Recursive Polynomial Approach to Fault Prediction in Technical Process Control System

[1*]Ugo Donald Chukwuma; [2]Ugochukwu Jane Ijoemoa, [3]Okafor C.J.

[1]Department of Mathematics, Faculty of Physical Sciences, ESUT

[2]Department of Pharmaceutical Microbiology and Biotechnology, Faculty of Pharmaceutical Sciences, ESUT

[3]Department of Electrical and Electronic Engineering, Enugu State University of Science and Technology, Enugu State

[1*]Corresponding Author

DOI: https://doi.org/10.51584/IJRIAS.2025.100800005

Received: 15 July 2025; Accepted: 26 July 2025; Published: 29 August 2025

ABSTRACT

This paper presents an intelligent Process Control System (PCS) that integrates a neuro-logic solver with a recursive polynomial estimator for fault prediction and integrity monitoring in safety-critical environments. The system combines Artificial Neural Networks (ANNs) with recursive polynomial regression to enhance reliability and compliance with IEC 61069 and IEC 61508 standards. Using operational data from a distillation unit, a neural network was trained as the safety logic solver, while an ARMAX-based recursive estimator modeled plant dynamics for early fault detection. Real-time simulation and control were conducted in MATLAB and Simulink. The system achieved a Mean Squared Error of 2.98×10^{-9}, a regression coefficient of 0.9978, and a PFD of 9.00×10^{-2}, corresponding to Safety Integrity Level 4. Results show robust fault tolerance, accurate forecasting, and adaptive control, making the approach well-suited for industrial applications demanding high safety and availability. The limitation of the work is that the system's performance depends on the quality and representativeness of the training data, and its computational complexity may pose challenges for real-time deployment in resource-constrained environments.

Keywords: Process Control System (PCS); Recursive Polynomial Estimation; Neuro Logic Solver; Artificial Neural Network; Safety Integrity Level (SIL)

INTRODUCTION

Basic elementary science defines production as the creation of goods and services for satisfaction of human wants. This concept can be dated back to the early men, who employed mechanized tools for the production of goods especially in the agricultural sector. This early idea of mechanized tool for production, extended to other manufacturing of simple mechanical system which expanded more in the mid 70's to more complex mechanical systems powered with steam and used for the optimization of manufacturing process. This era was tagged the first industrial revolution (Anthony, 2018). Beginning from the late 80's new sources of power like electricity, oil and gas began to make its way into the economic industry. Due to the efficiency of these energy sources, better mechanical systems manufactured like the internal combustion engine, began to reach its full potential. This era also resulted to other important inventions like automobiles, air planes, and early communication systems among others. This was the second industrial revolution stage (Marc et al., 2019).

In the 20th century, the third industrial revolution began to make way with advancement in the field of science and technology. According to the institute of entrepreneurship development (IED, 2019), this era brought forth the rise of telecommunications, electronics, computers and programming. As a result, intelligent system like the Programmable Logic Controllers (PLC) and robots find its way into the industries which helped gave rise to the first era of industrial automation (IED, 2019).

According to (Mallikarjun, 2017), industrial automation is the process of operating industrial machineries and other related equipment with digital logic control devices and reduction of human intervention via manual command process and decision making. This process has recorded great success in the industrial sector such as maximizing production, reducing the cost of labour, reducing risk reduction factor of technical process, among others.

Today, Industrial automation has taken over technical production process lines in manufacturing industries due to its potential for high production efficiency, high quality products, and high expectation in reliability of products, increased production volume, etc. However, to fully reach this potential, certain issues resulting from the dynamics nature of process operations and process planning like process design configuration challenges, uncertain elements in process control like the rate of reaction, endalphies, heat transfer etc, high implementation cost, faults, security, programming complexities among others have to be considered as constraints and hence solve. These constraints can be aligned with the three layers of industrial automation which are the fields or process design layer, the process control layer and the supervisory layer (Realpears, 2017).

The field layer involves the configurations and specifications for sensors, actuators, controllers, transducers and other process design equipment. The process layer ensures the logical control and right amount of process flow from the various field devices used in the process design while the supervision layer is the remote monitoring of the process control using internet of things, communication protocols and Supervisory Data Acquisition And Control (SCADA) systems. However, the process layer is the coordinator of the other layers and is the key section for optimal technical process efficiency. This layer is responsible for the control of the Basic Process Control System (BPCS) configured with the process design components like the plant reactor, sensors and controllers aforementioned.

In this BPCS the process control ensures stability of the plant and regulated flow rate during the technical process, however the high level of nonlinearities attributed with process plants like the continuous stir tank reactor during thermodynamic process and mass transfer often overwhelmed the process controller and hence results to system failure. According to (Inyama and Azubuike, 2015; Skogestad, 20011), the Programmable Logic Controller (PLC), dominates over 95% of the process industries automation settings and despite the huge application suffers many limitations identified in (Rahul and Rajesh, 2020; Obaid et al., 2018 and Mallikarjun et al., 2017; Fallahi and Azadi, 2011; Okafor et al., 2017) and hence not reliable guarantee optimal process operation. These technical limitations coupled with the aging of the process design components results to common cause problem and independent component problems which leads to system failure.

To address this problem, many methods such as complete replacement of the process design components, safety instrument systems, redundancy control system, emergency short down systems, etc was recommended (Tesfabirhan et al., 2017; Ashok et al., 2017), however these solutions are very cost intensive, time consuming and hence not affordable by many industries.

To solve this problem, this research proposes a fault estimation algorithm which will be able to monitor and detect step ahead fault in the technical process and then control the fault and at the same time notify the operator using machine learning-based logic solver algorithm and recursive polynomial function. This fault estimator when developed and integrated with the PLC, will go a long way to improve the reliability of the process control and optimize technical process operations.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

RESEARCH METHODOLOGY

The methodology used for the research was guided by the International Electrochemical Commission (IEC) standard IEC61069 for the modeling of process control systems. This methodology allowed for good process design workflow for optimal process control system performance. The research methods used were characterization for data collection, development of the fault prediction algorithms using recursive polynomial function and neural network algorithms and the optimization of the BPCS via system integration. The new system was implemented with simulation, then the performance evaluated. The new system was then deployed on the testbed and evaluated to analyze the performance in real world scenario.

Risk Assessment Test

To perform the risk assessment test of the system, the two methods obtainable are the inductive and deductive approaches respectively. The former employed structural analysis like fault tree to determine the PFD of the PCS considering causes of failure, degree of failure, time of failure, while the later used self-defining equations approved by the IEC 61508 to extract parameters such as reliability, failure rate, and then determine the safety integrity level of the system. The two methods were employed for the risk assessment test as they both complement each other in identifying all probabilities of failure in the PCS. The fault tree provided more insights on the causes of system failure using Boolean logic, while the later provided data of the system PFD over a given period of time per hour, when the component is in operational condition.

The Fault Analysis

This fault tree employed three major logical components which are the OR gate, AND gate and OR- VOTE gate respectively as shown in Table 1 for the development of the fault tree model.

Table 1: Logic table for fault tree analysis

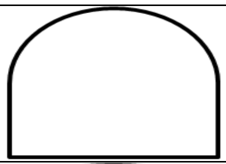
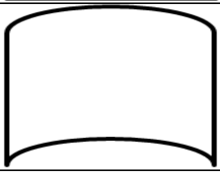
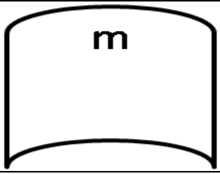
Logic gate	Type	Description
	AND	True if all inputs are true
	OR	True if any input is true
	OR- VOTE	True if m input are true

Table 1 present the logic components used to develop the fault tree. Unlike the IEC standard which focused on the operability of the components, the fault tree analysis focused on the failure rate of the components and then determines the average PFD of the PCS. The model of the fault tree for the PCS was developed identifying the main components of the PCS which are the sensors, the logic gate and solenoid valves and then used the logic gate as in the Table 1 to perform the fault tree analysis and identify all possibilities of failure based on independent input to the system. The fault tree model of the PCS system is presented in the Figure 1;

