National University of Singapore

Electrical & Computer Engineering

EE5111 Selected Topics in Industrial Control & Instrumentation

CA4

LUO ZIJIAN

A0224725H

In this CA4 report, you will find the context of Assignment one and Assignment two. My answer all follow from their question description.

Assignment 1

Question 1

It is observed that a sensor has fault. The input/output data is shown in the table below. Please fit the nonlinear relationship between the input and output in an equation below

My Answer

Part a

For non-linear linear relationship between input and output, we can use these formulas to calculate the final result.

$$X = \begin{bmatrix} \mathbf{1} & x_{11} & x_{12}^2 & \dots & x_{1p}^{\mathbf{p}} \\ \mathbf{1} & x_{21} & x_{22}^2 & \dots & x_{2p}^{\mathbf{p}} \\ & \vdots & & & \\ \mathbf{1} & x_{n1} & x_{n2}^2 & \dots & x_{np}^{\mathbf{p}} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{1}$$

$$\mathrm{y} = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \cdots + b_p x^p \tag{2}$$

For the Estimated coefficients, it was expressed as:

$$\hat{b} = \left(X^T X\right)^{-1} X^T Y \tag{3}$$

For the Model output, we can calculate through this equation

$$\hat{Y} = X\hat{b} \tag{4}$$

When it turns to the quality of estimation, the direct result is to compare the actual y with the estimated output.

$$e = Y - X\hat{b} \tag{5}$$

Follow the steps in the lecture notes, we can get the results

- when p=1, the model is $\hat{y} = 0.6931x + 0.2853$ in Figure 1
- when p=2, the model is $\hat{y} = 0.5526 + 0.5005x + 0.0253x^2$ in Figure 2
- when p=3, the model is $\hat{y} = 0.7781 + 0.2126x + 0.1153x^2 - 0.0079x^3$ in Figure 3
- when p=4, the model is $\hat{y} = 2.4868 - 2.7794x + 0.16487x^2 - 0.3067x^3 + 0.0196x^4$ in Figure 4
- when p=5, the model is $\hat{y} = -2.3474 + 7.4798x - 5.5457x^2 + 1.9029x^3 - 0.2881x^4 + 0.0159x^5$ in Figure 5

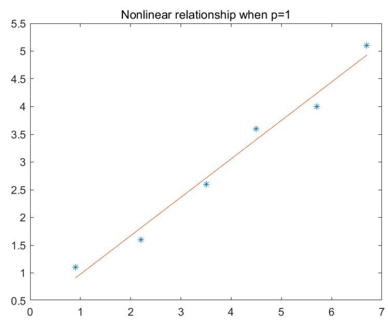

Figure 1: Nonlinear relationship graph(p=1)

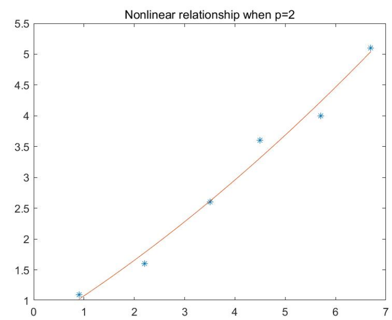

Figure 2: Nonlinear relationship graph(p=2)

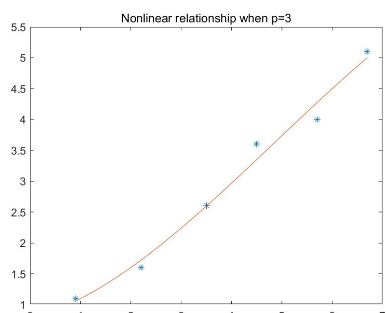

Figure 3: Nonlinear relationship graph(p=3)

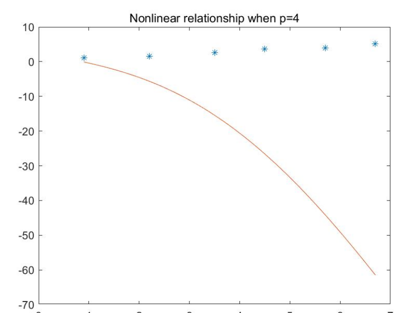

Figure 4: Nonlinear relationship graph(p=4)

