

Ejercicio 1

Profesora: Doris Saez Hueichapan

Auxiliar: Javier Ocaranza

Ayudantes: Luis Jiménez, Felipe Meyer, Benjamín Moreno

Autores: Benjamín Brito, Jorge García, Diego Sanz, Bruno Masserano

Pregunta 2

Parte a

Para esta parte se usó la ecuación 1 con el fin de obtener una señal a caracterizar mediante modelos tipo Takagi-Sugeno y redes neuronales. Para caracterizar el sistema se empleó una señal de entrada tipo APRBS con tiempo de muestreo en 0.01 segundos, 100 segundos de simulación, lo que genera 10.000 muestras de largo, frecuencia mínima 0.5 Hz y máxima de 5 Hz, amplitud entre -2 y 2 y ruido blanco aditivo de desviación estándar 0.1. La figura 1.a muestra la señal de entrada entregada al modelo, mientras que la figura 1.b muestra la señal de salida resultante a lo largo del tiempo. Dicha señal servirá posteriormente para caracterizar el sistema.

$$\begin{aligned} y(k) = & (0.5 - 0.3e^{-y^2(k-1)})y(k-1) \\ & -(0.2 + 0.8e^{-y^2(k-1)})y(k-2) \\ & +u(k-1) + 0.2u(k-2) + 0.1u(k-1)u(k-2) + \epsilon(k) \\ \epsilon(k) = & 0.5e^{-y^2(k-1)}\beta(k) \\ \beta \sim & N(0, 1) \end{aligned} \quad (1)$$

Parte b

A partir de la señal generada en la parte a se construyeron un modelo Takagi-Sugeno y una red neuronal. En primera instancia se dividieron los datos en:

- 60 % Entrenamiento
- 20 % Test
- 20 % Validación

Con esto se procede al entrenamiento de ambos modelos, la descripción del proceso de construcción de los modelos se dará a continuación.

Modelo Takagi-Sugeno

Utilizando el conjunto de entrenamiento se obtuvieron clusters difusos con los que luego se pueden computar los grados de pertenencia de los distintos regresores a las reglas definidas inicialmente,

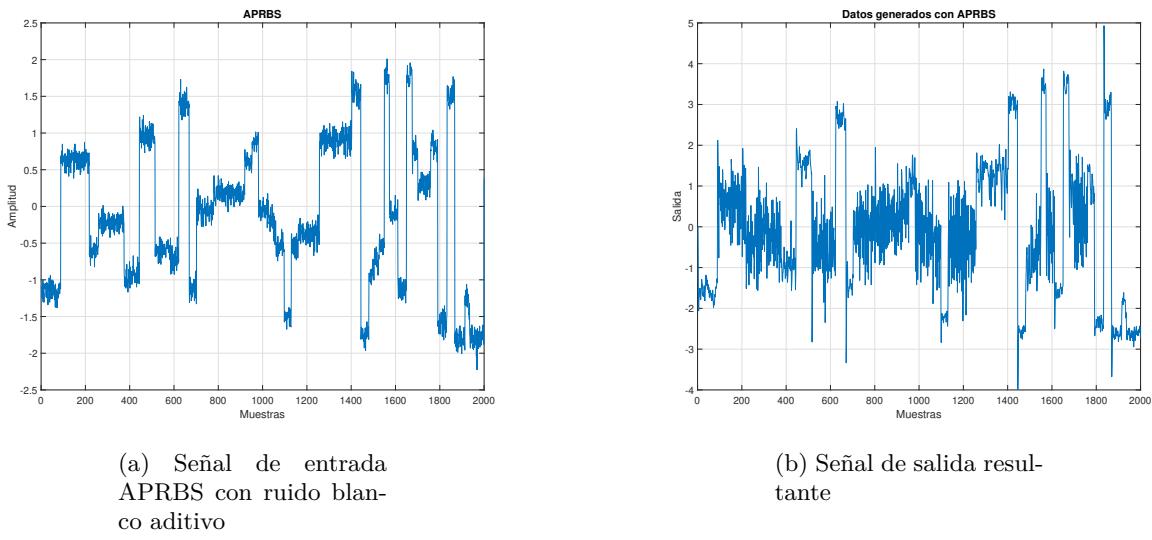


Figura 1: Señales de entrada y salida

esto se traduce en un modelo ARX con distintas variables que es sintonizado por mínimos cuadrados.

Para optimizar la estructura del modelo se inició por buscar el número óptimo de clusters, fijando de forma preliminar la estructura del modelo con 10 autoregresores y 10 regresores para la variable exógena.

Posteriormente, se fue variando la cantidad de clusters del sistema entre 2 y 10, obteniendo distintos modelos cuyo desempeño fue evaluado con el dataset de “test” en base al indicador RMSE. El desempeño de los modelos se observa en la figura 2 y se concluye que la cantidad óptima de clusters es de 6.

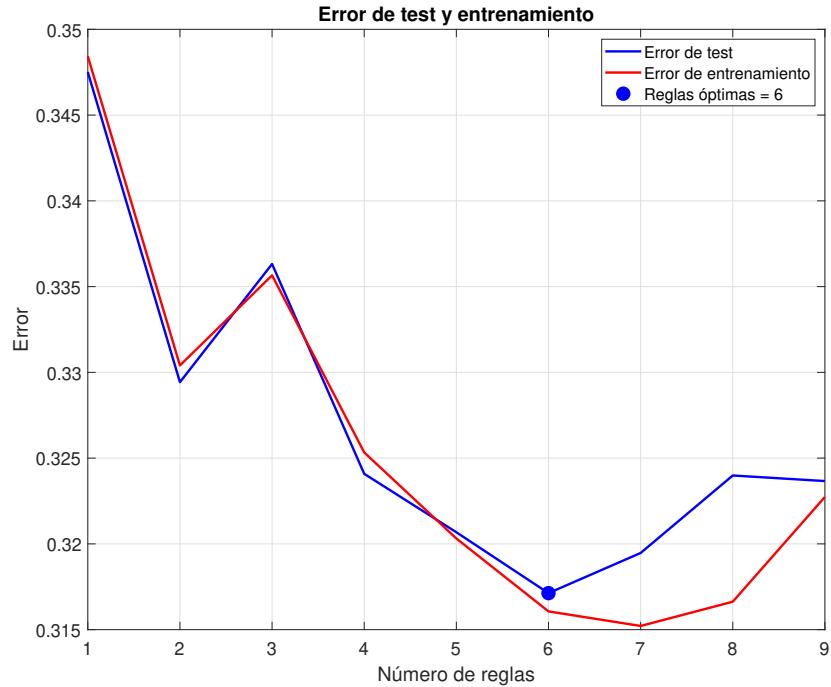


Figura 2: Error de test y entrenamiento

Con la cantidad óptima de grupos ya definida se procedió a optimizar la cantidad de regresores que conforman el modelo a partir del análisis de sensibilidad de la salida con respecto a cada regresor. Para ello se calcula la sensibilidad de la salida con respecto a cada regresor para cada dato en conjunto de validación y se construye una métrica de importancia de la forma $\bar{\delta}^2 + \sigma_{\delta}^2$ con $\bar{\delta}$ el promedio de las sensibilidades y σ_{δ}^2 la desviación estándar. En función de la métrica de sensibilidad se procede a eliminar iterativamente los regresores con menor importancia siempre y cuando la eliminación de dicho elemento mejore la métrica de error del modelo en términos de RMSE y se reentrena el para volver a computar análisis de sensibilidad en una nueva iteración.

La figura 3 muestra la distribución de sensibilidad para cada regresión en distintas iteraciones, donde se elimina el regresor que presenta la menor métrica de sensibilidad para esta iteración. Por su parte la figura 4 resume los resultados del proceso, pues muestra el RMSE en función de la iteración y por consiguiente su evolución al quitar regresores progresivamente. Se puede verificar que al quitar regresores el RMSE baja hasta el punto en el que quedan 6, momento en el que se llega a un mínimo local.

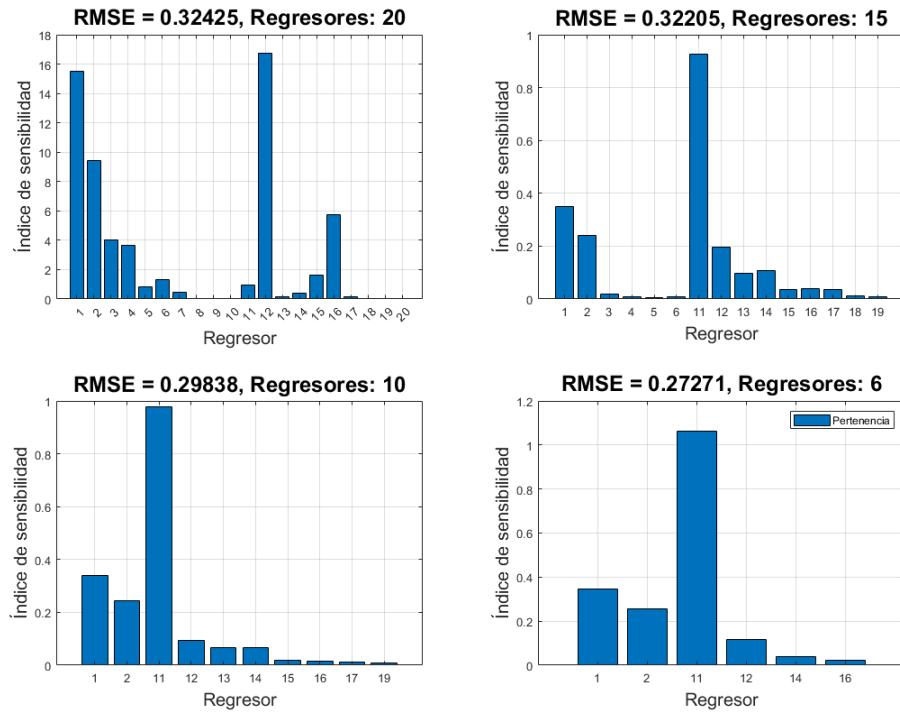


Figura 3: Evolución de la métrica de sensibilidad a medida que se quitan regresores.

