

Diseño de un controlador adaptativo neuronal usando Filtro Kalman

Por: Diego Fernando Cano *

Introducción

Este documento tiene como objetivos presentar la aplicación de un algoritmo recursivo para el entrenamiento de redes neuronales basado en Filtro Kalman y aplicar una ley de control adaptativo neuronal a una planta no lineal; inicialmente, se habla de la técnica de control directo inverso, el cual está basado en la identificación del modelo inverso del sistema; una vez obtenido, se prosigue con el tema de la actualización de los pesos en las redes neuronales mediante el Filtro Kalman. Esta técnica permite obtener mejores resultados computacionales, en comparación con otros métodos de entrenamiento de estas y, finalmente, se concluye con la aplicación de la estructura de Control Adaptativo Neuronal por Modelo de Referencia¹ sobre el modelo de una caldera.

Resumen

El control, basado en modelo de referencia (MRAS), ha tenido gran acogida en los últimos años, especialmente en los campos de investigación e industria debido a su fácil implementación y capacidad para controlar diferentes procesos. Esta técnica no hace referencia a un método de diseño específico sino a una variedad de algoritmos que permiten la construcción de diversos controladores. Una de las características de esta técnica es el uso directo del modelo del proceso, el cual es utilizado para comparar la salida del sistema, con la del modelo, y así efectuar ajustes sobre los parámetros del controlador. Estos ajustes son conocidos como adaptación. Dada la existencia de diversas dinámicas dentro de los sistemas, se propone utilizar las redes neuronales como representación del sistema en consideración (caldera) y de la misma manera se utilizan para encontrar el modelo inverso del sistema, el cual cumplirá la función de controlador; por otra parte, el algoritmo de adaptación de parámetros del controlador corresponde al Filtro Kalman Extendido (EKF), que permite obtener mejores resultados computacionales en comparación con otros métodos.

Control Inverso Directo (DIC – Direct Inverse Control)

La técnica DIC, utiliza el modelo inverso de la planta como controlador. El funcionamiento del sistema en malla abierta, de este esquema, puede describirse así: el controlador se coloca en cascada con el proceso, de esta forma, al utilizar la referencia como entrada al modelo inverso se espera que la salida de éste sea la señal de control adecuada para la planta. Este esquema se ilustra en la figura 1.

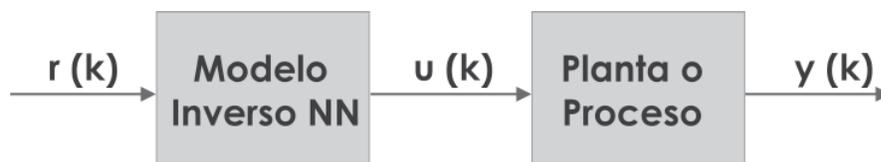


Figura 1. Esquema de Control Inverso Directo

Si denotamos el proceso como $P(k)$ y, de igual forma, denotamos el modelo inverso como $I(k)$, entonces se observa que la relación de entrada

* Docente programa de tecnología en Electrónica UNIMINUTO
Estos modelos se conocen como MRAS (Model Reference Adaptive System).

salida corresponde a:

$$y(k) = \frac{P(k)}{I(k)} \cdot r(k)$$

Esto nos lleva a las siguientes observaciones sobre el esquema:

- Si $I(k)$ es el modelo inverso exacto de $P(k)$, entonces la salida del sistema debe seguir a la referencia.
- Dado que no hay realimentación, no es posible determinar si el sistema sigue a la referencia.
- El controlador no es sensible a perturbaciones que ocurran sobre la planta.
- Si el modelo inverso corresponde a una red neuronal, se deben tener en cuenta las limitaciones de generalización de estas estructuras.

Filtro Kalman Extendido (EKF) y el entrenamiento de las redes neuronales

El Filtro Kalman es un conjunto de ecuaciones matemáticas que proveen un medio recursivo computacional eficiente para estimar el estado de un proceso, tomando como base la minimización del error cuadrado medio, el algoritmo básico del EKF puede resumirse como:

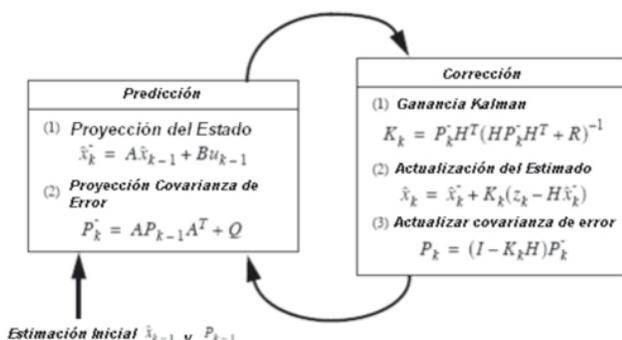


Figura 2. Algoritmo Filtro Kalman

Donde:

\hat{x}_k^- = Estados futuros estimados

\hat{x}_k = Estados futuros

P_k^- = Co varianza error estimada

P_k = Co varianza error

H = Relación Estados / Salidas Actuales

z_k = mediciones

K_k = Ganancia Kalman

Algunas de las ventajas que tiene el Filtro Kalman sobre otros métodos de entrenamiento radican en los cálculos a hacerse, que no son complejos. El modelo que se pretende minimizar está siendo linealizado en el punto de operación donde se encuentre y, además, como se está monitoreando todo el tiempo la evolución del proceso, es posible hacer correcciones sobre las predicciones realizadas.

Partiendo de esta base podemos representar un sistema no lineal como:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\ z_k &= h(x_k, v_k) \end{aligned}$$

Donde:

x_k = Estados futuros

x_{k-1} = Estados Actuales

u_{k-1} = Entradas Actuales

w_{k-1}, v_k = Ruido

z_k = Salidas

De acuerdo con lo anterior, una red neuronal puede ser modelada como se muestra a continuación:

$$\begin{aligned} x_k &= x_{k-1} \\ z_k &= \phi(x_k, u_k) \end{aligned}$$

Donde:

x_k = Pesos futuros

x_{k-1} = Pesos Actuales

u_{k-1} = Entradas Actuales

z_k = Salida

$\phi(\cdot)$ = Función de la red

Así, se observa que la estimación de los pesos futuros, corresponde a los pesos actuales y la salida de la red es una función de los pesos y las entradas actuales de la red.

El algoritmo para la actualización de los pesos de la red neuronal consiste en:

1. Determinar el vector de pesos (en este documento se le llama w).
2. Determinar la matriz H , que corresponde al gra-

- diente de la salida de la red con respecto a los pesos.
3. Realizar los siguientes cálculos:
 - a. Calcular R (escalar)
 - b. Calcular la ganancia Kalman (K)
 - c. Actualizar el vector de pesos W
 - d. Estimar la covarianza del error.
 4. Regresar al paso 1.

Diseño de controladores adaptativos neuronales

Los controladores adaptativos son aquellos que están diseñados para "adaptar" sus parámetros basándose en un método de ajuste, en el caso de los controladores neuronales, los parámetros a ajustar corresponden a los pesos de la red neuronal².

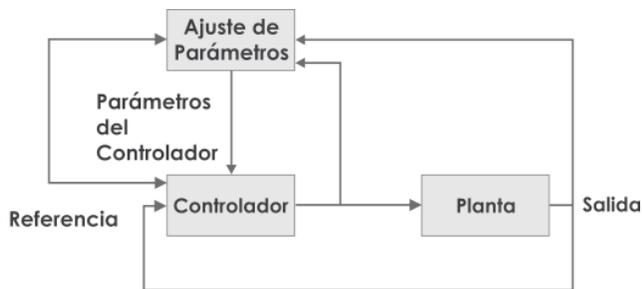


Figura 3. Sistema adaptativo

Para la aplicación de una ley de control adaptativo neuronal es necesario obtener el modelo inverso de la planta (se requiere dado que va a ser aplicado sobre un esquema de control directo); para esto, se lleva a cabo un proceso de identificación del sistema el cual puede hacerse de forma ON-LINE u OFF-LINE. Sin embargo, es deseable poseer un modelo inverso previamente sintonizado (identificación OFF-LINE) ya que, al implementar una red con parámetros (pesos) aleatorios, podemos hacer que nuestro sistema salga de la región de operación deseada³. En términos generales, el procedimiento para la identificación del modelo inverso puede resumirse como:

1. Definición de la estructura neuronal a utilizar; es decir, cuál va a ser el conjunto de entradas y salidas del modelo. En la práctica el modelo más utilizado corresponde al modelo NNARX, que es una extensión del modelo auto-regresivo ARX.
2. Generar el conjunto de señales para la identificación del sistema, éstas deben garantizar la excitación del sistema tanto a bajas como a altas frecuencias.
3. Una vez obtenidos los datos entrada-salida, se en-

trena la red neuronal teniendo en cuenta que la salida de ésta será la señal de control para la planta.

Una vez obtenido el modelo inverso preliminar, se define el método de actualización de pesos ON-LINE a utilizar entre los cuales se encuentran back-propagation ON-LINE, Gauss-Newton Recursivo, el método recursivo de predicción de error verdadero, entre otros⁴; en este documento se elabora el algoritmo recursivo utilizando Filtro Kalman.

