

## Sensor virtual neuronal con variables instrumentales y su aplicación en un Convertidor Teniente

Alejandro Suárez S.\* Zacarías Gómez U.\*\*

\*Departamento de Electrónica, Universidad Técnica Federico Santa María, Av. España n°1680, Valparaíso, Chile, (email: alejandro.suarez@usm.cl)

\*\*Unidad de Automatización, Superintendencia Ingeniería de Procesos y Desarrollo, Codelco Chile División El Salvador, Av. Bernardo O'Higgins n° 103, El Salvador, Chile (e-mail: zgomez001@codelco.cl)

Resumen: Se propone el diseño de un sensor virtual para medir la temperatura del baño de metal blanco en un Convertidor Teniente, utilizando redes neuronales a través de un nuevo método de entrenamiento que utiliza el concepto de variables instrumentales. Este nuevo tipo de entrenamiento será comparado con el método de gradiente descendente, el cual al estar basado en ajuste por mínimos cuadrados presenta sesgo en sus parámetros, producto del ruido de medición. La apuesta es que el método de variables instrumentales resuelva este problema, entregando una red con parámetros ajustados sin sesgo, lo que se verá reflejado en que la salida de esta red, se ajustará de mejor forma a la señal real que el método tradicional de gradiente descendente. Los resultados demuestran que la propuesta planteada entrega un sensor con mejor ajuste que el algoritmo tradicional cuando el instrumento real no se encuentra disponible. La aplicación específica del sensor virtual de temperatura para el Convertidor Teniente presenta gran interés para la industria debido al alto costo de los instrumentos que actualmente pueden cumplir dicha función. © 2011 CEA

Palabras Clave: sensores virtuales, redes neuronales, variables instrumentales, procesos de producción de cobre, instrumentación y mediciones virtuales.

### 1. INTRODUCCIÓN

Los métodos tradicionales de regresión de lineal por mínimos cuadrados presentan problemas cuando las variables medidas, utilizadas como entrada de un modelo, tienen ruido de medición, puesto que se genera un sesgo en el ajuste de los parámetros. Así mismo, los métodos tradicionales de entrenamiento de las redes neuronales, que utilizan los mínimos cuadrados para el ajuste de los pesos de ponderación, presentan también este sesgo al tratarse de datos con ruido de medición.

Una de las formas de solucionar esta problemática en los métodos de regresión lineal tradicionales, es por medio de los métodos de variables instrumentales, los que utilizan variables nuevas, creadas de manera artificial, y que deben cumplir con determinadas condiciones para permitir que el ajuste de los parámetros no se vea afectado por el ruido de medición que genera el sesgo en la estimación de los parámetros.

En este trabajo se propone utilizar una metodología similar para el ajuste de los pesos de ponderación de las redes neuronales, de forma tal que el ruido de medición no afecte en el ajuste de dichos pesos de ponderación de las neuronas. La metodología es aplicada en la implementación de un Sensor Virtual (SV) para la medición de temperatura en un Convertidor Teniente, perteneciente a la Fundición Potrerillos ubicada en la tercera región de Chile.

En la sección 2 se describe en forma general el proceso y las variables que participan en él. La sección 3 y 4 entregan una breve descripción de los métodos de regresión lineal y la introducción de variables instrumentales. Esto permite realizar una analogía con los métodos de ajuste en redes neuronales,

descrito en las secciones 5 y 6, para llegar al algoritmo a utilizar en el presente trabajo, descrito en la sección 7. La sección 8 comenta la implementación del SV. En la sección 9 se presentan los resultados obtenidos, para llegar a las conclusiones que se presentan en la sección 10.

#### 1.1. Identificación de Problemas

Los principales problemas que se abordan en este trabajo, son los siguientes:

- Adaptación del método de variables instrumentales al entrenamiento de una red neuronal.
- Estudio de un modelo adecuado de red neuronal para la aplicación específica.
- Recopilación de datos de proceso suficientes para el entrenamiento de las redes.
- Implementación de método de entrenamiento propuesto.

#### 1.2. Soluciones y abordajes realizadas por otros autores.

A la fecha, existen múltiples aplicaciones de sensores virtuales modelados con redes neuronales en la industria, que abarcan campos tales como: Simulación y control de procesos químicos, Predicción del clima, Biotecnología, Identificación de polímeros, Mantenimiento preventivo en la localización incipiente de fallas, entre otros. En la industria del cobre, se encuentran desarrollos en su mayoría en Plantas concentradoras, como por ejemplo los siguientes: sensor virtual para el índice de trabajo de mineral sobre la base de su litología (Ceballos, 2001), sensor virtual de concentración de sólidos en el rebose de hidrociclones en

circuitos de molienda (Bustos 1999), sensor virtual de la concentración de sólidos en la descarga de hidrociclones (Jordan, 1998), sensor virtual de ley de concentrado rougher (Barrera, 2007), sensores virtuales mediante análisis de conglomerados (Espinoza, 1995); además se han desarrollado plataformas de software que facilitan la implementación de nuevos sensores (Tapia, 2000).

En este trabajo se propone implementar un sensor virtual utilizando una metodología nueva para el entrenamiento de las redes neuronales, que consiste en aplicar la idea de las variables instrumentales en combinación con el algoritmo de entrenamiento tradicional de gradiente descendente. El objetivo es eliminar el sesgo que se produce al tener ruido de medición en las variables utilizadas para la estimación de parámetros de modelos de tipo dinámico. La utilización de filtros para eliminar el ruido de medición es ampliamente conocida, sin embargo la técnica propuesta a través de variables instrumentales es más amplia y el filtraje de señales se puede considerar como un caso particular de la metodología propuesta.

El avance de las técnicas desarrolladas para evitar los errores en el entrenamiento de las redes, en términos generales han apuntado a evitar el estacionamiento de los algoritmos en mínimos locales, para lo cual existen desarrollos que a estas alturas forman parte de la teoría general del entrenamiento de redes neuronales. Dentro de estas técnicas tradicionales, se encuentra el algoritmo de gradiente descendente, que consiste en realizar el cálculo de error para cada muestra que se presenta y ajustar los pesos para reducir al mínimo la diferencia entre la salida de la red y la señal deseada. Para evitar el estacionamiento en mínimos locales, este algoritmo se modifica a través de la incorporación de una razón de aprendizaje (Haykin, 1998), la que otorga a la red la capacidad para moverse desde estos mínimos locales y alcanzar el mínimo global.

Otra alternativa, la ofrece el método de Filtro de Kalman Extendido, que se puede utilizar para el entrenamiento de las redes, considerando que los pesos son los “estados” del sistema. Esta metodología usa la arquitectura feedforward, pero usa una estimación recurrente del gradiente, no es instantáneo como en el caso de gradiente descendente. Esto produce un entrenamiento que es menos propenso a ser atrapado en mínimos locales, pero mucho más complejo en cuanto a los cálculos involucrados. Una modificación consiste en considerar que los pesos pueden ser agrupados en grupos lógicos y ser entrenados en forma independiente (Pasika *et al.*, 1998). Con esto disminuye considerablemente la complejidad computacional y permite aplicar esta técnica en redes de gran envergadura.

