



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E S C I E N T I A H O M I N I S S A L U S "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DISEÑO Y SIMULACIÓN DE UN SISTEMA PARA DETECCIÓN
DE FALLAS UTILIZANDO EL FILTRO DE KALMAN COMO
ESTIMADOR DE PARÁMETROS PARA UN TANQUE DE
MEZCLA DE FLUIDOS EN DOS CONFIGURACIONES.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y CONTROL**

GUILLERMO ANTONIO CARPIO MONTEROS
guillermo.carpio@epn.edu.ec

DIRECTOR: ING. DANILO GEOVANNY CHÁVEZ GARCIA, Dr
danilo.chavez@epn.edu.ec

CO- DIRECTOR: ING. OSCAR EDUARDO CAMACHO QUINTERO, PhD
oscar.camacho@epn.edu.ec

Quito, Noviembre 2017

DECLARACIÓN

Yo, Guillermo Antonio Carpio Monteros declaro bajo juramento que el trabajo aquí descrito es de mi autoría: que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, Por su Reglamento y por la normatividad institucional vigente.

Guillermo Antonio Carpio Monteros

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Guillermo Antonio Carpio Monteros, bajo nuestra supervisión.

Ing. Danilo Chavez, Dr
DIRECTOR DEL PROYECTO

Ing. Oscar Camacho, PhD
CO-DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Aprovecho estas líneas para expresar mi agradecimiento a todas las personas que me apoyaron este proceso, a todos mis amigos y familia. También cabe reconocer el esfuerzo y dedicación de mis instructores, el Ing. Oscar Camacho, PhD el Ing. Danilo Chávez, Dr. y el Ing. Marco Herrera, Msc quienes me guiaron durante la realización del trabajo.

DEDICATORIA

A mi familia, con mucho cariño.

CONTENIDO

RESUMEN	viii
PRESENTACIÓN	ix
CAPÍTULO 1	1
MARCO TEÓRICO	1
1.1 DETECCIÓN DE FALLAS	1
1.1.1 NOMENCLATURA	2
1.1.1.1 FALLA	2
1.1.1.2 AVERÍA	2
1.1.1.3 ERROR	2
1.1.1.4 PERTURBACIÓN	3
1.1.1.5 RESIDUO	3
1.1.1.6 SÍNTOMA	3
1.1.1.7 DETECCIÓN DE FALLAS	3
1.1.1.8 AISLAMIENTO DE FALLAS	3
1.1.1.9 IDENTIFICACIÓN DE FALLAS	3
1.1.2 CLASIFICACIÓN DE LOS ESQUEMAS DE DETECCIÓN DE FALLAS.....	3
1.1.2.1 ESQUEMAS DE REDUNDANCIA DE HARDWARE	4
1.1.2.2 ESQUEMAS DE PROCESAMIENTO DE SEÑAL	4
1.1.2.3 ESQUEMAS BASADOS EN MODELO	4
1.1.2.3.1 GENERACIÓN DE RESIDUOS VÍA ESTIMACIÓN DE PARÁMETROS	5
1.1.2.3.2 GENERACIÓN DE RESIDUOS VÍA OBSERVADORES DE ESTADO	6
1.1.2.3.3 GENERACIÓN DE RESIDUOS VÍA ECUACIONES DE PARIDAD	6
1.1.3 MODELADO DE FALLAS	7
1.1.4 FALLAS EN LOS INSTRUMENTOS.....	8
1.2 FILTRO DE KALMAN	9
1.2.1 FILTRO DE KALMAN LINEAL	10
1.2.2 FILTRO DE KALMAN EXTENDIDO	11

1.2.3 MÉTODO DE RUNGE KUTTA	13
1.3 REDES NEURONALES FEEDFORWARD	14
1.3.1 PERCEPTRÓN	14
1.3.2 RED ADALINE	16
1.3.3 PERCEPTRÓN MULTICAPA	16
CAPÍTULO 2	18
DISEÑO Y SIMULACIÓN DEL ALGORITMO	18
2.1 MODELO DE LA PLANTA	18
2.1.1 CONFIGURACIÓN CON UN LAZO DE CONTROL	20
2.1.1 CONFIGURACIÓN CON DOS LAZOS DE CONTROL	21
2.1.3 MODELO DE LOS ACTUADORES Y TRANSMISORES	21
2.2 ALGORITMO DE DETECCIÓN DE FALLAS	22
2.2.1 SIMULACIÓN DE LA PLANTA	23
2.2.2 OBTENCIÓN DE RESIDUOS	26
2.2.2.1 CONFIGURACIÓN CON UN CONTROLADOR	38
2.2.2.2 CONFIGURACIÓN CON DOS CONTROLADORES	30
2.2.3 AISLAMIENTO DE FALLAS.....	31
CAPÍTULO 3	34
PRUEBAS Y RESULTADOS	34
3.1 IDENTIFICACIÓN DE PARÁMETROS	34
3.1.1 CONFIGURACIÓN CON UN CONTROLADOR	34
3.1.1.1 FALLAS EN LA INSTRUMENTACION	36
3.1.1.1.1 FALLA EN EL ACTUADOR	37
3.1.1.1.2 FALLA EN EL SENSOR	39
3.1.1.2 FALLAS EN LA PLANTA	42
3.1.1.2.1 FUGA TÉRMICA	43
3.1.1.2.2 FUGA DE MATERIAL	44
3.1.1.2.3 CAMBIO EN LA DENSIDAD	45
3.1.2 CONFIGURACIÓN CON DOS CONTROLADORES	46
3.1.2.1 FALLAS EN LA INSTRUMENTACION	51
3.1.2.1.1 FALLA EN TRANSMISOR DE TEMPERATURA	51
3.1.2.1.2 FALLA EN TRANSMISOR DE NIVEL	53
3.1.2.1.3 FALLA EN ACTUADOR DE FLUJO FRÍO	56

3.1.2.1.4 FALLA EN ACTUADOR DE FLUJO CALIENTE	59
3.1.2.2 FALLAS EN LA PLANTA	62
3.1.2.2.1 FUGA DE MATERIAL	62
3.1.2.2.2 FALLA EN EL AISLAMIENTO TÉRMICO	63
3.1.2.2.3 CAMBIO EN LA DENSIDAD DEL LÍQUIDO	64
3.2 AISLAMIENTO DE FALLAS.....	65
CAPÍTULO 4	70
CONCLUSIONES Y RECOMENDACIONES	70
4.1 CONCLUSIONES	70
4.2 RECOMENDACIONES	71
REFERENCIAS BIBLIOGRÁFICAS	72
ANEXO	

RESUMEN

La habilidad para detectar una falla de forma temprana es de gran importancia para los procesos industriales. Permite evitar daños en la maquinaria, incrementar la calidad y robustez del sistema o incluso prevenir accidentes.

El objetivo de este trabajo es diseñar y simular un sistema para detección de fallas de un tanque de mezcla de dos fluidos a distintas temperaturas en dos configuraciones, la primera con un controlador de temperatura y la segunda con control de temperatura y nivel.

El sistema de detección de fallas consiste en estimar los parámetros de la planta bajo la suposición de que una falla produce una variación de estos. Para realizar la estimación de parámetros se emplea el filtro de Kalman extendido.

Los parámetros estimados se comparan con los conocidos de la planta para generar residuos y estos son clasificados por una red neuronal del tipo feedforward que entrega información de la falla.

La simulación se lleva a cabo utilizando Matlab y Simulink.

PRESENTACIÓN

El trabajo está dividido en cuatro capítulos.

En el primer capítulo (Marco teórico) comienza exponiendo las bases de la teoría de detección de fallas, así como del Filtro Kalman y su aplicación en la detección de fallas, finalmente se da una breve explicación del funcionamiento de las redes neuronales del tipo feedforward.

El segundo capítulo (Diseño y simulación del algoritmo) presenta el modelo de la planta en ambas configuraciones y diagrama el sistema de detección de fallas que se simulará en el proyecto.

El tercer capítulo (Pruebas y resultados) presenta los resultados de la simulación, comienza con los residuos obtenidos al introducir fallas a la planta en sus dos configuraciones y posteriormente se presenta el resultado del entrenamiento de la red neuronal con una base de datos extensa.

El cuarto capítulo (Conclusiones y recomendaciones) se exponen algunas conjeturas a las que se llegó junto con recomendaciones futuras que podrían aportar a mejorar el sistema, o ayudar a implementar un sistema similar en un caso práctico.

CAPÍTULO 1

MARCO TEÓRICO

En este capítulo se presenta los conceptos básicos de la teoría de detección de fallas basada en modelo, se introduce el filtro de Kalman y el filtro de Kalman Extendido aplicado a la estimación de parámetros, finalmente se exploran las redes neuronales del tipo feedforward y su aplicación en clasificación de patrones.

1.1 DETECCIÓN DE FALLAS

La detección y diagnóstico de fallas tiene una gran importancia en los procesos industriales. Una temprana detección de estas permite evitar daños que pueden ser irreparables para la maquinaria, disminuir el desempeño del sistema de control o reducir la eficiencia del proceso lo que trae como consecuencia la disminución de la producción [1].

El desarrollo de las técnicas de detección y diagnóstico de fallas basado en modelo comenzó a principios de la década de 1970, el uso de las técnicas de estimación de parámetros para detección de fallas en sistemas químicos fue demostrado por Hohmann en 1978 [2], un esquema del desarrollo histórico de las técnicas basadas en modelo para detección y diagnóstico de fallas puede encontrarse en [3], ahí se detalla que en un comienzo las contribuciones a la teoría de detección y diagnóstico de fallas basado en modelo fueron principalmente de la comunidad de control, con métodos basados en observadores o identificación de parámetros. Posteriormente la teoría fue ampliada por métodos de la comunidad de ciencias de la computación con técnicas de inteligencia computacional o métodos basados en conocimiento.

La necesidad de obtener procesos más confiables, eficientes y robustos junto con el incremento de la complejidad de procesos de producción ha motivado el desarrollo de sistemas de diagnóstico y detección de fallas.

El presente trabajo presenta una aplicación del filtro de Kalman extendido como identificador de parámetros para una planta no lineal, un ejemplo de aplicación de este esquema de detección de fallas puede encontrarse en [2]. Otro identificador

de parámetros basado en el filtro de Kalman para sistemas no lineales es el filtro de Kalman Unscented UKF, en [4] se presenta una aplicación del filtro de Kalman Unscented para detección de fallas.

En [5] se desarrolla una aplicación de control tolerante a fallas activo, en este tipo de control se realimenta la información generada por el sistema de detección y diagnóstico de fallas al controlador.

En [6] a través de la información entregada por los sensores de un robot se diferencia los comportamientos de este ante las diferentes fallas.

1.1.1 NOMENCLATURA

En esta sección se presenta la terminología relacionada a detección y diagnóstico de fallas. Es el comité técnico SAFEPROCESS quien discute y trata de buscar definiciones comúnmente aceptadas [2], estas pueden ser encontradas en el diccionario RAM (Reliability, Availability and Maintainability).

1.1.1.1 Falla

Desviación no permitida de al menos una propiedad, característica o parámetro del sistema de su condición aceptable, usual o estándar.

1.1.1.2 Avería

Una interrupción permanente de la habilidad del sistema a desempeñar una función requerida bajo las condiciones de operación especificadas.

1.1.1.3 Error

Una desviación entre un valor medido o computado de una variable de salida y su valor verdadero o patrón.

1.1.1.4 Perturbación

Entrada no controlada actuando en un sistema.

1.1.1.5 Residuo

Indicador de falla basado en una discrepancia entre mediciones y los valores obtenidos basados en el modelo.

1.1.1.6 Síntoma

Un cambio observable de una cantidad con respecto a su comportamiento normal.

1.1.1.7 Detección de Fallas

Determinación de fallas presentes en un sistema y el tiempo de detección.

1.1.1.8 Aislamiento de Fallas

Determinación del tipo, ubicación y tiempo de detección de una falla. Sigue al proceso de la detección de fallas.

1.1.1.9 Identificación de Fallas

Determinación del tamaño y comportamiento variante en el tiempo de una falla. Sigue al proceso de aislamiento de fallas.

1.1.2 CLASIFICACIÓN DE LOS ESQUEMAS DE DETECCIÓN DE FALLAS

Las técnicas de detección y diagnóstico de fallas pueden tradicionalmente clasificarse en tres categorías [3]:

- Esquemas de redundancia de hardware.
- Aproximaciones basadas en procesamiento de señal y
- Esquemas basados en modelo (redundancia analítica).

1.1.2.1 Esquemas de Redundancia de Hardware

Los esquemas basados en hardware consisten en la reconstrucción del proceso utilizando componentes de hardware idénticos al proceso, por ejemplo, emplear múltiples sensores para medir una sola variable, con múltiples sensores al comparar la salida de los componentes redundantes se puede conseguir la detección de fallas.

La ventaja de este tipo de esquema es que, además de robustecer al sistema al disponer de componentes de respaldo, el aislamiento de fallas se consigue directamente pero el incremento en costo de utilizar componentes redundantes limita su aplicación a sistemas más críticos [3][2].

1.1.2.2 Esquemas de Procesamiento de Señal

Asumiendo que algunas señales del proceso llevan información acerca de las fallas de interés y que esta información se presenta en la forma de síntomas, se puede lograr diagnóstico de fallas por un procesamiento de señales adecuado. Los síntomas típicos son funciones en el dominio del tiempo como magnitudes, valores aritméticos o medias cuadráticas, valores límites, momentos estadísticos de la amplitud o funciones en el dominio de la frecuencia [3].

Para el análisis de señales periódicas estacionarias se puede utilizar un filtro pasa bandas o análisis de Fourier, las señales periódicas no estacionarias pueden ser analizadas por ejemplo con la transformada wavelet [7]. Este método es empleado en [1] para detección de fallos en plantas químicas.

1.1.2.3 Esquemas Basados en Modelo

También llamados de redundancia analítica porque se ejecuta un modelo de software paralelo al proceso. Un sistema de detección de fallas estándar está compuesto de un generador de residuos y una función de detección como se muestra en la Figura 1.1.

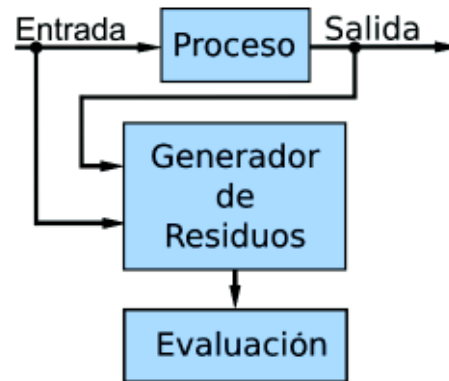


Figura 1.1 Detección de Fallas Basado en Modelo

Los esquemas basados en modelo más comunes son [3][2]:

- Generación de residuos con identificación de parámetros.
- Generación de residuos con observadores de estado y estimación de estado.
- Generación de residuos con ecuaciones de paridad.

1.1.2.3.1 Generación de Residuos Vía Estimación de Parámetros

Este método se basa en la suposición de que las fallas se reflejan en los parámetros del sistema y la idea básica es estimar estos parámetros en línea usando métodos de estimación de parámetros bien conocidos [2], por ejemplo, el filtro de Kalman o mínimos cuadrados recursivos.

Los parámetros obtenidos se comparan con los del modelo de referencia que pueden ser obtenidos inicialmente en condiciones libres de fallas, una discrepancia indica la presencia de una falla.

Cuando las mediciones son afectadas por ruido se puede generar una falsa alerta de falla, para limitar este tipo de errores se puede utilizar un filtro de Kalman para la estimación de parámetros [2]. Se debe hacer un compromiso entre la capacidad de detectar una falla y la capacidad de eliminar el ruido.

1.1.2.3.2 Generación de Residuos Vía Observadores de Estado.

Cuando se conoce con exactitud los parámetros del proceso se puede aplicar observadores de estado o estimadores de estado en caso de que el ruido en el sistema sea significativo [2].

En este tipo de generación de residuos las fallas deben modelarse como variaciones en los estados Δx .

Una alternativa es utilizar observadores de salida también se llaman observadores de entrada desconocida. Estudios en la generación de residuos desacoplados de entradas desconocidas recibió considerable atención durante la segunda mitad de la década de 1980[2]. Una transformación lineal $E(t)$ es utilizada para obtener nuevas variables de estado que son independientes de las entradas desconocidas, entonces los residuos dependen solamente de las fallas aditivas.

1.1.2.3.3 Generación de Residuos Vía Ecuaciones de Paridad.

La idea básica de las relaciones de paridad es proveer de un chequeo de la paridad (consistencia) de las mediciones adquiridas del sistema monitoreado [2].

Se considera el siguiente modelo del sistema con sus fallas:

$$dx(t) = Ax(t) + Bu(t) + Fv(t) + Lf_l(t) \quad (1.1)$$

$$y(t) = Cx(t) + Nn(t) + Lf_m(t) \quad (1.2)$$

donde:

$v(t)$ y $n(t)$ son señales de perturbación,

f_l y f_m son señales de falla aditivas.

Un método directo de detección de fallas es tomar a un modelo fijo y correrlo en paralelo con el proceso formando un error de salida [8].

$$r'(s) = [G_p(s) - G_M(s)]u(s) \quad (1.3)$$

Si el proceso $G_p(s)$ es igual al modelo utilizado $G_M(s)$, entonces el error de salida es igual a las fallas aditivas de entrada y salida, la siguiente es la ecuación de paridad que se genera.

$$r'(s) = G_p(s) f_l(s) + f_m(s) \quad (1.4)$$

1.1.3 MODELADO DE FALLAS

Se considera el siguiente modelo de entrada y salida [3].

$$y(s) = G_{yu}(s)u(s) + G_{yd}(s)d(s) + G_{yf}(s)f(s) \quad (1.5)$$

G_{yd} es una matriz de transferencia conocida, f es un vector desconocido que representa las fallas, G_{yd} es la matriz transferencia de las perturbaciones y d el vector de entradas desconocidas. Una realización en espacio de estado para 1.5 es la siguiente.

$$\begin{aligned} dx &= Ax + Bu + E_d d + E_f f \\ y &= Cx + Du + F_d d + F_f f \end{aligned} \quad (1.6)$$

E_f y F_f indican el lugar de ocurrencia de una falla y su influencia sobre los componentes del sistema.

Las fallas se dividen en tres categorías:

- fallas en sensor fs .
- fallas en actuador fa .
- fallas en el proceso fp .

Una falla en sensor se puede modelar con $F_f = I$ con lo que se obtiene:

$$y = C \cdot x + D \cdot u + F_d d + fs \quad (1.7)$$

Para una falla en el actuador se ajusta $E_f = B$, $F_f = D$, lo que genera el siguiente resultado:

$$dx = Ax + B(u + fa) + E_d d, y = Cx + D(u + fa) + F_d d \quad (1.8)$$

Las fallas en el proceso dependen de su ubicación. Para un sistema con fallas en sensores, actuadores y en el proceso se tiene:

$$f = [fa; fp; fs], E_f = [B \ E_p \ 0], F_f = [D \ F_p \ I] \quad (1.9)$$

Las fallas descritas en 1.6 se denominan fallas aditivas [3]. La ocurrencia de este tipo de fallas no afecta la estabilidad del sistema, en la práctica las fallas en el proceso, sensores y actuadores cambia los parámetros del sistema, una falla de ese tipo es una falla multiplicativa. Una falla multiplicativa puede ser modelada como una falla aditiva, pero se debe tener en cuenta que la función de falla es una función del estado y las variables de entrada del sistema y por lo tanto afectará la estabilidad del sistema.

1.1.4 FALLAS EN LOS INSTRUMENTOS

Los sistemas de medición son elementos de transferencia dinámicos en los que solamente la variable de salida es accesible [8].

El comportamiento estático de un sensor puede ser lineal.

$$Vp(t) = c_0 + c_1M(t) \quad (1.10)$$

o no lineal

$$Vp(t) = c_0 + c_1M(t) + c_2M(t)^2 \quad (1.11)$$

$Vp(t)$ es la variable dependiente, $M(t)$ la independiente y los coeficientes c_n son constantes. El comportamiento dinámico del sensor puede, frecuentemente, ser aproximado por un modelo lineal de primer orden para pequeñas variaciones de la variable independiente.

$$\tau_v \frac{dVp(t)}{dt} + Vp(t) = \left(\frac{Vp_{max} - Vp_{min}}{M_{max} - M_{min}} \right) (M(t) - M_{min}) + Vp_{min} \quad (1.12)$$

Los valores Vp_{max} , Vp_{min} , M_{max} y M_{min} son los valores mínimos y máximos de las variables dependiente e independiente respectivamente. El instrumento modelado de esta manera está libre de falla cuando la relación entre valores reales dentro del campo de medida y los valores de lectura es lineal.

Conforme se van presentando desviaciones en la recta de calibración se tendrán desviaciones de la recta de valor real- valor medido, lo que se traduce en fallas del instrumento Las fallas de medida del instrumento pueden ser de cuatro tipos: Falla en el span, Falla en la zona Muerta del span, Falla en la calibración de cero y Falla en la zona muerta de cero [1].

Variaciones en Vp_{max} , Vp_{min} , M_{max} y M_{min} representan fallas en el instrumento. Falla en el span se produce cuando se tiene una variación en Vp_{max} , falla en la zona muerta del span se presenta cuando se tiene una variación en M_{max} , falla en la calibración de cero se presenta cuando se tiene una variación en Vp_{min} , falla en la zona muerta de cero se presenta con una desviación en M_{min} .

La Figura 1.2 muestra los diferentes tipos de falla

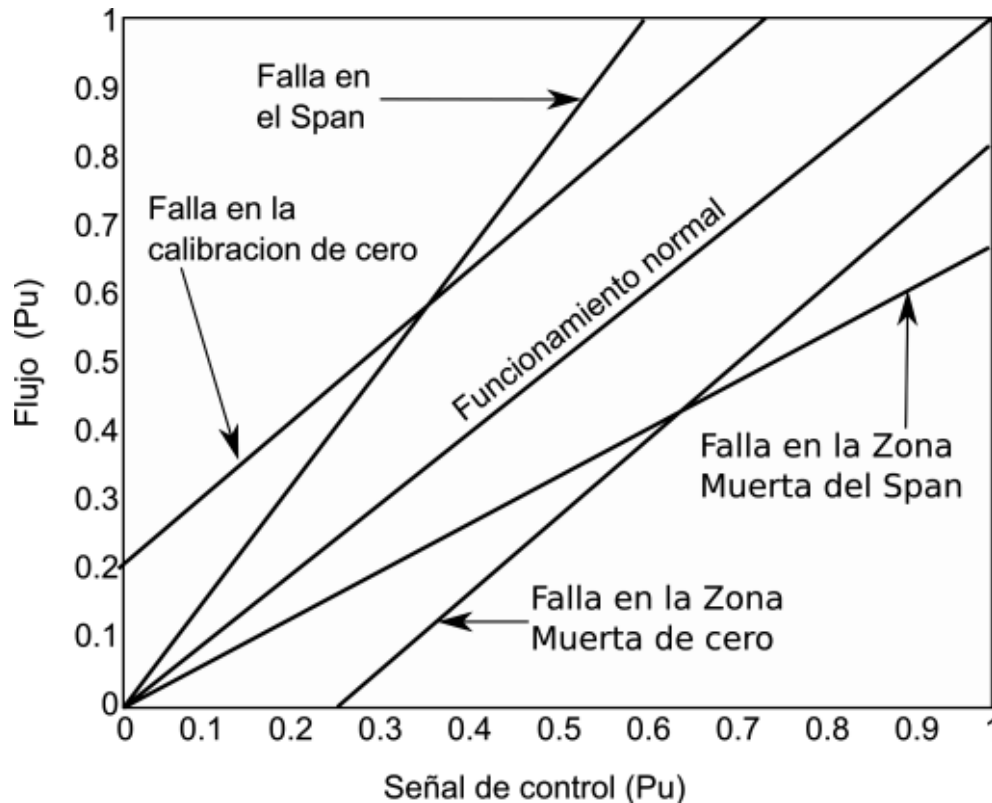


Figura 1.2 Fallas de medida en los Instrumentos [1]

1.2 FILTRO DE KALMAN

El filtro de Kalman fue presentado por primera vez en 1960 por R. E. Kalman en "A New Approach to Linear Filtering and Prediction Problems", teóricamente ha sido llamado estimador de mínimos cuadrados lineal [9].

1.2.1 FILTRO DE KALMAN LINEAL

Se asume que tenemos un proceso a estimar que se puede modelar en la forma:

$$X_{k+1} = \Phi_k X_k + w_k \quad (1.13)$$

$$z_k = H_k x_k + v_k \quad (1.14)$$

donde:

X_k = Vector de estado del proceso en el tiempo t_k

z_k = Vector de mediciones en el tiempo t_k

H_k = Matriz que conecta al vector de estado con el vector de mediciones

Φ_k = Matriz que relaciona X_k con X_{k+1}

v_k = Error de medición

w_k = Ruido de entrada al vector de estado

Las matrices de covarianza para los vectores w_k y v_k están dados por:

$$E[w_k w_i^T] = Q_k, i = k \quad (1.14)$$

$$E[v_k v_i^T] = R_k, i = k \quad (1.15)$$

$$E[w_k v_i^T] = 0, \text{ para todo } k \text{ e } i \quad (1.16)$$

El error de estimación a priori está dado por:

$$e_k^- = x_k - \hat{x}_k^- \quad (1.17)$$

$$P_k^- = E[e_k^- e_k^{-T}] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \quad (1.18)$$

El algoritmo recursivo del filtro de Kalman está compuesto por las siguientes ecuaciones [13]:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (1.19)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \quad (1.20)$$

$$P_k = (I - K_k H_k) P_k^- \quad (1.21)$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \quad (1.22)$$

Para entrar al lazo del algoritmo se asume que se conoce la matriz P_k^- y \hat{x}_k^- el vector de estados, después se calcula la ganancia K (eq 1.15) y se actualiza el

vector de estado utilizando la medición z (eq 1.20), el siguiente paso es calcular la covarianza del error P_k (eq 1.21) para finalizar se proyecta P_{k+1} (eq 1.22) y x (eq 1.8) y se procede a repetir el lazo como se indica en la Figura 1.3.

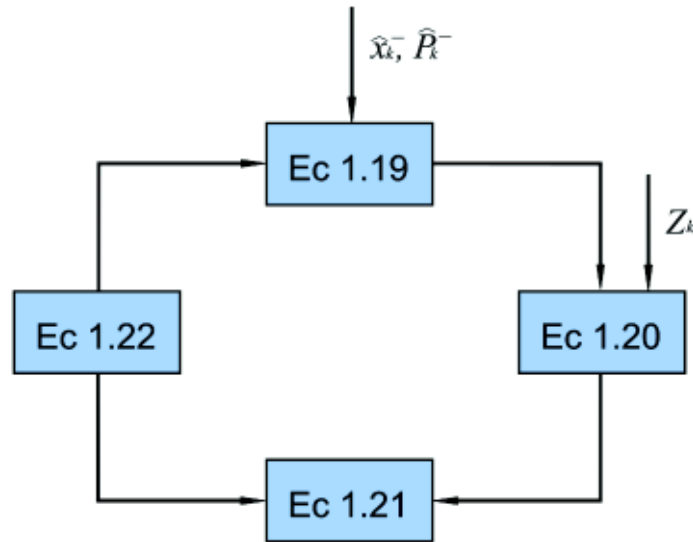


Figura 1.3 Filtro de Kalman

1.2.2 FILTRO DE KALMAN EXTENDIDO

El filtro de Kalman extendido asume que el estado verdadero del sistema es cercano a la estimación por lo que la dinámica del modelo no lineal puede ser expandido utilizando series de Taylor de primer orden alrededor del estimado actual [11], esto requiere que se calcule una matriz jacobiana en cada iteración del algoritmo, a pesar de no ser precisamente “óptimo” el filtro de Kalman extendido ha sido aplicado con éxito a lo largo de varios años [12].

En caso de que las estimaciones se alejen demasiado de los valores verdaderos del sistema el filtro de Kalman extendido al no ser óptimo puede divergir.

