an industrial gas turbine. *IEEE Transactions on Industrial Informatics*, 1(3):202–216, 2005.

[30] X. Roger, Z. Guangfan, L. Haynes, C. Kwan, and K. Semega. Sensor validation using nonlinear minor component analysis.

[31] A. F. Sayda. *Intelligent Control and Asset Management of Oil and Gas Production Facilities*. PhD thesis, University of New Brunswick, 2008.

[32] L. Wong. On-line fault diagnosis using signed digraphs. Master's thesis, University of New Brunswick, 2006.

[33] J. He. Neuro-fuzzy based fault diagnosis for nonlinear processes. Master's thesis, University of New Brunswick, 2006.

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

[35] 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*, pages 2560–2566, 2006.

[36] 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*, pages 970–975, 2007.

[37] M. Omana. Robust fault detection and isolation using a parity equation implementation of directional residuals. Master's thesis, Univeristy of New Brunswick, 2006.

[38] P. J. Antsaklis. Proper stable transfer matrix factorization and internal system descriptions. *IEEE Transactions on automatic control*, 31(7):634–638, 1986.

[39] M. Vidyasagar. *Control System synthesis: A Factorization Approach*. The MIT Press, 1985.

[40] C. C. MacDuffee. *Vectors and Matrices*. Math. Assoc. Amer., 1943.

[41] G. E. Dullerud and F. G. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Springer, 2000.

[42] The MathWorks Inc. *System Identification Toolbox 7 User's Guide For Use with* MATLAB®, 1988 - 2008.

[43] G. Welch and G. Bishop. An introduction to the kalman filter. *Department of Computer Science, University of North Carolina*, 2006.

[44] M. S. Grewal and A. P. Andrews. *Kalman Filtering Theory and Practice*. Prentice Hall, 1993.

[45] D. C. Lay. *Linear Algebra and its Applications*. Addison-Wesley, 2^{nd} edition, 1997.

[46] J. H. Taylor and A. Sayda. An intelligent architecture for integrated control and asset management for industrial processes. *Proc. IEEE International Symposium on Intelligent Control, Limassol, Cyprus*, 2005.

[47] 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.

[48] A. Sayda and J. H. Taylor. Toward a practical multi-agent system for integrated control and asset management of petroleum production facilities. *Proc. IEEE International Symposium on Intelligent Control, Singapore*, 2007.

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

[50] 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, Saint-Etienne, France*, 2006.

[51] P. Moreno and J. H. Taylor. Data reconciliation in the abscence of a dynamic nonlinear model. M.sc. proposal, University of New Brunswick, 2009.

[52] E. Durfee and T. Montgomery. Mice: A flexible test bed for intelligent coordination experiments. *Proceedings of the 9th workshop on distributed AI, Rosario, Washington*, 1989.

[53] M. J. Wooldridge. *An introduction to multiagent systems*. Chichester, England: Wiley, 2002.

[54] P. H. Menold, R. K. Pearson, and F. Allgower. Online outlier detection and removal. *Proc. of the 7th Mediterranean Conference on Control and Automation, Haifa, Israel*, 1999.

[55] M. J. Liebman, T. F. Edgar, and L. S. Lasdon. Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers chem. Eng*, 16(10/11):963–986, 1992.

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

[57] H. J. A. F. Tulleken. Generalized binary noise test-signal concept for improved identification-experiment design. *Automatica*, 26(1):37–40, 1990.

[58] The MathWorks Inc. *Optimization Toolbox For Use with* MATLAB®, 1990 - 2004.

[59] S. Simani, C. Fantuzzi, and R. J. Patton. *Model-based Fault diagnosis in Dynamic Systems Using Identification Techniques*. Springer, 2003.

[60] P. M. Frank. Residual evaluation for fault diagnosis based on adaptive fuzzy thresholds. *IEE Colloquium on Qualitative and Quantitative Modelling Methods for Fault Diagnosis*, April 1995.

[61] A. Sayda. A benchmark model of a jacketed stirred tank heater for fault detection and isolation. May 2004.

Appendix A

Complementary Time Histories

This appendix shows the corresponding time histories for different scenarios presented through this thesis. For simplicity, the fault scenario is briefly described in the figure's caption. A more complete description is given in the corresponding section where FDI was performed for each scenario.

It should be noticed that Fig. A.1 does not show any changes after the fault in the treator water outflow valve is applied at t=160 sec. because there is no feedback from the treator to the separator. Also, in Fig. A.2, the stuck actuator causes the tank to drain, giving the appearance of instability.