## 2. DESCRIPCIÓN DEL PROCESO

El Convertidor Teniente (*CT*) es un reactor basculante de tecnología desarrollada por Codelco, en Chile. Dentro de este reactor se produce el proceso de fusión y conversión de concentrado de cobre, el que se transforma en Metal Blanco y Escoria. El *CT* tiene forma cilíndrica y se mantiene en forma horizontal, por lo que le es posible girar cuando lo requiera. Sus dimensiones son 5 [m] de diámetro por 22 [m] de largo. Está construido en acero, y es recubierto interiormente con ladrillo refractario, lo que le permite alargar su vida útil.

A este Convertidor se le inyecta Concentrado Seco, en forma continua, directo al baño de cobre (en forma sumergida), a través de toberas especiales diseñadas para este propósito. También mediante toberas sumergidas se inyecta aire de baja presión, que se encuentra enriquecido con oxígeno en alrededor de un 35%.

El concentrado se funde en forma térmicamente autógena, no siendo necesario el uso de quemadores más que para calentar el *CT* y llevarlo al punto de operación, principalmente después de un período de mantenimiento prolongado. Por medio de correas transportadoras es posible agregar concentrado húmedo (que permite controlar las temperaturas del líquido), circulante (material que retorna desde la Fundición y se reprocesa para disminuir las pérdidas) y sílice (que facilita la formación de escorias).

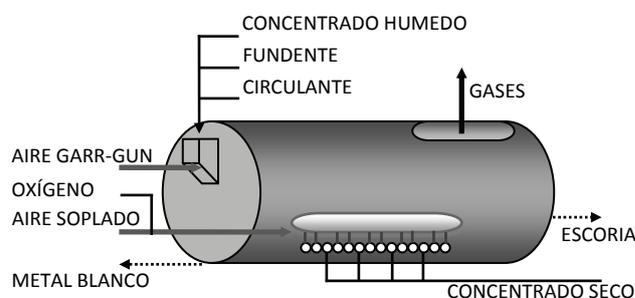


Figura 1. Convertidor Teniente

El *CT* trabaja con temperaturas entre 1200 y 1300 °C en su interior. Una vez que los materiales están en fase líquida, se produce la conversión y posterior separación de fases, debido a las diferentes densidades entre los materiales. Los siguientes son los materiales derivados de este proceso:

- Metal Blanco líquido, con 72-75% Cu, 4%Fe, 21%S
- Escoria líquida, con 8% Cu, 40%Fe y 24%SiO<sub>2</sub>.
- Gases con alto contenido de SO<sub>2</sub> y contenido moderado de polvo.

El Metal Blanco es trasladado hacia los Convertidores Peirce-Smith, que es la etapa siguiente dentro del Proceso Fundición. Por su parte, la escoria derivada del *CT*, tiene altos contenidos de cobre, por lo que es tratada en Hornos de Limpieza de Escorias, en los que se recupera gran parte de dicho cobre. Los gases emanados del Proceso de Fusión-Conversion tienen altos contenidos de azufre y polvos, por lo que son tratados para recuperar polvos (recirculados a la Fundición) y producir ácido sulfúrico.

La temperatura del baño de cobre al interior del *CT* se puede medir en forma directa, a través de un pirómetro óptico ubicado en una de las toberas de soplado. Esta medición se ajusta en forma periódica mediante la contrastación con termocuplas. Este pirómetro es dañado ocasionalmente por diversos motivos, tanto operacionales como de mantenimiento, quedando fuera de servicio. Debido a la criticidad de este sensor, se requiere contar con una gran cantidad de repuestos para evitar que esta medición permanezca en esta condición por periodos de tiempo prolongados. Por ello, se estima que esta aplicación es apropiada para la construcción de un sensor virtual, tal que pueda estimar la temperatura del baño en los periodos en que el pirómetro se encuentra fuera de servicio, bajando la criticidad de los repuestos y asegurando la medición de temperatura para el control apropiado de la operación del *CT*.

Para implementar este sensor virtual, es necesario encontrar señales que estén correlacionadas con la variable a inferir. Se dispone de las mediciones que aparecen en la tabla 1, en adelante se utilizará el identificador (TAG) para hacer mención a cada variable. Utilizando señales registradas en un día de operación normal, se obtienen los coeficientes de correlación que se

presentan en la tabla 2. Dichos coeficientes son calculados entre las variables mencionadas en la tabla 1 y la medición TIT-319.

**Tabla 1. Señales disponibles en el Convertidor Teniente**

TAG	Señal	Rango
FIT-318	Flujo Aire de Soplado	0 - 60.000 Nm <sup>3</sup> /h
FIT-323	Flujo Aire Garr-gun	0 - 5.000 Nm <sup>3</sup> /h
FIT-329	Flujo Oxígeno	0 - 20.000 Nm <sup>3</sup> /h
WIT-436	Flujo Concentrado Seco	0 - 150 ton/h
FIT-412	Aire Inyección Conc. Seco	0 - 900 Nm <sup>3</sup> /h
WIT-201	Flujo Concentrado Húmedo	0 - 60 ton/h
WIT-202	Flujo Sílice	0 - 32 ton/h
WIT-203	Flujo Circulante	0 - 32 ton/h
TIT-101B	Temperatura Gases Salida	0 - 800 °C
PIT-101B	Presión Gases de Salida	0 - (-)12 KPa
TIT-319	Temperatura Baño Cobre	750 - 1.500 °C

**Tabla 2. Coeficientes de correlación entre Señales disponibles y medición TIT-319**

TAG	Coef. de Correl.	TAG	Coef. de Correl.
FIT-318	0,544	TIT-101B	0,469
FIT-323	0,539	PIT-101B	0,459
FIT-329	0,537	WIT-202	0,436
WIT-436	0,491	WIT-201	0,315
FIT-412	0,471	WIT-203	0,059

### 3. MÉTODOS DE REGRESIÓN LINEAL

Antes de profundizar en la teoría de las redes neuronales, es de interés comentar algunos conceptos de los métodos de identificación tradicionales, con foco en los métodos de regresión lineal con ajuste por mínimos cuadrados. Esto para mostrar la gran similitud que se presenta con los métodos de entrenamiento de las redes neuronales.

Se consideran modelos tipo ARX, los que pueden escribirse de la siguiente forma

$$y(t) = \varphi^T(t) \cdot \theta + \varepsilon(t) \quad (1)$$

donde  $y(t)$  es la variable medida,  $\varphi(t)$  es un vector de cantidades conocidas y  $\theta$  es un vector de parámetros desconocidos. La señal  $\varepsilon(t)$  representa las diferencias producidas entre el resultado del modelo y la señal real, lo que se debe a dinámicas no modeladas, errores de medición, etc. Expresado en forma matricial queda de la siguiente forma

$$Y = \phi \cdot \theta_0 + \varepsilon \quad (2)$$

con

$$Y = \begin{pmatrix} y(1) \\ \vdots \\ y(N) \end{pmatrix} \quad \phi = \begin{pmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(N) \end{pmatrix} \quad \varepsilon = \begin{pmatrix} \varepsilon(1) \\ \vdots \\ \varepsilon(N) \end{pmatrix} \quad (3)$$

Al trabajar con cuadrados mínimos se considera la siguiente función de costo

$$J(\theta) = \frac{1}{2} \sum_{t=1}^N \varepsilon^2(t) = \frac{1}{2} \varepsilon^T \varepsilon \quad (4)$$

Con esto, el punto mínimo de la función de costo está dado por el estimador  $\hat{\theta}$

$$(\phi^T \cdot \phi) \cdot \hat{\theta} = \phi^T \cdot Y \quad (5)$$

Lo que también se puede expresar como

$$[\sum_{t=1}^N \varphi(t) \varphi^T(t)] \cdot \hat{\theta} = [\sum_{t=1}^N \varphi(t) y(t)] \quad (6)$$