La ecuación 1.23 a continuación especifica el cálculo de la matriz jacobiana.

$$f_{ij} = \left[\frac{df_i(t, \hat{x}(t), u(t))}{d\hat{x}_j} \right]_{t=tk-1} \quad (1.23)$$

El filtro de Kalman puede ser utilizado para la estimación de parámetros. El vector de estado $x(k)$ se aumenta con un vector de parámetros $\theta(k)$ obteniéndose el siguiente vector de estados:

$$\begin{pmatrix} \hat{x}(k+1) \\ \theta(k+1) \end{pmatrix} = \begin{pmatrix} f(\hat{x}(k), \theta(k), u(k)) \\ \theta(k) \end{pmatrix} + \begin{pmatrix} Fn(k) \\ \xi(k) \end{pmatrix} \quad (1.24)$$

el vector de parámetros $\theta(k)$ obedece a la dinámica: $\theta(k+1) = \theta(k) + \xi(k)$ donde, se puede observar que los parámetros están modelados como perturbados por ruido blanco, si no fuera este el caso el filtro asumiría que estos valores se conocen exactamente y no los ajustaría durante el filtrado [13]. Las ecuaciones para el Filtro de Kalman Extendido para un sistema dinámico continuo-discreto son las siguientes [14]:

$$\hat{x}(t_k|t_{k-1}) = \hat{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} f(\tau, x(\tau), u(\tau))d\tau \quad (1.25)$$

$$\hat{P}(t_k|t_{k-1}) = A(t_k)P(t_k)A(t_k)^T + Q(t_k) \quad (1.26)$$

Corrección:

$$K(t_k) = P(t_k|t_{k-1})H(t_k)^T [H(t_k)P(t_k|t_{k-1})H(t_k)^T + R(t_k)]^{-1} \quad (1.27)$$

$$\hat{x}(t_k) = \hat{x}(t_k|t_{k-1}) + K(t_k)[y(t_k) - h(t_k, \hat{x}(t_k|t_{k-1}), u(t_k))] \quad (1.28)$$

$$P(t_k) = (I - K(t_k)H(t_k))P(t_k|t_{k-1})(I - H(t_k)^TK(t_k)^T) + K(t_k)R(t_k)K(t_k)^T \quad (1.29)$$

donde A es igual a la exponencial de la matriz jacobiana F multiplicada por T , el tiempo entre ejecuciones.

En muchas aplicaciones de estimación no se trabaja con un modelo de planta lineal, cuando se tiene un modelo descrito por ecuaciones diferenciales no lineales se puede emplear el filtro de Kalman extendido para resolver el problema de estimación.

1.2.3 MÉTODO DE RUNGE KUTTA

En el presente trabajo este método es utilizado para realizar la resolución de las ecuaciones diferenciales, necesaria en la obtención del vector de estados estimado a partir del modelo, como se requiere en la ecuación 1.26.

El método de Runge-Kutta evalúa la función diferencial $f(t,z)$ en más puntos, este método es bastante fácil de programar y es uno de los métodos más utilizados para resolver problemas de valor inicial.

A continuación, se presentan las ecuaciones que describen al método de Runge-Kutta explícito con s etapas como se las encuentra en [15]

$$z_1 = y_n,$$

$$z_2 = y_n + ha_{2,1}f(t_n, z_1),$$

$$z_3 = y_n + h[a_{3,1}f(t_n, z_1) + a_{3,2}f(t_n + c_2h, z_2)],$$

↓

$$z_s = y_n + h[a_{s,1}f(t_n, z_1) + a_{s,2}f(t_n + c_2h, z_2) + \dots + a_{s,s-1}f(t_n + c_{s-1}h, z_{s-1})], \quad (1.30)$$

$$y_{n+1} = y_n + h[b_1f(t_n, z_1) + b_2f(t_n + c_2h, z_2)$$

$$+ \dots + b_{s-1}f(t_n + c_{s-1}h, z_{s-1}) + b_s f(t_n + c_s h, z_s)] \quad (1.31)$$

Los coeficientes a, b y c empleados en la ecuación anterior se presentan usualmente utilizando una tabla Butcher como la mostrada en la Figura 1.4.

$0 = c_1$				
c_2	$a_{2,1}$			
c_3	$a_{3,1}$	$a_{3,2}$		
	\vdots			
c_s	$a_{s,1}$	$a_{s,2}$	$a_{s,s-1}$	
	b_1	b_2	b_{s1}	b_s

Figura 1.4 Tabla butcher, utilizada para mostrar los coeficientes del método Runge-Kutta [15].

los coeficientes C_i y $a_{i,j}$ deben satisfacer las siguientes condiciones.

$$\sum_{j=1}^{i-1} a_{i,j} = c_i, i = 2, \dots, s. \quad (1.32)$$

La ventaja del método de Runge-Kutta con respecto a las series de Taylor es que no se necesita obtener derivadas de orden superior que son tediosas de calcular y pueden inducir errores.

1.3 REDES NEURONALES FEEDFORWARD

Una red neuronal artificial es un modelo matemático que imita la función de una neurona biológica, transmite una señal desde sus entradas y la procesa a través de varias conexiones para obtener en su salida un resultado esperado [16].

La arquitectura de una red neuronal feedforward se caracteriza porque su flujo de información es en una sola dirección, desde su capa de entrada hacia la capa de salida.

Los principales tipos de red neuronal que pertenecen a esta arquitectura son el perceptrón y el adaline en redes de una sola capa neuronal, cuyos algoritmos de entrenamiento son la regla de Hebb y la regla Delta respectivamente [16]. Para redes neuronales de múltiples capas los ejemplos de red principales que pertenecen a esta arquitectura son la red de función de base radial y el perceptrón multicapa.

1.3.1 PERCEPTRÓN.

El perceptrón, creado por Rosenblatt, es la configuración de red neuronal más simple jamás creada, su propósito era implementar un modelo computacional basado en la retina. Una de sus aplicaciones era identificar patrones geométricos [16].

El principio de operación del perceptrón es bastante simple, el valor compuesto por la suma de todas las entradas x_i multiplicadas por un peso w_i menos el valor de un umbral θ es utilizado por la función de activación. La ecuación matemática resultante es descrita por la siguiente expresión.

$$u = \sum_{i=1}^n w_i x_i - \theta \quad (1.33)$$

La función de activación en un perceptrón suele ser una función paso o paso bipolar de tal manera que su salida y puede tomar dos valores.

El perceptrón cumple la función de un discriminador lineal, un perceptrón con dos entradas separará a las clases en las que se encuentran en las diferentes direcciones desde una línea. un perceptrón con tres entradas separará a las clases utilizando un plano.

$$y = g(u) \quad (1.34)$$

En la Figura 1.5 se presenta un diagrama del perceptrón de capa simple.

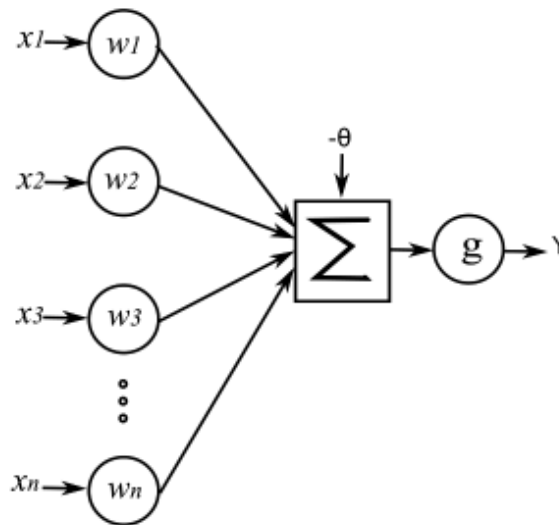


Figura 1.5 Red Neuronal Perceptrón de capa simple

Se concluye que para un perceptrón de capa simple sea utilizado como clasificador las clases del problema deben ser linealmente separables. Este principio se llama teorema de convergencia del perceptrón y fue presentado por Minsky en 1969[16].

El proceso de entrenamiento del perceptrón se describe en la siguiente ecuación:

$$w^{actual} = w^{anterior} + n \cdot (d^{(k)} - y) \cdot x^{(k)} \quad (1.35)$$

Donde:

$w = [\theta \ w_1 \ w_2 \ \dots \ w_n]^T$ es el vector que contiene los pesos y el valor umbral.

$x^{(k)} = [-1 \ x_1^{(k)} \ x_2^{(k)} \ \dots \ x_n^{(k)}]^T$ es la muestra de entrenamiento k .

$d^{(k)}$ es el valor deseado para la muestra de entrenamiento k .

y es la salida producida por el perceptrón.

n es una constante que define la velocidad de aprendizaje $0 < n < 1$.

1.3.2 RED ADALINE

La red ADALINE es el nombre simplificado de elemento lineal adaptativo, tiene el mismo funcionamiento que un perceptrón de capa simple, descrito en la ecuación 1.33, pero se diferencia en el proceso de entrenamiento. En esta red se aplica la regla delta que efectúa una minimización del error cuadrado entre el valor a la salida de la red y el valor deseado [16].

Para p muestras de entrenamiento se tiene la siguiente expresión de error en función de los pesos w :

$$E(w) = \frac{1}{2} \sum_{k=1}^p (d^{(k)} - u)^2 = \frac{1}{2} \sum_{k=1}^p (d^{(k)} - (w^T x^{(k)} - \theta))^2 \quad (1.36)$$

Aplicando el operador gradiente al error se obtiene la siguiente expresión:

$$\Delta E(w) = \sum_{k=1}^p (d^{(k)} - (w^T x^{(k)} - \theta)) (-x^{(k)}) \quad (1.37)$$

El ajuste de los pesos w debe realizarse se ejecuta en la dirección opuesta a la del gradiente y se ajusta con un valor η .

1.3.3 PERCEPTRÓN MULTICAPA.

Un perceptrón multicapa tiene al menos una red neuronal intermedia que se coloca entre la capa de entrada y la de salida, por lo tanto, los perceptrones multicapa están compuestos de al menos dos capas neuronales [16].

En la Figura 1.6 se presenta la arquitectura del perceptrón multicapa que cuenta con dos capas neuronales escondidas aparte de la capa de entrada y salida, como se puede ver en esta figura, las señales de entrada se propagan de capa a capa hasta la salida.

El proceso de entrenamiento del perceptrón multicapa consiste en dos etapas, en la primera se propaga el valor de las entradas capa por capa hasta alcanzar la capa de salida sin modificar ningún peso w . Esta etapa se llama etapa de propagación hacia adelante, La segunda etapa se llama de propagación hacia atrás, en esta se computará el error considerando el valor deseado y el valor obtenido en la primera etapa para modificar los pesos w , en las capas intermedias a diferencia de la capa de salida, no se puede acceder al valor deseado por lo que el ajuste de los pesos w debe realizarse a través de estimaciones de los errores producidos en la capa siguiente [16].

En aplicaciones de clasificación, la ventaja del perceptrón multicapa sobre el de capa simple es que se puede clasificar los patrones en superficies más complejas como por ejemplo una superficie convexa.

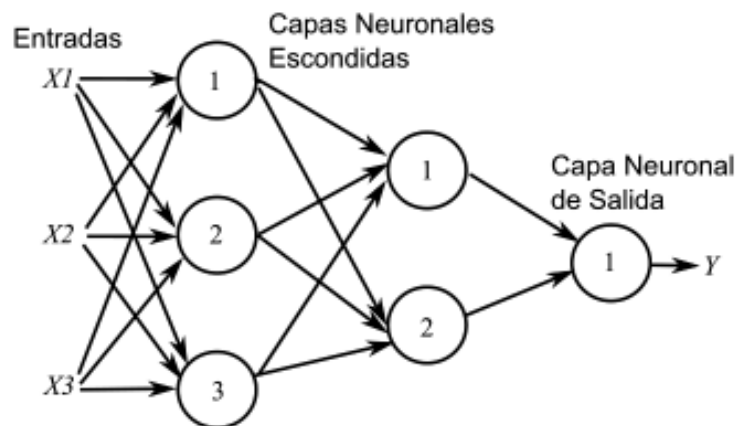


Figura 1.6 Perceptrón Multicapa

CAPÍTULO 2

DISEÑO Y SIMULACIÓN DEL ALGORITMO

Este capítulo comienza presentado el modelo de la planta en las dos configuraciones analizadas, después se introduce la estructura del algoritmo de detección de fallas, que está compuesto por dos etapas, la primera etapa consiste en la generación de residuos empleando el filtro de Kalman extendido y la segunda etapa consiste en la evaluación de los residuos, generados en la etapa anterior, mediante el uso de una red neuronal.

2.1 MODELO DE LA PLANTA

La planta utilizada en el trabajo es un tanque de mezclado de fluidos a distintas temperaturas en dos configuraciones, la primera configuración tiene un lazo de control de temperatura y la segunda configuración tiene dos lazos de control, uno de temperatura y otro de nivel en el tanque. Esta planta es similar a la que se utiliza en [1] donde se la emplea en detección de fallas con un esquema de análisis de señal, además un modelo de tanque de mezclado se encuentra en [17] donde es linealizado para el diseño de controladores.

Asumiendo:

- Un área del tanque A_c Constante.
- Propiedades ρ (densidad) y C_p (Coeficiente Calórico) de líquido constantes.
- Un buen aislamiento térmico del tanque.
- Que los contenidos se mezclan bien.
- Que el flujo de salida del tanque es $F = K\sqrt{h}$ (K es la constante del flujo de salida).

Se obtienen las siguientes ecuaciones del balance de energía y de material del tanque [17].

$$Ac \frac{dh}{dt} = \frac{(W_1 + W_2)}{\rho} - K\sqrt{h} \quad (2.1)$$

$$\rho C_{p3} A_c \frac{d(hT)}{dt} = C_{p1} W_1 T_1 + C_{p2} W_2 T_2 - C_{p3} \rho K \sqrt{h} T \quad (2.2)$$

Se tiene un sistema no lineal, por la presencia de los términos de raíz cuadrada y la multiplicación hT , multivariable que está constituido por dos ecuaciones diferenciales.

En [18] se puede encontrar valores de estado estable para las variables y parámetros de un tanque de mezclado, a continuación, se presenta un resumen de estos en la tabla 2.1.

Tabla 2.1 Variables y parámetros de estado estacionario de la planta.

Variable	Valor	Descripción
W_1	250 lb/min	Flujo de masa del fluido caliente.
W_2	191.17 lb/min	Flujo de masa del fluido fría.
C_{p1}	0.8 Btu/lb-°F	Capacidad calorífica a presión constante.
C_{p2}	1.0 Btu/lb-°F	Capacidad calorífica a presión constante.
C_{p3}	0.9 Btu/lb-°F	Capacidad calorífica a presión constante.
T_1	250 °F	Temperatura del fluido caliente.
T_2	50 °F	Temperatura del fluido Frío.
ρ	62.4 lb /pie ³	Densidad de los fluidos.
ΔP_v	16 psi	Caída de presión en la válvula.

C_{VL}	12 gpm / $\text{psi}^{1/2}$	Coeficiente de flujo de la planta.
τ_{vp}	0.4 min	Constante de tiempo del actuador.

2.1.1 CONFIGURACIÓN CON UN LAZO DE CONTROL

Las ecuaciones 2.1 y 2.2 describen el comportamiento dinámico para las dos configuraciones de la planta, estas se diferencian en la cantidad de instrumentos que los componen.

En la Figura 2.1 se muestra el diagrama de la planta en la configuración con un controlador. La nomenclatura TT en la figura significa transmisor de temperatura, LT es el transmisor de nivel.

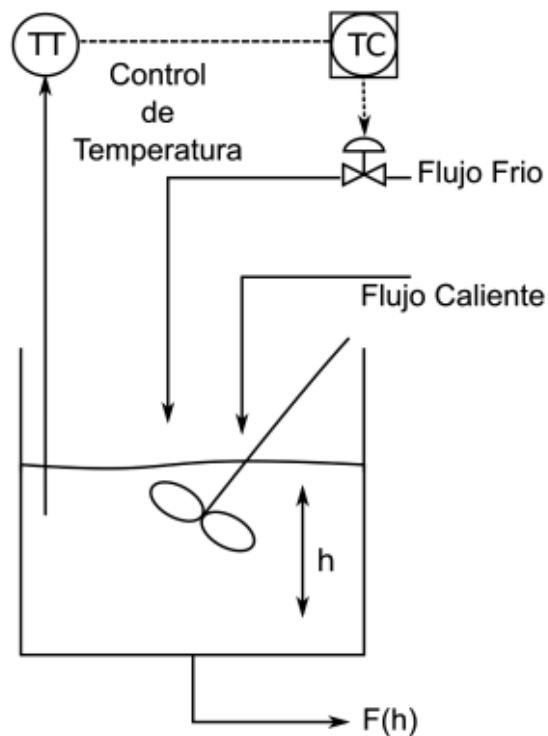


Figura 2.1 Configuración de un controlador.

Esta configuración tiene solamente un lazo de control para regular la temperatura de la mezcla, como se tiene dos flujos a distinta temperatura, existen dos opciones para cumplir el objetivo de control, en este caso se emplea un actuador

para controlar el flujo de entrada frío. Para la configuración en la siguiente sección al tener dos lazos de control se tiene cuatro opciones para cumplir con los objetivos de control. En esta configuración la planta cuenta con un sensor de temperatura de la mezcla que es la variable controlada.

2.1.1 CONFIGURACIÓN CON DOS LAZOS DE CONTROL

La segunda configuración realiza control de temperatura y nivel en la mezcla por lo tanto en esta configuración como se observa en la Figura 2.2 tiene dos sensores uno de temperatura y uno de nivel y dos actuadores, uno para cada flujo, el actuador de flujo frío se emplea en el control de temperatura y el de flujo caliente en el control de nivel.

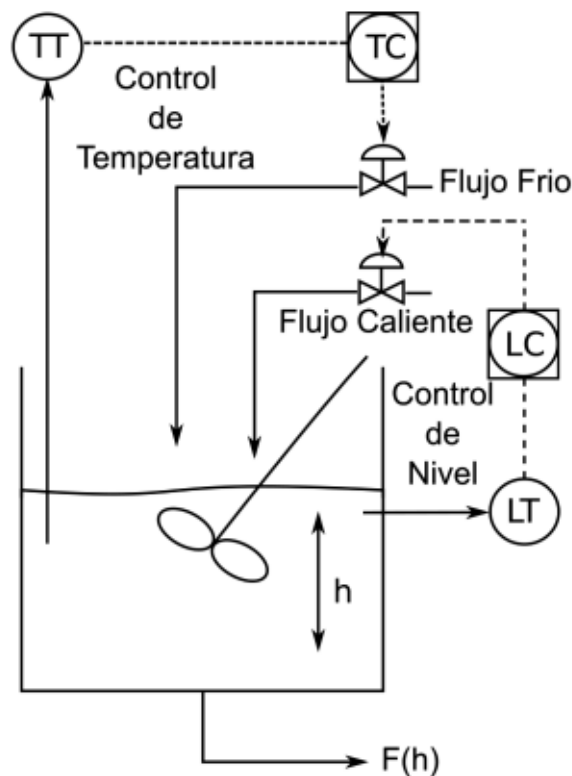


Figura 2.2 Configuración de dos controladores.

2.1.3 MODELO DE LOS ACTUADORES Y TRANSMISORES

La dinámica de las válvulas es considerada como un sistema de primer orden, en las ecuaciones 2.3 y 2.4 se expone esta dinámica como presenta en [10].

Posición de la válvula:

$$\frac{dV_p(t)}{dt} = \frac{1}{\tau_{vp}} [m(t) - V_p(t)] \quad (2.3)$$

$m(t)$ es la señal de control que recibe el actuador y V_p es la posición del actuador.

Ecuación de la válvula:

$$W(t) = \frac{500}{60} C_{VL} V_p \sqrt{\Delta P_v} \quad (2.4)$$

los valores de las constantes considerados en la ecuación 2.4 se encuentran en la Tabla 2.1 junto con sus explicaciones.

La siguiente ecuación representa el modelo matemático de los transmisores.

$$S_o = S_i - \tau_T \frac{dS_o}{dt} + R \quad (2.5)$$

R es una señal de ruido que afecta el desempeño de los transmisores. τ_T es la constante de tiempo del transmisor.

2.2 ALGORITMO DE DETECCIÓN DE FALLAS

Se utiliza un esquema de detección de fallas basado en modelo con generación de residuos por medio identificación de parámetros.

En caso de que existan fallas en el sistema, se considera que los parámetros obtenidos en la etapa de identificación van a divergir de los conocidos del modelo lo que genera residuos. Al tratarse de una planta no lineal se emplea el Filtro de Kalman Extendido para realizar la identificación de parámetros.

Los residuos dependen de la falla que los genera y deben ser analizados para poder determinar en qué elemento se originó la falla. Para realizar el análisis de residuos se utiliza redes neuronales que clasifican a los residuos en los diferentes tipos de falla.

En la Figura 2.3 se puede observar un esquema en el que se ilustra la estructura del sistema de detección de fallas. Como se observa en la figura la etapa de generación de residuos tiene como entradas la salida de la planta que es leída por los transmisores y las señales de control de los actuadores. El filtro de Kalman extendido no incluye el modelo de los actuadores por lo que las entradas $W1$ y $W2$

son desconocidas, para solucionar este problema se utiliza las señales de control junto al modelo de las válvulas para obtener los flujos.

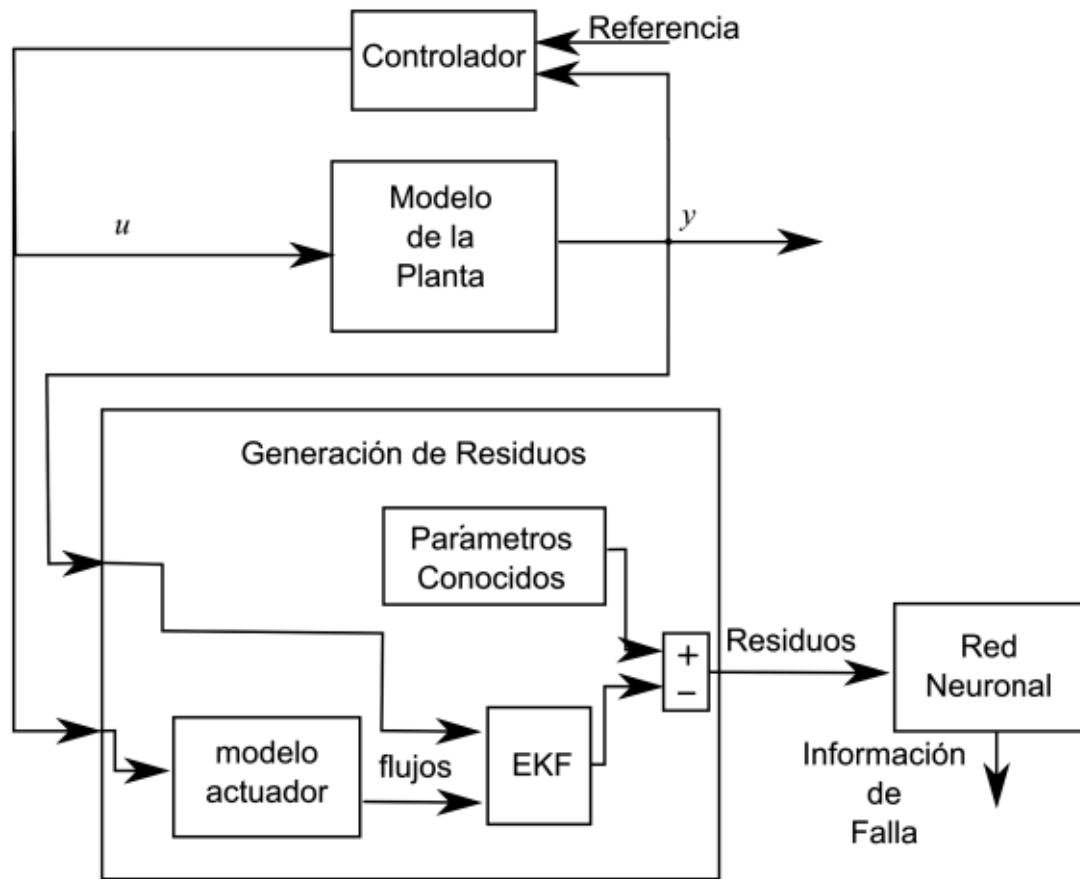


Figura 2.3 Estructura del sistema de detección de fallas.

Las redes neuronales que se utilizan son del tipo feedforward, para que el algoritmo pueda funcionar el proceso de entrenamiento debe realizarse con datos de entrenamiento obtenidos previamente.

Este esquema de detección requiere que el modelo de la planta sea conocido, para compararlo con los valores obtenidos en la estimación de parámetros, en caso de no conocer los parámetros de la planta se puede ejecutar el algoritmo de identificación de parámetros bajo condiciones libres de falla para obtener los parámetros que servirán para la comparación.

2.2.1 SIMULACIÓN DE LA PLANTA

La simulación de la planta se lleva a cabo utilizando un bloque de función definido por el usuario en Simulink de Matlab que contiene las ecuaciones diferenciales

que describen al comportamiento de la planta presentadas en este capítulo, las salidas de este bloque se integran utilizando la herramienta de integral y se realimentan al bloque para realizar el siguiente cálculo. Para el controlador se utiliza el bloque PID de Simulink.

La dinámica del sistema se simula al calcular los estados en pasos de tiempo sucesivos, estos pasos de. Un solucionador aplica algún método numérico para resolver el grupo de ecuaciones diferenciales, este determina el paso de tiempo de la simulación.

Los solucionadores de ecuaciones se clasifican de acuerdo con el tipo de paso de tiempo que utilizan para los cálculos y a la naturaleza de los estados en el modelo. Los tipos de paso puede ser fijo o variable y la naturaleza del estado puede ser continua o discreta.

El solucionador utilizado es automáticamente seleccionado por el programa de simulación.

En las siguientes figuras se muestran los modelos de simulación de la planta para ambas configuraciones.

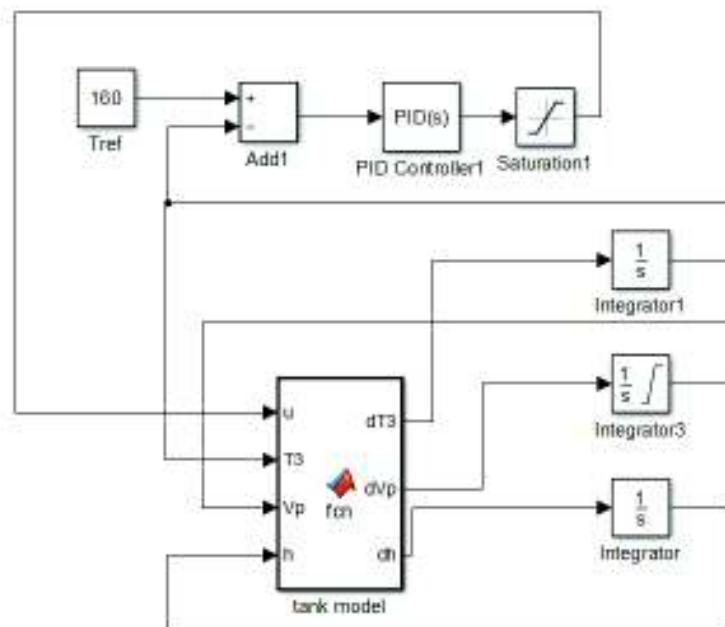


Figura 2.4 Simulación de la planta, configuración de un controlador.

Como se observa en la Figura 2.4 en el bloque de posición de la válvula también se integra y es realimentada al bloque por lo que las ecuaciones que representan su comportamiento dinámico presentadas en la sección 2.1.3 también se incluyen en la simulación.

La configuración de un controlador solamente necesita un lazo de control, el de temperatura y un actuador, en la Figura 2.5 se presenta el modelo de simulación para esta configuración, este bloque de simulación para el tanque de mezclado tiene cuatro entradas en lugar de seis y tres salidas en lugar de cuatro.

La entrada no controlada se mantiene en un valor fijo.

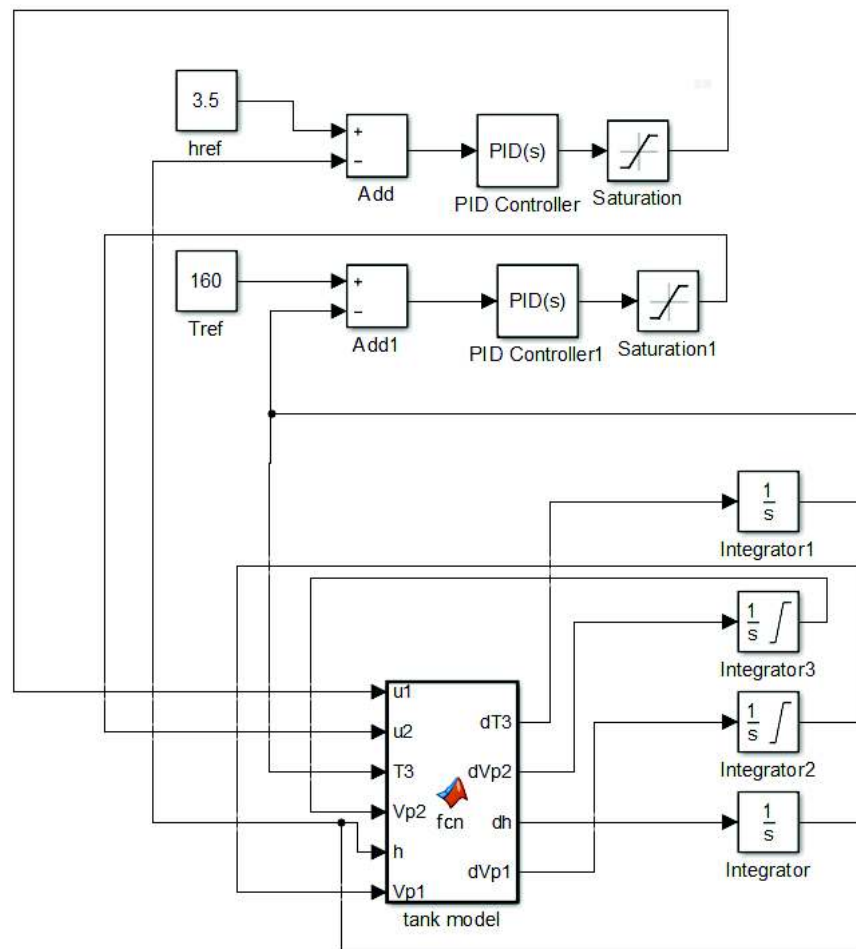


Figura 2.5 Simulación de la planta, configuración de dos controladores.