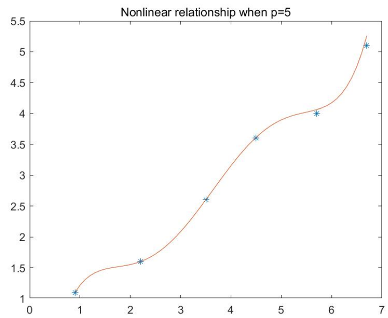

Figure 5: Nonlinear relationship graph(p=5)

Part b

From the equation 5, we can calculate the error result in these conditions in Table 1.

	p=1	p=2	p=3	p=4	p=5
error(x1)	0.1909	0.0764	0.0430	-0.0100	$-0.0159 * 10^{-8}$
error(x2)	-0.2102	-0.1762	-0.1197	0.0555	$-0.0175 * 10^{-8}$
error(x3)	-0.1112	-0.0142	0.0041	-0.1428	$-0.2011 * 10^{-8}$
error(x4)	0.1957	0.2829	0.2500	0.1555	$-0.1751 * 10^{-8}$
error(x5)	-0.2360	-0.2274	-0.2737	-0.0766	$-0.5274 * 10^{-8}$
error(x6)	0.1708	0.0585	0.0963	0.0185	$-0.8572 * 10^{-8}$

Table 1: Error in different p

We have six coefficients from 0 to 6th power of x because we employ most variants of the equation, allowing us to more exactly replicate the variance of input and output data. The gap between the estimated number and the output data is also the lowest when the error is calculated from the accurate value.

In a word, we choose the model(p=5) as the best fault model, the relationship is $\hat{y} = -2.3474 + 7.4798x - 5.5457x^2 + 1.9029x^3 - 0.2881x^4 + 0.0159x^5$

Question 2

Hardware sensor fault diagnosis method can be used for detecting various faults in machines. Now we use one sensor to detect the possible faults in one machine. From the experimental test, we collected the sensor readings from the healthy state as shown below and wanted to do fault diagnosis.

Please choose a suitable technique to check if the machine has a fault (if you select the threshold decision method, the tolerance number is 0.15) and explain why you select this method. You may use the graph to illustrate your conclusion. Also, please give the fault detection time if a fault is detected.

My Answer

In this part, I use trend analysis as the technique to detect the fault.

For the healthy state (time from 1 to 9), we can calculate the sensitivities(0.1, 0.1, -0.15, 0.51, 0.04, 0.1, -0.05, 0.05) in Table 2 from this formula.

$$\frac{\Delta_O}{\Delta_I} = \frac{O_i - O_{i-1}}{I_i - I_{i-1}} \tag{6}$$

sensitivity	0.1	0.1	-0.15	0.51	0.04	0.1	-0.05	0.05

Table 2: Sensitivity from healthy state

From the results, we can know that the minimum value of sensitives is -0.15, the maximum value of sensitivities is 0.51. Therefore, the normal range of sensitivity is from -0.15 to 0.51.

Turning to the sensor readings from 100 to 108, we can calculate the sensitives(0.1, -0.05, 0.35, 0.2, 0.5, 0.2, 1.1, 1.3) in Table 3.

sensitivity	0.1	-0.05	0.35	0.2	0.5	0.2	1.1	1.3

Table 3: Sensitivity from time 100 to 108

Obviously, we can find that the last two values are beyond the healthy state maximum sensitivities. Through the key idea in trend analysis, we know that there is a fault in the system and the detection time is 107 s.

I choose trend analysis because its pre-defined range of sensitivity can tell there is a fault when the sensitivity exceeds it. If the hardware is in the normal state, its sensitivity should keep similar trend as before.

Question 3

For model-based fault detection method, please draw a block diagram to demonstrate it and explain it briefly.

