



UNIVERSIDAD DE CHILE

UNIVERSIDAD DE CHILE

EL7012

CONTROL INTELIGENTE DE SISTEMAS, OTOÑO

Ejercicio N°1

Autor:

Elias Obrequé
Gustavo Ceballo
Maibeth Sánchez

15 de mayo de 2020

Índice

1. Introducción	1
1.1. Problema 1	1
2. Generación de Datos	3
3. Modelos de predicción	5
3.1. Modelo lineal	5
3.2. Modelo difuso Takagi-Sugeno Tipo-1	6
3.3. Modelo de red neuronal	15
4. Resultados	26
5. Conclusión	27

1. Introducción

La mayoría de los sistemas tienen un comportamiento no lineal, excepto en un determinado rango de operación donde pueden ser considerados lineales. En ocasiones un modelo lineal es insuficiente para explicar un fenómeno por lo que se debe recurrir a Modelos No Lineales.

Los modelos basados en redes neuronales y sistemas difusos han sido empleados para la identificación, dado que han mostrado una grancapacidad para aproximar funciones no lineales desconocidas, además de que ofrecen una estructura general tan compleja o sencilla como el problema lo demande. Por un lado, las redes neuronales son estructuras matemáticas inspiradas en las estructuras biológicas compuestas por neuronas, en donde a través de elementos relativamente simples (llamados neuronas) que efectúan operaciones y procesos muy sencillos logran, mediante interconexiones, realizar procesos complejos y procesamiento de información en paralelo. En general se consideran redes neuronales conformadas por “capas” de neuronas, donde una capa procesa la información recibida en paralelo y la envía a la siguiente capa de neuronas. Por otro lado, los sistemas difusos son también estructuras matemáticas, aunque estos se encuentran inspirados en los procesos de pensamiento y toma de decisiones llevados a cabo por los humanos, así como en la teoría de conjuntos difusos propuesta por Lofti Zadeh.

En este ejercicio se dará solución al Problema 1.

1.1. Problema 1

Considere la siguiente serie no lineal dinámica:

$$\begin{aligned} y(k) = & (0,8 - 0,5\exp\{-y^2(k-1)\})y(k-1) \\ & - (0,3 + 0,9\exp\{-y^2(k-1)\})y(k-2) \\ & + u(k-1) + 0,2u(k-2) + 0,1u(k-1)u(k-2) + e(k) \end{aligned} \quad (1)$$

donde el ruido del sistema

$$e(k) = 0,5\exp\{-y^2(k-1)\}\beta(k) \quad (2)$$

depende del estado previo de la salida del modelo, y $\beta(k)$ es un ruido blanco.

Como usted sabe existen varias técnicas que se pueden emplear para la modelación a partir de estos datos, por lo que debe seleccionar el tipo de modelo más adecua-

do para este tipo de sistema. Para este trabajo se le pide detallar la metodología utilizada para:

- a) Generar 600 datos a partir de esta serie. Considere 55 % para entrenamiento, 25 % test y 20 % validación.
- b) Obtener un modelo de predicción lineal, difuso tipo-1 (T&S) y neuronal para la salida. Evaluar las predicciones a 1, 8 y 16 pasos. Comparar el desempeño de todos los modelos a partir de las métricas más apropiadas tales como RMSE, MAPE, MAE, entre otras. Comente.
- c) Construir el intervalo de predicción de los modelos obtenidos en b) utilizando el método de la covarianza.
- d) Evaluar los intervalos de predicción obtenidos en b) realizando predicciones a 1, 8, y 16 pasos. Comparar el desempeño de los modelos a partir de las métricas más apropiadas tales como ancho del intervalo, probabilidad de cobertura, entre otras.
- e) Construir el intervalo de predicción del modelo difuso encontrado en a) con el método de optimización min-max. Compare este intervalo de predicción con el intervalo obtenido utilizando el método de la covarianza. Comente.
- f) Construir el intervalo de predicción neural utilizando el método de Joint Supervision. Compare con los métodos anteriores.
- g) Seleccione el modelo más apropiado y justifique.

2. Generación de Datos

En esta etapa es necesario generar datos que representen la dinámica del sistema en la mayor cantidad de rangos de operación posibles, ya que el modelo obtenido tiene un ancho de banda acotado, y por lo tanto las dinámicas definidas por fuera de dicha banda podrían no ser representadas adecuadamente. Para lo cual se debe diseñar una entrada $u(k)$ que excite a la planta en el rango de frecuencias en que se encuentran los fenómenos de interés.

En este trabajo se propone el uso de señales binarias pseudo aleatorias (Pseudo Random Binary Signal, PRBS), ya que es una de las señales más utilizadas en identificación de sistemas. Esta es una señal periódica, determinística y que posee principalmente propiedades similares al ruido blanco (contenido muy rico en frecuencias)

Para generar la señal se suponen los siguientes parámetros de interés $f_{min} = 0,2Hz$, $f_{max} = 1Hz$ y tiempo de muestreo $T_s = 0,01$. Con los parámetros anteriores, y utilizando la expresión

$$n = \frac{\log(f_c/f_{min} + 1)}{\log(2)} \quad (3)$$

con $f_c = 2,5 * f_{max} = 2,5Hz$, se genera una PRBS de orden $n = 4$, por lo que el largo máximo corresponde a $N = 2^n - 1 = 15$. A su vez, la cantidad de muestras por bit son $N_s = 40$. Luego, el tiempo de un bit, $\Delta t = N_s * T_s = 0,4s$, por lo que la PRBS dura en total 6s y debe ser replicada 400 veces con diferentes condiciones iniciales para obtener los 6000 datos de interés. Finalmente se genera la APRBS variando la amplitud aleatoriamente de la PRBS generada, Fig.1 y se aplica a la serie no lineal como se muestra en la Fig.2.

Una vez obtenidos los datos experimentales de entrada-salida, éstos son clasificados en tres conjuntos con distinta información: datos de entrenamiento, datos de validación y datos de prueba; esto con el fin de evaluar adecuadamente los modelos generados. El conjunto de entrenamiento se utiliza para determinar los parámetros del modelo. El conjunto de prueba permite comparar distintas estructuras de los modelos generados. Finalmente, el conjunto de validación permite verificar el sobreajuste del modelo óptimo obtenido, evaluándolo en un nuevo conjunto de datos (distintos a los datos del conjunto de entrenamiento y validación), analizando su capacidad de generalización. En este caso se utiliza una división de 55 % de los datos para entrenamiento, 25 % para prueba y 20 % validación.

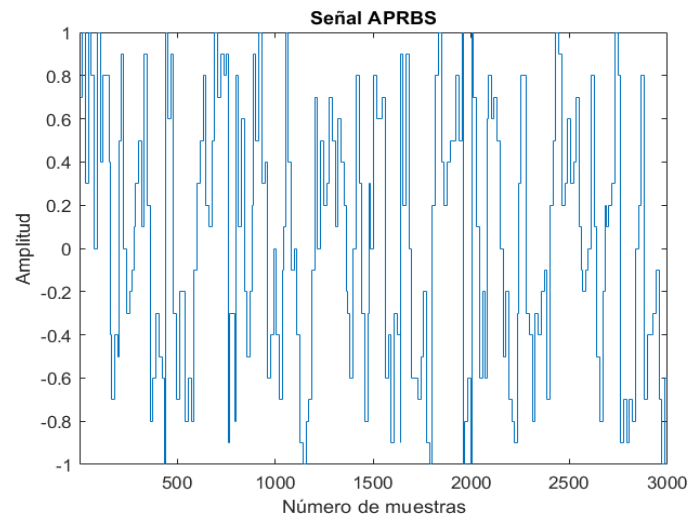


Figura 1: Señal APRBS con Amplitud entre -1 y +1.