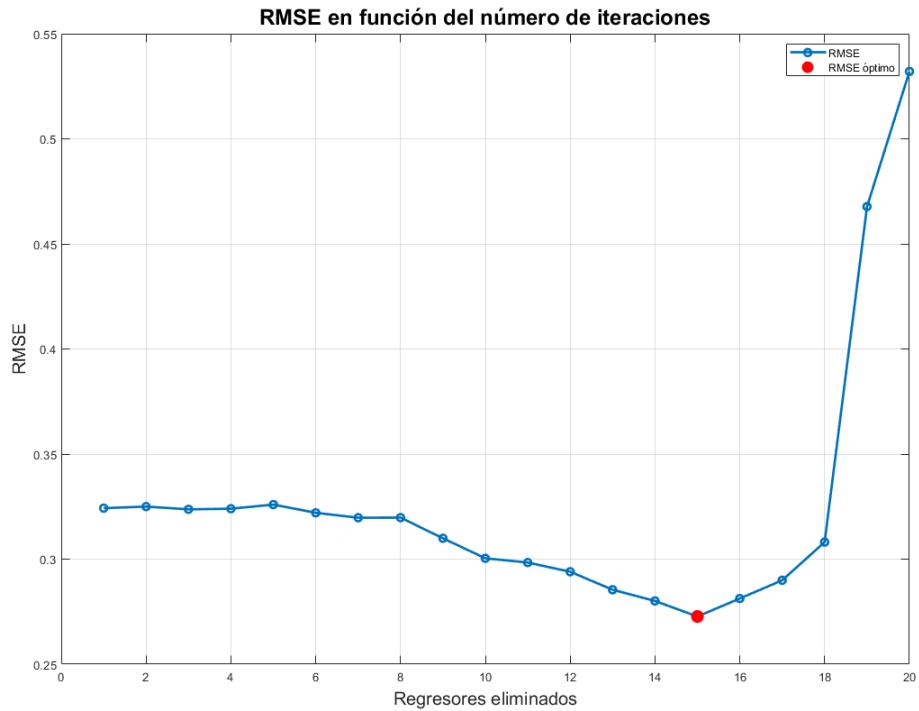


Figura 4: Evolución del RMSE en función del número de regresores eliminados.

Con la estructura óptima encontrada se reentrena el modelo y se obtienen los resultados de la figura 5, donde se puede verificar que la salida real y predicha es similar en varios puntos y transientes para el dataset de validación. Un análisis más profundo de la efectividad del modelo será provisto en la **Parte c**. Adicionalmente, al analizar el modelo resultante de la optimización estructural se llegó a que los autoregresores más importantes son a uno y dos pasos, mientras que para la variable exógena son a uno, dos, tres y cuatro pasos. Se puede verificar que los regresores más importantes encontrados para la salida y entrada efectivamente corresponden a los que definen el comportamiento del sistema dado por la ecuación 1.

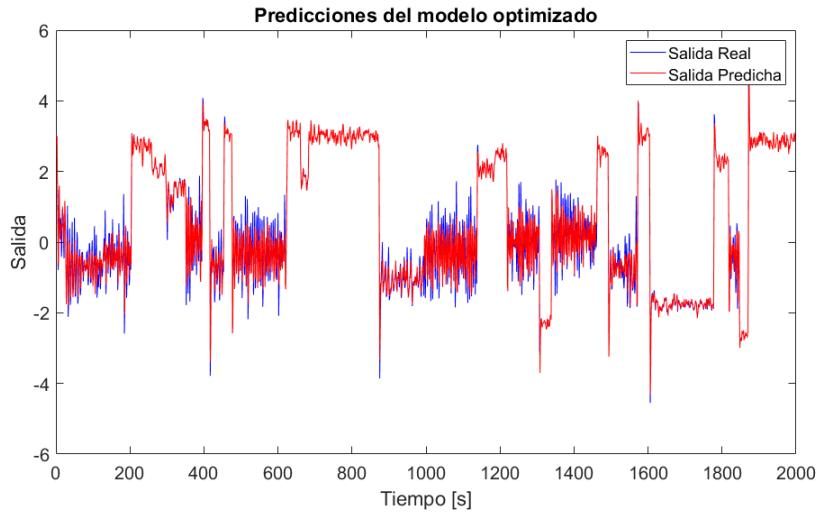


Figura 5: Comparación salida real y predicha del modelo TS óptimo.

Modelo Red Neuronal

Para la obtención del modelo neuronal se usaron los mismos datasets de entrenamiento, test y validación que en la parte anterior, además de realizar un proceso similar para la optimización estructural de la red. Para este tipo de modelos se usó el toolbox de Matlab para emplear la variante “SCG” de backpropagation con tasa de aprendizaje y épocas por defecto.

Para lo optimización estructural se procedió a calcular la sensibilidad de la salida con respecto a cada neurona en la capa oculta para cada dato del conjunto de test y construir la métrica de sensibilidad $\bar{\delta}^2 + \sigma_\delta^2$ presentada en la parte anterior. La figura 6 muestra la distribución de sensibilidad para cada neurona inicialmente, se nota que todos los regresores tienen alto índice de sensibilidad sobre el rendimiento de la red, por lo que no se elimina ningún regresor, resultando en una red con 20 regresores de entrada, 10 autoregresores y otros 10 de la entrada.

Para ajustar la cantidad de neuronas de la capa oculta se entrenaron varios modelos con distintas arquitecturas y se conmutó su RMSE con el dataset de test, seleccionando así el modelo que minimiza la métrica. Los resultados del proceso se encuentran en la figura 7.

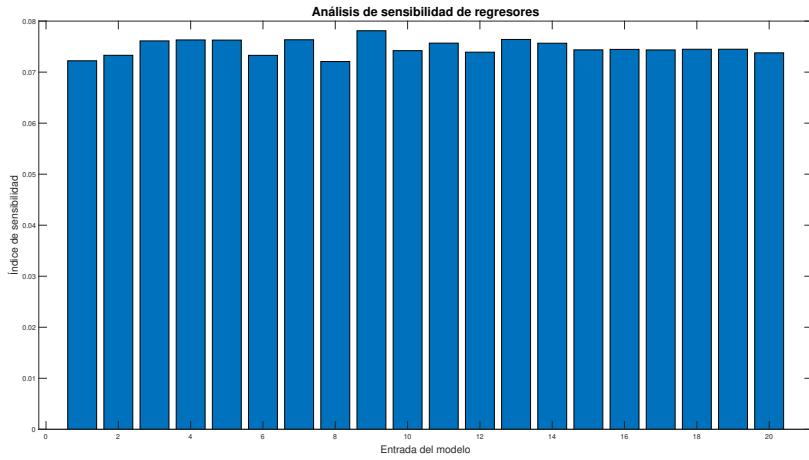


Figura 6: Sensibilidad de regresores para modelo NN.

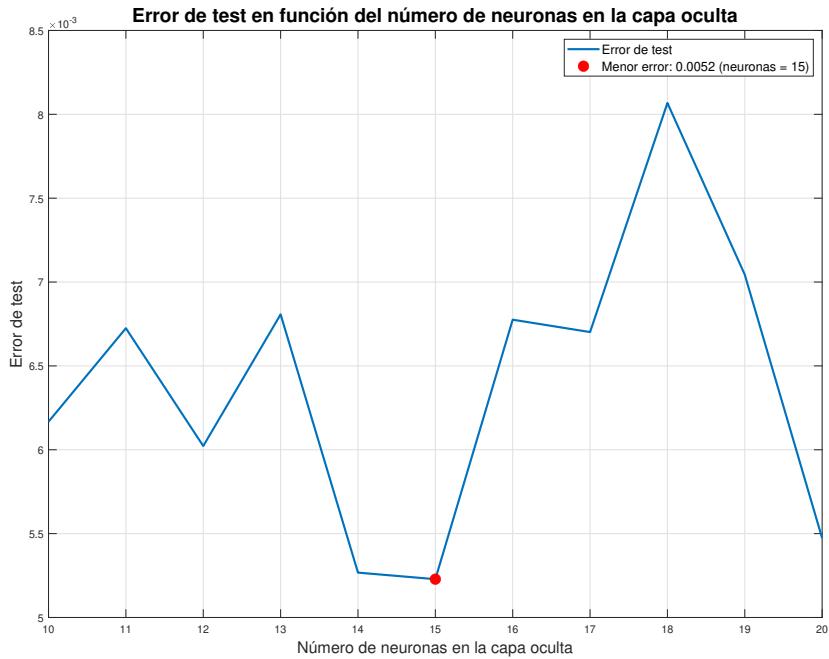


Figura 7: Resultado optimización del número de capas ocultas.

Con el modelo entrenado se generaron predicciones a un paso con el dataset de validación resultando en el gráfico de la figura 8, se puede verificar cómo las predicciones a un paso efectivamente logran seguir la salida del modelo para varios puntos de operación, siendo las zonas de estabilización de la señal APRBS donde el modelo se acerca más a los resultados reales y los zonas de cambio brusco de amplitud aquellos donde más falla.

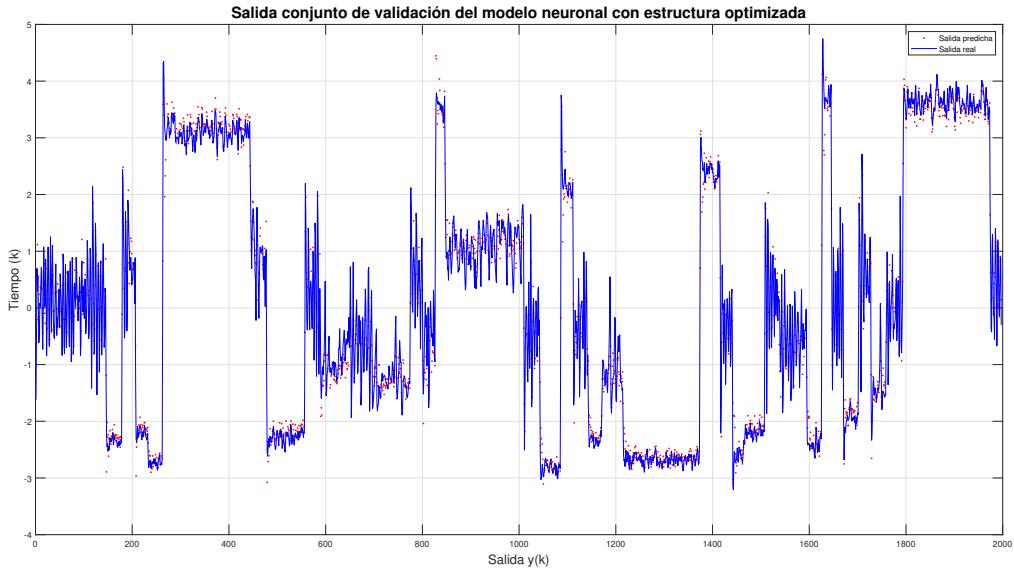


Figura 8: Comparación salida real y predicha del modelo NN óptimo.

Parte c

Para comparar el desempeño de los modelos se usaron el RMSE y MAE, descritas en las ecuaciones 2 y 3, como métricas de error de predicción a un paso. En particular, se usaron esas métricas por su capacidad para representar la diferencia promedio entre cada punto predicho y el valor real de formas distintas. La tabla 1 sintetiza los resultados obtenidos.

$$RMSE = \sqrt{\frac{1}{N} \sum_i^N (y_i - \hat{y}_i)^2} \quad (2)$$

$$MAE = \frac{1}{N} \sum_i^N |y_i - \hat{y}_i| \quad (3)$$

Horizonte de Predicción	Métrica	Takagi-Sugeno	Neural Network
1 paso a futuro	RMSE	0.2761	0.3443
	MAE	0.1566	0.2336

Tabla 1: Métricas de rendimiento para modelos TS y NN optimizados a 1 paso.

A partir de los resultados se puede verificar cómo si bien ambos modelos logran una buena métrica de error en términos de RMSE y MAE, el modelo Takagi-Sugeno presenta mejores resultados para los datos de prueba. En las secciones siguientes se procederá a construir intervalos sobre los modelos optimizados y ses comparará su desempeño a distintos horizontes de predicción.