Esto se encuentra desarrollado en extenso en la referencia (Ljung, 1999). Así también, en la misma referencia se analiza la consistencia del estimador  $\hat{\theta}$  para converger al vector de parámetros verdadero  $\theta_0$ . Llegándose a que

$$E\{\hat{\theta}\} = E\{\theta_0\} + E\{(\phi^T \cdot \phi)^{-1} \cdot \phi^T \cdot \varepsilon\} \quad (7)$$

Por lo tanto, si se tiene que  $\varphi(t)$  es independiente de  $\varepsilon(t)$ , se tiene que

$$E\{\hat{\theta}\} = E\{\theta_0\} + E\{(\phi^T \cdot \phi)^{-1} \cdot \phi^T\} \cdot E\{\varepsilon\} \quad (8)$$

Por lo que basta con que  $E\{\varepsilon\} = 0$ . Condición que se obtiene cuando  $\varepsilon(t)$  es ruido blanco de valor medio cero. Luego, se tiene que  $E\{\varepsilon\} = 0$ , y con ello,

$$E\{\hat{\theta}\} = \theta_0 \quad (9)$$

Cuando el vector  $\varphi(t)$  está correlacionado con  $\varepsilon(t)$ , evento que ocurre por ejemplo, cuando se utilizan ecuaciones recursivas, no puede lograrse el resultado (8). Además, si el ruido no puede considerarse como ruido blanco, tampoco se obtiene el resultado (9).

### 4. MÉTODOS DE VARIABLES INSTRUMENTALES

En la teoría de identificación clásica, los métodos de Variables Instrumentales (VI) entregan una solución a este problema, modificando la ecuación normal de mínimos cuadrados mediante la introducción de un nuevo vector  $\zeta(t)$ , de la siguiente forma (Ljung, 1999)

$$[\sum_{t=1}^N \zeta(t) \varphi^T(t)] \cdot \hat{\theta} = [\sum_{t=1}^N \zeta(t) y(t)] \quad (10)$$

En esta ecuación la elección del vector  $\zeta(t)$  juega un rol importante. Existen dos condiciones que  $\zeta(t)$  debe cumplir (Ljung, 1999) para asegurar la consistencia del estimador  $\hat{\theta}$

$$1. \quad E[\zeta(t) \cdot \varphi^T(t)] \text{ sea no singular} \quad (11)$$

$$2. \quad E[\zeta(t) \cdot \varepsilon(t)] = 0 \quad (12)$$

Esto implica que en la elección de  $\zeta(t)$ , se debe formar un vector correlacionado con el vector de regresores, pero no correlacionado con el ruido.

La elección del vector  $\zeta(t)$  es de importancia en los resultados que pueden obtenerse con este método. En general, se busca un vector similar a  $\varphi(t)$ , el cual se modifica para cumplir con las condiciones recién nombradas, que indican que no debe ser influenciado por el error (Ljung, 1999).

### 5. REDES NEURONALES

Con esta información, se revisa parte de la teoría clásica de la Redes Neuronales (Haykin, 1998). A modo de definir la notación a utilizar, considérese el siguiente modelo de neurona

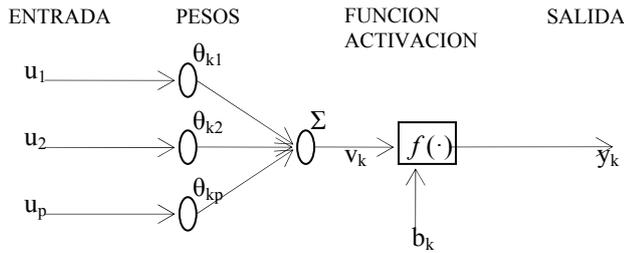


Figura 2. Modelo de una Neurona

Las ecuaciones que describen esta neurona son las siguientes

$$v_k = \sum_{j=1}^p \theta_{kj} u_j \quad y_k = f(v_k - b_k) \quad (13)$$

En este caso, se denomina como  $d$  la respuesta deseada de la red. Con esto, la señal de error es la siguiente  $e = d - y$ , y se busca minimizar la función de costo siguiente

$$J = \frac{1}{2} E[e^2] \quad (14)$$

expandiendo esta expresión, e igualando el gradiente de dicha expresión a cero, se llega a que

$$\nabla_{\theta_k} J = 0 \Rightarrow \sum_{j=1}^p \theta_{0j} E[u_j u_k] = E[u_k d] \quad k = 1, 2, \dots, p \quad (15)$$

Estas ecuaciones, se conocen como las “ecuaciones de Wiener-Hopf”. La solución de estas ecuaciones implica invertir una matriz de  $p \times p$ , que contiene los valores asociados a la función de autocorrelación  $E[u_j u_k]$ . Para evitar la inversión matricial, se utiliza el método de Gradiente Descendente que se describe en la próxima sección. Una limitación para resolver estas ecuaciones, es que se debe conocer las funciones de autocorrelación para  $u$  y la correlación cruzada entre  $u$  y  $d$ . Cuando el comportamiento del filtro es desconocido, estas funciones no están disponibles y por lo tanto, se deben utilizar aproximaciones de ellas. Una alternativa es usar las aproximaciones siguientes (Haykin, 1998)

$$\hat{E}[u_j u_k]_n = u_j(n) u_k(n) \quad \hat{E}[u_k d]_n = u_k(n) d(n) \quad (16)$$

Con estas ecuaciones, los cálculos asociados a las ecuaciones anteriores, conducen a la inversión de una matriz singular, y por lo tanto, no presentan una solución única.

Si se comparan estas últimas ecuaciones, con las expresiones obtenidas para los métodos de regresión por mínimos cuadrados, se observan las similitudes que se explican debido a que ambos métodos utilizan el error cuadrático como funcional de costo a optimizar. A pesar de ello, las expresiones no son equivalentes, y esto se produce porque en este último caso se consideran las aproximaciones (16).

El algoritmo de gradiente descendente, utiliza las ecuaciones derivadas para la neurona, pero evita la inversión matricial utilizando formas variantes en el tiempo para los estimadores de los pesos de ponderación  $\hat{\theta}_i$ .

$$\Delta \hat{\theta}_k(n) = -\eta \nabla_{\theta_k} J(n) \quad k = 1, 2, \dots, p \quad (17)$$

Donde  $\eta$  es el factor de aprendizaje. Luego, dependiendo del gradiente que se obtenga será la dirección y la magnitud de la

corrección  $\hat{\theta}_k$ . Utilizando la expresión para el gradiente y reemplazando las aproximaciones de (16), se puede llegar desde la expresión (17) a la (18) (Haykin, 1998).

$$\hat{\theta}_k(n+1) = \hat{\theta}_k(n) + \eta [d(n) - y(n)] u_k(n) \quad k = 1, 2, \dots, p \quad (18)$$

Una estructura similar se obtiene cuando se llevan los métodos de regresión a una forma variante en el tiempo.

## 6. ALGORITMO GRADIENTE DESCENDENTE

La derivación de este algoritmo también se considera parte de teoría clásica para redes neuronales (Haykin, 1998), para efectos de este trabajo, sólo interesa explicar en qué consiste y cómo utilizar dicho algoritmo.