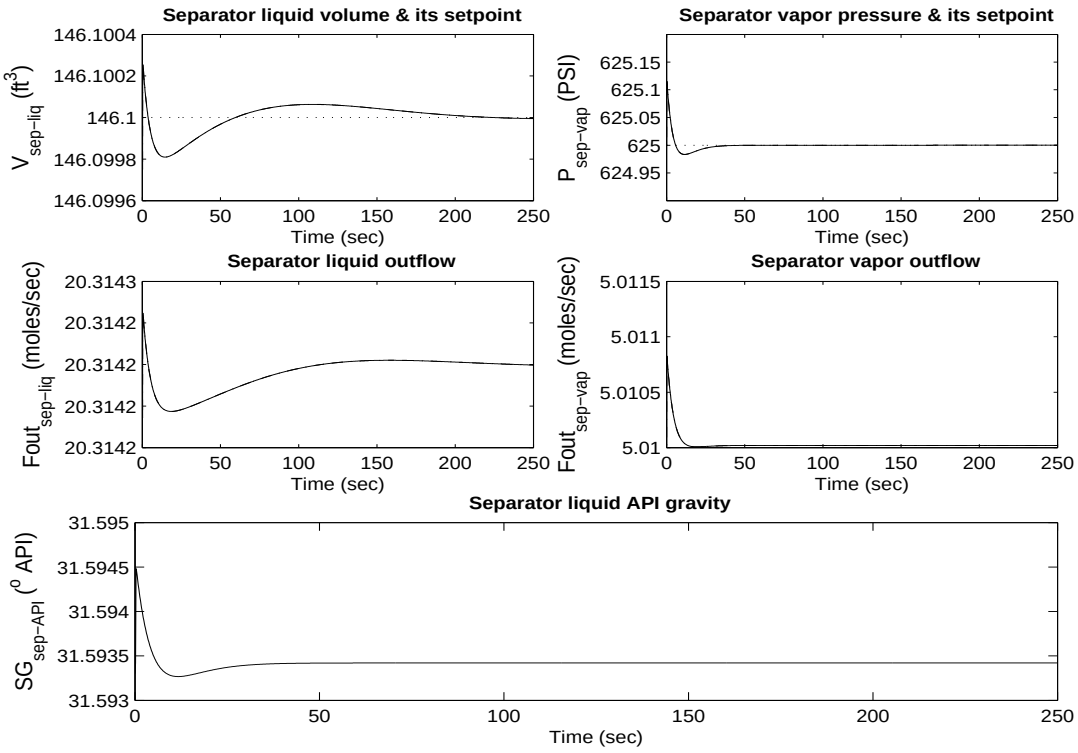

Figure A.1: +9% bias A_3 applied at t=160 sec. - Nominal Operating Point

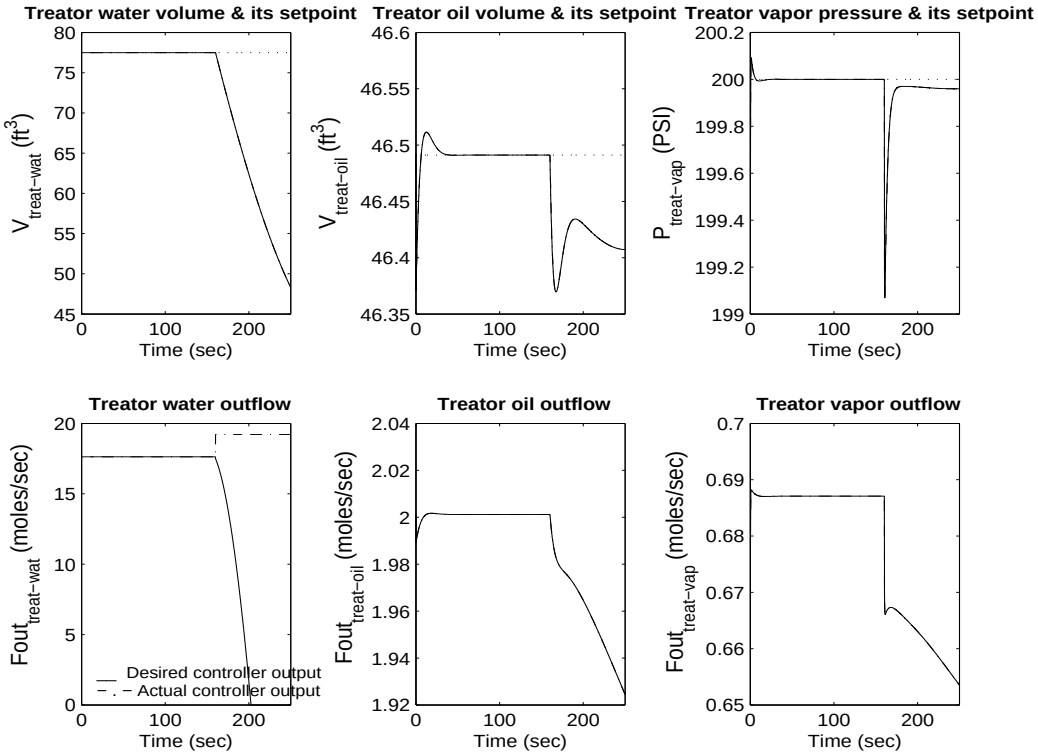

Figure A.2: +9% bias A_3 applied at t=160 sec. - Nominal Operating Point

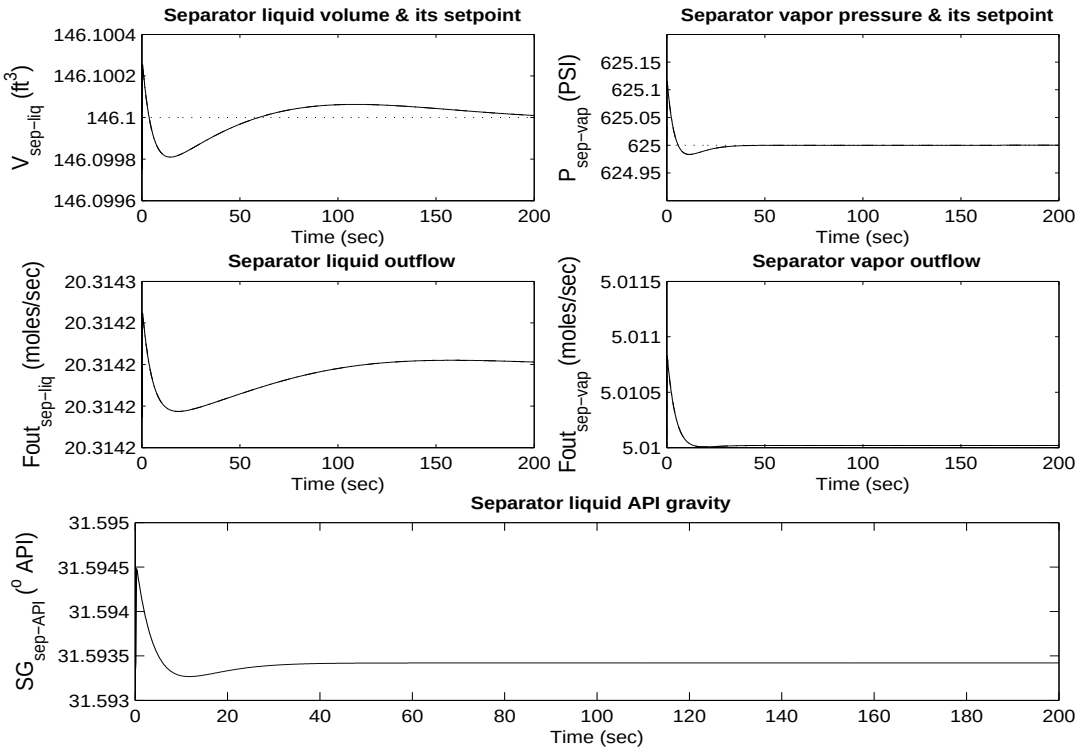

Figure A.3: +1% bias S_4 applied at t=160 sec. - Nominal Operating Point

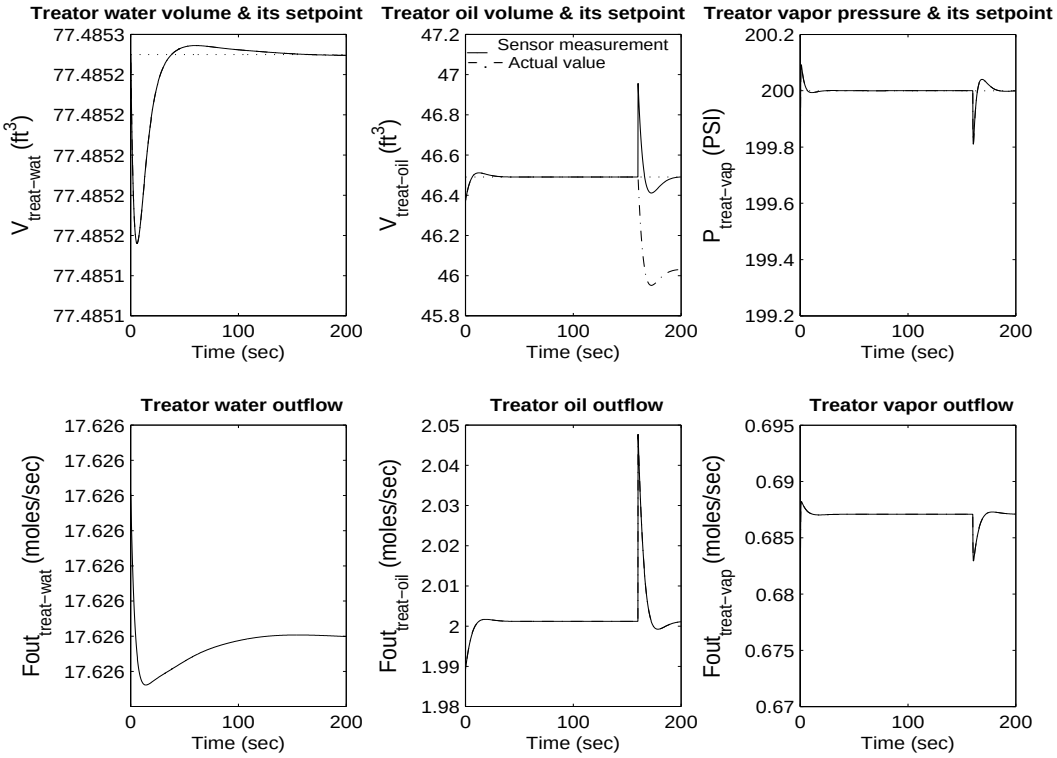

Figure A.4: +1% bias S_4 applied at t=160 sec. - Nominal Operating Point

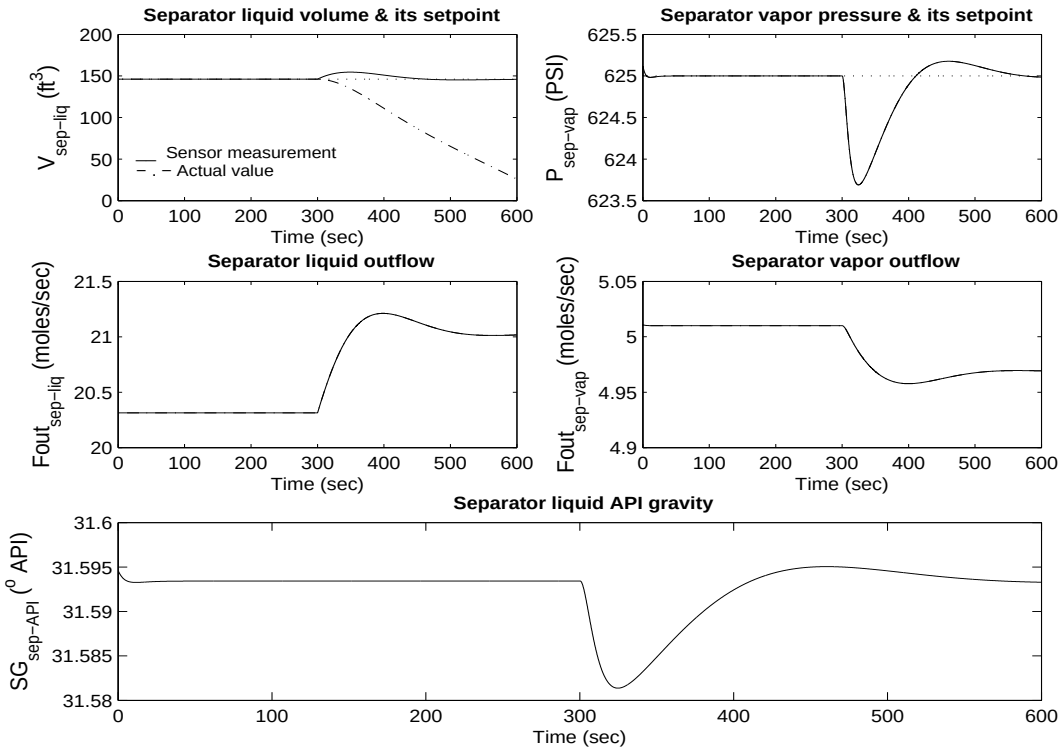

