http://dx.doi.org/10.12785/ugric/2025006

An Embedded Fault Detection, Isolation, and Accommodation System in a Model Predictive Controller for an Industrial Process

Bassam Mohamed Alhamad
Department of Chemical Engineering
University of Bahrain
Isa Town, Bahrain
balhamad@uob.edu.bh

Fadhel Abbas Hasan
Department of Chemical Engineering
University of Bahrain
Isa Town, Bahrain
fadhel.abbas2002@gmail.com

Sayed Yousif AbdulJalil
Department of Chemical Engineering
University of Bahrain
Isa Town, Bahrain
Yousif.sayed49@gmail.com

Muath Ahmed Abubaker
Department of Chemical Engineering
University of Bahrain
Isa Town, Bahrain
Moathahmed003@gmail.com

Abstract*— Fault detection, isolation, and accommodation (FDIA) systems play a critical role in ensuring the smooth operation of complex industrial processes with no faults. However, traditional model-based approaches face challenges in capturing and maintaining the complexity of modern systems. This paper presents a data-driven alternative for FDI and fault-tolerant control (FTC) systems that overcome these limitations. This paper tackles the fault identification and detection in the Shell heavy oil fractionator using two control approaches, the Model Predictive Control (MPC) and Proportional-Integral (PI) controller. Two types of fault behaviors (drift and bias faults) are applied to the validated model that was validated with historical data collected during normal process operation. To indicate potential faults in the measurement system, the Squared Prediction Error (SPE) is calculated to observe the possible faults. While introducing the two faults, the constructed FDI unit showed successful detection of the fault using PCA. For identification, the fault is isolated and identified through SPE calculations, where the top endpoint y1 clearly showed a high spike of 240 and 18 compared to the other outputs (for drift fault and bias fault, respectively). Finally, fault-tolerant control is constructed and applied to compensate for the fault behaviors introduced. The two FTC techniques used are measurement reconstruction and measurement replacement in both MPC and PI control algorithms. The two techniques showed great results in compensating faults, where the fault compensation for bias fault occurred at time 905 min after introducing the fault and around 25-30 minutes after introducing the fault measurement as drift fault. The results demonstrate the system's adaptability and versatility in real-world industrial settings. When combined with PI and MPC controllers, the FDI system exhibits robust performance, providing valuable insights into its capabilities. By leveraging data-driven methodologies and assessing their performance in a simulated environment, this paper paves the way for more effective FDI systems in complex industrial processes.

Keywords*— Fault Detection and Isolation (FDI), Fault Tolerant Control (FTC), Model Predictive Control (MPC), Principal Component Analysis (PCA), Squared Prediction Error (SPE), drift and bias faults, industrial process monitoring

I. Introduction

The Shell Control problem conducted by [1] in 1987 was a pivotal event in the field of process control, aimed at addressing challenges in the oil and gas industry. The problem focused on bridging the gap between theoretical advancements and practical implementation of control strategies. Participants discussed topics such as model-based control, multivariable control, and optimization while exchanging knowledge and experiences through case studies [1].

In oil refineries, heavy oil fractionators are essential because they separate crude oil into distinct product draws by efficiently chilling the mixed-phase oil supply. It plays a significant role in processing and separating hydrocarbons' complex mixtures that in turn creates heavy oil. In industrial processes, there are several fractionators working together, each specialized in fractionating products from different draws, that are frequently observed in operation. However, this complex multi-input/multi-output (MIMO) system is vulnerable to measurement errors that can have a significant impact with further procedures in the refining process. Thus, keeping precise measurements is essential to guaranteeing operational effectiveness and averting possible losses throughout the refining process.

Fig. 1 illustrates the Shell Heavy Oil Fractionator presented by Prett and Morari in 1987 at the Shell process control workshop [1]. As shown in Fig. 1, there are seven output variables (top end point y1, side end point y2, top temperature y3, upper reflux temperature y4, side draw temperature y5, intermediate reflux temperature y6, bottom reflux temperature y7), three input variables (top draw flow rate, u1; side draw flow rate, u2; bottom reflux head transfer rate, u3; intermediate reflux heat transfer rate, l2; upper reflux heat transfer rate, l1), and two measured disturbance variables (intermediate reflux heat transfer rate, l2; upper reflux heat transfer rate, l1). This process is modeled using a set of linear models, first-order transfer functions, that can satisfy the dynamic behavior of the fractionator shown in Table I. [2]

TABLE I. Shell Heavy Oil Fractionator Model Transfer Functions

	Outputs				
	Top draw flow rate	**Side draw flow rate, u2**	**Bottom reflux head transfer rate, u3**	**Intermediate reflux heat transfer rate, l2**	**Upper reflux heat transfer rate, l1**
Heavy oil fractionator top end point, y1	$\frac{4.05e^{-27s}}{50s+1}$	$\frac{1.77e^{-28s}}{60s+1}$	$\frac{5.88e^{-27s}}{50s+1}$	$\frac{1.20e^{-27s}}{45s+1}$	$\frac{1.44e^{-27s}}{40s+1}$

	Outputs				
	Top draw flow rate	**Side draw flow rate, u2**	**Bottom reflux head transfer rate, u3**	**Intermediate reflux heat transfer rate, l2**	**Upper reflux heat transfer rate, l1**
Heavy oil fractionator side end point, y2	$\frac{5.39e^{-18s}}{50s+1}$	$\frac{5.72e^{-14s}}{60s+1}$	$\frac{6.90e^{-15s}}{40s+1}$	$\frac{1.52e^{-15s}}{25s+1}$	$\frac{1.83e^{-15s}}{20s+1}$
Top temperature, y3	$\frac{3.66e^{-2s}}{9s+1}$	$\frac{1.65e^{-20s}}{30s+1}$	$\frac{5.53e^{-2s}}{40s+1}$	$\frac{1.16e^{-0s}}{11s+1}$	$\frac{1.27e^{-0s}}{6s+1}$
Upper reflux temperature, y4	$\frac{5.92e^{-11s}}{12s+1}$	$\frac{2.54e^{-12s}}{27s+1}$	$\frac{8.10e^{-2s}}{20s+1}$	$\frac{1.73e^{-0s}}{2s+1}$	$\frac{1.26e^{-0s}}{22s+1}$
Side draw temperature, y5	$\frac{4.13e^{-5s}}{8s+1}$	$\frac{2.38e^{-7s}}{19s+1}$	$\frac{6.23e^{-2s}}{10s+1}$	$\frac{1.31e^{-0s}}{19s+1}$	$\frac{1.26e^{-0s}}{22s+1}$
Intermediate reflux temperature, y6	$\frac{4.06e^{-20s}}{13s+1}$	$\frac{4.18e^{-4s}}{33s+1}$	$\frac{6.53e^{-1s}}{9s+1}$	$\frac{1.19e^{-0s}}{19s+1}$	$\frac{1.17e^{-0s}}{32s+1}$
Bottom reflux temperature, y7	$\frac{4.38e^{-20s}}{33s+1}$	$\frac{4.42e^{-22s}}{44s+1}$	$\frac{7.20e^{-0s}}{19s+1}$	$\frac{1.14e^{-0s}}{27s+1}$	$\frac{1.26e^{-0s}}{32s+1}$