La corrección que se realiza en cada uno de los pesos se basa en la regla delta, que es la aplicación de la ecuación (17) para cada neurona (Haykin, 1998)

$$\begin{pmatrix} \text{corrección} \\ \text{peso} \\ \Delta \theta_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{razón} \\ \text{aprendizaje} \\ \eta \end{pmatrix} \begin{pmatrix} \text{gradiente} \\ \text{local} \\ \delta_j(n) \end{pmatrix} \begin{pmatrix} \text{entrada} \\ \text{neurona} \\ y_i(n) \end{pmatrix} \quad (19)$$

Para una neurona de una capa de salida, el cálculo del error es directo puesto que se compara la salida de dicha neurona con respecto a la salida deseada de la red. Para una neurona de una capa oculta, el cálculo del error debe seguir algún criterio de cálculo. Este problema es resuelto por el algoritmo de *backpropagation* de la siguiente forma.

### 6.1. Neurona perteneciente a la capa de salida de la red.

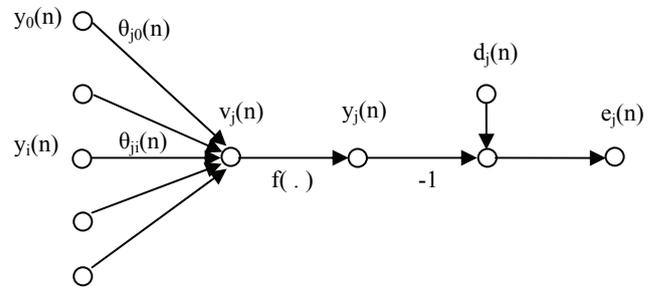


Figura 3. Modelo general para una Neurona

En este caso, la salida de la neurona  $y_j(t)$  se compara directamente con la señal deseada de salida, calculando el error de la siguiente forma

$$e_j(n) = d_j(n) - y_j(n) \quad (20)$$

Recordando que se debe minimizar el error cuadrático, se puede llegar a la siguiente expresión para la regla delta (Haykin, 1998)

$$\Delta \theta_{ji}(n) = -\eta \frac{\partial \xi(n)}{\partial \theta_{ji}(n)} = \eta \delta_j(n) y_i(n) \quad (21)$$

donde  $\delta_j$  es el gradiente local

$$\delta_j(n) = -\frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) f'(v_j(n)) \quad (22)$$

## 6.2. Neurona perteneciente a las capas ocultas de la red.

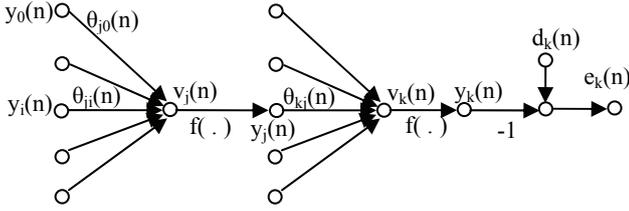


Figura 4. Modelo para neurona de capa oculta.

Para las neuronas de las capas ocultas, no hay una respuesta deseada específica para dicha neurona, por lo que no se puede determinar el error en forma directa. En este caso, el error se calcula en términos de todas las señales de error a las cuales dicha neurona se encuentra conectada.

$$\xi(n) = \frac{1}{2} \sum_{k \in c} e_k^2(n) \quad (23)$$

Con esta consideración y en forma similar al caso anterior, se puede llegar a la siguiente expresión para el gradiente local (Haykin, 1998)

$$\delta_j(n) = f_j'(v_j(n)) \sum_k \delta_k(n) \theta_{kj}(n) \quad (24)$$

El primer término es la derivada de la función de activación en  $v_j$ . El segundo término corresponde a la suma ponderada de los  $\delta$ 's de las neuronas de la capa siguiente, ya sea capa oculta o capa de salida, a las cuales se encuentra directamente conectada la neurona en cuestión.

## 7. ALGORITMO DE VARIABLES INSTRUMENTALES

El método propuesto consiste en la adaptación del método de variables instrumentales a las redes neuronales, de tal forma de lograr que los parámetros de la red puedan ser estimados sin sesgo.

En forma análoga a lo realizado en los métodos paramétricos, la modificación que se debe realizar al algoritmo se basa en introducir un vector de entradas para el ajuste de los pesos de las neuronas. Anteriormente se había determinado que los pesos de las neuronas son determinados por

$$\sum_{j=1}^p \{\theta_{0j} E[u_j u_k]\} = E[u_k d] \quad (25)$$

donde  $u$  es el vector de las entradas y  $d$  es el vector de salidas deseadas de la red. Esta forma es similar a la ecuación (6), por lo que la forma de introducir el vector de  $VI$  en este caso es también similar a lo presentado en la ecuación (10) con el vector  $\zeta(t)$ . Esto es válido porque las redes que estamos utilizando son lineales, por lo tanto las ecuaciones que determinan el ajuste de los pesos son las mismas. Para este caso, el vector de  $VI$  se denomina  $z$ , por lo que la forma de obtener los pesos de ponderación de las neuronas estará dada por

$$\sum_{j=1}^p \{\theta_{0j} E[u_j z_k]\} = E[z_k d] \quad (26)$$

Luego, interesa desarrollar la modificación para el algoritmo de gradiente descendente con variables instrumentales.

$$\hat{\theta}_k(n+1) = \hat{\theta}_k(n) - \eta \nabla_{\theta_k} J(n)$$

$$\hat{\theta}_k(n+1) = \hat{\theta}_k(n) - \eta \left[ -d(n) z_k(n) + \sum_{j=1}^p \hat{\theta}_j(n) u_j(n) z_k(n) \right]$$

$$\hat{\theta}_k(n+1) = \hat{\theta}_k(n) + \eta \left[ d(n) - \sum_{j=1}^p \hat{\theta}_j(n) u_j(n) \right] z_k(n)$$

$$\hat{\theta}_k(n+1) = \hat{\theta}_k(n) + \eta [d(n) - y(n)] z_k(n) \quad (27)$$

Con esto, si se trata de una neurona, el algoritmo de cálculo con variables instrumentales es directo, puesto que lo único que se requiere es realizar la siguiente modificación

$$\Delta \hat{\theta}_{ji}(n) = \eta \delta_j(n) z_i(n) \quad (28)$$

donde  $z_i$  son los elementos del vector de entrada de la neurona que utiliza la modificación de las  $VI$ .

Si existe una red de neuronas cuyos datos de entrada se conectan directamente con una capa de salida, se puede utilizar la misma forma de ajuste de las neuronas puesto que se debe considerar el error específico de cada neurona en comparación con la señal deseada para dicha salida.

En el caso de una red con neuronas ocultas, ocurre algo similar a lo indicado en la sección (6.2). El algoritmo no presenta diferencias respecto al de gradiente descendente, los pesos se ajustan de forma similar.

Al llegar a ajustar los pesos de ponderación de la capa de entrada, se encuentra la diferencia entre los métodos.

$$\Delta \hat{\theta}_{ji}(n) = \eta \delta_j(n) u_i(n) \rightarrow \Delta \hat{\theta}_{ji}(n) = \eta \delta_j(n) z_i(n) \quad (29)$$

Con esto, la única modificación que se debe realizar al algoritmo de cálculo consiste en considerar variables instrumentadas para el ajuste de los pesos de las neuronas de entrada de la red.