Figure A.5: +16.43%/min ramp S_1 applied at t=300 sec. - Nominal Operating Point

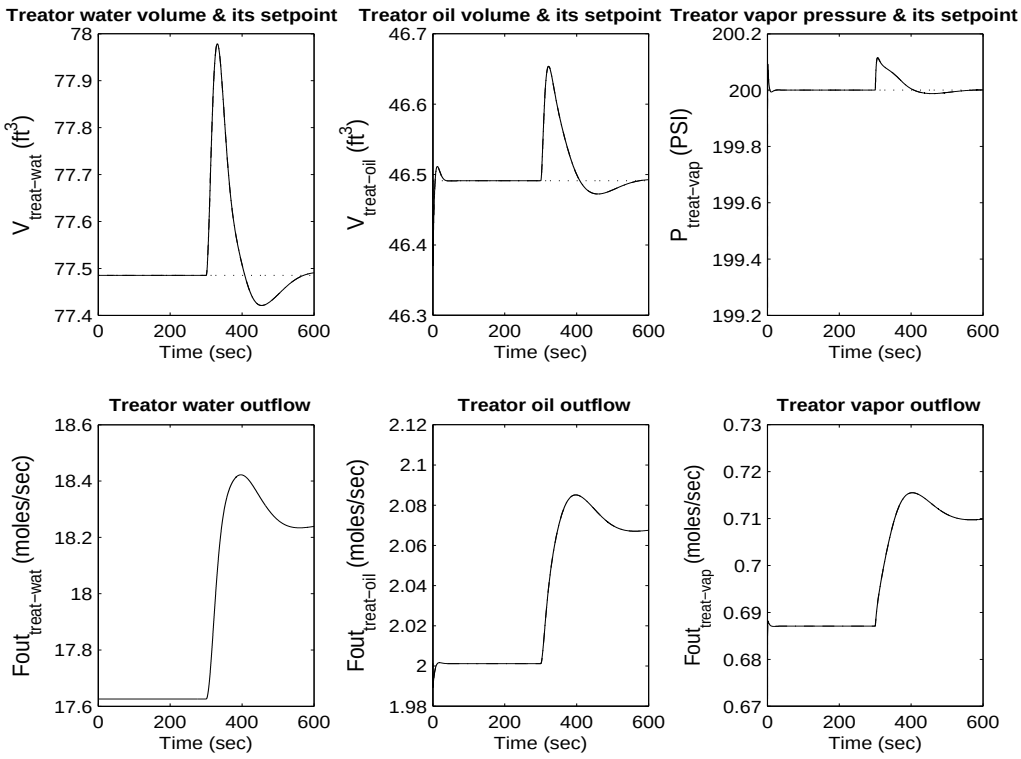

Figure A.6: +16.43%/min ramp S_1 applied at t=300 sec. - Nominal Operating Point

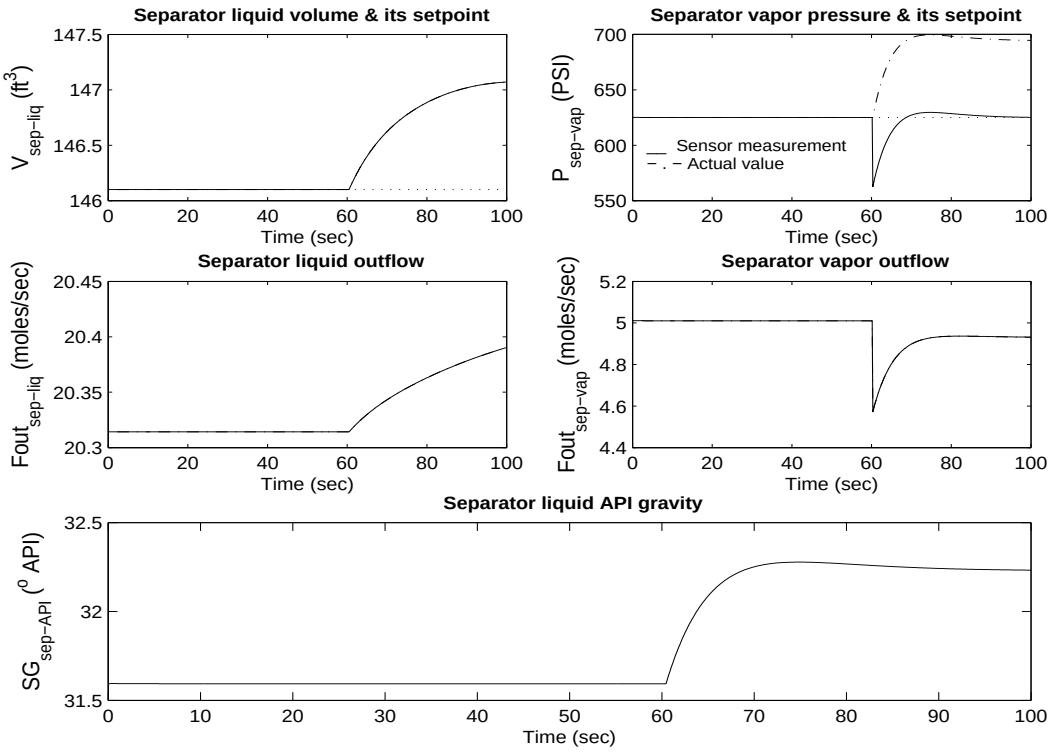

Figure A.7: -10% bias S_2 applied at t=60 sec. - Nominal Operating Point

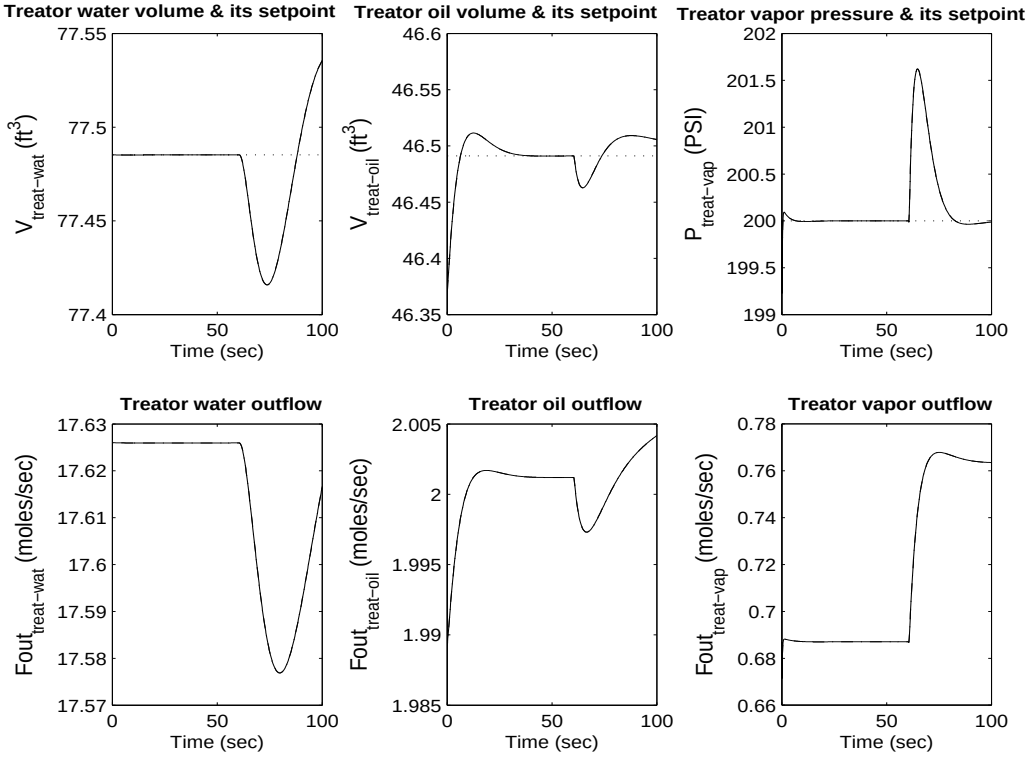

Figure A.8: -10% bias S_2 applied at t=60 sec. - Nominal Operating Point

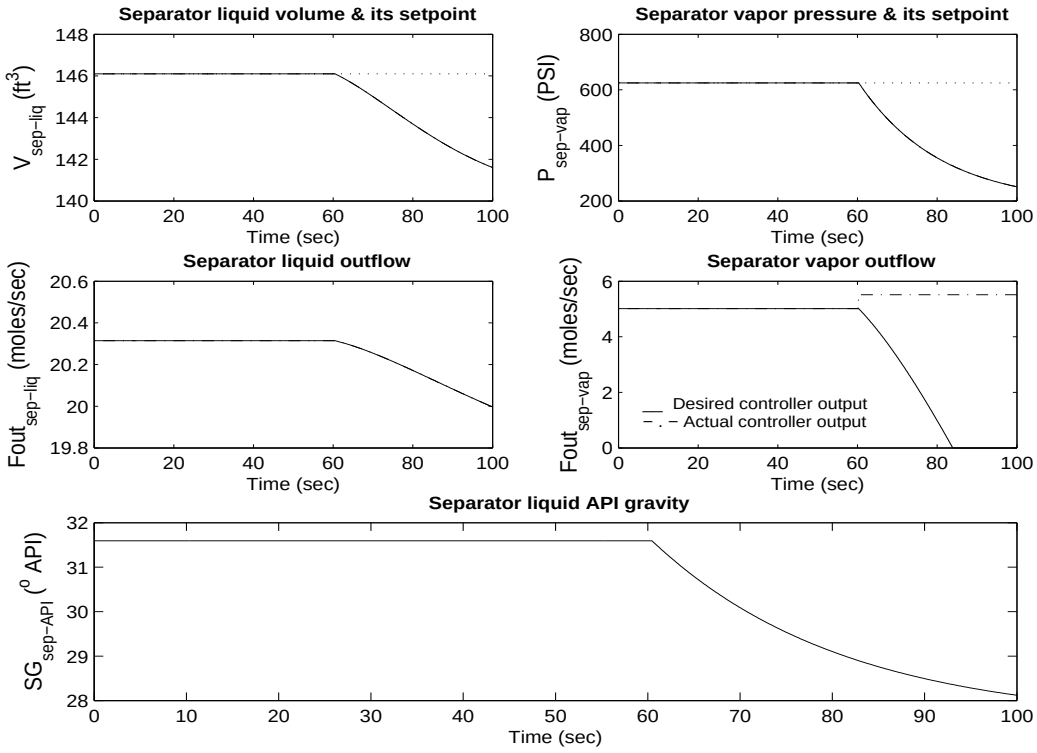

Figure A.9: +10% bias A_2 applied at t=60 sec. - Nominal Operating Point

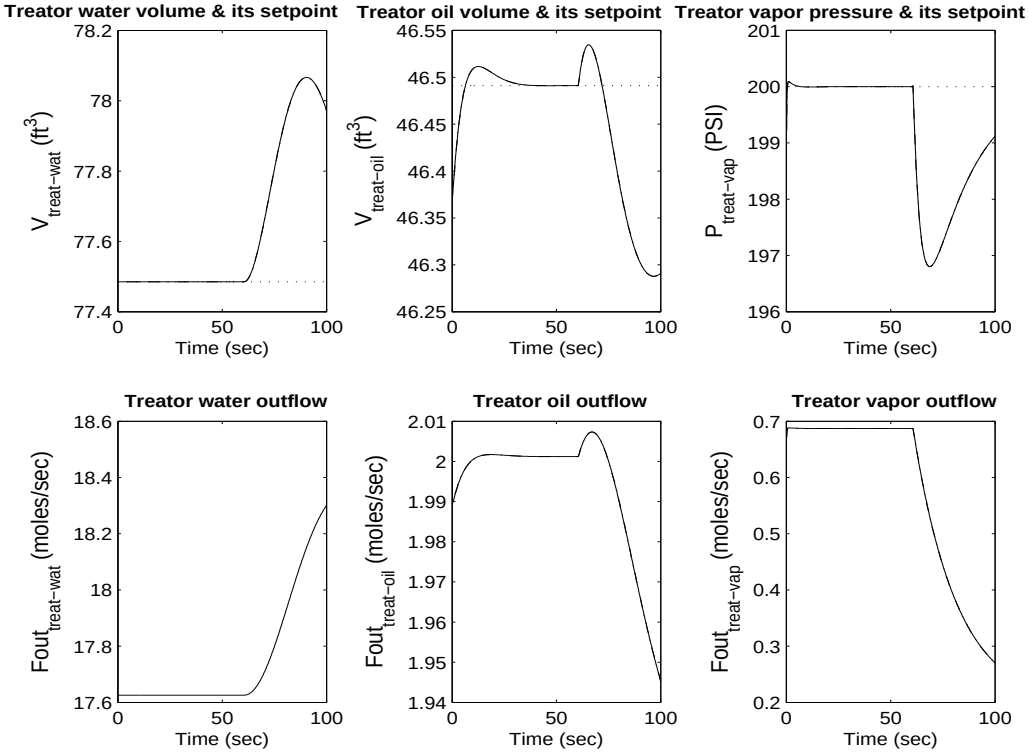