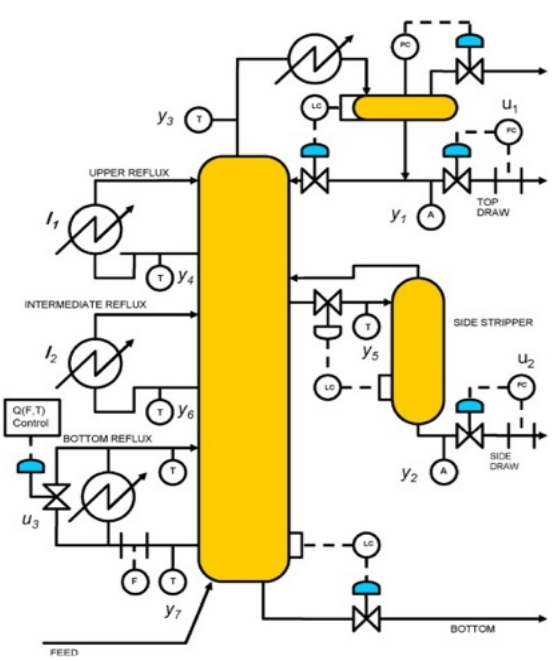

Fig. 1. The Shell heavy oil fractionator. [2]

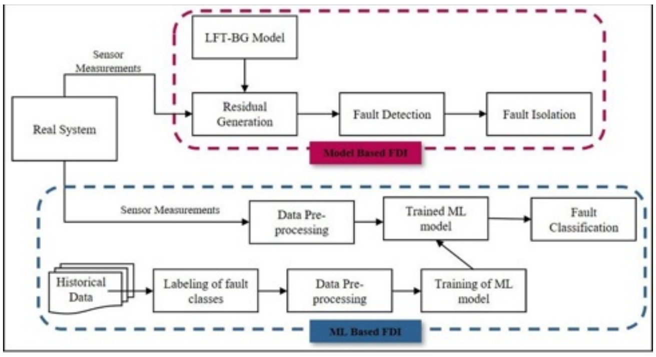

Fig. 2. Model-based FDI vs. Data-based FDI.

A. *Aim of the paper*

The aim of this paper is to design a reliable FDI & FTC system to detect and accommodate the shell heavy oil fractionator faults, plus implement process control strategies such as PID & MPC. First, it is crucial to design the control loops that are the base of the control process for the plant. The FDI unit was then constructed using the training data with and without the fault tests. The FDI unit should also send out the faulty value in a different route (i.e., to the supervisory unit) for compensation to occur. Finally, to accommodate the faults temporarily until the faulty instrument or part of the process is addressed, the FTC or a supervisory unit is designed so that when a fault is detected, the system manipulates this fault temporarily.

B. *Motivation*

Fault diagnostics has been a crucial aspect of process plant supervision for decades, and most recently it has been eye-catching when combined with process accommodators or compensators to optimize the process performance and prevent process instability. The fault detection, isolation, and accommodation system is generally an online system that could be implemented in various industrial processes. The Shell Heavy Oil Fractionator serves as a great example in demonstrating a real-life industrial process that has various factors that could eventually become faulty or fail in operation. The system designed in this process detects the fault using statistical analysis, isolates the faulty value, then temporarily compensates it using optimization techniques.

II. Literature Review

A. *Fault detection and isolation*

In recent years, various industrial companies have tried reducing operational costs and enhancing safety in complex control systems, leading to increased research in the field of fault detection and isolation (FDI). There are various methods that have been proposed for FDI, which can be categorized into two main groups: model-based and data-driven methods [2], [3], [4], [5], [6].

Model-based methods assume prior knowledge of the mathematical model describing the system dynamics. Kalman filtering is a well-known example of a model-based method that has been used in various applications. However, limitations of these methods, such as the difficulty in accurately capturing the complexities of real-world systems using first-principal equations. As the process becomes more complex, obtaining an accurate system model for FDI becomes much harder [7].

Data-based models provide an alternate solution for cases when the model-based approach to fault detection and isolation (FDI) becomes harder to obtain. Data-based models rely on large archives of process data that are available in many real-world applications, and these can be used to build data-based models. These models are trained with input-output data sets and are designed to handle non-linear systems. [3][4], [7].

Principal Component Analysis (PCA) is a prevalent method in multivariate data analysis, where it converts correlated variables into a reduced collection of uncorrelated principal components. Principal Component Analysis (PCA) is utilized across several domains and frequently serves as a first phase in the examination of extensive datasets [5], [6], [8]. The primary aim of PCA is:

- Derive the salient information from the data table.
- Reduce the data set's size by retaining only the essential information.
- Simplify the characterization of the data set.
- Examine the composition of the observations and the variables.
- Compress the data by decreasing the dimensionality while preserving the majority of the information. This approach is employed in picture compression.

To achieve these objectives, PCA computes new variables termed principal components, which are linear combinations of the original variables. To calculate the primary components, the subsequent procedures may be employed.

- Standardize the data: This involves standardizing the dataset by removing the mean and dividing by the normal deviation for each variable.
- Calculate the covariance matrix: Compute the covariance matrix of the normalized data. The covariance matrix illustrates the interrelationships and variances among various variables in the dataset.
- Determine the eigenvectors and eigenvalues: Calculate the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors denote the main components, whereas the eigenvalues indicate the variance accounted for by each principal component.
- Sort the eigenvectors. Arrange the eigenvectors in descending order according to their associated eigenvalues. This stage aids in identifying the primary components that account for the most variance in the data.
- Select the preferred quantity of major components: Ascertain the quantity of principal components to preserve based on the proportion of variance elucidated. In most applications, the primary components are selected based on the substantial proportion of overall variation accounted for; for instance, the leading two components encompass up to 80% or 90% [5], [6].

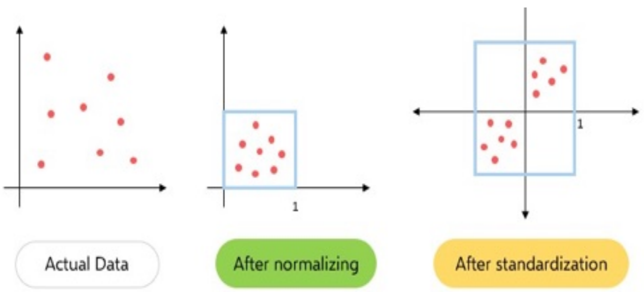

Fig. 3. Model for Data Standardization.

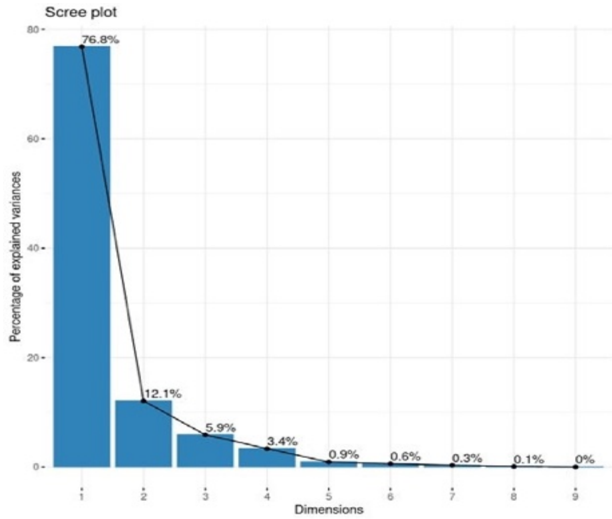

Fig. 4. Chart depicting explained variance.