Los modelos de simulación de transmisores se crearon utilizando un subsistema de simulink que toma de entrada el valor de la simulación de la planta que es

medido y tiene a la salida esta señal después de pasar por un sistema de primer orden, además se le suma señal de ruido gaussiano que permitirá probar la sensibilidad del algoritmo frente al ruido.

Las fallas en transmisores son simuladas utilizando una ganancia adicional o sumando un valor en la entrada de señal desde la planta.

En la configuración de dos controladores se emplean dos subsistemas de transmisor, los valores de ruido se ajustan a un porcentaje del valor de la variable que se conecta al subsistema.

En la Figura 2.6 se presenta el subsistema de transmisor.

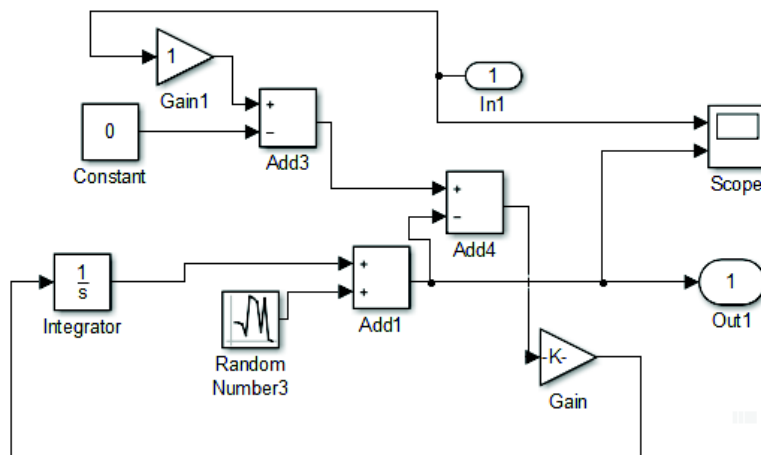


Figura 2.6 Simulación de Transmisor

2.2.2 OBTENCIÓN DE RESIDUOS

Para ejecutar el algoritmo de obtención de residuos se emplea al igual que en la simulación de la planta un bloque de función definido por el usuario, pero en este caso el bloque debe ejecutarse de manera periódica con un tiempo entre ejecuciones fijo definido de acuerdo con la aplicación por lo cual se emplea un bloque que se ejecuta con el filo de subida de una señal externa. El valor máximo del período de esta señal no debe exceder una fracción de las constantes de tiempo de la planta, un valor demasiado bajo está limitado por el poder de procesamiento disponible, sin embargo, el algoritmo de identificación de

parámetros tiene un mejor desempeño cuando se disminuye el tiempo entre ejecuciones por lo que se utiliza un valor de diez milisegundos.

El Filtro de Kalman Extendido que se utiliza en la obtención de parámetros toma como entrada los flujos frío y caliente, estos no son accesibles al algoritmo a no ser que se cuente con instrumentos de medición en los actuadores, en este caso se utiliza el modelo de las válvulas junto con las señales de control $u1$ y $u2$ a las cuales si se puede acceder para obtener los valores de entrada al algoritmo de identificación $W1$ y $W2$.

Para propagar el valor de posición del actuador entre un tiempo de muestreo y el siguiente se emplea el método Runge-Kutta de cuatro pasos para integrar las ecuaciones diferenciales que describen a los actuadores.

Los coeficientes del método de Runge-Kutta de cuatro pasos se calculan de acuerdo con la regla descrita en la sección 1.2.3, en las siguientes ecuaciones se presenta el método utilizado.

$$\begin{aligned} z_1 &= x_n, \\ z_2 &= x_n + 0.5Tf(z_1, u), \\ z_3 &= x_n + 0.5Tf(z_2, u), \\ z_4 &= x_n + 0.5Tf(z_3, u), \end{aligned} \tag{2.6}$$

$$x_{n+1} = x_n + (1/6)T(f(z_1, u) + 2f(z_2, u) + 2f(z_3, u) + f(z_4, u)) \tag{2.7}$$

En las ecuaciones 2.6 y 2.7 la función f es la ecuación diferencial 2.3 que describe la posición de la válvula.

El siguiente diagrama en la Figura 2.7 describe la obtención de los valores $W1$ y $W2$ aplicando el método de Runge-Kutta.

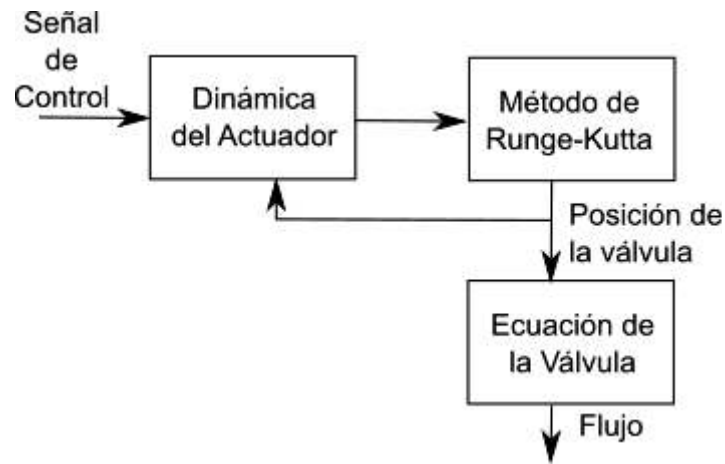


Figura 2.7 proyección de flujos a partir de señales de control.

2.2.2.1 Configuración con un Controlador

En esta configuración se extiende el vector de estado con los parámetros $T1$, $T2$ y p . Se obtiene el siguiente modelo en espacio de estados.

$$x1 = T1 \cdot Cp1, x2 = T2 \cdot Cp2, x3 = T3, x4 = h, x5=W1, x6 = \rho.$$

$$u_1 = W2.$$

$$dx3 = \left(\frac{x1 \cdot x5 + x2 \cdot u_1}{Cp3} - x3 \cdot (x5 + u_1) \right) / (x4 \cdot x6 \cdot Ac) \quad (2.8)$$

$$dx4 = (x5/x6 + u_1/x6 - K\sqrt{x4})/Ac \quad (2.9)$$

Se cuenta con una entrada controlada mientras que la otra entrada se mantiene fija, por lo que se la puede considerar como un parámetro de la planta y se la añade al vector de estados.

El parámetro K no se incluye en el vector de estado porque al no tener mediciones de nivel del líquido se vuelve más crítica la necesidad de estimar de manera correcta la variable de nivel en el tanque $x4$.

En esta configuración se tiene un sensor para medir la temperatura de la mezcla por lo tanto la planta tiene una sola salida y su matriz H es la siguiente:

$$H = [0 \ 0 \ 1 \ 0 \ 0 \ 0] \quad (2.10)$$

La matriz Jacobiana F descrita en la sección 1.2.2 está compuesta por la siguiente ecuación:

$$f_{ij} = \left[\frac{df_i(t, \hat{x}(t), u(t))}{d\hat{x}_j} \right]_{t=tk-1} \quad (2.11)$$

Lo que da como resultado la siguiente matriz:

$$F = \begin{bmatrix} 0,0,0,0,0,0; \\ 0,0,0,0,0,0; \\ \frac{x^5}{x^6 C_p^3 x^4 A_c}, \frac{u}{x^6 C_p^3 x^4 A_c}, \frac{x^5+u}{x^6 x^4 A_c} \dots \\ - \frac{x^5 \cdot x^1 + u x^2}{x^6 C_p^3 A_c x^4} + \frac{x^3(x^5+u)}{x^6 A_c x^4}, \frac{x^1/C_p^3 - x^3}{x^4 x^6 A_c} \dots \\ - \frac{x^5 x^1 + u x^2}{x^6 C_p^3 A_c x^4} + \frac{x^3(x^5+u)}{x^6 A_c x^4}, \\ 0,0,0, \frac{K x^4}{2A_c} \frac{1+u}{x^6 A_c}, \frac{x^5+u}{A_c x^6}; \\ 0,0,0,0,0,0; \\ 0,0,0,0,0,0; \end{bmatrix} \quad (2.12)$$

Para obtener el valor estimado de x a partir del valor anterior y el modelo de la planta, ec 1.24 en la sección 1.2.2 es necesario resolver una integral por lo que se utiliza nuevamente el método de Runge-Kutta de cuatro pasos. La función por resolver en este caso corresponde al sistema de ecuaciones de la planta extendida con los parámetros.

Para obtener la matriz A a partir de la matriz F se emplea la función de matlab `expm`, que computa el exponencial utilizando la aproximación de Pade.

Los valores de la matriz Q y de la matriz R que corresponden a la matriz de covarianza asociada al ruido en el sistema y la matriz asociada al ruido de medición serán ajustados en la etapa de obtención de resultados para obtener mejores respuestas con diferentes niveles de ruido o mejor sensibilidad a errores del algoritmo.

En la Figura 2.8 se presenta un diagrama del algoritmo de estimación de parámetros.

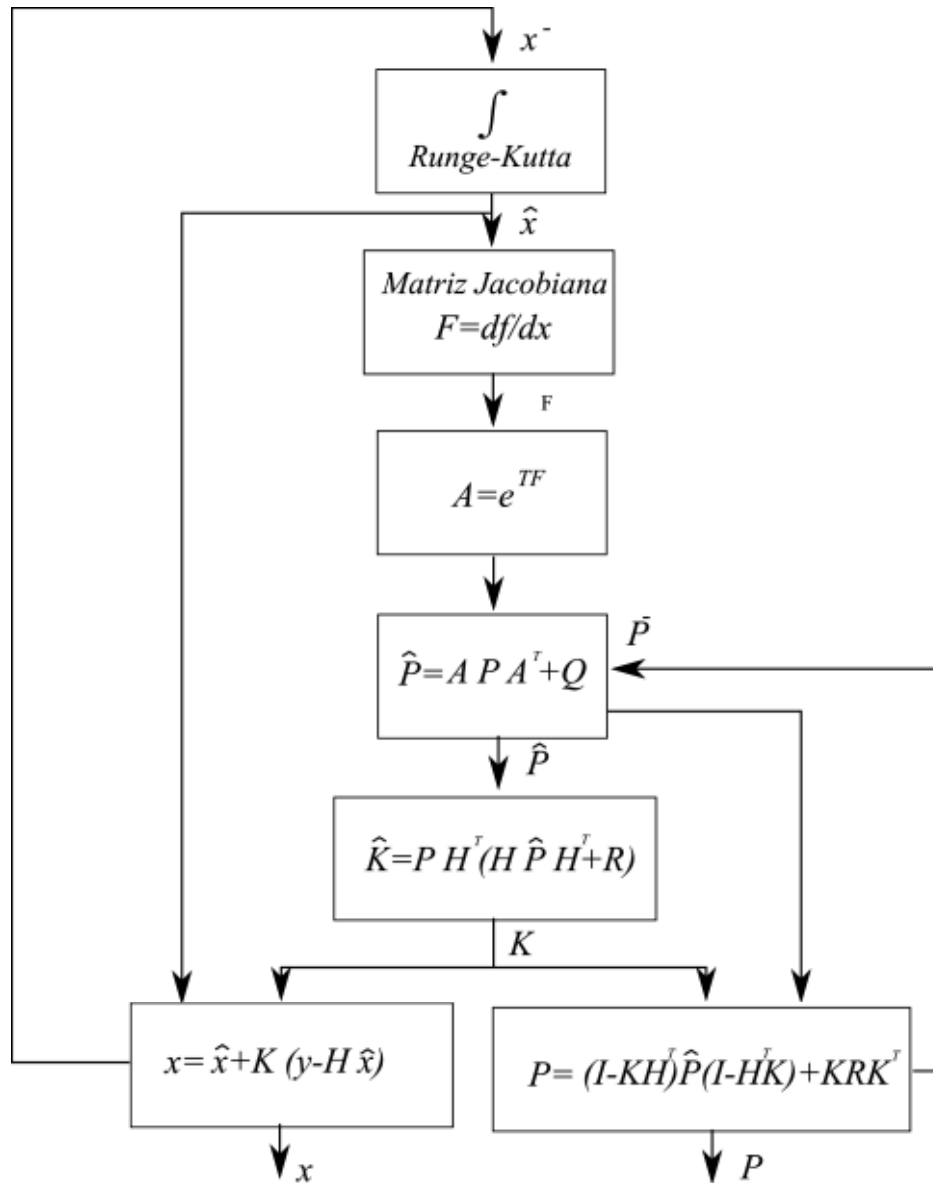


Figura 2.8 Diagrama algoritmo de identificación de parámetros

2.2.2.2 Configuración con dos controladores.

Se extiende el vector de estado con los parámetros estáticos $T1$, $T2$, p y K . Obteniéndose el siguiente modelo en espacio de estado de la planta.

$$x_1 = T1 \cdot Cp1, x_2 = T2 \cdot Cp2, x_3 = T3, x_4 = h, x_5 = K, x_6 = \rho.$$

$$u_1 = W1, u_2 = W2.$$

$$dx3 = \left(\frac{x1 \cdot u_1 + x2 \cdot u_2}{C_{p3}} - x3 \cdot (u_1 + u_2) \right) / (x4 \cdot x6 \cdot Ac) \quad (2.13)$$

$$dx4 = (u_1/x6 + u_2/x6 - x5\sqrt{x4})/Ac \quad (2.14)$$

En esta configuración se tiene dos sensores para medir la temperatura de la mezcla $T3$, que es $x3$ en el modelo de estado y la altura del fluido en el tanque, $x4$ en el modelo de estado, por lo que la matriz H es la siguiente:

$$H \begin{bmatrix} 0,0,1,0, \end{bmatrix} \quad (2.15)$$

La matriz jacobiana se calcula de la misma manera que en el literal anterior, para esta configuración es la siguiente.

$$F = \begin{bmatrix} 0,0,0,0,0,0; \\ 0,0,0,0,0,0; \\ \frac{u1}{x6 \cdot Cp3 \cdot x4 \cdot Ac}, \frac{u2}{x6 \cdot Cp3 \cdot x4 \cdot Ac}, \frac{u1+u2}{x6 \cdot x4 \cdot Ac} \dots \\ - \frac{u1 \cdot x1 + u2 \cdot x2}{x6 \cdot Cp3 \cdot Ac \cdot x4} - \frac{x3(u1+u2)}{x6 \cdot Ac \cdot x4^2}, 0 \dots \\ - \frac{u1 \cdot x1 + u2 \cdot x2}{x6 \cdot Cp3 \cdot Ac \cdot x4} - \frac{x3(u1+u2)}{x2 \cdot Ac \cdot x6^2}, \\ 0,0,0, \frac{x5 \cdot x4^{1/2}}{2Ac}, - \frac{\sqrt{x4}}{Ac}, - \frac{u1+u2}{Ac \cdot x6^2}; \\ 0,0,0,0,0,0; \\ 0,0,0,0,0,0; \end{bmatrix} \quad (2.16)$$

El vector x que se obtiene con el algoritmo de filtro de Kalman extendido contiene los parámetros estimados, para obtener residuos simplemente se los compara con los obtenidos del modelo.

$$Residuos = Parámetros_{Modelo} - Parámetros_{Filtro} \quad (2.17)$$

2.2.3 AISLAMIENTO DE FALLAS

El aislamiento de fallas se lleva a cabo utilizando redes neuronales para clasificar a los residuos generados con el Filtro de Kalman Extendido.

La red neuronal es creada con la función `patternnet()` de Matlab, el argumento de entrada de esta función es el número de capas escondidas y el algoritmo de ajuste de error. La Figura 2.10 representa la red neuronal empleada para la configuración de dos controladores.

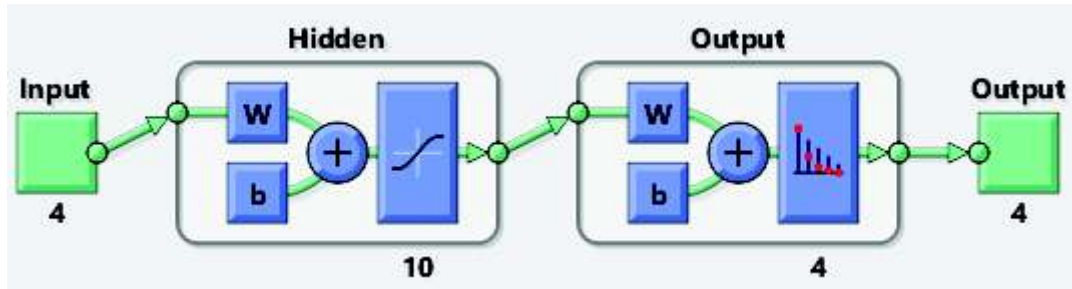


Figura 2.10 Red Neuronal Patternnet.

El número de capas escondidas para la red `patternnet` se puede cambiar de acuerdo con el desempeño del entrenamiento, su valor por defecto es diez capas escondidas. El algoritmo utilizado para entrenar a la red neuronal es “`trainrp Resilient Backpropagation`”.

En la Figura 2.9 se muestran las entradas a la red neuronal para la configuración con un controlador y con dos controladores.

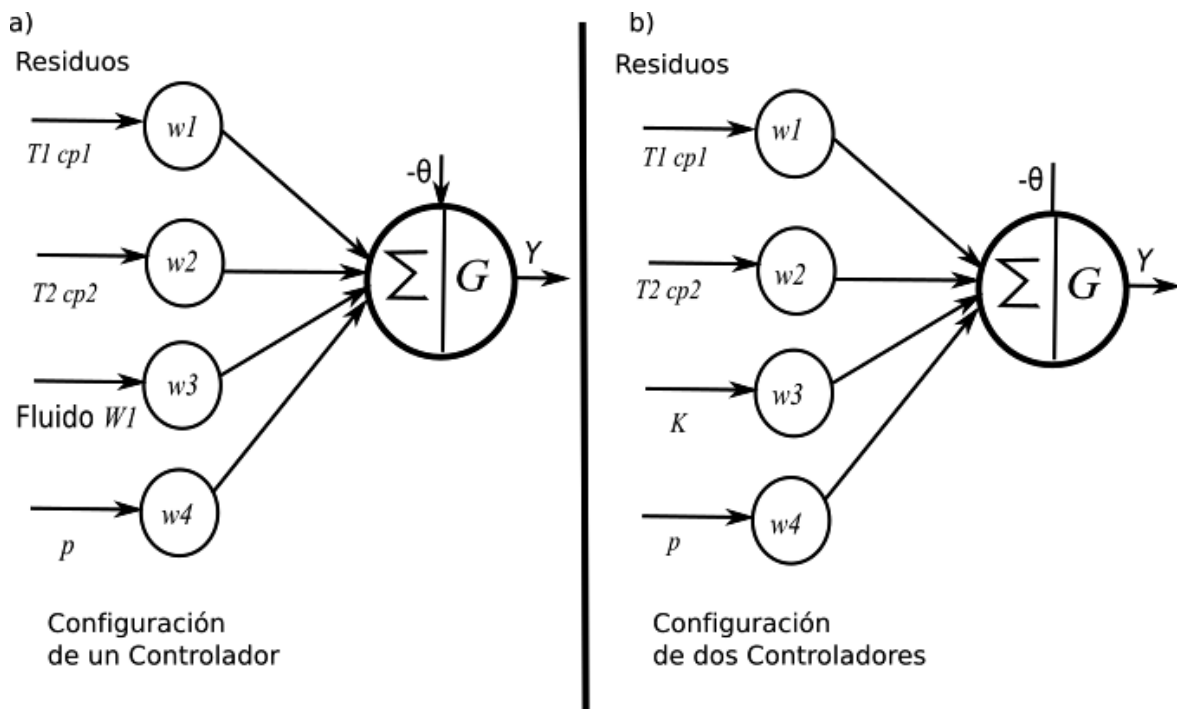


Figura 2.9 Entradas a la Red Neuronal.

La función `train()` de matlab es la utilizada para realizar el entrenamiento de la red neuronal. Para entrenar una red de reconocimiento de patrones se debe tener un vector de entradas y un vector de objetivo, el vector de objetivo contiene todas las clases a las que pertenece la entrada y su valor debe ser de cero en todos menos en la clase objetivo para la entrada actual. En el presente trabajo el vector objetivo contiene las clases de falla en las que se puede aislar los patrones encontrados con el algoritmo de identificación.

En la Figura 2.11 se presenta el desempeño del algoritmo de entrenamiento como su valor de entropía cruzada en función de las épocas de ejecución. La entropía cruzada calcula el desempeño de la red dado las salidas y el objetivo, al minimizar la entropía cruzada se obtiene mejores clasificadores. El número de épocas es determinado automáticamente por el algoritmo.

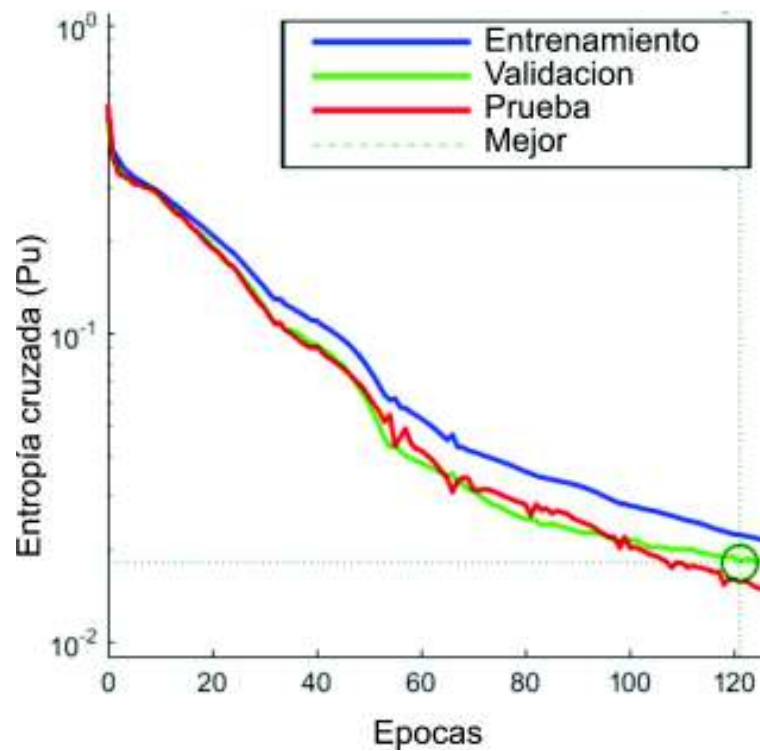


Figura 2.11 Desempeño de la red.

CAPÍTULO 3

PRUEBAS Y RESULTADOS

Este capítulo expone los resultados obtenidos en las simulaciones realizadas, se divide en dos secciones, la primera dedicada a estimación de parámetros y la segunda al aislamiento de fallas.

3.1 IDENTIFICACIÓN DE PARÁMETROS.

En esta sección se presentan las respuestas del algoritmo de identificación de parámetros para las dos configuraciones de la planta. Se consideran diversos escenarios de nivel de ruido en sensores, cambios en parámetros de la planta y fallas introducidas en los transmisores y actuadores.

3.1.1 CONFIGURACIÓN CON UN CONTROLADOR

En la Figura 3.1 se presenta la respuesta de la planta en esta configuración.

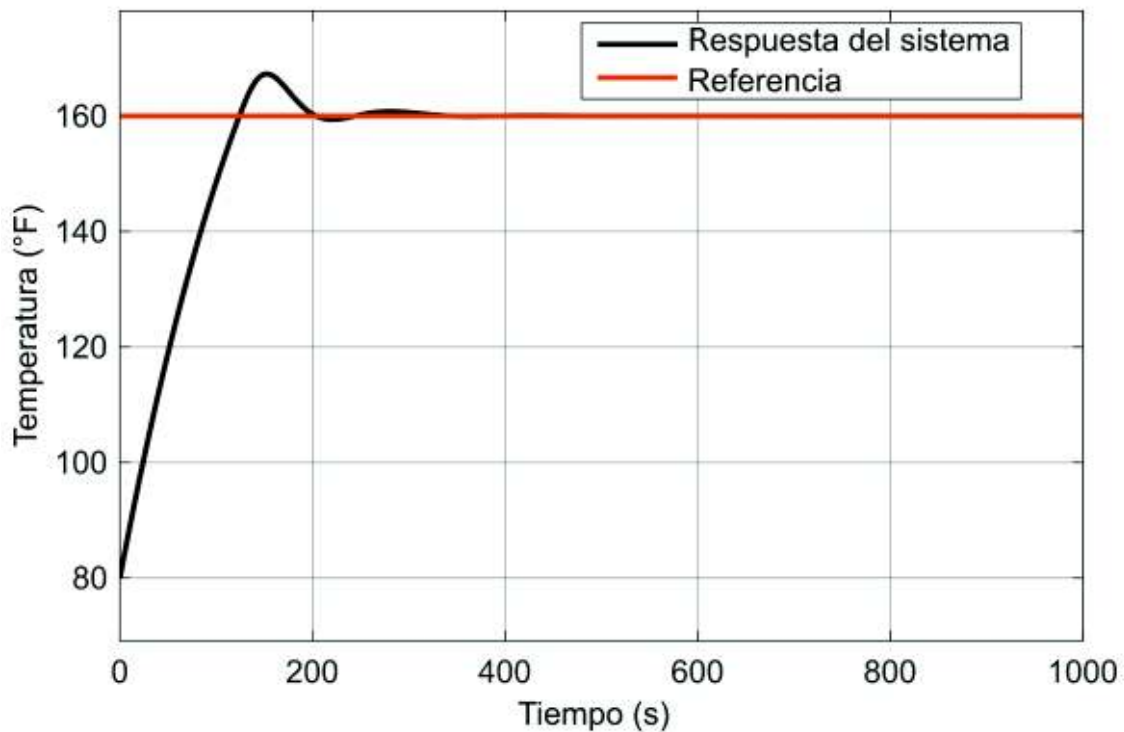


Figura 3.1 Respuesta de la planta en la configuración de un controlador.

Los valores de los parámetros del controlador PI para temperatura se ajustaron de forma manual a $K_p=0.01$ y $K_i=0.0015$ y su referencia es $160\text{ }^\circ\text{F}$.

El nivel del tanque se inicializó con un valor de 5 ft, se debe tener en cuenta que con un valor de cero el modelo de la planta no se mantiene, se produce una división para cero.

En aplicaciones prácticas no existen sensores perfectos. Para efectuar las simulaciones se considera un escenario con tiempo de respuesta de los transmisores diferente cero y afectados por ruido gaussiano de media cero. La varianza del ruido en el sensor de temperatura se configura en 1.6 con una constante de tiempo de 0.1 segundos. En la Figura 3.2 se presenta una comparación entre la respuesta del transmisor y la planta.