Las condiciones que debe cumplir  $z$ , son las indicadas en las ecuaciones (11) y (12), puesto que, como mencionamos anteriormente, al trabajar con redes de función de activación lineal se llega al mismo problema que para el caso de regresión lineal. Son múltiples las formas en que se puede seleccionar el vector de  $VI$ . Una forma es reemplazar en el vector de entrada, las señales que son realimentadas del mismo proceso, por ejemplo las salidas retardadas, situación que se presenta cuando se quiere modelar un sistema dinámico. Las señales reemplazantes, son generadas por filtros lineales. En este trabajo, se utiliza la misma forma para obtener el vector de  $VI$ . Se utilizan señales generadas por un modelo lineal para reemplazar las señales del vector de entradas que sean señales del pirómetro retardadas en uno y dos periodos de muestreo. Sin embargo, se destaca que esta no es la única forma de elegir un vector de  $VI$ , por lo que este método no puede identificarse solamente como un filtro de señales, sino que es más amplio y éste es sólo un caso particular.

El modelo generador de las nuevas señales para el vector de entradas será llamado Modelo 1. Este modelo debe ser entrenado con el método tradicional de gradiente descendente y luego generar los datos con su propia realimentación para poder entrenar al Modelo 2, que es el que utilizará la variable instrumentada.

La figura 5 muestra la forma en que se generan los datos que serán utilizados para entrenar al Modelo 1.

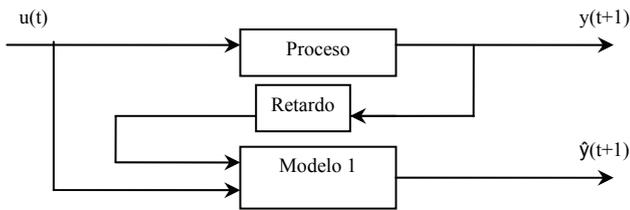


Figura 5. Esquema de datos para el entrenamiento del Modelo 1

Para ser consecuentes con el método de variables instrumentales, debemos utilizar el Modelo 1 para generar los datos  $\hat{y}(t)$  que se utilizarán en el entrenamiento del Modelo 2, pero estos deben generarse con el Modelo 1 realimentando su propia salida. Esto se grafica en la siguiente figura.

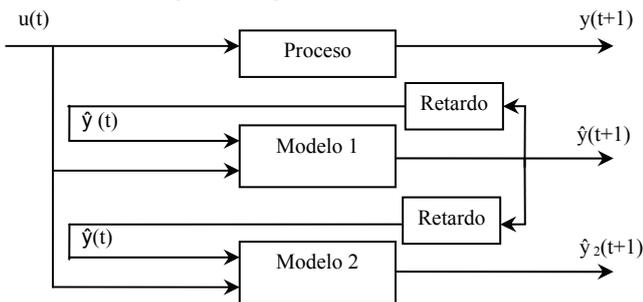


Figura 6. Esquema de datos para el entrenamiento del Modelo 2

### 8. IMPLEMENTACIÓN DEL SENSOR VIRTUAL

La estructura final de la red utilizada se determina por prueba y error. Una vez que se alcanza un resultado satisfactorio en base a criterios cuantitativos (error cuadrático) y cualitativos (seguimiento de la tendencia), se realiza un ajuste desechando las señales que no aportan mejoramiento al resultado de la red. Para la realización de las pruebas se utilizan dos modelos, el primero es entrenado de forma tradicional. El segundo modelo es entrenado con el algoritmo de *VI*, utilizando el primer modelo como variable instrumental para la medición de temperatura.

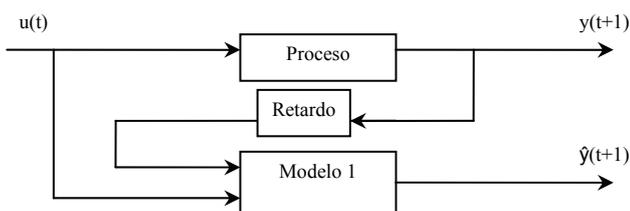


Figura 7. Esquema del primer modelo para el Sensor Virtual

En el Modelo 1 se buscan dos resultados. Por una parte, el modelo debe presentar el mejor ajuste posible a los datos de entrenamiento y validación, tanto cuando el pirómetro está disponible como cuando el *SV* funciona reemplazando totalmente al Pirómetro. El segundo resultado es que el Modelo 1 sea estable, para que cuando deba realimentar su propia salida, ésta última no se vuelva oscilante.

#### 8.1. Estructura de la Red Neuronal

El sensor virtual, implementado mediante redes neuronales, considera la utilización de señales de entrada y salida del

proceso. Las señales de entrada están consideradas dentro del vector denominado  $u(t)$ , que también considera señales de salida del proceso que se encuentran correlacionadas con la medición del pirómetro, y que sirven para realizar un mejor ajuste del modelo. La señal de salida para este modelo, es la medición de temperatura del baño de cobre entregada por el pirómetro Noranda, por lo que será denominada como  $y(t)$ .

El proceso de fusión-conversión que se desarrolla en el *CT* es de tipo dinámico, por lo tanto, la red neuronal también debe tener características dinámicas. Es por ello, que cada una de las señales que aparecen en la Tabla 1 se utilizan con tiempos de retardo. En este caso, se utilizan las señales en  $(t-1)$  y  $(t-2)$ , para estimar la medición  $y(t)$ . Esto responde a que en las pruebas realizadas en Planta se observa un comportamiento cercano a un sistema de primer orden, por lo tanto, utilizando dos periodos de retardo se asegura el poder rescatar la dinámica predominante.

Se utilizan redes de tipo multicapa con alimentación hacia adelante, para ser utilizada con algoritmos de entrenamiento de propagación hacia atrás.

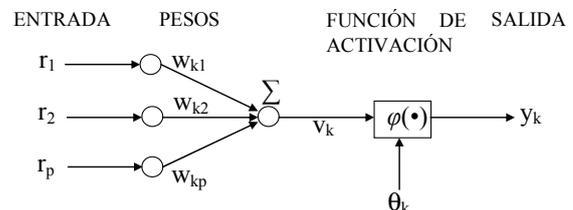


Figura 8. Modelo de Neurona

El modelo de neurona de la Figura 8, se encuentra descrito por las siguientes ecuaciones

$$v_k = \sum_{j=1}^p w_{kj} r_j \quad y_k = \varphi(v_k - \theta_k) \quad (30)$$

Las funciones de activación  $\varphi(\cdot)$  que se utilizan en este trabajo son lineales, de forma que sea aplicable la solución existente para los métodos de regresión lineal por medio de *VI*. Para mantener las ventajas de las características no lineales de las redes neuronales, se recurre a utilizar señales de entrada elevadas al cuadrado. Si no se obtienen resultados satisfactorios se pueden agregar términos elevados a las potencias siguientes.

#### 8.2. Preparación de datos

Los datos que se utilizan en estas simulaciones corresponden a 9 horas con 20 minutos, que son los resultantes de un conjunto mayor al que se aplican filtros para eliminar datos fuera de rango y periodos transientes de partida y detención de la planta. El periodo de muestreo es de 30 segundos, con lo que se cuenta con un total de 1122 muestras. Este conjunto de datos fue dividido para generar los datos de entrenamiento (6 horas) y los datos de validación (3 horas 20 minutos, equivalentes a 400 muestras). Debido a la naturaleza diversa de las variables, se utiliza la normalización de señales en el rango 0,1 – 0,9.