Interpreting PCA findings necessitates comprehending the information encapsulated by the principal components and its correlation to the original variables. Consequently, two primary methodologies must be evaluated to have a deeper understanding of the outcomes of the principal component analysis. The first is to calculate the loadings that denote to the correlations between the original variables and the main components. Each major component possesses an associated set of loadings that signify the magnitude and orientation of the link between the variables and the component. It indicates the extent to which each variable contributes to the primary components.

The second step is to calculate the scores that denote to the changed values of the original data projected onto the major components. Every observation in the dataset is allocated a score for each main component. These scores denote the location of an observation inside the reduced-dimensional space established by the principal components [9], [10], [11], [12], [13].

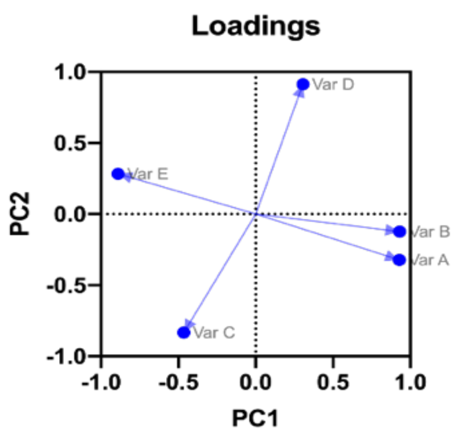

Fig. 5. Principal Component Analysis Loadings.

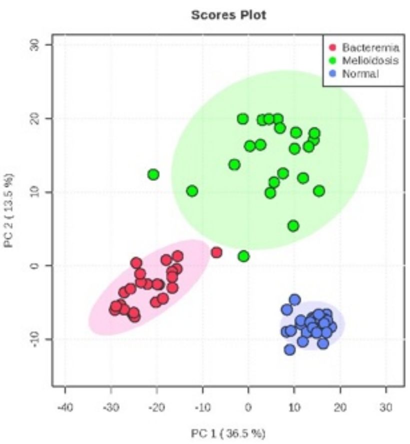

Fig. 6. Plot of PCA Scores.

B. Fault detection and isolation

Industrial processes are intrinsically susceptible to failures, mostly stemming from sensors, actuators, the process itself, or the controller. The controller's control action may substantially decline due to the fault, even if the system operates correctly, resulting in a situation where the fault escalates to the extent that the control loop cannot achieve stability, leading to undesirable outputs or, in the worst-case scenario, catastrophic system failure [6].

Due to the public's need for the uninterrupted operation of automated systems and their apprehensions regarding possible system breakdowns, guaranteeing the dependability of these systems is essential. A technique to do this is by the application of fault-tolerant control (FTC). FTC is a discipline within control engineering that amalgamates many disciplines of knowledge to improve system dependability. The principal aim of FTC is to avert small faults from amplifying and culminating in complete system failure. By enacting this policy, the FTC reduces the likelihood of production disruptions or safety risks stemming from system failures [2], [3].

A variety of approaches are available for the application of FTC in automated systems, including the following [2], [3]:

- Fault accommodation techniques aim to modify the system's control strategy or settings in response to detected issues. This may involve adjusting setpoints, restructuring control loops, or using adaptive control algorithms. The aim is to alter the system's behavior to reduce failures and maintain optimal performance.
- Reconfiguration involves modifying the system's design or structure in response to problems. This may entail shifting to supplementary components, rerouting signals, or modifying the system's operational mode. Reconfiguration solutions aim to maintain system functionality in the event of faults.
- The execution of redundancy involves the replication of critical components or subsystems inside the system. Redundancy enables a backup component or subsystem to take over in the case of a breakdown, ensuring continuous operation. Redundancy can be achieved at several levels, encompassing sensor redundancy and actuator redundancy. [5]

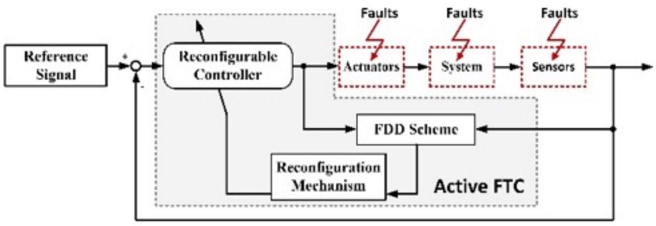

Fig. 7. Reconfiguration-based FTC system [1].

III. Methodology and System Design

The objective of controlling the fractionator is to maintain the desired values for the endpoints of the top-draw product (y1), side-draw product (y2), and bottom reflux temperature (y7). This is achieved by manipulating the flow rates of the top draw (u1), side draw (u2), and the heat transfer rate of the bottom reflux (u3). The heat transfer rate (u3) is further adjusted using a control loop that utilizes the hot steam flow rate as a control variable. Additionally, there are two measured disturbances in the system: the heat transfer rate of the upper reflux (l1) and the intermediate reflux (l2). These flows remove heat from the system and are subsequently reboiled in other sections of the plant.

The study first determines the most suitable control configuration for the PI (Proportional-Integral) controlled system. To achieve this, the relative gain array (RGA) was calculated, as shown in Table II, a method to analyze the interaction between control variables and process outputs. Based on the RGA matrix presented in Table II, the analysis recommends pairing the top draw product endpoint (y1) with the top draw product flow rate (u1) for control. Similarly, the side draw product endpoint (y2) should be controlled by the side draw product flow rate (u2), and the bottom reflux temperature (y7) by the bottom reflux heat transfer rate (u3).

A. Control Using PI

With the control structure defined based on the RGA analysis, the research moves to testing the control strategy. This is done within a MATLAB Simulink environment. Three separate PI controllers are added to the system, each dedicated to controlling one of the key process variables: y1 (top draw product endpoint), y2 (side draw product endpoint), and y7 (bottom reflux temperature). The tuning of the PI controllers utilizes the Internal Model Control (IMC) method. This approach helps determine the ideal settings for the PI controllers. The final values obtained through IMC tuning are presented in Table III.

TABLE II. RGA Matrix for the Process

RGA Matrix	**Values**		
	Top draw flowrate, u1	***Side draw flowrate, u2***	***Bottom reflux head transfer rate, u3***
Top endpoint, y1	2.0757	-0.7289	-0.3468
Side endpoint, y2	3.4242	0.9343	-3.3585
Bottom reflux temperature, y7	-4.4999	0.7946	4.7503

TABLE III. PI TUNING PARAMETERS

PI Parameters	K values	τ_i
Controller 1	0.4	72
Controller 2	0.8	89
Controller 3	1.8	22

B. Control Using MPC

The MPC-based control strategy is subsequently developed in the MATLAB platform. Real-time optimization of a cost function is conducted using quadratic programming to solve the constrained optimization problem, while analytical approaches are employed for the unconstrained scenarios. As certain process states cannot be directly measured, a state estimator is necessary for the controller. For this study, the default state estimation method employed by the MPC toolbox is the Kalman filter.