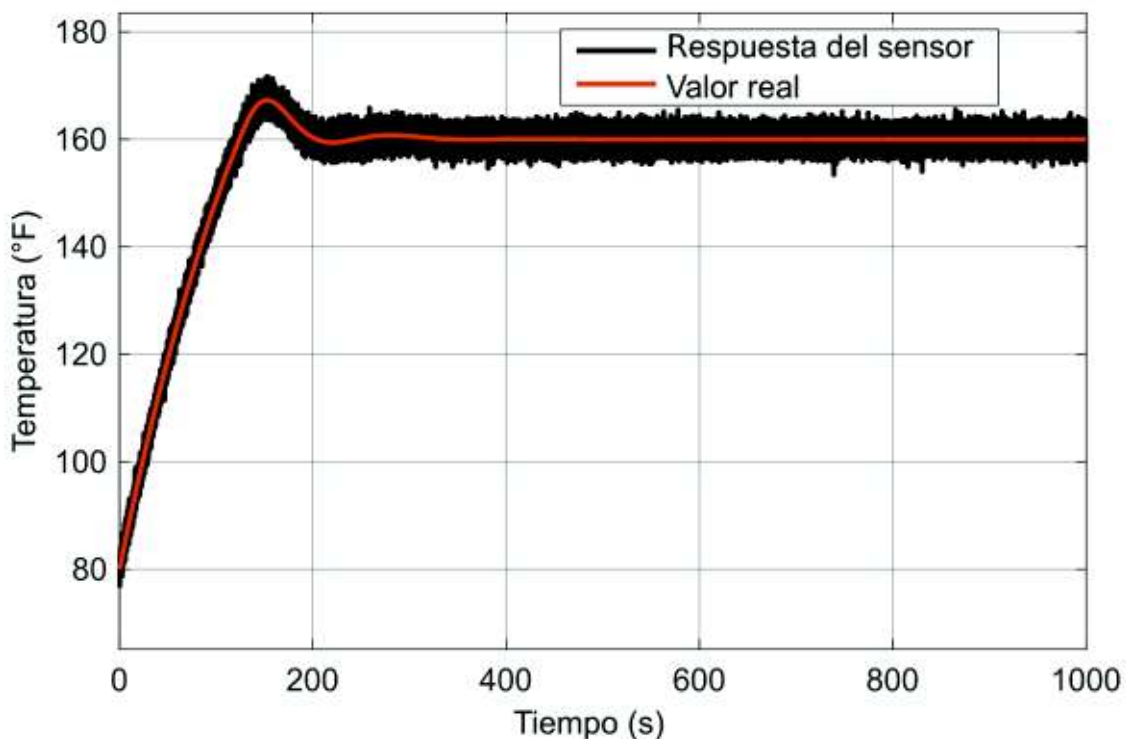


Figura 3.2 Respuesta del transmisor de temperatura en configuración de un controlador.

El algoritmo de detección de fallas debe ser capaz de ignorar variaciones en parámetros debidas al funcionamiento normal de los transmisores.

El valor de posición del actuador se inicializa en cero.

En la Figura 3.3 se presentan las variaciones en los parámetros cuando no se introducen fallas en la planta. Las variaciones que se observan no son de un valor

considerable sin embargo son ocasionadas por el ruido que se añade al sensor y por la constante de tiempo del sensor.

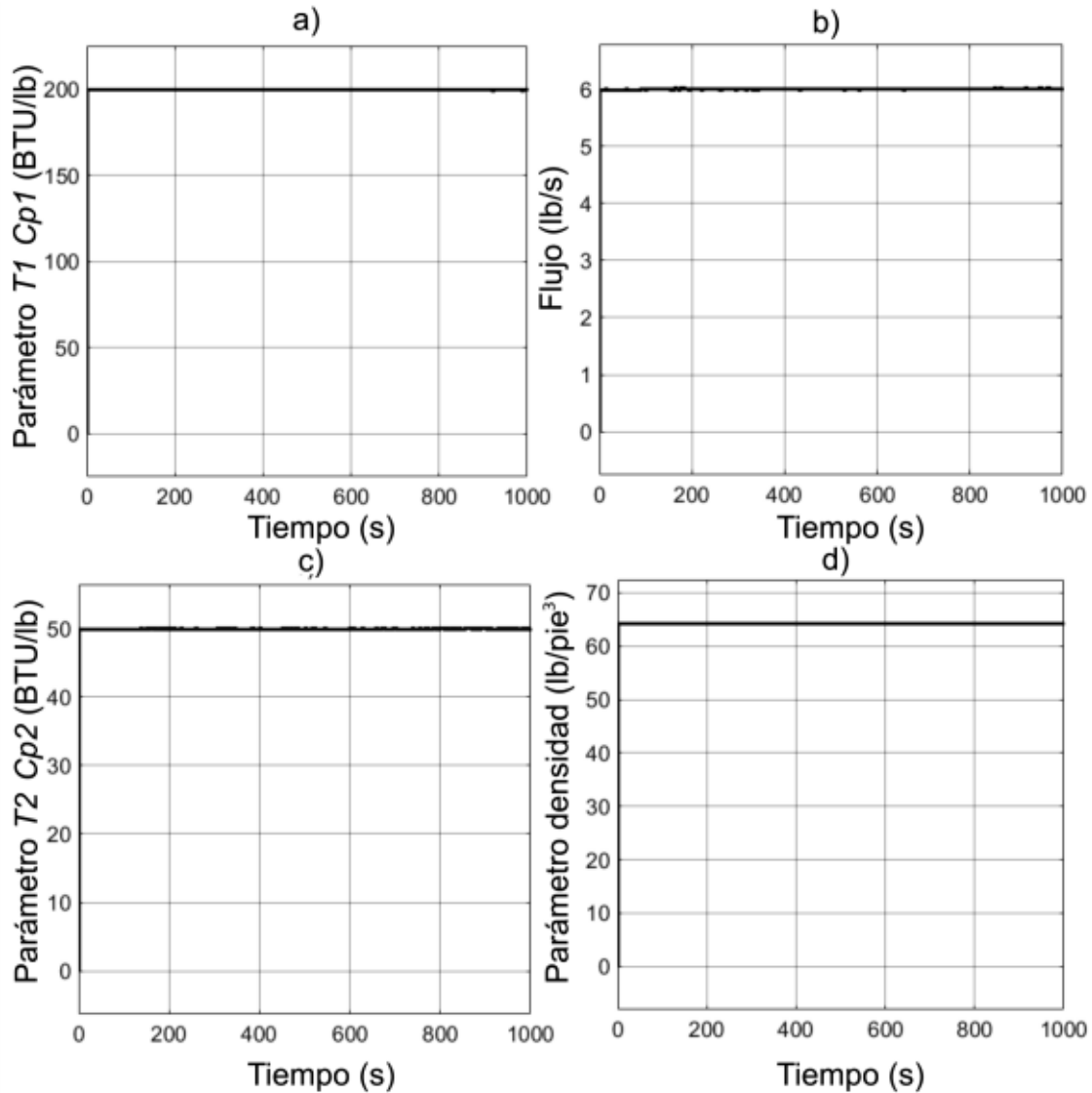


Figura 3.3 Estimación de parámetros libre de fallas, configuración de un controlador

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro $w1$, c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.1.1 Fallas en la Instrumentación

En esta sección se introduce fallas en el actuador y en el transmisor para presentar las variaciones de parámetros que producen.

La convergencia de parámetros depende del nivel de excitación de las ecuaciones dinámicas de la planta y esta depende de las entradas al algoritmo, por lo que se utiliza el mismo intervalo de tiempo entre 200 (s) y 500 (s) para introducir fallas en

todas las simulaciones presentadas en la sección de estimación de parámetros. De esta forma el valor en la entrada va a ser el mismo mientras se añade la señal de falla.

3.1.1.1.1 Falla en el Actuador

Se introduce una falla aditiva en el flujo del actuador con una pendiente de 0.003 que comienza a los 200 (s) y crece hasta los 500 (s). En la Figura 3.4 se presenta el valor de flujo que se calcula utilizando la señal de control comparada con el valor obtenido de la simulación de la planta bajo la presencia de esta falla.

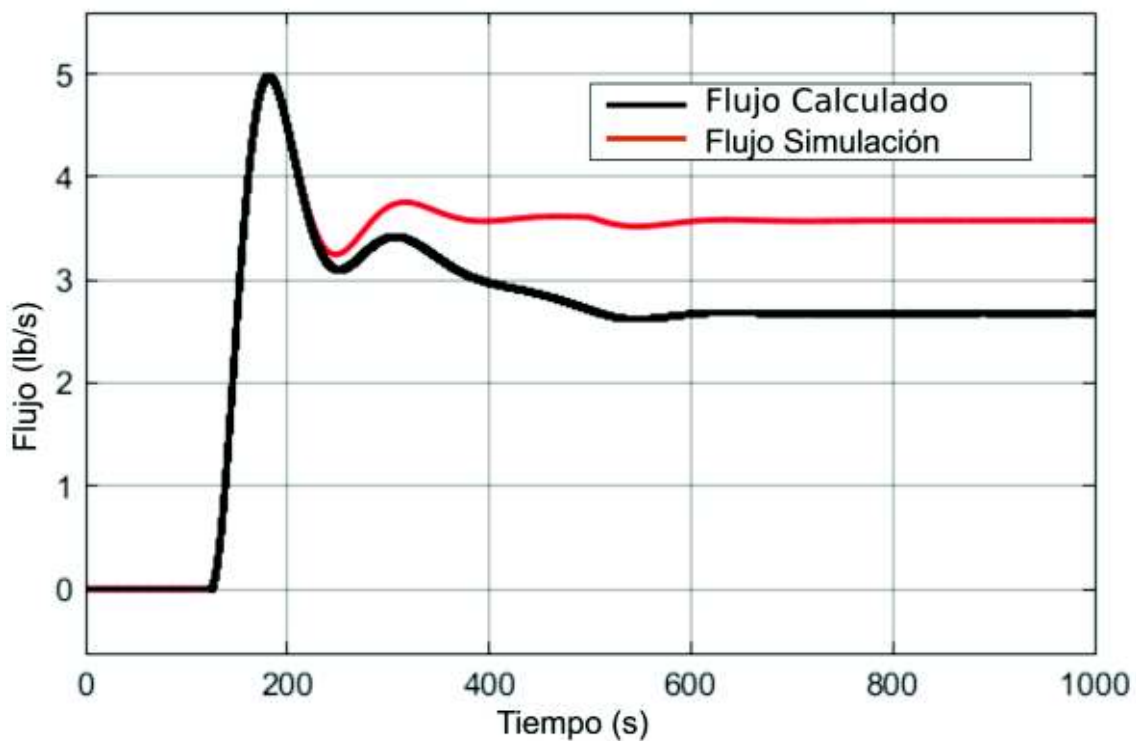


Figura 3.4 Falla en el actuador, configuración con un controlador.

El flujo en el actuador a pesar de verse afectado por la falla aditiva se mantiene constante debido a la acción del controlador, sin embargo, la señal de control si cambia su valor por lo tanto se da una discrepancia entre ambos valores. Como consecuencia de esto se producen variaciones en los parámetros.

En la Figura 3.5 se muestra la respuesta del algoritmo de estimación de parámetros en este escenario y en la Figura 3.6 se presentan los parámetros que se obtienen cuando se introduce una falla similar pero con una pendiente negativa.

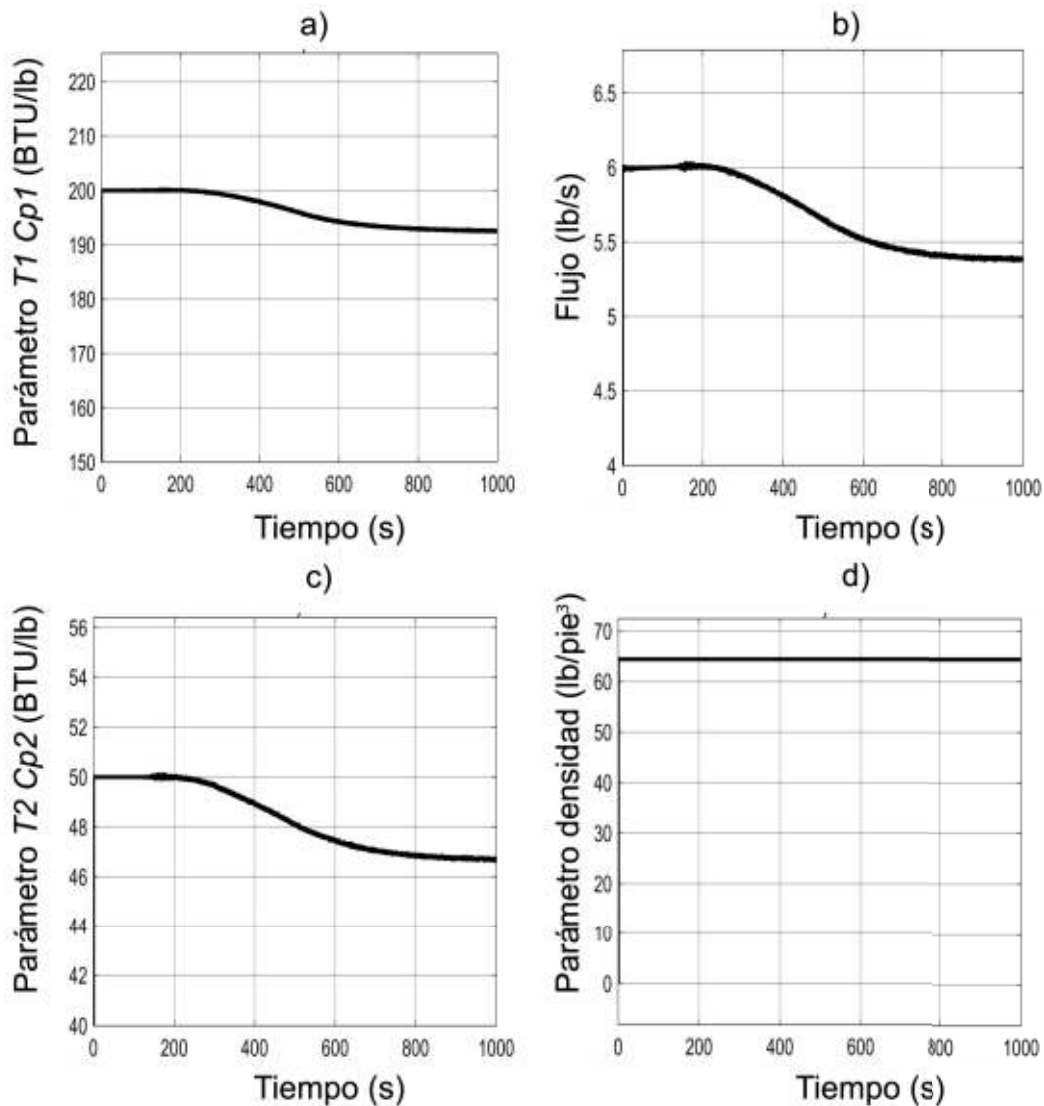


Figura 3.5 Estimación de parámetros con falla positiva en el actuador a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro w_1 , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

La diferencia entre las variaciones de los parámetros obtenidas al cambiar el signo a la pendiente de la falla en el actuador es simplemente un cambio en la dirección en que varían los parámetros.

Se observa en las figuras que la dirección en que cambian los parámetros es opuesta al signo de la pendiente de falla.

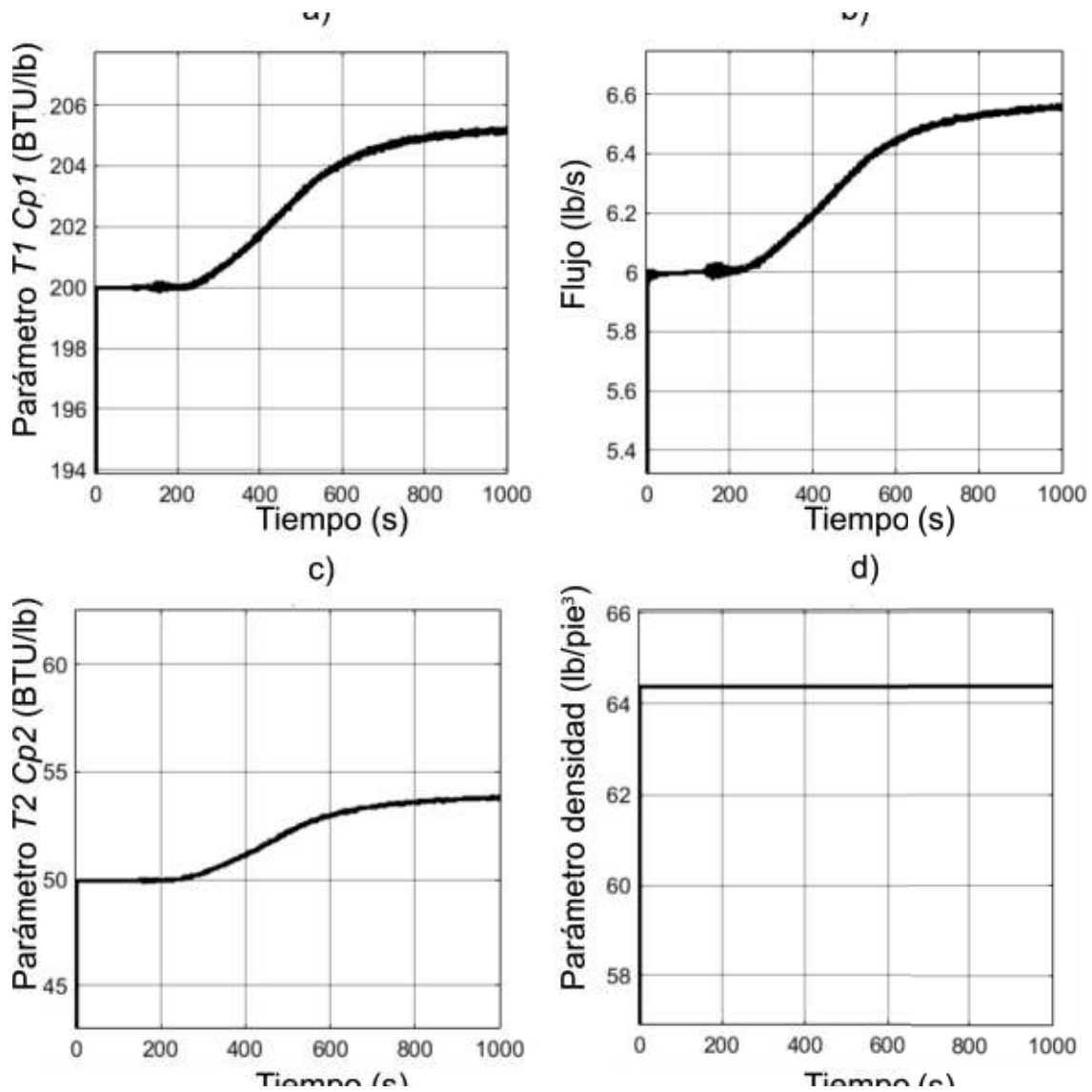


Figura 3.6 Estimación de parámetros con falla negativa en el actuador a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro w_1 , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.1.1.2 Falla en el Sensor

Se añade una señal de falla que crece desde 200 (s) a 500 (s) con una pendiente de 0.05 unidades.

En esta falla se emplea un valor de pendiente mayor porque el valor estable del sensor de temperatura es de 160 mientras que el valor estable para el actuador es mucho menor. En la Figura 3.6 se presenta la discrepancia entre el valor de

temperatura en la planta y el valor que el algoritmo de estimación recibe del sensor de temperatura con el cual va a trabajar.

En este caso la falla que se introduce afecta directamente al valor del sensor por lo que la discrepancia que se observa en la figura tiene el valor de la pendiente introducida.

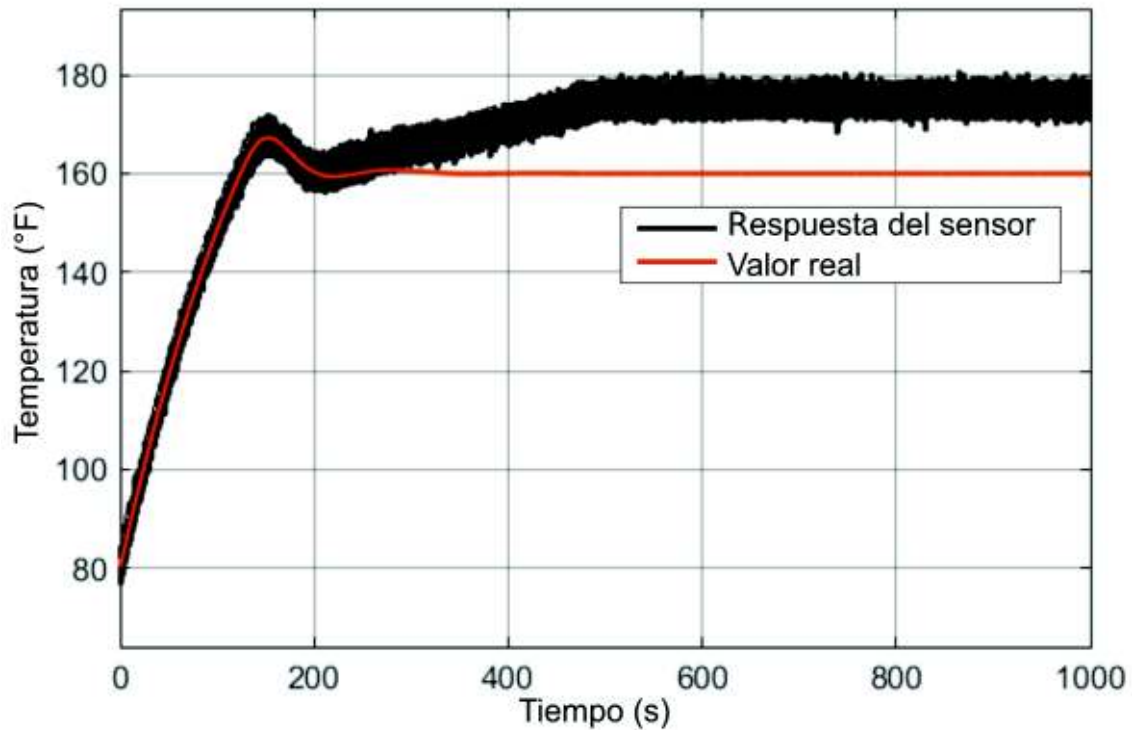


Figura 3.7 Falla en el sensor, configuración con un controlador.

La variación de los parámetros que se obtiene al introducir esta señal de falla se presenta en las figuras 3.8 y 3.9 para pendiente positiva y negativa respectivamente.

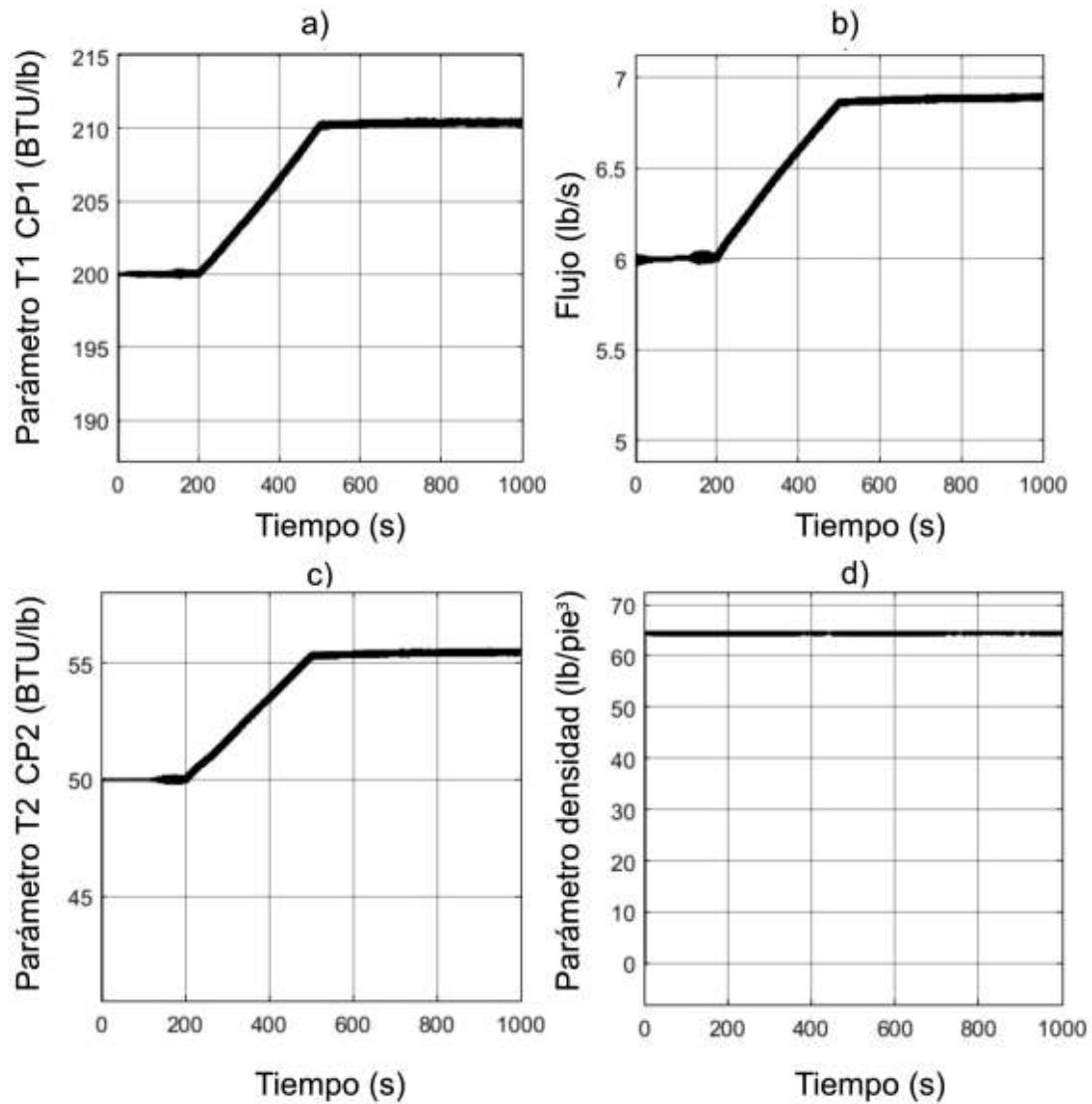


Figura 3.8 Estimación de parámetros con falla negativa en el actuador a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro $w1$, c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

Con respecto a la variación de parámetros este tipo de falla se diferencia de la falla en el actuador, presentada en la sección anterior, en que los parámetros varían en la misma dirección que la pendiente. Otra diferencia es que el parámetro TIC_{p1} varía en una cantidad mayor en comparación con los demás parámetros.

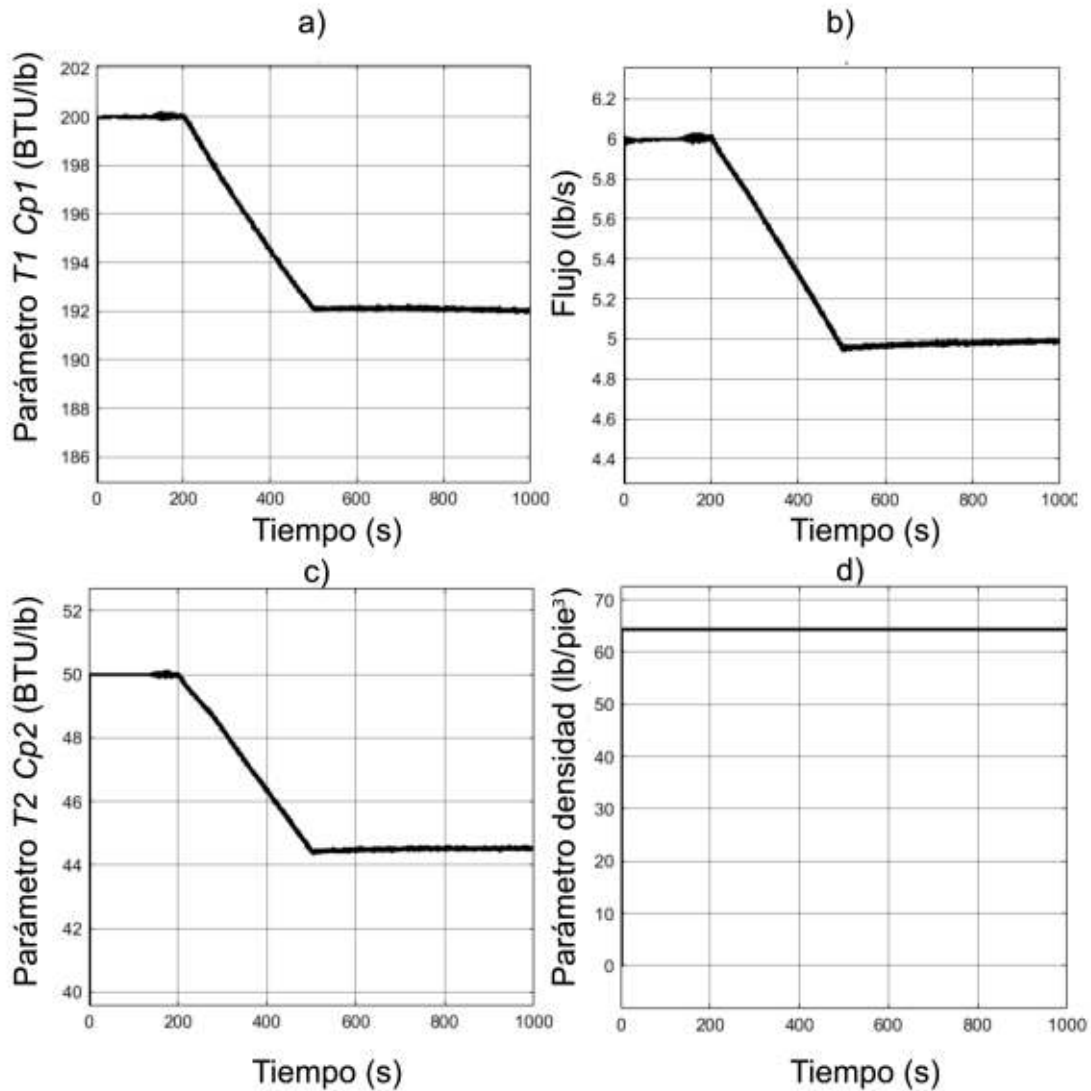


Figura 3.9 Estimación de parámetros con falla negativa en el actuador a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro w_1 , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.1.2 Fallas en la Planta

En esta subsección se presentan variaciones en los parámetros producidas por defectos en el proceso como fugas de material o de energía en el tanque.

