“Técnicas de monitoreo en línea para optimización de mantenimiento en centrales nucleares”

CARRERA: ESPECIALIZACIÓN EN REACTORES NUCLEARES Y SU CICLO DE COMBUSTIBLE

Alumno: Ing. Pablo Crubellier

Director: Lic. Guillermo Urrutia

Diciembre – 2018

Contenido

1. OBJETIVOS 4
2. INTRODUCCIÓN GENERAL 5
2.1. INTRODUCCIÓN 5
2.2. PRINCIPIOS DE MONITOREO ON LINE 5
2.3. BENEFICION DEL MONITOREO EN LA CALIBRACIÓN 6
2.4. DESCRIPCIÓN BASICA DE SISTEMAS DE MONITOREO 6
2.5. ANALISIS PARA SISTEMAS DE MONITOREO EN LINEA 7
2.6. MODELOS DE PREDICCIÓN: TÉCNICAS ESTÁTICAS 9
2.6.1. MODELOS NO PARAMÉTRICOS 9
2.6.2. MODELOS PARAMÉTRICOS 10
3. TÉCNICAS APLICADAS 11
3.1. REGRESIÓN KERNEL 11
3.1.1. REGRESIÓN KERNEL HETEROASOCIATIVA 16
3.1.2. REGRESIÓN KERNEL AUTOREGRESIVA 16
3.2. TÉCNICAS PARAMÉTRICAS APLICADAS 17
3.2.1. REDES NEURONALES 17
3.2.2. DEFINICIÓN DE REDES NEURONALES 18
3.2.2.1. APRENDIZAJE DE UNA RED NEURONAL 18
3.2.2.2. APRENDIZAJE SUPERVISADO 18
3.2.2.3. APRENDIZAJE NO-SUPERVISADO 19
3.2.2.4. MODELO BÁSICO DE UNA NEURONA 19
3.2.2.5. TIPOS DE FUNCIÓN DE ACTIVACIÓN 20
3.2.2.6. TOPOLOGÍA DE UNA RED 22
3.2.2.7. PROPAGACIÓN DE LAS SEÑALES 23
3.2.3. REDES NUEURONALES AUTOREGRESIVAS 23
3.3. MODULO DE COMPARACION 24
3.4. LOGICA DE DECISION 25
4. RESULTADOS REGRESIÓN KERNEL AUTOREGRESIVA 28
4.1. CONSTRUCCION DE MATRIZ DE ENTRENAMIENTO 30
4.2. SELECCIÓN DE ANCHO DE BANDA 30
4.3. ANALSIS DE α, β Y μ 31
4.4. ESTIMACIÓN DE VARIABLES 33
4.4.1. CASO SIN FALLA 34
4.4.2. DRIFT EN TEMPERATURA 1 36
4.4.3. DRIFT EN TEMPERATURA 1 Y 2 38
4.4.4. DRIFT EN TEMPERATURA 2 40
4.4.5. DRIFT EN PRESIÓN 42
4.4.6. DRIFT EN TEMPERATURA 1, 2 Y PRESIÓN 44
4.5. CONCLUSIÓN REGRESIÓN KERNEL AUTOREGRESIVA 46
5. RESULTADOS REDES NEURONALES AUTOREGRESIVAS 47
5.1. ESTIMACIÓN DE VARIABLES 47
5.1.1 CASO SIN FALLA 48
5.1.2 DRIFT EN TEMPERATURA 1, 2 Y PRESIÓN 50
5.2. CONCLUSION REDES NEURONALES AUTOREGRESIVA 52
6. CONCLUSIÓNES 53

7. REFERENCIAS 54
8. ANEXOS 55

1. OBJETIVOS

El presente trabajo final tiene por objetivo el estudio, análisis y desarrollo de sistemas de monitoreo en línea, que facilitan el diagnostico e identificación de anomalías presentes en sensores y transmisores pertenecientes a cadenas de instrumentación de centrales nucleares.

La mayoría de los sensores y transmisores requieren de validación y calibración para asegurar el buen desempeño de los componentes involucrados, estos procesos implican tiempo de mantenimiento, costos monetarios y dosis al personal originados por aquellos sensores que se encuentren en zona controlada. Los sistemas de monitoreo pueden proveer información continua en cuanto el estado de la instrumentación disminuyendo la intervención del personal, rediciendo costos y exposición radiológica.

En el presente trabajo se estudian diversas técnicas matemáticas para el desarrollo de algoritmos de monitoreo, se evalúa el desempeño de cada una de ellas tomando como sistema a modelar las señales pertenecientes a la instrumentación de uno de los Generadores de Vapor de la Central nuclear Atucha II. El resultado de este estudio busca determinar pros y contras de las técnicas usadas, como así también, cual de ellas es la mas apta para ser aplicada en el sistema seleccionado.

2. INTRODUCCIÓN GENERAL

Se describen brevemente los principios de monitoreo on line adoptados y proceso básico de un sistema de monitoreo.

2.1. INTRODUCCIÓN

Actualmente las prácticas de mantenimiento sobre la instrumentación de centrales nucleares son abordadas de manera conservativa. En la mayoría de los instrumentos, tales como, transmisores de temperatura y transmisores de presión se realizan calibraciones completas en cada para programada, esto trae aparejado costos de mantenimiento, tiempos fuera de servicio y dosis al personal.

Actualmente hay más de 30 años de experiencia operativa en más de 400 centrales nucleares en todo el mundo. Esta experiencia muestra que la mayoría de las centrales nucleares presentan un buen desempeño en cuanto a la estabilidad de la instrumentación, lo que lleva a reevaluar algunas prácticas de mantenimiento.

El monitoreo en línea se refiere a métodos de monitoreo de señales mientras la planta esta en operación, sin interferir en la instrumentación de esas señales. El sistema de monitoreo en línea provee un continuo estado del canal de instrumentación informando cualquier anomalía o falla. Esto permite diagnosticar, identificar y anticipar problemas sobre la instrumentación, como así también, validar el buen funcionamiento sin intervención del personal.

2.2. PRINCIPIOS DE MONITOREO ON LINE

Los procesos de calibración convencional sobre la instrumentación implican básicamente dos pasos. El primer paso es establecer el estado de calibración, esto se realiza mediante la desconexión del transmisor del proceso y posterior ensayo del instrumento en base a criterios de aceptación establecidos. El segundo paso es realizar la calibración en caso de que sea necesario, es decir, si el instrumento no cumple con el criterio de aceptación preestablecido, se realiza la calibración.

Los sistemas de monitoreo en línea se basan en datos históricos de señal, la colección de datos implica la revisión de distintos períodos de tiempo en los cuales la planta esta en operación, durante este proceso la instrumentación no se ve afectada, el proceso de recolección de datos es pasivo, no intrusivo pero a su vez in situ.

Los sistemas de monitoreo realizan el seguimiento de las salidas de los canales de instrumentación al lo largo de todo el ciclo de operación, identifican corrimientos, errores, ruido y otras anomalías. La ventaja de esto, es que identifica problemas de calibración que puedan ocurrir, como efectos propios del proceso sobre la calibración y evita calibraciones innecesarias.

2.3. BENEFICION DEL MONITOREO EN LA CALIBRACIÓN

El uso de sistemas de monitoreo en línea permite reducir el número de calibraciones requeridas en cada parada programada de una central nuclear, pero además se destacan otros beneficios:

- Reducir la exposición a la radiación del personal.
- Reducir la posibilidad de daño sobre el instrumento durante la calibración.
- Reducir los tiempos y costos de mantenimiento.
- Verificación y disponibilidad continua del estado de la instrumentación.
- Identificación de condiciones anormales.
- Acción inmediata en respuesta a la identificación de condiciones anormales.

2.4. DESCRIPCIÓN BASICA DE SISTEMAS DE MONITOREO

Antes de discutir características especificas del análisis de sistemas de monitoreo, es necesario comprender las funciones básicas que componen un sistema de monitoreo en línea. Los métodos de análisis que son usados en este trabajo son de característica empírica y son explicados en detalle en las secciones subsiguientes.

Los modelos empíricos usan datos históricos de planta para construir el modelo predictivo, una vez que el modelo predictivo es construido, es puesto en modo de monitoreo para proveer el mejor estimado de la variable.

La figura 1 representa en bloques el sistema de monitoreo. En esta figura la señal *(x)* proviene de los sensores, es la entrada del modelo predictivo, el cual calcula el mejor estimado de la variable *(x').* Los valores estimados son comparados con los valores medidos formando los valores diferencias llamados residuos *(r).* Un bloque de decisión determina si los residuos son estadísticamente diferentes de cero y establece el estado del sensor (normal, anormal, error, necesidad de calibración, etc.) según se defina bajo criterio establecido.

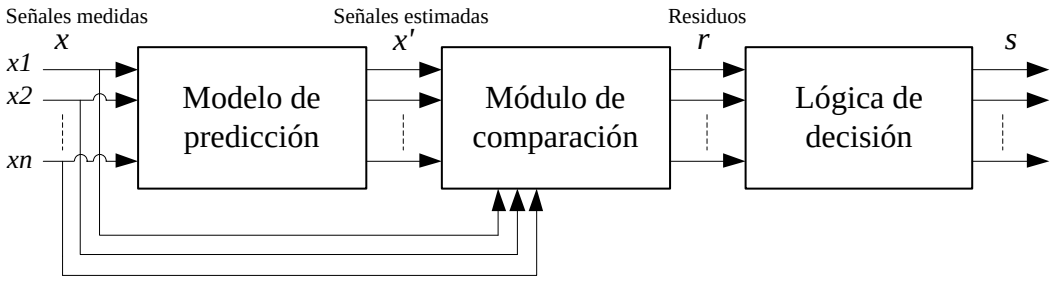

Figura 1. Bloque básico del sistema

2.5. ANALISIS PARA SISTEMAS DE MONITOREO EN LINEA

El análisis de datos usado en monitoreo en línea esta dividido en dos grandes categorías: métodos dinámicos y métodos estáticos. Los métodos dinámicos se refieren a aquellos que realizan el análisis de altas frecuencias de la respuesta del sensor, estas frecuencias son debidas a fluctuaciones inherentes a parámetros de los procesos tales como, turbulencias, vibraciones, transferencia de calor, etc. Los métodos estáticos se refieren a los que implican análisis de baja frecuencia, es decir, al estudio de series temporales historiadas. La figura 2 muestra la clasificación general de los distintos métodos de monitoreo en línea.

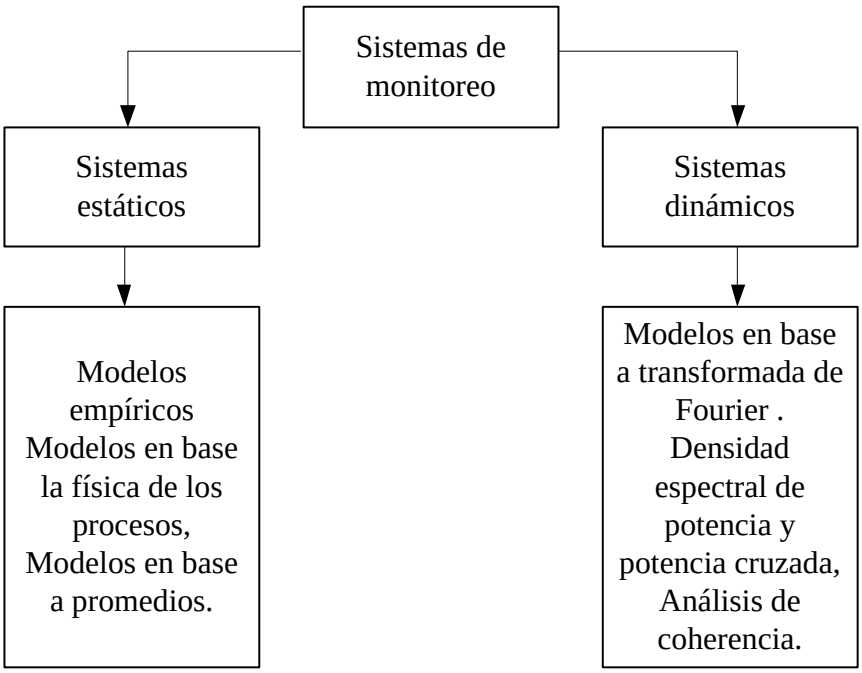

Figura 2. Métodos dinámicos y estáticos.

En el presente trabajo se hace focos en los métodos estáticos y su análisis de desempeño en busca de optimizar su funcionamiento. La figura 3 muestra una subdivisión de los distintos criterios adoptados para métodos estáticos.

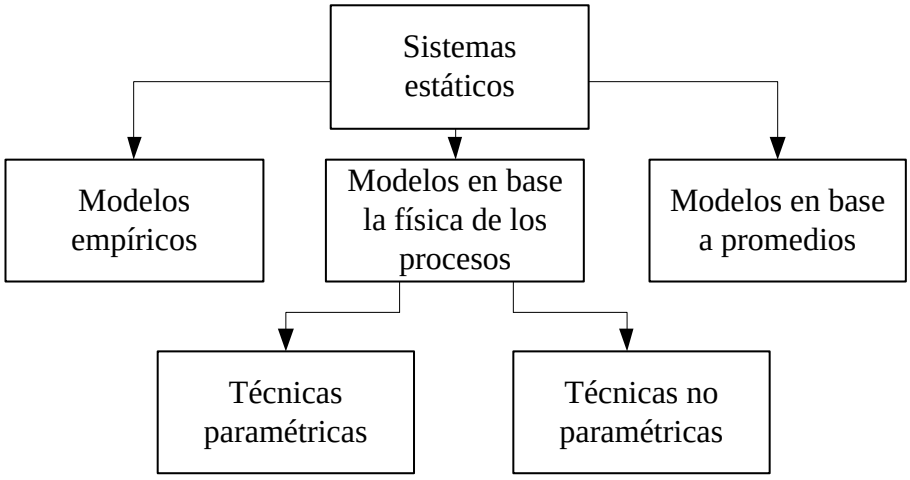

Figura 3. Métodos estáticos.

Los métodos estáticos se puede dividir en tres categorías: técnicas físicas, técnicas promediales y técnicas empíricas. En el presente trabajo se realiza el análisis comparativo de técnicas empíricas.

2.6. MODELOS DE PREDICCIÓN: TÉCNICAS ESTÁTICAS

Las técnicas empíricas hacen uso de datos históricos para realizar la predicción de datos futuros. Las técnicas en base a los datos físicos, son modelos construidos en base a ecuaciones derivadas de la física del sistema en análisis, el uso de modelos físicos implica mayor tiempo de desarrollo siendo poco favorable en comparación con modelos empíricos. Por otro lado, las técnicas que usan promedios son las mas simples de desarrollar pero de poco desempeño.

En este trabajo se usan técnicas empíricas cuyos modelos derivan directamente de los datos de proceso. La mayor ventaja de estas técnicas es que son capaces de modelar sistemas complejos con moderado costo de desarrollo. Existen dos grandes grupos de modelos a partir de técnicas empíricas: paramétricos y no paramétricos.

2.6.1. MODELOS NO PARAMÉTRICOS

Las técnicas no paramétricas se basan en criterios estadísticos que no pueden ser definidos con anterioridad, los datos observados no definen el modelo a priori. La utilización de estos métodos se hace recomendable cuando no se puede asumir que los datos se ajusten a una distribución conocida. Las técnicas no paramétricas usadas en sistemas de monitoreo, hacen uso de datos históricos almacenados en memoria y los procesan para obtener una predicción de valor.

Las técnicas basadas en regresión no paramétrica son formas de análisis de la regresión, en el que el predictor no tiene una forma predeterminada, sino que se construye de acuerdo a la información derivada de los datos. La Regresión Kernel pertenece a los métodos de regresión no paramétricos y se compone de tres pasos: un paso de ajuste, en el que intentamos encontrar la mejor combinación del tipo de modelo, la función kernel, y el ancho de banda, utilizando una muestra de prueba; un paso de validación, que permite validar el modelo sobre nuevas observaciones para las que se conoce una predicción; y un paso de aplicación, en que el modelo se aplica a un nuevo conjunto de datos para los cuales se desconoce la predicción. Los detalles de las técnicas usadas serán expuestos en las siguientes secciones.