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

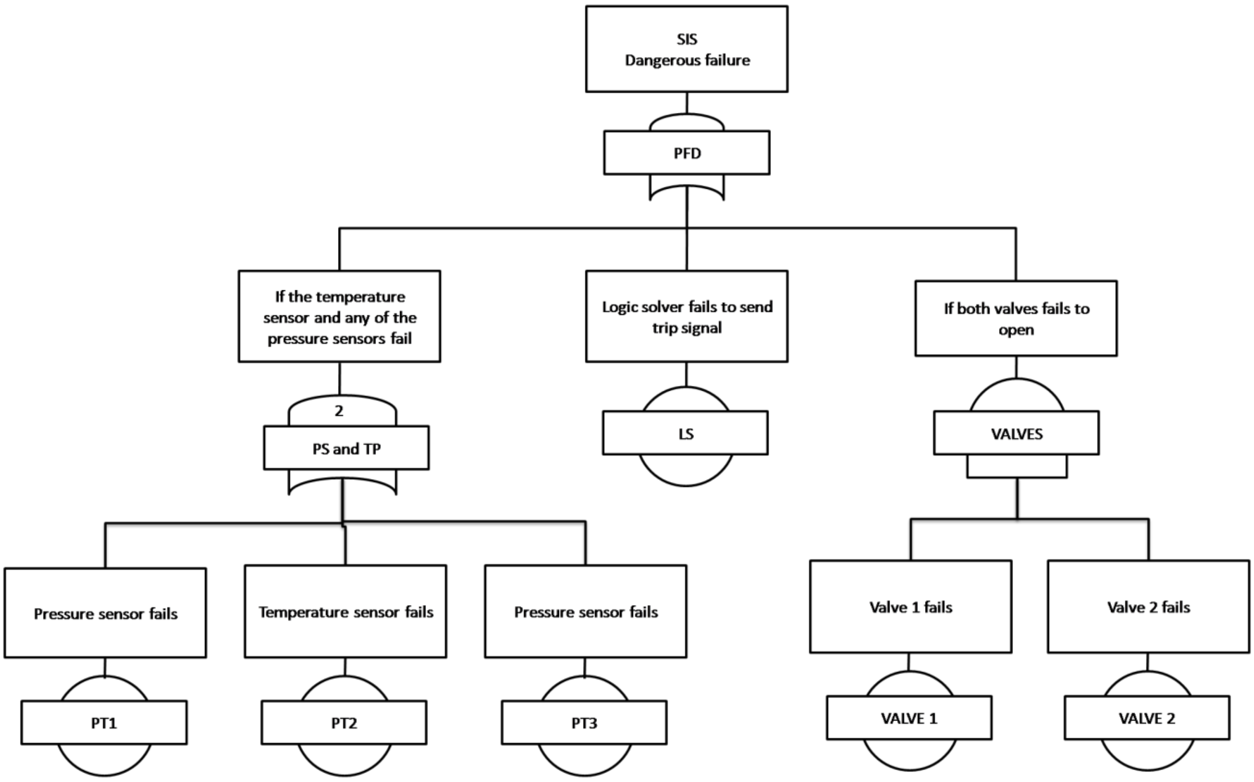


Figure 1: Model of the PCS fault tree

The fault tree model in Figure 1 presents the risk analysis of the PCS considering the three critical components which are the sensors, logic solver and valves. The PFD for each component was identified using the logic gate in Table 1. The risk assessment of the sensors used the OR-VOTE GATE to identify two sensors out of the three sensors unable to send signal to the logic solver as dangerous. The logic solver when unable to send failure signal from the sensor is also dangerous (most dangerous as it has no redundancy). Then the AND gate was used to present the logical operation of the valves shown if both cannot close as dangerous. In the fault tree modelled in Figure 1 the component failures are treated independently, however there are cases where the same problem can cause more than one component to fail at the same time. For instance, poor condition of the components, common maintenance strategy that is incorrect (Joseph, 2003) and these can all have great impact on the PFD of the PCS and has to be accounted for to ensure good result for PFD. To handle these issues, an explicit event in the fault tree was used to represent the Common Cause Problem (CCP) which presented an independent component failure and also a simultaneous component failure as shown in the fault tree model. From the assessment, it was deduced that the logic solver is the most critical component in the PCS with potential for dangerous failure as it has no redundancy when compared to failure of some sensors or control valves. However, despite the success of this fault tree analysis, it is possible that the researcher did not identify all failures due to uncertainties, but to cover for this, the inductive risk assessment technique was also employed which considered key risk assessment parameters like availability of the components, reliability, Mean Time Between Failure (MTBF) and PFD to determine the safety integrity level of the PCS.

Method of Data Collection

Having successfully performed the risk assessment test on the PCS, the data was collected considering the PFD of the system components, the detected and undetected common cause failures, diagnostic coverage for each component for a period of 39 days. The data are reported in the next chapter and analysed considering the usage of safety integrity level and risk reduction factor according to the IEC standard. Another data of the fractional distillation plant was also collected from the case study containing attributes such as the temperature and pressure behaviour of the plants and was used later in the work for development of the machine learning based algorithm proposed.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

Machine Learning Based Logic Solver System

From the risk assessment test conducted, the PLC logic solver remains one of the most critical components of the PCS and it has no redundancy. This component is the coordinator of all control operation based on data collected from the sensors to ensure safety is achieved in the technical process and requires the most attention to reduce the failure probability to approximately zero. Other components like the final control elements which are the valves, the sensors such as the pressure and temperature sensors all have redundancy (if one fails, another can complement) and can operate in 1003 or 1002 modes respectively, however the logic solver which does not have redundancy is the most important components in the PCS need to guarantee functional safety.

The characterized test bed was modelled with PLC based logic solver, which was developed using Proportional and Integral (PI) mathematical functions, have the probability of early failure due to many technical problems such as it depends on open loop response as it cannot be trained Poor recovery rate during load disturbance, no improvement with time due to output oscillation, poor generalization property with set point change, module failure, bad network connection issues with the communication module, grounding integrity, electromagnetic interference, overheating, power outages, memory corruption, etc (Cory, 2013; Yuvraj, 2012).

To this end, machine learning based logic solver is developed using artificial neural network, activation function and training algorithm to reduce most of these technical problems attributed with the conventional PLC based logic solver and hence reduce failure probability to the minimum. The neural network model was developed using the interconnection of neurons, activation functions, training algorithms as shown in the Figure 2;

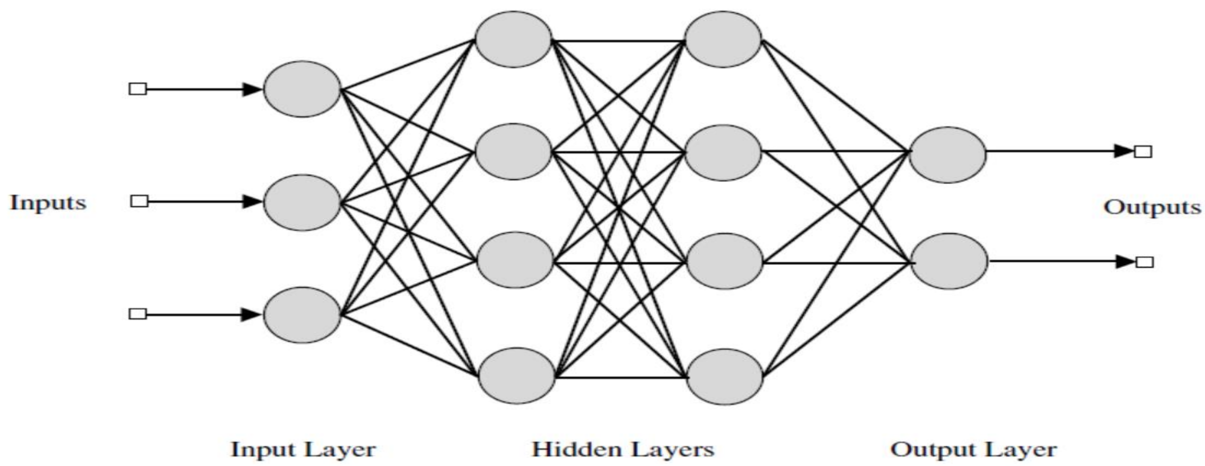


Figure 2: The neural network architectural model

The model showed how the neurons which have weights and bias was configured according to the input data class of the training set, the activation function and training algorithm to learn the distillation plant data collected and generate a neural network-based logic solver algorithm. The activation function used is the Tangent sign mode (tansig) activation function which enables the neurons to activate and also ensure data convergence between (-1 and 1). The training algorithm used in the study, is the Gradient descent back propagation type as it allows the neurons to learn, check its learning rate and feedback for adjustment and continuous learning until the least error is achieved as shown in the flowchart of Figure 3. This figure 3 presents the training algorithm used to train the neural network model in figure 2. To achieve this, the plant data was loaded into the neural network for configuration and training using the algorithm in Figure 3. The neural network training model which shows how the neural network identified the loaded plant data and then train the neurons with the training algorithm presented above is shown in Figure 4;