The MPC parameters are adjusted based on the dynamics of the simulated process. The prediction horizon (p) is set to a duration that allows for an effective response to most situations encountered in the simulated process. Given the varying dead times (ranging from 0 to 28 minutes) and time constants (ranging from 6 to 60 minutes) in the process, the prediction horizon is established as 120 minutes. Similarly, the control horizon (m) is set to a longer duration. However, to account for computational time, the control horizon length is defined as 40 minutes. The sample time for both the process and the MPC is configured as 1 minute.

The weights assigned to the controlled variables and manipulated variables (MVs) are adjusted to fine-tune the performance and behavior of the MPC. The controlled variables y1, y2, and y7 are assigned weights of 45. The MV weights are set to 0.01. To mitigate the impact of noise and abrupt changes in output values, the weights assigned to the MV rates (u1, u2, and u3) are established as 1000. These weight value configurations result in more stable and reliable control actions compared to lower weight values. A summary of the MPC parameter values is provided in Table IV.

C. Designing FDI and FTC

In fault detection and isolation (FDI) systems, Principal Component Analysis (PCA) helps identify process issues. By analyzing historical data under normal conditions, PCA builds a model of expected behavior. During operation, new data is compared to this model. Deviations exceeding a threshold, calculated using squared prediction error, signal potential faults. PCA can even assist in fault location by analyzing which variables contribute most to the deviation [14].

TABLE IV. MPC PARAMETERS

Parameter	Values
Prediction Horizon, p	120
Control Horizon, m	40
Weights, CV [y1 y2 y7]	[45 45 45]
Weights, MV	[0.01 0.01 0.01]
Weights, MV rates	[1000 1000 1000]

To determine whether the data significantly differs from normal operating data, the T^2 method is used as expressed in (1).

$$T^2=\sum_{j=1}^{a}\frac{T_j^2}{\sigma_j^2} \tag{1}$$

If the calculated value exceeds the threshold, it suggests a potential fault.

D. Squared Prediction Error (SPE)

The Squared Prediction Error (SPE) defines the error by comparing the actual values of data with its reconstruction of the reduced data by PCA as expressed in (2).

$$SPE=\sum_{j=1}^{m}(X_j - X_{reconstructed,j})^2 \tag{2}$$

SPE Contribution helps isolate which specific features are responsible for the error and to calculate this using (3):

$$\text{SPE Contribution}=(X_j - X_{reconstructed})^2 \tag{1}$$

SPE Contribution Analysis helps identify the cause of a fault by breaking the total Squared Prediction Error (SPE) into parts contributed by each feature. It compares the real value of a feature to its predicted (reconstructed) value. The feature with the largest difference is flagged as the main contributor to the fault. This method makes it easier to pinpoint which specific variable in the data is behaving abnormally, allowing precise fault isolation.

E. Thresholds

The thresholds for T-squared and Squared Prediction Error (SPE) statistics are critical for fault detection in multivariate analysis. These thresholds are set to determine when the process deviates significantly from normal behavior, indicating a potential fault. The T-squared threshold is determined based on the confidence level, the number of principal components, and the number of samples in the dataset. This threshold establishes the critical limit for the T-squared statistic, beyond which the data is considered abnormal. Similarly, the Squared Prediction Error threshold is calculated using the confidence level and the number of variables in the dataset, setting the limit for the reconstruction error. These thresholds depend on the specific characteristics of the dataset, including the number of samples and variables, and can be adjusted based on the desired sensitivity of the fault detection system. A higher confidence level results in a stricter threshold, making the system more sensitive to smaller deviations. In contrast, a lower confidence level leads to a less sensitive system, detecting fewer faults. By adjusting these thresholds, the fault detection system can be tailored to the unique requirements of the monitored process.

When the system calculates the Squared Prediction Error (SPE) and Hotelling's T^2 statistic for each PCA model, it compares these values to pre-defined limits. If an SPE value surpasses its corresponding limit, a fault is flagged on the associated variable (y1, y2, or y7) with the highest exceeding SPE. It's important to note that Hotelling's T^2 statistic acts more as a confirmation and comparison tool, not the primary trigger for declaring a fault.

Oil refineries experience two main types of sensor and analyzer faults: abrupt changes (bias faults) and gradual shifts (drift faults). Contamination in the analyzer sample usually causes bias faults. Drift faults, on the other hand, can arise from a slow buildup of materials within sensors, analyzers, or sample lines.

The test data includes 1600 minutes of simulated process data with these measurement faults incorporated. The simulation injects a positive bias fault of 0.5 into the top product quality variable (y1) at the 900-minute mark, lasting until the 1100-minute mark. Another fault, a positive drift fault, is also introduced into y1 at the 900-minute mark. This drift fault gradually increases to a maximum value of 0.5 by the 1100-minute mark. It's important to note that 0.5 represents the highest allowable limit for the final quality of the top product (y1).

To train the PCA-based fault detection method, a dataset of closed-loop process data without faults is used. This data incorporates information about both external influences (disturbance variables like upper and intermediate reflux heat duty) and the system's response (controlled variables like product end points and temperature). Additionally, it includes details on the control inputs (manipulated variables like flow rates and heat transfer) used to maintain the desired process state. By learning the normal relationships between these variables, the FDI method can identify deviations that might signal a fault in the system.

The fault detection method employs three separate PCA models, each focusing on a specific controlled variable (y1, y2, or y7) along with its corresponding manipulated variables (u1, u2, u3) and the disturbance variables (l1, l2). This structure (PCA1, PCA2, PCA7) allows the models to consider the impact of external process disturbances while identifying potential faults. To achieve this, the system calculates two metrics for each model: Squared Prediction Error (SPE) and Hotelling's T2 statistic, both with 95% confidence intervals. These metrics help identify deviations from normal behavior that might indicate a fault in the system.

Once a fault is successfully detected, located, and its severity is determined in the top product quality variable (y1), the system attempts to compensate for it. The study compares two compensation methods: measurement reconstruction using the PCA-based FDI system and measurement replacement. Both methods are tested under two control strategies: a traditional PI controller and a more advanced model predictive control (MPC) strategy.

Regarding designing FTC, the FTC (Fault Tolerant Control) is proposing a three-part system to address faults. The system has a detection element (FDI), a control element, and a supervisory element [12], [13], [15]. The supervisory element takes action to minimize the impact of the fault, using two possible methods: replacing faulty measurements or reconstructing them. This study focuses on these two methods for comparison with other control systems. Fig. 8 illustrates both approaches.

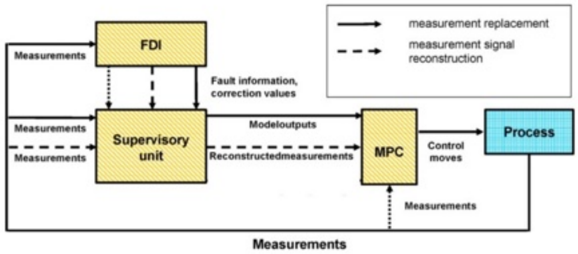

Fig. 8. FTC structure.

Measurement replacement is a strategy used in fault tolerance control (FTC) systems to address situations where sensor malfunctions provide unreliable data. This method relies on a mathematical model of the system's behavior, built using a technique called subspace identification (SID). The training data for this model consists of system inputs and corresponding correct measurements collected during normal operation. When a sensor fault is detected, the supervisory unit within the FTC system activates the model. This model, essentially acting as a virtual sensor, generates an estimate to replace the faulty measurement. The effectiveness of measurement replacement hinges on the quality of the training data used for SID. High-quality training data ensures the model accurately reflects the system's behavior, allowing it to provide reliable estimates and maintain control system operation even in the presence of sensor faults.