2.6.2.MODELOS PARAMÉTRICOS

Los modelos paramétricos son aquellos en los cuales la correlación de datos históricos es incorporada en el modelo en forma de parámetros. Se basan en especificar una forma de distribución y de los estadísticos derivados de los datos. Se asume que la población de la cual la muestra es extraída es conocida y los parámetro pueden definirse.

Las técnicas basadas en redes neuronales son un caso de modelos paramétricos, operan de forma análoga al comportamiento observado en las neuronas biológicas. Cada neurona está conectada con muchas otras y los enlaces entre ellas, mediante factores de peso, pueden incrementar o inhibir el estado de activación de las neuronas adyacentes, es decir, incorporan una relación causa efecto entre los parámetros de peso que las constituyen. Los detalles sobre redes neuronales serán expuestos en secciones subsiguientes.

3. TÉCNICAS APLICADAS

Se describen las dos técnica no paramétricas aplicadas, Regresión Kernel en las versiones Heteroasociativa (HKR), Autoregresiva (AAKR), y redes neuronales (RN) respectivamente.

3.1. REGRESIÓN KERNEL

La Regresión Kernel es una técnica no paramétrica que usa datos históricos para realizar la predicción.

Si se consideran *n* observaciones de la entrada *(X)* y la salida *(Y),* la estimación de la salida para una entrada se realiza en base a la siguiente ecuación:

$$\hat{y}(x) = \frac{\sum_{i=1}^{n} [K(X_i - x) Y_i]}{\sum_{i=1}^{n} K(X_i - x)} \qquad (1)$$

Donde, *n* es el número de muestras de observaciones del modelo, *Xi* y *Yi* son la entrada y salida del i-ésimo valor observado respectivamente, *x* es la entrada de consulta, que se corresponde con el nuevo dato medido, *K(Xi –x)* es el peso o función de Kernel, la cual genera factores de peso para una diferencia dada entre la consulta y la muestra de observación y *y'(x)* es la estimación del valor *x*.

Básicamente, es un mecanismo de construcción de la variable *y'(x)* ponderando los valores muestreados de esta variable, según la distancia que haya desde el valor de *x* a los valores observados de la variable independiente. La ponderación se materializa mediante la función de Kernel.

La estimación mediante Regresión Kernel consta de tres etapas. Primero se calcula la distancia entre la variable de consulta x y las observaciones de entrada X, posteriormente los valores de distancia son suministrados como entrada de la función Kernel, la cual convierte las distancias en factores de peso (similaridad). Finalmente, los factores de peso son usados para calcular la estimación de salida por medio del promedio ponderado de los factores de peso con las observaciones de salida Y. Estos pasos se manifiestan en la figura 4.

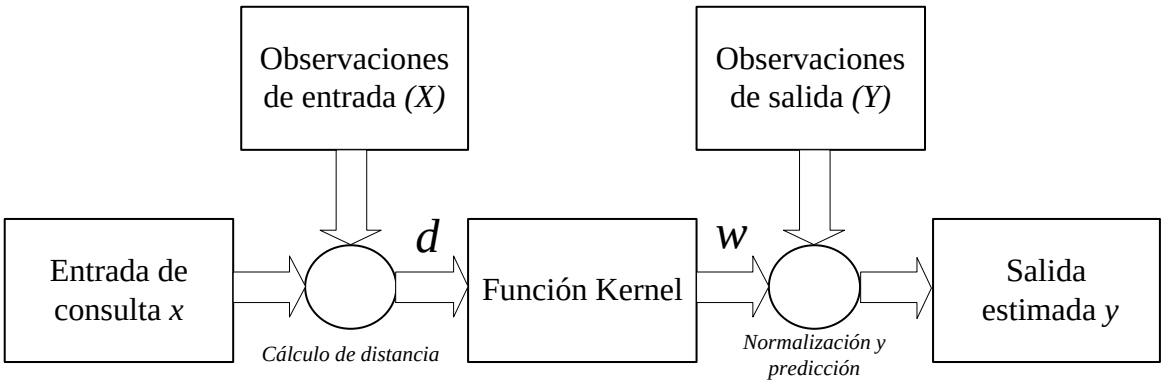

Figura 4. Diagrama en bloques de proceso de Regresión Kernel para predicción

Las entrada *X* y la salida *Y*, representan valores observados, estos valores se almacenan en arreglos matriciales usualmente denominados matriz memoria o matriz de entrenamiento. Las matrices son de dimensiones $n \times p$ y $n \times r$, respectivamente, donde n es la cantidad de datos observados, p la cantidad de señales involucradas en la entrada X y r la cantidad de señales involucradas en la salida Y. Por otro lado, los valores de consulta x estan representados por un vector de dimensión p igual a la cantidad de variables analizadas.

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} \cdots X_{1,p} \\ X_{2,1} & X_{2,2} \cdots X_{2,p} \\ \vdots & \vdots \ddots \vdots \\ X_{n,1} & X_{n,2} \cdots X_{n,p} \end{bmatrix} \qquad Y = \begin{bmatrix} Y_{1,1} & Y_{1,2} \cdots Y_{1,r} \\ Y_{2,1} & Y_{2,2} \cdots Y_{2,r} \\ \vdots & \vdots \ddots \vdots \\ Y_{n,1} & Y_{n,2} \cdots Y_{n,r} \end{bmatrix} \tag{2}$$

$$x = \begin{bmatrix} x_1 & x_2 \cdots x_p \end{bmatrix}$$

Para determinar la distancia entre la variable de consulta x y las observaciones de entrada X, se usa la distancia euclidiana o la distancia absoluta entre el i-ésimo valor observado representado Xi,j y el i-ésimo valor de consulta representado por xi.

$$d(X_i, x) = \sqrt{(X_{i,1} - x_i)^2 + (X_{i,2} - x_2)^2 + \cdots + (X_{i,p} - x_p)^2} \tag{3}$$

$$d(X_i, x) = X_i - x$$

En el cálculo de distancia, los valores de consulta que se encuentren por fuera de la región de entrenamiento, es decir, por debajo o por encima de los valores mínimos y máximos de la matriz de entrenamiento, tienen que ser excluidos del cálculo de distancia. Esto permite que el cálculo de distancia sea más robusto en referencia a corrimientos o anomalías (valores espurios-outliers) de las señales analizados. Si el valor de consulta *x* está fuera de la región de entrenamiento *X*, el cálculo de distancia dará resultados muy grandes, lo que posteriormente traerá aparejado factores de peso muy chicos que desencadenan en estimaciones poco confiables. El resultado final del cálculo de distancia entrega un vector columna *d* con *n* distancias.

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \tag{4}$$

Posteriormente, los valores de distancias son cuantificados por similiaridad, esto se logra por medio de la conversión de los valores de distancia en factores de peso o similaridades a través de la función Kernel.

Para el vector columna *d* de *n* distancias, el resultado de la función Kernel será en un vector columna *w* de *n* pesos, los cuales representan la similaridad entre el valor de consulta *x* y cada una de las entradas *X*. La ecuación que cuantifica la similaridad es:

$$w_i = K\left(d\left(X_i, x\right)\right) \tag{5}$$

En términos generales, la función Kernel entrega factores de peso de valor alto para pequeñas distancias y factores de peso de valor chico para grandes distancias. En otras palabras, cuando un valor de consultas *x* es cercano o idéntico al punto de referencia dado por *X*, la distancia será pequeña y el factor de peso será grande. La función Kernel usada es la función gaussiana.

$$K_h\left(d\right) = \frac{1}{\sqrt{2 \cdot \pi \cdot h^2}} e^{\left(\frac{-d^2}{2 \cdot h^2}\right)} \tag{6}$$

El parámetro *h* es el ancho de banda de la función Kernel, y es usado para controlar que las distancias sean consideradas similares. Cuando se focaliza en los valores medios, los valores de *h* son por lo general entre $0 < h < 1$.

En la figura 5 se muestran dos Kernel gaussianos con dos anchos de banda distintos. Para el caso de $h=0.1$ se producirán factores de peso altos cuando las distancias estén cercanas a cero, mientras que valores grandes de anchos de banda entregarán resultados menos específicos.

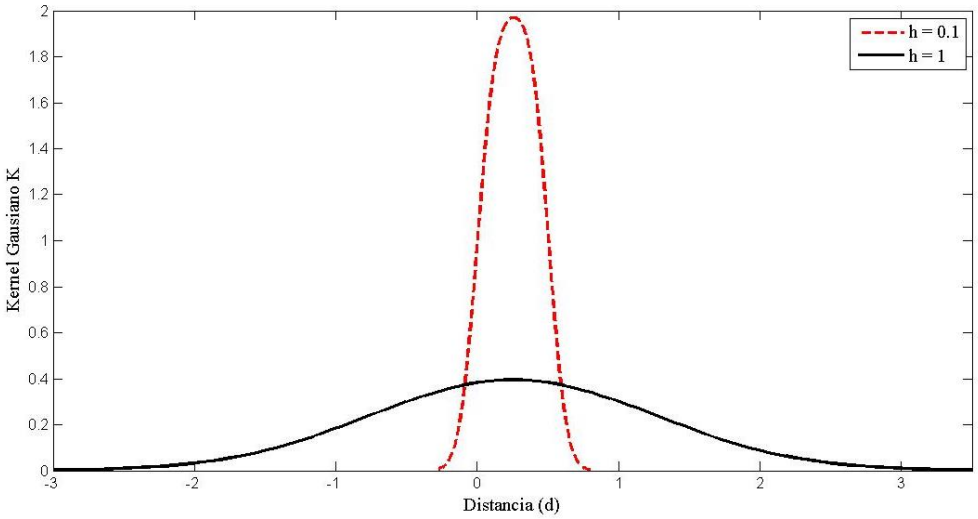

Figura 5. Ejemplos de Kernel gaussiano para distintos valores de ancho de banda *h*.

La optimización del valor de ancho de banda *h* es muy importante, ya que tiene gran implicancia en el desempeño de la predicción de valores. Una de las formas de optimizar el valor de *h* es por medio de minimizar el error cuadrático medio integrado de la función Kernel respecto a *h*, es decir, evaluada para distintos valores de *h*. Este análisis permite seleccionar el valor de *h* que minimiza la función Kernel.

Existen otras formas de optimizar el valor de *h* tales como minimizar el error cuadrático asintótico medio integrado, validación cruzada, reglas generalizadas por expertos y otros, pero debido a que el criterio de minimizar el error cuadrático medio integrado entrega resultados aceptables, hacemos uso de éste.

La cuantificación de cada uno de los valores de distancia, como se mencionó anteriormente, se calcula mediante la expresión:

$$w_i = K_h(d_i) = \frac{1}{\sqrt{2 \cdot \pi \cdot h^2}} e^{\left(\frac{-d_i^2}{2 \cdot h^2}\right)} \qquad (7)$$

El resultado de esta operación entrega un vector columna *w* con *n* factores de peso (similaridades) calculadas.

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \qquad (8)$$

En la Regresión Kernel los factores de peso son combinados con la salida *Y*, la cual es normalizada por la suma de los factores de peso, esta suma entrega un escalar *a*.

$$a = \sum_{i=1}^{n} w_i \qquad (9)$$

Según la definición de la ecuación (5) la estimación de la salida se puede escribir como:

$$\hat{y}(x) = \frac{1}{a} \sum_{i=1}^{n} (w_i Y_i) \qquad (10)$$

A partir de este punto, se pueden plantear las distintas formas del tercer paso del proceso de regresiones de Kernel, para la predicción de valores, estas son: Regresión Kernel Heteroasociativa (HKR) y Regresión Kernel Autoregresiva (AAKR).

3.1.1. REGRESIÓN KERNEL HETEROASOCIATIVA

La característica de la Regresión Kernel Heteroasociativa es que usa *p* entradas para estimar *r* salidas. Los primeros dos paso del proceso de Regresión Kernel miden la distancia y los factores de peso o similaridades de la consulta, teniendo en cuenta la matriz de entrenamiento de entrada *X*, el tercer paso usa las similaridades para sopesar la salida por medio del uso de la matriz *Y*. La ecuación que describe esto es:

$$\hat{y}(x) = \frac{1}{a}\left[\sum_{i=1}^{n}(w_i Y_{i,1}) \;\; \sum_{i=1}^{n}(w_i Y_{i,2}) \;\; \cdots \sum_{i=1}^{n}(w_i Y_{i,r})\right] \qquad (11)$$

Para le vector *w* de *n* valores, la predicción es calculada para las *r* salidas donde *Yi,j* es el i-esimo valor de la salida j-esima.

3.1.2. REGRESIÓN KERNEL AUTOREGRESIVA

En este caso no hay matriz de salida de valores observados *Y*, en cuyo caso las entradas *X* realizan la misma tarea que lo harían las salidas *Y*. Los valores de salida estimados de este modelo son la versión corregida de las entradas sopesadas y normalizadas por los factores de peso *w*, la ecuación que lo describe es:

$$\hat{x}(x) = \frac{1}{a}\left[\sum_{i=1}^{n}(w_i X_{i,1}) \;\; \sum_{i=1}^{n}(w_i X_{i,2}) \;\; \cdots \sum_{i=1}^{n}(w_i X_{i,p})\right] \qquad (12)$$

Una forma simplificada es la ecuación 13, donde los factores de peso son combinados con la matriz de entrenamiento de entrada *X* y normalizada por el escalar *a:*

$$\hat{x}(x) = \frac{\sum_{i=1}^{n}(w_i X_i)}{a} \qquad (13)$$

El diagrama de proceso modificado para una Regresión Kernel Autoregresiva se muestra en la figura 6.

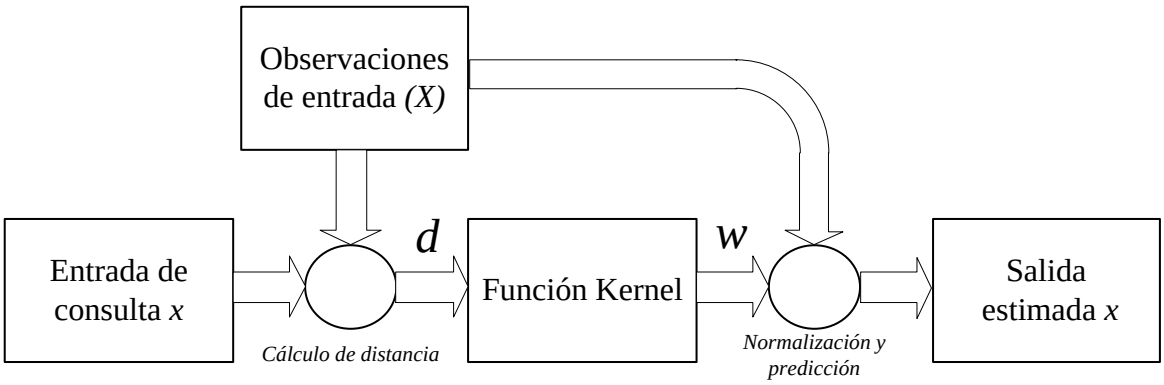

Figura. 6 Diagrama en bloques de proceso de Regresión Kernel Autoregresivo

3.2. TÉCNICAS PARAMÉTRICAS APLICADAS

En la siguiente sección se describen las características principales de la técnica paramétrica aplicada, que en este trabajo se refiere al uso de redes neuronales.

3.2.1. REDES NEURONALES

Las redes neuronales conforman un área de gran importancia en el campo de la inteligencia artificial. Originalmente los estudios se centraron en entender como funcionaba el cerebro y simular alguna de sus propiedades.

El desarrollo de sistemas basados en redes neuronales se debe al hecho de que si bien el cerebro es más torpe para efectuar cálculos matemáticos que un computador, es mucho más eficiente desde el punto de vista del procesamiento cuando lleva a cabo tareas como el reconocimiento de una imagen, la identificación de un tipo de patrón, etc. A este tipo de tareas el cerebro, por medio de sus neuronas, las realiza a velocidades muy superiores que el de los computadores. El cerebro logra esta eficiencia debido a un elevado paralelismo en el procesamiento de la información, y al gran número de neuronas que posee. El cerebro humano posee aproximadamente 10^{11} neuronas y cada una de estas está conectada con aproximadamente otras 10^{4}.

Los sistemas basados en redes neuronales se basan, como se mencionó más arriba, en estudios biológicos y fisiológicos del cerebro. Si bien los modelos usados en ingeniería distan bastantes de los reales (las redes neuronales se implementan con elementos electrónicos o software, etc.) presentan buenos resultados prácticos.