Parte d

Para generar un modelo con intervalos de confianza se aplicó el método de la covarianza imponiendo un porcentaje de cobertura del 95 %. Sabemos que en el método de la covarianza para modelos tipo Takagi-Sugeno se predicen los límites $\hat{y}(k)$ y $\bar{y}(k)$, las cuales permiten generar una cota superior e inferior que actúan como intervalo de confianza. Las ecuaciones 4 y 5 rigen el comportamiento de los modelos a implementar, donde \hat{y} corresponde al modelo base calculado anteriormente, α un parámetro que ajusta el porcentaje de cobertura del intervalo e $I^{TS}(z(k))$ la desdiferenciación de la covarianza difusa, la cual corresponde a la ecuación 6 donde $\Delta\hat{y}_j$ se refiere a la covarianza difusa de la j-ésima regla.

$$\hat{y}(k) = \hat{y} - \alpha I^{TS}(z(k)) \quad (4)$$

$$\bar{y}(k) = \hat{y} + \alpha I^{TS}(z(k)) \quad (5)$$

$$I^{TS}(z(k)) = \sum_j h_j(z(k)) \Delta\hat{y}_j \quad (6)$$

Para obtener el valor de la covarianza y ajustar el valor de α se usó el dataaset de test. Para ajustar α se realizó una exploración lineal partiendo en $\alpha = 0.1$ con paso 0.1, computando el porcentaje efectivo de cobertura y detenerse cuando llegue al valor deseado. La figura 9 muestra los resultados obtenidos al sintonizar α , donde se puede verificar cómo el porcentaje de cobertura se acerca progresivamente al deseado a medida que aumenta el parámetro, llegando hasta el 90 %.

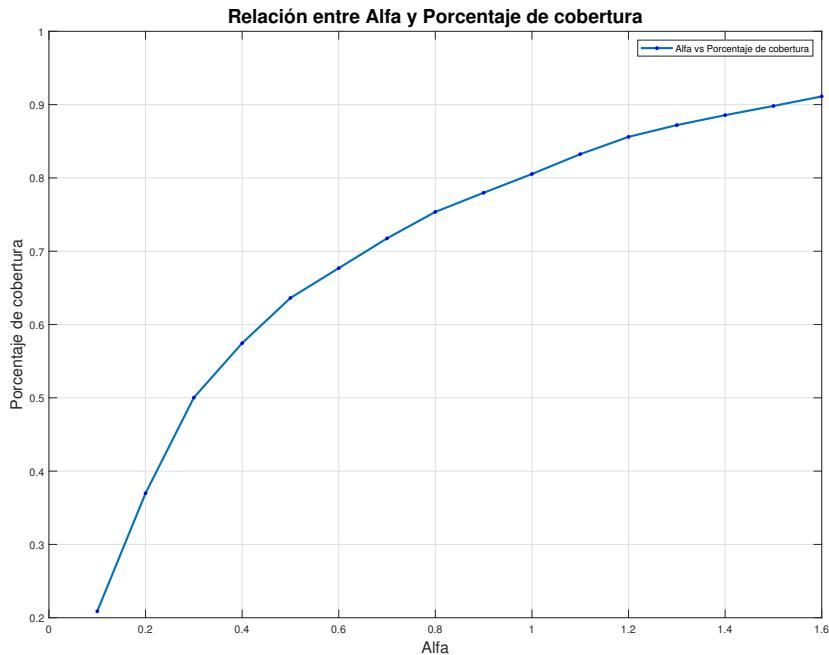


Figura 9: Alfa en función del porcentaje de cobertura para 1 paso para TS

A partir de las funciones de intervalos construidas anteriormente se generaron predicciones a 1, 8

y 16 pasos con intervalos difusos sobre el dataset de validación, considerando un valor de α distinto para cada horizonte de predicción. Además de inspección visual en las predicciones se consideraron las métricas de error señaladas en la sección anterior con el fin de analizar cómo se mueven el error de la predicción media y las métricas de cobertura efectiva con este nuevo conjunto de datos.

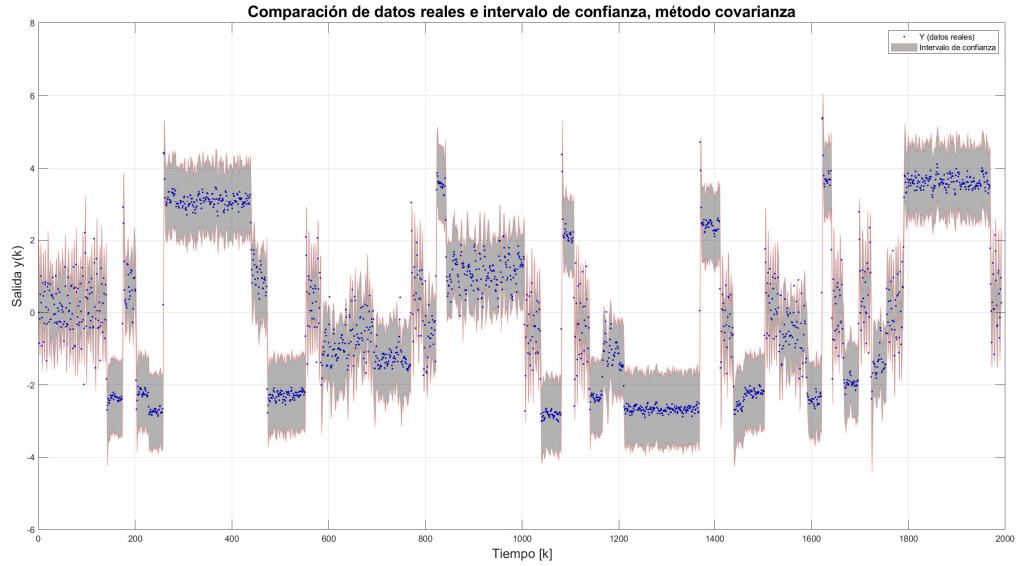


Figura 10: Intervalos calculados con el método de la covarianza TS a 1 paso

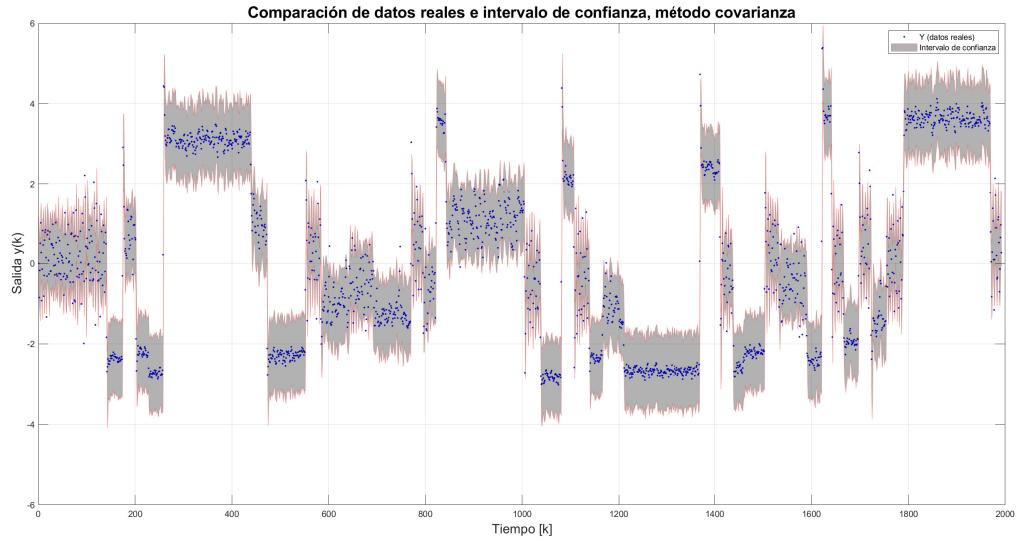


Figura 11: Intervalos calculados con el método de la covarianza TS a 8 pasos

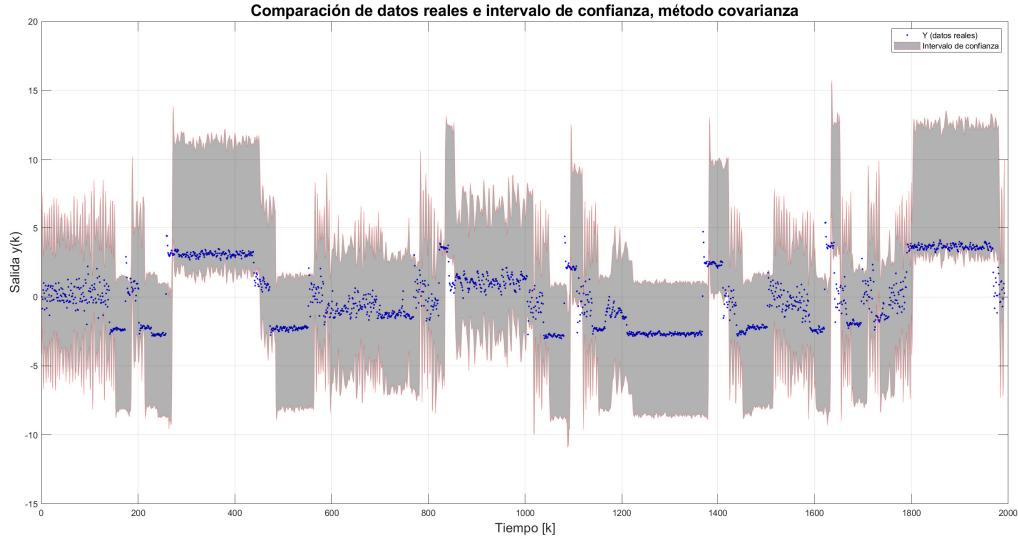


Figura 12: Intervalos calculados con el método de la covarianza TS a 16 pasos

A partir de las figuras 10, 11 y 12 se puede observar cómo los intervalos de predicción aumentan a medida que realizamos predicciones más adelante en el futuro. Dicho resultado se condice con lo esperado, pues es razonable que a medida que predecimos más instantes a futuro, mayor debe ser la incertidumbre asociada a dicha predicción.

También se puede verificar que la mayoría de los datos efectivamente se encuentran en el intervalo de predicción para el dataset de validación, lo que a priori permite afirmar que los intervalos construidos son efectivamente informativos. En secciones posteriores se comparará de forma más precisa estos resultados para los otros métodos de predicción mediante intervalos.

Para el modelo basado en MLP también se generaron predicciones usando el método de la covarianza, definiendo cotas superiores e inferiores para cada predicción siguiendo las ecuaciones 4 y 5, con a diferencia que para estos modelos el término $I^{TS}(z(k))$ sigue la ecuación 7, donde Z^* corresponde a la matriz de salida de la capa oculta para cada dato en el conjunto de entrenamiento, mientras que Z_k hace referencia a los valores de la capa oculta para el k-esimo valor del conjunto en el que se está computando el método. Por último, σ_e corresponde a la raíz de la varianza del ruido en las muestras de test.

$$I^{NN}(z(k)) = \sigma_e(1 + Z_k^T (Z^* Z^{*T})^{(-1)} Z_k^T)^{\frac{1}{2}} \quad (7)$$

A partir de ello volvemos a sintonizar el parámetro α para las cotas superior e inferior siguiendo el método descrito anteriormente, llegando así a los resultados de la figura 13, obteniendo un porcentaje de cobertura del 90 % para el conjunto de test.