Measurement reconstruction aims to salvage the faulty measurement itself. It achieves this by calculating an iterative correction value that minimizes the squared prediction error (SPE). This correction value is derived using a PCA model built from a separate training dataset. The supervisory unit then applies this correction to the faulty measurement, essentially attempting to "fix" it. Fig. 8 illustrates this approach with a dashed line.

IV. Results and Discussions

The experiment examined the system's reaction to a setpoint alteration for the primary product quality (y1). At the 100-minute mark, the target end value (setpoint) of y1 was elevated by 0.4 units. Fig. 9 illustrates that the PI controller had a progressive response, ultimately attaining a steady adjustment and meeting the new objective. Nevertheless, the reaction was rather sluggish. The rising time for y1 to reach 0.4 was 400 minutes. The system required 600 minutes to completely stabilize at the new setpoint value, signifying the settling period. This alteration in the setpoint also influenced other measured parameters inside the process, indicating possible interconnections among various components of the system.

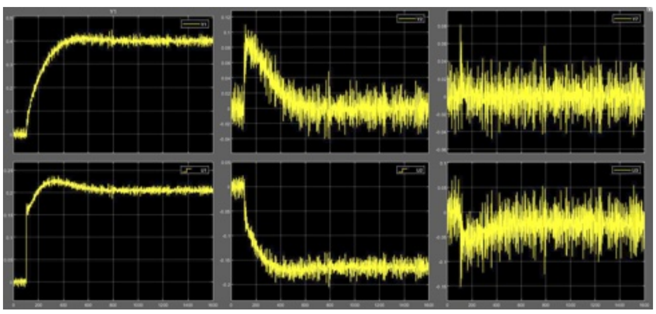

Fig. 9. Output of PID control.

The efficacy of the control system utilizing Model Predictive Control (MPC) was assessed. Fig. 10 depicts the reaction to a 0.4-unit step adjustment in the intended ultimate quality (setpoint) of the top product (y1) at the 100-minute interval. The MPC exhibited robust performance. It attained the new setpoint of 0.4 for the highest product quality (y1) in a markedly reduced duration relative to the PI controller (achieved at 264 minutes). Moreover, as seen in Fig. 10, the alteration of the setpoint for y1 affected other measured variables in the process. Nonetheless, these modifications were seamless, and the other variables reverted to their setpoints with reasonable rapidity, signifying effective overall regulation by the MPC system. The rise time, defined as the duration for y1 to attain 90% of its new setpoint, is 200 minutes in this instance.

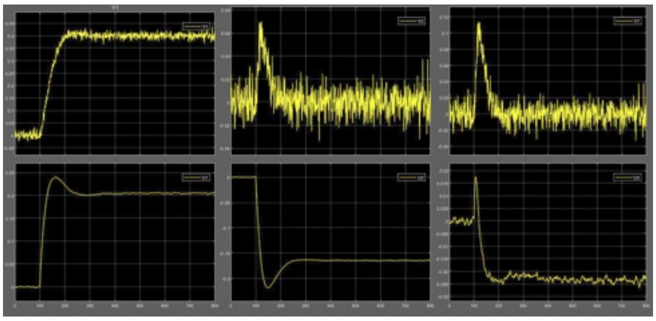

Fig. 10. Output of the Model Predictive Control (MPC).

A. Proportional-Integral Controller vs. Model Predictive Controller

Although PI controllers have fulfilled their function, optimally calibrated Model Predictive Control (MPC) presents a more advantageous option, clearly outperforming in both precision and reaction time (Fig. 9 & Fig. 10). MPC attains the target setpoint markedly more rapidly, resulting in expedited process modifications and possibly enhanced results. MPC has additional advantages beyond speed: more straightforward tuning methods than PI controllers, wider application across various processes, and automated dead time correction due to its integrated model. This paradigm facilitates intrinsic multivariable control, essential for intricate processes with interrelated variables. Moreover, MPC can accommodate measurable disruptions, self-regulating to sustain optimal performance. Nonetheless, the precision of the MPC's internal model is crucial; a more precise model results in enhanced overall system performance, as imperfections might constrain the MPC's full capabilities.

B. Foreign Direct Investment Unit Outcomes

The Fault Analyzer Toolbox in MATLAB Simulink was employed to replicate authentic operating circumstances and evaluate the system's resilience. This toolkit serves as an effective instrument for introducing controlled defects into the system. Two prevalent fault types had been identified: bias and drift. Bias faults denote a persistent deviation in the measured value, resembling scenarios such as sensor calibration discrepancies. Drift faults, conversely, represent a steady alteration in the observed value over time, maybe resulting from sensor deterioration or environmental influences. These controlled failures are included, as shown in Fig. 11 and Fig. 12, to assess the system's capacity to detect, isolate, and potentially correct for interruptions, therefore assuring seamless functioning under non-ideal situations.

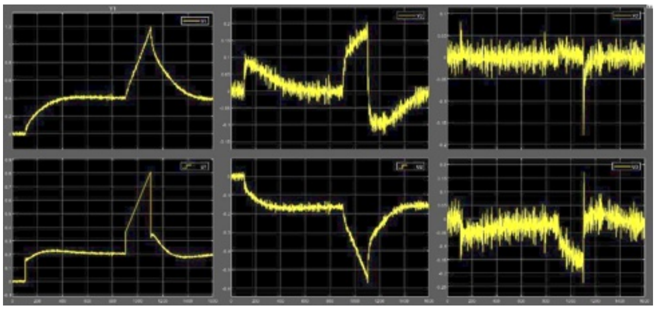

Fig. 11. PID Fault Output (Drift Fault)

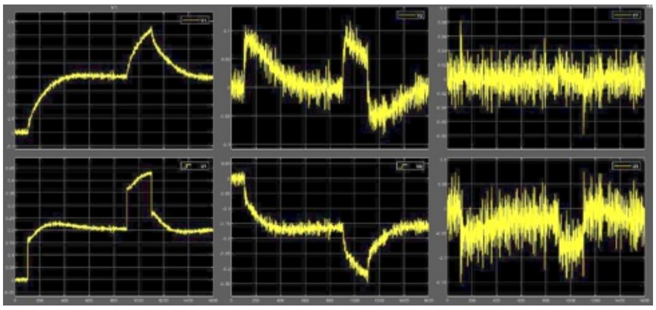

Fig. 12. PID Fault Output (Bias Fault)

C. T^2 and SPE Outcomes

The research delineates a failure detection and isolation (FDI) system grounded in Principal Component Analysis (PCA). This approach uses two primary metrics: the Squared Prediction Error (SPE) and Hotelling's T^2 statistic. The SPE functions as the principal instrument for detecting anomalies in the controlled variables. When an SPE value is over a certain threshold, a defect is issued. An iterative procedure utilizing SPE data is employed to identify the precise fault type. The Hotelling's T^2 index serves as a supplementary verification instrument.