A continuación, se describen las herramientas básicas y necesarias sobre redes neuronales utilizadas en este trabajo.

3.2.2. DEFINICIÓN DE REDES NEURONALES

Teniendo en cuenta lo expuesto anteriormente, establecemos que: una red neuronal es un sistema constituido por un conjunto de elementos idénticos, que se denominan elementos de procesamiento, neuronas o nodos, que están masivamente interconectados entre sí y tienen la potencialidad de mejorar su desempeño mediante un aprendizaje dinámico ante la presentación de una señal de entrada patrón y una respuesta deseada.

3.2.2.1. APRENDIZAJE DE UNA RED NEURONAL

La característica más importante de las redes neuronales es la capacidad de aprender del medio en el cual la red se desempeña, y mejorar dicho desempeño a través del aprendizaje. Este, es un proceso durante el cual se presenta un conjunto de datos (que dependen del medio) a la red neuronal, la red aprende mediante un proceso repetitivo e iterativo de ajustes, cambiándose el valor de los pesos sinápticos y de las bias para mejorar su desempeño, este proceso dura una cantidad determinada de iteraciones y puede ser supervisado o no-supervisado.

3.2.2.2. APRENDIZAJE SUPERVISADO

Aprendizaje en el cual un sistema se entrena mediante un modelo de referencia (o maestro) que le muestra las respuestas deseadas (o correctas) ante un estímulo de entrada definido, es decir, que la red es entrenada para obtener en la salida un valor previamente conocido.

3.2.2.3. APRENDIZAJE NO-SUPERVISADO

Aprendizaje en el cual no es necesario mostrar o disponer de las respuestas correctas para estímulos de entrada definidos, el sistema debe auto-organizarse exclusivamente en base a los estímulos recibidos de la entrada. En otras palabras, este tipo de aprendizaje intenta obtener propiedades o características comunes de los datos presentados a la entrada del sistema.

3.2.2.4. MODELO BÁSICO DE UNA NEURONA

Una neurona es una unidad de procesamiento de información la cual es fundamental para la operación de una red neuronal. En la figura 7 se muestra el modelo de una neurona. A continuación se identifican cuatro elementos básicos del modelo:

1. Un conjunto de sinapsis o intensidad de conexión, cada una está caracterizada por un peso. Específicamente, una señal x_j en la entrada de una sinapsis j conectada a la neurona k es multiplicada por el peso o sinapsis w_{kj}. Este peso es el que se ajusta en el proceso de entrenamiento de una red. El primer subíndice se refiere a la neurona en cuestión y el segundo subíndice se refiere a la entrada de la sinapsis y por ende a la sinapsis misma. Distinto a lo que ocurre con la sinapsis en el cerebro, el peso en una neurona artificial puede estar tanto en un rango positivo como negativo.
2. Un sumador para sumar las señales de entrada pesadas por las respectivas sinapsis de la neurona. La operación que se describe aquí constituye una combinación lineal.
3. Una función de activación para limitar la amplitud de la salida de la neurona. Los niveles límites de la función de activación se encuentran en el intervalo [0,1] o alternativamente [-1,1].
4. Una señal bias, denotada por b_k. La señal bias es un valor constante (generalmente 1) que tiene un efecto de tendencias, es decir, de incrementar o disminuir la entrada de la red de la función de activación, dependiendo si esta es positiva o negativa. La señal bias es opcional, no siempre está en el modelo de una neurona.

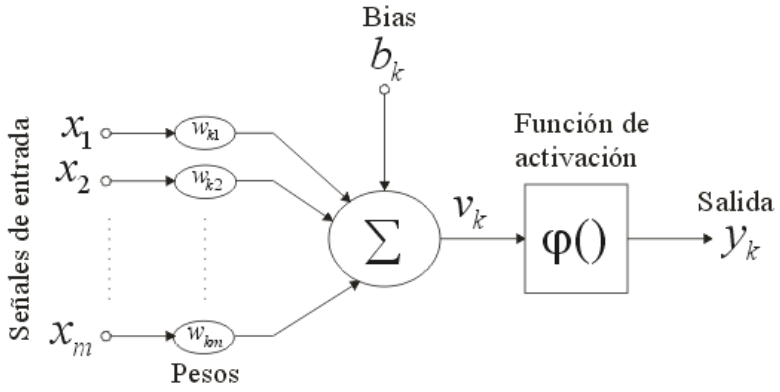

Figura 7: Modelo de una neurona *k*-ésima

Para una neurona *k*, los pesos o sinapsis se pueden representar por medio de un escalar w_k, esto se ve claramente en la figura 7.

3.2.2.5. TIPOS DE FUNCIÓN DE ACTIVACIÓN

La función de activación indicada por $\varphi(v)$, define la salida de la neurona. A continuación se muestran tres tipos básico de función de activación:

1. *Función umbral*. Este tipo de función de activación, se describe en la siguiente expresión y tiene la forma:

$$\varphi(v) = \begin{Bmatrix} 1 & si & v \geq 0 \\ 0 & si & v < 0 \end{Bmatrix} \qquad (14)$$

Correspondientemente, la salida de la neurona *k* es expresada como:

$$y_k = \begin{Bmatrix} 1 & si & v_k \geq 0 \\ 0 & si & v_k < 0 \end{Bmatrix} \qquad (15)$$

La siguiente figura muestra este tipo de función de activación:

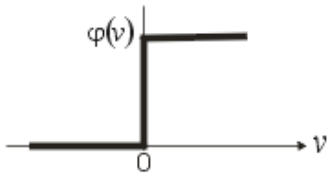

Figura 8: Función de activación umbral.

En este modelo la salida la neurona toma el valor 1 si v_k es no negativa y cero en otro caso.

2. *Función lineal bipolar*. Esta función se describe a continuación:

$$\varphi(v)=\begin{cases}1 & v\geq\frac{1}{2}\\ v & \frac{1}{2}>v>-\frac{1}{2}\\ 0 & v\leq-\frac{1}{2}\end{cases} \qquad (16)$$

Donde el factor de amplificación dentro de la región lineal de operación es asumido unitario. De esta manera la función de activación es vista como una aproximación a un amplificador no-lineal. La siguiente figura que se muestra a continuación. muestra esta función de activación.

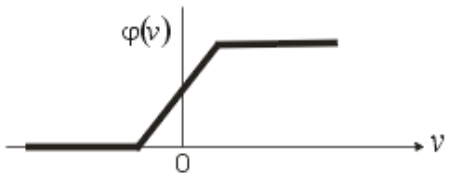

Figura 9: Función de activación bipolar.

3. *Función sigmoidea*. Esta es una de las funciones más usadas en redes neuronales. Se define como una función estrictamente creciente que exhibe un balance entre una conducta lineal y no-lineal. Existen diferentes tipos de funciones sigmoideas, un ejemplo de ésta es la función logística, definida por:

$$\varphi(v)=\frac{1}{1+e^{(-av)}} \qquad (17)$$

Donde *a* es el parámetro de inclinación o pendiente. Variando este parámetro, la función adquiere diferentes inclinaciones. La siguiente figura muestra esta función de activación:

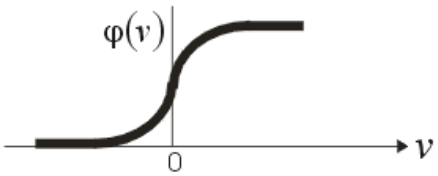

Figura 10: Función de activación sigmoidea.

3.2.2.6. TOPOLOGÍA DE UNA RED

Independientemente del tipo de elemento de procesamiento, es necesario definir una estructura o topología de la red neuronal, es decir, definir el conexionado entre las neuronas, lo que define el modelo de conexiones. Este modelo se organiza en niveles y al conjunto de neuronas que se encuentren en un mismo nivel se las denomina capa (layer). Estas capas se interconectan mediante los pesos o sinapsis. La figura 11 muestra un conexionado de tres capas. El sentido de las flechas indica el sentido de propagación de las señales.

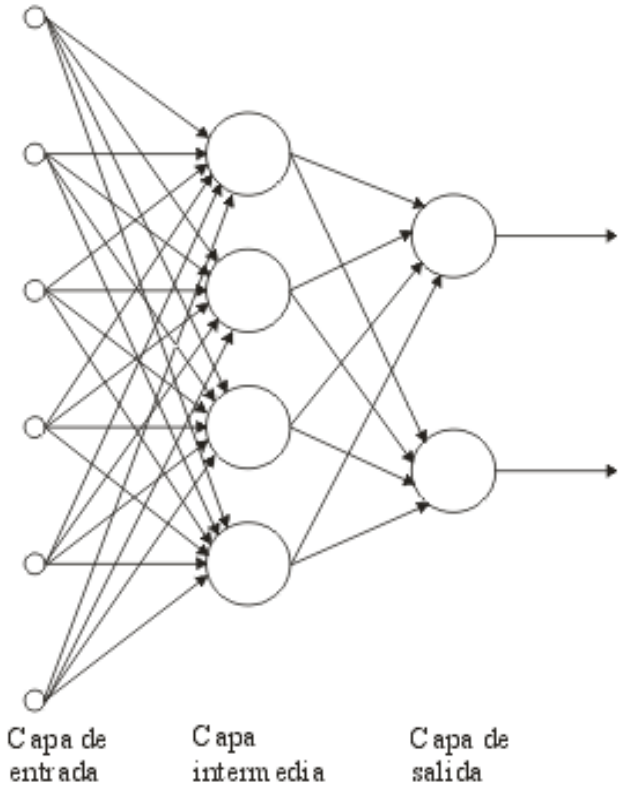

Figura 11: Red feedforward de tres capas

3.2.2.7. PROPAGACIÓN DE LAS SEÑALES

En general existen dos formas de propagar las señales en una red neuronal: propagación de las señales hacia delante (feedforward signals) y realimentación de señales (feedback signals). El sistema utilizado en este trabajo hace uso de propagación mediante realimentación.

3.2.3. REDES NUEURONALES AUTOREGRESIVAS

Las red neuronal utilizada en este trabajo es del tipo autoregrsiva, ARNN (Auto Regressive Neural Network). Este tipo de red utiliza datos históricos como target o referencia para la estimación de los valores. El patrón de interconexión esta definido por la siguiente ecuación:

$$\hat{y} = f(y(t-1) + y(t-2) + \ldots + y(t-d)) \qquad (18)$$

Donde *d* es el tiempo de retardo, *f()* es la función no lineal que solo depende de los últimos *d* valores previos.

La red neuronal consta de dos capas, una capa interna y otra de salida. El número de capas internas es 10 y el valor de *d* es 2. La función de activación calcula las salidas y se corresponde con la siguiente ecuación:

$$\hat{y} = f(b + \sum_i w_i x_i) \qquad (19)$$

Donde *bi* son los bias, *wi* los pesos y *xi* las entradas. La figura 12 muestra la estructura de la red ARNN implementada.

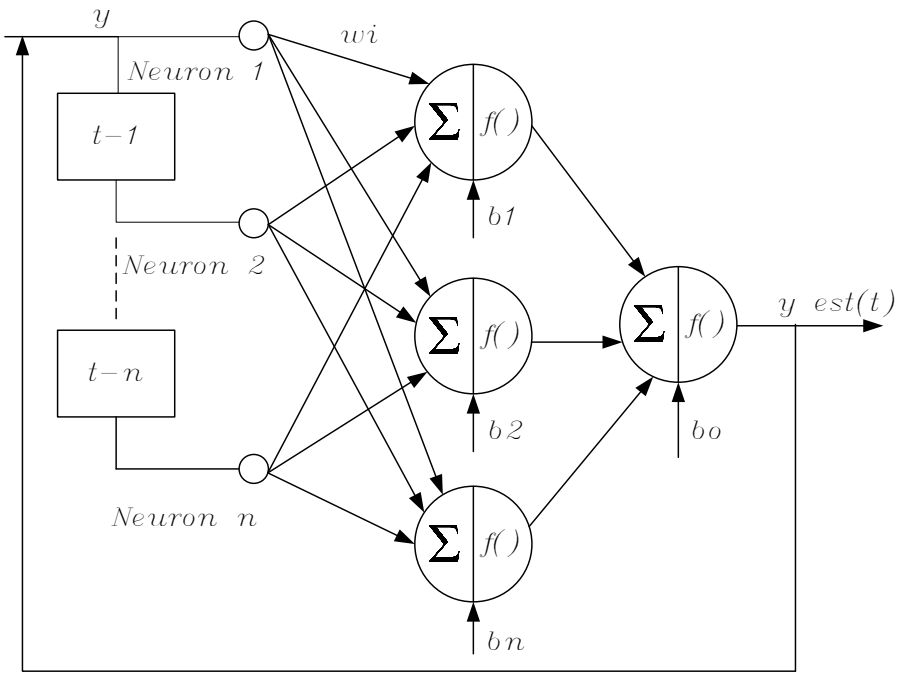

Figura 12: Estructura ARNN implementada.

La red neuronal está basada en un proceso de entrenamiento, éste es un proceso cíclico donde los pesos *w* y los bias *b* son modificados mediante el seguimiento de una regla. La regla de aprendizaje busca minimizar el error entre la salida y la referencia, una vez que el error se ha alcanzado por medio de un proceso de minimización, el entrenamiento de la red finaliza. El proceso de entrenamiento es de lazo abierto y el proceso de estimación posterior al entrenamiento es un proceso de lazo cerrado.

No es objeto de este trabajo ahondar en características técnicas de redes neuronales, por lo que los detalles pueden ser estudiados en la bibliografía especializada.

3.3. MODULO DE COMPARACION

Este bloque del sistema solamente se refiere a la generación de los residuos, que son la diferencia entre los valores medidos y los valores estimados. Estos residuos son los que ingresan a la lógica de decisión quien se encarga de informar alarma (estado de inconsistencia) o funcionamiento normal (condición de no información).

3.4. LOGICA DE DECISION

El cálculo en la lógica de decisión se realiza por medio de una prueba secuencial de razón de probabilidades, SPRT (Sequential Probability Ratio Test). La detección de anomalía sobre el sensor o instrumento se realiza a partir de los residuos calculados en la etapa anterior, y el problema que esta lógica aborda, es cómo diferenciar si los residuos se refieren a cambios en el proceso, anomalías en el instrumento o condiciones normales de operación con variaciones propias de medición.

SPRT es una técnica estadística que busca detectar la anomalía tan pronto como sea posible con una muy baja probabilidad de tomar la decisión incorrecta. El proceso consiste en evaluar si el instrumento acepta la hipótesis H_0 que refiere a modo normal de operación o acepta la hipótesis H_1 que se refiere a modo degradado o anomalía. La técnica SPRT realiza esto por medio del cálculo de la razón de verosimilitud, cuya ecuación se muestra a continuación:

$$L_n = \frac{probabilidad\ de\ observaciones\ \{r_n\}\ dado\ H_1\ sea\ verdadero}{probabilidad\ de\ observaciones\ \{r_n\}\ dado\ H_0\ sea\ verdadero} = \frac{p\left(\{r_n\}/H_1\right)}{p\left(\{r_n\}/H_0\right)} \qquad (20)$$

Donde r_n es la secuencia consecutiva de n residuos r, y L_n es la razón de verosimilitud que posteriormente es comparada con dos limites de aceptación, el limite inferir $ln(A)$ y el limite superior $ln(B)$. Las ecuaciones que definen esos límites son las siguientes:

$$\ln(A) = \ln\left(\frac{\beta}{1-\alpha}\right)$$

$$\ln(B) = \ln\left(\frac{1-\beta}{\alpha}\right) \qquad (21)$$

El parámetro α se corresponde con la probabilidad de falsa alarma y el parámetro β se refiere a la probabilidad de alarma perdida.

Si la razón de verosimilitud es menor que *ln(A)* se acepta la hipótesis H_0 que implica condición normal y si la razón de verosimilitud es mayor que *ln(B)* se acepta la hipótesis H_1, es decir, modo degradado o anomalía.