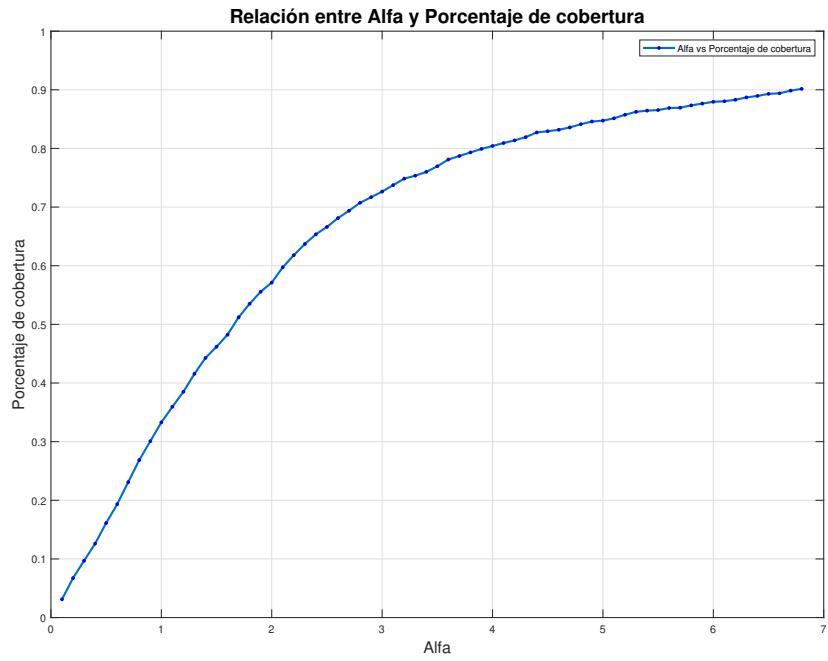


Figura 13: Alfa en función del porcentaje de cobertura para 1 paso para NN

Con todos los parámetros sintonizados se procedió a generar predicciones a 1, 8 y 16 pasos de forma concordante con la parte anterior, computando métricas de error y porcentaje de cobertura. Para este parte se verifica que, al igual que para redes neuronales los intervalos efectivamente cubren gran parte de la data de validación, sin embargo, también se observa cómo el intervalo efectivamente crece a medida que se predice a más pasos en el futuro.

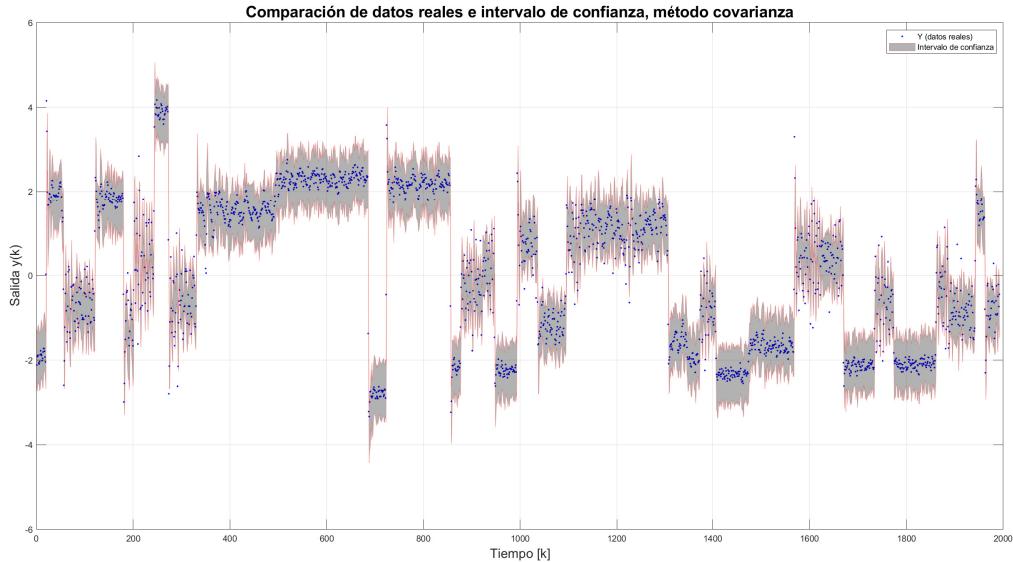


Figura 14: Intervalos calculados con el método de la covarianza NN a 1 pasos

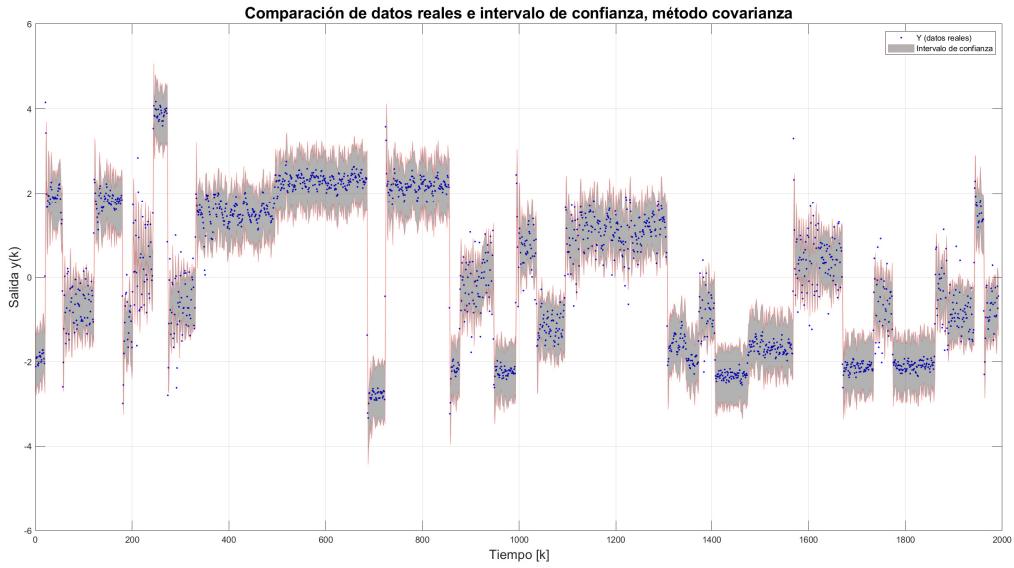


Figura 15: Intervalos calculados con el método de la covarianza NN a 8 pasos

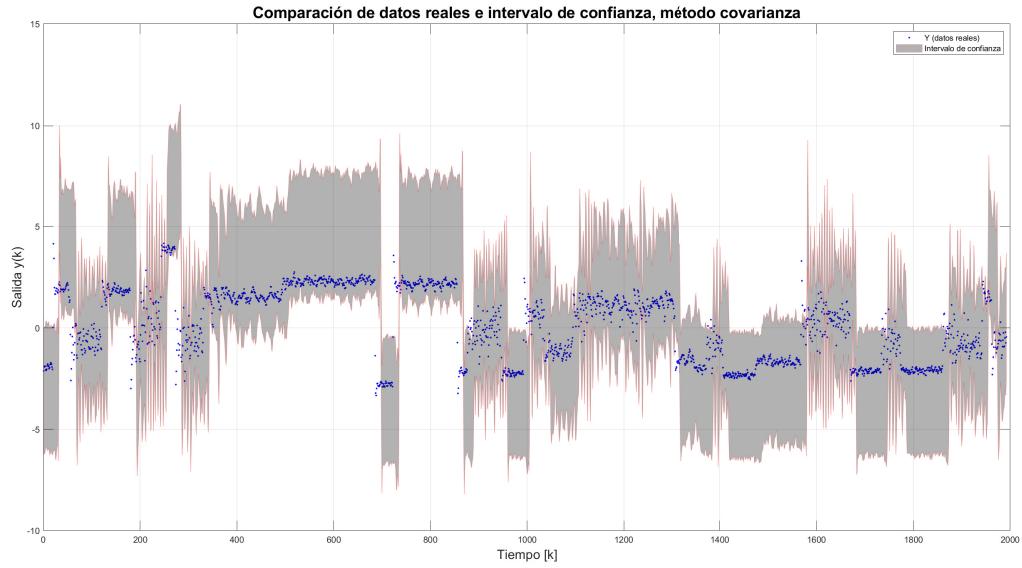


Figura 16: Intervalos calculados con el método de la covarianza NN a 16 pasos

Parte e

Tras ajustar modelos capaces de generar intervalos difusos mediante el método de la covarianza se procedió a usar intervalos difusos con la intención de comparar el desempeño de ambos métodos con modelos tanto difusos como neuronales con el dataset de validación.

Para construir los intervalos con modelos difusos es necesario considerar que cada parámetro de la consecuencia es modelado como un número difuso tal que $a_i^j = [m_i^j - \underline{s}_i^j, m_i + \bar{s}_i^j]$ con m_i^j el valor

medio del i-ésimo parámetro y s_i^j las cotas superiores e inferiores del intervalo para el i-esimo parámetro, para la j-ésima regla. Al ser un método secuencial, primero se debe entrenar un modelo tipo Takagi-Sugeno convencional con el fin de obtener los valores de los parámetros de las consecuencias y en base a ello calcular los valores de los límites, también llamados *spreads*. De esta forma se obtienen dos modelos TS para predecir las cotas inferiores y superiores del intervalo difuso, las consecuencias de las j-ésimas reglas se obtienen a partir de las ecuaciones 8 y 9, respectivamente.

$$\hat{y}_L(k) = \sum_{i=1}^p m_i^j z_i(k) + m_0^j - \sum_{i=1}^p |z_i(k)| \underline{s}_i^j \quad (8)$$

$$\hat{y}_U(k) = \sum_{i=1}^p m_i^j z_i(k) + m_0^j + \sum_{i=1}^p |z_i(k)| \bar{s}_i^j \quad (9)$$

Para el caso de modelos neuronales los modelos para las cotas inferiores y superiores se obtienen a partir de las ecuaciones 10 y 11.

$$\hat{y}_L(k) = \sum_{j=1}^L m_j \tilde{Z}_j(k) + b_0 - \sum_{j=1}^L |\tilde{Z}_j(k)| \underline{s}_j \quad (10)$$

$$\hat{y}_U(k) = \sum_{j=1}^L m_j \tilde{Z}_j(k) + b_0 + \sum_{j=1}^L |\tilde{Z}_j(k)| \bar{s}_j \quad (11)$$

El proceso de sintonización de los *spreads* consiste en encontrar los valores que generan el menor ancho promedio en los intervalos, PINAW, sujeto al cumplimiento de la probabilidad de cobertura deseada (PICP). Con lo anterior, el problema de optimización es el siguiente:

$$\begin{aligned} & \min_{\underline{s}(k), \bar{s}(k)} PINAW \\ & st. PICP = 1 - \alpha \end{aligned} \quad (12)$$

Donde los valores de PINAW y PICP se calculan con las ecuaciones 13 y 14, respectivamente.

$$PINAW = \frac{1}{N \cdot R} \sum_{k=1}^N (\hat{y}_U(k) - \hat{y}_L(k)) \cdot 100 \% \quad (13)$$

$$PICP = \frac{1}{N} \sum_{k=1}^N \delta_k \cdot 100 \% \quad (14)$$

Resolver este problema de optimización puede ser complejo, por lo que su resolución típicamente se logra mediante algoritmos iterativos como PSO. Además, se relaja la condición de $PICP = 1 - \alpha$ y se agrega al funcional que se quiere minimizar, con un factor de penalización asociado. Es así como se forma la ecuación 15, la que se termina usando como función a minimizar en el método *particleswarm()* de Matlab que implementa el algoritmo de optimización de PSO.

$$J = \eta_1 \cdot PINAW + e^{-\eta_2(PICP - (1 - \alpha))} \quad (15)$$

Tras sintonizar los parámetros se generaron predicciones a 1, 8, y 16 pasos con el dataset de test, al igual que para el caso del método de la covarianza, resultando así en las figuras 17, 18 y 19, donde se tienen las predicciones de los intervalos difusos de un modelo TS sobre el conjunto de validación

como benchmark, ya que el conjunto de test fue utilizado para encontrar los valores de los spreads.