Figure A.10: +10% bias A_2 applied at t=60 sec. - Nominal Operating Point

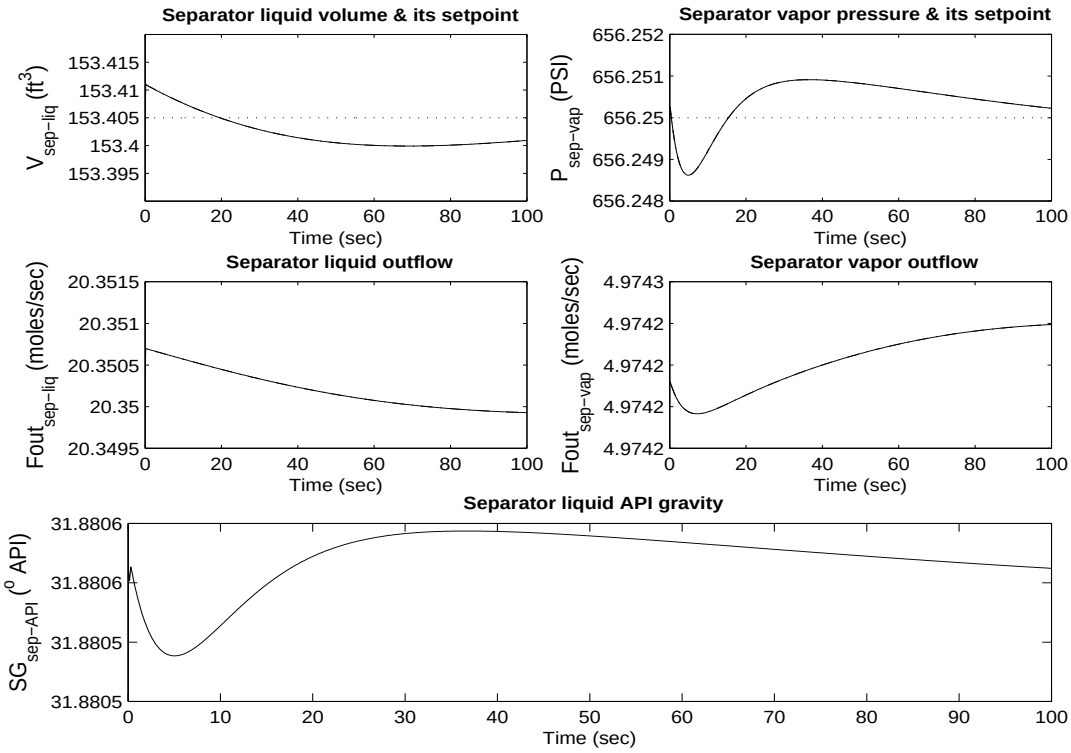

Figure A.11: -5%/min ramp S_3 applied at t=60 sec. - ΔSP=5%

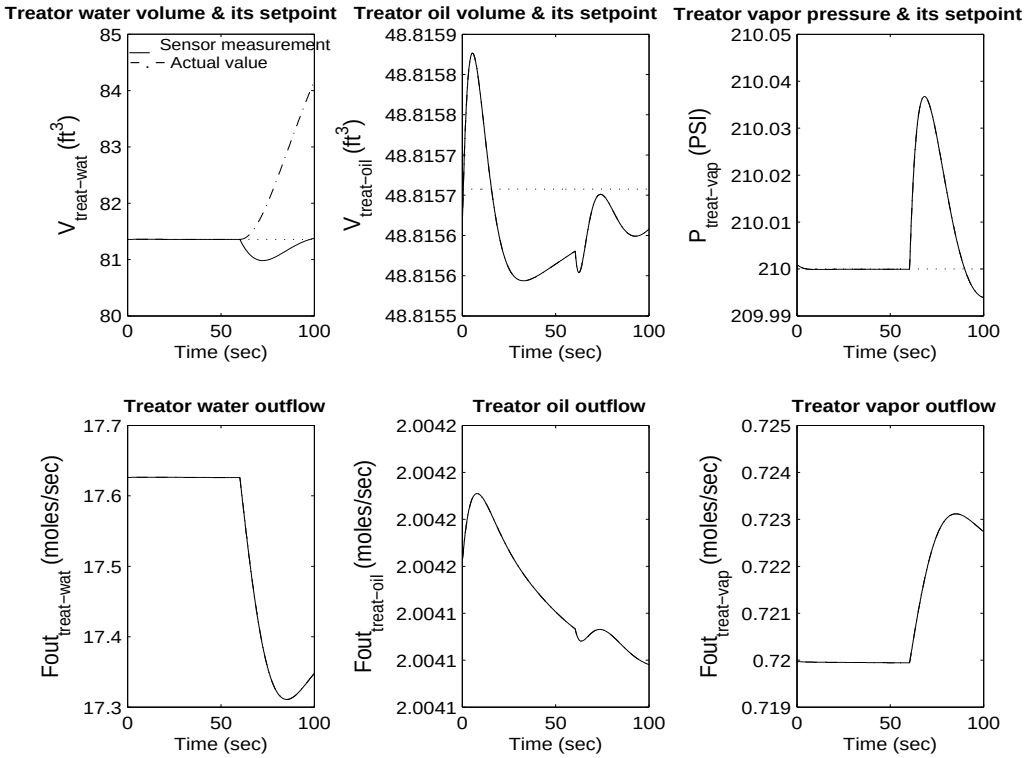

Figure A.12: -5%/min ramp S_3 applied at t=60 sec. - ΔSP=5%

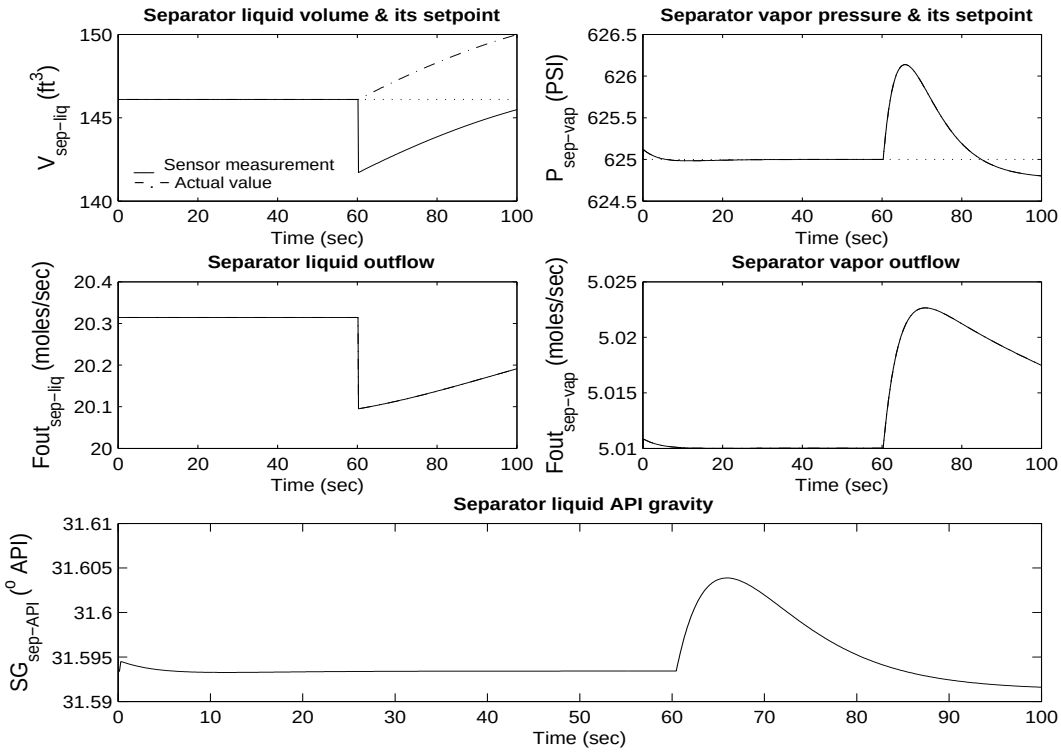

Figure A.13: -3% bias S_1 applied at t=60 sec. - Nominal Operating Point

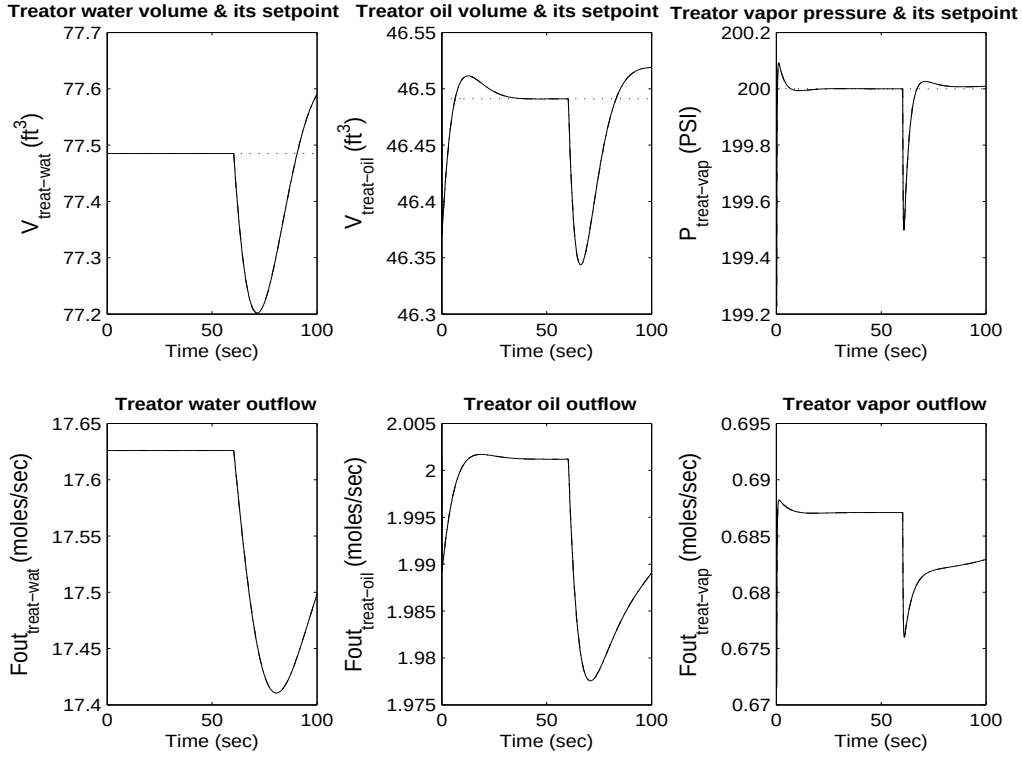

Figure A.14: -3% bias S_1 applied at t=60 sec. - Nominal Operating Point