Modelo de la caldera⁵

La figura 4 ilustra un modelo disponible simplificado en simulink de una caldera de vapor, de uso común en la industria en todo proceso que requiera calor. Todos los datos están en P.U. (1 significa que trabaja en el valor nominal de diseño, 100%). Los parámetros de la caldera son:

(CV) Válvula de control de turbina (0-2.0). Determina la cantidad de vapor a extraer de la caldera.

(Q) Agua, Aire, Combustible (0-2). Representa el accionamiento, la adición de calor. La relación entre estos tres elementos se ajusta automáticamente.

(PT) Presión en el tambor. Variable a controlar. El valor de 1 p.u. es el valor deseado a mantener.

(RV) Rata de flujo de vapor. Vapor que sale de la caldera, depende de la demanda de vapor del proceso.

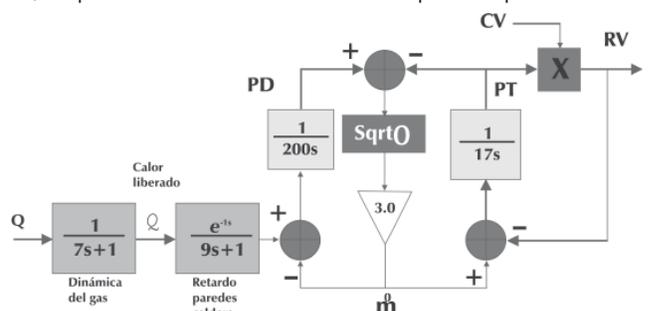


Figura 4. Modelo de la caldera

Aplicación de la ley de control al modelo

En esta sección se presentan varias de las pruebas realizadas bajo diferentes parámetros de simulación.

2.Referencia:Åström, Karl J, Björn Wittenmark, Adaptive Control , 1995.
 3.Referencia: M. Norgaard, O. Ravn, N. K. Poulsen y L. K. Hansen. Neural Networks for Modelling and Control of Dynamic Systems. Springer. 2003
 4.NOTA: Estos algoritmos son mencionados en M. Norgaard, O. Ravn, N. K. Poulsen y L. K. Hansen. Neural Networks for Modelling and Control of Dynamic Systems. Springer. 2003 , sin embargo, el algoritmo con el cual se trabaja en este documento corresponde a la actualización con Filtro Kalman.
 5. Tomado de: Curso Control por Computador. Universidad de los Andes. Segundo Semestre. 2005.

Prueba 1

Cambio del valor de CV en $t=5000$ a $CV=0.5$. No existe ruido.

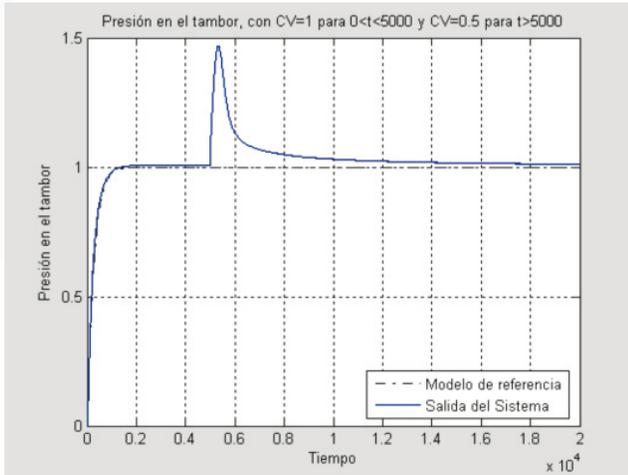


Figura 5. Respuesta del sistema frente a un cambio en CV

El sistema sigue al modelo de referencia hasta $t=5000$, lo cual se esperaba dado que el modelo inverso fue construido con las características que tiene la caldera hasta este tiempo; en $t=5000$ ocurre un cambio en CV, lo cual hace que el controlador deba ajustarse al nuevo modelo, proporcionando una nueva señal de control que consigue llevar el sistema a la referencia nuevamente.

En la figura 6 se muestra la evolución de la matriz de ganancia K, con el fin de ilustrar la adaptación del controlador.