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

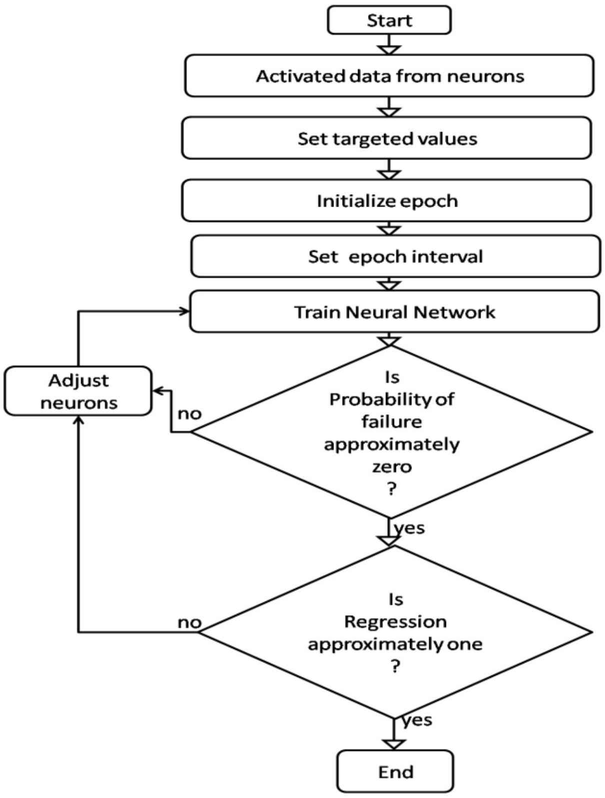


Figure 3: Back-propagation Algorithm

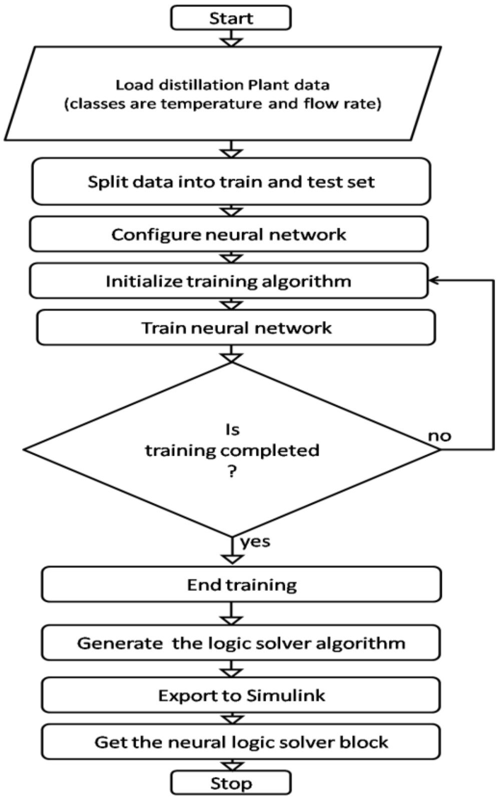


Figure 4: The neural network training model

Figure 4 present the flow chart of the neural network training process used in generating the neuro logic solver algorithm and hence the neural logic solver system Simulink, the data of the plant loaded into the neural network was used to configure the network and then train the neuron with the training algorithm to generate the neurologic solver algorithm. During the training, at each epoch the regression and training error was checked until least error is achieved and then the neuro logic solver algorithm developed as shown in the pseudocode below;

The Logic Solver Algorithm

Start
Load plant data
Configure neural network with Table 2
Initialize training algorithm
Train neural network
Check for training failure
If
Failure probability ≈ 0
Generate logic solver algorithm
Else
Back-propagation
Adjust neuron
Repeat step (4,5.6 and 7)
Generate logic solver algorithm
Else
Do (step 13) until step 8 is true
Generate neurologic solver algorithm
Generate the neurologic solver block
End if
End if
End

Table 2: The training parameters

Training Parameter	Assumed Value
Learning Rate	0.001
Number of Epochs	100
Batch Size	32
Activation Function	ReLU
Loss Function	Mean Squared Error (MSE)
Optimizer	Adam
Regularization Techniques	L2 Regularization (weight decay)

The training parameters in Table 2 present the neural network properties which values for input layer and hidden layers were inspired by the plant attributes (class in the training set collected), other values were standard neural network properties auto input by the neural network tool used for training. Figure 5 present the neuro-based PCS system which collect data from the temperature and pressure sensor and used to monitor the behaviour of the distillation plant for tank overflow when the process control logic solver fails. The PCS detects the problem and activates the control valves to stabilize the plant and prevent the problem. The system block diagram is presented in Figure 6;

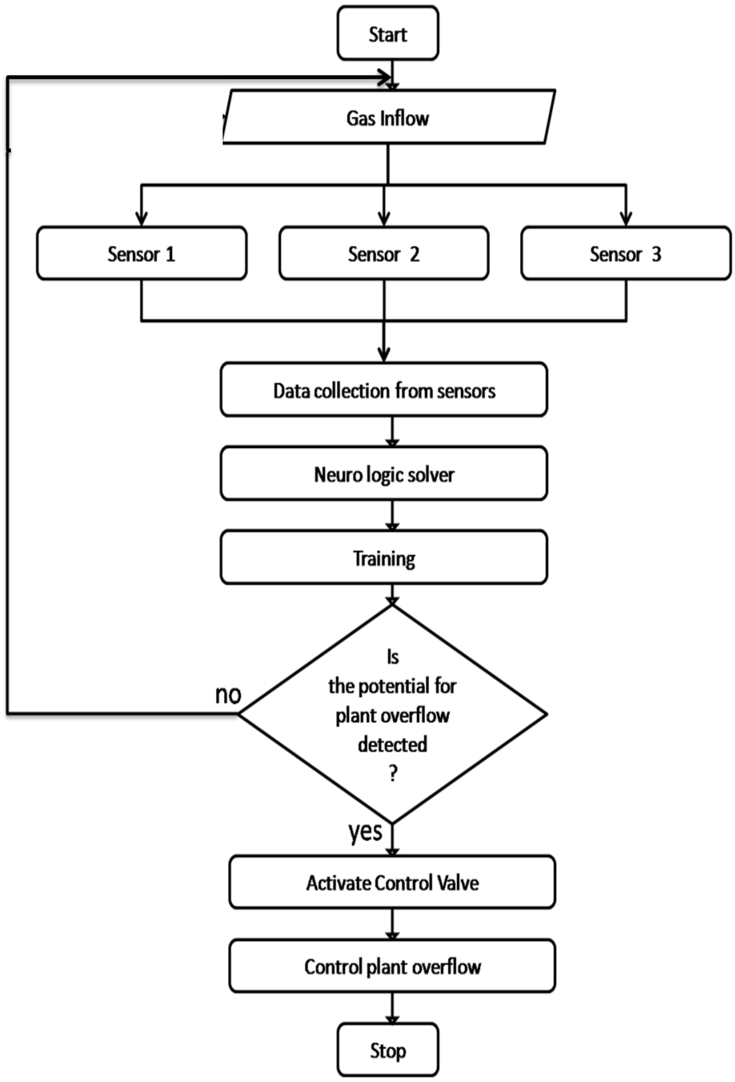


Figure 5: Neuro-based PCS operation

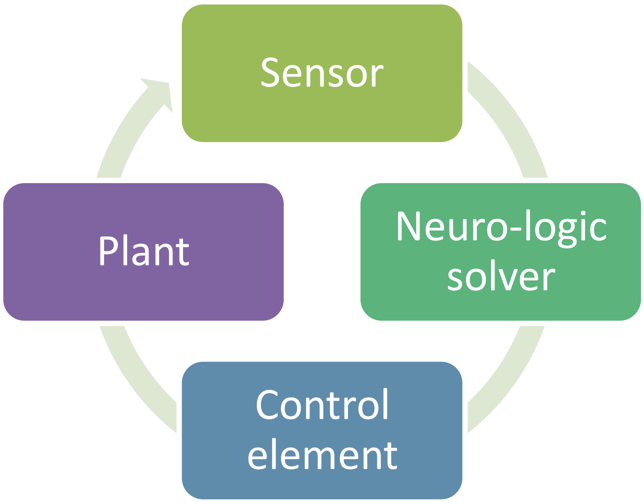


Figure 6: Block Diagram of the neuro-based PCS

The Figure 6 shows how the plant behaviour was collected by the sensors and fed forward to the neuro-based logic solver algorithm which trains the data to detect the distillation plant overflow problem and then activate the final control element which are the valves to control the plant behaviour.