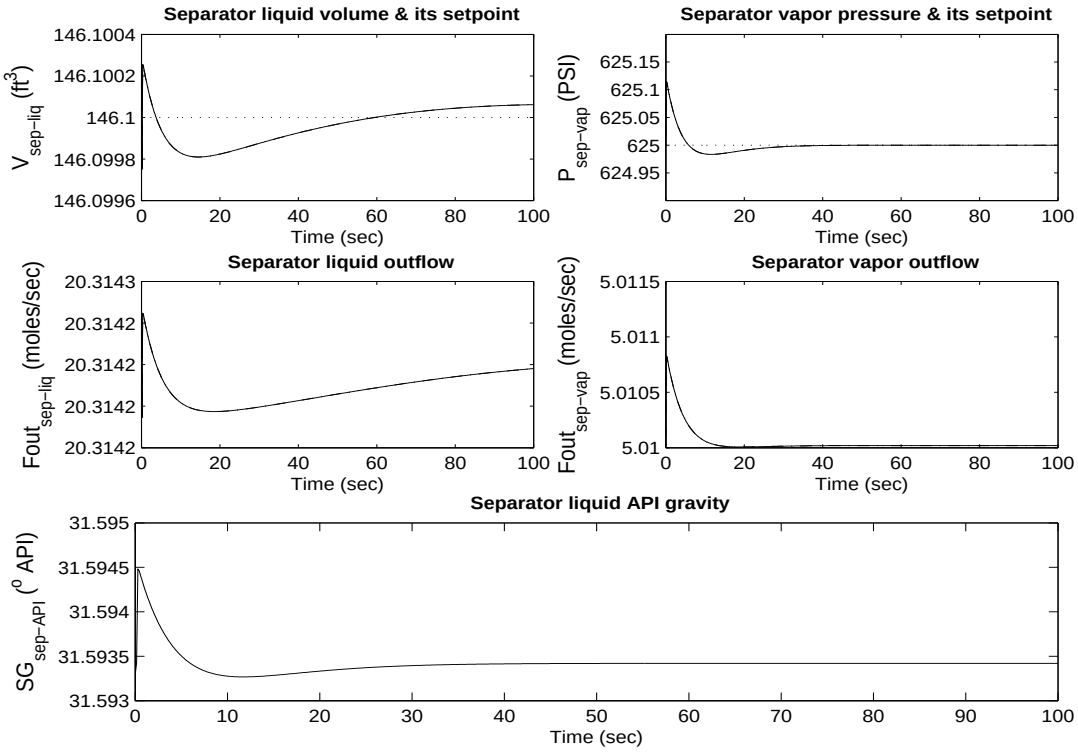

Figure A.15: -10%/min ramp S_4 applied at t=60 sec. - Nominal Operating Point

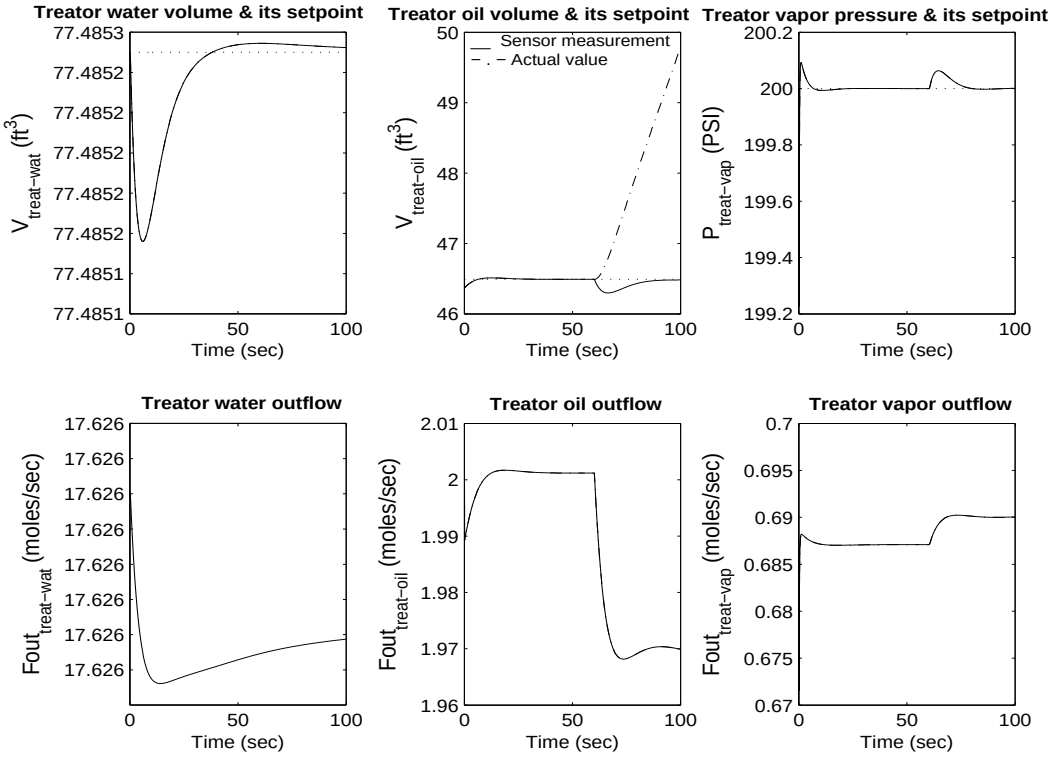

Figure A.16: -10%/min ramp S_4 applied at t=60 sec. - Nominal Operating Point

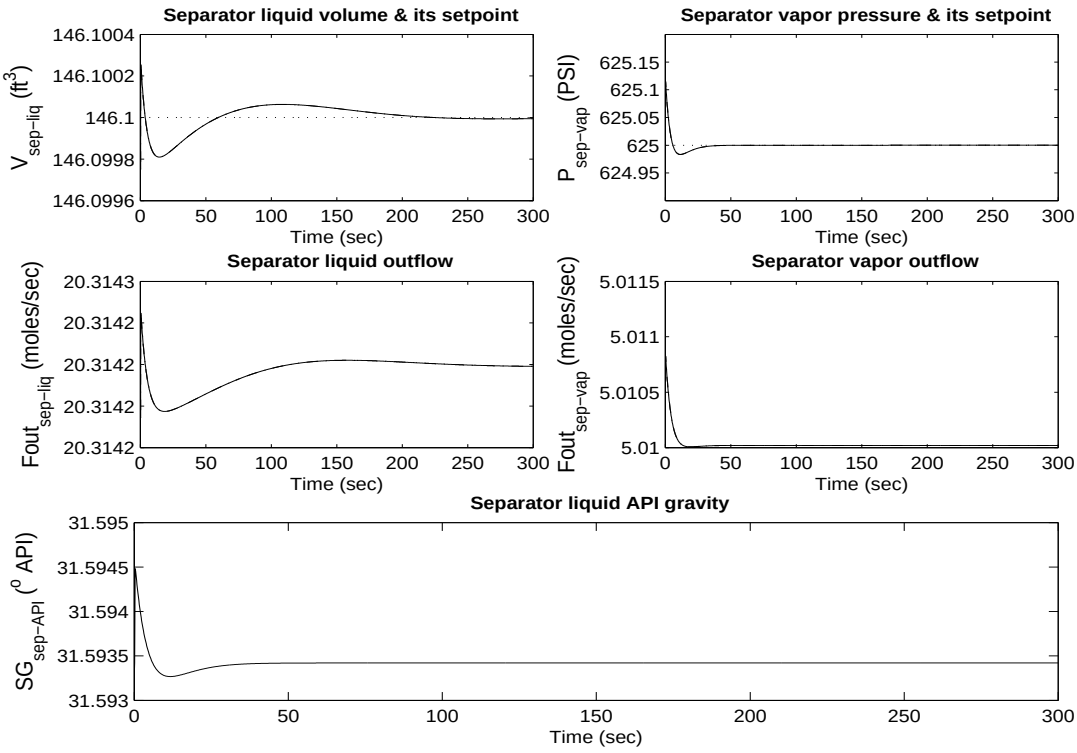

Figure A.17: +1%/min ramp S_3 applied at t=200 sec. - Nominal Operating Point

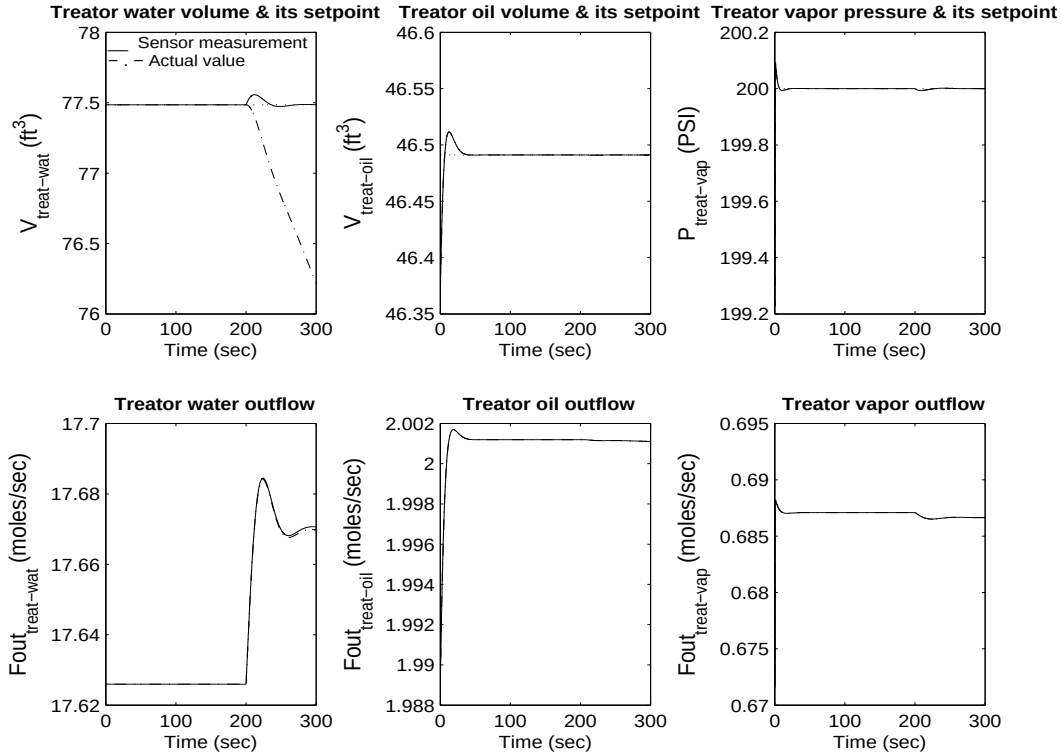

Figure A.18: +1%/min ramp S_3 applied at t=200 sec. - Nominal Operating Point

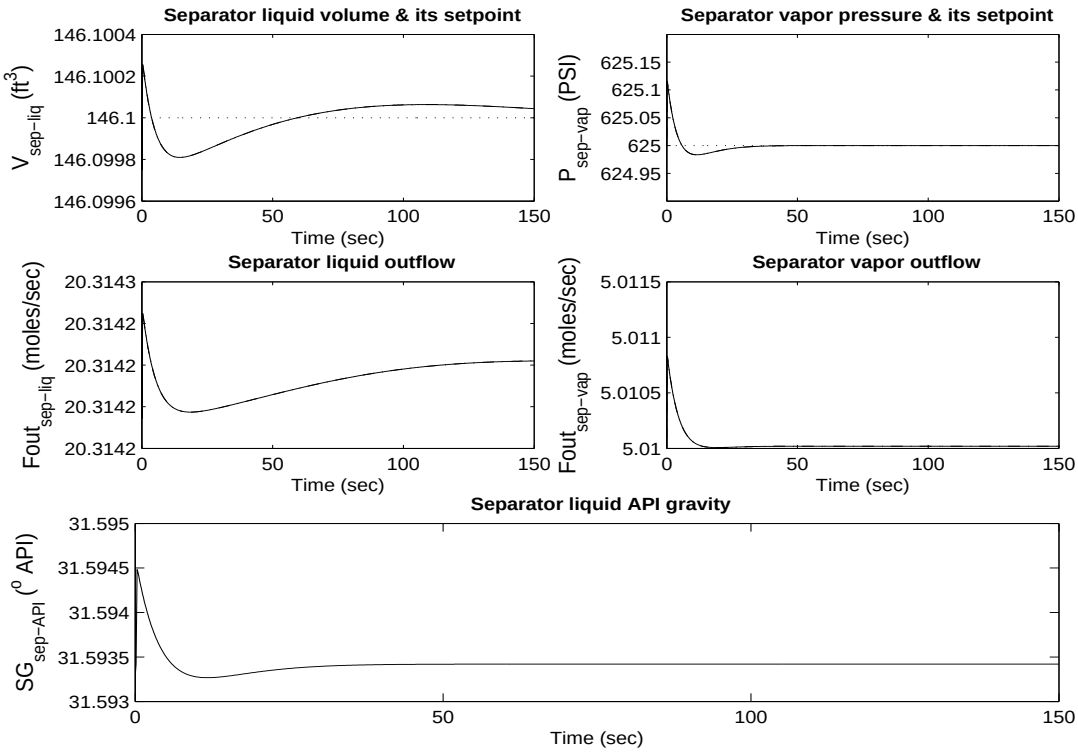