My Answer

For the steps of the model-based fault detection method in Figure 6, firstly we should create a nominal system model to simulate the estimated output. Then we compare the result with the actual outcome. Based on the residual result through this equation

$$r(k) = Y_{true}(k) - Y_{estimated}(k) \tag{7}$$

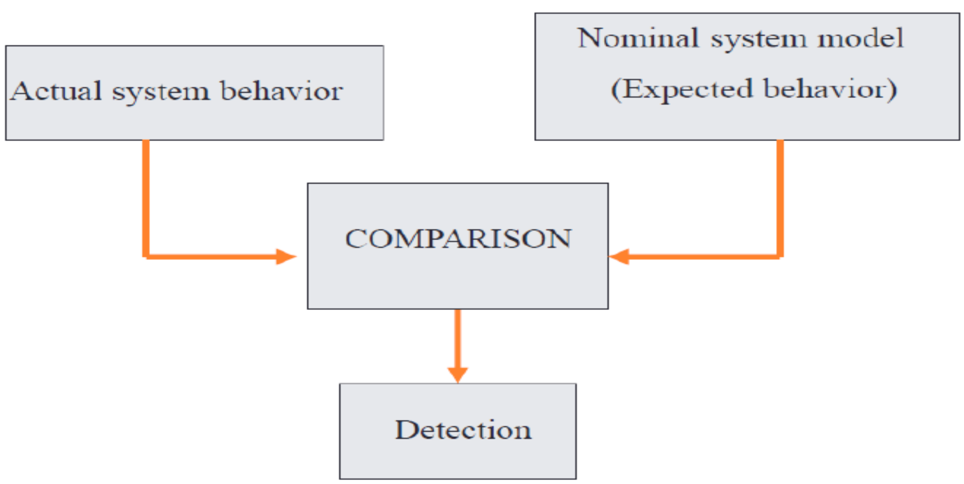

Figure 6: Working principle of model based approaches in lectures:

and then we should compare the residual result with a given threshold. If the residual result is larger than the given threshold, we declare that we detect the fault, otherwise no fault detected.

As for the pre-defined threshold, we can represent it(given threshold = Mean + deviation + small tolerance number) or (given threshold = Mean + $\alpha * deviation(\alpha > 1)$

To build up a model, we need an estimator which can be a linear or nonlinear model observer; filters, which are usually linear or nonlinear Kalman filters and nonlinear neural network model. The choice of model is depends the system requirement. Finally, I prepare a work flow graph below Figure 7, you can see more details about the basic structure of model-based fault detection.

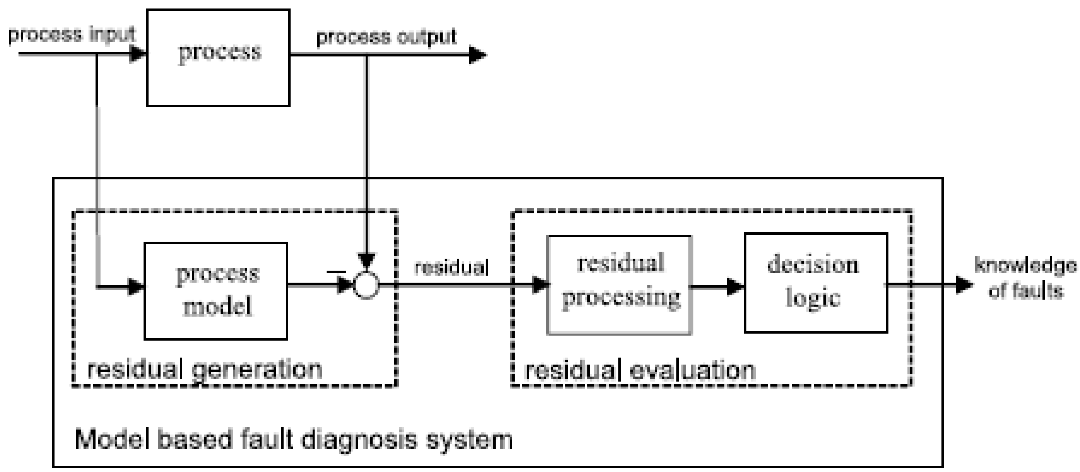

Figure 7: Working flow of model based fault detection approaches

Assignment 2

Based on the lecture notes, and any additional reading materials from literature, answer the following questions:

Question 1

What is fault-tolerant control? Please give one or two application examples.