Safety Integrity Algorithm for the Neuro PCS

The previous section developed a neurologic based PCS system to monitor the distillation plant overflow and maintain stability, however despite the high-level intelligence of the logic solver as it has been trained with the plant data, there is still probability of failure due to common cause problems. To address this, a recursive polynomial estimation model was developed which identifies common cause problems for individual component error ahead of time and control. The model of the recursive polynomial estimation was developed from the general linear dynamic model of the PCS behaviour as an Auto Regressive Moving Average (ARMAX) (Petr, 2014) in Equation 1; The flow chart of the algorithm was presented in Figure 7;

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})P\,(q^{-1})}u(k) + \frac{C(q^{-1})}{A(q^{-1})D(q^{-1})}n(k) \qquad (1)$$

Where y(k) is the output signal, u(k) is input signal, n(k) is noise with constant variance, A, B, C, D, and P are all shift transfer operators' polynomial as shown in the transfer functions below;

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots\ldots\ldots.. + a_{na} q^{-na}$$

$$B(q^{-1}) = b_1 q^{-1} + b_2 q^{-2} + \ldots\ldots\ldots\ldots + b_{nb} q^{-nb}$$

$$C(q^{-1}) = 1 + \mathrm{c}_1 q^{-1} + \ldots\ldots\ldots.. + c_{nc} q^{-nc}$$

$$D(q^{-1}) = 1 + d_1 q^{-1} + \ldots\ldots\ldots.. + d_{nd} q^{-nd}$$

$$P(q^{-1}) = 1 + p_1 q^{-1} + \ldots\ldots\ldots.. + d_{np} q^{-np}$$

The ARMAX in Equation 1 presented the polynomial regression of the PCS, while the corresponding predictor is presented as(Chan and Zhang, 2011);

$$\check{y}(t_k|p) = \phi^T\,(t_k.p)\check{p}(t_{k-1})` \qquad (2)$$

Where $\phi^T(t_k)$the repressor, p is the parameter vector and are defined as;

$$\phi^T(t_k) = [-y(t_{k-1}) \ldots\ldots \;\; \text{-}\,y(t_{k-n_a})\; u(t_k) \ldots\ldots\ldots\ldots u(t_{k-n_b}).^T$$

$$p = [a_1 \ldots\ldots\ldots\ldots a_{na} b_o \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots b_{nb]^T}$$

The model in equation 2 was rewritten as a general recursive algorithm in Equation 3(Cao and Schwartz, 1999; Chan and Zhang, 2011) which is the estimated step ahead prediction model of the PCS error;

$$\check{p}(t_k) = \check{p}(t_{k-1}) + u(t_k)L(t_k)\varepsilon\,(t_k) \qquad (3)$$

Where$L(t_k)$ is the adaptation gain, $u(t_k)$ is the scalar,$\check{p}(t_k)$ is the estimated time varying vector parameter, $\varepsilon\,(t_k)$ the predictor error and given as Equation 4 with $\check{y}(t_k|p)$ defined (3)

$$\varepsilon\,(t_k) = y(t_k) - \check{p}(t_k|p) \qquad (4)$$

The Recursive Polynomial Predictor algorithm
1. Start 2. Identify the PCS as ARMAX in Equation 1 3. Define the shift transfer polynomials (A, B, C, D, P) and noise function n(k) 4. Get the equivalent PCS predictor model with Equation 2 5. Transform to recursive form with Equation 3 6. Identify the estimated time varying vector $\check{p}(t_k)$ 7. Identify the predictor error $\varepsilon\ (t_k)$ 8. Return 9. Stop

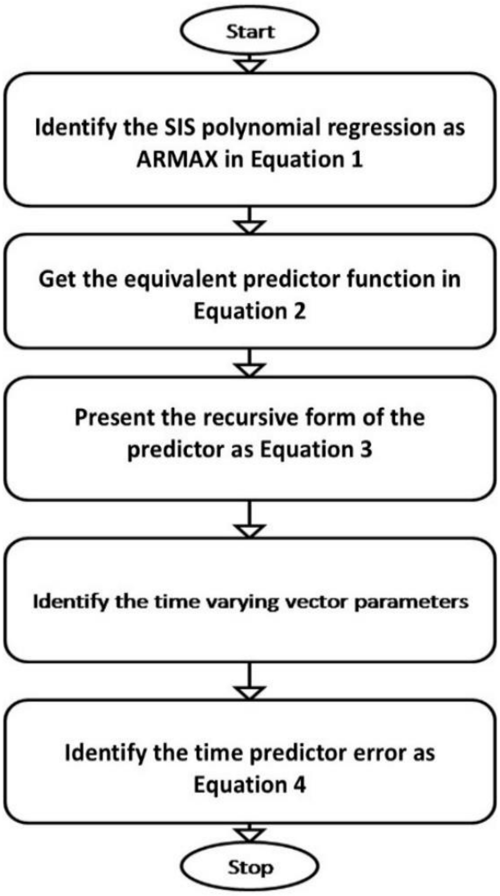


Figure 7: Flow chart of the polynomial estimator

Figure 7 present the flow chart of the polynomial model developed for the optimization of the PCS integrity level. The PCS model was identified as an autoregressive moving average function using the model in equation 1 with the polynomial equivalent shift operators, noise and input functions. The predictor of the ARMAX in Equation 1 is presented as Equation 2 which was used to estimate the next behaviour of the PCS and identifying any error in the components using the recursive form in Equation 3 with the error estimated defined as Equation 4.

Figure 8 present the complete system flow chart which shows the neuro-based PCS and the polynomial estimation model was used to monitor the technical process for tank overflow. When the process control fails, then sensors send the signal to the neurologic solver which then activates the control valve for stability of the reactor. However, when the controller suffers common cause problem the estimation algorithm detects it and then control the plant to prevent failures.

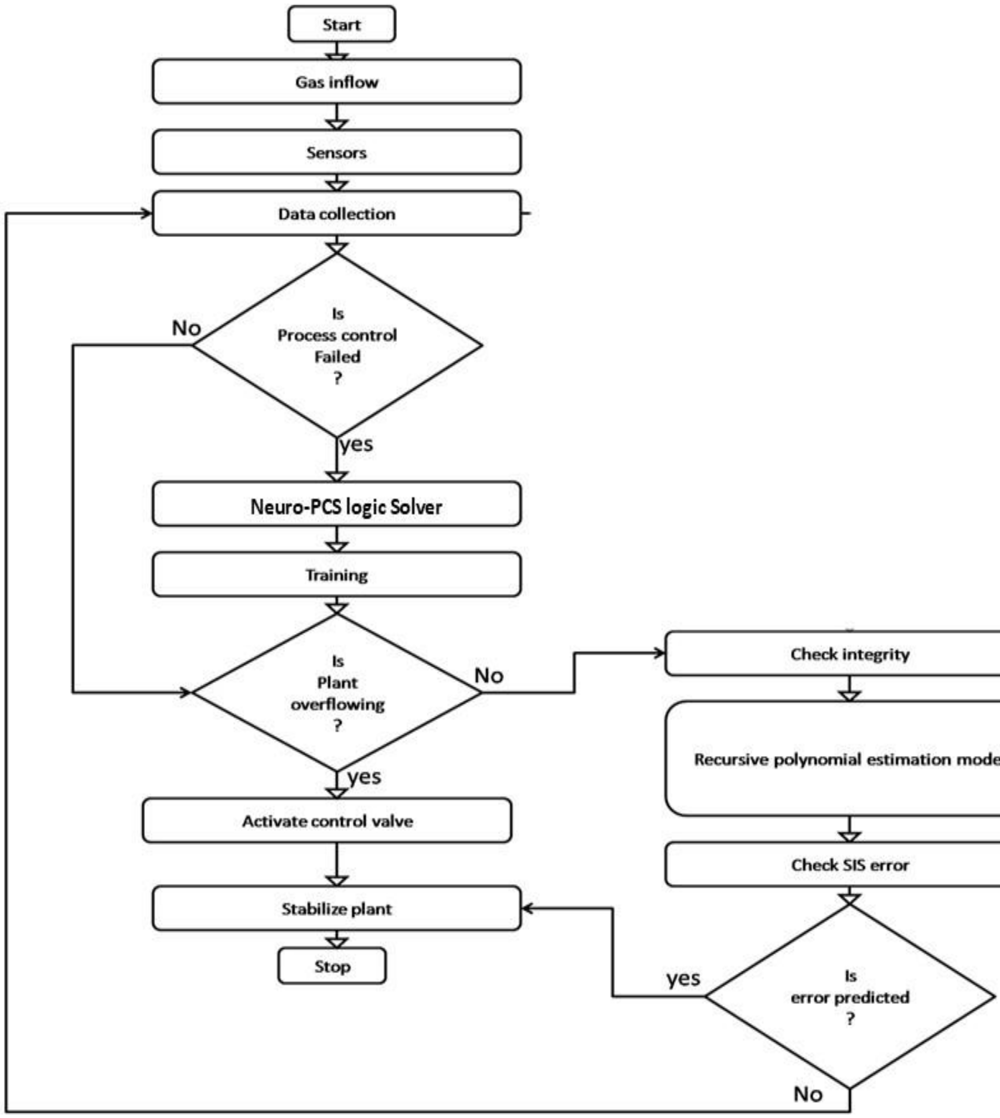


Figure 8: The complete system flow chart

Model of the PLC controller

The model of the PLC was developed from basic Proportional Integral Differential (PID) function of the PCS as shown in the model of Equation 5 (Inyama and Azubuike, 2015);

$$P = K_{p.error\,(t)} \quad (5)$$

$$I = K_I \int_0^t error\,(t)dt \quad (6)$$

$$D = K_{D\,\frac{derror(t)}{dt}} \quad (7)$$

Where the Equation 5 presented the Proportional functions, (6) presented the integral function and then the Equation 7 presented the derivative function. The relationship between the three mathematical functions presented the PLC control model as in Equation 8;

$$G = K_P\,(1 + \frac{1 + T_I.T_D.S^2}{T_I S}) \quad (8)$$

Where K_P is the proportional gain, T_I is the integral time constant, T_D is the derivative constant, $Ku_I = K_P/T_I$ is the integral gain and the $K_D = K_P/T_D$ is the derivative gain.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

SYSTEM IMPLEMENTATION

Model of the new PCS was developed using artificial neural network and the mathematical transfer function presented in Figure 9.The interconnection of the neurons with the activation functions and training algorithm to train the data of the distillation plant collected and generates the neuro logic solver algorithm. During the training process the recursive polynomial estimation function was used to check the system integrity via identification of the PCS as ARMAX in Equation 1 and then used the recursive model in Equation 4 to identify possible problem for control measures. The transfer function of the recursive polynomial estimation model.