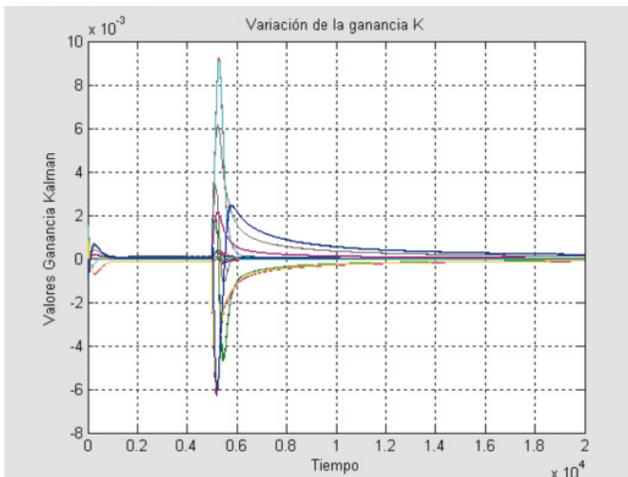


Figura 6. Variación de los valores de la ganancia Kalman

Prueba 2

El sistema es sometido a las siguientes situaciones:

- Cambio de referencia de 1 a 1.5 en $t=12000$.
- Cambio en el parámetro CV de la planta en $t=3000$. El valor de CV después de $t=3000$ es cero.
- Ruido gaussiano con media cero y varianza 0.1 en la salida del sistema después de $t=10000$.

Estas condiciones sacan al sistema de las condiciones ideales, y permiten evaluar el desempeño del controlador neuronal ya que éste debe adaptarse para seguir no sólo las referencias sino también identificar el cambio del modelo de la planta y filtrar el ruido, para no considerarlo como un error de modelo. La respuesta de la caldera se muestra en la figura 7.

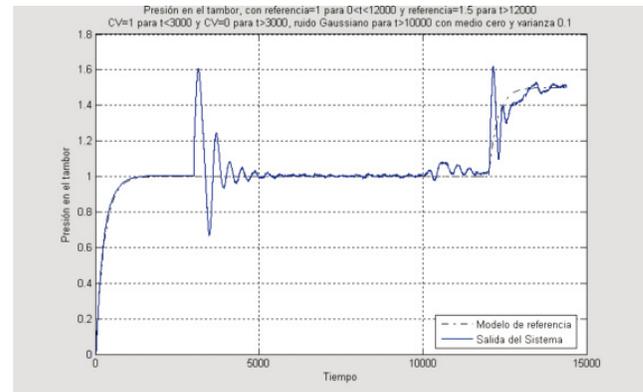


Figura 7. Respuesta del sistema en presencia de cambio de modelo y ruido.

El controlador se adapta permitiendo llevar al sistema a la referencia deseada; sin embargo, se aprecia que al cambiar el modelo de la planta y al agregar la señal de ruido la respuesta del sistema empieza a oscilar alrededor de la referencia. Este comportamiento se debe a la adaptación que presenta el controlador en presencia del ruido; este ruido es considerado como error de modelo. La variación en los parámetros de la matriz K para el ajuste del controlador se muestra en la figura 8.

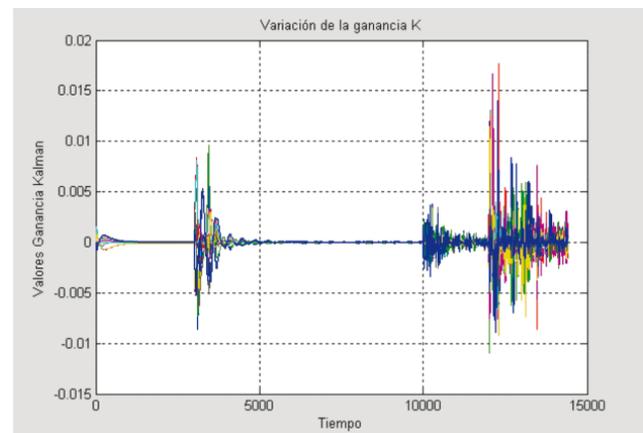


Figura 8. Variación de la Matriz K con cambio de modelo y en presencia de ruido

Prueba 3

La figura 9 exhibe el comportamiento del sistema en tres regiones de operación diferentes (1, 2 y 1,5), se aprecia que en el momento de hacer el cambio de modelo ($t=5000$) el sistema reacciona bruscamente pero luego logra estabilizarse, de acuerdo con la adaptación del controlador; posteriormente, con la presencia de ruido, el controlador intenta compensar este comportamiento y hace que el sistema empiece a presentar oscilaciones alrededor del punto de operación. Al cambiar el valor de la referencia, el controlador se adapta y lleva al sistema hacia la referencia deseada eliminando las oscilaciones producidas por la presencia del ruido; la respuesta del sistema se mantiene en un rango de error de $\pm 2\%$.