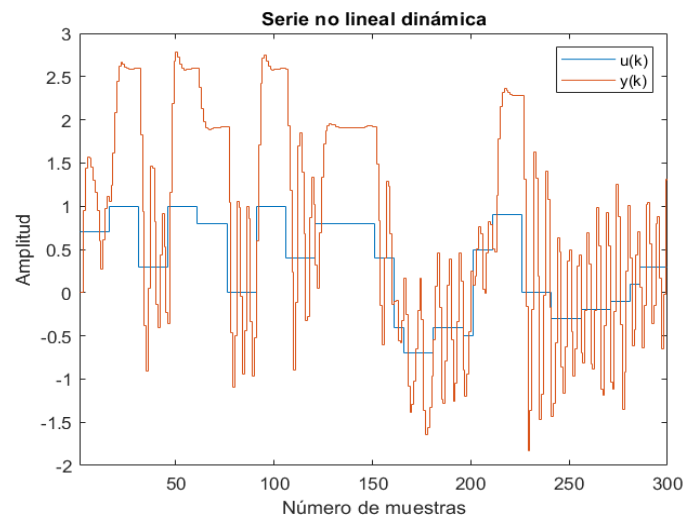


Figura 2: Respuesta de la serie no lineal.

3. Modelos de predicción

3.1. Modelo lineal

En este caso, supondremos que se ajustará un modelo lineal suponiendo que el sistema real es lineal con ruido blanco gaussiano aditivo, es decir,

$$y(k) = a_1y(k-1) + a_2y(k-2) + b_1u(k-1) + b_2u(k-2) + e(k) \quad (4)$$

Luego, se propone un modelo lineal para llevar a cabo la predicción a 1 paso, de modo tal que:

Predicción a 1 paso:

$$\hat{y}(k) = \hat{a}_1y(k-1) + \hat{a}_2y(k-2) + \hat{b}_1u(k-1) + \hat{b}_2u(k-2) \quad (5)$$

Este modelo no considera un valor constante o bias dado el supuesto que el sistema es lineal con ruido blanco aditivo. En caso que se sospechara que existe un bias o tendencia (trend) en el sistema, se puede agregar otro vector de unos a la matriz de regresores (o matriz de información).

Para llevar a cabo la estimación de los parámetros del modelo se utilizó la técnica de mínimos cuadrados, es decir:

$$\hat{\theta} = (Xent^T * Xent)^{-1} * Xent^T * \hat{y}(k) \quad (6)$$

En que $\hat{\theta} = [\hat{a}_1 \quad \hat{a}_2 \quad \hat{b}_1 \quad \hat{b}_2]^T$ es el vector de parámetros y $Xent$ es la matriz de regresores con los valores de las n muestras ordenados por filas.

Los valores que se obtuvieron de los parámetros fueron los siguientes:

$$\hat{\theta} = \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} 0,8601 \\ -0,6930 \\ 0,9724 \\ 0,3486 \end{pmatrix} \quad (7)$$

A continuación, en la Tabla 1, se presentan las métricas de bondad del ajuste o errores en los diversos conjuntos de datos, a saber, conjunto de datos de entrenamiento, prueba o test y validación.

Tabla 1: Errores o Métricas de bondad de ajuste a 1 paso

Métricas	Conjunto En- trenamiento	Conjunto de Prueba	Conjunto de Valida- ción
RMSE	0.0115	0.019	0.0241
MAPE	123.1323	101.9383	169.4761
MAE	0.3313	0.3402	0.3524

3.2. Modelo difuso Takagi-Sugeno Tipo-1

Los modelos difusos de Takagi-Sugeno son estructuras basadas en la lógica difusa que permiten representar procesos con dinámicas no lineales mediante la combinación de información otorgada por modelos locales. Estos tipos de modelos pueden ser expresados a partir de una base de reglas del tipo “Si-Entonces”, de la forma

$$R_r : \text{Si } z_1(k) \text{ es } MF_1^r \text{ y...y } z_p(k) \text{ es } MF_p^r \text{ entonces } y_r(z(k)) = f_r(z(k)) \quad (8)$$

donde R_r denota la r -ésima regla del modelo difuso, con $r \in 1, \dots, N_r$ y N_r el número total de reglas; $y_r(z(k))$ es su consecuencia o modelo local; $z(k) = [z_1(k), \dots, z_p(k)]$ es el vector de premisas en el tiempo k , las cuales por lo general son regresores de la entrada y/o salida del sistema; $f_r(z(k))$ es una función de las premisas del modelo; y MF_i^r es el conjunto difuso (función de pertenencia) de la i -ésima premisa correspondientes a la r -ésima regla.

Sea $\mu_r(z_i(k))$ el grado de pertenencia de la i -ésima premisa $z_i(k)$ al conjunto difuso MF_i^r , donde $\mu_r(z_i(k)) \in [0, 1]$, siendo 0 cuando la premisa no pertenece en ningún grado al conjunto MF_i^r , y siendo 1 si pertenece completamente a dicho conjunto. Luego, se define el grado de activación de la r -ésima regla, $w_r(z(k))$, como

$$w_r(z(k)) = \text{oper}(\mu_r(z_1(k)), \dots, \mu_r(z_p(k))) \quad (9)$$

donde $\text{oper}(\cdot)$ puede ser el operador mínimo o el producto. Se denota $h_r(z(k))$ al grado de activación normalizado de la r -ésima regla, es decir,

$$h_r(z(k)) = \frac{w_r(z(k))}{\sum_{l=1}^{N_r} w_l(z(k))} \quad (10)$$

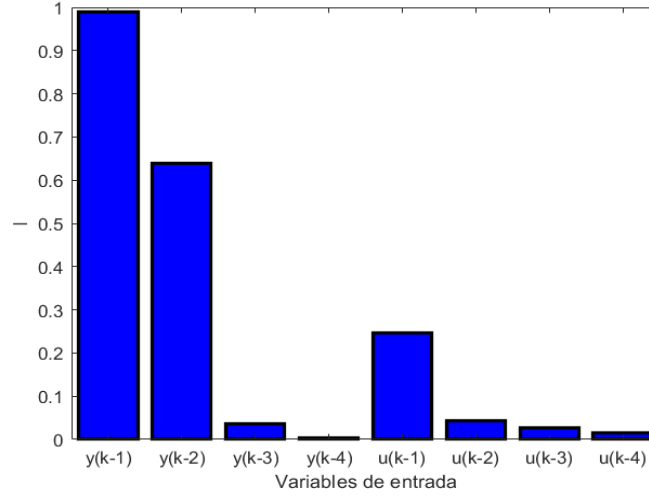


Figura 3: Índice de Sensibilidades.

Ya definido el grado de activación de cada regla, la salida del modelo difuso, $y_{fuzzy}(k)$, está dada por una suma ponderada de cada modelo local por su grado de activación normalizado, de la forma

$$y_{fuzzy}(k) = \sum_{r=1}^{N_r} h_r(z(k)) * y_r(z(k)) \quad (11)$$

Los modelos difusos TS son una clase de sistemas no lineales, cuya formulación requiere definir una serie de variables que no se conocen a priori, por lo que una estructura adecuada para representar un sistema es desconocida y, como consecuencia, un proceso de identificación debe ser llevado a cabo para determinar la estructura y cada uno de los parámetros del modelo.

- **Selección de variables:** Para seleccionar las variables que actúan como entrada al sistema difuso, se realiza un análisis de sensibilidad. Suponiendo una estructura del modelo inicial difuso con 8 variables de entrada $y(k-1), \dots, y(k-4), u(k-1), \dots, u(k-4)$. En la Fig. 3 se muestran los índices de las sensibilidades del modelo inicial para las 8 variables de entrada, comprobándose que las variables $y(k-4)$, $u(k-3)$ y $u(k-4)$ presentan menores índices de las sensibilidades, por lo cual no son incluidos en el modelo difuso. A pesar de que los regresores $y(k-3)$ y $u(k-2)$ presentan una sensibilidad similar se decidió escoger $u(k-1)$ ya que el mismo es un parámetro de la serie no lineal dinámica.

La Tabla 2 indica el valor de la Raíz del Error Cuadrático Medio (RMSE) para los modelos con 8 y 4 regresores, y se puede notar que son muy semejantes, y por consiguiente se selecciona el modelo más sencillo, que coincide con el número de regresores de la serie no lineal original.