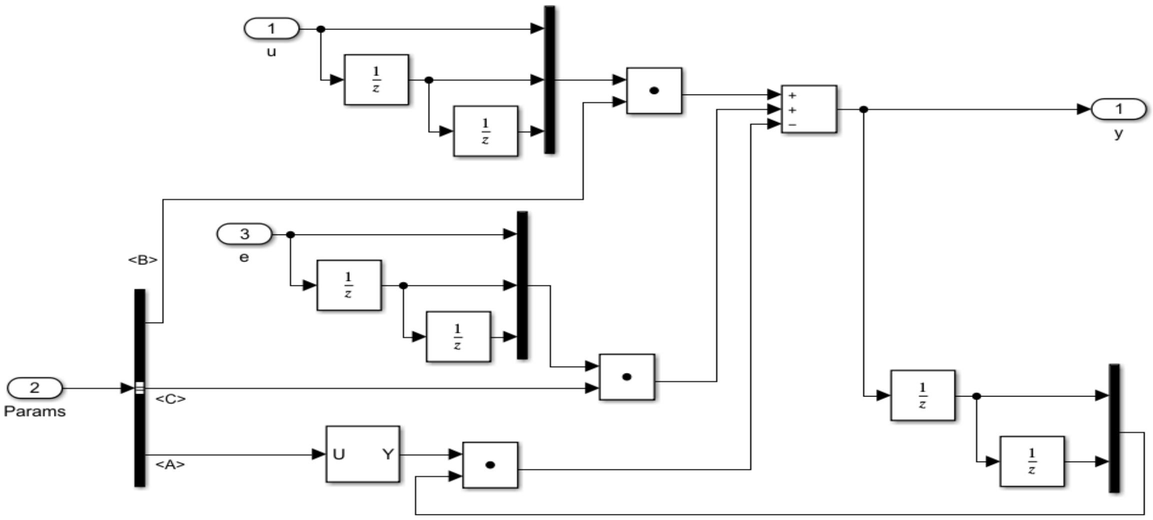


Figure 9: Transfer function model of the recursive polynomial estimator

The Figure 9 present the recursive polynomial mathematical transfer function which identify the PCS as an ARMAX, then regression was used to get the predictor and then apply recursive polynomial model in the Equation 4 used for the detection of PCS error and estimation. These models were all implemented using neural network toolbox, system identification toolbox, optimization toolbox and Simulink. The neural network tool was loaded with the plant data and then trained for the generation of the neurologic solver algorithm. The optimization toolbox was used to implement the recursive polynomial algorithm developed, which identified the plant behaviour using the system identification toolbox as ARMAX for monitoring and control measures. The Simulink block of the neuro-based PCS is shown in Figure 10;

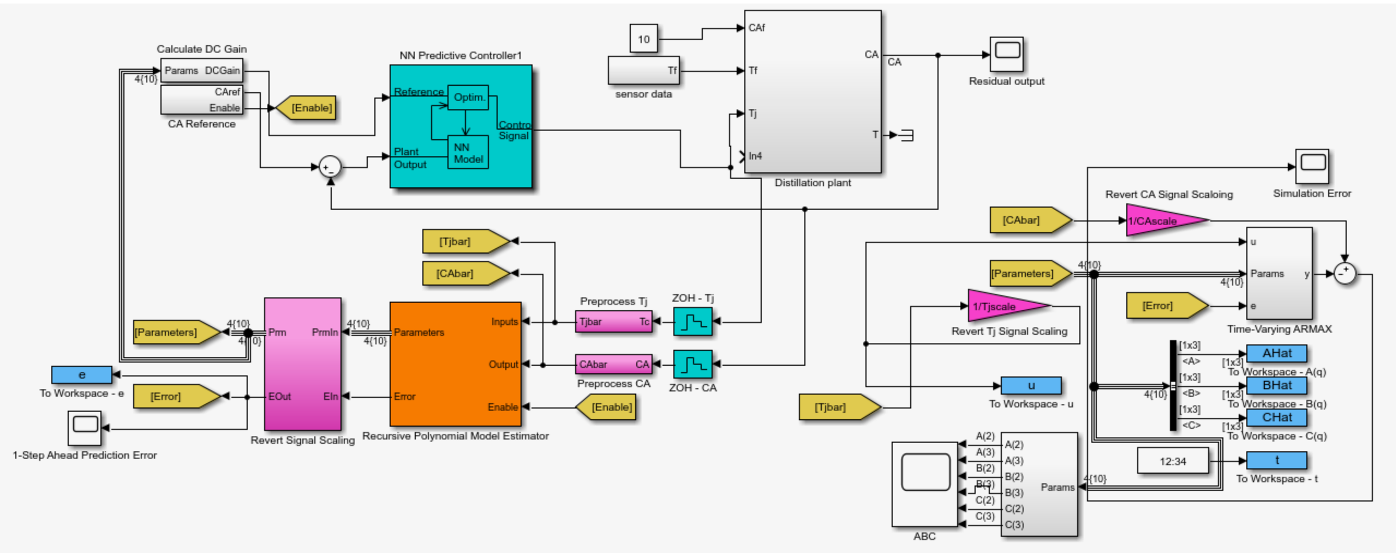


Figure 10: The Simulink model of the Neuro based PCS system

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

The Simulink implementation of the system was presented in the Figure 10, showing how the recursive polynomial estimation model developed was used to improve the safety integrity level of the PCS via monitoring and detection of errors. The neural network was trained as the safety integrity logic solver to monitor the behaviour of the distillation plant, while the polynomial was used to monitor various errors through the time varying Auto Regressive Moving Average (ARMAX) which can occur within process design and the notify through the revert signal scaling scope. While the neural network collects plant data from the process design, the compares with the neural network reference model to control the plant.

RESULTS OF THE NEUROLOGIC SOLVER ALGORITHM

From the risk assessment test conducted, it was uncovered that the PLC based logic solver has potential for dangerous failure as it is one of the most vital components of the PCS. This study developed neural network-based logic solver as shown in the Figure 4 and used to improve the integrity of the PCS. The performance of the neurologic solver was evaluated using regression and Mean Square Error (MSE) model as appeared in (Inyama and Azubuike, 2015). The MSE performance were presented in Figure 11;