También se considera un escenario en el que existe un cambio en la densidad del líquido.

3.1.1.2.1 Fuga Térmica

En este caso se resta una falla en la ecuación de balance térmico, esta falla al igual que las anteriores comienza en 200 segundos con una pendiente de -0.001 hasta los 500 segundos. En la siguiente figura se presentan las variaciones en los parámetros que se obtiene del algoritmo para esta falla.

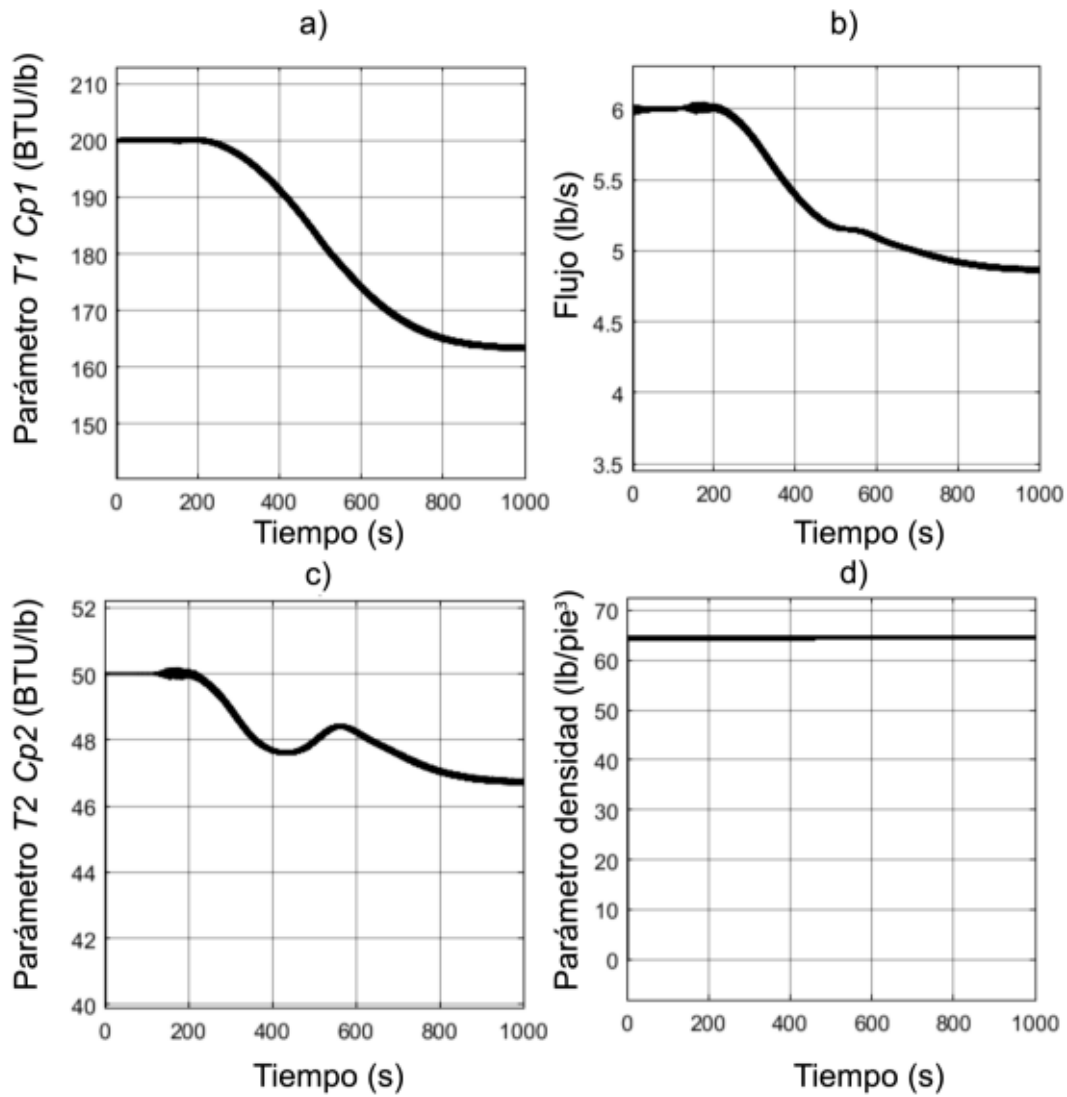


Figura 3.10 Estimación de parámetros con fuga térmica, configuración con un controlador
 a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro w_1 , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

Las respuestas que se obtienen con esta falla son similares a las que se obtiene al añadir una falla en el sensor de temperatura.

3.1.1.2.1 Fuga de Material

Se resta una señal de falla con pendiente $-1e-4$ que comienza a los 200 (s) y termina en 500 (s). En la siguiente figura se presentan las variaciones en los parámetros obtenidas.

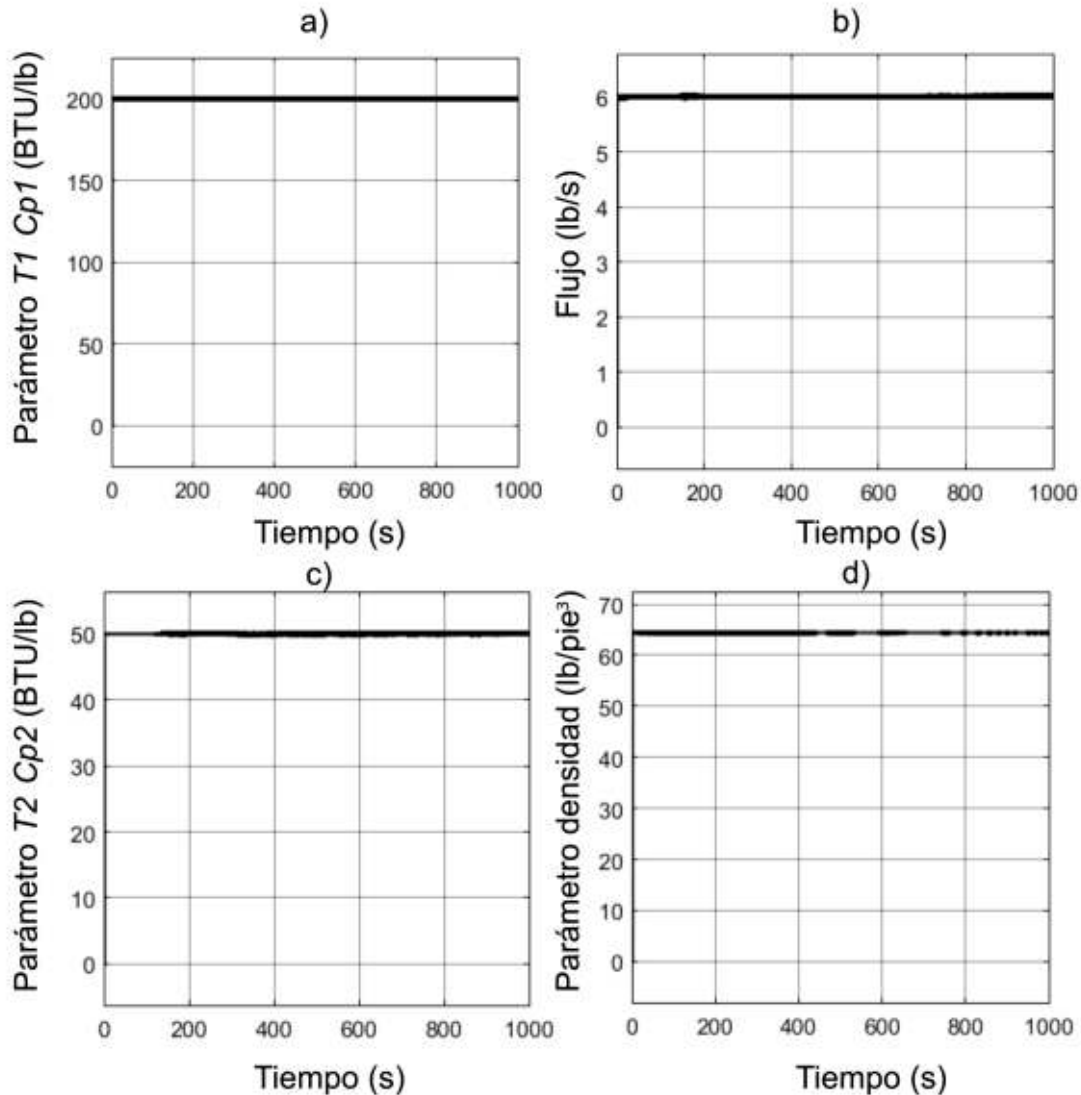


Figura 3.11 Estimación de parámetros con fuga de material, configuración un controlador
 a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro w_1 , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

En esta configuración, al no contar con un sensor de nivel no se cumple con la condición de persistencia de excitación lo que ocasiona que los parámetros no convergen. Para poder detectar este tipo de falla se necesita que existan cambios en la entrada del proceso lo que produce excitación de las ecuaciones dinámicas.

La estimación de nivel por parte del filtro si cambia su valor por lo que otra alternativa puede ser un comparador con límite.

3.1.1.2.1 Cambio en la Densidad

Se añade una señal de falla con pendiente 0.1 entre los 200 (s) y 500 (s) al parámetro de densidad. En la Figura 3.12 se observa la variación que la falla produce en los parámetros.

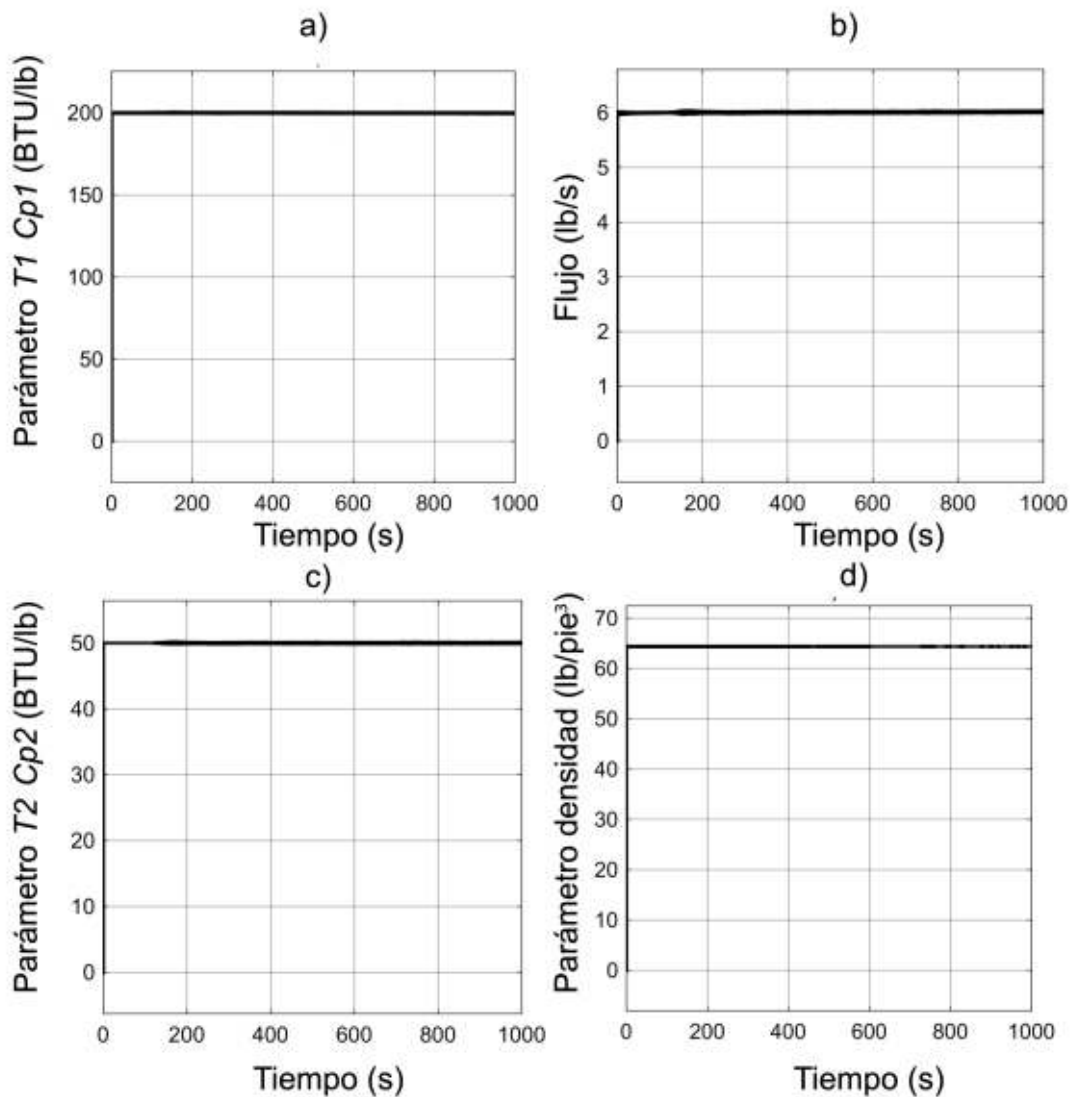


Figura 3.12 Estimación de parámetros con cambio de densidad, configuración de un controlador a) Parámetro $T_1 C_{p1}$, b) Parámetro w_1 , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro p

Al igual que en el caso anterior no existe variación en los parámetros debido a la falta de excitación del sistema.

3.1.2 CONFIGURACIÓN CON DOS CONTROLADORES

Las válvulas se inicializan en un valor de 0, pero el valor de nivel en el tanque se inicializa en dos porque el modelo de la planta es inconsistente para un nivel de mezcla de cero. Los valores K_p y K_i para los controladores se ajustaron manualmente a los valores que se indican en la Tabla 3.1.

Tabla 3.1 Parámetros de los Controladores.

	K_p	K_i
PI Nivel	2	0.0034
PI Temperatura	-1	-1e-5

En las figuras 3.13 y 3.14 se presenta la respuesta de la planta con parámetros de los controladores detallados en la Tabla 2.1 y sin introducir fallas en el sistema.

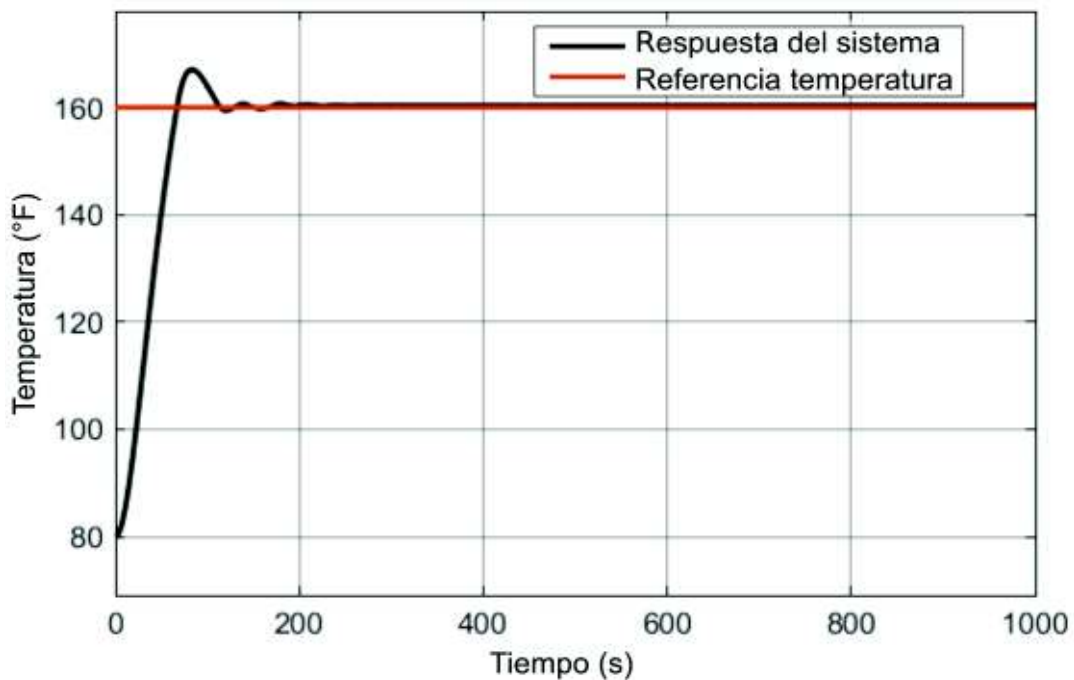


Figura 3.13 Respuesta de Temperatura de la mezcla, Configuración dos controladores.

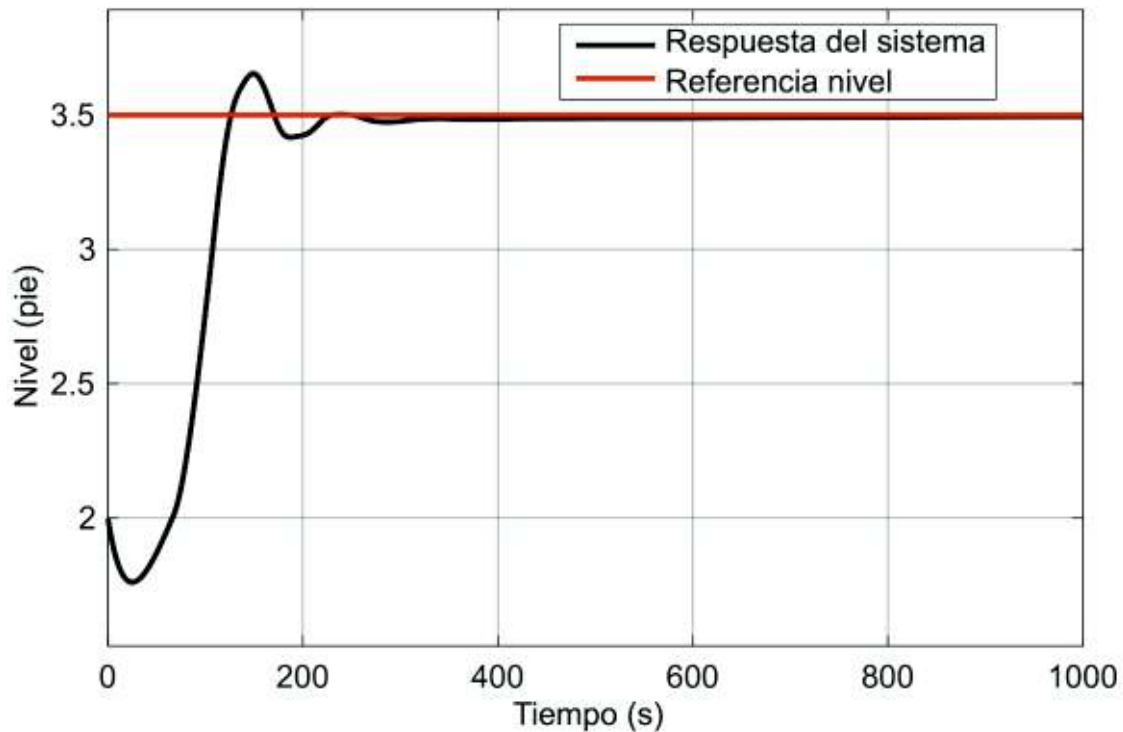


Figura 3.14 Respuesta de Nivel en el tanque, configuración de dos controladores.

Al igual que en la configuración con un controlador, en esta configuración también se realiza las simulaciones con ruido y tiempo de respuesta distinto de cero en los transmisores.

En esta configuración se realizan las simulaciones de fallas considerando una varianza del ruido en los transmisores alrededor al 1% del valor de referencia de las variables, 1.6 para el transmisor de temperatura y 0.001 para el transmisor de nivel en el tanque. Se asigna una constante de tiempo de 0.01 segundos para ambos.

En la Figura 3.15 se presenta la respuesta de los bloques de transmisor en comparación con los valores de simulación de la planta

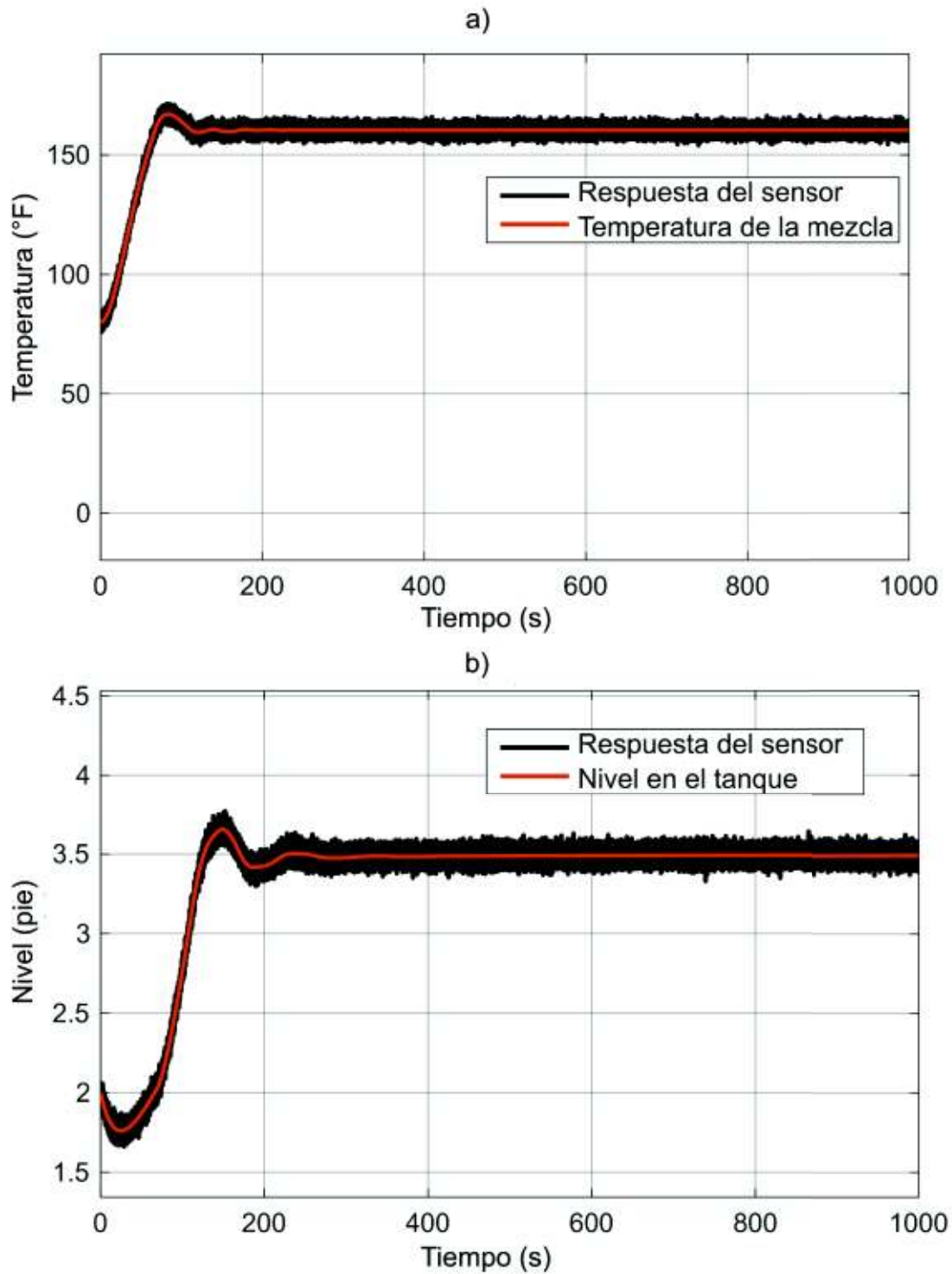


Figura 3.15 Respuesta de los transmisores, configuración de dos controladores
a) Transmisor de temperatura, b) Transmisor de nivel.

El valor de la matriz R se ajusta en el algoritmo para permitir estimar los parámetros de la planta bajo estas características,

En la Figura 3.16 se presenta la estimación de los parámetros cuando no se introduce ninguna falla en la simulación, los parámetros se mantienen en sus valores especificados en la Tabla 2.1. Las variaciones en los parámetros son de baja magnitud, estas se deben a la parte dinámica del error de los sensores causado por tiempo de respuesta diferente de cero y variaciones que corresponden al ruido.

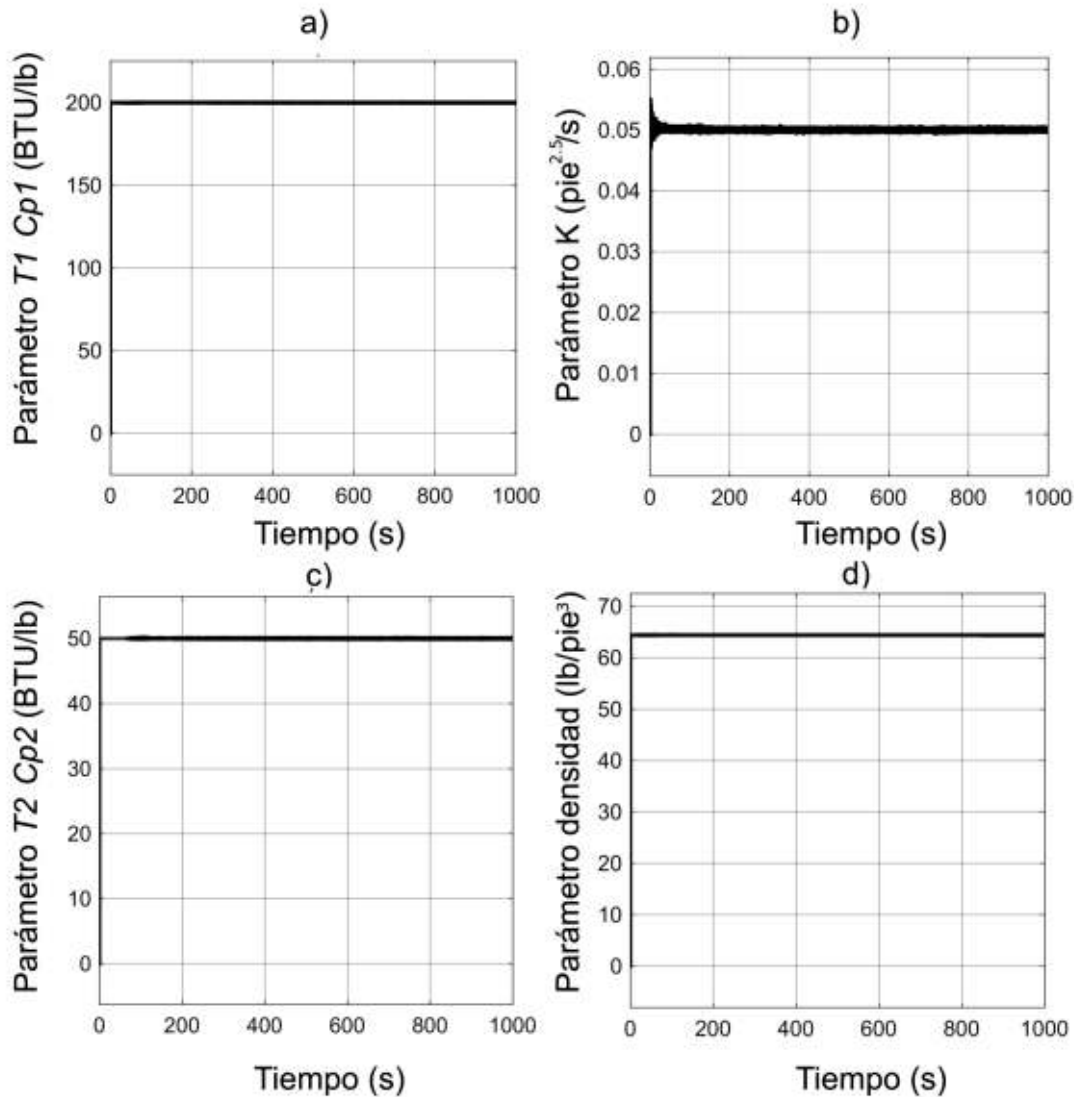


Figura 3.16 Estimación de parámetros sin fallas, configuración de dos controladores. a) Parámetro $T_1 C_{p1}$, b) Parámetro K , c) Parámetro $T_2 C_{p2}$, d) Parámetro ρ .