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - y_{fuzzy}(k))^2} \quad (12)$$

donde N es la cantidad total de datos, $y(k)$ es la salida de la planta real en el instante k , e $y_{fuzzy}(k)$ es la predicción realizada por el modelo difuso en el instante k .

Tabla 2: Índices de Error para el Análisis de Sensibilidades.

Modelo	Variables de entrada	RMSE
1	$y(k-1), y(k-2), y(k-3), y(k-4),$ $u(k-1), u(k-2), u(k-3), u(k-4)$	0.2177
2	$y(k-1), y(k-2), u(k-1), u(k-2)$	0.2109

- **Optimización de la estructura:** La optimización de la estructura del modelo difuso consiste principalmente en determinar el número óptimo de reglas del modelo difuso. En este caso se definió un número máximo de 20 clusters y se entrenó el modelo para cada uno de los posibles valores de clusters, utilizando como algoritmo de clustering el Fuzzy C-Means.

La Fig. 4 muestra el RMSE para los conjuntos de entrenamiento y prueba. Si no importa la complejidad, el mejor modelo es aquel que tiene menor RMSE. Sin embargo, es posible que un modelo con peor índice de desempeño, pero menos complejo que el modelo óptimo, pueda obtener resultados aceptables bajo un estándar de rendimiento definido preliminarmente. Por lo antes expuesto para este problema se escoge como número de clusters 5, por lo que el modelo difuso contará con 5 reglas, Fig 5.

Además del RMSE, se pueden definir el Error Porcentual Absoluto Medio (MAPE, Mean Absolute Percentage Error) y Error Absoluto Medio (MAE, Mean Absolute Error) para comprobar el modelo difuso obtenido.

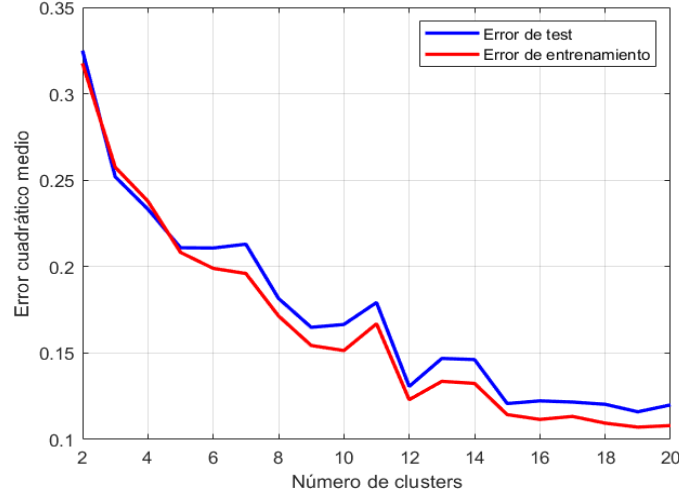


Figura 4: Índice de Sensibilidades.

$$MAPE = \frac{1}{N} \sum_{k=1}^N \frac{y(k) - y_{fuzzy}(k)}{y(k)} \quad (13)$$

$$MAE = \frac{1}{N} \sum_{k=1}^N |y(k) - y_{fuzzy}(k)| \quad (14)$$

Tabla 3: Índices de Error para el Modelo Difuso

Índices	Conjunto		
	Entrenamiento	Prueba	Validación
RMSE	0.2082	0.2109	0.1938
MAPE	70.92	65.65	67.71
MAE	0.1454	0.1505	0.1344

En la Tabla 3 se puede ver como el RMSE se mantiene de similar para los conjuntos de prueba y validación lo que indica la capacidad de generalización del modelo. Por su parte el índice MAPE, mide el tamaño del error (absoluto) en términos porcentuales, indicando que el error porcentual promedio del modelo se encuentra alrededor del 70 %, si bien este es un valor grande, es válido destacar que se está trabajando con una señal cuya salida se encuentra

entre $[-2,3]$. El MAE tampoco presenta variaciones entre los diferentes conjuntos de datos. Considerando un compromiso entre complejidad y desempeño se determina que el modelo propuesto con 4 regresores y 5 reglas es suficiente.

- **Optimización de los parámetros:** Se utiliza el algortimo de clustering difuso Fuzzy C-Means obteniéndose siguientes parámetros de los consecuentes $\theta^T = [\theta_{1,1}, \dots, \theta_{N_r,1}, \dots, \theta_{1,n}, \dots, \theta_{N_r,1}]$

$$\theta^T = \begin{pmatrix} -0,1625 & 0,99755 & -0,7451 & 0,8430 & 0,2983 \\ 0,4561 & 1,1500 & -0,3126 & 0,5930 & 0,1362 \\ -0,2139 & 0,8892 & -0,2883 & 1,0148 & 0,2207 \\ -0,10934 & 0,8182 & -0,9810 & 0,6261 & 0,1752 \\ 0,0032 & 0,5939 & -0,9615 & 1,0893 & 0,3707 \end{pmatrix} \quad (15)$$

- **Predicciones a 1, 8, y 16 pasos:**

Las Fig. 6 muestra la salida del modelo difuso y la estimación para 1, 8 y 16 pasos respectivamente.

En la Tabla 4 se establece una comparación entre los modelos obtenidos, como se puede comprobar a amedida que aumenta el paso de la estimación el desempeño de del modelo difuso se deteriora.

Tabla 4: Desempeño del Modelo difuso a j-pasos

Índices	Modelos			
	Estimado	Predicción a 1 pasos	Predicción a 8 pasos	Predicción a 16 pasos
RMSE	0.1938	0.2432	0.7077	0.8847
MAPE	67.71	108.72	159.78	169.81
MAE	0.1344	0.1816	0.5076	0.6814