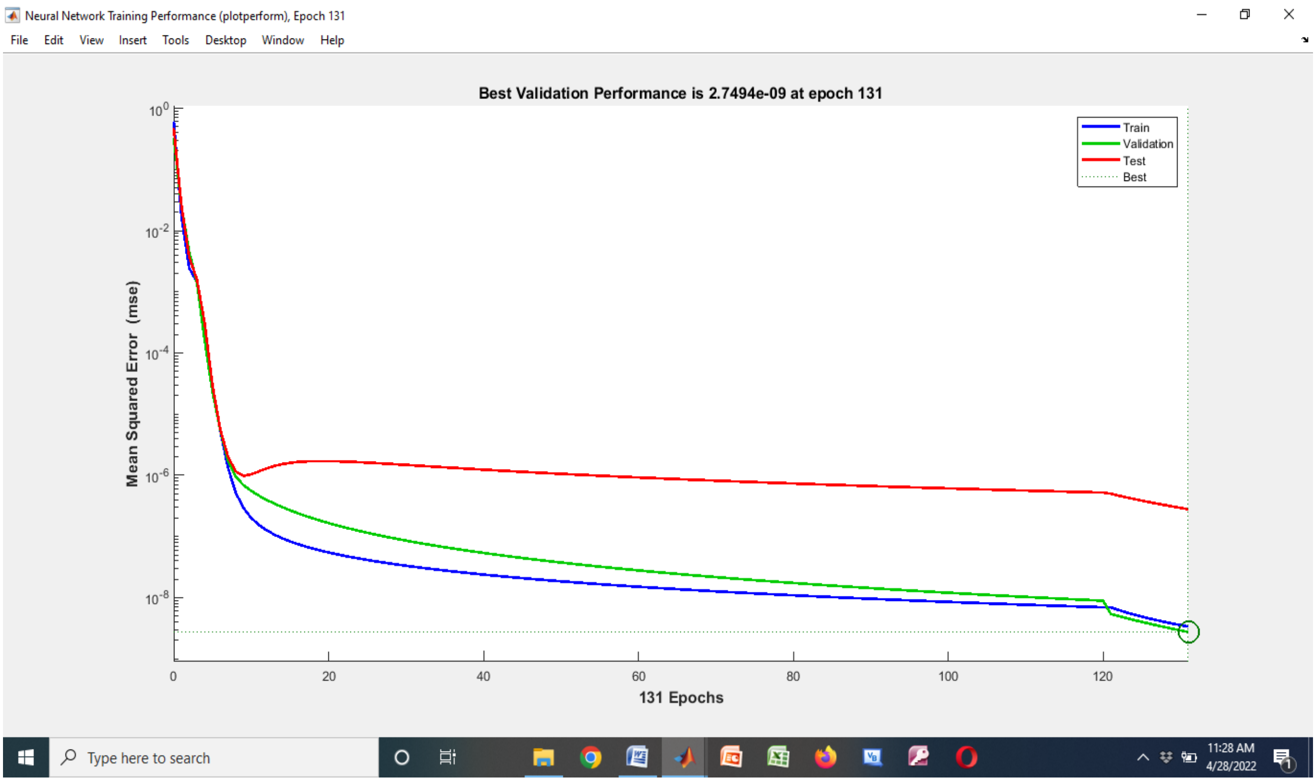


Figure 11: The MSE of the neurologic solver

The analysis of the results depicted in Figure 11 provided valuable insights into the accuracy and effectiveness of the neural network training and testing process. The primary objective of this assessment was to minimize the error associated with the neurologic solver algorithm. Remarkably, the achieved Mean Squared Error (MSE) of 2.7494E-09 indicated a level of error that can be considered practically negligible. This exceptional performance demonstrated the capability of the neurologic solver algorithm to generate highly precise and reliable outcomes.Furthermore, the subsequent evaluation focused on assessing the regression performance of the neurologic solver. This analysis aimed to determine the solver's ability to accurately detect and interpret signals from sensors, enabling it to make precise control decisions. Figure 12 visually presents the performance of the neurologic solver in this regard.The regression analysis involved comparing the predicted values generated by the neurologic solver with the actual sensor signals. By measuring the degree of correlation between the predicted and actual values, the regression performance of the neurologic solver was assessed. A high degree of correlation would indicate that the solver effectively captured and interpreted the sensor signals, leading to accurate control decisions.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

The results obtained from this evaluation provided crucial insights into the efficacy of the neurologic solver in detecting sensor signals and making precise control decisions. The close alignment between the predicted values and the actual sensor signals depicted in Figure 12 demonstrated the solver's ability to effectively analyze and interpret the data. This robust regression performance further substantiated the reliability and accuracy of the neurologic solver algorithm in the context of the PCS application.Overall, the combination of minimal error indicated by the MSE analysis and the strong regression performance showcased in Figure 12 reinforced the effectiveness of the neurologic solver algorithm. These results contribute to the overall confidence in the neurologic solver's ability to accurately process sensor signals and enable precise control decisions, thus enhancing the reliability and performance of the system.

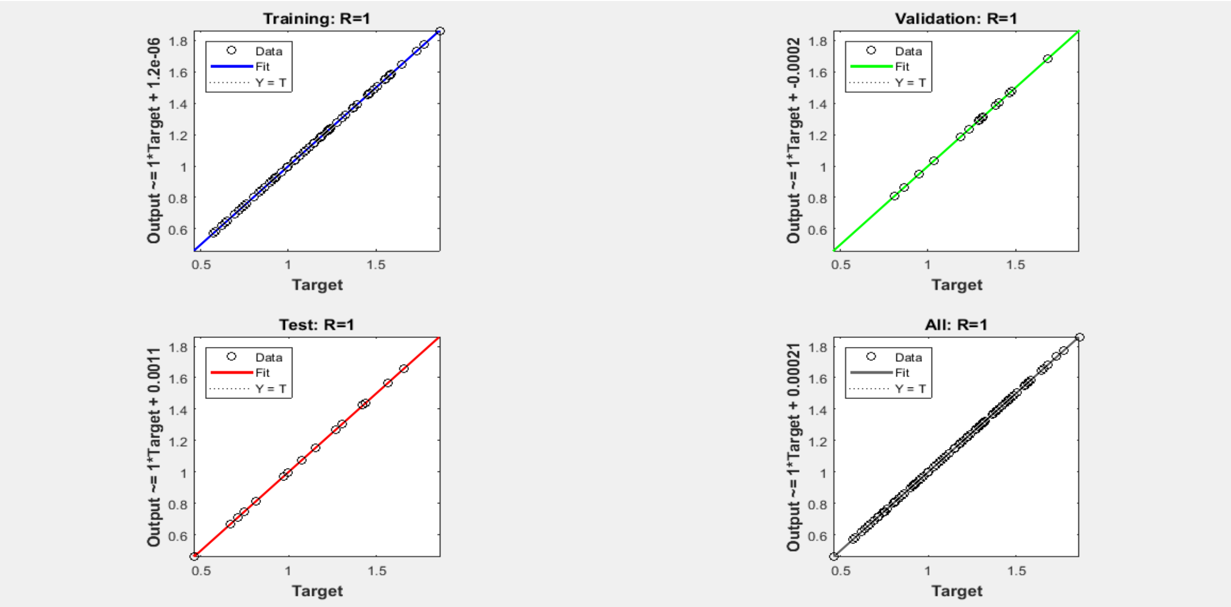


Figure 12: The Regression result

Figure 12 presents the regression performance of the neurologic solver. The aim here is to achieve a regression approximately or equal to one. The result here showed that the regression for the neurologic solver is 1, which implied reliability in controlling the tank overflow when error occurs in the process control section. To measure the failure rate of the neurologic solver, the neurologic solver was tested at operating time of 500hrs and the result presented in Figure 13;

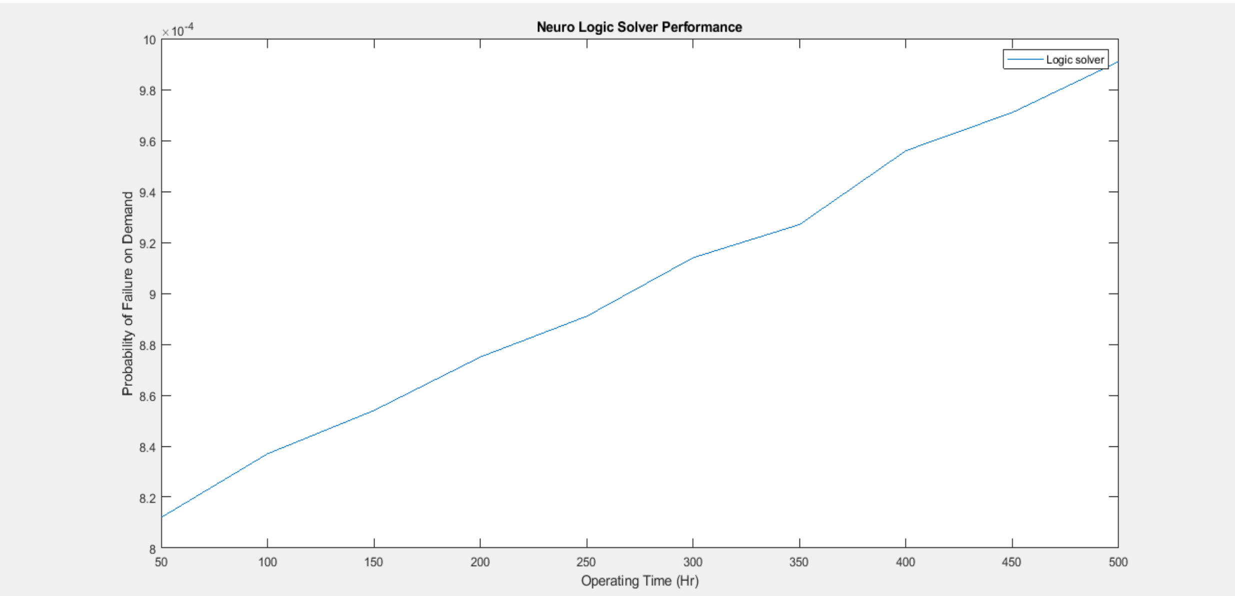


Figure 13: The neurologic solver PFD

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

Figure 13 presents the PFD of the neurologic solver using the PFD model to identify the failure of the neurologic solver over 500 hours of operation time. From the result, the PFD is 9.14E-04. This PDF shows that the neurologic solver has a SIL of 4 when referred to the IEC standard. This section is part of solution for the performance evaluation objective.

Validation of the Results

To validate the results of the neuro logic solver, the tenfold cross validation model in Dash et al. (2001) was adopted and used to validate the MSE, regression and PFD. The validation iteratively evaluated the respective logic solver performances and then computes their mean to determine the overall performance of the neuro logic solver as shown in Table 3;

Table 3: Validation Result of the Neurologic solver

S/N	**MSE**	**Regression**	**PFD**
1	2.7494E-09	1.0000	9.14E-04
2	2.9644E-09	0.9978	8.98E-04
3	3.7973E-09	0.9971	9.87E-04
4	2.8773E-09	1.0000	8.69E-04
5	3.1535E-09	0.9981	8.74E-04
6	3.0071E-09	0.9992	7.72E-04
7	3.6877E-09	0.9947	8.79E-04
8	2.4012E-09	0.9982	9.51E-04
9	2.4772E-09	0.9930	8.92E-04
10	2.7034E-09	1.0000	9.67E-04
Average	2.98E-09	0.99781	9.00E-04

Table 3 presents the validation performance of the neurologic solver algorithm developed for the optimization of the PCS safety integrity level. From the result, it was observed that the validated MSE result is 2.98E-09, Regression is 0.9978 and PFD is 9.00E-04 which implies safety integrity level of 4.