#### 8.3. Entrenamiento del Modelo 1

La primera alternativa, utiliza una red neuronal con realimentación de la señal de salida  $y(t-1)$  e  $y(t-2)$ . La red se incorpora de la siguiente forma

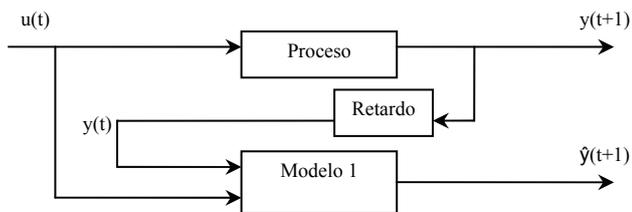


Figura 9. Modelo 1 de red con realimentación de señal de temperatura

En la figura 9 no aparecen mencionados en forma explícita los retardos aplicados a cada uno de los vectores, el detalle se encuentra en la tabla 3. La señal de salida y el vector de entradas están dados por

**Tabla 3. Configuración para el Vector de Entradas del Sensor Virtual Modelo 1.**

Vector Entrada	Vector Salida
TIT-319(t-1)	TIT-319(t)
TIT-319(t-2)	
FIT-318(t-1)	
FIT-318(t-2)	
FIT-323(t-1)	
FIT-323(t-2)	
FIT-329(t-1)	
FIT-329(t-2)	
FIT412(t-1)	
FIT412(t-2)	
PIT-101B(t-1)	
PIT-101B(t-2)	
TIT-101B(t-1)	
TIT-101B(t-2)	

Se debe recordar que por cada una de las señales del vector de entradas que aparece en la tabla 3, también se considera la señal elevada al cuadrado. Es decir, el vector de entradas está compuesto por 28 señales. Además, para todos los modelos realizados se considera *polarización* para cada neurona. El resultado final de la red del Modelo 1, utiliza la siguiente configuración.

**Tabla 4. Configuración de la red para el Modelo 1**

DESCRIPCION	VALOR PARÁMETRO
Neuronas de Entrada	28
Neuronas Capa Oculta	38
Neuronas de Salida	1
Función Activación Capa Oculta	lineal
Función Activación Capa Salida	lineal
Método de Entrenamiento	Gradiente descendente
Epochs	2000
Goal	0.001

Como algoritmo de entrenamiento se utiliza la rutina *traingd* de *MATLAB*, que corresponde al algoritmo tradicional de gradiente descendente.

Cuando el pirómetro no se encuentra disponible, la señal de temperatura entregada por el modelo debe ser realimentada al Modelo 1, por ello, es necesario que dicho modelo sea estable, lo que se verifica con las pruebas realizadas. Las gráficas de los resultados de las pruebas se entregan en la sección 9. La figura

siguiente muestra la realimentación que se efectúa cuando el pirómetro no está disponible y el Sensor Virtual debe reemplazar la medición.

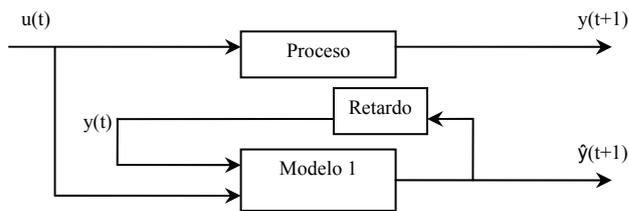


Figura 10. Modelo 1 funcionando como SV de Temperatura

8.4. Entrenamiento del Modelo 2

La segunda alternativa, utiliza una red neuronal entrenada con variables instrumentadas para la señal del Pirómetro Noranda  $\hat{y}(t-1)$  e  $\hat{y}(t-2)$ . La figura 11 muestra la forma en que esta red es entrenada.

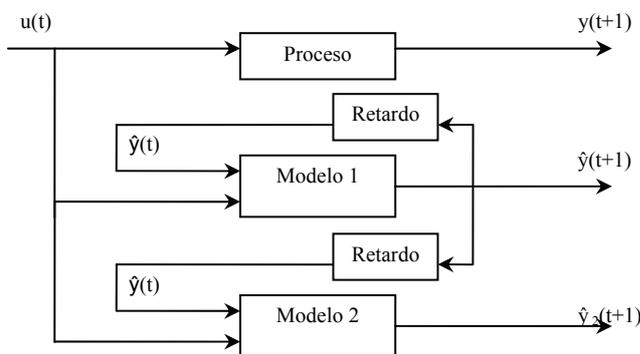


Figura 11. Entrenamiento para Modelo 2 con Variable Instrumentada para la Temperatura

El cuadro de señales utilizadas para el entrenamiento de la red se presenta en la tabla 5.

La señal de salida sigue estando dada por la temperatura real del Pirómetro, esto porque se requiere ajustar al comportamiento real del pirómetro, no al resultado entregado por el Modelo 1. El vector de entrada considera las modificaciones ya mencionadas, al introducir en lugar de mediciones pasadas del pirómetro, los datos entregados por el Modelo 1.

**Tabla 5. Configuración para el Vector de Entradas del Sensor Virtual Modelo 2.**

Vector Entrada	Vector Salida
TImod1(t-1)	TIT-319(t)
TImod1(t-2)	
FIT-318(t-1)	
FIT-318(t-2)	
FIT-323(t-1)	
FIT-323(t-2)	
FIT-329(t-1)	
FIT-329(t-2)	
FIT412(t-1)	
FIT412(t-2)	
PIT-101B(t-1)	
PIT-101B(t-2)	
TIT-101B(t-1)	
TIT-101B(t-2)	

En este caso también se agregan al vector de entrada cada una de las señales con sus valores elevados al cuadrado, y se debe considerar que cada una de las neuronas tiene polarización. La configuración final de la red del Modelo 2 es la siguiente.

**Tabla 6. Configuración de la red para el Modelo 2**

DESCRIPCION	VALOR PARÁMETRO
Neuronas de Entrada	28
Neuronas Capa Oculta	38
Neuronas de Salida	1
Función Activación Capa Oculta	Lineal
Función Activación Capa Salida	Lineal
Método de Entrenamiento	Gradiente descendente
Epochs	2000
Goal	0.001

Una vez entrenado el Modelo 2, es de interés realizar la comparación para las mismas situaciones que debe enfrentar el Modelo 1. Por lo tanto, se consideran los siguientes dos modos de funcionamiento. El primero es considerar que el pirómetro se encuentra disponible, para lo cual el modelo se puede utilizar como predictor de temperatura. El segundo caso, consiste en suponer que el pirómetro queda fuera de servicio, por lo que el modelo debe funcionar con realimentación, como se muestra en la figura siguiente.

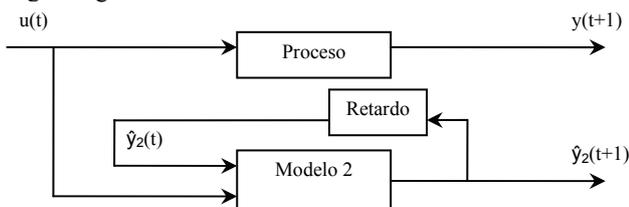


Figura 12. Modelo 2 funcionando como SV con realimentación de su salida

### 9. PRUEBAS Y ANÁLISIS DE RESULTADOS

En esta sección se realizan las pruebas a los modelos obtenidos en la sección anterior y se comparan los resultados utilizando el conjunto de datos de validación.

#### 9.1. Pruebas para el Modelo 1

La primera prueba se realiza sobre el Modelo 1, suponiendo que el pirómetro se encuentra disponible y alimenta la entrada del Sensor Virtual, lo que se representa gráficamente en la figura 13.