- **Covarianza:**
- **MinMax:**

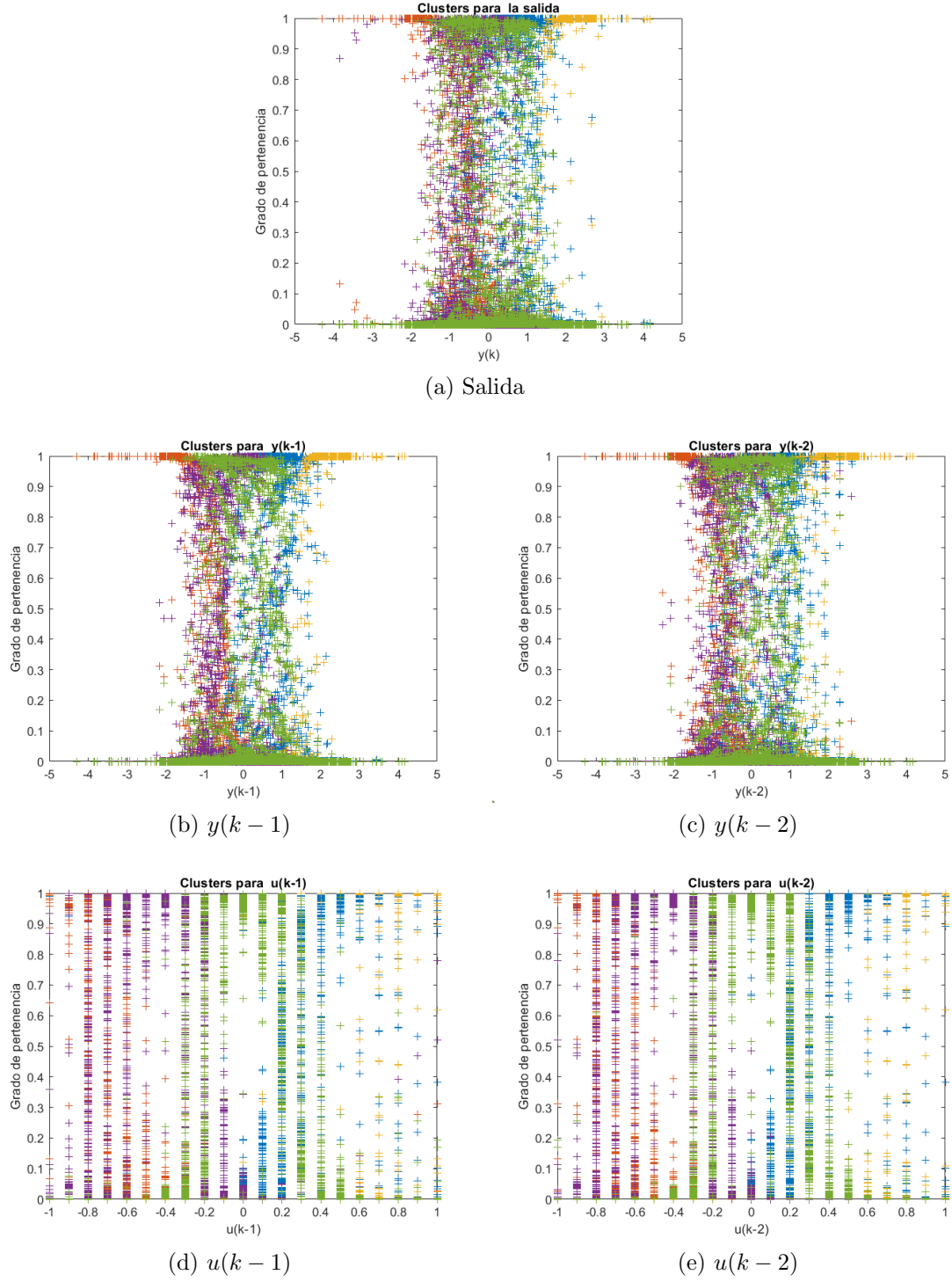
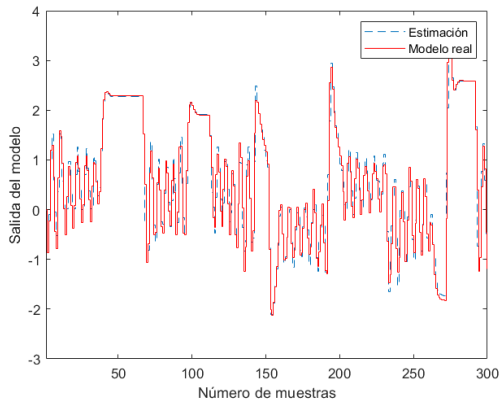
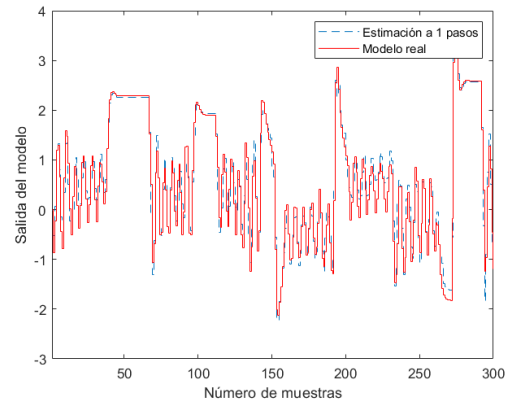


Figura 5: Clusters del Modelo Difuso Tipo 1.

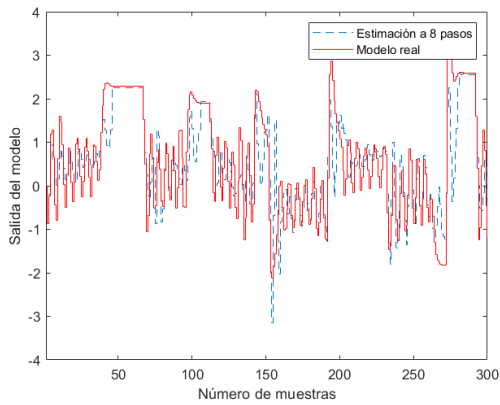
Ejercicio N°1



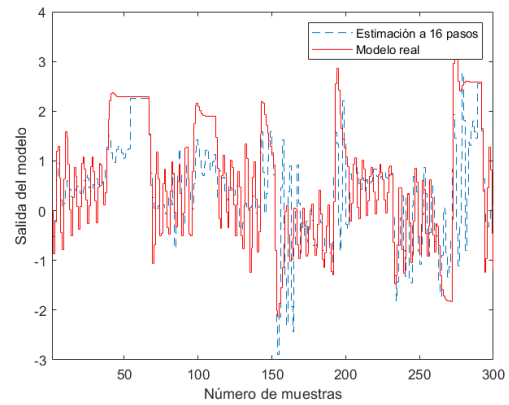
(a) Salida del Modelo



(b) Salida del modelo a 1 paso



(c) Salida del modelo a 8 pasos



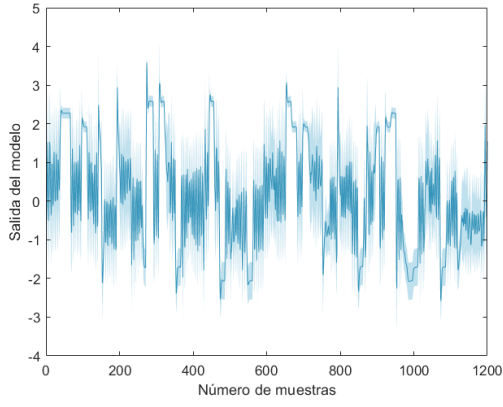
(d) Salida del modelo a 16 pasos

Figura 6: Respuesta del Modelo Difuso Tipo 1.

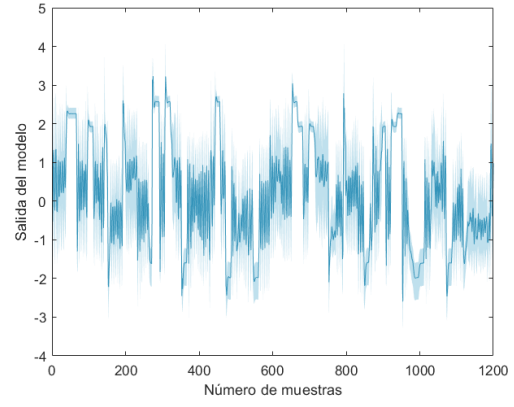
Tabla 5: Intervalos de predicción. Método de la Covarianza

Índices	Modelos			
	Estimado	Predicción a 1 pasos	Predicción a 8 pasos	Predicción a 16 pasos
PICP	100	99.33	84.67	72
PINAW	27.59	29.2	29.55	28.1

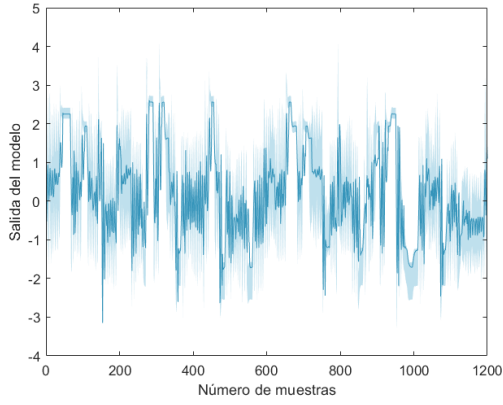
Ejercicio N°1



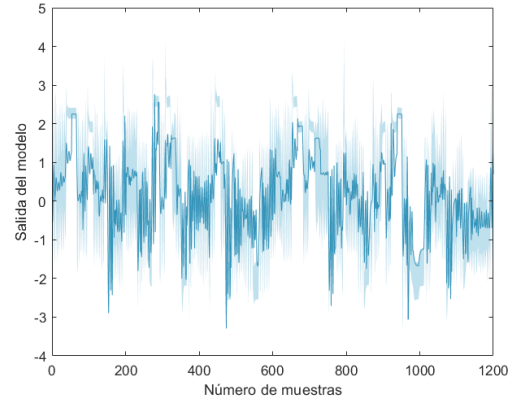
(a) Salida del modelo. Método de la Covarianza