Para una secuencia de residuos que aceptan la hipótesis H_0 la media de los residuos es cero y para una secuencia de residuos que aceptan la hipótesis H_1 la media de los residuos es distinta de cero (positiva o negativa). La distribución que se adopta para el cálculo de razón verosimilitud es:

$$p\left(r_i \middle/ H_0\right) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma^2}} e^{\left(\frac{r_i^2}{2 \cdot \sigma^2}\right)} \tag{22}$$

Sin ser objeto del presente trabajo, para el desarrollo matemático que deduce el cálculo de la razón verosimilitud, solo indicamos que la varianza se calcula como la varianza de los residuos en condición normal y las ecuaciones se deducen del cálculo de la razón de verosimilitud a partir de la ecuación 22.

La hipótesis H_0 igual a media de residuos cero y H_1 igual a cualquier valor distinto de cero, son las siguientes:

$$H_0 = \mu_0 = 0 \qquad H_1 \neq \mu_{neg,pos} \tag{23}$$

$$SPRT_{pos} = \frac{\mu_{pos}}{\sigma^2}\left(r_i - \frac{\mu_{pos}}{2}\right) \tag{24}$$

$$SPRT_{neg} = \frac{\mu_{neg}}{\sigma^2}\left(-r_i - \frac{\mu_{neg}}{2}\right) \tag{25}$$

La magnitud de cambio sobre el instrumento causada por alguna anomalía o falla puede ser confiablemente detectada por el índice *SPRT*. Si el índice *SPRT* es mayor que el límite superior *ln(B)* entonces se puede concluir que la hipótesis H_1 es verdadera, lo que implica condición de anomalía, si el indicie *SPRT* es menor que el *ln(A)* entonces se puede

concluir que la hipótesis H_0 es verdadera, lo que implica condición normal, si el índice *SPRT* esta entre *ln(A)* y *ln(B)* no hay suficiente información para tomar la decisión.

El valor óptimo de μ_{pos} y/o μ_{neg} son determinados numéricamente calculando el índice *SPRT* en condición normal sin falla, se evalúan los datos y se localiza el valor de μ_{pos} y/o μ_{neg} que resulte en una probabilidad de falsa alarma que sea lo más cercana posible a la probabilidad de falsa alarma α previamente adoptada.

4. RESULTADOS REGRESIÓN KERNEL AUTOREGRESIVA

En la siguiente sección se realiza la evaluación de la técnica de Regresión Kernel Autoregresiva. El proceso de análisis se efectúa mediante estimación de tres variables físicas típicas, dos variables de temperatura y una variable de presión. Estas variables son creadas a partir de la dinámica de variables reales del mismo tipo.

En lo que corresponde a centrales nucleares, tanto variables de temperaturas como presión, son algunas de las variables que necesitan ser medidas para la correcta operación.

La figura 13 muestra un esquema de central tipo PHWR (Pressurized heavy water reactor) y en este tipo de centrales las variables de temperatura pueden ser temperaturas de rama fría y caliente del sistema refrigerante, o bien temperaturas correspondientes al sistema moderador, por otro lado la variable de presión puede ser la presión del sistema.

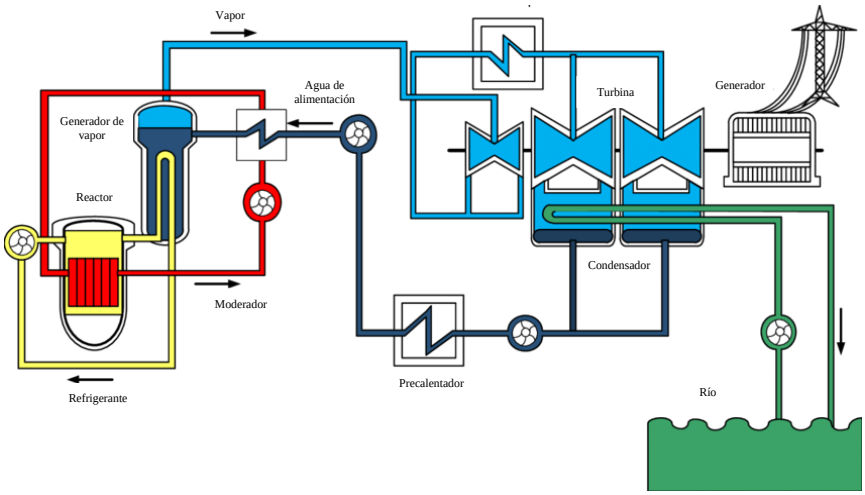

Figura 13. Esquema de central PHWR.

Las siguientes características de datos se corresponden con las variables de temperatura y presión adoptadas para el análisis.

Variable	Rango	Span	Tolerancia
Temperatura 1	0 – 280 ºC	280 ºC	1 % Span
Temperatura 2	0 – 280 ºC	280 ºC	1 % Span
Presión	60 – 110 Bar	50 bar	2 % Span

Tabla 1. Características de variables.

Las señales usadas se correspoden a un periodo de estimación de 30000 minutos, lo que concierne aroximadamente a 21 días.

Las siguientes figuras muestras las señales a ser usadas.

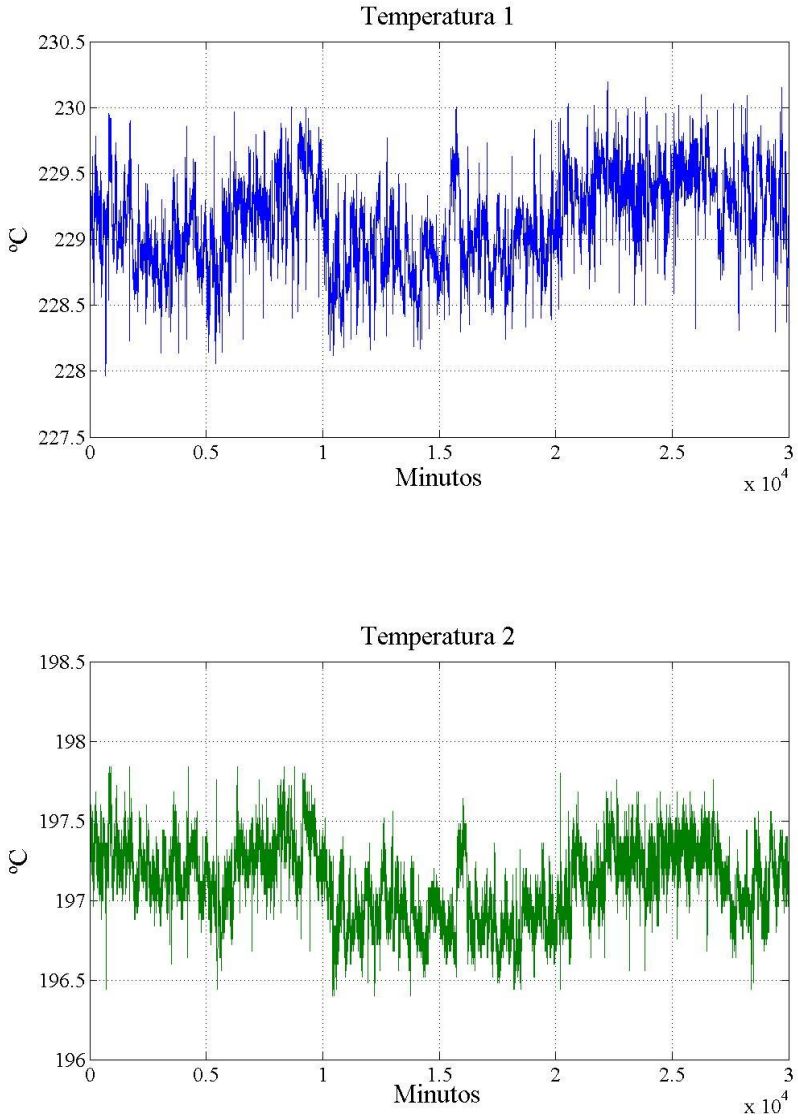

Figura 14. Temperatura 1 y Temperatura 2.

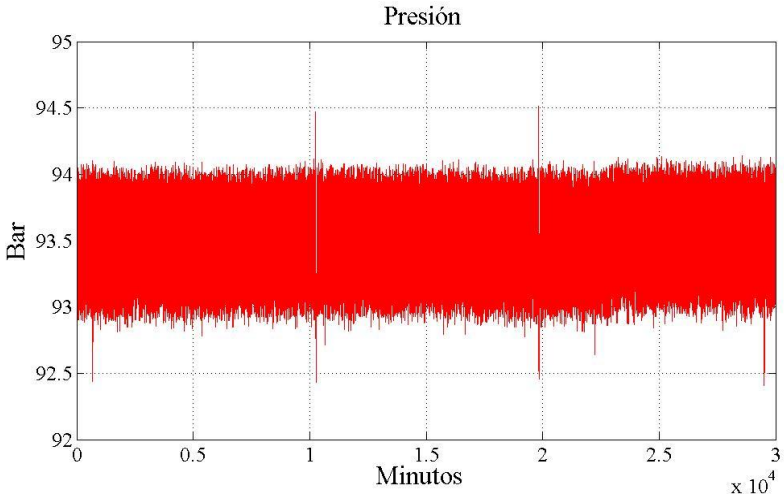

Figura 15. Presión.

4.1. CONSTRUCCION DE MATRIZ DE ENTRENAMIENTO

El proceso de construcción de la matriz de entrenamiento es el más importante de todo el modelo, ya que ésta contendrá los valores más representativos del proceso en análisis. La selección de los datos más representativos se refiere a que estos datos posean las características más distintivas en cuanto a su histórico de funcionamiento, cambios dinámicos y valores de diseño.

La selección de datos para la construcción de la matriz de entrenamiento se realiza mediante dos técnicas combinadas de selección, técnica de selección por máximos y mínimos y técnica de selección por ordenamiento de vector. Estas técnicas y su descripción pueden ser estudiadas en la bibliografía presente al final de este trabajo.

4.2. SELECCIÓN DE ANCHO DE BANDA

La selección del ancho de banda *h* es otro parámetro importante que influye en el modo de comportamiento del modelo. Para obtener el óptimo valor de *h*, se realizan *N* estimaciones con *N* valores de *h* que arrancan con un valor muy chico, por ejemplo 0.0001. Para cada cálculo de estimación se evalúa el error entre el valor estimado y el valor real. La selección del óptimo valor de *h* será el que se corresponde con el mínimo error calculado. La figura 16 muestra la relación entre el error y el ancho de banda. El valor óptimo *h*=1.81 es donde la función se minimiza.

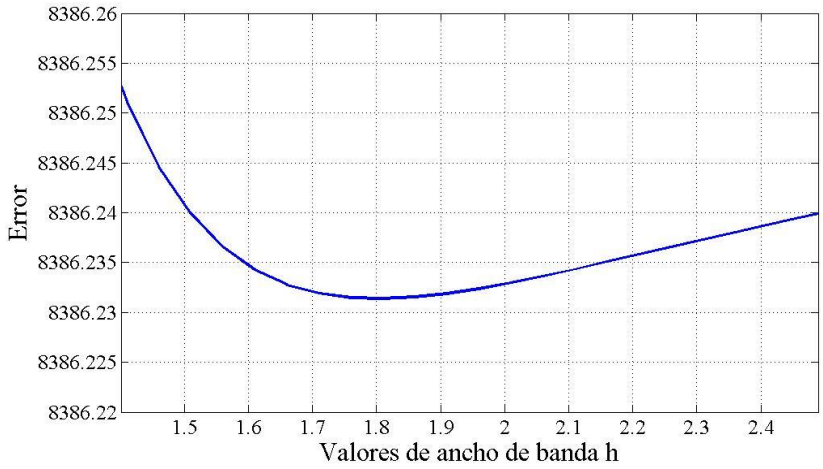

Figura 16. Error vs valores de ancho de banda evaluados

4.3. ANALSIS DE α, β Y μ

La correcta selección de los parámetros α, β y μ se realiza sobre el cálculo de estimaciones en condiciones previamente conocidas, una de ellas es sin falla, es decir, operación normal y la otra en condición de falla, es decir, drift sobre la señal.

Analizando la condición de operación normal, se evalúan los datos y se localiza el valor de μ que resulte en una fracción de falsa alarma que sea menor o igual a la probabilidad de falsa alarma α. La fracción de falsa alarma se calcula como la relación entre la cantidad de aceptaciones de la hipótesis H_1 y la cantidad de aceptaciones de la hipótesis H_0 (cantidad aceptaciones H_1 / cantidad aceptaciones H_0), los resultados de las fracciones de falsa alarma son las que se muestran en la tabla 2.

Variable en condición normal	Fracción de falsa alarma	α	μ	Comentarios
Temperatura 1	0,00408	0,05	2,8	0,00408 < 0,05
Temperatura 2	0,00077	0,05	2,5	0,00077 < 0,05
Presión	0,000515	0,004	1	0,000515 < 0,004

Tabla 2. Resultados de análisis de parámetros α y μ

Las fracciones de alarmas en los tres casos, es menor que sus respectivos valores de α. Esto implica la aceptación de los valores α adoptados y la aceptación de los valores μ utilizados en cada caso. Es importante destacar que los valores de μ adoptados en el inicio del análisis se corresponden con los porcentajes de tolerancia referidos al Span de cada caso.

Para la adopción el parámetro β, se analizan los datos a partir de una condición de falla previamente conocida, se calcula la fracción de alarma perdida como la relación entre la cantidad de aceptaciones de la hipótesis H_0 y la cantidad de aceptaciones de la hipótesis H_1 (cantidad aceptaciones H_0 / cantidad aceptaciones H_1) y se la compara con el valor de β. Si los valores calculados de fracción de alarma perdida son menores o iguales a los valores de β adoptados, se consideran valores óptimos. La tabla 3 muestra los resultados obtenidos para cada caso.

Variable en condición de falla	Fracción de alarma perdida	β	μ	Comentarios
Temperatura 1	0,00403	0,05	2,8	0,00403 < 0,05
Temperatura 2	0,00349	0,05	2,5	0,00349 < 0,05
Presión	0,04749	0,05	1	0,04749 < 0,05

Tabla 3. Resultados de análisis de parámetros β

Las fracciones de alarmas en los tres casos es menor que sus respectivos valores de β. En las figuras 17 y 18, correspondientes a la temperatura 1, se muestran las condiciones de operación y estimaciones necesarias para el análisis de los parámetros α, β y μ.

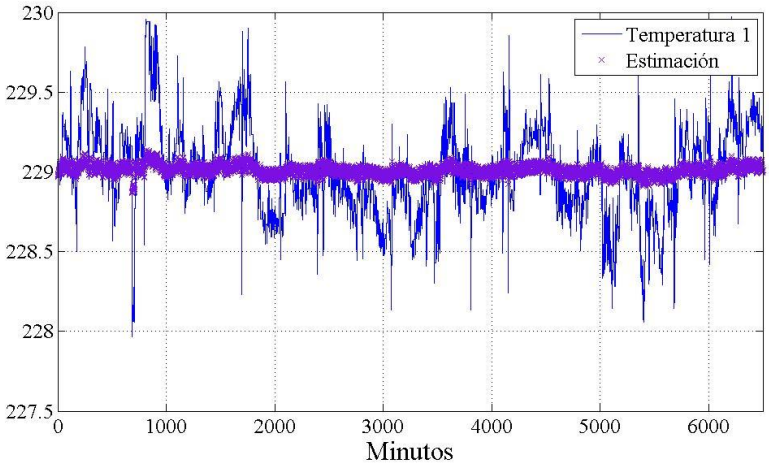

Figura 17. Temperatura 1 y estimación en condición normal.

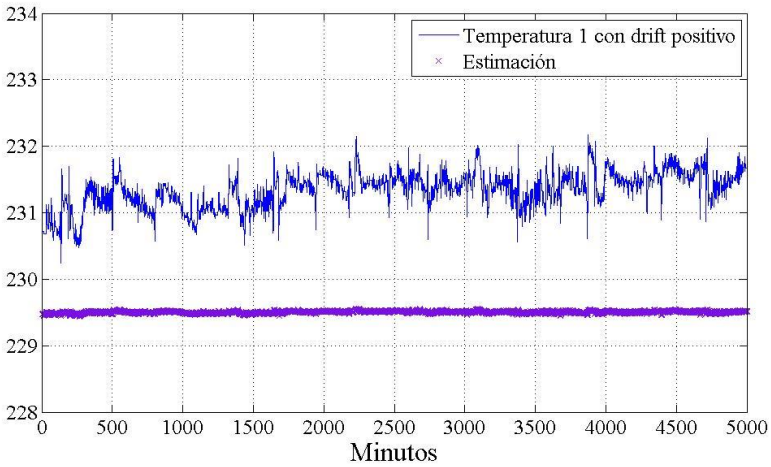

Figura 18. Temperatura 1 y estimación en condición de falla (drift positivo).