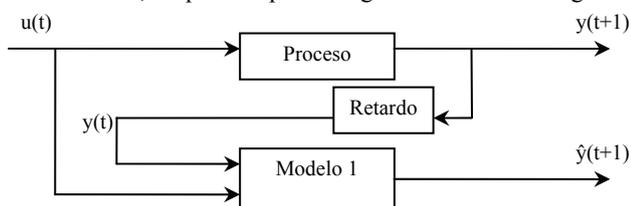


Figura 13. Modelo 1 funcionando como predictor de temperatura

La figura 14 muestra el ajuste del modelo a los datos de validación. Se aprecia que el Modelo 1 presenta un buen ajuste a los datos de validación, salvo algunos sectores en los cuales se estima que son dinámicas no modeladas. Los gráficos que se presentan arrojan los resultados para las 400 muestras que conforman el conjunto de datos de validación. El periodo de

muestreo es de 30 segundos, por lo que cada gráfico abarca un periodo de 3 horas y 20 minutos.

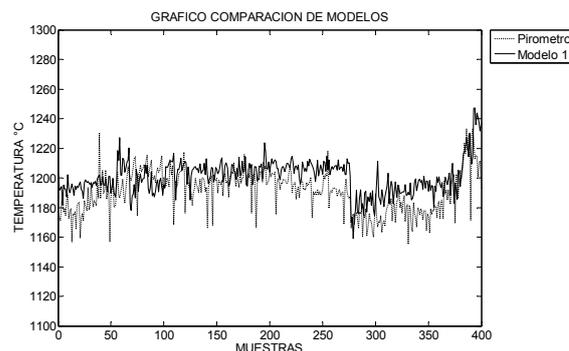


Figura 14. Ajuste del Modelo 1 a los datos de validación

La figura 16 presenta el funcionamiento del Sensor Virtual cuando el pirómetro no está disponible, es decir, cuando debe realimentar su propia salida, lo que se representa gráficamente en la figura 15. Se aprecia que el ajuste no es satisfactorio desde el punto de vista operacional, pese a ello, se debe destacar que el modelo sigue la tendencia del pirómetro, lo que es información relevante para la operación del CT considerando que el Pirómetro no se encuentra disponible.

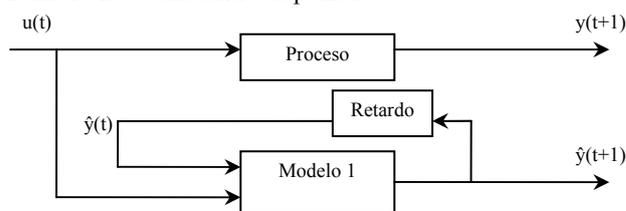


Figura 15. Modelo 1 funcionando como Sensor Virtual de temperatura

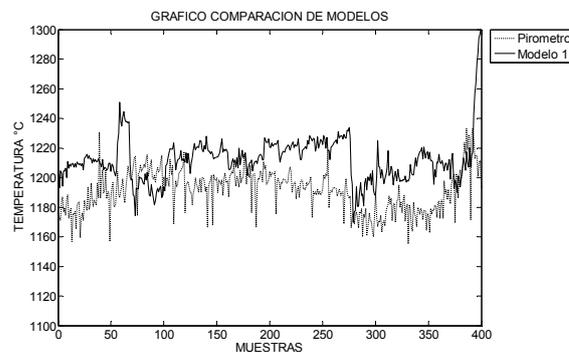


Figura 16. Estimación del Modelo 1 como SV de temperatura

#### 9.2. Pruebas para el Modelo 2

El Modelo 2 es entrenado con la señal del Modelo 1, pese a ello, la primera prueba considera que el pirómetro se encuentre disponible. En tal caso, existen dos alternativas, alimentar al Modelo 2 con la señal proveniente del Modelo 1, o bien, directamente desde el pirómetro. Para establecer una buena comparación entre ambos, el Modelo 2 se alimenta directamente desde la señal del Pirómetro, lo que se muestra gráficamente en la figura 17.

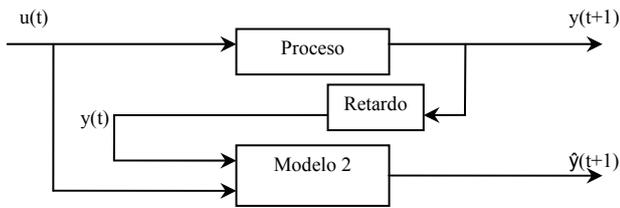


Figura 17. Modelo 2 como predictor de temperatura con señal desde pirómetro

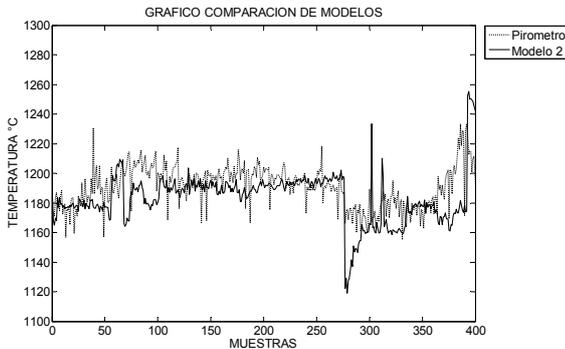


Figura 18. Modelo 2 alimentado con señal desde el Pirómetro

Ahora, suponiendo que el Pirómetro no se encuentra disponible, el Modelo 2 debe cumplir su función de Sensor Virtual, como se muestra en la figura 19.

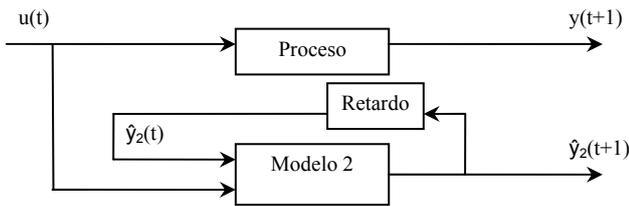


Figura 19. Modelo 2 como Sensor Virtual realimentando su propia salida

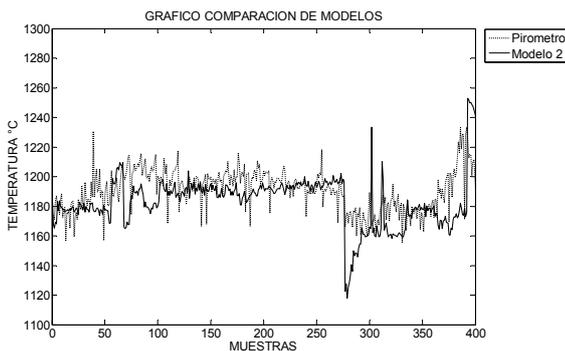


Figura 20. Modelo 2 como SV realimentando su propia señal de salida

9.3. Análisis de los gráficos

El Modelo 1 presenta un ajuste que se considera satisfactorio en cuanto a seguimiento de la tendencia (criterio cualitativo). Cuando el pirómetro se encuentra disponible el Modelo 1 entrega una señal de predicción de temperatura bastante ajustada a los datos reales entregados por el instrumento, por lo que se puede concluir que el Modelo 1 sirve para cumplir un rol de predictor de temperatura.