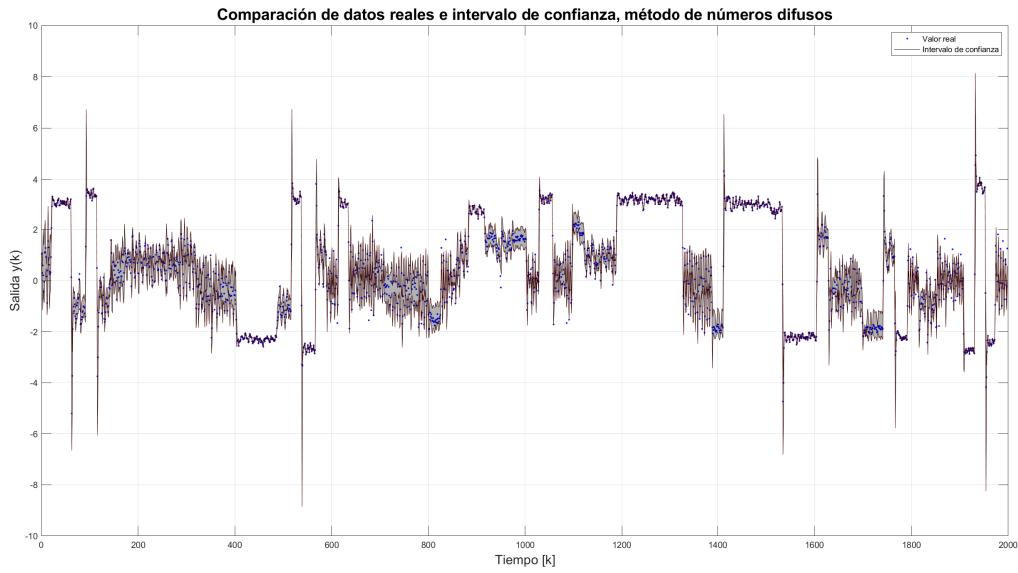


Figura 17: Intervalos de confianza con números difusos con predicción a 1 paso de modelo TS.

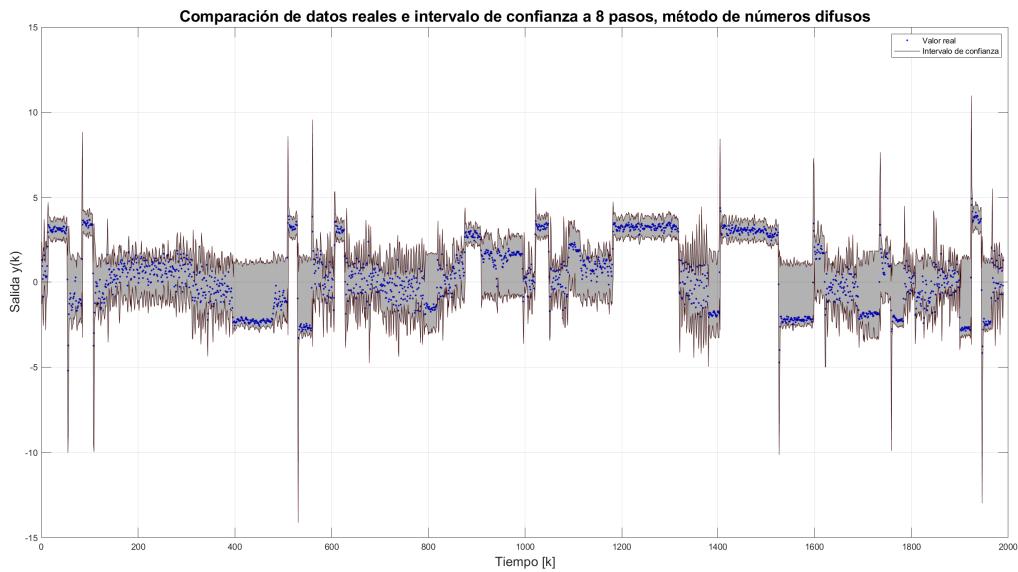


Figura 18: Intervalos de confianza con números difusos con predicción a 8 pasos de modelo TS.

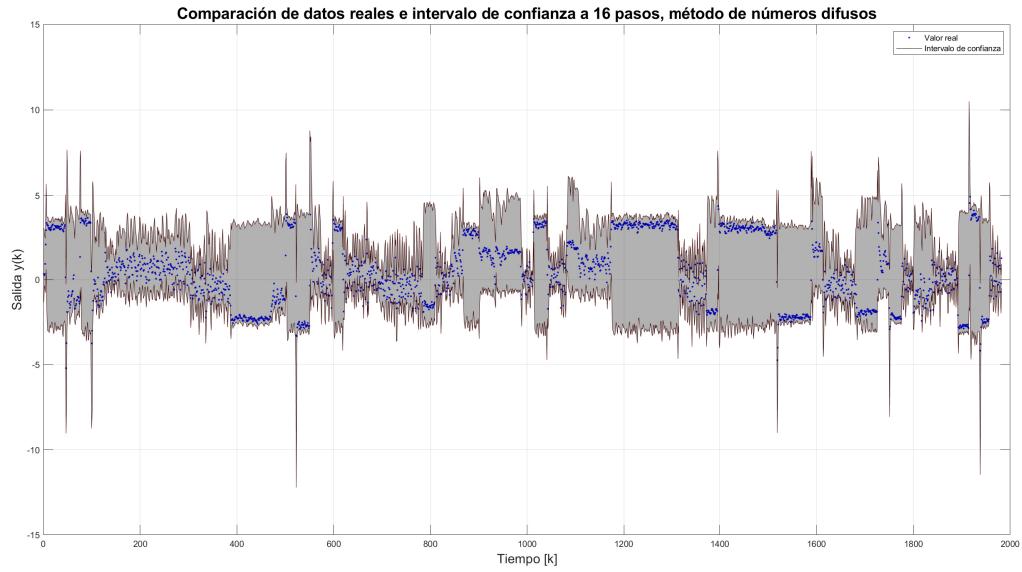


Figura 19: Intervalos de confianza con números difusos con predicción a 16 pasos de modelo TS.

De igual manera se obtienen las figuras 20, 21 y 22 que muestran los intervalos difusos de un modelo NN sobre el conjunto de validación como benchmark.

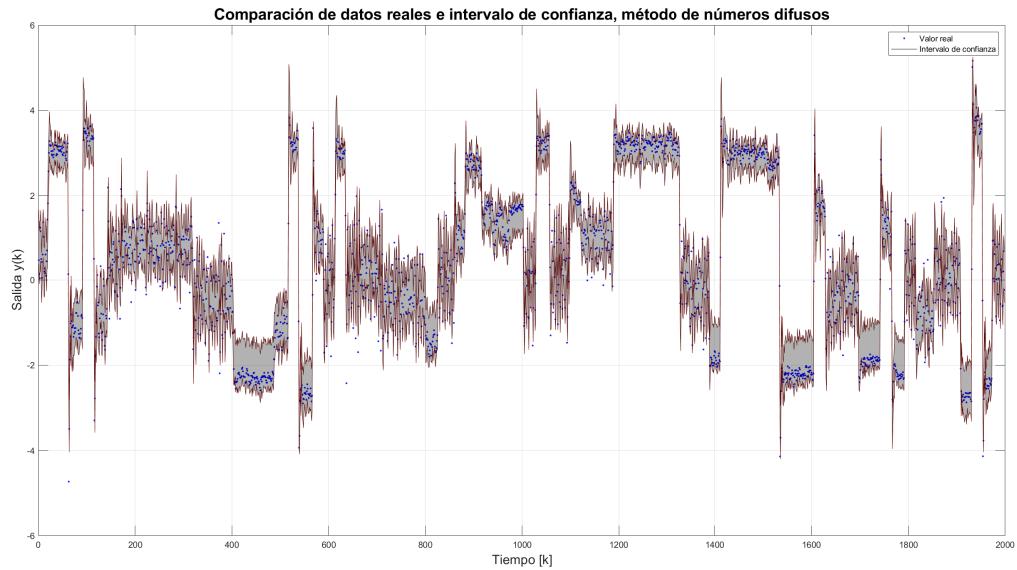


Figura 20: Intervalos de confianza con números difusos con predicción a 1 paso del modelo NN.

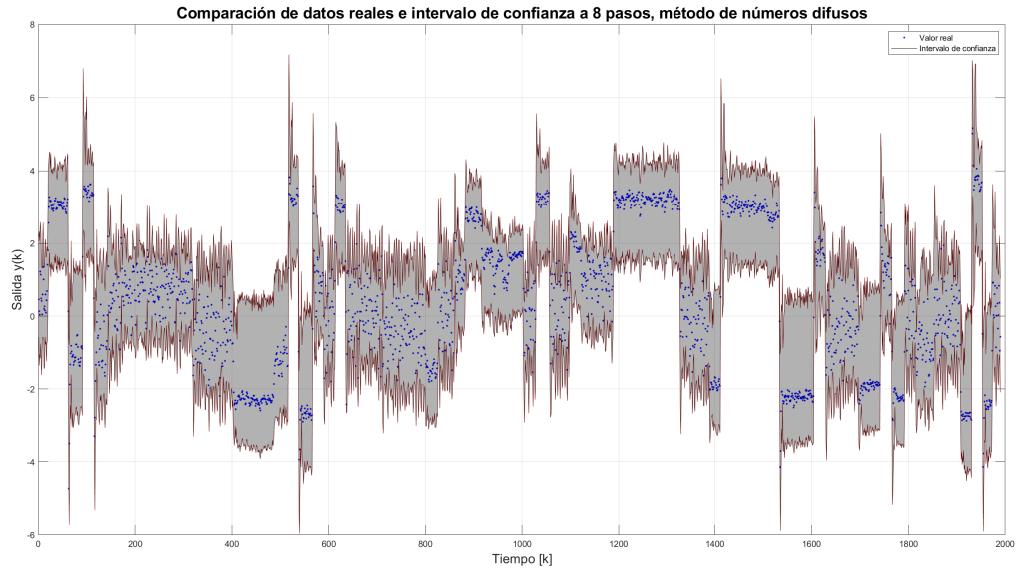


Figura 21: Intervalos de confianza con números difusos con predicción a 8 pasos del modelo NN.