El algoritmo de detección de fallas debe ignorar las variaciones de parámetros dentro un rango que corresponde a lo presentado en la figura anterior.

En la Figura 3.17 se presenta la estimación del algoritmo para la temperatura de la mezcla y el nivel en el tanque (x_3 y x_4 en el modelo de estado del algoritmo).

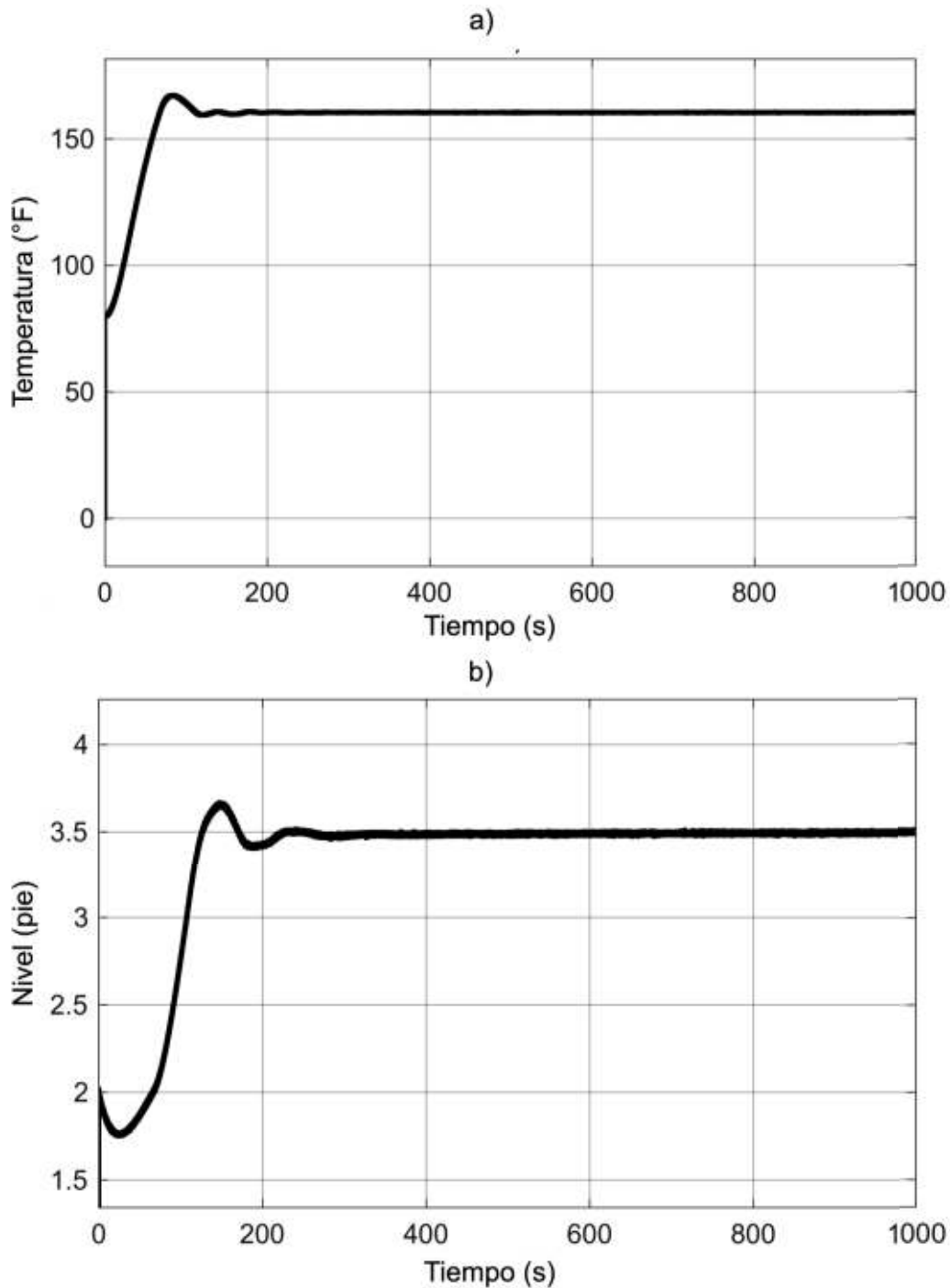


Figura 3.17 Estimación de estados con ruido en transmisores a) Temperatura de la mezcla, b) Nivel del Tanque.

Como se observa de la comparación entre las respuestas de los sensores (Figura 3.15) y del algoritmo (Figura 3.17), en las últimas el algoritmo logra filtrar un gran porcentaje del ruido.

3.1.2.1 Fallas en la Instrumentación

En esta sección se presentan las variaciones en los parámetros que presenta el algoritmo de estimación cuando se introduce fallas en las válvulas de fluido frío y caliente y en los sensores.

3.1.2.1.1 Falla en el Transmisor de Temperatura

Empezamos con una falla en el sensor de temperatura. Se añade una señal de falla que crece a partir de 200 (s) hasta 500 (s) con una pendiente de 0.05 unidades por segundo.

En la Figura 3.18 se muestra la respuesta del sensor de temperatura en conjunto con el valor de temperatura en el tanque.

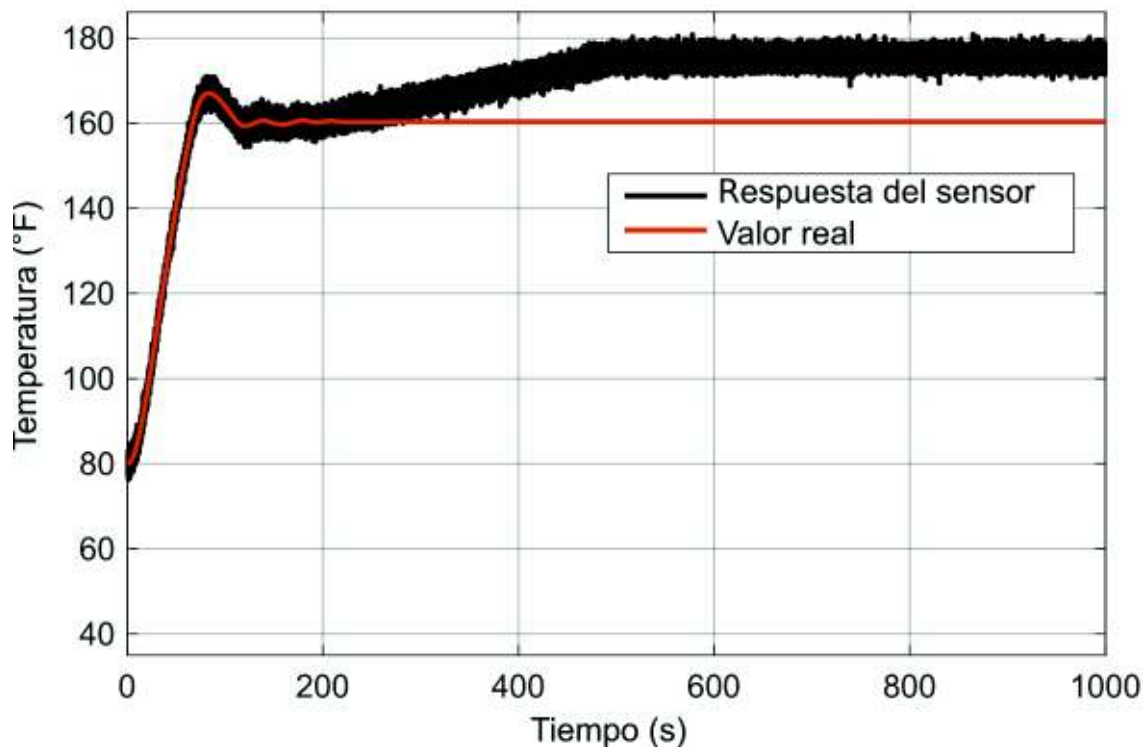


Figura 3.18 Falla positiva en el sensor de temperatura.

En la Figura 3.19 se presenta la variación en los parámetros que corresponde a la introducción de esta falla.

Se observa que la introducción de la falla provoca variaciones en los parámetros $T1Cp1$ y $T2Cp2$ mientras que los demás parámetros no presentan variaciones,

esto se debe a que estos parámetros están directamente relacionados con el valor de T_3 que es la temperatura de la mezcla.

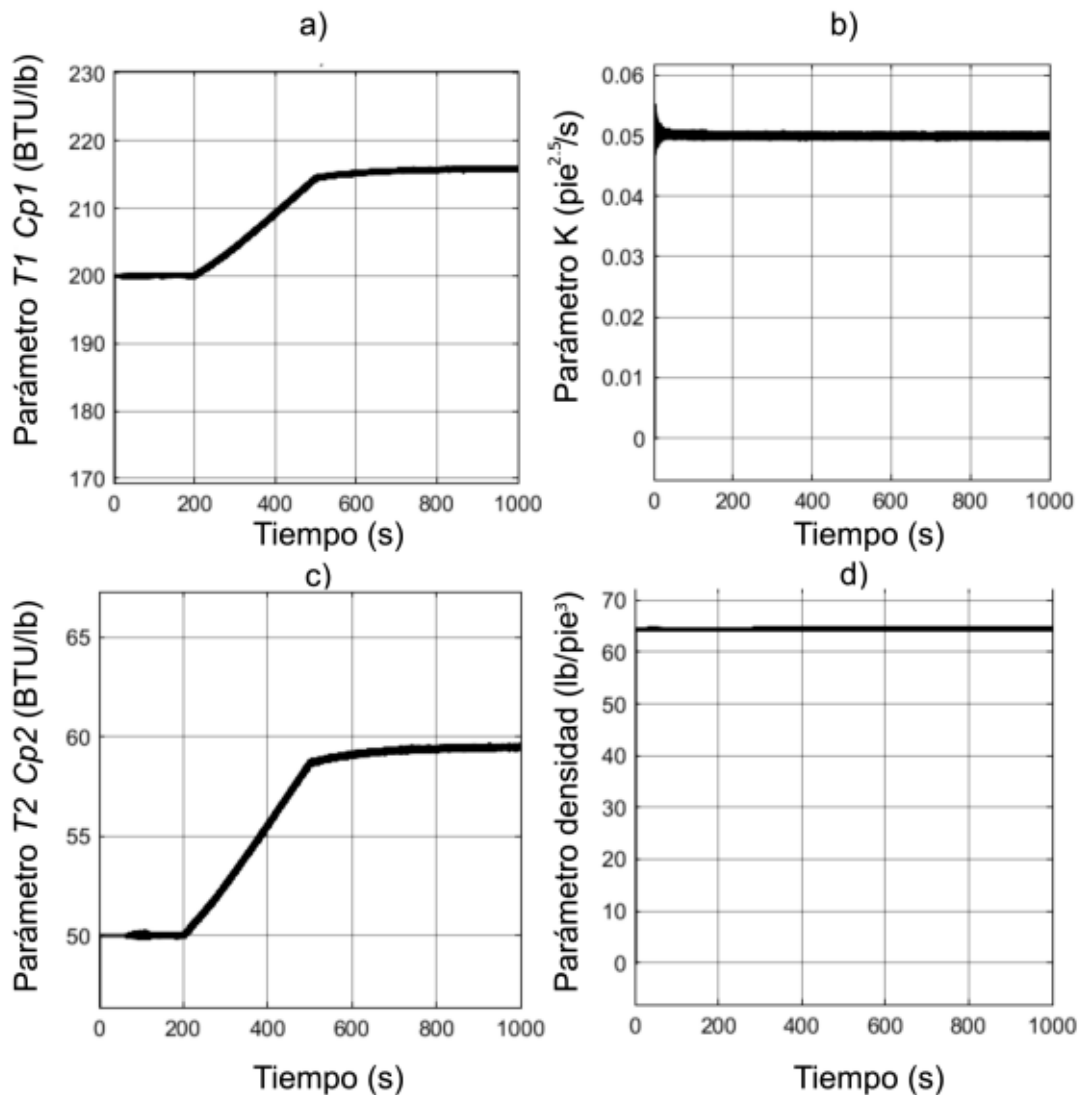


Figura 3.19 Estimación de parámetros con falla positiva en el sensor de temperatura,

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

En la Figura 3.20 se presenta la respuesta para una falla en el mismo elemento, el sensor de temperatura, pero en este caso la pendiente es negativa.

La diferencia entre las variaciones de parámetros presentados en la Figura 3.19 y la Figura 3.20 es que las gráficas de $T_1 C_{p1}$, y $T_2 C_{p2}$, se observa una variación positiva en lugar de negativa.

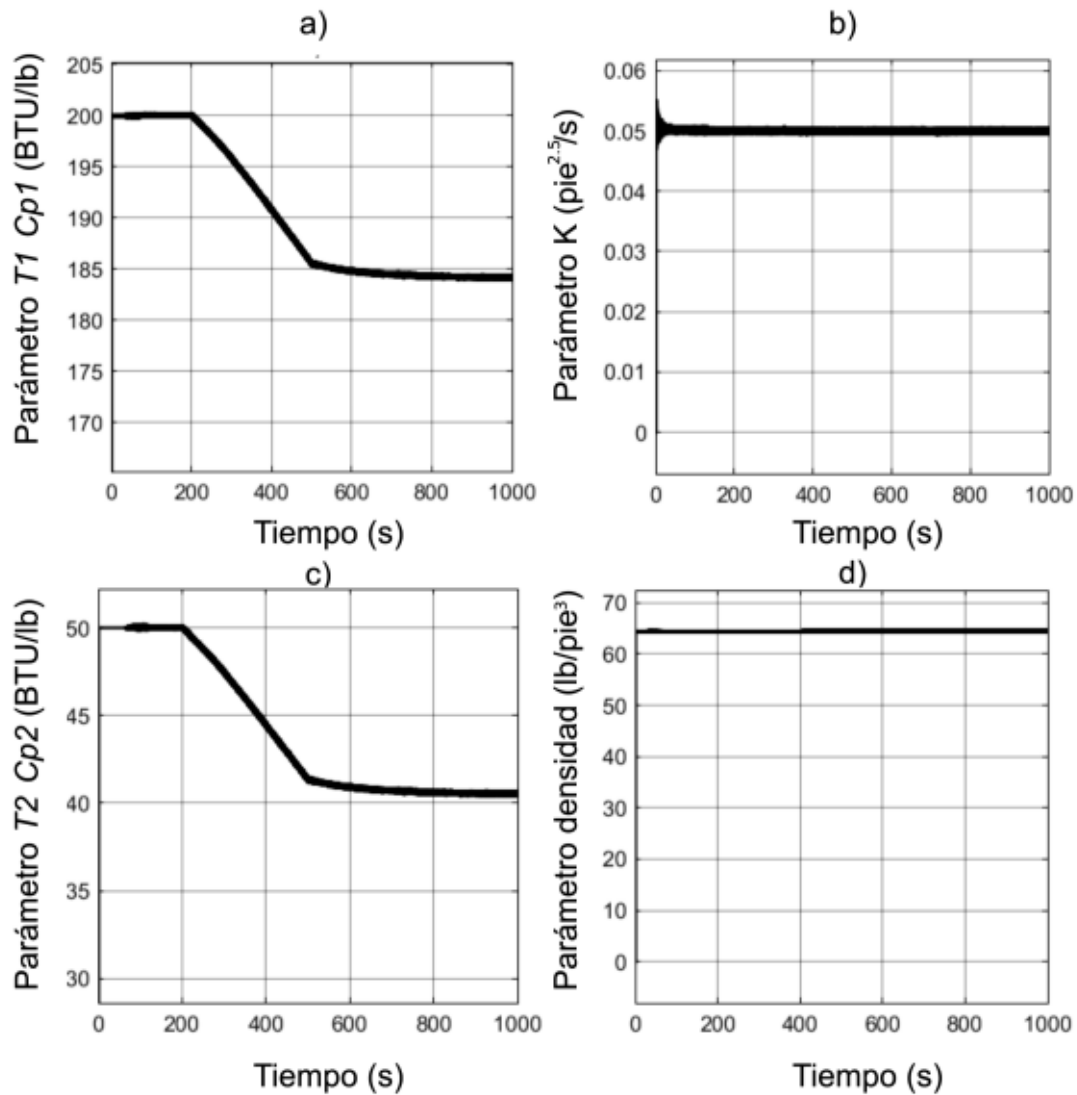


Figura 3.20 Estimación de parámetros con falla negativa en el sensor de temperatura,

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.2.1.2 Falla en el Transmisor de Nivel

En este escenario se considera una falla en el sensor de nivel, la falla comienza en 200 (s) con una pendiente de 0.002 y dura los 500 (s).

En la Figura 3.21 a continuación se presenta la respuesta del sensor en conjunto con el valor de la planta.

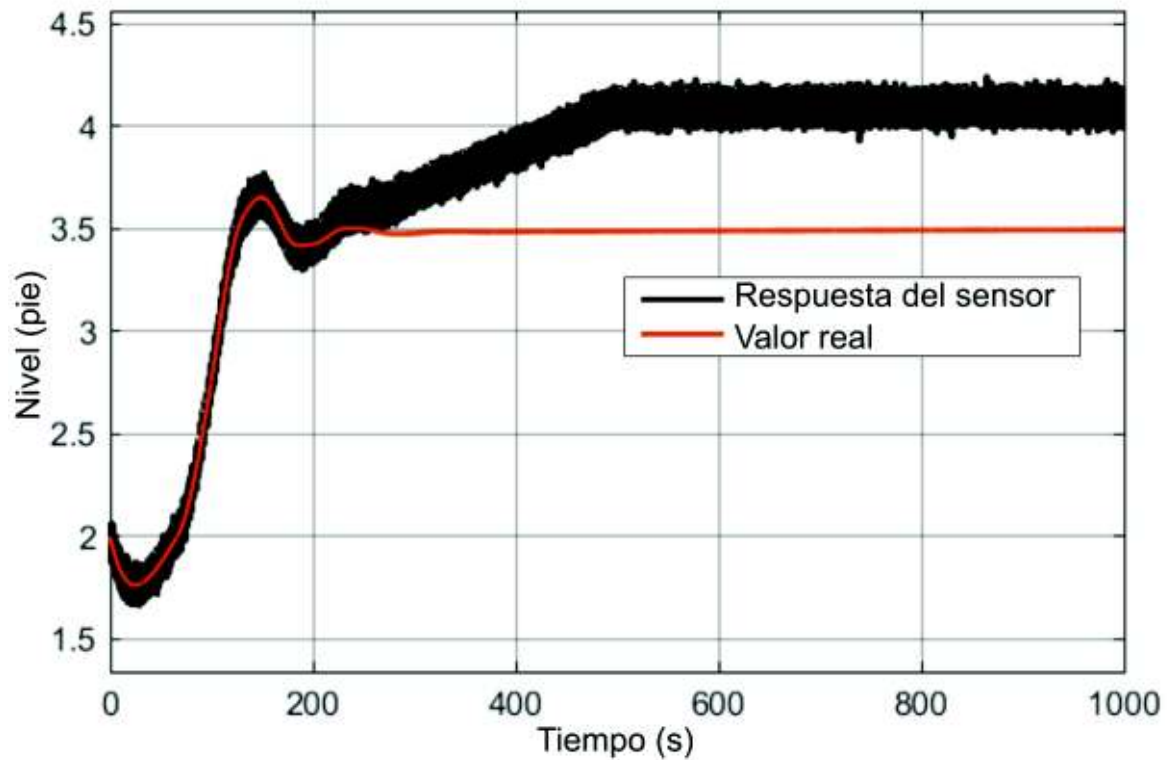


Figura 3.21 Falla en el sensor de nivel.

En la Figura 3.22 se presentan las variaciones en los parámetros para este escenario. y en la Figura 3.23 para la misma falla, pero con pendiente negativa.

La diferencia entre los parámetros presentados en la Figura 3.22 y 3.23 es la dirección en la que cambia el parámetro K , los demás parámetros no tienen variaciones significativas.

En este escenario de falla el algoritmo de identificación recibe un valor de nivel diferente del que se espera como se observa en la Figura 3.20 por lo que el parámetro K que se relaciona de forma directa con el valor de nivel es ajustado.

Una falla aditiva en el sensor de nivel produce una variación del parámetro K en el sentido inverso.

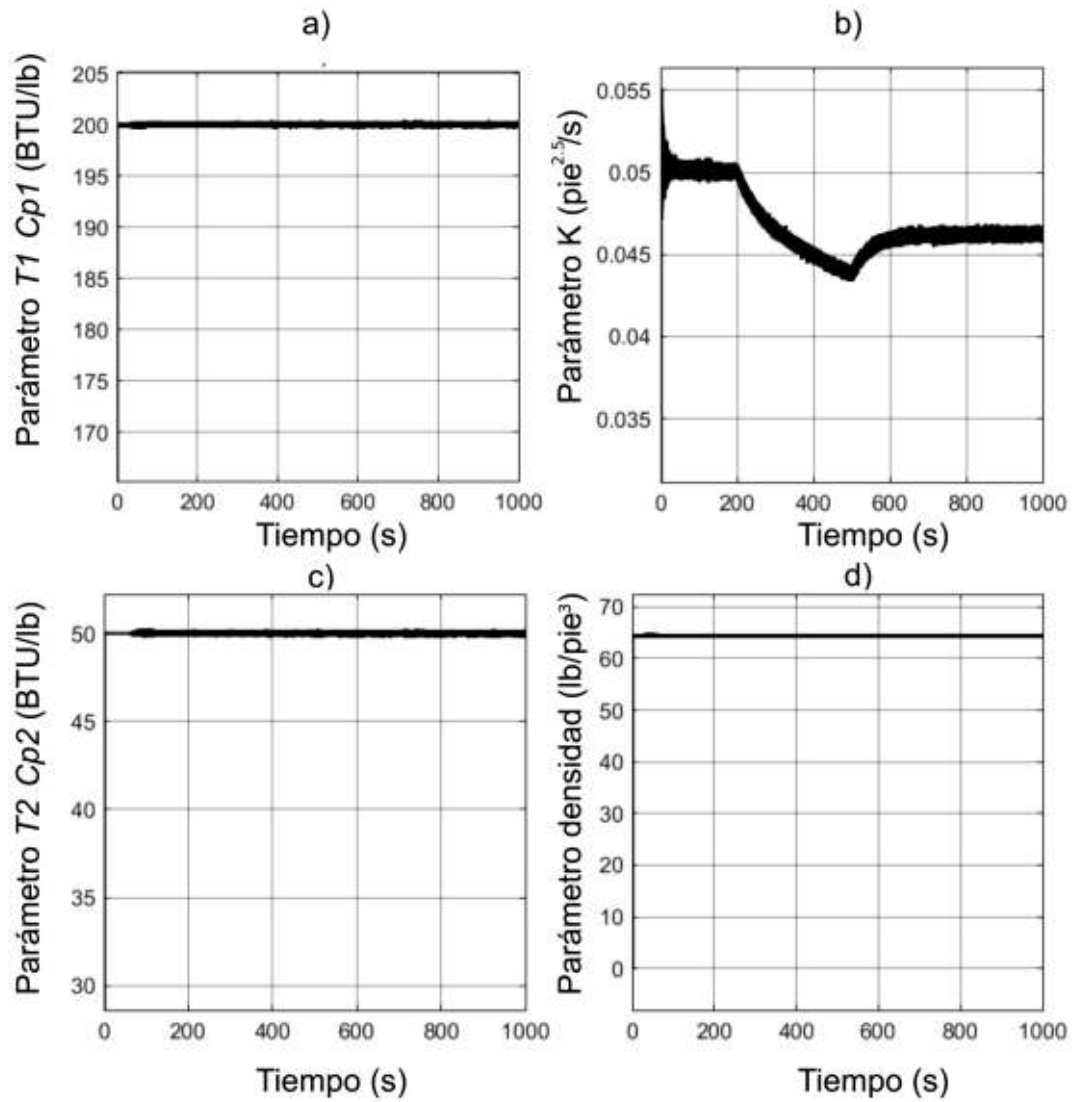


Figura 3.22 Estimación de parámetros con falla positiva en el sensor de nivel, a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

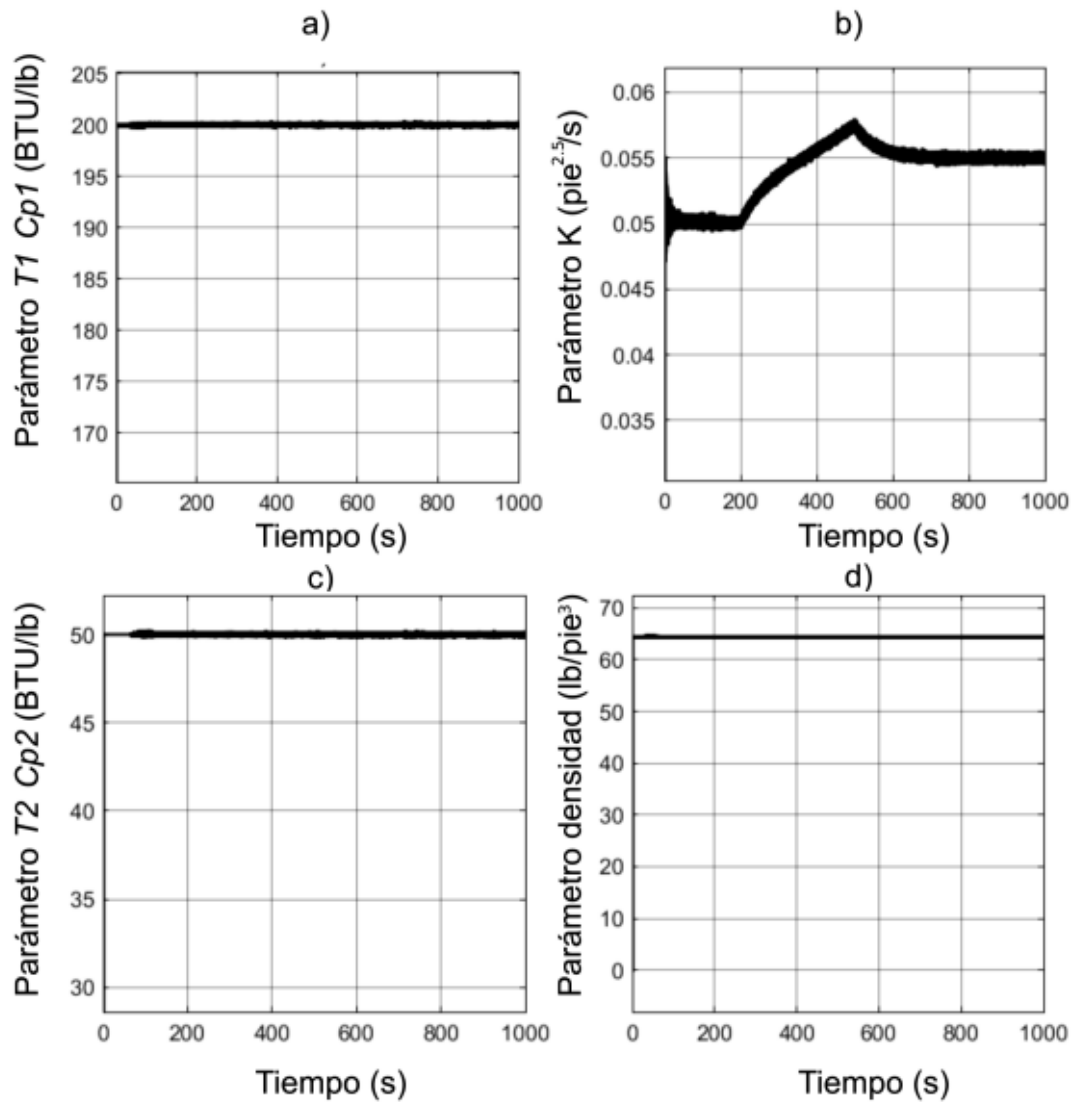


Figura 3.23 Estimación de parámetros con falla positiva en el sensor de nivel,

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.2.1.3 Falla en el Actuador de Flujo Frío

A continuación, se presenta la simulación de una falla en el actuador de flujo frío, la señal de error comienza a los 200 (s) y aumenta con una pendiente de 0.003 hasta los 500 (s).

La Figura 3.24 presenta la comparación entre el valor de la planta y el valor generado por el modelo del actuador en el bloque de generación de residuos.