Results of Neuro PCS with the Recursive Polynomial Estimator

This section presents the performance of the neuro PCS with the recursive polynomial estimation algorithm developed for the estimation of error in the system performance. The result showed the error identification of the PCS as ARMAX according to the Equation 1with the equivalent polynomial regression (which contained the white noise as the simulated error).

Figure 14 provides a visual representation of the system identification results obtained from the polynomial estimation model. The purpose of this model is to accurately determine and understand the behaviours exhibited by the PCS plant. By employing a polynomial regression transformation, the model can effectively capture and represent the underlying patterns and dynamics of the PCS plant.

The process of system identification involves analysing the input-output relationship of the PCS plant and extracting relevant information from the collected data. The polynomial regression transformation, as defined in Equation 3, serves as a mathematical framework for mapping the input signals to the corresponding output responses. This transformation enables the model to capture the complex relationships and non-linearities inherent in the PCS plant's behaviour.Figure 15 showcases the application of the recursive polynomial function in predicting the step ahead error of the signal. This function plays a crucial role in estimating the error that may occur in future time steps. By leveraging historical data and the identified polynomial model, the recursive polynomial function can forecast the potential deviations or

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

discrepancies between the expected and actual signal values.

The ability to predict the step ahead error is of great significance in ensuring the robustness and reliability of the PCS. It enables proactive measures to be taken in response to potential errors or anomalies, thereby preventing system failures or hazards. By continuously monitoring and analysing the predicted errors, appropriate corrective actions can be implemented in a timely manner to maintain the optimal performance of the PCS.Overall, the utilization of the polynomial estimation model and the recursive polynomial function facilitates a comprehensive understanding of the PCS plant's behaviours and enhances the system's ability to identify and address potential errors. This contributes to the overall reliability, safety, and effectiveness of the PCS in critical operational environments. The Figure 15 presents the step ahead error prediction.

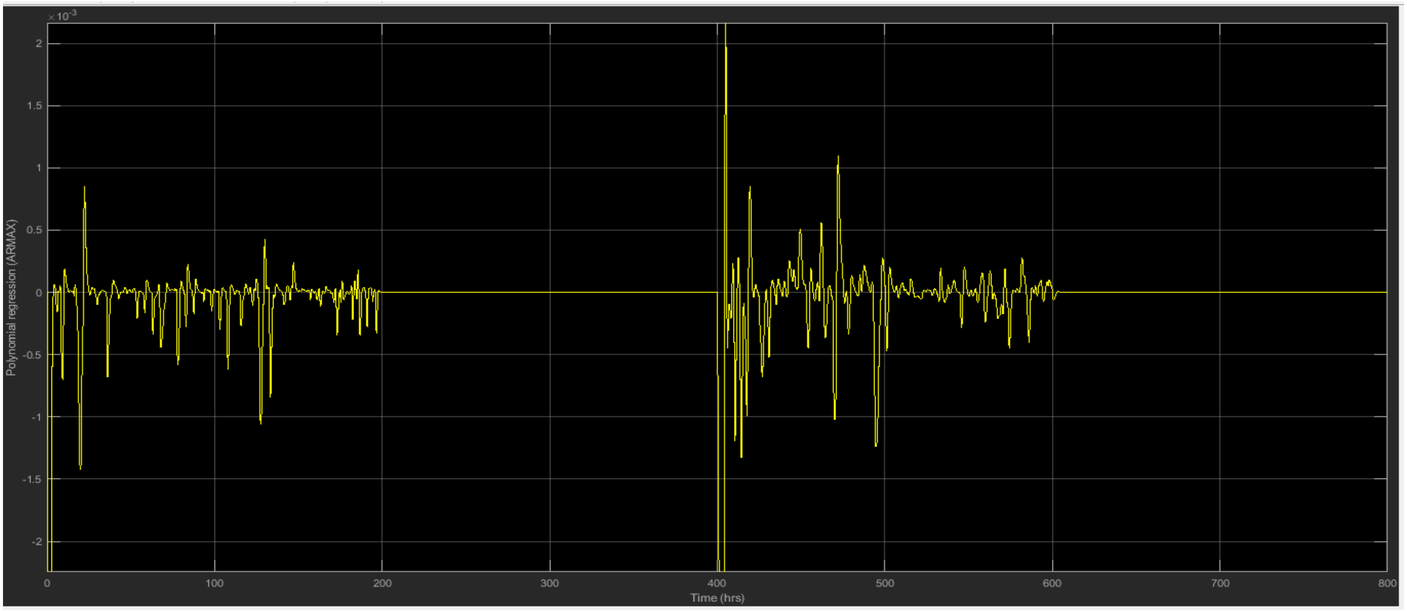


Figure 14: The PCS error identification result

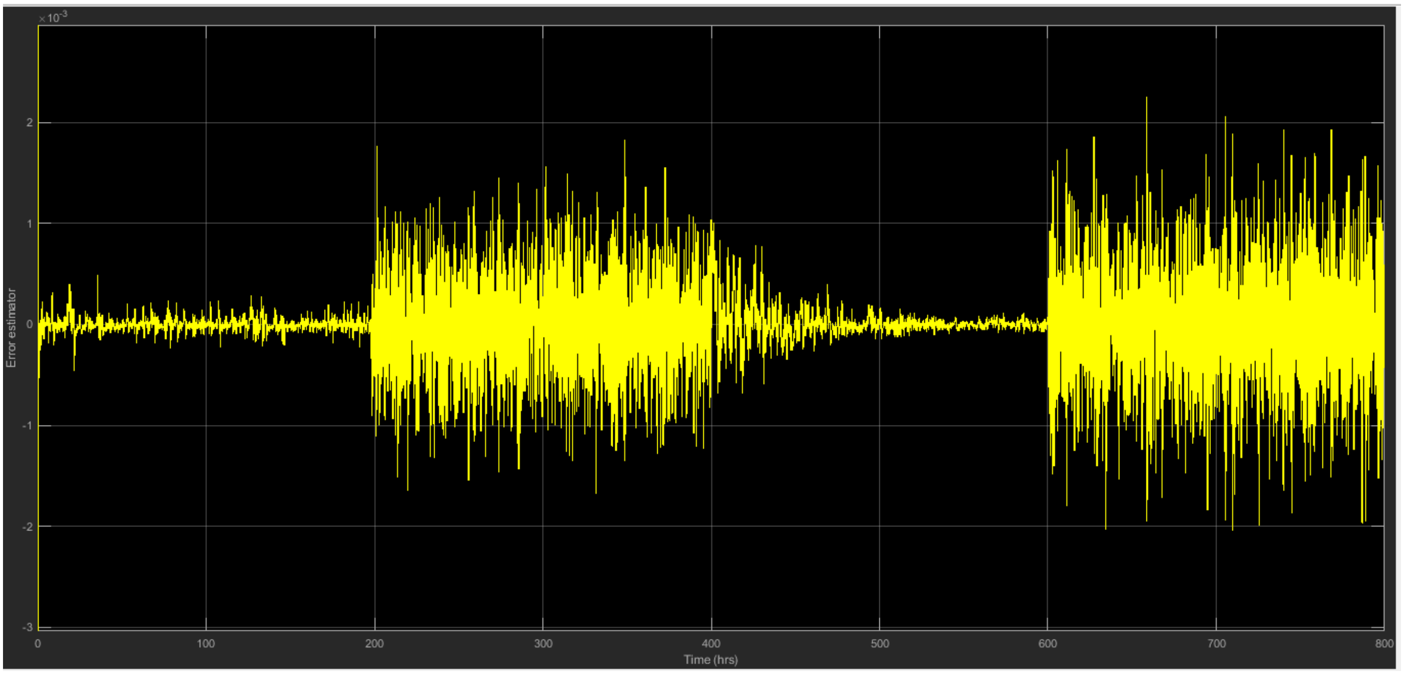


Figure 15: The Step ahead Estimated Error

Figure 15 present the step ahead error predictor of the PCS. From the model, the error in the PCS was estimated with Equation 4 while the controlled output to stabilize the plant is presented in figure 16;