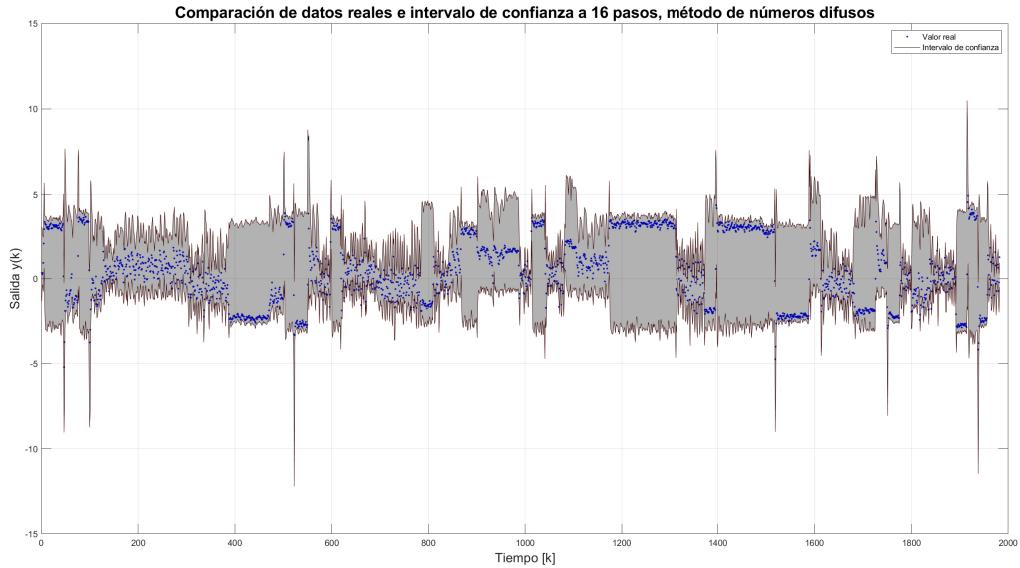


Figura 22: Intervalos de confianza con números difusos con predicción a 16 pasos del modelo NN.

Además se calculan los PINAW y PICP resultantes de todas las predicciones, estos resultados se resumen en la tabla 3.

De los resultados se puede apreciar cómo se ensancha el intervalo de predicción a medida que predecimos más intervalos en el futuro, este comportamiento es esperable dado que al predecir más

a futuro los resultados tendrán más incertidumbre. Esto se nota en los gráficos y es evidente de la tabla 3, donde el PINAW crece considerablemente para mantener la condición de que la cobertura sea del 90 %. También se observa que, a diferencia de los intervalos obtenidos por el método de la covarianza, números difusos entrega intervalos más acotados en zonas donde la salida del sistema no tiene mucho ruido, y ensancha los intervalos cuando la salida es más errática.

Parte f

Para comparar el desempeño de los distintos modelos se computaron métricas de MAE y RMSE, a partir de las predicciones del modelo base, y PICP y PINAW para los intervalos difusos computados en las secciones anteriores. A partir de las tablas 2 y 3 se puede observar cómo el modelo Takagi-Sugeno tiene una mejor métrica en el error de predicción que la red neuronal, tal como se comentó en la parte c del reporte.

En cuanto a los intervalos, se verifica que para los distintos modelos el PICP se mantiene cercano al valor impuesto, lo que a priori asegura un intervalo correcto, sin embargo también se debe considerar el ancho del intervalo. En términos de PINAW, se puede verificar que para ambos modelos el intervalo más acotado es logrado mediante números difusos, siendo Takagi-Sugeno el mejor partidario en todos los horizontes de predicción.

En base a lo mencionado anteriormente, se puede concluir que el mejor tipo de modelo basado en intervalos para este dataset es el Takagi-Sugeno con números difusos, tanto por su precisión en las predicciones, como por el ancho de los intervalos resultantes.

Horizonte de Predicción	Métrica	Takagi-Sugeno	Neural Network
1 paso a futuro	RMSE	0.2761	0.3443
	MAE	0.1566	0.2336
	PICP (%)	90.71	90.16
	PINAW(%)	24.00	31.97
8 pasos a futuro	RMSE	0.6414	1.3322
	MAE	0.3818	0.7767
	PICP (%)	90.86	90.11
	PINAW(%)	22.58	28.77
16 pasos a futuro	RMSE	0.6408	1.8380
	MAE	0.3815	1.1491
	PICP (%)	90.01	90.01
	PINAW(%)	104.25	100.58

Tabla 2: Métricas de rendimiento para modelos TS y NN a 1, 8 y 16 pasos considerando intervalos por método de la covarianza.

Horizonte de Predicción	Métrica	Takagi-Sugeno	Neural Network
1 paso a futuro	RMSE	0.2761	0.3443
	MAE	0.1566	0.2336
	PICP (%)	90.84	90.64
	PINAW(%)	6.51	10.53
8 pasos a futuro	RMSE	0.6414	1.3322
	MAE	0.3818	0.7767
	PICP (%)	89.80	89.75
	PINAW(%)	30.36	41.84
16 pasos a futuro	RMSE	0.6408	1.8380
	MAE	0.3815	1.1491
	PICP (%)	90.36	89.66
	PINAW(%)	48.70	70.24

Tabla 3: Métricas de rendimiento para modelos TS y NN a 1, 8 y 16 pasos considerando intervalos por números difusos.

Pregunta 3

Parte a

El sistema elegido para el ejercicio es el levitador magnético, este es un sistema no lineal altamente estudiado en la literatura que cuenta con datos tanto de entrada como de salida, esta característica lo hace ideal para este tipo de análisis. Su funcionamiento se describe en la figura 23 donde existen divisiones entre la planta en sí y los actuadores con tal de dar detalles del funcionamiento interno, sin embargo, el comportamiento se puede reducir a una planta que tiene como entrada una corriente y una altura h como salida.

Para este ejercicio se trabajó con una planta real, en particular el levitador magnético de la marca Feedback Modelo 33-210, equipo de donde provienen los datos de entrada y salida analizados en las preguntas futuras.

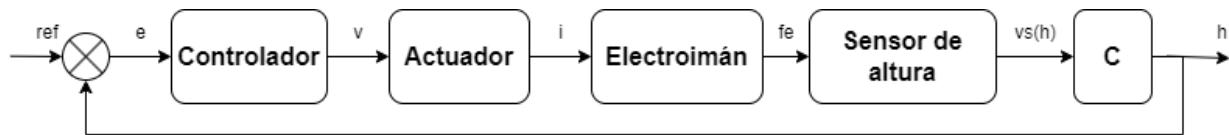


Figura 23: Diagrama de bloques del levitador magnético

El sistema se alimenta de un voltaje de entrada que energiza un electroimán para generar la fuerza que eleva la bola de metal y en consecuencia, se obtiene la altura que se busca controlar.

Este sistema aparte de ser no lineal es inherentemente inestable, para su identificación es necesario el uso de un controlador base para analizar la planta a lazo cerrado.

Con el controlador a lazo cerrado se comenzó analizando la respuesta del sistema en distintos rangos de operación, obteniendo los resultados de la figura 24

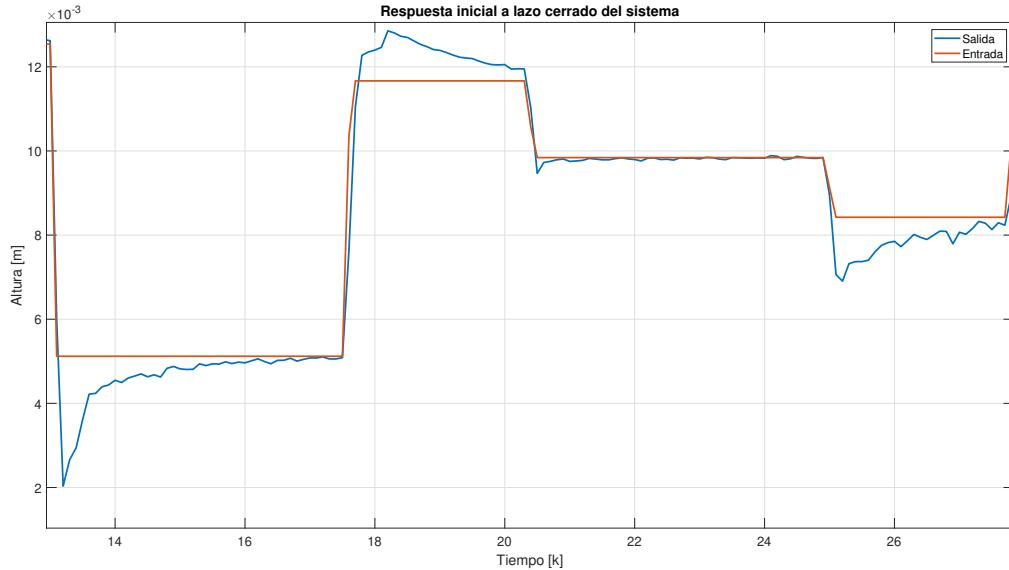


Figura 24: Respuesta a lazo cerrado para identificación de parámetros

Los datos fueron recopilados entre el 12 de abril hasta el 19 de abril de 2023.

Los parámetros extraídos del sistema son:

- Tiempo de establecimiento $\approx 2[s]$
- Rango de operación de entrada entre $-2.2[V]y - 0.8[V]$

La utilidad de analizar este tipo de sistemas son las potenciales aplicaciones que tienen en los sistemas mecánicos. La eliminación del roce es una de sus características más destacables, al tener la capacidad de ejercer fuerza sin necesidad de contacto, se puede eliminar el roce y reducir en gran medida el desgaste de piezas de un sistema que utilice levitación magnética. Tal es el caso de los trenes bala, estos trenes tienen un riel magnético que le permite al usuario alcanzar velocidades de hasta $580 \frac{km}{h}$ gracias a la eliminación del roce con el riel.

Parte b

Para esta parte del ejercicio se elige la construcción de un modelo difuso Takagi-Sugeno. Se eligió esta opción por su capacidad de modelar sistemas altamente no lineales como el del caso de estudio.