Figure A.19: -25% bias S_5 applied at t=60 sec. - Nominal Operating Point

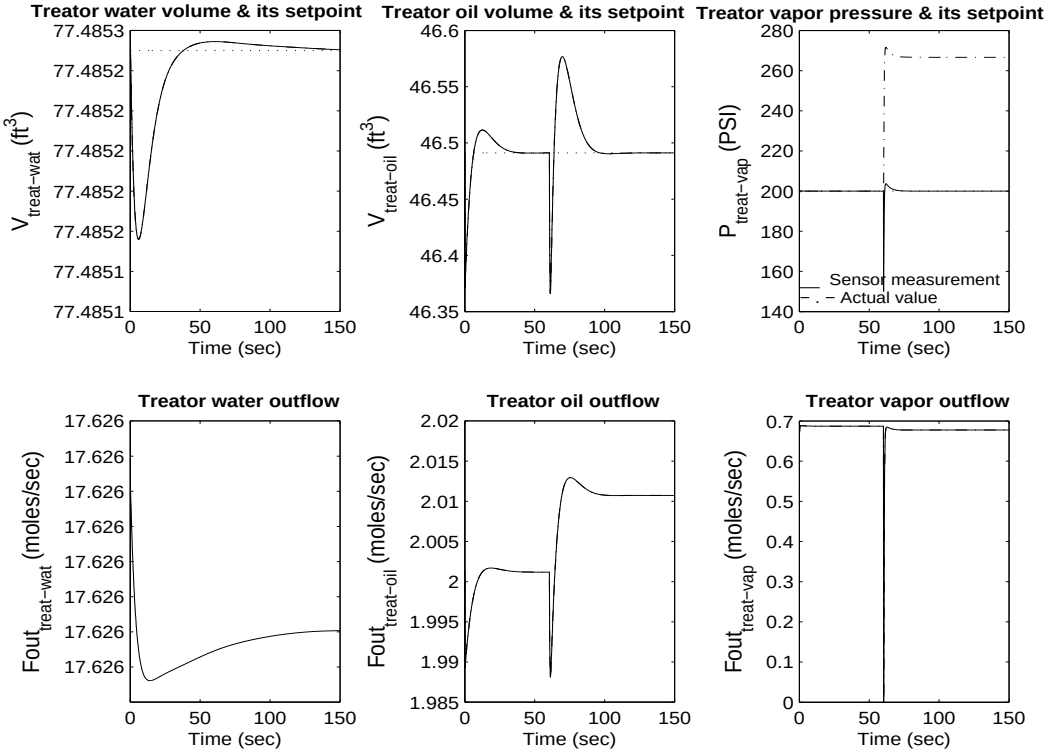

Figure A.20: -25% bias S_5 applied at t=60 sec. - Nominal Operating Point

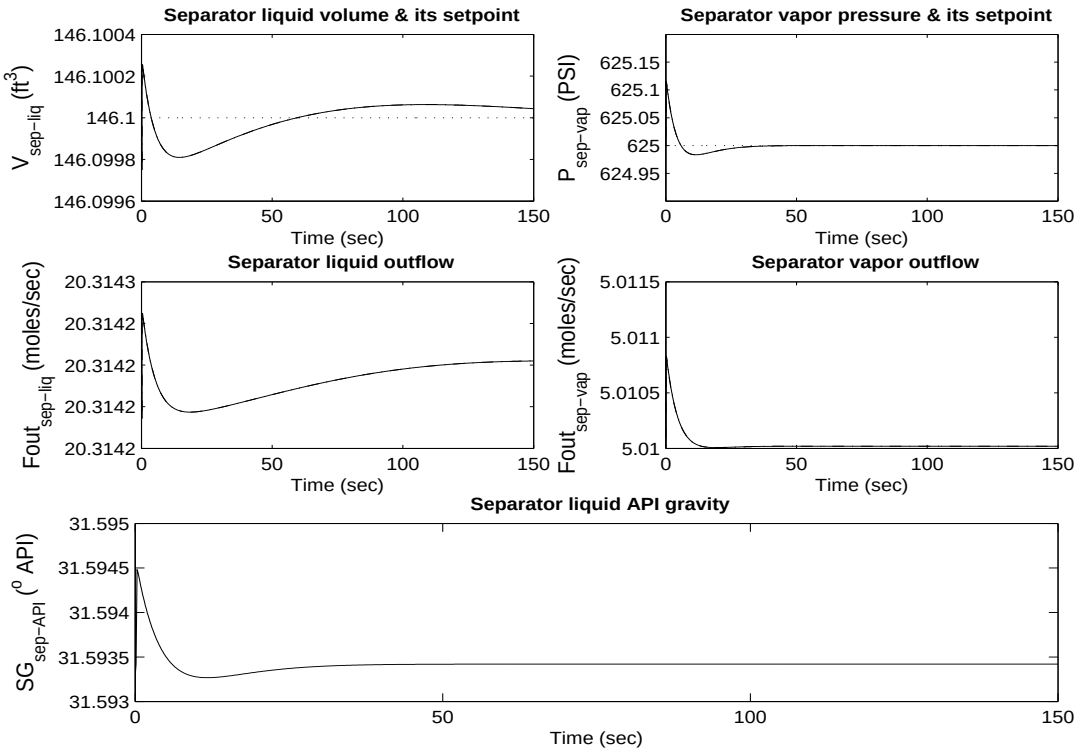

Figure A.21: -25% bias A_5 applied at t=60 sec. - Nominal Operating Point

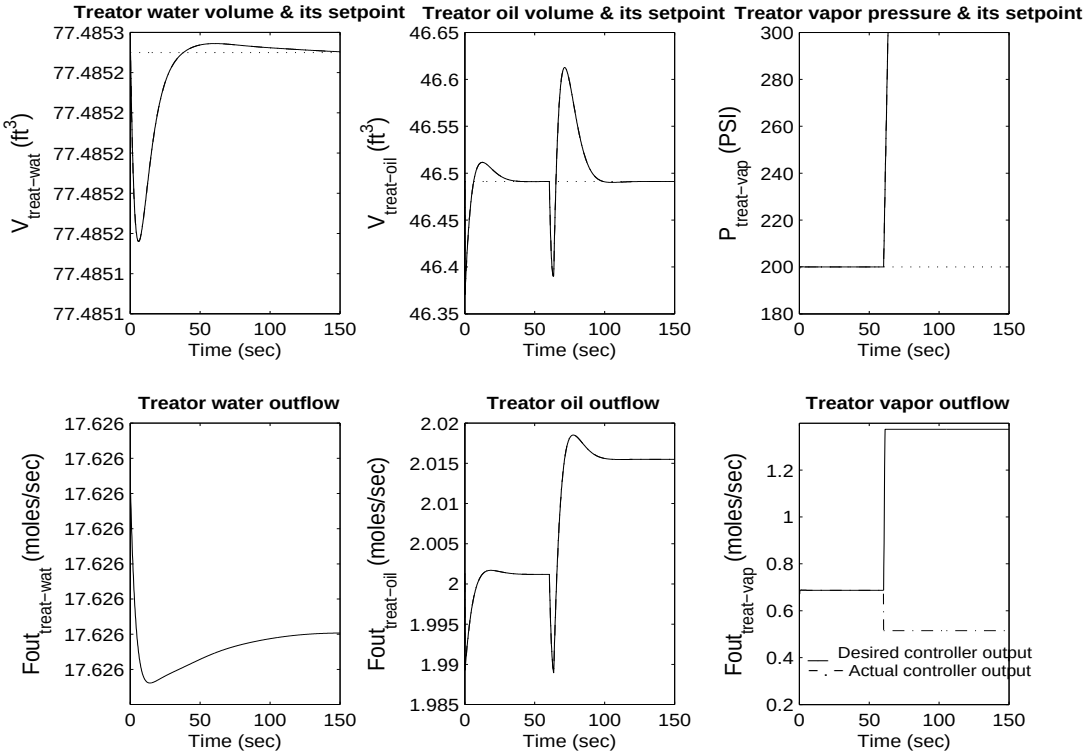

Figure A.22: -25% bias A_5 applied at t=60 sec. - Nominal Operating Point

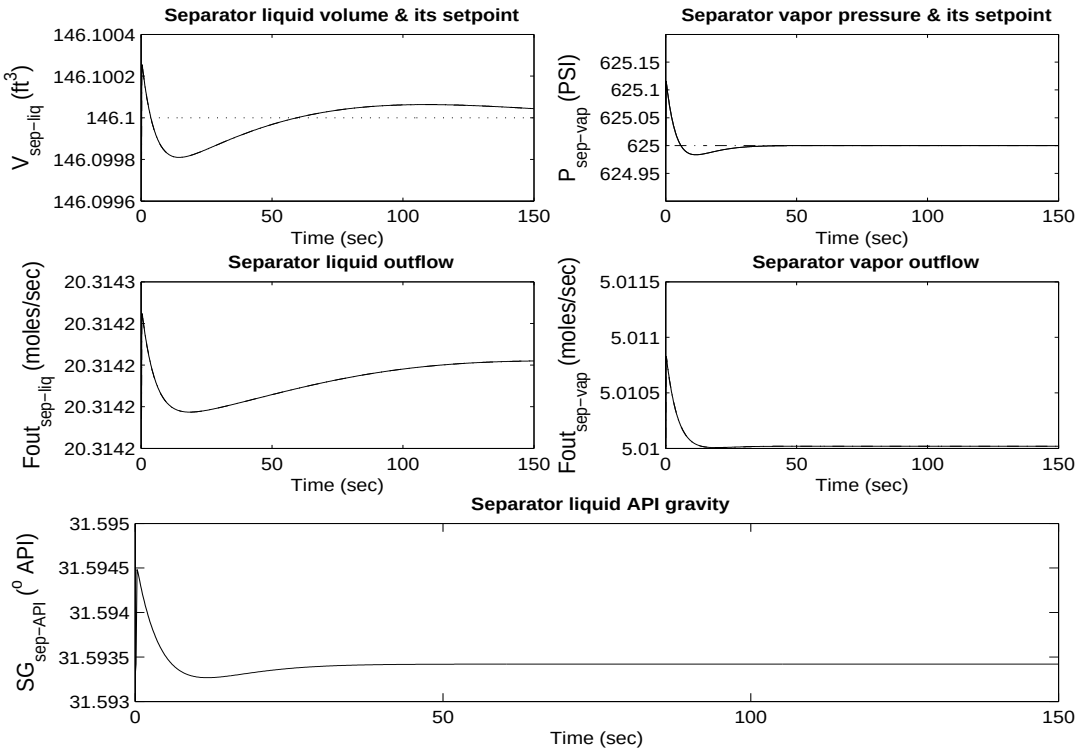

Figure A.23: -15% bias S_4 applied at t=60 sec. - Nominal Operating Point

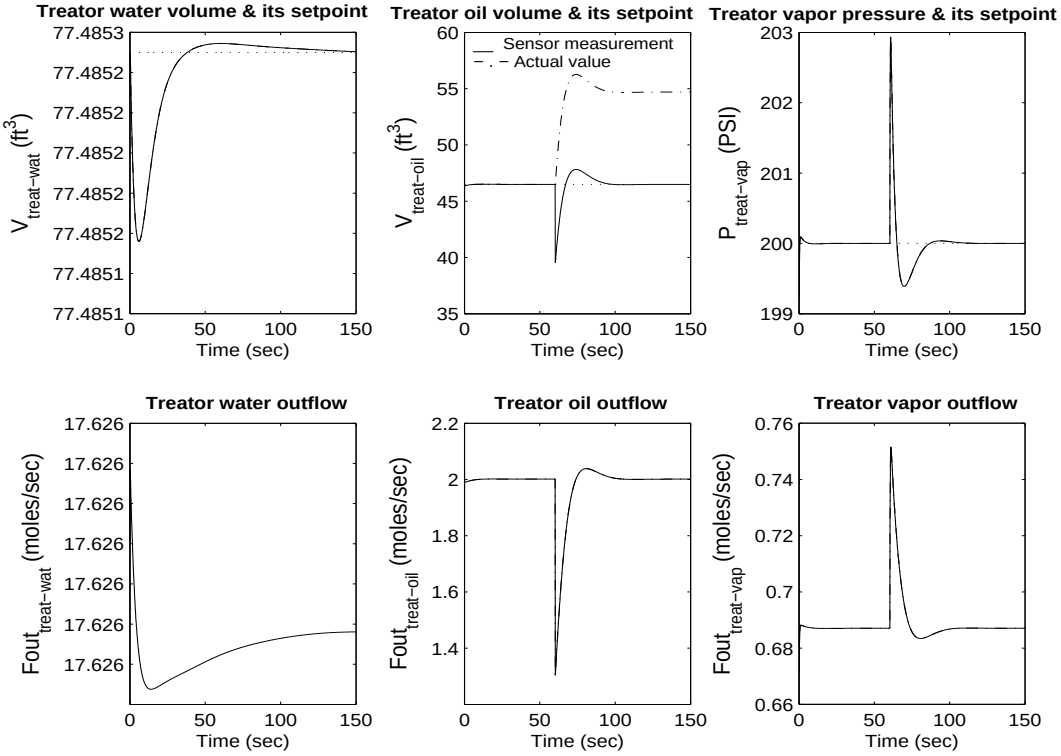

Figure A.24: -15% bias S_4 applied at t=60 sec. - Nominal Operating Point

Appendix B

JCSTR Dynamic Model

In this JCSTR, see figure B.1, the tank inlet stream is received from another process unit and there is a heat transfer fluid circulating through the jacket to heat the fluid in the tank. The objective is to control the temperature and the volume inside the tank by varying the jacket inlet valve flow rate (the temperature control or TC loop) and tank outlet valve flow rate (the level control or LC loop) respectively.