4.4. ESTIMACIÓN DE VARIABLES

En la siguiente sección se muestran los distintos casos analizados y sus respectivos desempeños en cuanto a la estimación y generación de alarmas mediante resultados de Regresión Kernel Autoregresiva. Se plantea en este análisis una condición de alarma con categoría de *'importante a revisar'* a aquella condición que posee un disparo continuo de alarma por más de 10 minutos seguidos, esta condición podría ser establecida para cada caso según la naturaleza del proceso.

4.4.1. CASO SIN FALLA

A continuación se muestran las respectivas estimaciones y alarma para el caso de señales sin falla.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	0	0
Temperatura 2	0	0
Presión	2	0

Tabla 4. Caso sin falla

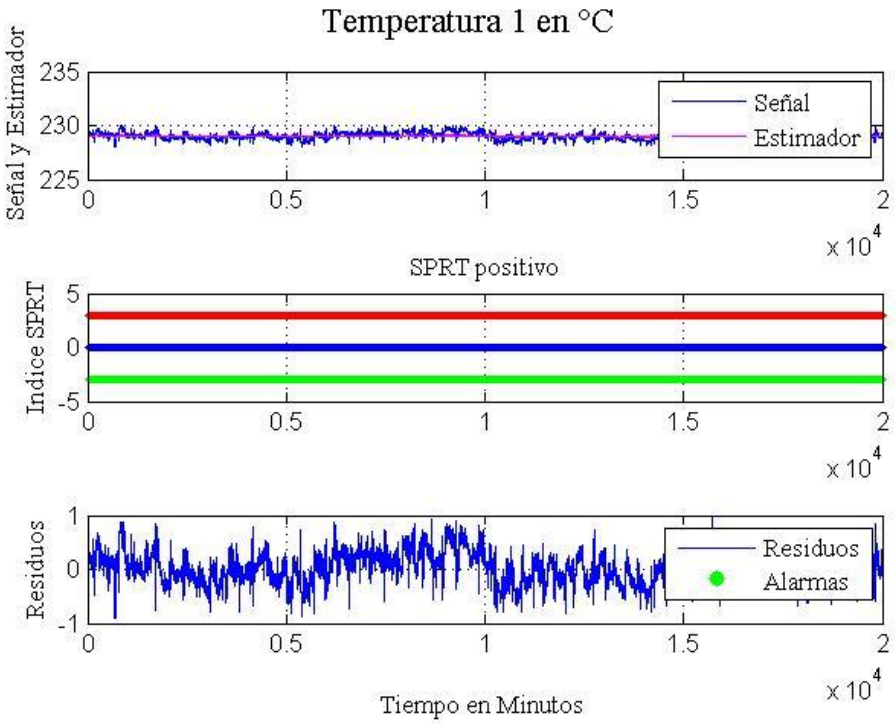

Figura 19. Resultados de Temperatura 1 sin falla.

En este caso se observa claramente que las tres estimaciones siguen a la señal, (figuras 19, 20 y 21), el índice SPRT para las dos temperaturas no supera el límite superior, para el caso de la presión se generan 2 alarmas consideradas falsas alarmas, los residuos en todos los casos están alrededor de cero. En este caso no hay generación de alarma continua que implique categoría de *'importante a revisar'*.

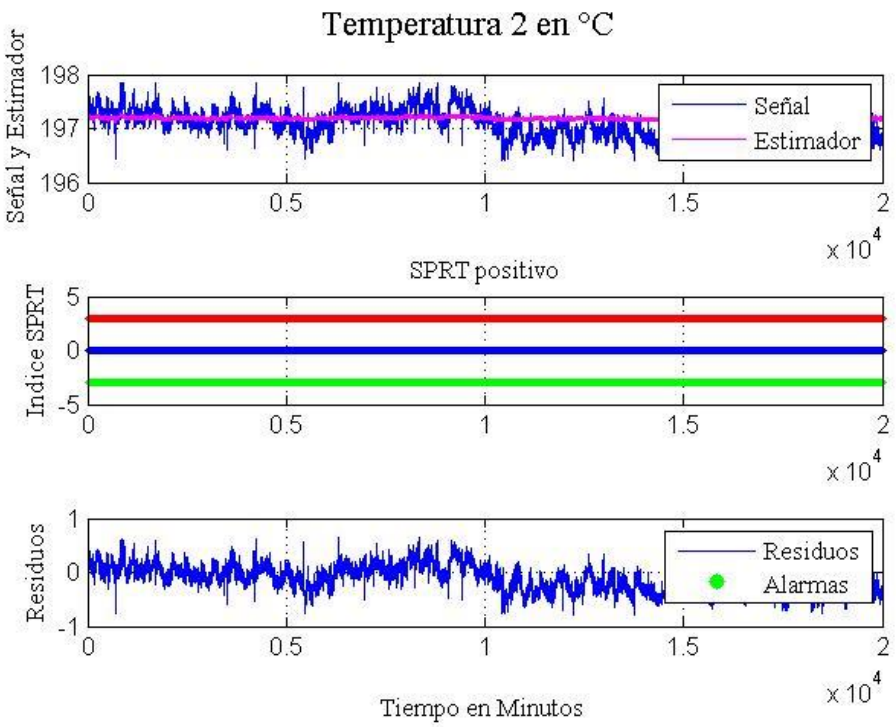

Figura 20. Resultados de Temperatura 2 sin falla.

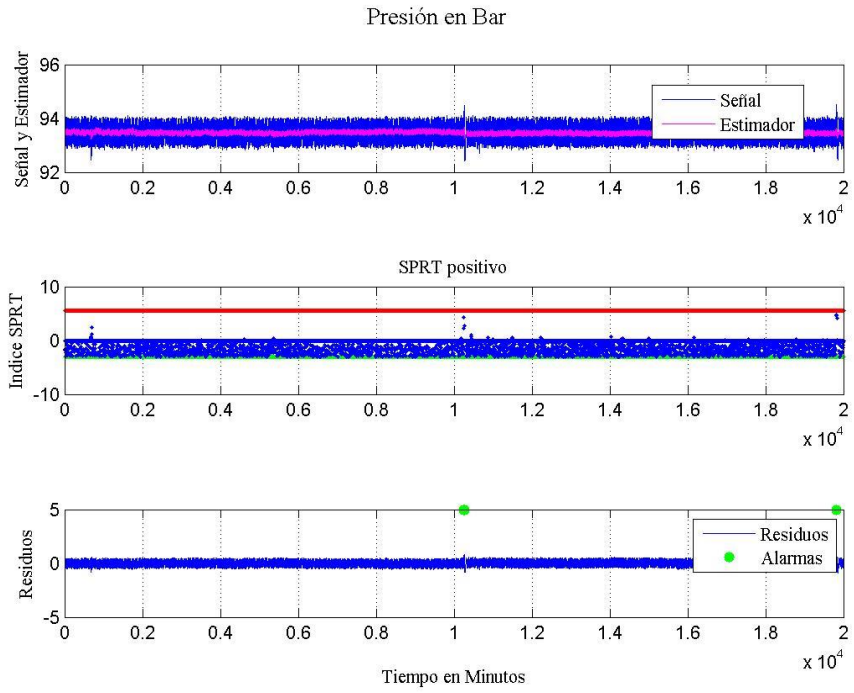

Figura 21. Resultados de Presión sin falla.

4.4.2. DRIFT EN TEMPERATURA 1

Estimaciones y alarma para el caso de una señal con drift.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	2276	1390
Temperatura 2	0	0
Presión	2	0

Tabla 5. Caso de una señal con drift

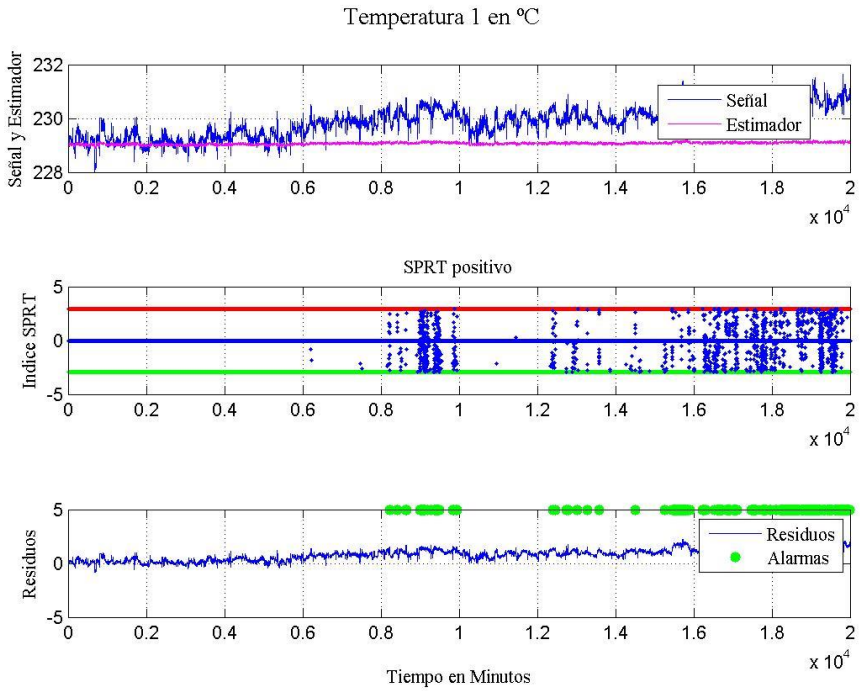

Figura 22. Resultados de temperatura 1 con temperatura 1 drift positivo.

En la figura 22 se manifiesta que la estimación de la temperatura 1 sigue el valor esperado, el índice SPRT en distintos casos sobrepasa el límite superior, los residuos se desvían y se generan alarmas (puntos verdes en la figura), la categoría de *'importante a revisar'* se manifiesta con la secuencia continua de puntos al final en la gráfica de residuos. Las figuras 23 y 24 muestran condiciones sin falla.

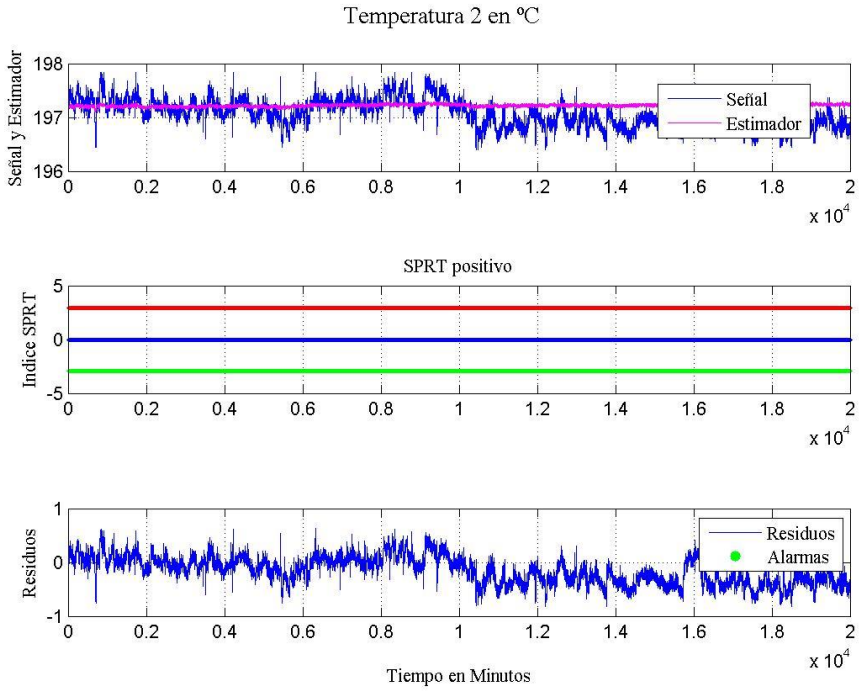

Figura 23. Resultados de temperatura 2 con temperatura 1 drift positivo.

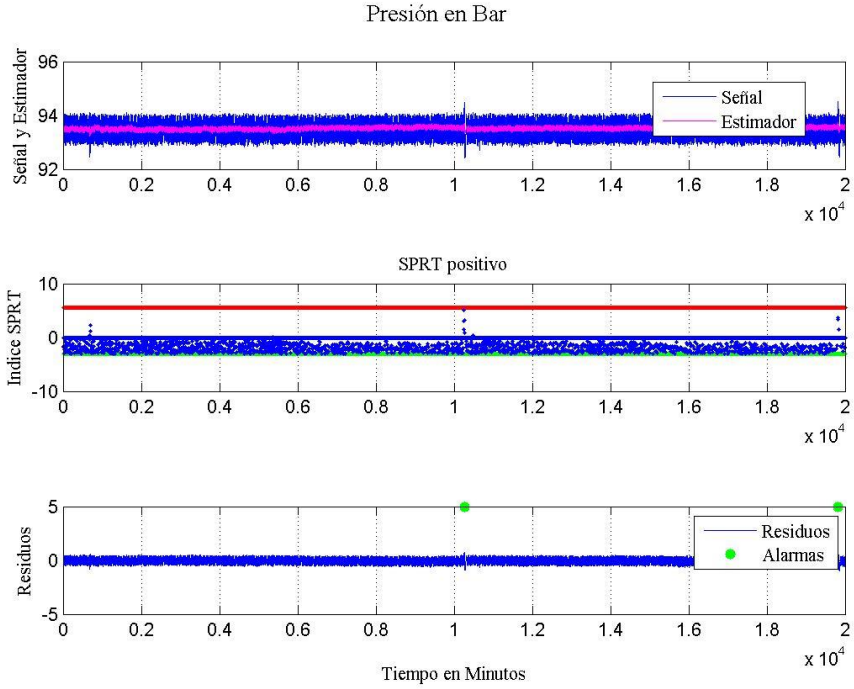

Figura 24. Resultados de presión con temperatura 1 drift positivo.

4.4.3. DRIFT EN TEMPERATURA 1 Y 2

Estimaciones y alarma para el caso de dos señales con drift.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	1871	1040
Temperatura 2	981	350
Presión	2	0

Tabla 6. Caso de dos señal con drift

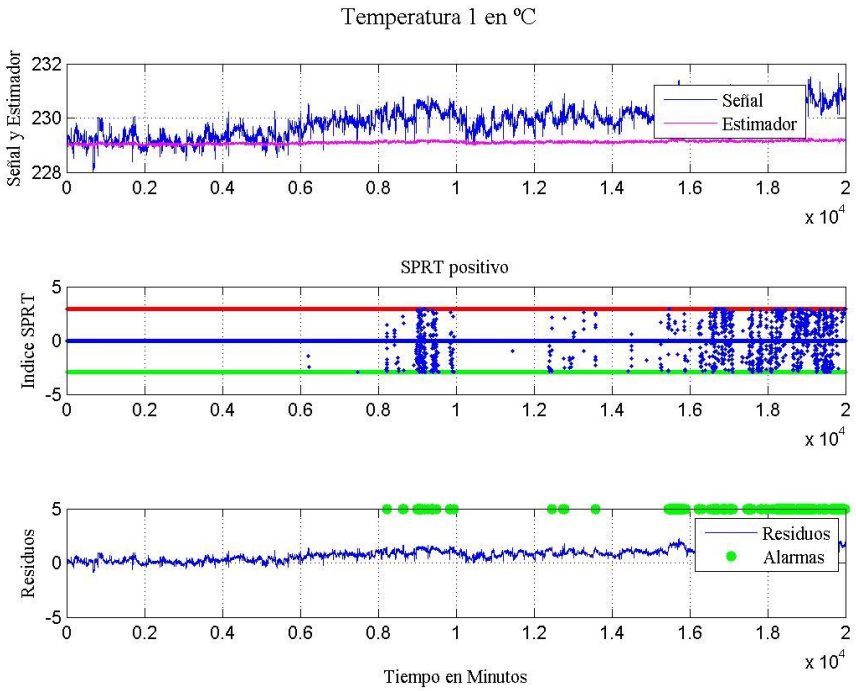

Figura 25. Resultados de temperatura 1 con temperatura 1 y 2 drift positivo.