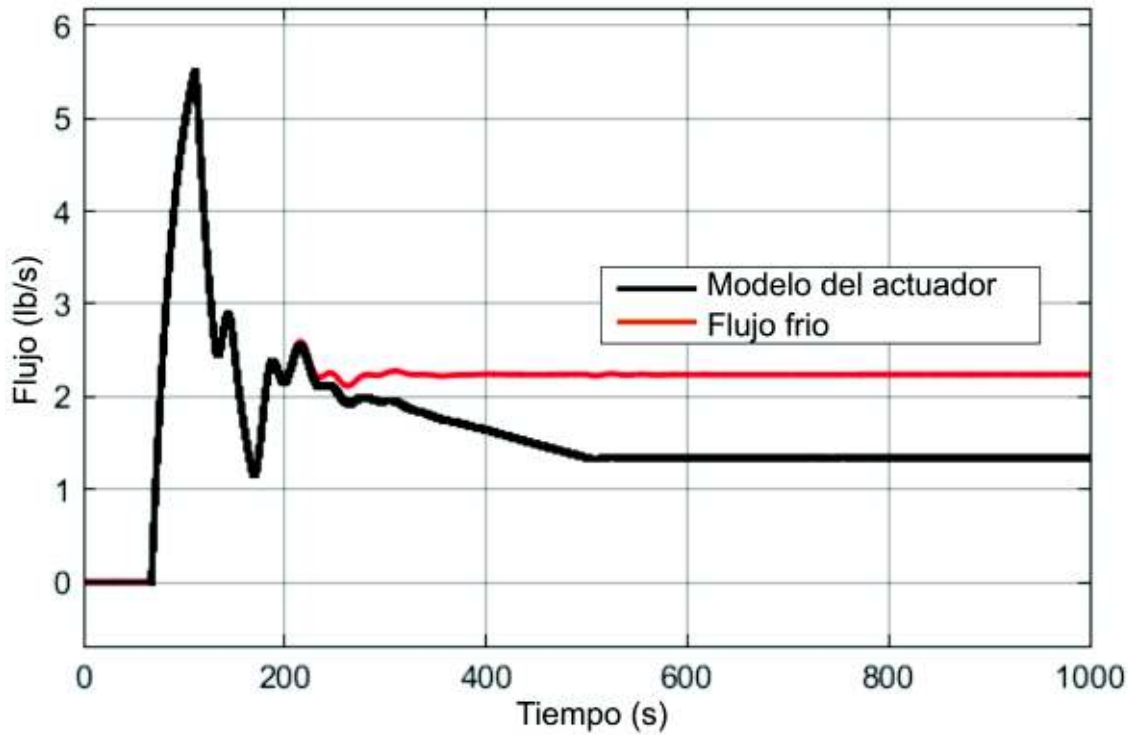


Figura 3.24 Falla positiva en la válvula de flujo frío.

A pesar de que se suma una perturbación en la válvula de flujo frío este permanece casi constante porque el controlador se encarga de compensar el error producido.

El modelo del actuador toma la señal de control para calcular el valor del flujo, esta señal fue regulada por el controlador por lo que se genera una discrepancia y el algoritmo de identificación recibe un valor de flujo diferente del real de la planta.

En la Figura 3.25 se presentan las variaciones de parámetros para este escenario. y en la Figura 3.26 se presenta la estimación de parámetros para una falla negativa en la válvula de flujo frío.

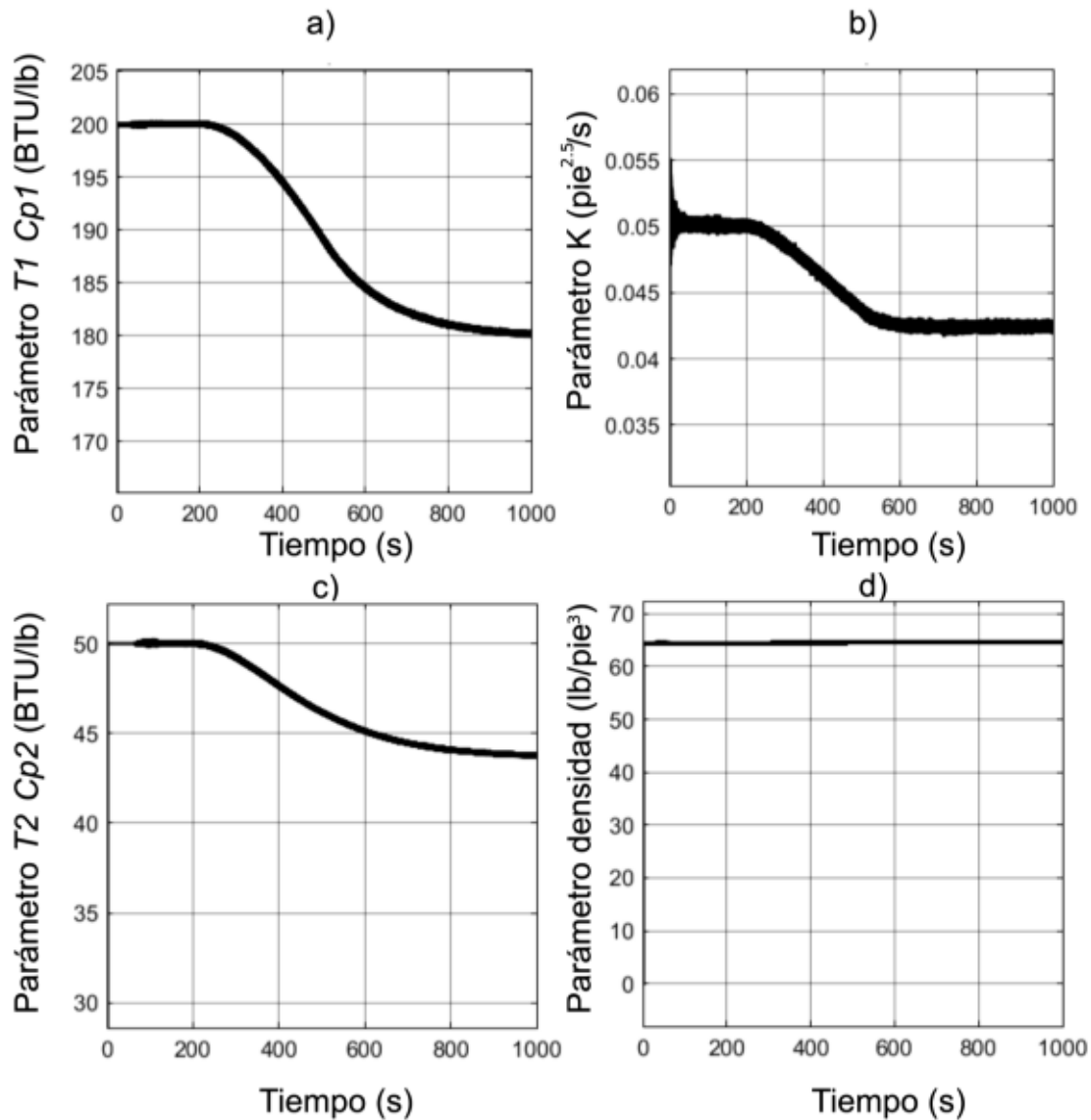


Figura 3.25 Estimación de parámetros con falla positiva en la válvula de flujo frío,

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

La diferencia entre la Figura 3.25 y 3.26 es la dirección de la variación en los parámetros, con la falla negativa los parámetros crecen mientras que con la falla positiva estos decrecen. Una falla en este componente provoca la variación de tres parámetros.

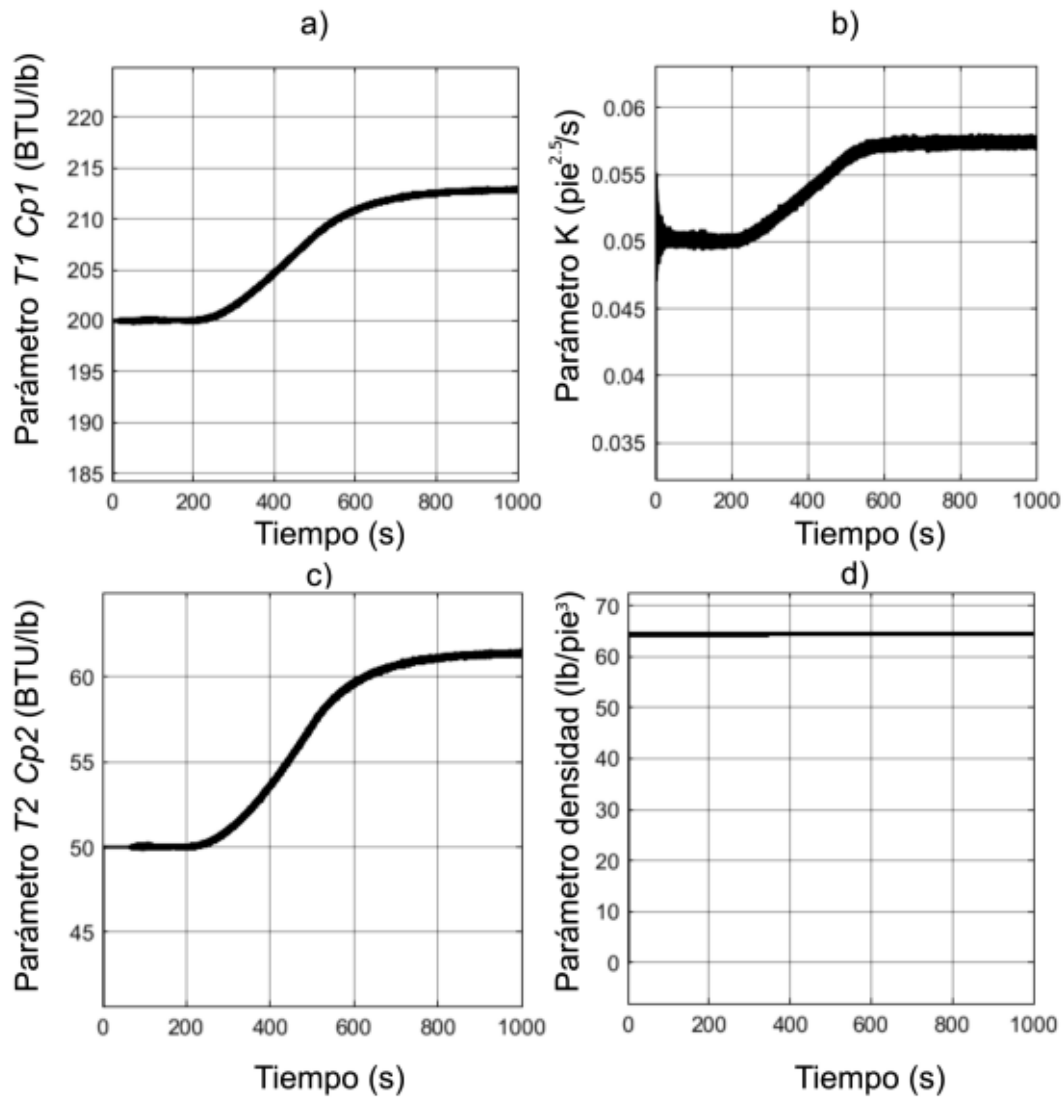


Figura 3.26 Estimación de parámetros con falla negativa en la válvula de flujo frío,

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.2.1.4 Falla en el Actuador de Flujo Caliente

En la Figura 3.27 se muestra la discrepancia entre el flujo en la válvula de flujo caliente y el valor calculado por el algoritmo cuando se añade una falla positiva, esta es la misma falla que se añadió a la válvula de flujo frío.

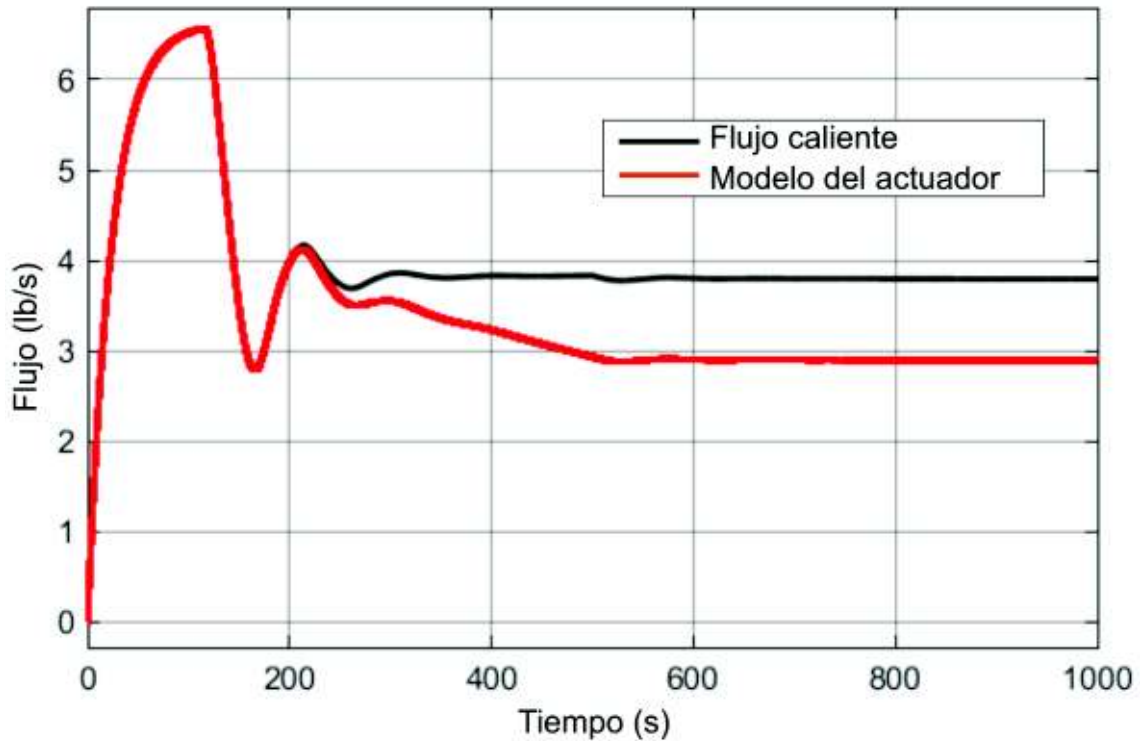


Figura 3.27 Falla en válvula de flujo Caliente.

Las desviaciones en los parámetros para este escenario de falla se muestran en 3.28, En la Figura 3.29 se presenta la variación en parámetros ante una falla negativa en la válvula de flujo caliente.

De los resultados expuestos en la Figura 3.28 y 3.29 se observa que en presencia de una falla de flujo caliente se producen variaciones en los parámetros $TICp1$, $T2Cp2$ y K . Mientras los primeros varían en la dirección de la falla K varía en la dirección opuesta.

La diferencia entre una falla en el actuador de flujo caliente y de flujo frío es la dirección de variación del parámetro K .

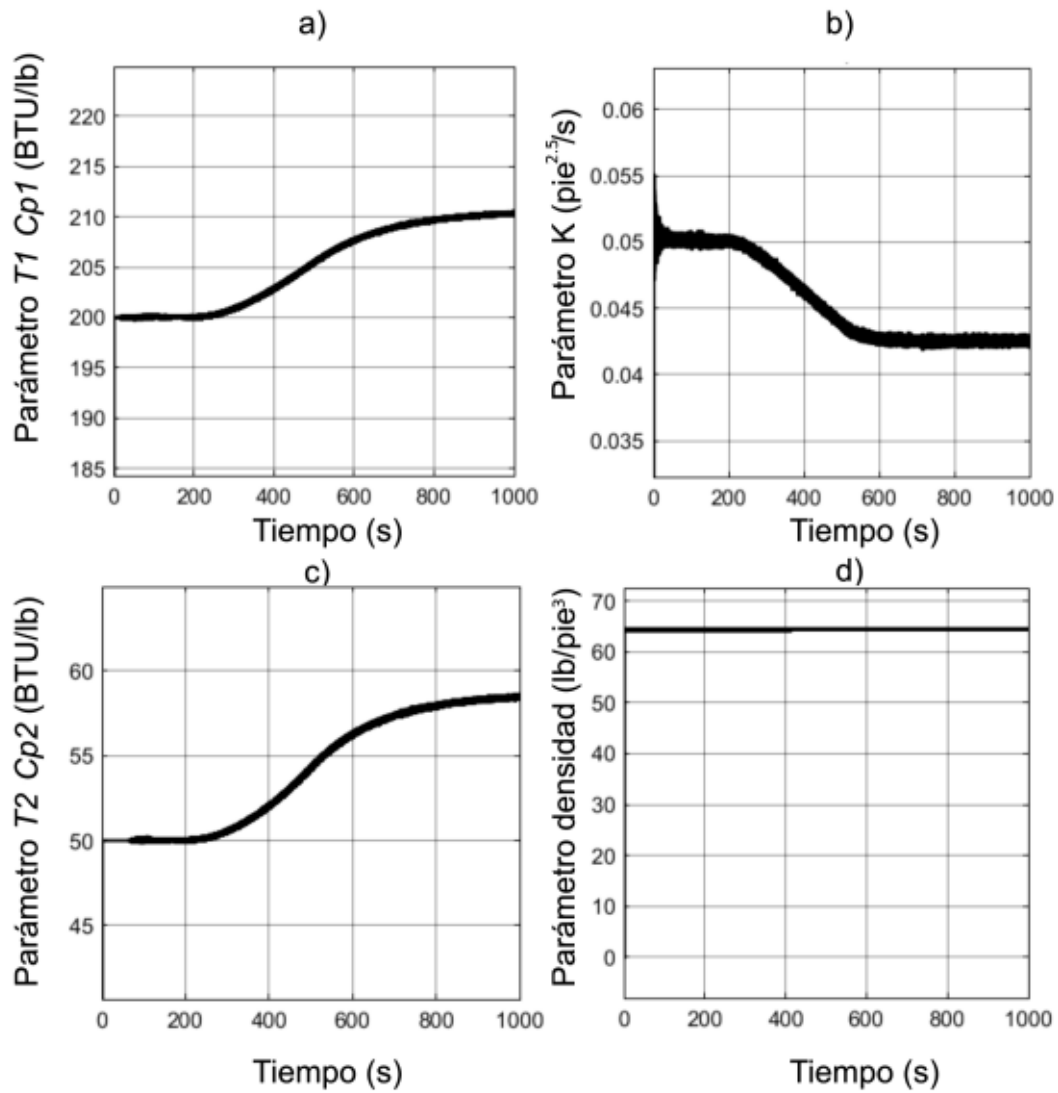


Figura 3.28 Estimación de parámetros con falla positiva en la válvula de flujo caliente,

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

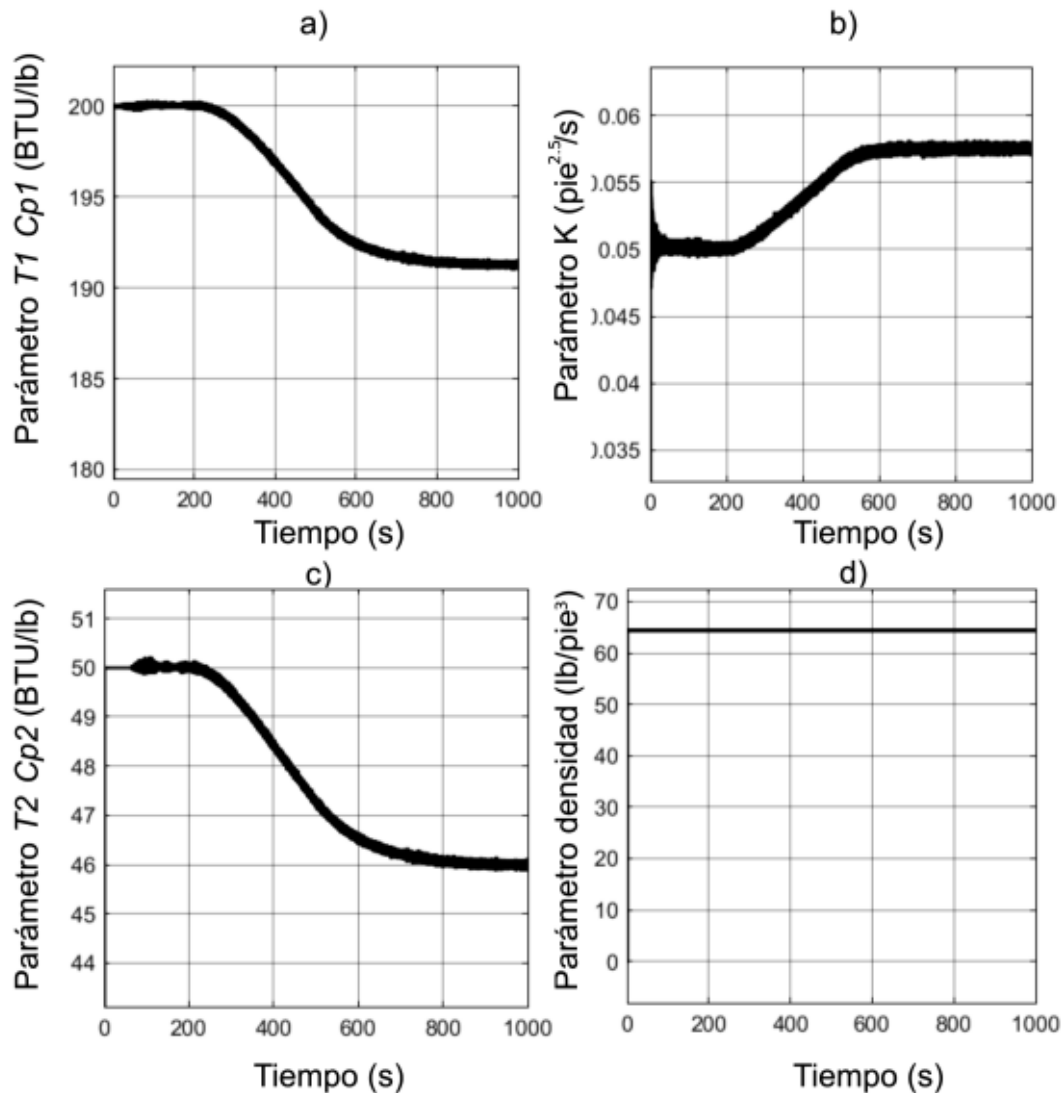


Figura 3.29 Estimación de parámetros con falla negativa en la válvula de flujo caliente,

a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.2.2 Fallas en la Planta

En esta subsección se presenta la respuesta del algoritmo de identificación de parámetros cuando se presentan variaciones de características de los líquidos como la densidad del líquido. También se considera una falla en el aislamiento térmico y una fuga de material del tanque.

3.1.2.2.1 Fuga de Material

Para simular una fuga de material en el tanque se añade en la ecuación diferencial de balance de material una falla con pendiente negativa de $2e-5$ entre

200 (s) y 500 (s). La variación de parámetros causada por esta falla se presenta en la Figura 3.30, en esta imagen se observa que se obtiene una variación en los parámetros similar al escenario de falla en el sensor de nivel. Esta superposición de los escenarios tiene como consecuencia que no se puede diferenciar entre una falla en el sensor de nivel y una fuga de material.

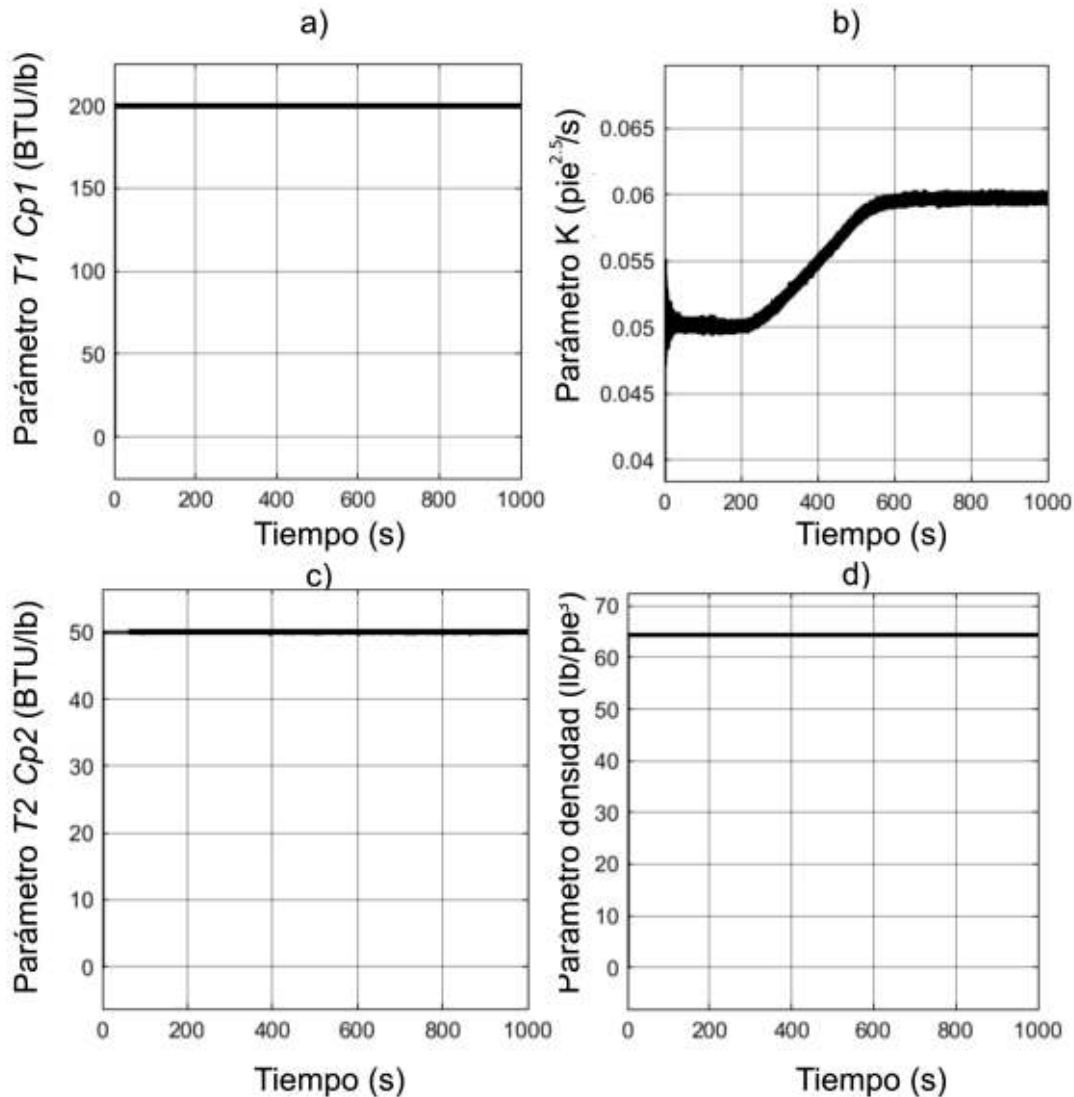


Figura 3.30 Estimación de parámetros con fuga de material a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.2.2.2 Falla en el Aislamiento Térmico

En este tipo de falla se añade una falla aditiva con pendiente negativa en la ecuación diferencial de balance térmico, la variación de parámetros causada por la presencia de la falla en el aislamiento se presenta en la Figura 3.31. En la figura se observa que en este escenario se obtiene variaciones paramétricas

similares a las que se obtiene con una falla en el sensor de temperatura por lo que no es posible diagnosticar una diferencia entre estas dos fallas.