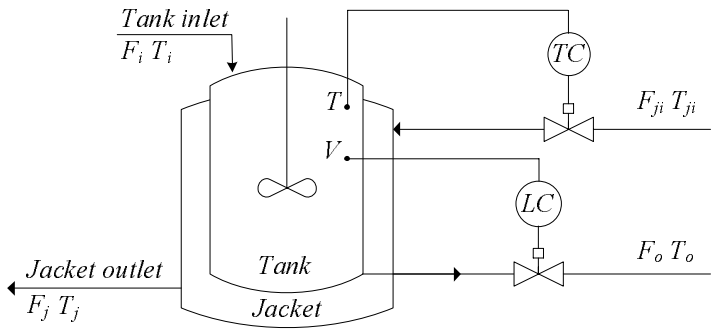

Figure B.1: Jacketed continuously stirred tank reactor

In order to derive the dynamic modeling equations of the tank and jacket temperatures, the following assumptions were made:

- Liquids have constant density and heat capacity.

- Mixing in both the tank and jacket are perfect.
- The tank inlet flow rate, jacket flow rate, tank inlet temperature and jacket inlet temperature may change (these are the inputs).
- The rate of heat transfer from the jacket to the tank is governed by the equation $Q = U\,A(T_j - T)$, where U is the overall heat transfer coefficient and A is the area for heat transfer and is given by equation $A = \frac{\pi D_r^2}{4} + \frac{4V}{D_r}$.

The following equations describe the ODE model for the JCSTR reactor [61] and the notation used is listed below.

$$\dot{V} = F_{in} - F_{out} \tag{B.1}$$

$$\dot{T} = \frac{F_{in}\,(T_{in} - T)}{V} + \frac{U\,A\,(T_j - T)}{V\,\rho\,C_p} \tag{B.2}$$

$$\dot{T_j} = \frac{F_{Jin}\,(T_{Jin} - T_j)}{V_j} + \frac{U\,A\,(T_j - T)}{V_j\,\rho\,C_p} \tag{B.3}$$

NOTATION

Subscripts

i	Inlet
j	Jacket
J_{in}	Jacket inlet

Variables

A	Area for heat transfer
D_r	Diameter of the reactor
C_p	Heat capacity (energy/mass*temp)
F	Volumetric flowrate (volume/time)
ρ	Density (mass/vol)
T	Temperature
Q	Rate of heat transfer (energy/time)
U	Heat transfer coefficient (energy/time*area*temp)
V	Volume

Parameter values

D_r	5 m
C_p	4.1868*1000 *(J/kg.K)*
F_{in}	0.1 (m^3/s)
ρ	997.95 (kg/m^3)
T_{in}	283 ^oK
F_{Jout}	0.15 (m^3/s)
V_j	9 m^3
T_{Jin}	419 ^oK
U	851.74 $(W/m^2.K)$

Operating point 1

V_o	180 m^3
T_o	306.5824 ^oK
T_j	377.2784 ^oK

Operating point 2

V_o	126 m^3
T_o	306.6937 ^oK
T_j	403.2040 ^oK

Appendix C

Fault detection during transients using the GPV technique

In this appendix the parity vector technique is implemented for an arbitrary second order system *G(s)* given by equation (C.1), with left coprime factors $\tilde{N}$ and $\tilde{D}$ given by equations (C.2) and (C.3).

$$G(s) = \frac{s+1}{(s+2)(s+10)} \tag{C.1}$$

$$\tilde{N}(s) = \frac{s+1}{(s+3)^2} \tag{C.2}$$

$$\tilde{D}(s) = \frac{(s+2)(s+10)}{(s+3)^2} \tag{C.3}$$

The purpose of this study is to clarify the $|GPV|$ behaviour during the transient following a small setpoint variation, for the ideal linear case. The fault scenario is described as follows: A unit step input is applied at t=0 sec followed by a +1% variation at t=10 sec. During its transient, a +10% bias sensor fault is applied at t=11 sec. It is observed in Fig. C.1 that the $|GPV|$ increases from its $|GPV|_{ff}$ value of $4.918\times^{-13}$ to $1.458\times^{-6}$ at t=10.13 sec after the +1% input variation is

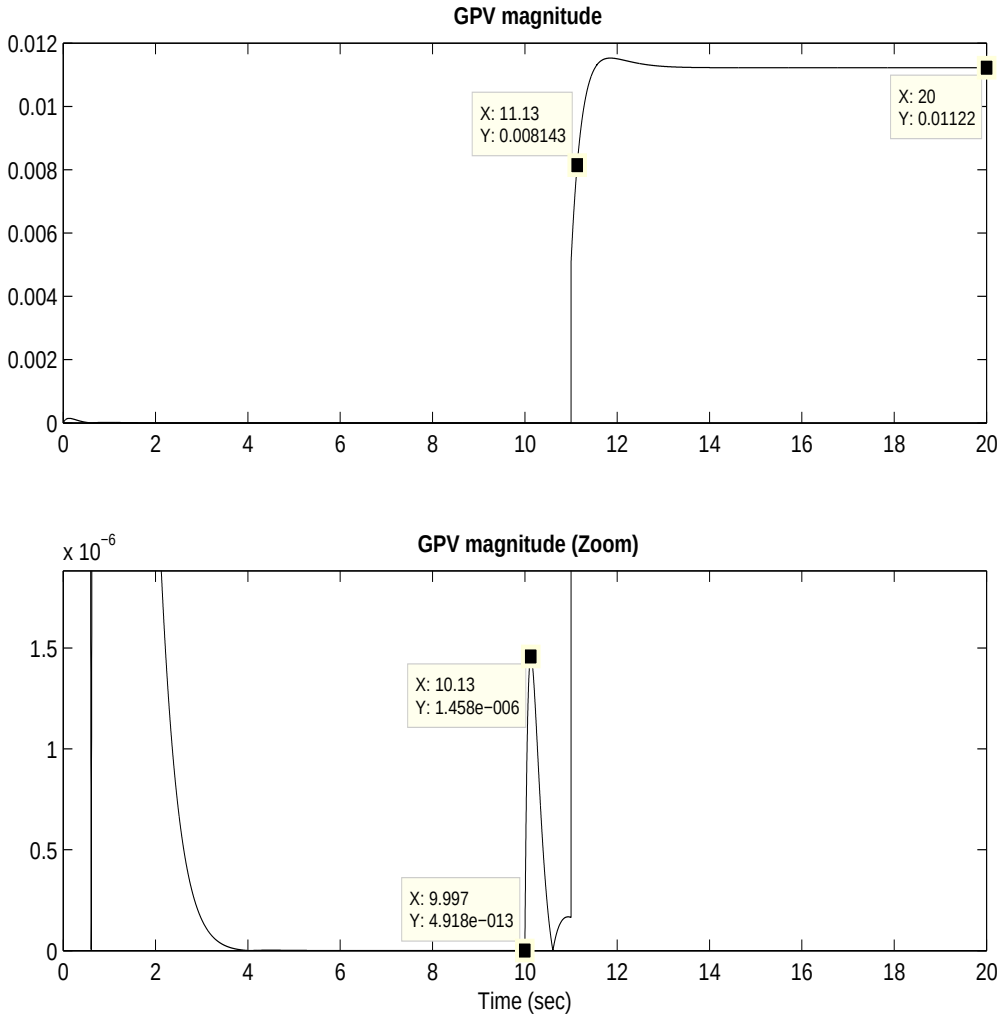

Figure C.1: Dynamic $|GPV|$ for a +10% sensor fault

applied. However, this $|GPV|$ is still very small as expected for a fault free scenario. Conversely, the GPV magnitude increment is more significant right after the fault happens at t=11.13 sec.

It can be established from this ideal case study that the GPV magnitude threshold can be easily selected within a wide range and still be able to correctly detect faults during transients without having false fault alarms.

Vita

Candidate's full name:	Maira Omana
Universities attended:	Universidad Industrial de Santander, Bachelor in Electrical Engineering, 2001
	University of New Brunswick, Master of Science in Electrical Engineering, 2006
Publications:	Robust Fault Detection and Isolation Using a Parity Equation Implementation of Directional Residuals, *Proc. APC 2005*
	Enhanced Sensor/Actuator Resolution and Robustness Analysis for FDI Using the Extended Generalized Parity Vector Technique, *Proc. ACC 2006*
	Fault Detection and Isolation Using the Generalized Parity Vector Technique in the Absence of an a Priori Mathematical Model, *Proc. CCA 2007*
	Fault Detection, Isolation and Accommodation Using the Generalized Parity Vector Technique, *Proc. IFACWC 2008*

Conference Presentations: IEEE Advanced Process Control Applications for Industry Workshop *(APC 2005)*, Vancouver, BC, May 9-11, 2005.

IEEE American Control Conference *(ACC 2006)*, Minneapolis, MN, June 14-16, 2006.

IEEE Conference on Control Applications *(CCA 2007)*, Singapore, October 1-3, 2007.

17^{th} IFAC World Congress *(IFACWC 2008)*, Seoul, Korea, July 7-11, 2008.