Se comienza con el diseño del experimento de identificación. El objetivo es obtener data representativa del sistema en los rangos de operación seleccionados. En particular para este sistema se notaron particularidades en el método de obtención de data real:

- Los primeros 15[s] de simulación no hay seguimiento de referencia, posiblemente por calibración interna del sistema
- Los datos recopilados del sistema se sobreescreiben cada 60[s], razón por la cual se extraen 5 simulaciones de 60 segundos que posteriormente son concatenadas para la correcta identificación del sistema

- El tiempo de establecimiento aumenta al acercarse a los límites de operación del sistema, por lo que la frecuencia de la señal APRBS se redujo.
- El tiempo de muestreo del sistema es de $0.01[S]$

En respuesta a estos problemas se procedió a obtener 5 datasets distintos de 60 segundos con una APRBS con límites más aterrizados en comparación a los valores obtenidos inicialmente, se eliminaron los primeros 1800 datos y además, debida a la alta frecuencia con la que el sensor genera muestras, se generó 1 dato nuevo promediando cada 10 puntos muestreados para reducir la cantidad de datos y aumentar el tiempo de muestreo a $0.1[s]$.

La señal APRBS utilizada como experiencia de identificación cuenta con las siguientes características:

- Tiempo de establecimiento $\approx 2 - 3[s]$
- Rango de operación de entrada entre $-2.1[V]y - 0.9[V]$
- Frecuencia Mínima $0.2[Hz]$
- Frecuencia Máxima $0.4[Hz]$

Una muestra de la respuesta del sistema a la señal APRBS se observa en la Figura 25.

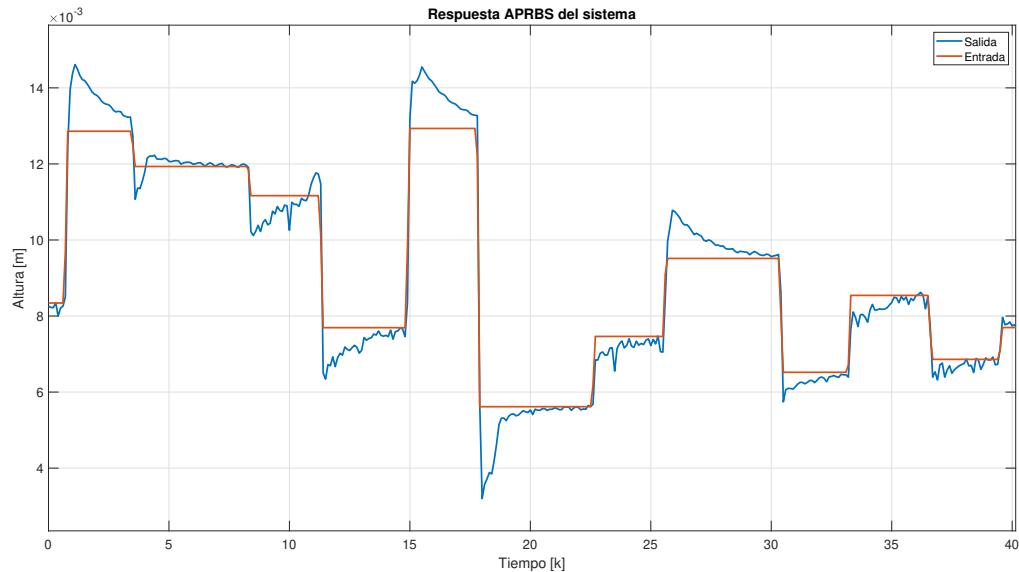


Figura 25: Respuesta a señal APRBS del sistema

Con la data obtenida se tienen cerca de 10.000 datos para su análisis, se divide la data en entrenamiento-test-validación en los porcentajes 60 %-20 %-20 % respectivamente.

Utilizando el conjunto de entrenamiento se obtuvieron clusters difusos con los que luego se pueden computar los grados de pertenencia de los distintos regresores a las reglas definidas preliminarmente, esto se traduce en un modelo ARX con distintas variables que es sintonizado por mínimos cuadrados.

Para optimizar la estructura del modelo se inició por buscar el número óptimo de clusters, fijando de forma preliminar la estructura del modelo con 10 autoregresores y 10 regresores para la variable exógena.

Cabe destacar que como propuesta inicial se trata de cubrir la mayor cantidad de regresores del sistema, sin embargo, al sobrepasar los 4 regresores por variable, el error al optimizar la cantidad de clusters aumenta sustancialmente y el rango de la matriz es deficiente, lo que da a entender que pasado 4 regresores hay dependencias lineales en el sistema.

Posteriormente, se fue variando la cantidad de clusters del sistema entre 2 y 4, obteniendo distintos modelos cuyo desempeño fue evaluado con el dataset “test”, Usando como métrica de medición el RMSE de la señal. El desempeño de los modelos se observa en la figura 26 y se concluye que la cantidad óptima de clusters es de 2 por regla.

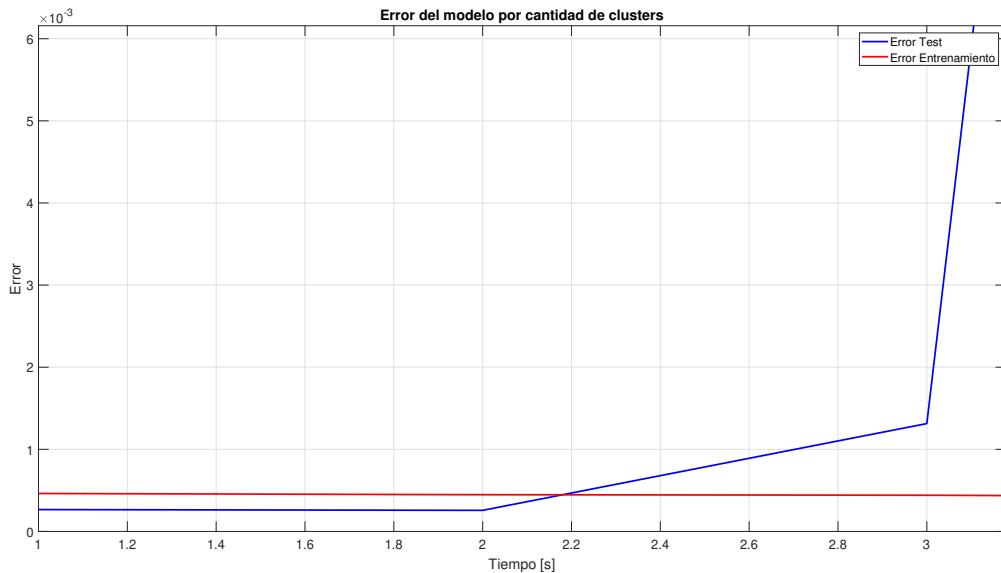


Figura 26: Error del sistema en función de la cantidad de clusters

Con la cantidad óptima de clusters ya definida se procedió a optimizar la cantidad de regresores que conforman el modelo. Esto se logró mediante un análisis de sensibilidad de la salida con respecto a cada regresor dado a priori.

Se calcula la sensibilidad de la salida con respecto a cada regresor para cada dato en conjunto de validación y se construye una métrica de importancia de la forma $\bar{\delta}^2 + \sigma_{\delta}^2$ con $\bar{\delta}$ el promedio de las sensibilidades y σ_{δ}^2 la desviación estándar. Posteriormente se elimina iterativamente aquellos regresores que disminuyan el RMSE del sistema.

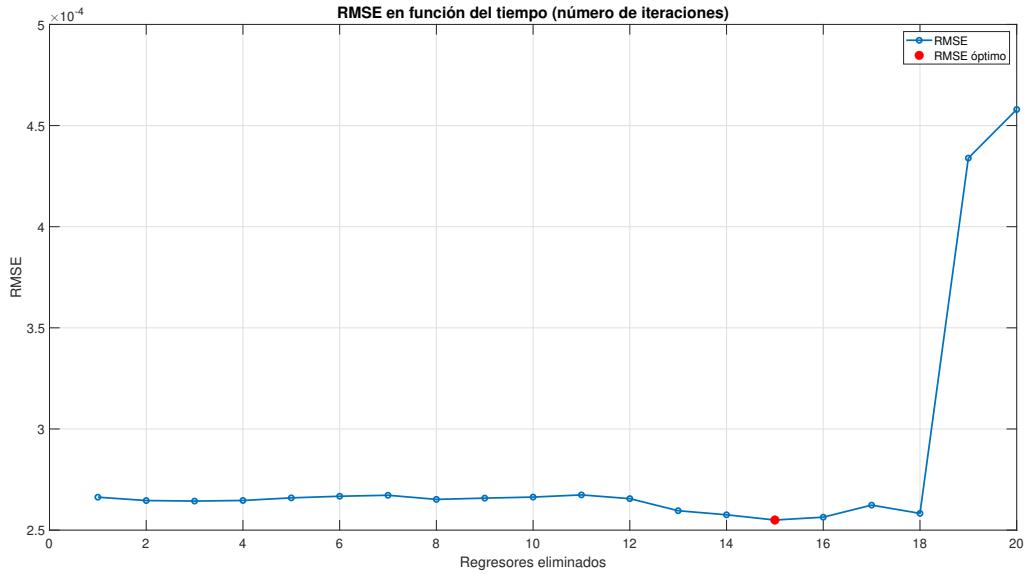


Figura 27: Errores RMSE del sistema en función de la cantidad de regresores

El desempeño del sistema en base al numero de regresores se observa en la Figura 27 donde el número óptimo de regresores se obtiene al eliminar 15 de los 20 iniciales. La evolución de regresores del sistema se observa en la Figura 28

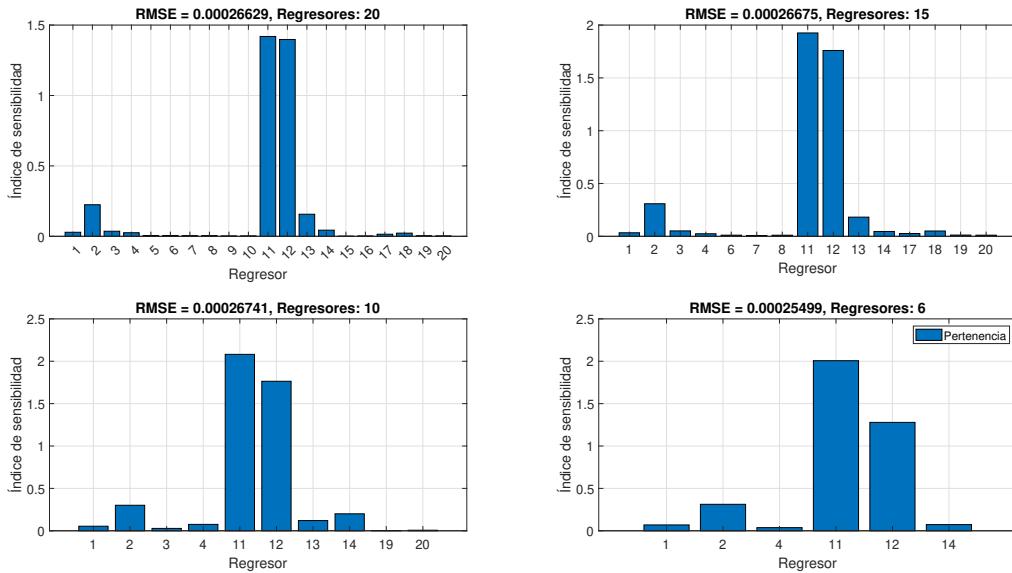


Figura 28: Eliminación de regresores en base al RMSE del modelo

Parte c

Para evaluar el desempeño del modelo se decide utilizar las métricas MAE y RMSE, ambos dado que dan resultados en la misma unidad que la variable que se quiere controlar y no discriminan si el error es positivo o negativo, lo que otorga resultados intuitivos de analizar. Además, por un lado el

RMSE da gran importancia a los errores altos y no tanto a los bajos, falencia que se puede compensar con el uso del MAE, que mide de forma absoluta el error y en consecuencia, castiga todos los errores por igual.