The evaluation of bias and drift faults demonstrated the efficacy of the PCA-based Fault Detection and Isolation system. Both SPE and Hotelling's T^2 identified the bias problem at the 905-minute mark, indicating a little delay relative to the actual introduction of the fault. SPE detected the issue at 925 minutes for both PI and MPC control techniques in the context of the drift fault. Hotelling's T^2, however, exhibited a delay, identifying the defect just at 929 minutes. These data illustrate the markedly superior detection rate of SPE relative to Hotelling's T^2, despite equivalent confidence levels. Fig. 13 and Fig. 14 display the SPE and Hotelling's T^2 indices for drift and bias faults, respectively.

Ultimately, the algorithm isolates the defect by pinpointing the variable with the largest SPE value that surpasses the detection threshold. When many variables exhibit SPEs over the threshold, the variable with the highest value is designated as defective. The graphic illustrates that the variable y1 possesses the greatest SPE value. This identifies y1 as the defective variable, facilitating focused troubleshooting and maintenance activities.

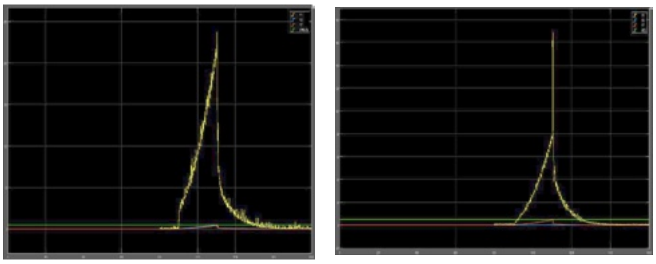

Fig. 13. FDI SPE and T2 hoteling output for drift fault.

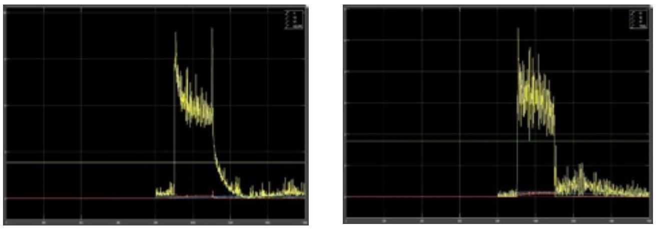

Fig. 14. Output of FDI SPE and T^2 hoteling for bias defect.

D. *FTC Unit Outcomes*

Following the examination of fault detection and isolation in both fault behaviors (bias fault and drift fault), the subsequent phase is the fault compensation procedure to achieve the fault-tolerant control (FTC) of the system [2], [3], [16], [17]. The methods employed for fault compensation include measurement reconstruction and measurement replacement. Measurement reconstruction is fundamentally a technique aimed at deriving the optimum value by optimizing the SPE value at a given moment. Measurement replacement adopts an alternative methodology, depending on the process model to substitute the erroneous value with the ideal one. Both measurement reconstruction and measurement replacement eliminate the impact of faults and deliver a more precise measurement for control reasons, as well as to mitigate the problem until the defective process unit is rectified [2], [16].

E. *Reconstruction of Measurements with Bias Fault*

The system effectively identified the bias fault simultaneously (905 minutes) for both Proportional-Integral (PI) and Model Predictive Control (MPC) techniques. Fault correction was promptly attained with minimal interference to other measures, illustrating the system's efficacy in managing bias faults. Fig. 15 and Fig. 16 illustrate the measurement reconstruction in a bias fault PCA-FDI unit [12], [15].

F. *Reconstruction of Measurements using Drift Fault*

Additionally examined the system's reaction to drift faults (progressively altering defects). The system required a little longer duration to identify and rectify the error in comparison to bias faults. In both PI and MPC instances, the issue was detected and rectified within an acceptable timeframe of around 25-30 minutes following its impact on measurements. This underscores the system's capacity to manage various error types while preserving process stability. Fig. 17 and Fig. 18 illustrate the measurement reconstruction within a drift fault.

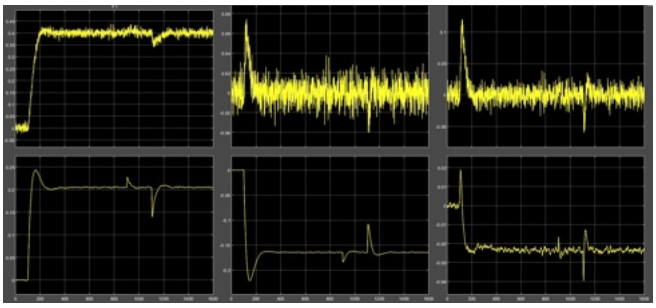

Fig. 15. Reconstruction of measurements for model predictive control with fault-tolerant control in the presence of bias faults.

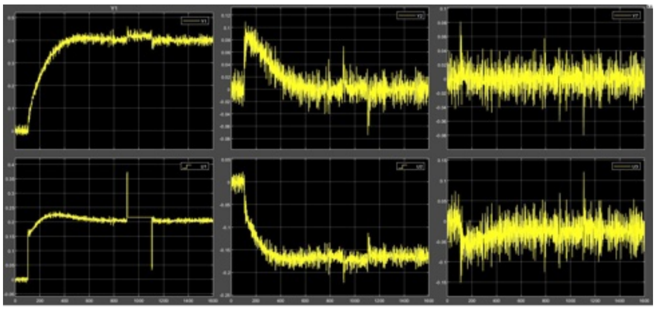

Fig. 16. Reconstruction of measurements for proportional-integral fault-tolerant control in response to bias faults.

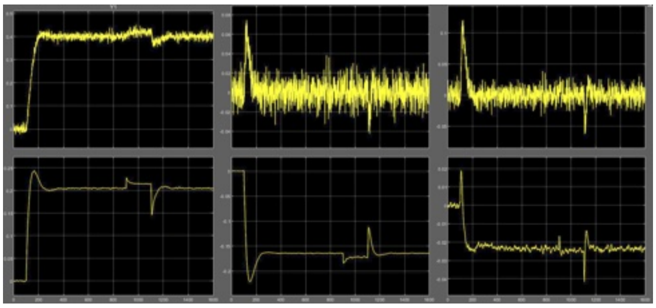

Fig. 17. Reconstruction of measurements for model predictive control with fault-tolerant control addressing drift faults.

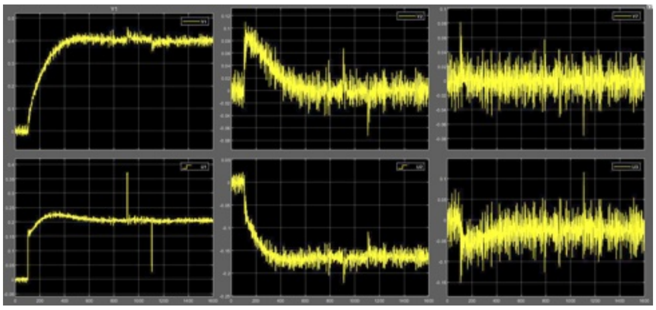

Fig. 18. Reconstruction of measurements for proportional-integral-based fault-tolerant control addressing drift faults.

G. *Measurement Substitution with Bias Error*

The system efficiently addressed bias errors throughout the measurement replacement process, swiftly attaining fault compensation while reducing disruptions to other measurements. The system's capability in addressing bias errors is evidenced by the swiftness of fault rectification. The substitution of measurements in a bias fault PCA-FDI unit is depicted in Fig. 19 and Fig. 20.