(b) Salida del modelo a 1 paso



(c) Salida del modelo a 8 pasos

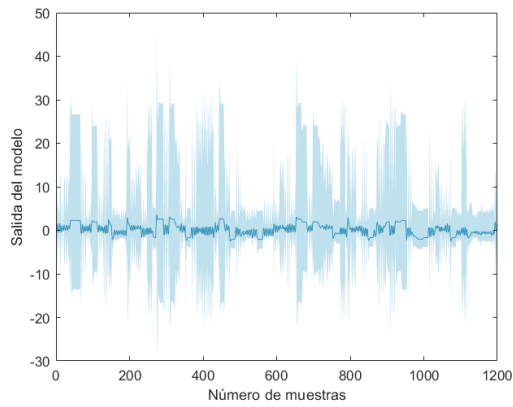


(d) Salida del modelo a 16 pasos

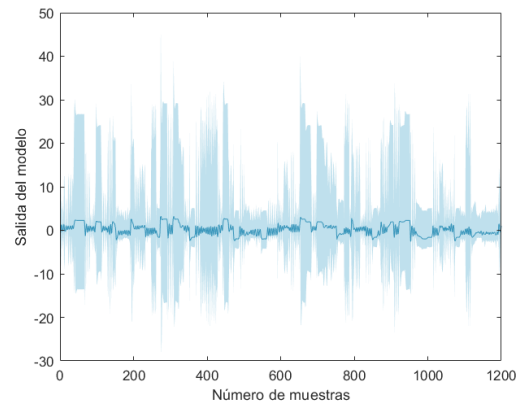
Figura 7: Intervalo difuso. Método de la Covarianza.

Tabla 6: Intervalos de predicción. Método MinMax

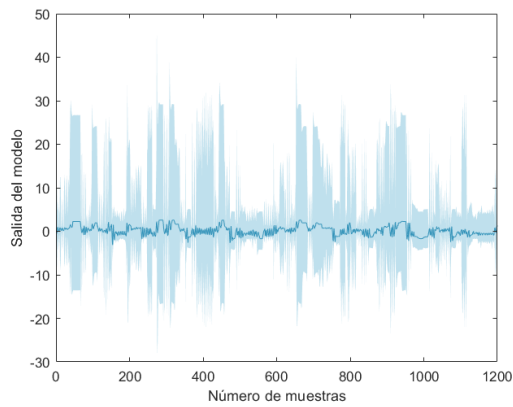
Índices	Modelos			
	Estimado	Predicción a 1 pasos	Predicción a 8 pasos	Predicción a 16 pasos
PICP	98.5	98.08	96.42	95.67
PINAW	296.69	313.96	317.784	302.19



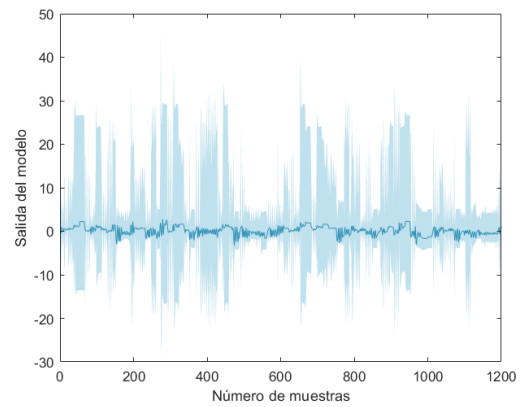
(a) Salida del modelo. Método de la Covarianza



(b) Salida del modelo a 1 paso



(c) Salida del modelo a 8 pasos



(d) Salida del modelo a 16 pasos

Figura 8: Intervalo difuso. Método MinMax

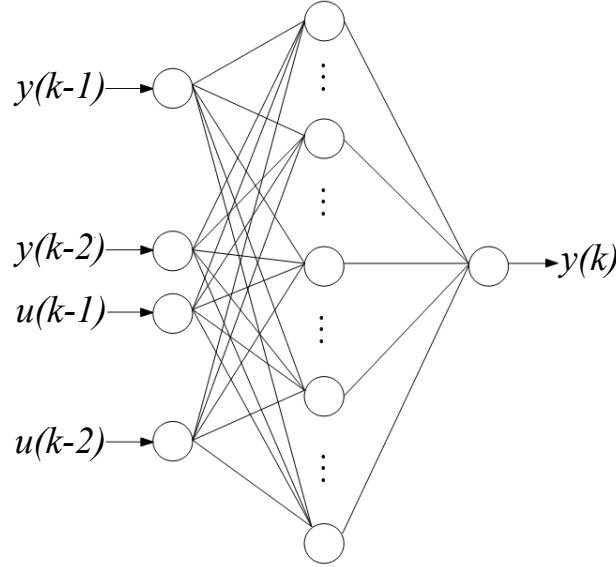


Figura 9: Estructura de la red neuronal perceptrón

3.3. Modelo de red neuronal

Para encontrar un buen modelo de red neuronal se propone seguir los pasos de identificación.

- **Obtención de datos:** Para ello se utiliza el set de datos creado en el punto a) del problema.
- **Selección de datos:** Análogo a los modelos anteriores, se utilizan los datos repartido con un 55 % en el conjunto de entrenamiento, 25 % en el conjunto de prueba y 20 % en el conjunto de validación.
- **Definición de la estructura de la red:** Se propone una red con una capa oculta, función de activación \tanh en la salida de la capa oculta y algoritmo de aprendizaje Levenberg-Marquardt. Las variables de entradas son iguales al número de regresores (4 entradas) como muestra la Figura 9. Por otro lado, el modelo matemático de la red, queda expresado como se muestra en la ecuación 16.

$$\hat{y}(k) = \sum_{i=1}^{N_h} r w_i \left(\tanh \left(\sum_{j=1}^{N_I} l w_{ji} x_j + b_i \right) \right) + c \quad (16)$$

donde, $x = [y(k-1), y(k-2), u(k-1), u(k-2)]$ es el vector de entrada, N_h es el número de neuronas en la capa oculta, N_I es el número de variables en la

entrada, rw_i es el peso que conecta la i -ésima neurona de la capa oculta con el nodo de salida y lw_{ji} corresponde al peso que una la entrada j con la i -ésima neurona en la capa oculta. Los sesgos de cada neuronas en la capa oculta y para el nodo de salida son b_i y c respectivamente.

- **Selección de entradas relevantes:** De los 4 regresores presente en el sistema se debe analizar cuál tiene mayor peso en el modelo. Un método para encontrar dichos regresores es mediante un análisis de sensibilidad evaluando la derivada de la salida de la red por cada premisa de nuestros datos, es decir,

$$\xi_j = \frac{\partial \hat{y}(k)}{\partial x_j} \quad (17)$$

Como la funcion de activacion es \tanh y en la salida es lineal, se tiene,

$$\begin{aligned} \xi_j &= \frac{\partial \hat{y}(k)}{\partial x_j} \\ &= \sum_{i=1}^{N_h} rw_i \left(1 - \tanh \left(\sum_{m=1}^{N_I} lw_{mi} x_m + b_i \right)^2 \right) lw_{ji} \end{aligned} \quad (18)$$

Como se tendrá un valor de ξ_j para cada dato vector de entrada, se genera un vector $\boldsymbol{\xi}_j$ del mismo largo que el número de datos de cada variable.

Luego, se hace uso de un *indicador* I_j para cada entrada j definido como,

$$I_j = \mu^2(\boldsymbol{\xi}_j) + \sigma^2(\boldsymbol{\xi}_j) \quad (19)$$

donde μ es la media del vector de datos y σ^2 es la varianza para cada entrada j .

- **Optimización paramétrica y estructural:** Para encontrar los valores óptimos de las parámetros peso y sesgo de la red neuronal se utiliza el algoritmo de Levenberg-Marquardt backpropagation [1]. Por otro lado, para encontrar el óptimo de la estructura se analiza cuantas neuronas debe tener la capa oculta. Para ello se evalúa el RMSE (Raíz del error cuadrático medio) del conjunto de prueba en la salida de la red para un número de neuronas entre $[2 - 41]$. Los resultados de sensibilidad para cada neurona en la capa oculta se muestran en las Figura 11 y 12 y el RMSE evaluado en los 3 conjuntos se muestra en la Figura 10. Se puede ver que el mínimo RMSE para el conjunto de prueba es para 6 neuronas y que el modelo es menos sensible a la entrada $u(k-2)$ el cual es eliminado del entrenamiento.