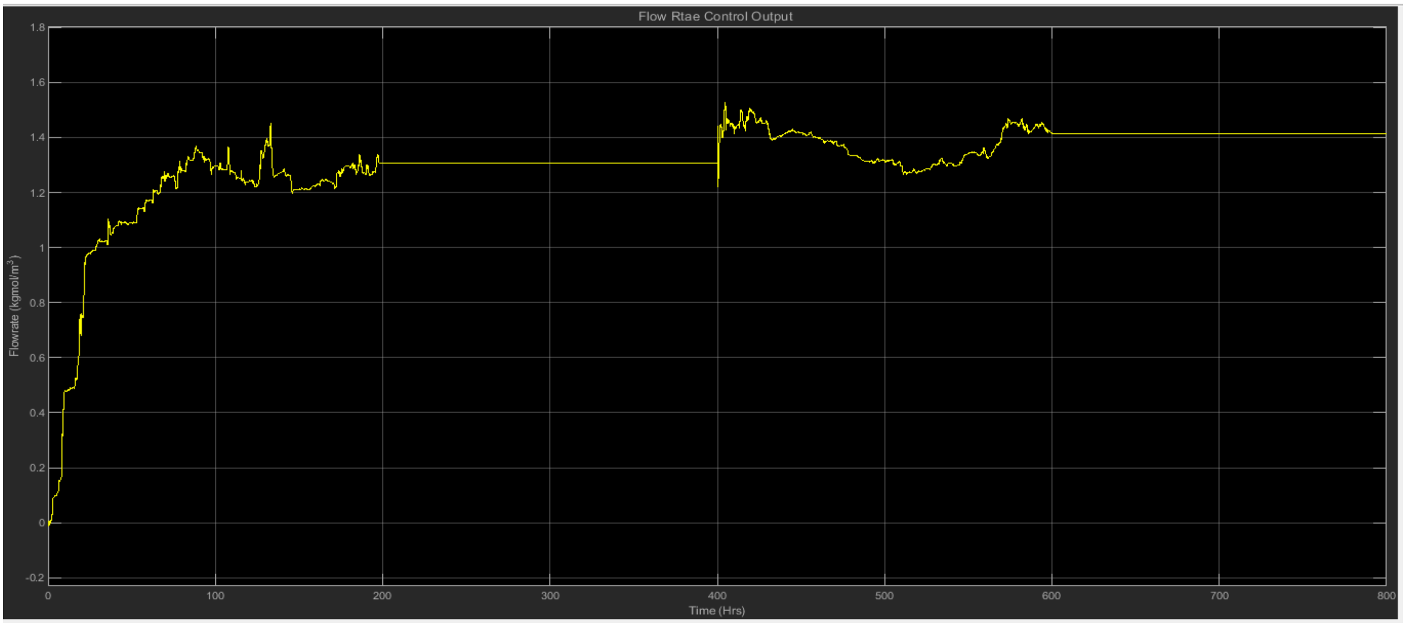


Figure 16: The Control output

Figure 16 provides a graphical representation of the control output of the plant, highlighting the significant impact of identifying and estimating errors ahead of time. By utilizing the neurologic solver algorithm and the error estimation techniques, the system was able to proactively identify and address potential errors, thereby minimizing the probability of system failure.

The integration of the neurologic solver algorithm allowed for the timely detection and estimation of errors in the system. This enabled proactive measures to be taken to mitigate the identified errors and ensure smooth and reliable system operation. As a result, the control output of the plant depicted in Figure 16 demonstrates a notable reduction in potential failures and an overall increase in the safety integrity level of the system.The ability to identify and estimate errors ahead of time provides a crucial advantage in terms of system reliability and safety. By addressing potential issues before they can escalate, the likelihood of system failure is significantly reduced, thereby enhancing the overall performance and integrity of the system.

The findings presented in Figure 16 highlight the positive outcomes achieved through the implementation of the neurologic solver algorithm and the error estimation techniques. By effectively managing errors and optimizing the control output of the plant, the system's reliability is greatly enhanced, and the safety integrity level is elevated.Overall, the utilization of these advanced techniques enables the system to operate with greater efficiency and resilience. The reduction in the probability of system failure demonstrated in Figure 16 reflects the successful implementation of proactive measures, ultimately leading to improved safety, increased reliability, and enhanced overall system performance.

CONCLUSION

In this paper, a neuro-based PCS is designed and implemented to integrate a recursive polynomial estimator into a distillation plant which is in turn utilized in predicting faults, safety integrity checking. The system conformed to the structured evaluation and design philosophy of IEC 61069, and the Safety Integrity Level (SIL-4) rating as per IEC 61508, which means the system had outstanding reliability and fault tolerance. A time-series data-set of inputs and outputs of the process was used to train and develop a feedforward neural network (FNN) to simulate and implement control of the dynamic behaviour of the process of distillation. The process stability and tracking ability provided by the neural solver had a mean square error (MSE) equal to 3.09 10^{-9} and the regression R approximately equal to 1.0, so the accuracy of the prediction in a wide range of operating conditions was very high.

To add to this, a recursive polynomial estimator with an ARMAX model structure was utilized to estimate the model in order to continuously monitor the differences between predicted and actual outputs to detect possible integrity violations in real time. Integrated system implementation was in a MATLAB/Simulink system and assessed under real-time conditions. The probability of failure on demand (PFD) taken after 500 hours of observation was taken out to be 9.00 10^{-4} which confirmed the system to have passed as per SIL 4 standards. The neuro-recursive hybrid structure delivered the right control and ahead-of-time fault prediction, thus improving functionality safety and system access to the safety-critical settings. In general, the suggested intelligent control and monitoring framework provides credible and scalable process of industrial PCS application, which is able to mitigate risks pro-actively and can also align with international safety standards set therein.

Limitation of this work

The performance of the system depends on the quality and representativeness of the training data, and its computational complexity may pose challenges for real-time deployment in resource-constrained environments.

REFERENCES

1. Anthony, E. W. (2018). Reconsidering the industrial revolution: England and Wales. *Journal of Interdisciplinary History*, *49*(1), 9–42.
2. Cao, L., & Schwartz, H. (1999). A novel recursive algorithm for directional forgetting. *Proceedings of the American Control Conference*, San Diego, California, 455–463.
3. Chan, C., & Zhang, Z. (2011). Local polynomial modeling and variable bandwidth selection for time-varying linear systems. *IEEE Transactions on Instrumentation and Measurement*, *60*(3), 1102–1117.
4. Cory, A. (2013). Top causes of PLC control system failure. *Technique Group*. http://www.lertechnqiue.com
5. Dash, P., Pradhan, A., & Panda, G. (2001). Application of minimal radial basis function neural network to distance protection. *IEEE Transactions on Power Delivery*, *16*(1), 68–74.
6. Fallahi, M., & Azadi, S. (2011). A novel adaptive sliding mode control for DC motor. *International Journal of Advanced Engineering Applications*, *4*(3), 19–27.
7. Institute of Entrepreneurship Development. (IED) (2019). *The 4 industrial revolutions*. IED Teams.
8. Inyama, H. C., &Agbaraji, C. (2015). A survey of controller design methods for a robot manipulator in harsh environments. *European Journal of Engineering and Technology*, *3*(3). https://doi.org/10.1000/ejengtech.v3i3.567
9. Joseph, R. (2003). Using fault trees to analyze safety-instrumented systems. *Isograph Inc.*, 1–12.
10. Mallikarjun, G. H., Umayal, R. M., Kabberalli, S. K., &Hittalamani, R. (2017). Programmable logic controller (PLC) in automation. *Advanced Journal of Graduate Research*, *2*(1), 37–45. https://doi.org/10.1000/ajgr.v2i1.123 (Note: Add DOI if available)
11. Marc, T., Jonathan, T., Lea, B., Peter, M., & Susanne, W. (2019). *Industrial robotics: Insights into the sector's future growth dynamics*. Advanced Industries, Visual Media Europe.
12. Obaid, A., Muhammad, N. M., &Houssem, J. (2018). A survey on control techniques of a benchmarked continuous stirred tank reactor. *Journal of Engineering Science and Technology*, *13*(10), 3277–3296.
13. Okafor, P. U., Eneh, P. C., & Arinze, S. N. (2017). Model reference adaptive control scheme for eliminating overshoot in DC motor. *International Journal of Advanced Research in Innovative Engineering (IJARIE)*, *6*(3). ISSN 2278–6244.
14. Petr, N. (2014). Recursive estimation algorithms in MATLAB and Simulink development environment. *WSEAS Transactions on Computers*, *13*, 691–702.
15. Rahul, U., & Rajesh, S. (2020). Analysis of CSTR temperature control with adaptive and PID controller (A comparative study). *IACSIT International Journal of Engineering and Technology*, *2*(5), October 2010. ISSN: 1793-8236.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VIII August 2025

16. Realpears Video. (2017). *Realpears tutorials*. http://www.Realpears.com/2017/tutorials
17. Skogestad, S. (2011). Probably the best simple PID tuning rules in the world. *Journal of Process Control*, 21(4), 1–27. https://doi.org/10.1016/j.jprocont.2011.01.003 (Replace with actual journal volume/issue if needed)
18. Yuvraj, V. (2012). Comparison of ANN and PID controllers for industrial water bath temperature control system. *International Journal of Computer Applications (IJCA, 53)*(2), 0975–8887.