My Answer

Faults in automated processes frequently cause unintended reactions and plant shutdowns, affecting product yield, quality, plant infrastructure, workers, and the environment. Fault tolerance is the ability of an operating system to respond to a hardware or software malfunction. The ability of a system to continue working in the face of faults or malfunctions is referred to as fault tolerance. A single point of failure cannot disturb an operating system that provides a robust definition for defects. In the event of a breakdown, it assures business continuity and high availability of critical applications and systems.

If a fault occurs, we perform manual service using traditional FTC procedures. Faults in industrial automation systems will be handled using modern fault tolerant control approaches. Hardware, software, and mixed techniques are all examples of modern fault tolerant control mechanisms. Hardware methodsâĂŤin order to accommodate defects, redundant hardware (spare) is used. To tolerate flaws, software approaches use software to implement the proposed algorithms (less sensors). Mixed approaches entails the employment of both automated and human interaction.

In this part, I want to use two examples to illustrate the applications of fault-tolerant control.

Firstly, I would like to show Pixhawk PX4 2.4.8 Flight Controller Kit Set.

Figure 8: Pixhawk PX4 2.4.8 Flight Controller Kit Set

The kit contains these sensors: L3GD20 3 axis digital 16-bit gyroscope, LSM303D 3 axis 14-bit accelerometer/magnetometer, MPU6000 6 axis accelerometer/ magnetometer (redundant component), MS5611 high precision barometer, GPS. For fault tolerant control, Pixhawk employs two accelerometers: ID1 and ID2. ID1 is typically used as the default. Accelerometer ID1 should change when velocity changes; if it does, it is working; otherwise, switch to ID2. In addition, the defect detection analyzes the accelerometer's range. Switch to ID2 if the range is exceeded.

Another example is the Sensor Fault Tolerant Control for Aircraft Engines.

The Kalman filters (KF) bank scheme of fault-tolerance control, shown in Figure 9, is an example of a typical notion. The KF bank, working as an FDI portion, detects and isolates the broken sensor, which is subsequently replaced with a virtual sensor signal solved by KF. However, the efficacy of FTC is heavily reliant on the quality of FDI, and a sluggish detection method may result in tardy FTC replies. As seen in Figure 10, several researchers have employed sliding mode theory to compensate for sensor errors. The sensor fault reconstruction signal can be obtained in real-time using sliding mode observers, and it is used to rectify corrupted measured signals before they are used by the controller.