En las figuras 25 y 26 se observa que las estimaciones siguen el valor esperado, los índices SPRT sobrepasan los límites superiores respectivos, los residuos se desvían y se generan alarmas (puntos verdes en las figuras), la categoría *'importante a revisar'* para temperatura 1 se manifiesta con 350 alarmas menos que el caso del punto 4.4.2, una reducción del 25%. La figura 27 muestra condición sin falla.

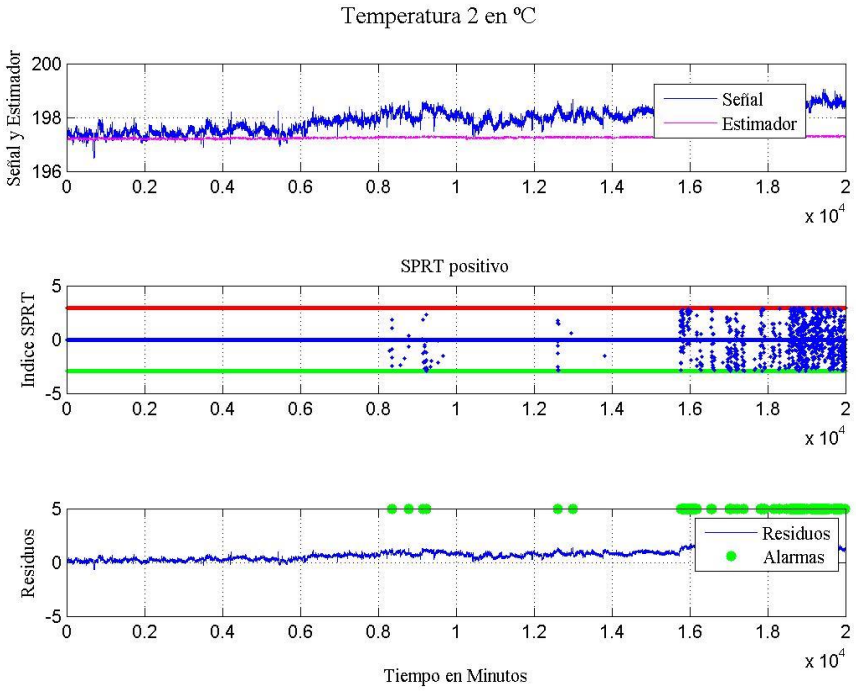

Figura 26. Resultados de temperatura 2 con temperatura 1 y 2 drift positivo.

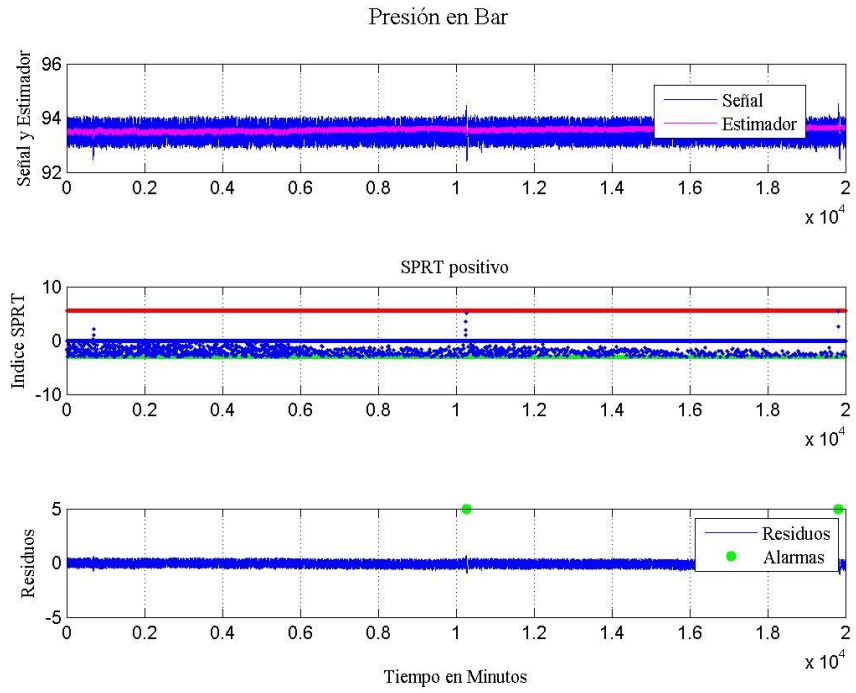

Figura 27. Resultados de presión con temperatura 1 y 2 drift positivo.

4.4.4. DRIFT EN TEMPERATURA 2

Estimaciones y alarma para el caso de una señal con drift.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	0	0
Temperatura 2	1421	630
Presión	3	0

Tabla 7. Caso de una señal con drift

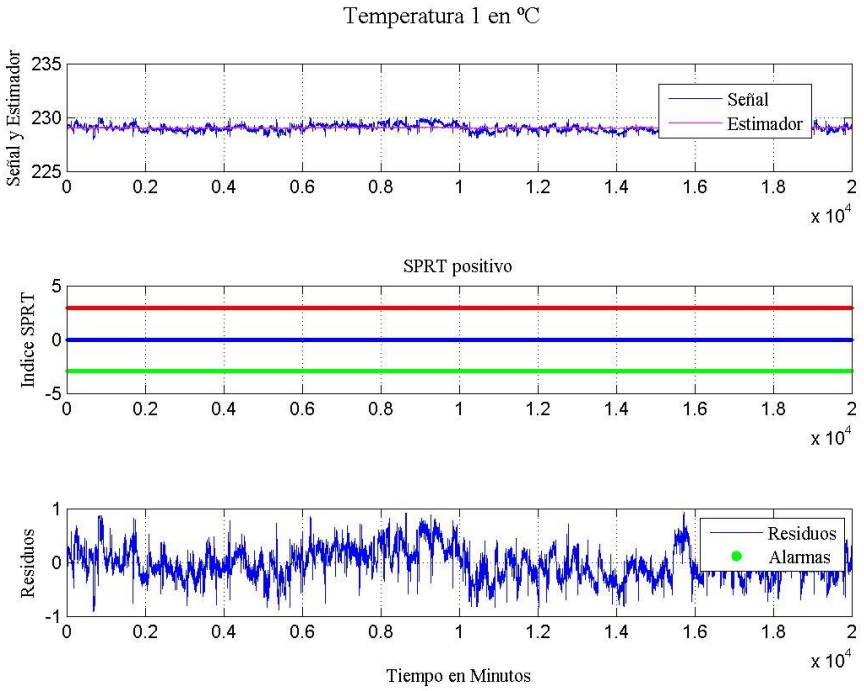

Figura 28. Resultados de temperatura 1 con temperatura 2 drift positivo.

En la figura 29 la estimación de la temperatura 2 sigue el valor esperado, el índice SPRT sobrepasa el límite superior, los residuos se desvían y se generan alarmas (puntos verdes en la figura), la categoría de *'importante a revisar'* se manifiesta con la secuencia continua de puntos al final en la gráfica de residuos, la cantidad de alarmas es menor que la cantidad de alarmas del caso 4.4.2, esto es porque temperatura 2 tiene menos ruido de proceso (ver figura 14). Las figuras 28 y 30 muestran condiciones sin falla.

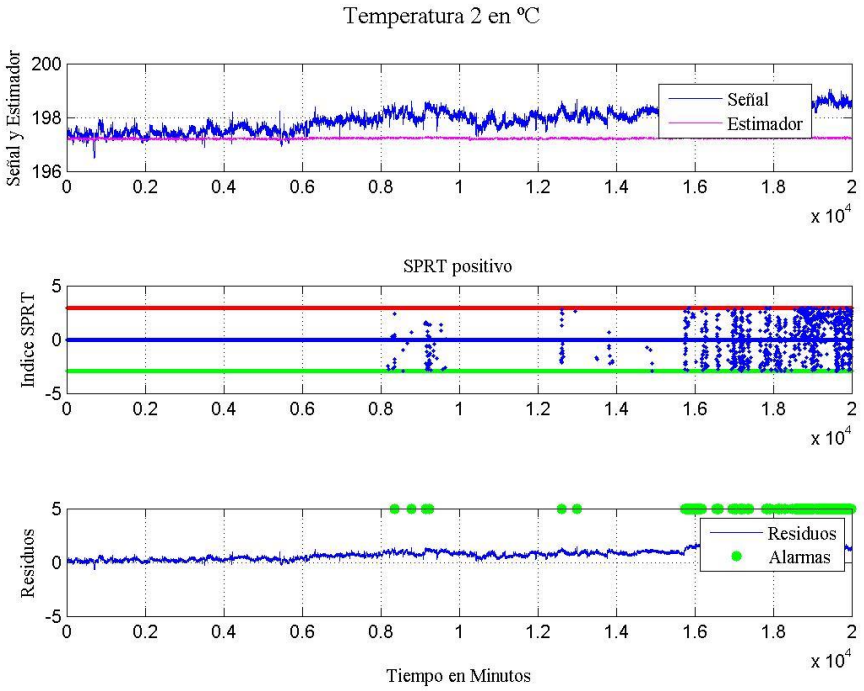

Figura 29. Resultados de temperatura 2 con temperatura 2 drift positivo.

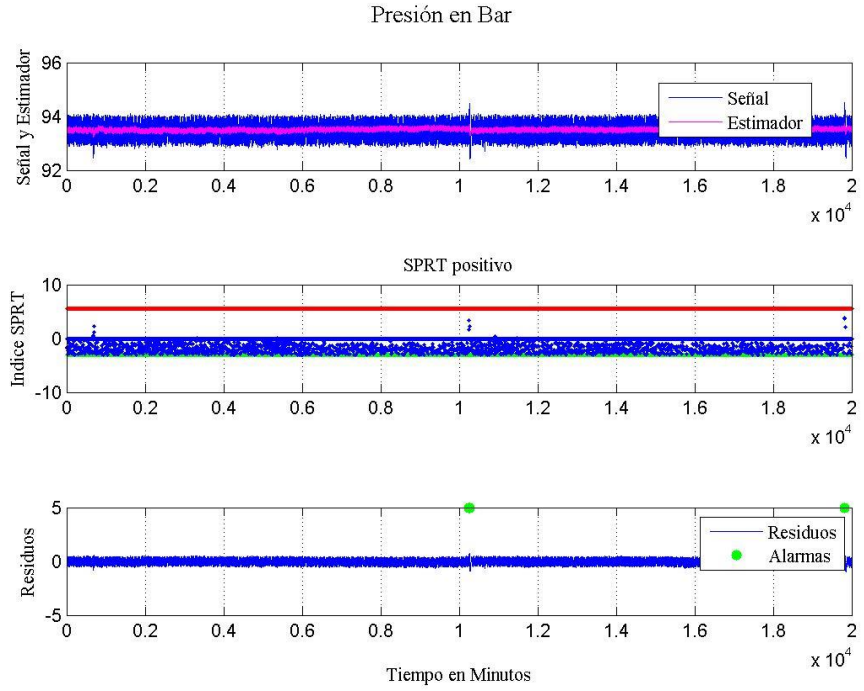

Figura 30. Resultados de presión con temperatura 2 drift positivo.

4.4.5. DRIFT EN PRESIÓN

Estimaciones y alarma para el caso de una señal con drift.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	0	0
Temperatura 2	0	0
Presión	210	2

Tabla 8. Caso de una señal con drift

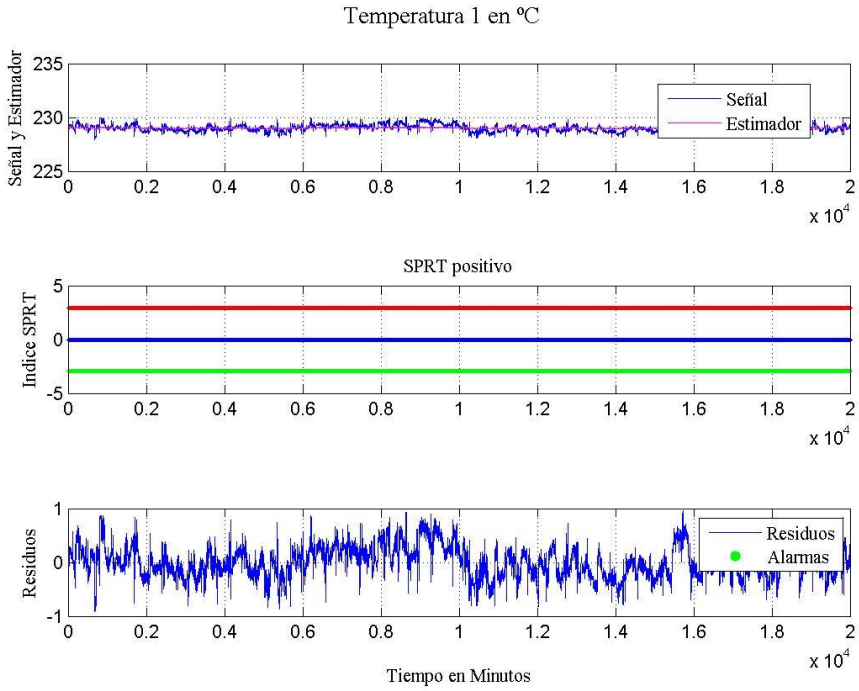

Figura 31. Resultados de temperatura 1 con presión drift positivo.

En la figura 33 la estimación de la presión sigue el valor esperado, el índice SPRT sobrepasa el límite superior, los residuos se desvían y se generan alarmas (puntos verdes en la figura), la categoría de *'importante a revisar'* se manifiesta con la secuencia continua de puntos al final en la gráfica de residuos, la cantidad alarmas en esta categoría es 2 alarmas debido a la naturaleza de la señal. Las figuras 31 y 32 muestran condiciones sin falla.

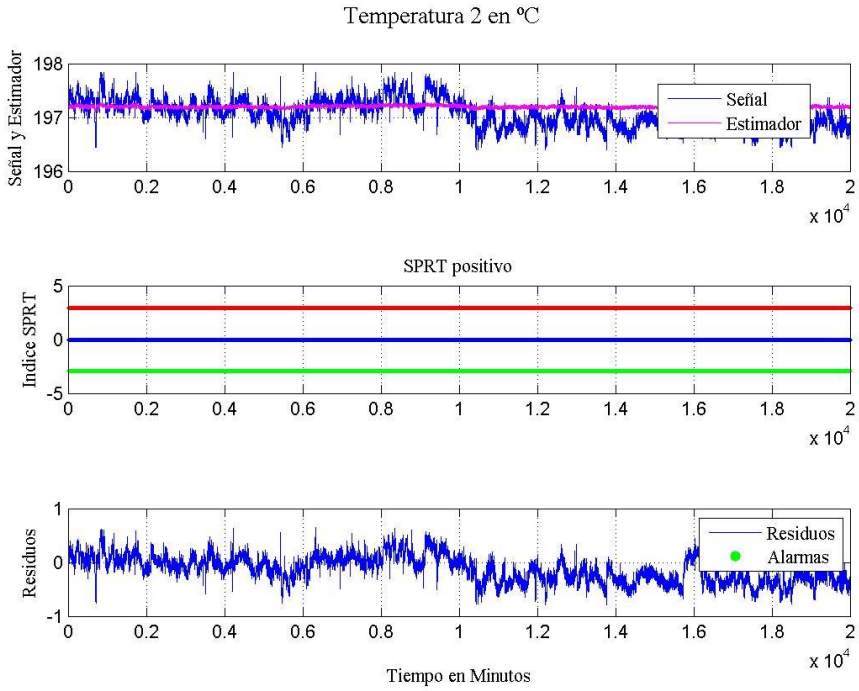

Figura 32. Resultados de temperatura 2 con presión drift positivo.