Cuando el pirómetro no se encuentra disponible, el Sensor Virtual entregado por el Modelo 1 presenta un ajuste desmejorado respecto al primer caso, en cuanto a seguimiento de tendencia y a error cuadrático. Esta señal a criterio del autor, no es confiable como para ser utilizada en reemplazo del pirómetro. Para el caso del Modelo 2, cuando el pirómetro está disponible y alimenta directamente la entrada del modelo, se nota un buen ajuste, pero respecto al primer modelo presenta una mayor variabilidad, por lo que se considera que el Modelo 1 es mejor que el Modelo 2 para este caso.

Cuando el pirómetro no se encuentra disponible, el Modelo 2 visualmente entrega un mejor ajuste que el primer modelo. Para tener una comparación objetiva se recurre al cálculo del error cuadrático. En las tablas siguientes se presenta una comparación del error cuadrático presentado por cada una de las soluciones.

Tabla 7. Comparación de Sumatoria de Error Cuadrático con Pirómetro Disponible

Con Pirómetro Disponible		Sin Pirómetro Disponible	
Modelo 1	Modelo 2	Modelo 1	Modelo 2
101000	137000	315000	138000

Con esto, se concluye que el Modelo 1 es la mejor alternativa para actuar como predictor de temperatura. Pero cuando el pirómetro no se encuentra disponible y se requiere un SV, el mejor ajuste lo entrega el segundo modelo.

10. CONCLUSIONES

El principal resultado de este trabajo es la demostración de que es posible aplicar el concepto de variables instrumentales a los algoritmos de entrenamiento de Redes Neuronales. Al realizar el desarrollo para redes lineales, se observa que la implementación es similar a la utilizada en los métodos de identificación tradicionales y consiste en crear un vector de entradas no correlacionado con el error. Una de las posibilidades de elegir el vector de VI es utilizar un filtro para las señales que se realimentan al mismo modelo, que es la forma utilizada en esta ocasión. Sin embargo, cabe destacar que no es la única forma de crear el vector de VI, por lo que el caso de la señal filtrada pasa a ser un caso particular de esta metodología.

Esta propuesta entrega resultados satisfactorios bajo criterios cualitativos y cuantitativos y permite, bajo determinadas condiciones, un mejor ajuste a la señal del pirómetro que el algoritmo tradicional de gradiente descendente. El criterio cuantitativo utilizado para discriminar cuál método es mejor, es el error cuadrático de la salida de la red respecto a la señal real.

Una de las limitaciones importantes es que el filtro utilizado para la creación de una señal de salida no correlacionada con el error, debe presentar un comportamiento estable al realimentar su propia salida. Esta es una condición para utilizar la metodología aquí presentada, puesto que de no ser así no sería posible generar datos para el entrenamiento del Modelo que utiliza las variables instrumentales.

Los modelos obtenidos tienen bajo error promedio respecto de la señal real. Las termocuplas con que se calibra el pirómetro tienen errores de +/- 5 °C. El caso del pirómetro puede ser un poco mayor en la práctica, ya sea por errores de calibración o por elementos que se encuentren con algún nivel de desgaste, de todas maneras la especificación indica también +/- 5 °C de error, por lo tanto se tiene un error acumulado de +/- 10 °C. Los modelos obtenidos están alrededor de los +/- 15 °C, es decir, un nivel de error absolutamente comparable.

La introducción de términos cuadráticos en la entrada de la red es fundamental para este caso, puesto que se trata de una red lineal en los parámetros, por lo tanto, al modelar sistemas no-lineales se debe tener alguna alternativa de incorporar no-linealidades al sistema.

La posibilidad de inferir la medición de temperatura dentro de un reactor es una alternativa que se ha presentado en el desarrollo de los modelos dinámicos de los procesos de los Convertidores. En estos modelos tradicionales, se requiere una gran cantidad de información que permita determinar en forma precisa el aporte de calor que realiza cada uno de los flujos que ingresan al CT. Para la Fundición de Potrerillos la posibilidad de obtener toda la información necesaria en línea es lejana, puesto que es una fundición maquiladora que trabaja mezclando concentrados provenientes de diversas fuentes, y en este momento, no se cuenta con un analizador en línea que entregue información confiable de la mezcla actual que está ingresando al Convertidor. El desarrollo de este Sensor Virtual demuestra que la estimación de temperatura no necesita toda la información requerida en estos modelos tradicionales, puesto que esta información puede ser entregada por otras señales de salida del proceso que se encuentren correlacionadas con los mismos datos de entrada que la medición de temperatura.

## 11. REFERENCIAS

- Arias H. (1995). *Diseño de sensores virtuales para uso industrial en una planta de flotación*. Tesis para optar al título de ingeniero civil electricista. Universidad de Chile. Santiago, Chile.
- Barrera R. (2007). *Desarrollo de un sensor virtual de ley de concentrado rougher en planta Las Tórtolas*. Tesis para optar al grado de magíster en ciencias de la ingeniería. Universidad de Chile. Santiago, Chile.
- Bustos M. (1999). *Desarrollo de un sensor virtual de concentración de sólidos en el rebose de hidrociclones en circuitos de molienda*. Tesis para optar al título de ingeniero civil de minas. Universidad de Chile. Santiago, Chile.
- Catipillán M. (2003). *Modelos metalúrgicos Fundición Potrerillos*. Tesis para optar al grado de Magíster en Ciencias de la Ingeniería. Universidad de Santiago de Chile. Santiago, Chile.
- Ceballos G. (2001). *Sensor virtual para el índice de trabajo de mineral sobre la base de su litología*. Tesis para optar al título de ingeniero civil electricista. Universidad de Chile. Santiago, Chile.
- Espinoza P. (1995). *Diseño de sensores virtuales mediante análisis de conglomerados*. Tesis para optar al título de ingeniero civil electricista. Universidad de Chile. Santiago, Chile.
- Freeman J.A. y D.M. Skapura (1993). *Redes Neuronales: Algoritmos, Aplicaciones y Técnicas de Programación*. Addison-Wesley.
- Haykin S. (1998). *Neural Networks: A Comprehensive Foundation*. Segunda edición. Prentice Hall.
- Ianiszewski I. (2007). *Análisis de la correlación existente entre los parámetros de un sistema con los pesos de una red neuronal artificial*. Tesis para optar al grado de Magíster en Ingeniería Electrónica. Universidad Técnica Federico Santa María. Valparaíso, Chile.
- Jain P., I. Rahman, B.D. Kulkarni (2007). Development of a Soft Sensor for a Batch Distillation Column Using Support Vector Regression Techniques. *Chemical Engineering Research and Design* **85**, pp. 283-287.
- Jordan H. (1998). *Sensor virtual de la concentración de sólidos en la descarga de hidrociclones*. Tesis para optar al título de ingeniero civil de minas. Universidad de Chile. Santiago, Chile.
- Ljung L. (1999). *System Identification - Theory For the User*. Segunda edición. PTR Prentice Hall. Upper Saddle River, N.J.
- Pasika H., S. Haykin, E. Clothiaux, R. Stewart (1998). Neural Networks for Sensor Fusion in Remote Sensing. *Neural Networks. International Joint Conference on* **4**, pp. 2772 – 2776.
- Rojas R. (2004). *Apuntes del curso: Modelos para Control*. Universidad Técnica Federico Santa María. Valparaíso, Chile.
- Solar M. (2000). *Manual de Proceso del Convertidor Teniente CT5*. “no publicado”
- Tapia H. (2000). *Plataforma de hardware y software para aplicaciones industriales de los sistemas de sensores virtuales*. Tesis para optar al título de ingeniero civil electricista. Universidad de Chile. Santiago, Chile.