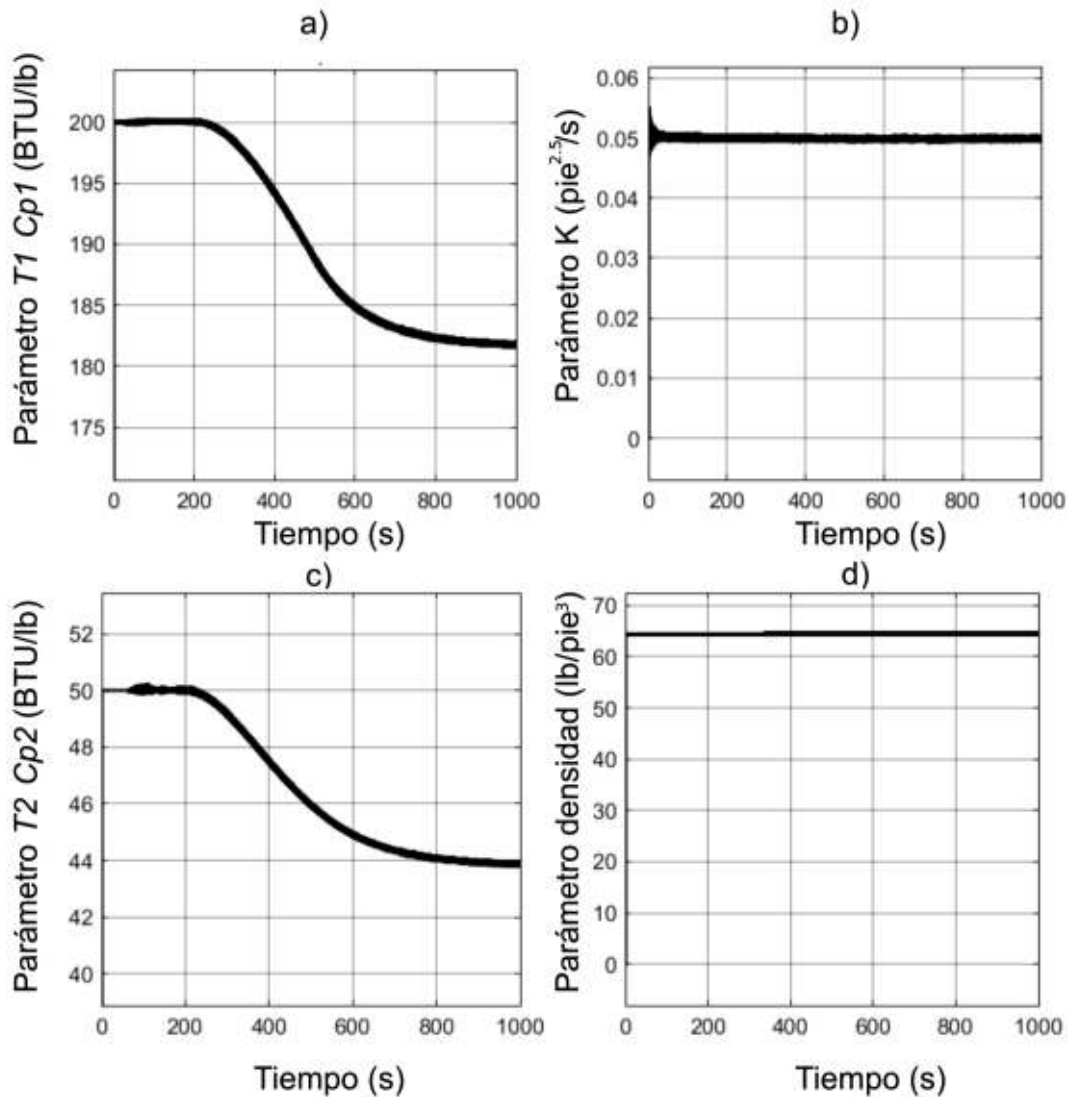


Figura 3.31 Estimación de parámetros con fuga de material a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.1.2.4.3 Cambio en la Densidad del Líquido

En este escenario se añade una perturbación al valor de la densidad en la planta, en la Figura 3.32 se presenta el cambio en parámetros que esta perturbación ocasiona, como se observa el estimador de parámetros no obtiene el equivalente que se esperaría, esto se debe a que no existe suficiente persistencia de excitación.

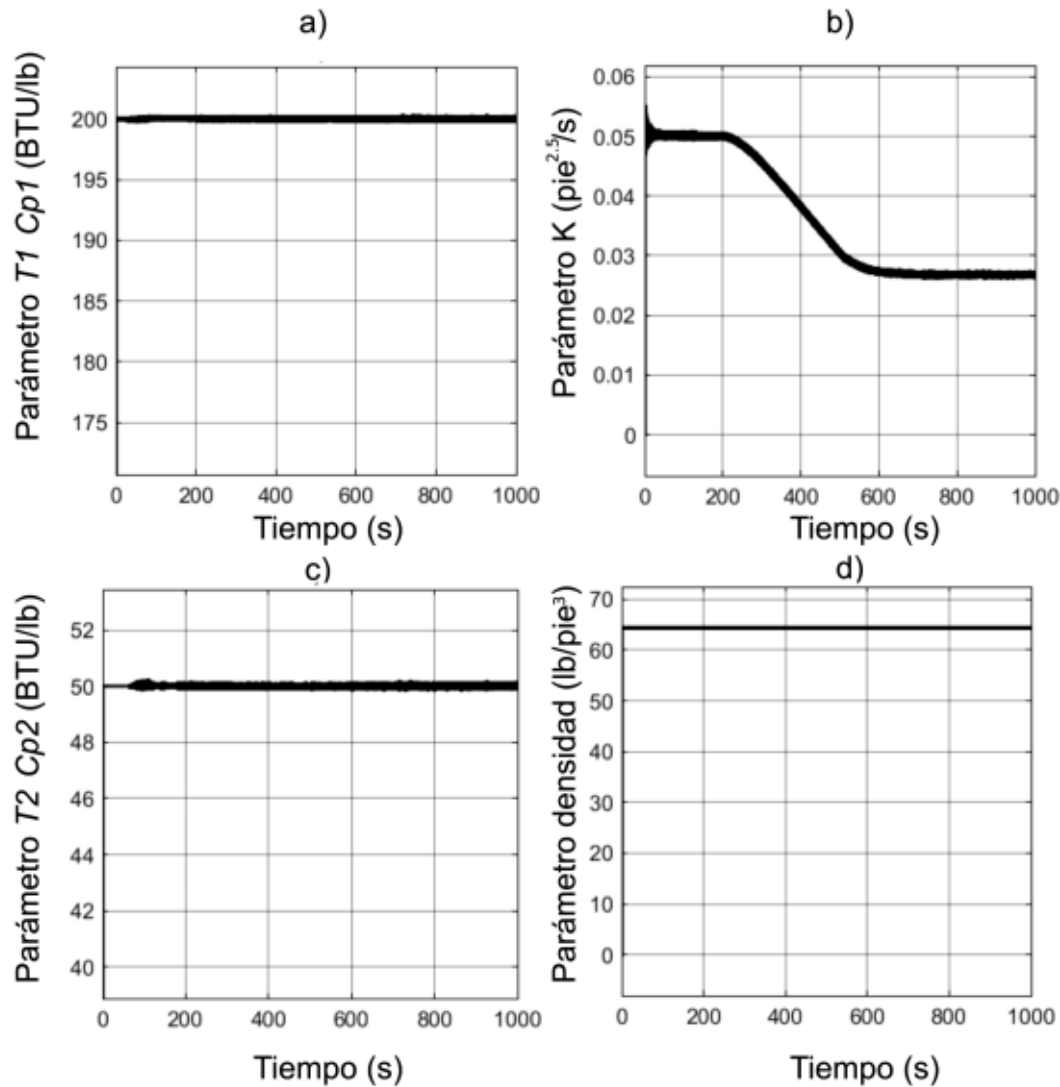


Figura 3.32 Estimación de parámetros con variación de densidad a) Parámetro $T_1 \cdot C_{p1}$, b) Parámetro K , c) Parámetro $T_2 \cdot C_{p2}$, d) Parámetro ρ .

3.2 AISLAMIENTO DE FALLAS.

Al momento de realizar una simulación de falla, como las ejecutadas en la sección 3.1, se conoce de antemano el tipo de falla que se introduce al sistema por lo que se puede emplear esta información junto a los residuos generados por el algoritmo de estimación para construir una base de datos para el entrenamiento de la red neuronal. En la Tabla 3.2 a continuación se presenta un resumen de los residuos obtenidos en la sección anterior.

Tabla 3.2 Ejemplo de Datos de entrenamiento

Residuos				Falla	Planta	Pendiente
7.5101	3.231	0.618	0.013	Sensor	Un Controlador	0.003
-5.1268	-3.7977	-0.5549	0.007	Sensor	Un Controlador	-0.003
-10.34	-5.43	-0.8853	0.002	Actuador	Un Controlador	0.05
8.0125	5.49	1.012	0.007	Actuador	Un Controlador	-0.05
36.7	1.15	3.69	0.01	Fuga Térmica	Un Controlador	-0.001
0.001	0.001	0.001	0.001	Fuga Nivel	Un Controlador	-0.0001
0.001	0.001	0.001	0.001	Densidad	Un Controlador	0.1
0.048	0.037	0.004	0.001	Sensor Nivel	Dos Controladores	0.002
0.049	0.037	-0.005	0.01	Sensor Nivel	Dos Controladores	-0.002
-15.78	-9.42	-9.2e-5	-0.059	Sensor Temperatura	Dos Controladores	0.05
15.87	9.502	-1.18e-4	-0.024	Sensor Temperatura	Dos Controladores	-0.05
-10.1	-8.85	0.008	0.008	Actuador Caliente	Dos Controladores	0.003
9.1	9.12	-0.008	0.008	Actuador Caliente	Dos Controladores	-0.003
19.89	6.253	0.0075	0.003	Actuador Frio	Dos Controladores	0.003
-12.828	-11.35	-0.0075	0.009	Actuador Frio	Dos Controladores	-0.003
17.45	6.01	0.0001	0.007	Fuga	Dos	2e-5

				Térmica	Controladores	
0.01	0.01	-0.001	0.006	Fuga Material	Dos Controladores	2e-4
-0.01	0.01	0.023	0.005	Densidad	Dos Controladores	0.1

Para realizar la clasificación se consideró fallas en cada uno de los instrumentos de la planta por lo que en la configuración de un controlador se tiene dos clasificaciones de falla y en la configuración de dos controladores se tiene cuatro.

Los datos presentados en la tabla son una parte pequeña del total de datos disponibles. Para entrenar a la red neuronal se realiza un mayor número de simulaciones, se emplea distintos valores para la pendiente y el tiempo inicial y final. Esto permite tener diferentes valores en la entrada lo que produce distintos niveles de convergencia de los parámetros.

En el anexo se presenta una guía de uso de la interfaz gráfica con la que se puede ejecutar más simulaciones para añadir valores a la base de datos.

El algoritmo de entrenamiento primero separa los datos en cada planta, para después entrenar una red neuronal por cada configuración de la planta.

Se emplea una división aleatoria de datos para entrenamiento 70/100, para validación de 15/100 y para prueba de 15/100.

En la Figura 3.33 se presenta la matriz de confusión que resume el resultado del entrenamiento para la configuración de un controlador. En la diagonal de la matriz de confusión se encuentra el número y porcentaje de estimaciones correctas efectuadas por la red neuronal entrenada. Los valores fuera de la diagonal son los valores en que el clasificador fallo.

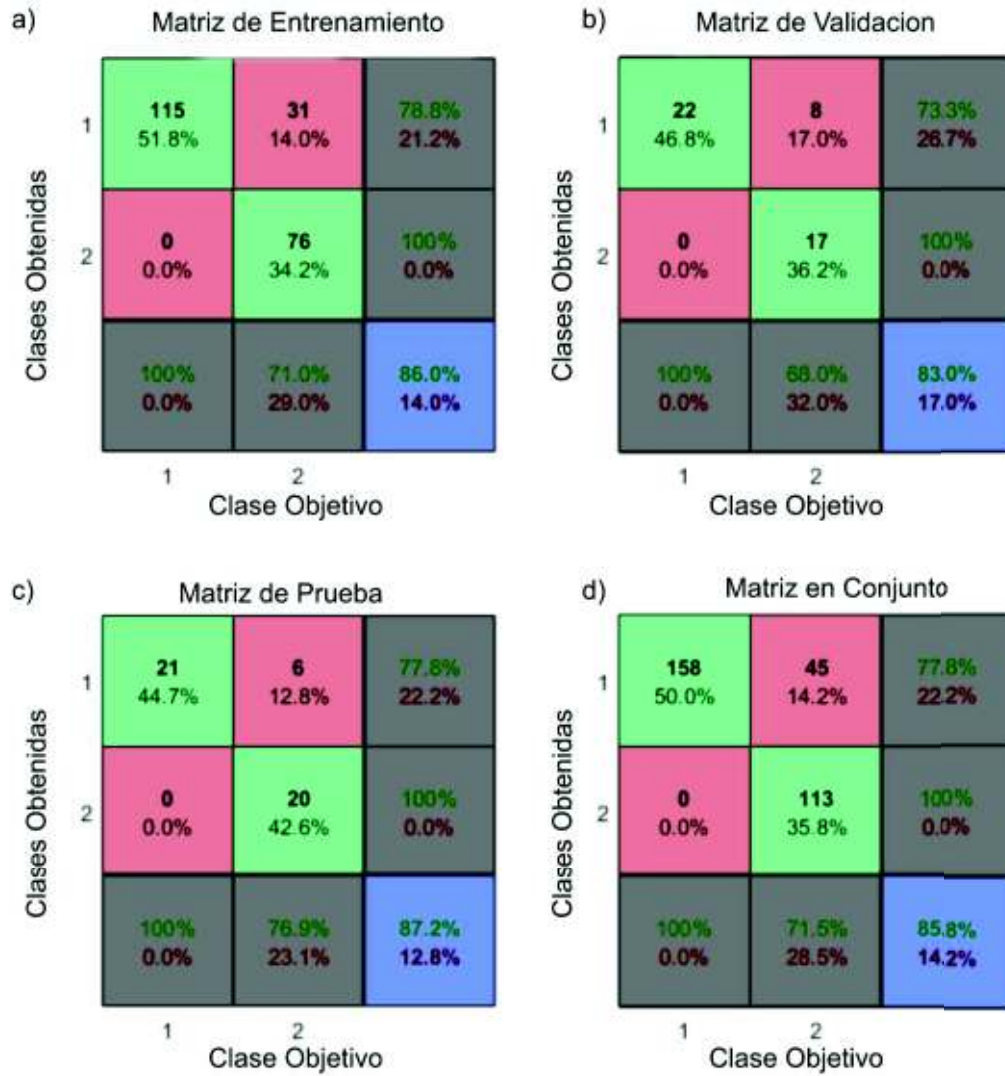


Figura 3.33 Clasificación, Configuración de un controlador.

En la Figura 3.34 se presenta la matriz de confusión que resume el resultado del entrenamiento para la configuración de dos controladores.

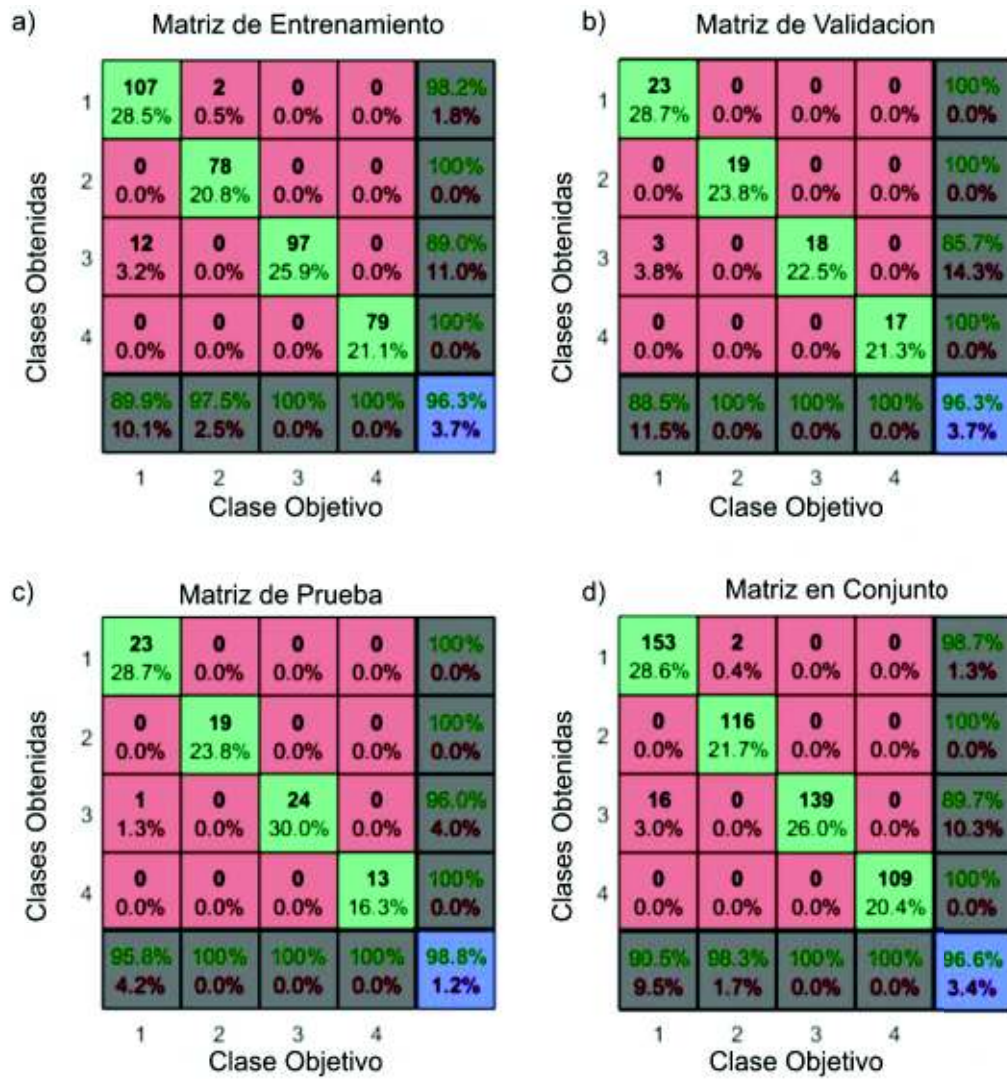


Figura 3.34 Clasificación, Configuración de dos controladores.

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Una falla introduce variaciones en la planta que pueden ser detectadas con un algoritmo estimador de parámetros como el filtro de Kalman Extendido.
- El filtro de Kalman permite tener un sistema de detección con una mejor resistencia frente al ruido o a perturbaciones en el sistema, pero también puede desensibilizar al algoritmo de fallas transitorias de pequeña magnitud por lo que se debe realizar un compromiso entre estos dos aspectos.
- Al momento de ordenar los datos empleados en el entrenamiento de la red neuronal también es posible obtener inmunidad a perturbaciones al considerar escenarios en los que los parámetros no tengan variaciones considerables como libres de falla, en este caso también se realiza un compromiso entre inmunidad y sensibilidad.
- Cuando se dispone de un mayor número de transmisores, el algoritmo de estimación de parámetros cuenta con una mayor información de la planta lo que brinda la posibilidad de obtener mejores estimados de una mayor cantidad de parámetros.
- Al contar con una mayor cantidad de datos para llevar a cabo el entrenamiento de la red neuronal se obtienen una clasificación más completa y por lo tanto existe menor posibilidad de errores al momento de aplicar la red en clasificaciones de datos posteriores.

4.2 RECOMENDACIONES

- Para realizar una estimación de parámetros utilizando el filtro de Kalman extendido es necesario tener valores iniciales de las variables de la planta, por lo que se recomienda para una implementación práctica realizar una lectura de sensores antes de comenzar el proceso para obtener los valores iniciales.
- El modelo de los actuadores y de los transmisores no se incluye en el algoritmo de estimación de parámetros, por lo que se recomienda explorar el desempeño de un algoritmo que cuente con un modelo de la planta más detallado.
- Se recomienda agregar más elementos de medición a la planta para mejorar el desempeño del algoritmo, por ejemplo, agregar sensores a las señales de entrada o incluir sensores redundantes.
- Se podría mejorar el desempeño de la clasificación utilizando técnicas de inteligencia artificial más avanzadas como redes neuronales autoajustables o evolutivas.

REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Guillén, J. Paredes, O. Camacho, "Una propuesta de detección y diagnóstico de fallas en la instrumentación de procesos químicos utilizando la transformada wavelet." Revista Técnica de la Facultad de Ingeniería Universidad del Zulia vol. 28, Enero 2005, pp 72-81.
- [2] S. Simani, C Fantuzzi, R. Patton, "Model based Fault Diagnosis in Dynamic Systems Using Identification Techniques", Springer 2003, pp 2-5, 32-40, 142-143.
- [3] S. Ding, "Model based Fault Diagnosis Techniques, Design Schemes, Algorithms, and Tools" springer 2008, pp 3-10.
- [4] K. Xiong, C. W. Chan, H. Y. Zhang, "Unscented Kalman Filter for fault detection"
IFAC 2005.
- [5] L. Li, "Fault Detection and Fault Tolerant Control for Nonlinear Systems" Springer Berlin 2015.
- [6] R. Carrasco "Detección y diagnóstico de fallas en robots móviles cooperativos" Pontificia Universidad Católica de Chile, Escuela de Ingeniería, Departamento de Ingeniería Eléctrica, Agosto 2004.
- [7] R. Isermann, "Fault-Diagnosis Applications, Model-based condition monitoring: Actuators, drives, machinery, plants, sensors and fault-tolerant systems" Springer, Berlin 2011, pp 34-40.
- [8] R. Isermann, "Fault-Diagnosis Applications, Model-based condition monitoring: Actuators, drives, machinery, plants, sensors and fault-tolerant systems" Springer, Berlin 2011, pp 34-40.
- [9] M. Grewal, A. Andrews, "Kalman Filtering, Theory and Practice Using MATLAB", Cuarta Edición, John Wiley & Sons, Inc. 2015 pp 1.
- [10] R. Grover, P. Hwang, "Introduction to Random Signals and Applied Kalman Filtering", Cuarta Edición, John Wiley & Sons, Inc. 2012 pp 143-148.

- [11] B. Gibbs, "Advanced Kalman Filtering Least Squares and Modeling" John Wiley & Sons, Inc. 2011, pp. 249-250.
- [12] J. Crassidis, J. Junkins, "Optimal Estimation of Dynamic Systems" 2nd ed, CRC Press 2012 pp 184 .
- [13] R. Isermann, M. Münchhof, "Identification of Dynamic Systems, An Introduction with Applications" Springer, Berlin 2011 pp 548-549.
- [14] K. Keesman, "System Identification An Introduction" Springer 2011 pp 212.
- [15] K. Atkinson, W. Han, D. Stewart "Numerical Solution of Ordinary Differential Equations" pp 70-74.
- [16] I. Nunes da Silva, D. Hernane, R. Andrade, L. Bartocci, S. Franco, "Artificial Neural Networks A Practical Course" Springer 2017, pp 29-30, 33-34 .
- [17] B. Ogunnaike, W. Harmon Ray, "Process Dynamics, Modelling and Control " Oxford University Press 1994, pp 691-693.
- [18] O. Camacho, C. Smith, "Sliding mode control: an approach to regulate nonlinear chemical processes" ISA Transaction 39 · Febrero 2000, pp. 205-218

ANEXO A.

MANUAL DE USO DE LA INTERFAZ

El software desarrollado consiste en una interfaz capaz de ejecutar la simulación de la planta en simulink con las condiciones iniciales preestablecidas. Se permite seleccionar entre las dos configuraciones de la planta y añadir una falla aditiva tipo rampa a los instrumentos de la planta.

Para abrir la interfaz en matlab se debe ejecutar el archivo inter.m que se incluye en la carpeta de interfaz. La siguiente figura presenta la pantalla que se abre al ejecutar el archivo.

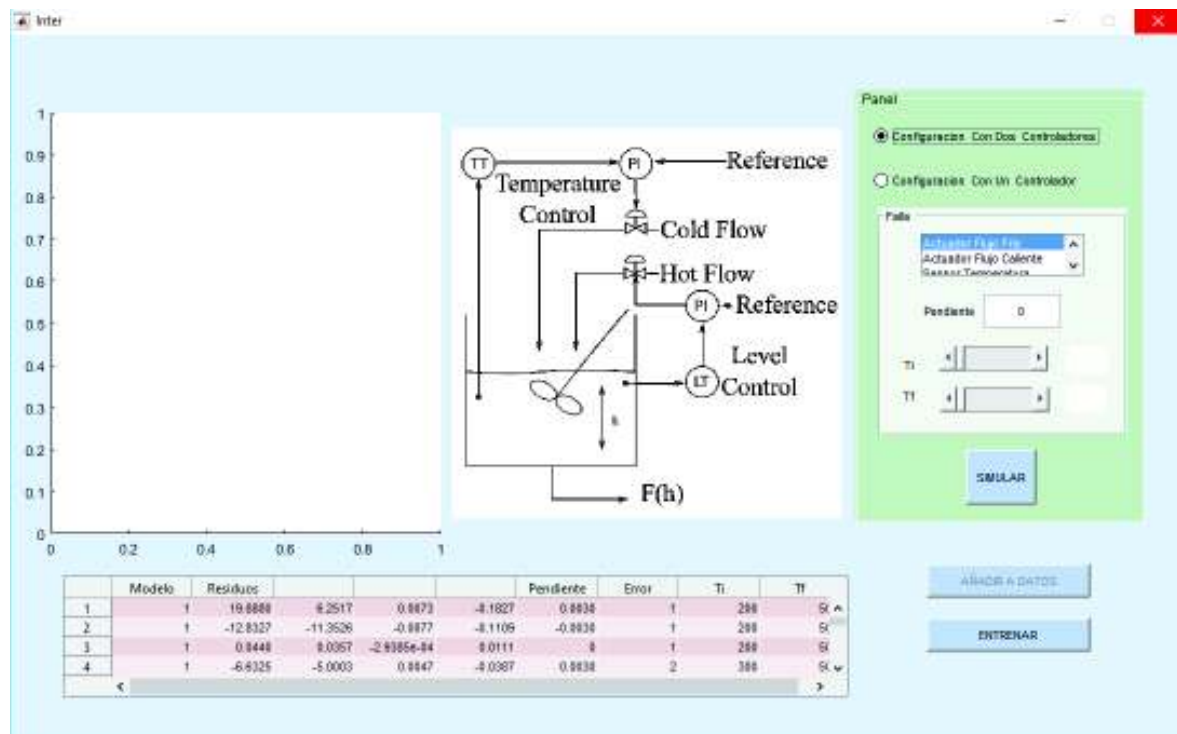


Figura A.1 Interfaz Gráfica.

En el sector derecho está ubicado el Panel donde se puede seleccionar qué planta se desea simular y las opciones de la falla, el valor de la pendiente y entre que tiempo se aplica esta pendiente.

Una vez que se lleva a cabo la simulación, que puede demorar varios segundos en ejecutarse, dependiendo de la máquina que corra la simulación, se actualiza el gráfico en la derecha para indicar el resultado de la clasificación de error de la red neuronal, como se presenta en la Figura A.2.

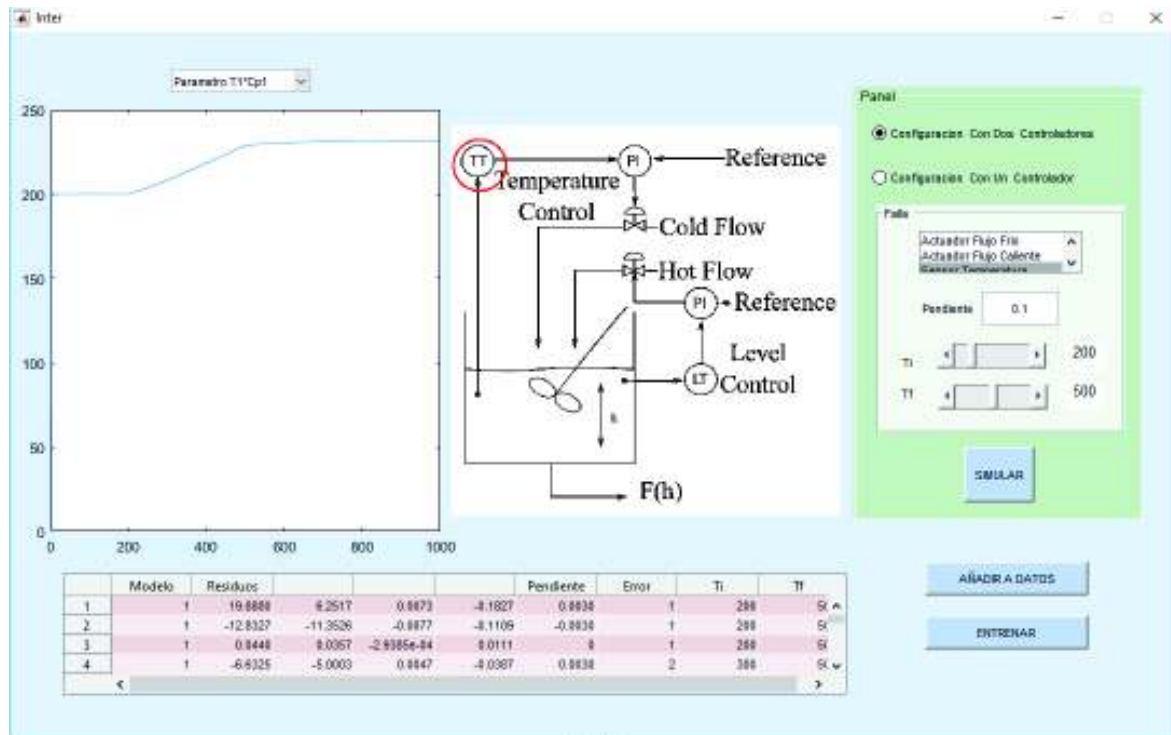


Figura A.2 Interfaz Gráfica.

En la parte superior de gráfico izquierdo aparece un selector que nos permite visualizar la evolución de los distintos parámetros que son estimados por el sistema de detección de fallas.

Ahora que se ha ejecutado una simulación, se activa la opción de añadir el resultado de esta a la base de datos que aparece en la tabla inferior. Solamente se puede añadir el resultado una vez, en caso de ser añadido el botón de añadir datos vuelve a desactivarse.

El botón de entrenar elimina las redes neuronales existentes y entrena nuevas empleando la base de datos que se presenta en la interfaz.

La interfaz fue realizada utilizando **MATLAB R2015a** por lo que se recomienda emplear esta versión o una superior para ejecutarla.