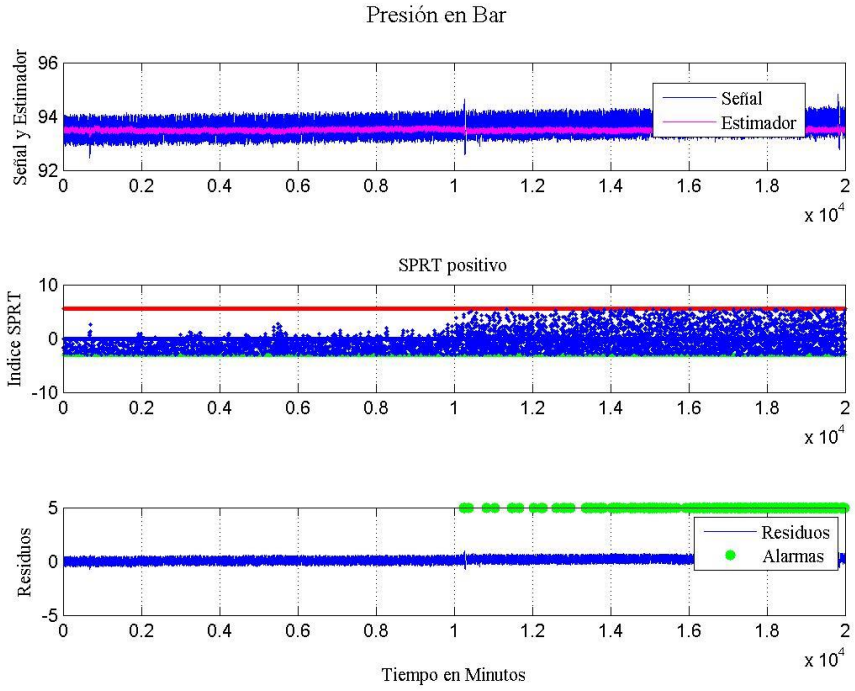

Figura 33. Resultados de presión con presión drift positivo.

4.4.6. DRIFT EN TEMPERATURA 1, 2 Y PRESIÓN

Estimaciones y alarma para el caso de tres señales con drift.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	1725	910
Temperatura 2	883	330
Presión	5	1

Tabla 9. Caso de tres señales con drift

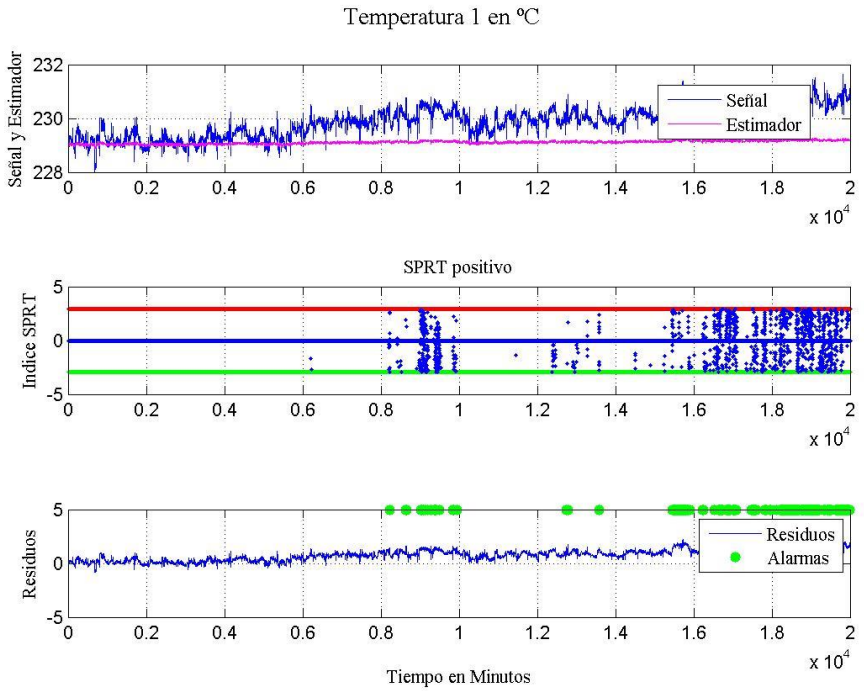

Figura 34. Resultados de temperatura 1 con temperatura 1, 2 y presión drift positivo.

Las estimaciones siguen el valor esperado, los índices SPRT sobrepasan el límite superior, los residuos se desvían y se generan alarmas (puntos verdes en la figura), la categoría de *'importante a revisar'* se manifiesta en los tres casos. Temperatura 1 se manifiesta con 480 alarmas menos que el caso del punto 4.4.2, una reducción del 35%, temperatura 2 tiene 330 alarmas que son 300 alarmas menos que en el punto 4.4.4, una reducción del 48% y la presión entrega 1 alarma en contra de 2 alarmas del punto 4.4.5.

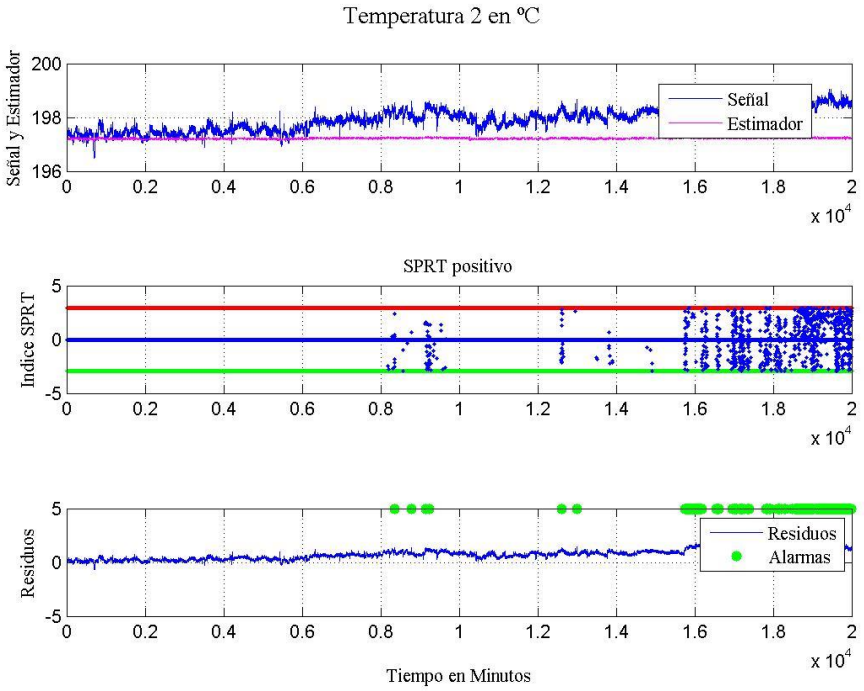

Figura 35. Resultados de temperatura 2 con temperatura 1, 2 y presión drift positivo.

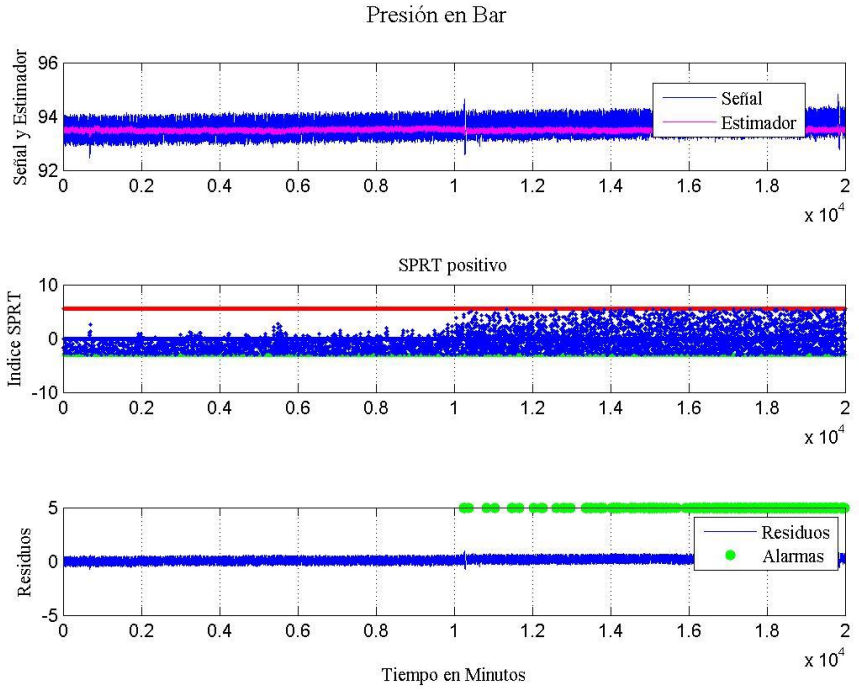

Figura 36. Resultados de presión con temperatura 1, 2 y presión drift positivo.

4.5. CONCLUSIÓN REGRESIÓN KERNEL AUTOREGRESIVA

Los distintos casos muestras las generación de alarmas para distintos casos, ningún drift, drift en una, dos o tres señales. Cuando el drift es solo sobre la temperatura 1 se generan 2276 alarmas, cuando es solo en temperatura 2 se generan 1421 alarmas y para el caso de drift solo en señal de presión se generan 210 alarmas. En cada uno de estos casos individuales se generan una cantidad significativa de alarmas que implican revisión de la señal y por consiguiente revisión de la instrumentación.

Para el caso de drift en temperatura 1 y temperatura 2 se tiene 1871 alarmas asociadas a temperatura 1, es decir una disminución de 405 alarmas respecto de caso de drift individual, para temperatura 2 se tienen 981 es decir una disminución de 409 alarmas. Claramente la cantidad de alarmas generadas en las señales de temperaturas, son menores que las generadas por drift individual. Esta disminución de cantidad de alarmas podría estar asociada a una posible variación del proceso y no a una anomalía de señal.

Para el caso de las tres señales con drift la cantidad de alarmas generadas disminuye aún más, si las comparamos con el caso individual de cada una, se tiene 1725 alarmas vs 2276 alarmas para temperatura 1 (reducción de 551 alarmas), 883 alarmas vs 1390 alarmas para temperatura 2 (reducción de 507 alarmas) y 5 alarmas vs 210 alarmas para la presión (reducción casi total de alarmas), este caso implica una variación de señales por cambios en las condiciones de proceso.

5. RESULTADOS REDES NEURONALES AUTOREGRESIVAS

Se detallan los resultados de la estimación mediante redes neuronales Autoregresivas, los datos de entrenamiento de la redes se corresponden con la matriz de entrenamiento utilizada en el método de Regresión Kernel Autoregresiva, ésto se hace para obtener una mejor comparación de desempeño de ambas técnicas.

Las señales de temperaturas y presión son las mismas que las utilizadas en el caso anterior.

5.1. ESTIMACIÓN DE VARIABLES

En la siguiente sección se muestran los distintos casos analizados y sus respectivos desempeños en cuanto a la estimación y generación de alarmas mediante resultados de redes neuronales Autoregresiva. El proceso correspondiente a la lógica de decisión es mediante el cálculo de índice *SPRT*, esta última etapa de proceso es idéntica a la usada en el caso de Regresión Kernel Autoregresiva. La diferencia comparativa radica en el proceso de estimación y no en la lógica de decisión. La selección de los valores α, β y μ se obtienen de como se describe en el punto 4.4. y los valores de estos parámetros se corresponden con los de dicho punto.

5.1.1 CASO SIN FALLA

Estimaciones y alarma para el caso sin drift.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	0	0
Temperatura 2	0	0
Presión	2	0

Tabla 10. Caso de tres señales sin drift

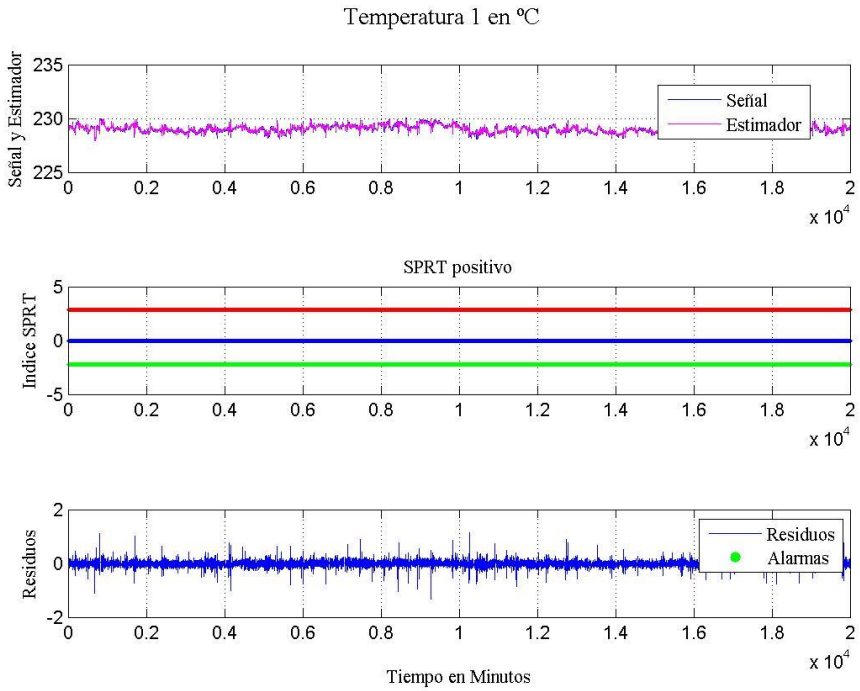

Figura 37. Resultados de Temperatura 1 sin falla.

Las figuras 37, 38 y 39 muestran que las tres estimaciones siguen a la señal, el índice *SPRT* para las dos temperaturas no supera el límite superior, para el caso de la presión se generan 2 alarmas consideradas falsas alarmas, los residuos en todos los casos están alrededor de cero. En este caso, no hay generación de alarma continua que implique categoría de *'importante a revisar'*.

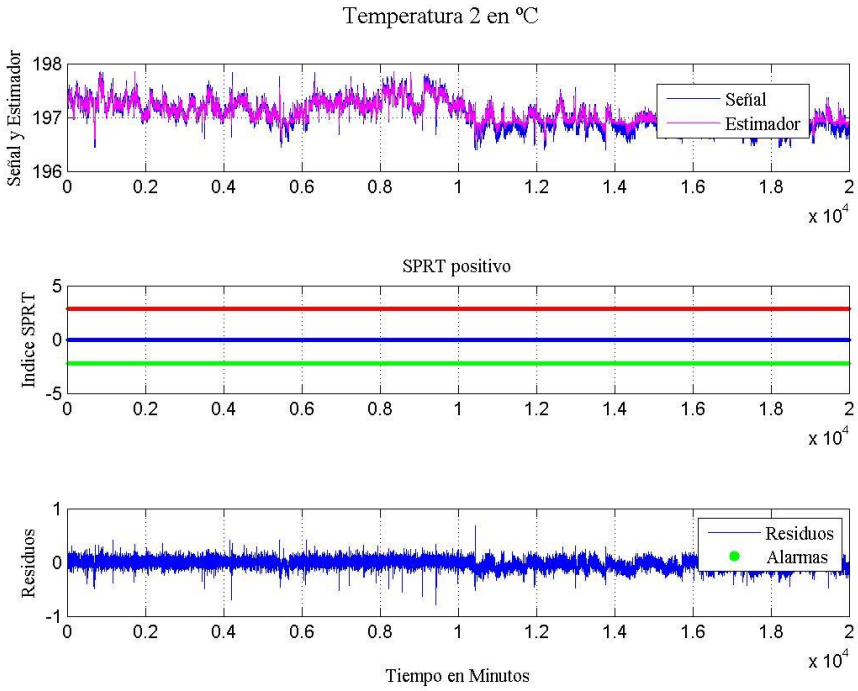

Figura 38. Resultados de Temperatura 2 sin falla.

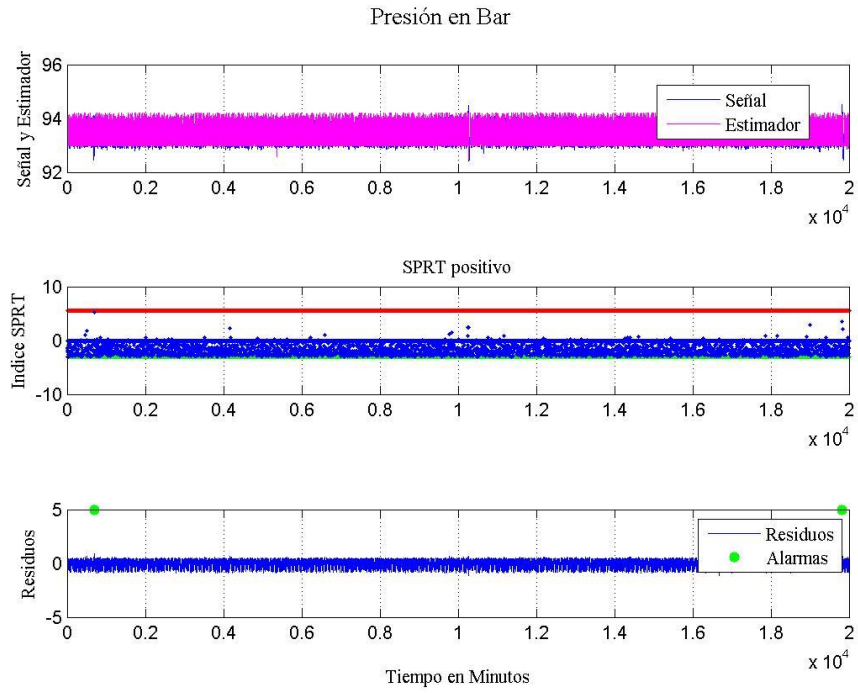

Figura 39. Resultados de Presión sin falla.