H. Substitution of Measurement Due to Drift Fault

The system affected by the drift faults was analyzed. In this instance, the system required longer time to identify and rectify the issue in comparison to bias faults. In both PI and MPC cases, the fault was detected and rectified, taking 25 to 30 minutes after introducing the faults in measurements. This illustrates the system's capacity to handle diverse fault types while maintaining process stability. Fig. 21 and Fig. 22 depict the measurement substitution in a drift fault PCA-FDI unit.

The experiments validated the efficacy of the PCA-based FDI utilizing measurement reconstruction and replacement for fault compensation in the process. The system effectively identified and rectified both bias and drift problems with negligible effect on other readings. Ultimately, the examination of measurement reconstruction and replacement impacts on both PI and MPC controllers reveals largely comparable outcomes, with minor discrepancies in the MPC controller attributed to its more aggressive actions relative to the PI controller, resulting in slight variations in efficiency regarding measurement reconstruction or replacement.

Further analysis has been made to make sure that the system could operate in various circumstances. This extended analysis is made by incorporating the process disturbances into the system prior to the FDI and FTC analysis. Fig. 23 shows the process output with PI control after incorporating the process disturbances I1 and I2 into the three outputs y1, y2, and y7.

As Fig. 23 shows, the process is successfully stable; however, the process outputs/inputs have exceeded the process constraints, which indicates that the system in reality is undesirable. Since MATLAB does not count process constraints unless they are implemented, the process stabilizes without process constraints under consideration.

For the FDI unit, the new system's training data is changed since the process output changed, and doing so, Fig. 24 shows that the faults are very much like the faults in the previous original analysis from the 900th min to the 1100th min.

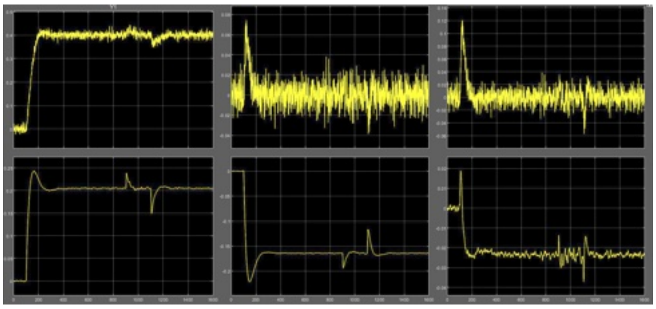

Fig. 19. Measurement substitution for model predictive control with fault-tolerant control addressing bias faults.

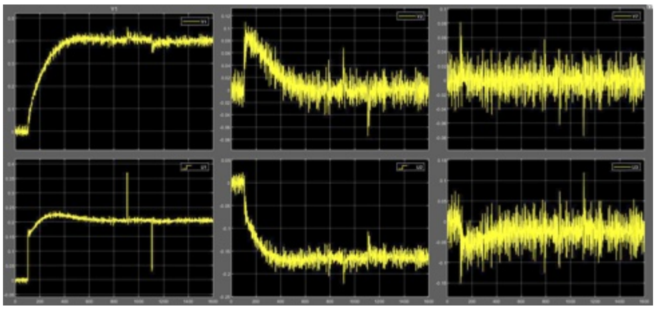

Fig. 20. Substitution of measurement for proportional-integral-based fault-tolerant control in the context of bias faults.

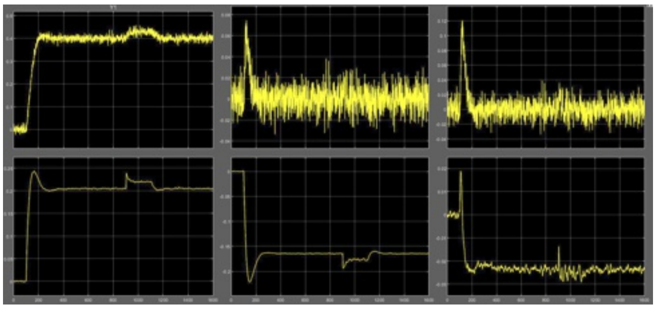

Fig. 21. Measurement substitution for model predictive control with fault-tolerant control addressing drift faults.

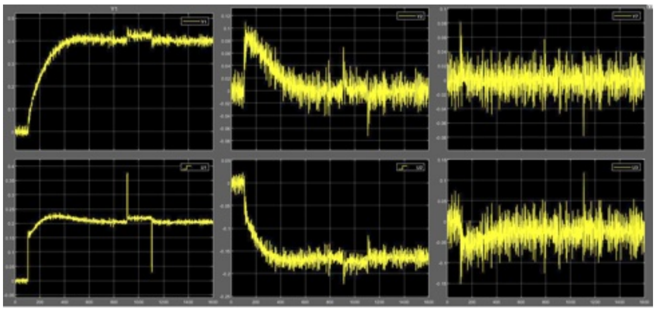

Fig. 22. Measurement substitution for proportional-integral-based fault-tolerant control addressing drift faults.

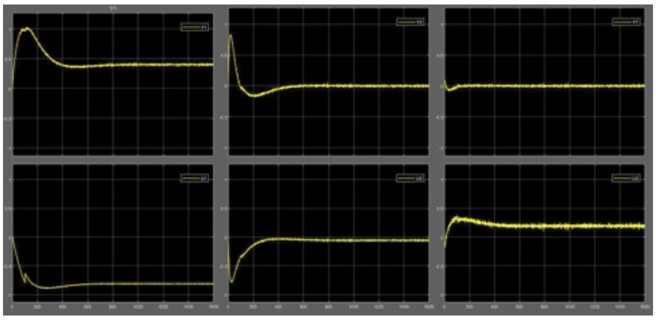

Fig. 23. PI output with disturbance.

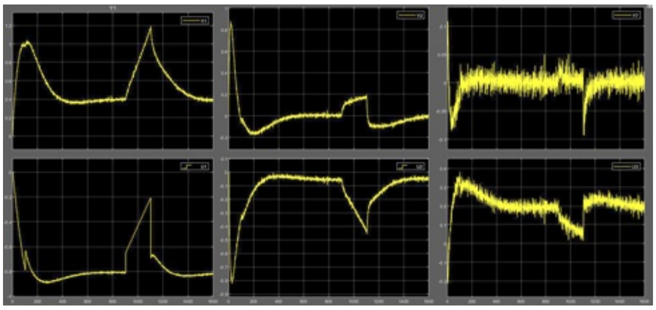

Fig. 24. PI output with disturbance after drift fault.

Fig. 25 shows the SPE and T^2 charts, where it has been observed that the results are also very similar to the original analysis.

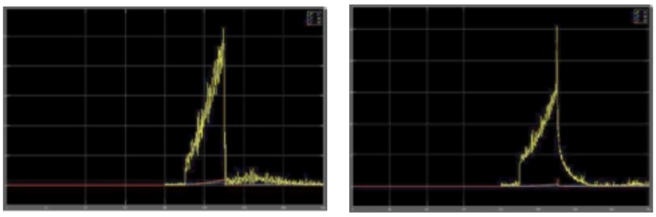

Fig. 25. SPE and T^2 charts for PI fault with disturbance.

Finally, the FTC unit is implemented with measurement replacement and measurement reconstruction, where it is observed that the results replicate the original analysis, as in Fig. 26 and Fig. 27.