El entrenamiento se configura a una velocidad inicial de aprendizaje de la red de 0,05 y un valor de épocas de 5000 con evaluación de *Overfitting* de 200 épocas de validación. Es importante señalar que durante el experimento se probaron diferentes épocas de validación para cuantificar el efecto del sobre ajuste en el número óptimo de neuronas con la hipótesis de que, independiente del RMSE obtenido, el mínimo global se da con la misma cantidad de neuronas óptimas afectando solo a los mínimos locales que pudieran aparecer. En la Tabla 7 se muestran los resultados de varios experimentos demostrando la hipótesis, es más, el valor RMSE obtenido con todo los modelos es casi el mismo y solo cambia el número de neuronas, donde, 6 demuestra ser un mínimo global y los demás un mínimo local. Por otro lado, se observo que tras varios experimentos el mejor valor RMSE del conjunto de prueba se observa cuando las épocas de validación por sobre ajuste se encuentran entre $[50 - 300]$, de esta manera, de aquí en adelante se utilizan 50 épocas de validación por sobre ajuste para disminuir los costos de calculo.

Tabla 7: Experimento para ver la influencia del sobre ajuste.

Número de épocas de validación	N_h óptimo	Valor RMSE
1	11	0.0025392
10	7	0.0025384
50	6	0.0025342
100	6	0.0025295
200	6	0.0025324
500	6	0.0025326
1000	10	0.0025299
5000	17	0.0025383

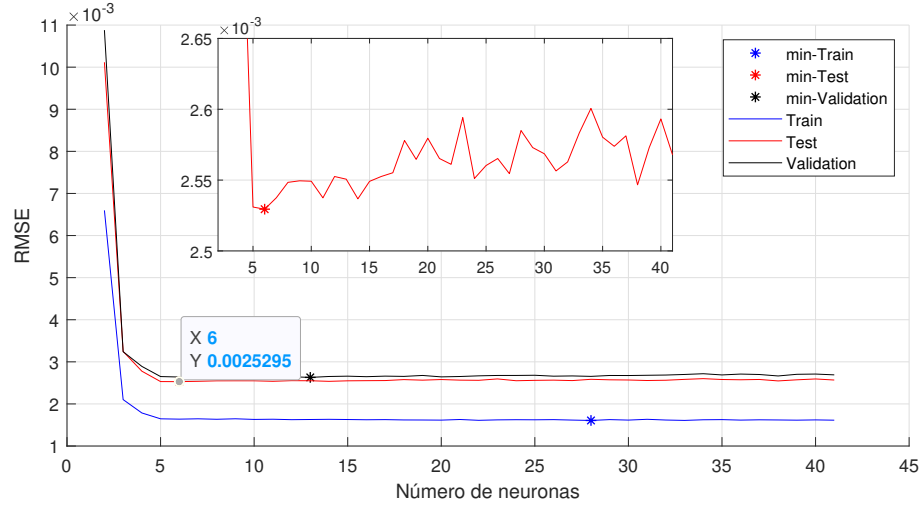


Figura 10: RMSE para diferente número de neuronas.

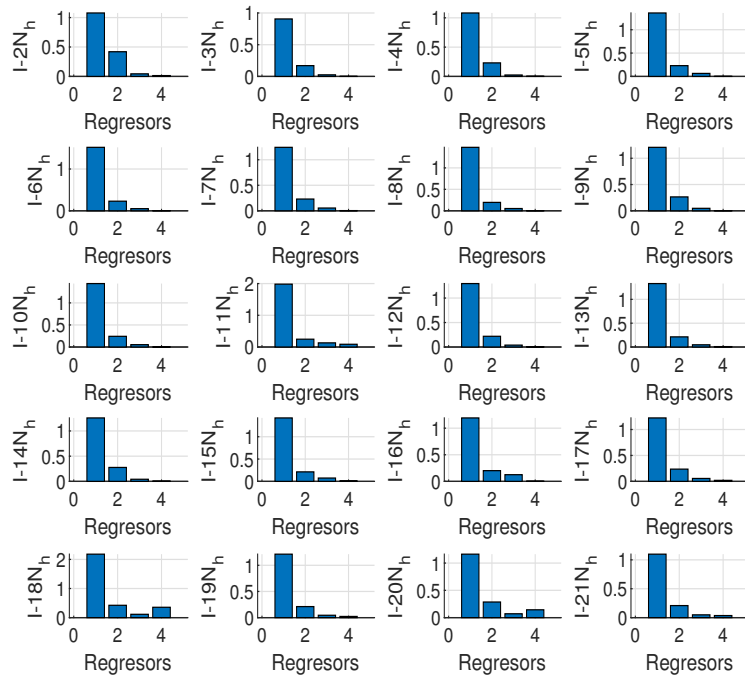


Figura 11: Sensibilidad para un número de neuronas entre $[2 - 21]$ en la capa oculta.

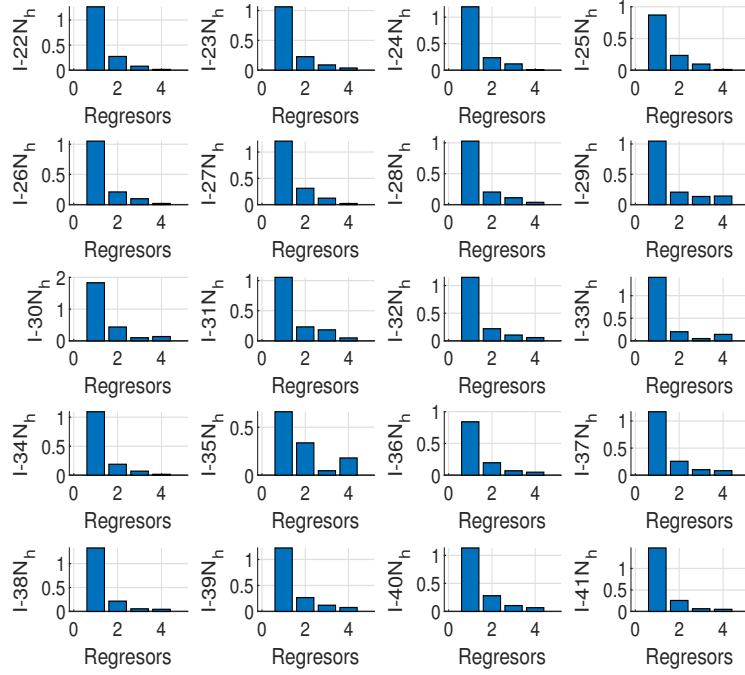


Figura 12: Sensibilidad para un número de neuronas entre $[22 - 41]$ en la capa oculta.

- **Desempeño de la red definida:** Se procede a evaluar el desempeño de la red con 8 neuronas en la capa oculta utilizando las cuatro entradas y luego solo con 3, $[y(k-1), y(k-2), u(k-1)]$, para notar las diferencias. Los resultados se muestran en las Figuras 13 hasta la 15 y los datos numéricos se agrupan en la Tabla 8

Tabla 8: Valores MSE para los 3 conjuntos de datos evaluados en una red con 4 entradas y otra con 3 entradas.