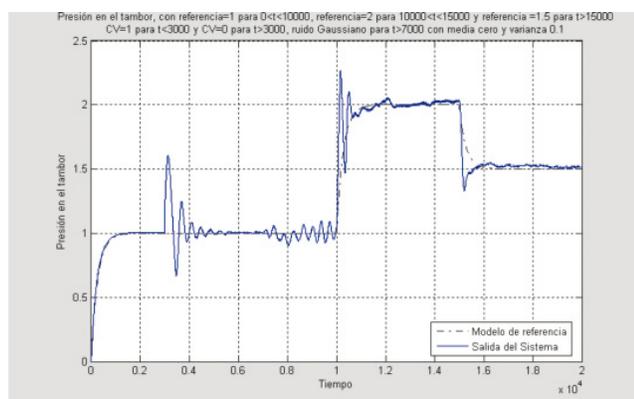


Figura 9. Respuesta del sistema bajo cambio de modelo, referencia y en presencia de ruido

Estos resultados muestran la fortaleza de los controladores adaptativos neuronales y en especial el método de actualización de pesos por medio del Filtro Kalman.

Conclusiones

De acuerdo a los resultados obtenidos se mostró que los controladores adaptativos neuronales son una herramienta de control útil cuando se tienen sistemas con parámetros variables, ya que el entrenamiento ON-LINE de la red neuronal permite identificar los cambios en los modelos y así ajustar la ley de control para el sistema; sin embargo, una de las debilidades de estos radica en la identificación de ruido, la cual hace que el controlador se adapte de acuerdo a este y haga que la planta salga del punto de referencia deseado. Un método para reducir este efecto consistiría en filtrar la señal de salida, antes de alimentar el controlador con la señal de la planta.

En cuando a las redes neuronales se sabe que permiten obtener modelos de sistemas en los cuales

sólo se cuenta con un conjunto de datos de entrada y de salida, y en casos especiales (cuando se usan funciones de activación $y=x$ o "línea pura" en las 2 capas la red), el modelo obtenido corresponde a un modelo afín al sistema. Por otra parte, en cuanto al entrenamiento, el Filtro Kalman reduce el costo computacional del cálculo de Jacobianos y Hessianas, reflejándose en un menor tiempo de cálculos y poniéndolo en ventaja, frente a algoritmos de entrenamiento como Back-propagation y Gauss-Newton.

Referencias

Åström, Karl J, Björn Wittenmark. 1995. Adaptive Control.

Cano, Diego Fernando. 2005. Diseño de Controladores Predictivos Neuronales. Universidad de los Andes.

Isermann, R., Lachmann, K.-H, Matko, D.. 1992. Adaptive Control Systems

Haykin, Simon. 1999. Neural Networks A comprehensive foundation.

Nguyen Hung, Prasad Nadipuram, Walker Carol, Walker Elbert. 2003. A first course in fuzzy and neural control.

Norgaard, M., Ravn, O. Poulsen, N. K. y Hansen, L. K. 2003. Neural Networks for Modelling and Control of Dynamic Systems. Springer.

- www.ici.ro/ici/revista/sic2003_4/art4.pdf
- alumnus.caltech.edu/~amir/control.pdf
- www.erudit.de/erudit/events/esit99/12598_p.pdf
- fluid.power.net/techbriefs/hanghzau/3_2.pdf
- <http://med.ee.nd.edu/MED7/med99/papers/MED026.pdf>
- www.eunite.org/eunite/events/eunite2001/look_back/13373_P_Isermann.pdf
- www.cdén.ca/2005/2ndCDEN-conference/data/10143.pdf
- www.cs.ele.tue.nl/MLazar/MATCOMpaper.pdf
- www.ee.unimelb.edu.au/conferences/ascc2004/proceedings/papers/P99.pdf
- db.s2.chalmers.se/download/publications/ngia_451.pdf
- www.esa.espci.fr/ARTICLES/1998kalm.pdf
- www.image.ece.ntua.gr/papers/197.pdf
- www.ias.ac.in/sadhana/Pdf2000Apr/Pe901.pdf
- <http://matwbn.icm.edu.pl/ksiazki/amc/amc16/amc1627.pdf>
- lorien.ncl.ac.uk/ming/robust/imc.pdf
- <http://mocha-java.uccs.edu/dossier/RESEARCH/1998thesis-.pdf>

- <http://mocha-java.uccs.edu/dossier/RESEARCH/1998thesis.html>
- <http://www.che.utexas.edu/~john/research/index.html>
- <http://www.engin.umich.edu/group/ctm/examples/ball/ball.html>
- matwbn.icm.edu.pl/ksiazki/amc/amc12/amc12210.pdf
- <http://zone.ni.com/devzone/cda/tut/p/id/4321>
- <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>
- <http://citeseer.ist.psu.edu/birgmeier94neural.html>