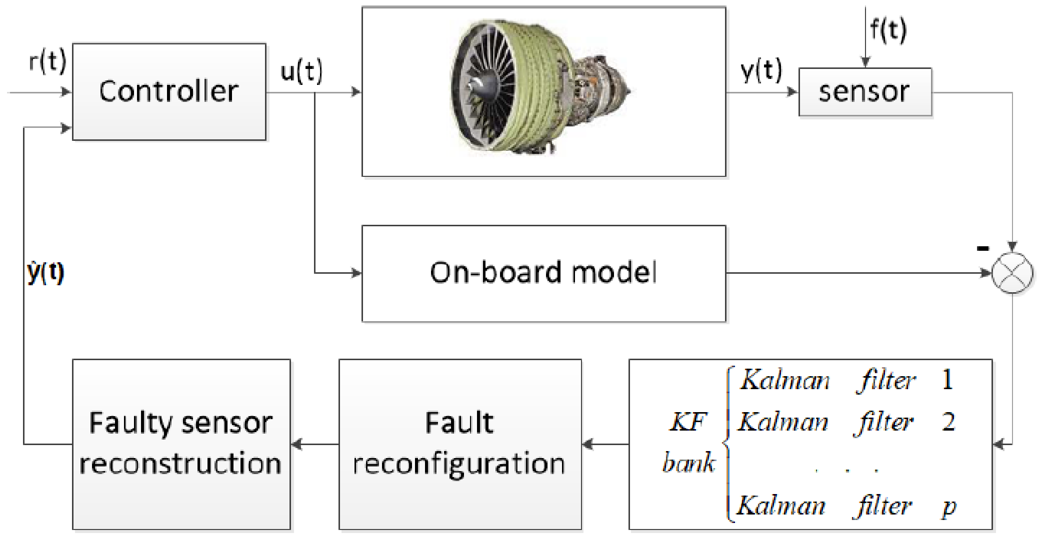

Figure 9: Schematic of FTC based on Kalman filters (KF) bank

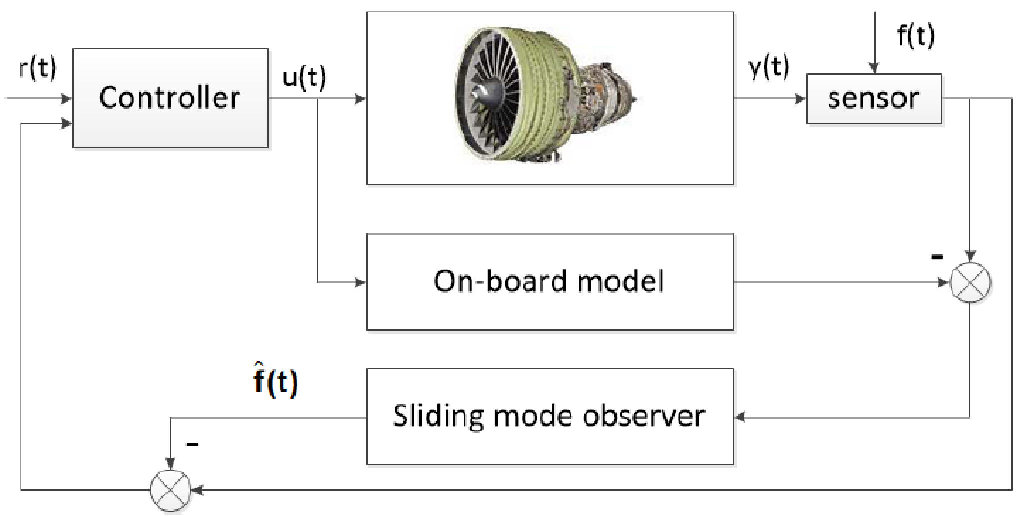

Figure 10: Schematic of FTC based on fault reconstruction of sliding mode observer (SMO)

Question 2

Please explain three soft-based fault-tolerant control methods, i.e., model-based fault-tolerant control methods, and draw the corresponding flowcharts, respectively.

My Answer

The three soft-based fault-tolerant control methods are Passive FTC, Active FTC, and Hybrid FTC.

Passive FTC systems

Passive FTC systems in Figure 11 do not rely on the fault information because it is a robust control against a set of preset problems, therefore no fault identification is required. The benefit of passive FTC is that the robust control can be developed to reach acceptable performance without the time delay. Passive FTC, on the other hand, allows only a certain amount of errors to be allowed, and the solution is often conservative, unable to achieve certain performance criteria.