Número de entradas	MSE - Entrenamiento	MSE - Prueba	MSE - Validación
4	0.0089	0.0097	0.0084
3	0.0108	0.0118	0.0108

Se puede notar de la Figura 13 que al quitar la entrada $u(k-2)$ aumenta el indicador I_1 de la entrada $y(k-1)$ para compensar la falta del autoregresor $u(k-2)$ manteniendo casi al mismo valor las otras dos entradas. Por otro lado, no se eliminan más entradas dado que el ancho del histograma de la Figura 15(b) comienza a aumentar de valor mostrando que más datos tienen

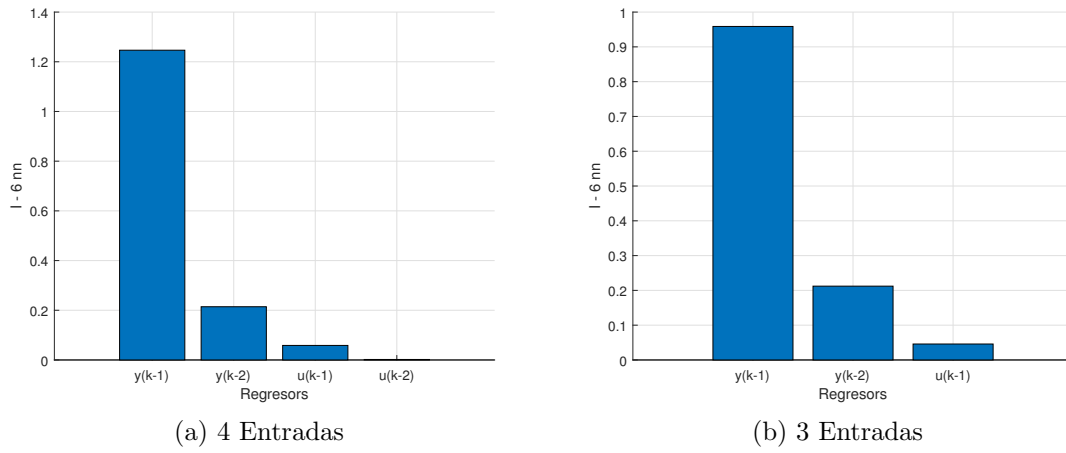
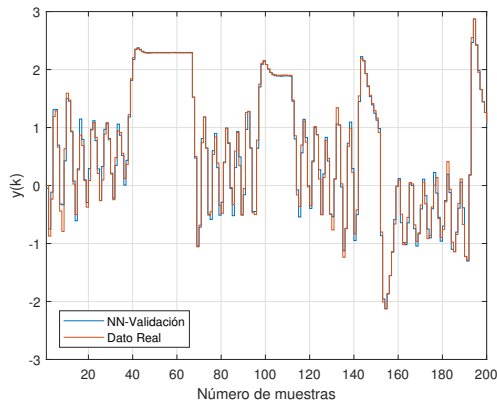


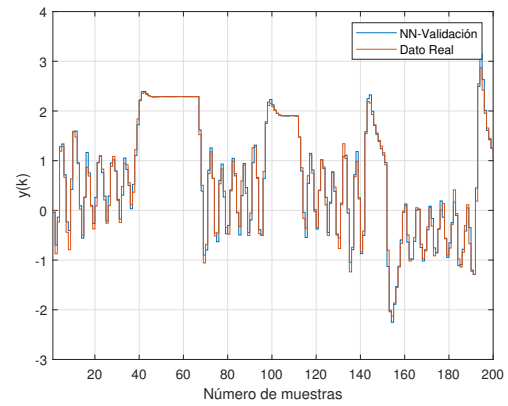
Figura 13: Índice de sensibilidad para 4 y 3 entradas.

errores grandes. Este error puede tener solución al utilizar un nuevo número de neuronas en la capa oculta, por lo tanto, en el siguiente ítem se procede nuevamente a encontrar el número óptimo de neuronas con 3 entradas.

Ejercicio N°1

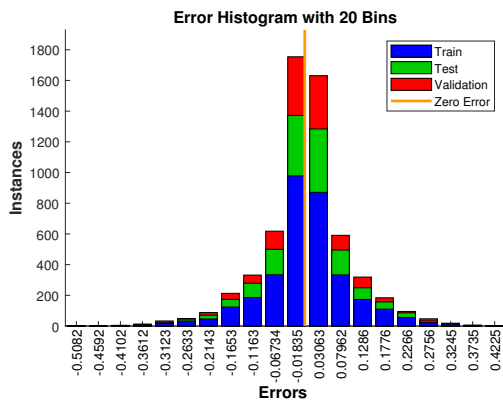


(a) 4 Entradas

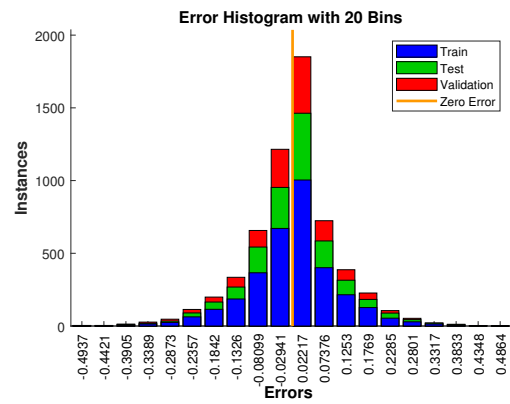


(b) 3 Entradas

Figura 14: Comparación entre la salida de la red neuronal y el valor real del conjunto de validación.



(a) 4 Entradas



(b) 3 Entradas

Figura 15: Histograma del error de cada conjunto de datos.

- **Optimizar estructura 2:** Se busca el número de neuronas óptimo en un rango de $[2 - 41]$. El resultado de la curva RMSE se muestra en la Figura 16 y el análisis de sensibilidad para cada neurona en las Figuras 17 y 18.

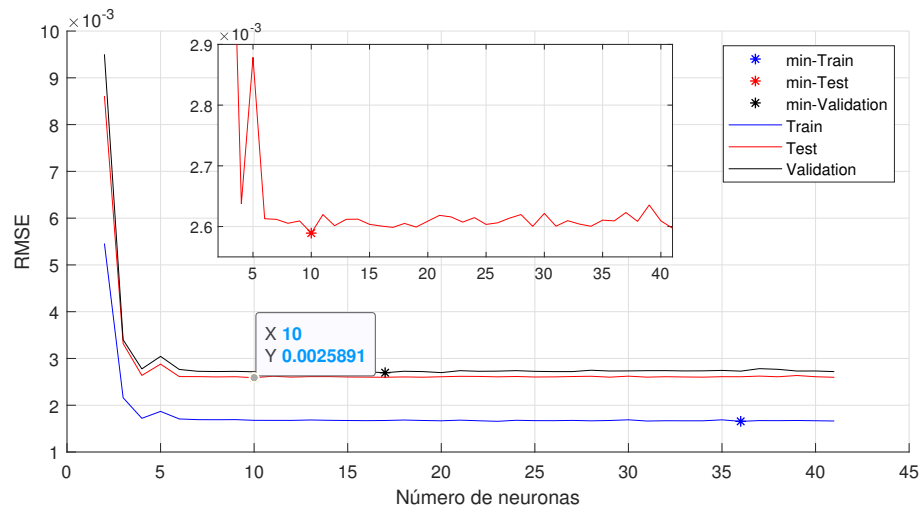


Figura 16: RMSE para diferente número de neuronas.

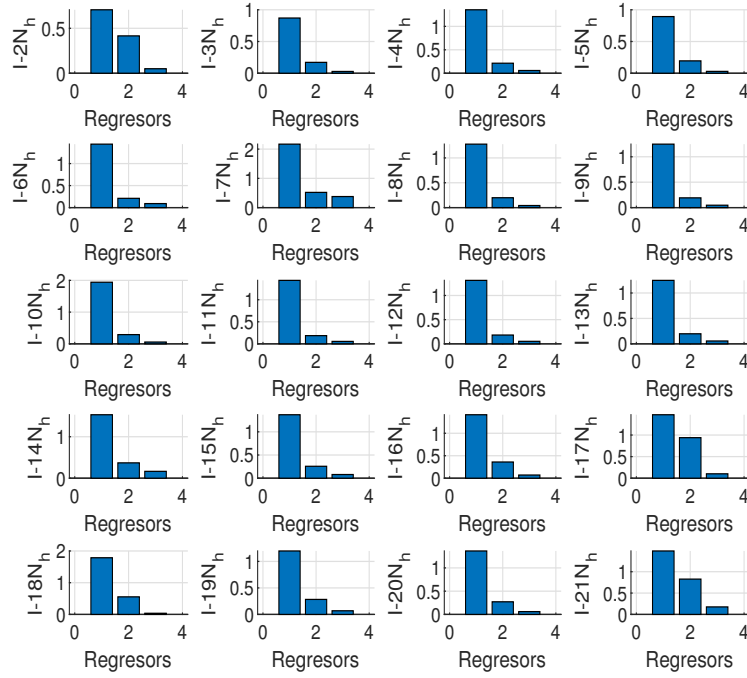


Figura 17: Sensibilidad para un número de neuronas entre $[2 - 21]$ en la capa oculta con 3 entradas.