The FDI and FTC units designed in this paper give similar results even if there are disturbances, and that will happen if the training data injected into the system takes into account everything the system could encounter other than faults (i.e., if the training data does not consider disturbances and the implemented disturbances, the FDI and FTC unit will consider it as faults).

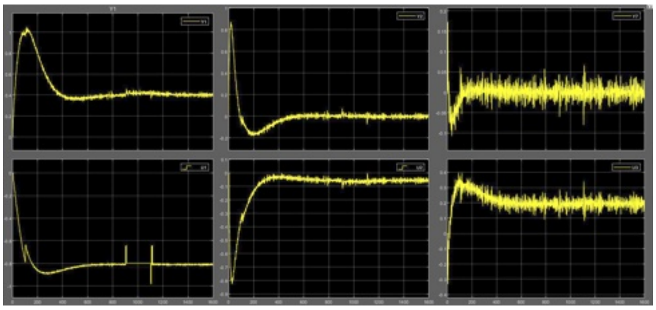

Fig. 26. Measurement reconstruction for PI with disturbance-based FTC unit.

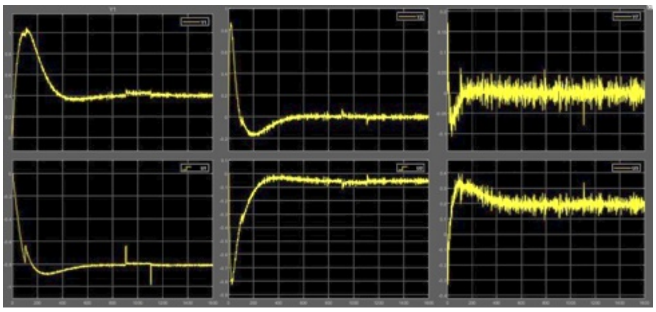

Fig. 27. Measurement replacement for PI with disturbance-based FTC unit.

V. Conclusions

This study investigates a data-driven methodology for issue detection and tolerance in a simulated heavy oil fractionation process (Shell Control issue). The methodology uses Principal Component Analysis (PCA) to analyze sensor data and identify abnormalities that may indicate problems. Two fault-tolerant control systems were developed, utilizing PCA-based fault detection and isolation. These systems employed two fault compensation strategies: measurement reconstruction and substitution. The effectiveness of these systems and different control methods (MPC compared to PI controllers) was then evaluated. The results demonstrate the effectiveness of the proposed approaches. Both measurement reconstruction and replacement significantly alleviated bias and drift issues in the simulated readings. PCA employing Squared Prediction Error (SPE) showed effectiveness in defect detection, whereas Hotelling's T^2 served as a comparable instrument. The FTC systems demonstrated effective performance with both MPC and PI controllers, showcasing their ability to detect and correct problems regardless of the control mechanism used. The research suggests that MPC offers greater flexibility in variable management, making it potentially more suitable for complex systems. The research presents a feasible data-driven approach for attaining fault tolerance in industrial processes.

This study demonstrates the effectiveness of the PCA-based FTC system in mitigating bias and drift faults; nonetheless, its future success relies on enhanced resilience. This necessitates an analysis of the system's response to a wider range of perturbations, including modifications to the process equipment, malfunctions in control actuators, significant deviations from the initial process model, and the ability to differentiate between sensor failures and external disturbances impacting the process. Implementing a self-tuning method for the controller would represent a substantial advancement. This may entail online adaptation algorithms that modify controller settings in response to identified defects, allowing the system to sustain optimal process performance under unexpected conditions.

References

[1] D. M. Prett and M. Morari, *The Shell Process Control Workshop*. Butterworth-Heinemann, 1987.

[2] J. J. Gertler, "Fault-diagnosis and fault-tolerant-control in industrial processes and manufacturing systems," in *Fault Diagnosis and Fault-Tolerant Control*, Springer, 2016. doi: 10.1007/978-3-319-45735-2_5.

[3] S. Narasimhan and R. Rengaswamy, "Actuator fault monitoring and fault tolerant control in distillation processes," *Int. J. Autom. Comput.*, vol. 13, no. 3, pp. 219–234, 2016, doi: 10.1007/s11633-016-1037-8.

[4] R. Isermann, *Fault Detection and Diagnosis in Industrial Systems*. Springer, 2006. doi: 10.1007/978-1-4471-0347-9.

[5] S. Lee, J. Lee, and J. Lee, "A review on fault detection and process diagnostics in industrial processes," *Processes*, vol. 8, no. 9, p. 1123, 2020, doi: 10.3390/pr8091123.

[6] S. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review on fault detection and diagnosis techniques: Basics, classification, and applications," *Artif. Intell. Rev.*, 2020, doi: 10.1007/s10462-020-09934-2.

[7] W. Sun, A. R. C. Paiva, P. Xu, A. Sundaram, and R. D. Braatz, "Fault detection and identification using Bayesian recurrent neural networks," *arXiv preprint*, 2019. [Online]. Available: https://arxiv.org/abs/1911.04386.

[8] M. Tommasi and F. Weinschelbaum, "Centralization vs. decentralization: A principal-agent analysis," *J. Public Econ. Theory*, vol. 9, no. 2, pp. 369–389, 2007.

[9] R. Isermann, "Fault detection with principal component analysis (PCA)," in *Fault-Diagnosis Systems*, Springer, pp. 267–278, 2006, doi: 10.1007/3-540-30368-5_13.

[10] Z. Morrison, B. P. Russo, and R. Kamalapurkar, "Fault detection via occupation kernel principal component analysis," *arXiv preprint*, 2023. [Online]. Available: https://arxiv.org/abs/2303.11138.

[11] A. Rehab, I. Ali, and M. N. Fors, "Bearings fault detection using hidden Markov models and principal component analysis enhanced features," *arXiv preprint*, 2021. [Online]. Available: https://arxiv.org/abs/2104.10519.

[12] J. Gertler and J. Cao, "PCA-based process diagnosis in the presence of control," in *Proc. 5th IFAC Symp. Fault Detect. Superv. Saf. Tech. Processes (SAFEPROCESS)*, 2003, pp. 849–854.

[13] J. Zhang, H. Chen, S. Chen, and X. Hong, "An improved mixture of probabilistic PCA for nonlinear data-driven process monitoring," *arXiv preprint*, 2020. [Online]. Available: https://arxiv.org/abs/2012.06830.

[14] J. V. Kresta, J. F. MacGregor, and T. E. Marlin, "Multivariate statistical monitoring of process operating performance," *Can. J. Chem. Eng.*, pp. 35–47, 1991.

[15] R. Dunia and S. Qin, "Joint diagnosis of process and sensor faults using PCA," *Control Eng. Pract.*, vol. 6, pp. 457–469, 1998.

[16] J. J. Gertler, "Fault-diagnosis and fault-tolerant-control in industrial processes and manufacturing systems," in *Fault Diagnosis and Fault-Tolerant Control*, Springer, 2016. doi: 10.1007/978-3-319-45735-2_5.

[17] S. Narasimhan and R. Rengaswamy, "Actuator fault monitoring and fault tolerant control in distillation processes," *Int. J. Autom. Comput.*, vol. 13, no. 3, pp. 219–234, 2016, doi: 10.1007/s11633-016-1037-8.