5.1.2 DRIFT EN TEMPERATURA 1, 2 Y PRESIÓN

Estimaciones y alarma para tres casos con drift.

Variable	Alarmas totales por drift positivo	Alarmas por drift positivo *'importante a revisar'*
Temperatura 1	6974	5150
Temperatura 2	5713	4030
Presión	1405	65

Tabla 11. Caso de tres señales con drift

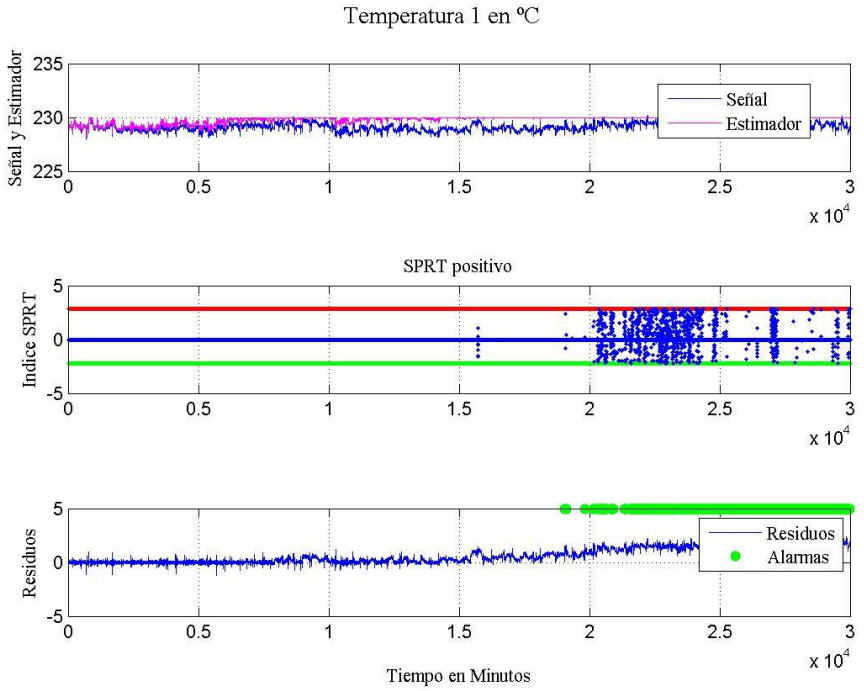

Figura 40. Resultados de temperatura 1 con temperatura 1, 2 y presión drift positivo.

Las estimaciones siguen el valor esperado, los índices *SPRT* sobrepasan el límite superior, los residuos se desvían y se generan alarmas (puntos verdes en la figura), la categoría de *'importante a revisar'* se manifiesta en los tres casos.

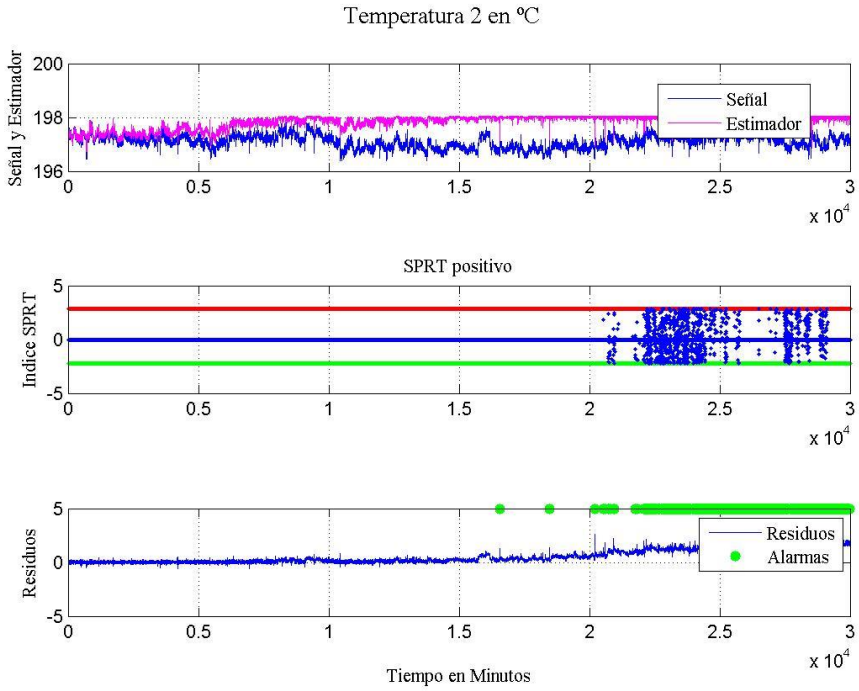

Figura 41. Resultados de temperatura 2 con temperatura 1, 2 y presión drift positivo

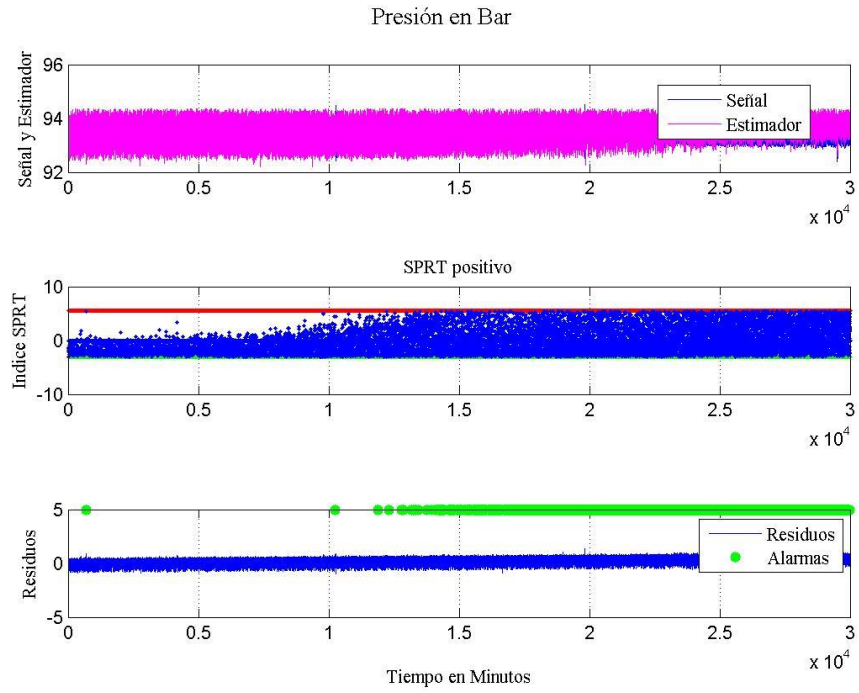

Figura 42. Resultados de presión con temperatura 1, 2 y presión drift positivo.

5.2. CONCLUSION REDES NEURONALES AUTOREGRESIVA

Los distintos casos muestran la generación de alarmas a ningún drift en tres señales y a drift sobre cada señal. La cantidad de alarmas generadas por temperatura 1 es de 6974 alarmas, para temperatura 2 es de 5713 alarmas y 1405 alarmas para la presión. El uso de redes neuronales genera estimaciones más azarosas alrededor del valor estimado, creando mayor cantidad de falsas alarmas.

Para el caso de condiciones normales de operaciones, las estimaciones siguen perfectamente la señal. Por otro lado, las figuras 40, 41 y 42 muestran claramente las alarmas generadas cuando el cambio de los residuos lo manifiesta. Se ve además, que la característica de los residuos es sustancialmente ruidosa, por lo que se entrega mayor cantidad de falsas alarmas.

6. CONCLUSIÓNES

Ambos métodos son aceptables para generación de estimaciones a partir de datos históricos conocidos y representativos de las condiciones del sistema a ser analizado.

Si observamos el modo de comportamiento en cuanto a las estimaciones, el método de Regresión Kernel genera menor cantidad de alarmas que las generadas por medio de la técnica de redes neuronales para un mismo caso. Esta característica puede mejorar, si previamente a procesar la estimación se filtra la señal a ser evaluada, de esta manera y mediante el uso de filtros pasa bajos se puede eliminar o disminuir el ruido de proceso dejando solamente la dinámica y tendencia de la señal, lo que disminuiría notablemente la cantidad de falsas alarmas. Esto se ve en la menor cantidad de alarmas generadas por temperatura 2 respecto de temperatura 1 (usando Regresión Kernel o redes neuronales), esto de debe a que la señal de temperatura 2 posee menos ruido, lo cual se ve claramente en la figura 14.

Otra característica a destacar, es que el tiempo de procesamiento que necesitan las redes neuronales es mayor a lo necesario por Regresión Kernel para una misma estimación, esta característica no es expuesta en detalle en este trabajo, ya que no afectaría la aplicación práctica.

El uso de Regresión Kernel es preferible, ya que la estimaciones presentan menor cantidad de falsas alarmas para una caso equivalente, por otro lado, es posible mejorar el desempeño de las dos técnicas por medio del ajuste de los parámetros α, β y μ de la lógica de decisión.

El uso de sistemas de monitoreo en línea permite detectar alarmas que según su cantidad generada se pueden asociar a condición de falla o condición normal, esta información aplicada a programas de mantenimiento permite reducir el número de calibraciones requeridas en cada para programada de una central nuclear, como así también reducir la exposición a la radiación, reducir tiempos y costos de mantenimiento.

7. REFERENCIAS

1. NUREG/CR-6895 Technical Review of On-Line Monitoring Techniques for Performance Assessment Volume 1: State-of-the-Art

2. NP-T-1.1 On-line Monitoring for Improving Performance of Nuclear Power Plants Part 1: Instrument Channel Monitoring

3. EPRI On-Line Monitoring of Instrument Channel Performance TR-104965-R1 NRC SER

4. Neural Networks and Learning Machines Third Edition Simon Haykin McMaster University
5. Hamilton, Ontario, Canada

6. Sequential probability ratio tests for reactor signal validation and sensor surveillance applications. Keith Humenik University of Maryland Baltimore, Kenny C. Gross Argonne National Laboratory

7. On-line calibration monitoring system based on data-driven model for oil well Sensors. Andre A. Boechat, Ubirajara F. Moreno,Decio Haramura, Jr. Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Norwegian University of Science and Technology, Trondheim, Norway, May 31 - June 1, 2012.

8. Introduction to signal processing. Sophocles J. Orfanidis Rutgers University 2010

9. Fault Detection in Nuclear Power Plants Components by a Combination of Statistical Methods. Francesco Di Maio, Piero Baraldi, Enrico Zio, Redouane Seraoui, 2014

10. Comparación de redes neuronales artificiales y análisis armónico para el pronóstico del nivel del mar. Erik Molino-Minero-Re, José Gilberto Cardoso-Mohedano, Ana Carolina Ruiz-Fernández, Joan-Albert Sanchez-Cabeza 2014

8. ANEXOS

ANEXO I

Código Matlab predicción Regresión Kernel Autoregresiva (código simplificado)

```
function [xestimadototal,res1,res2,res3]=AAKR3(x1,x2,x3,M,h)
%Estimador de señal AAKR (Auto-Associative Kernel Regression)
para redundancia de 3 señales
%x1      = señal de redundacia 1
%x2      = señal de redundacia 2
%x3      = señal de redundacia 3
%res1    = residuo entre estimacion AAKR de x1 y x1.
%res2    = residuo entre estimacion AAKR de x2 y x3.
%res3    = residuo entre estimacion AAKR de x3 y x3.
%xestimadototal = estimador en matriz
%M       = matriz de entrenamiento
%h       = ancho de banda seleccionado

%asignaciones de memoria para cada vector requerido
d=zeros(1,length(M));format long
w=zeros(1,length(M));format long
xestimadototal=zeros(length(x1),3); format long
res1 = zeros(1,length(x1));format long
res2 = zeros(1,length(x1));format long
res3 = zeros(1,length(x1));format long


j=1;
while j<=length(x1)

    %primera etapa
    for i=1:length(M);

        d(i)=sqrt((M(i,1)-x1(j))^2+(M(i,2)-x2(j))^2+(M(i,3)-
x3(j))^2);

    end

    %segunda etapa
    K = 1/sqrt(2*pi*(h^2));

    for i=1:length(M);

        w(i)=K*exp(-d(i)^2/h^2);

    end
```

```
%tercera etapa
    sumaw=sum(w);%sumo los elementos del arreglo w
    xestimado1 = 0;
    xestimado2 = 0;
    xestimado3 = 0;
    for i=1:length(M)
        xestimado1 = w(i)*M(i,1)+xestimado1;
        xestimado2 = w(i)*M(i,2)+xestimado2;
        xestimado3 = w(i)*M(i,3)+xestimado3;
    end
    xestimadototal(j,1)=xestimado1/sumaw;
    xestimadototal(j,2)=xestimado2/sumaw;
    xestimadototal(j,3)=xestimado3/sumaw;


    %generador de residuos para cada señal
    res1(j)=(x1(j)-xestimadototal(j,1));%residuos
    res2(j)=(x2(j)-xestimadototal(j,2));
    res3(j)=(x3(j)-xestimadototal(j,3));

    j=j+1;
end
```

ANEXO II

Codigo Matlab predicción redes neuronales Autoregresivas (código simplificado)

```
function  outputs2 = ARNN(x,m)
% x = señal
% m = target, señal linea base condición normal
% y = señal estimada

targetSeries = tonndata(m,false,false);
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narnet(feedbackDelays,hiddenLayerSize);
[inputs,inputStates,layerStates,targets] =
preparets(net,{},{},targetSeries);
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
% Entrenamiento
[net,tr] = train(net,inputs,targets,inputStates,layerStates);
% Evaluación
C = num2cell(x);
outputs2 = net(C',inputStates,layerStates);
```

ANEXO III

Codigo Matlab logica de decisión (código simplificado)

```
function [SPRT
ubic]=sprttestpos(x,x1,alfa,beta,mu1,var_r,nro)
%x     = resuido de la señal
%alfa  = probabilidad de falsa alarma, ej 0.05 es un 5% de
falsa alarma.
%beta  = probabilidad de alarma perdida, ej 0.1 ew un 10% de
alarma perdida.
%mu1   = valor asociado al drift adpotado como error,
%var_r = es la varianza de los residuos en zona estable.
%nro   = cantidad de muestras seguidas a partir de donde se
%        dispara alarma.

n=length(x);%long del vector residuos
A=(beta/(1-alfa));
B=((1-beta)/alfa);

%Contadores en cero.
cont_alarma=0;
cont_condicion_normal=0;
cont_no_decide = 0;

SPRT=zeros(1,n);
ubic=zeros(1,n);  ubic(ubic==0) = NaN;

%Primer calculo del indice SPRT inicia en cero.
SPRT(1) = 0.0;
i=2;

while (i<n)

    SPRT(i) = SPRT(i-1)+(mu1/var_r)*(x(i)-(mu1/2));

    if SPRT(i) <= log(A) %Acepto H0, condicion normal
cont_condicion_normal=cont_condicion_normal+1;
        SPRT(i) = 0.0;
    elseif  SPRT(i) >= log(B) %Acepto H1, Rechazo H0,
        cont_alarma=cont_alarma+1;
        ubic(i)= 5;
        SPRT(i) = 0.0;
    end
```

```
    if log(A)< SPRT(i) < log(B)
        cont_no_decide = cont_no_decide +1;
    end
    i=i+1;
end

cont = 0;
for i=(nro+1):length(x)
    if ubic(i-nro:i) == 5
        cont = cont + nro;
        ubic(i-nro:i) = 10;
    end
end
```