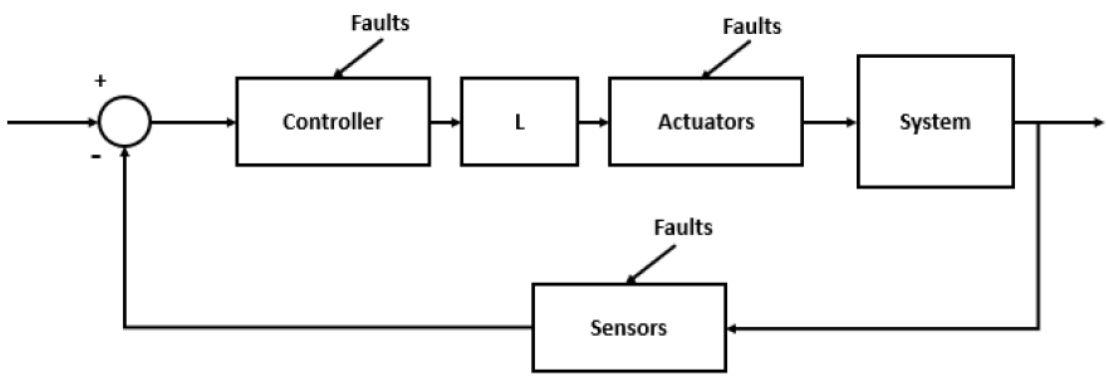

Figure 11: Flow chart of passive FTC systems

Active FTC systems

Active FTC systems in Figure 12, in contrast to passive FTC systems, respond to each fault by changing control operations. To achieve fault tolerance, an active FTC system, unlike a passive FTC system, mainly relies on fault diagnostics. Fundamentally, active FTC responds to detected defects by applying the appropriate controls, ensuring that stability is maintained and performance remains within acceptable limits. In general, an active FTC system is more effective at dealing with various types of problems. However, the controller's ability to provide fast and correct fault information is primarily reliant on the FDI unit.

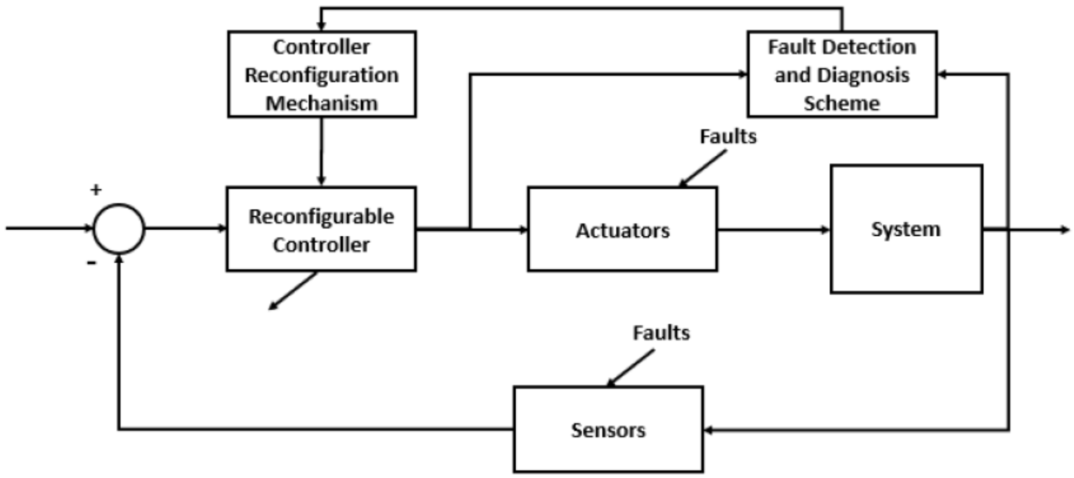

Figure 12: Flow chart of active FTC systems

If the fault function is unknown, artificial intelligence should be used to assist us discover the unknown fault and accomplish FTC. To construct an on-line estimator of the defect function, neural networks, fuzzy logic systems, and adaptive learning algorithms can be employed. The following is a flowchart of AI-based fault-tolerant control in Figure 13