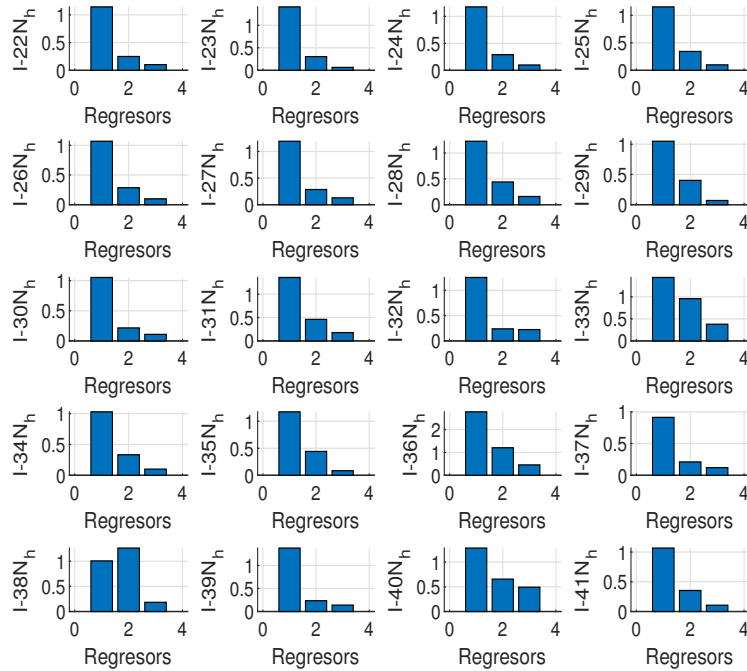


Figura 18: Sensibilidad para un número de neuronas entre $[22 - 41]$ en la capa oculta con 3 entradas.

Validación final de la estructura: Para finalizar con la identificación del modelo de predicción por red neuronal, se evalúan los parámetros de la red, MAE, MAPE, RMSE, MSE reagrupados en la Tabla ??.

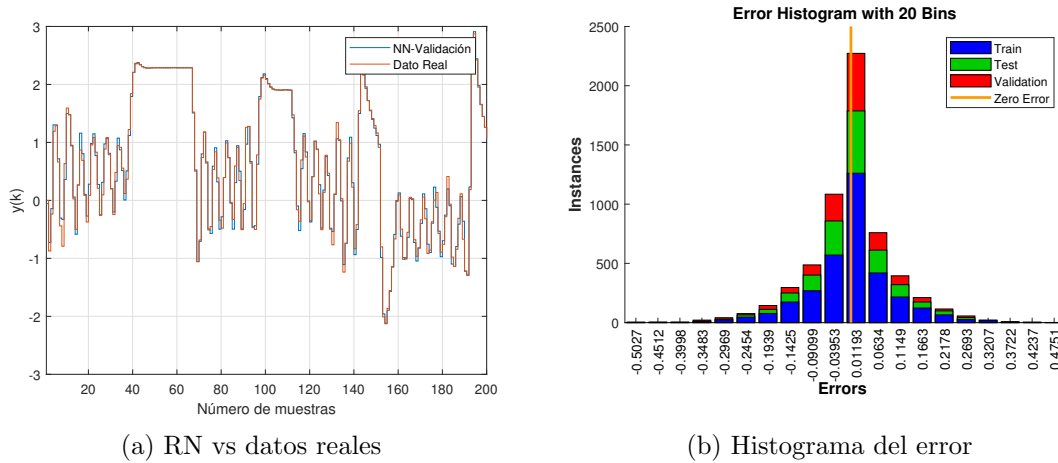


Figura 19: Resultados del modelo neuronal con 10 neuronas en la capa oculta en comparación con los datos reales.

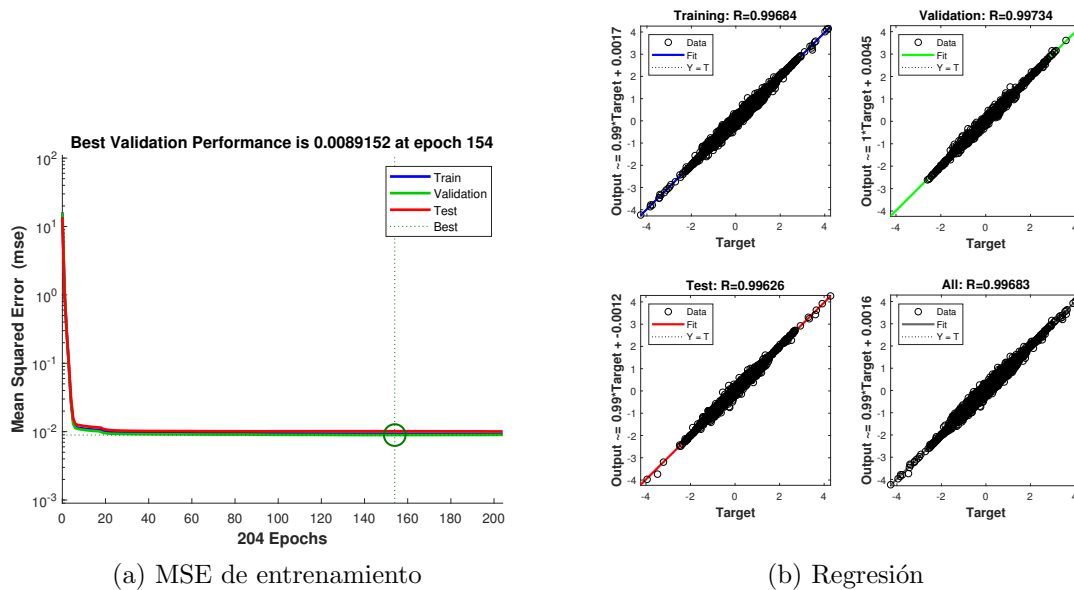


Figura 20: Resultados de entrenamiento mostrando la regresión de cada conjunto.

Tabla 9: Métricas finales del modelo neuronal

Métricas	Conjunto de entrenamiento	Conjunto de prueba	Conjunto de validación
RMSE	0.001681	0.002599	0.002726
MSE	0.009324	0.01013	0.008915
MAE	0.06538	0.06852	0.06117
MAPE	45.25	60.20	35.55

Modelo predictivo: Se comienza de la premisa que se conocen todos los valores hasta $\hat{y}(k)$.

- **A 1 paso:** Para realizar un modelo predictivo a 1 paso (un paso más y aparte del generado en sí por el modelo neuronal para pasar de $(k-1)$ a (k)), se tiene,

$$\hat{y}(k+1) = \sum_{i=1}^{N_h} r w_i \left(\tanh \left(\sum_{m=1}^{N_I} l w_{mi} x_m(k+1) + b_i \right) \right) + c \quad (20)$$

recordando la definición del vector x , se tiene, que $x(k+1) = [\hat{y}(k), y(k-1), u(k)]$. Existe una forma de determinar la señal de control futura $u(k)$ mediante modelo de control predictivo (MPC) [2], sin embargo, dado que se tiene un número grande de datos y que la señal APBRS se mantiene constante por lo menos en 40 puntos por bits antes de cambiar, se dará por conocido el control $u(k)$ requerido hasta el total de datos menos j .

- **A 8 paso:**
- **A 16 paso:**

4. Resultados

5. Conclusión

Referencias

- [1] Matlab, “trainlm, levenberg-marquardt backpropagation.” [Online]. Available: <https://www.mathworks.com/help/deeplearning/ref/trainlm.html>
- [2] M. Nørgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*, ser. Advanced Textbooks in Control and Signal Processing, M. J. Grimble and M. A. Johnson, Eds. London: Springer London, 2000. [Online]. Available: <http://link.springer.com/10.1007/978-1-4471-0453-7>