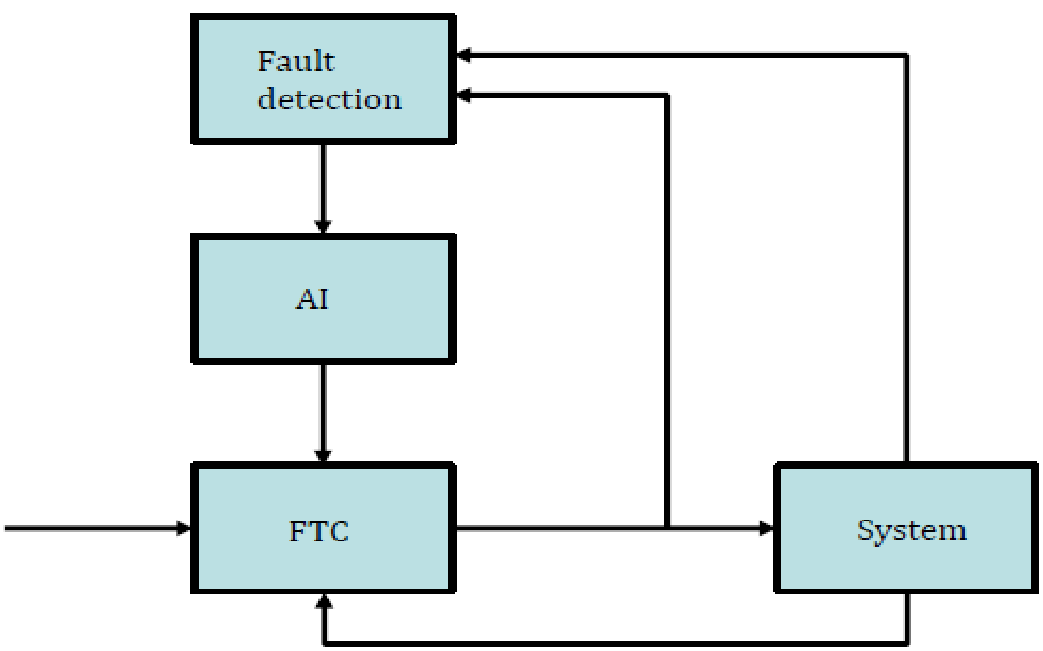

Figure 13: Flow chart of AI FTC systems

Hybrid FTC system

The real-time fault detection/diagnosis technique is the bottleneck in any active FTC. Because the control system must work in a real-time environment with a limited amount of measurements, noise will frequently contaminate this data, resulting in the probability of making a mistake. As a result, Hybrid FTC in Figure 14, which combines passive and active FTC systems, will be a viable option. In an active FTC system, a passive FTC will offer stability cushioning while the active FTC will boost system performance.

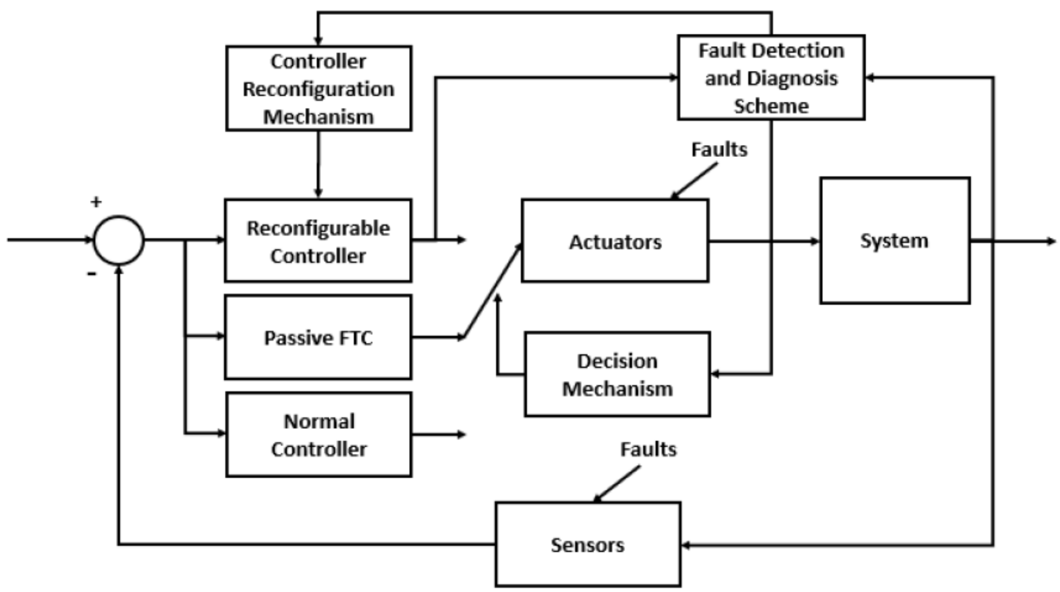

Figure 14: Flow chart of Hybrid FTC systems