Los valores de RMSE y MAE son $0.4 \cdot 10^{-3}$ y $3.987 \cdot 10^{-7}$ respectivamente, estos valores son particularmente bajos, lo que da a entender que el modelo Takagi-Sugeno es bastante acertado para predecir los resultados. Al mismo tiempo se pueden observar en la figura 29 una visual de los puntos en los cuales el valor del error es más alto, este dato es útil, dado que se sabe que los lugares en los que más falla el modelo es en la predicción de los cambios bruscos producto de la APRBS y del ruido del sistema.

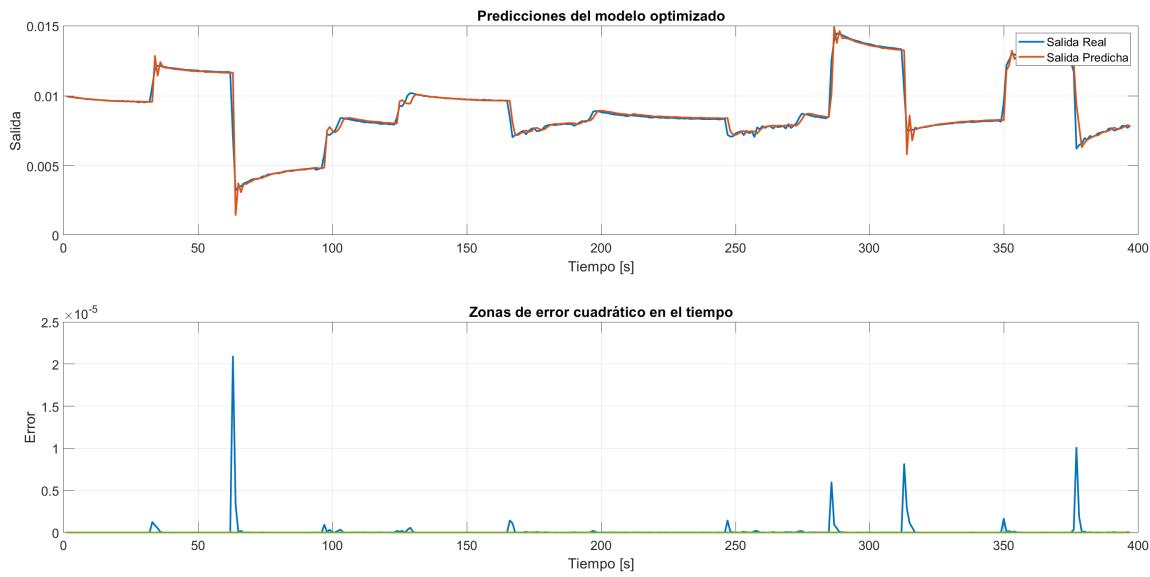


Figura 29: Errores por zona del modelo optimizado

Parte d

Se construyen intervalos de confianza del sistema en base al método de la covarianza. Este método busca generar un intervalo donde un porcentaje dado de los datos exista dentro de dicho intervalo, permitiendo así detectar comportamiento anómalo del sistema al obtener puntos fuera de ese rango.

En la Figura 30 se observa la predicción a 1 paso del modelo utilizando un valor de α tal que el 90 % de los datos se encuentren dentro del intervalo (este valor se mantendrá en futuras implementaciones de intervalos de covarianza).

Nro de pasos	RMSE	MAE
5	$0.8 \cdot 10^{-3}$	$9.5628 \cdot 10^{-7}$
10	$1.3 \cdot 10^{-3}$	$1.6481 \cdot 10^{-6}$
20	$4.0 \cdot 10^{-3}$	$4.6389 \cdot 10^{-6}$

Tabla 4: Métricas de error para las predicciones a n pasos

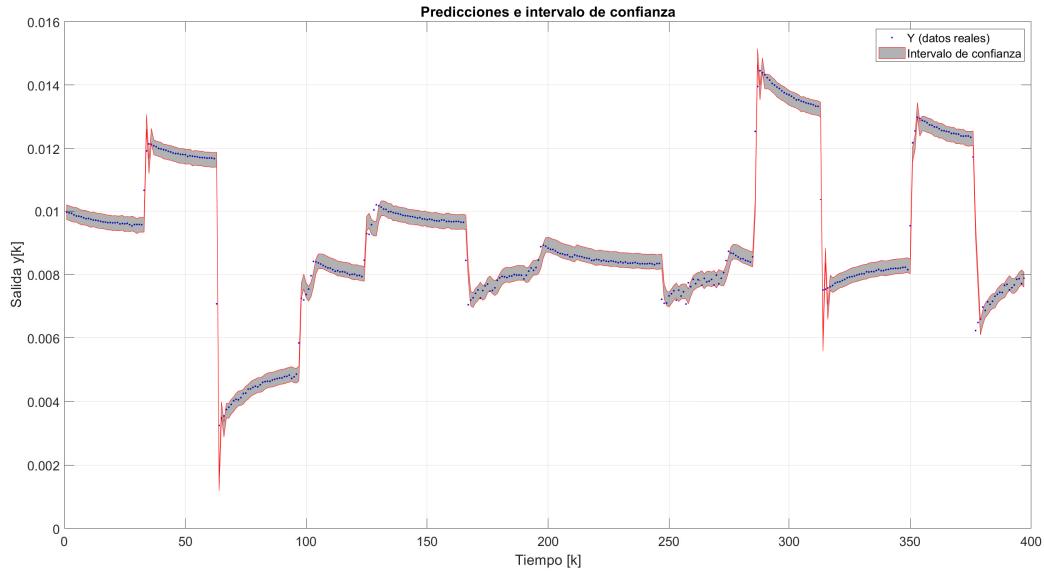


Figura 30: Comparación de salidas e intervalos de confianza

Adicionalmente se realizaron predicciones del sistema para 5, 10 y 20 pasos, esto corresponde a 0.5, 1 y 2[s] respectivamente, tiempo razonable considerando el tiempo de establecimiento del sistema. El desempeño del sistema se observa en la Tabla 4, donde claramente se ve el aumento del error a medida que se aumenta la cantidad de pasos, pero aún así manteniendo valores relativamente bajos incluso para 20 pasos de predicción.

A cada paso se realiza una implementación de intervalos de confianza con método de covarianza. En las Figuras 31, 32 y 33 se observan los intervalos para cada paso, se nota un aumento en la amplitud contenida por el intervalo de covarianza a medida que aumentan los pasos producto del aumento en error que tiene el modelo a mayor cantidad de pasos.

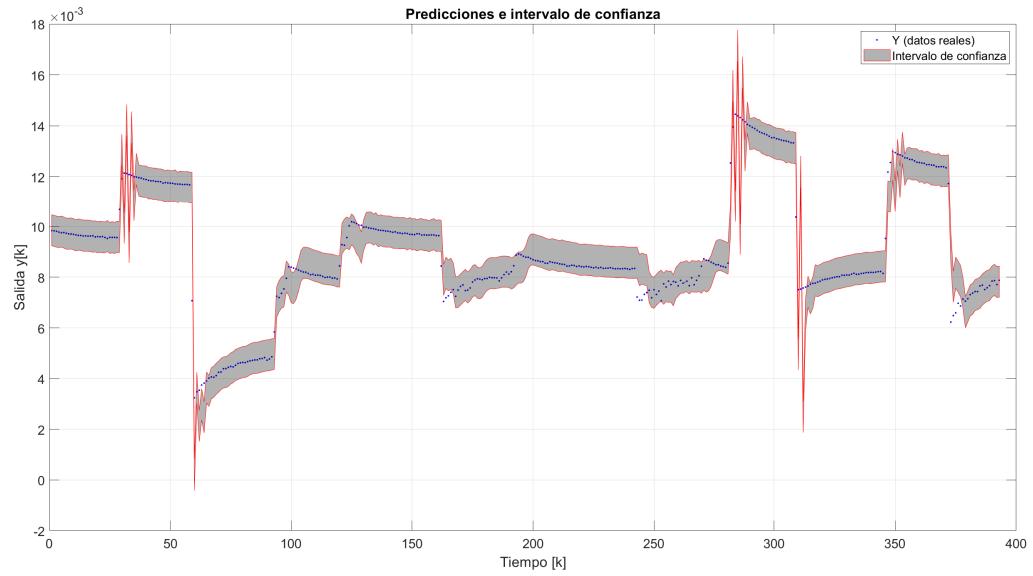


Figura 31: Intervalos de covarianza para 5 pasos

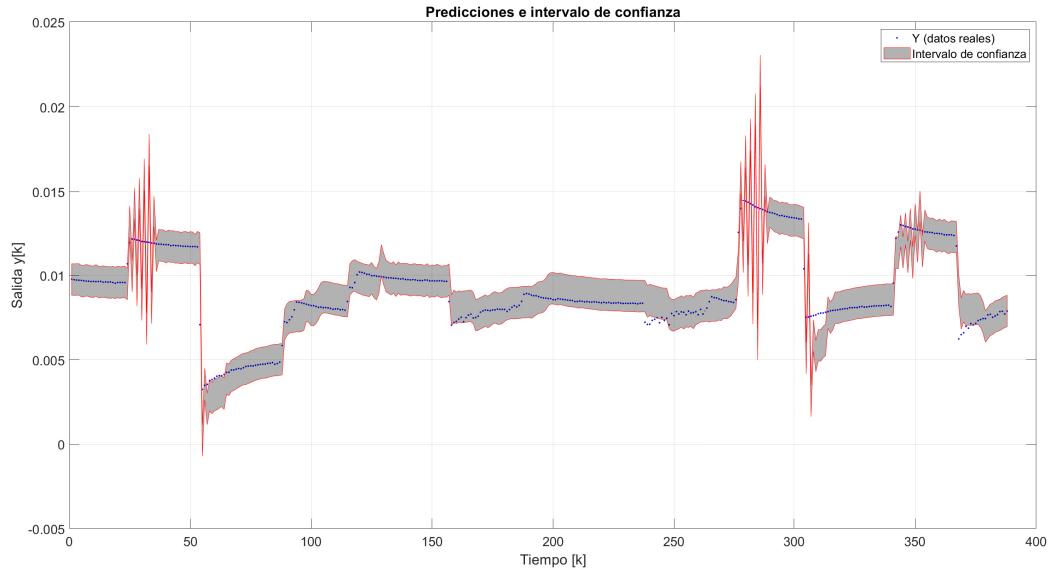


Figura 32: Intervalos de covarianza para 10 pasos

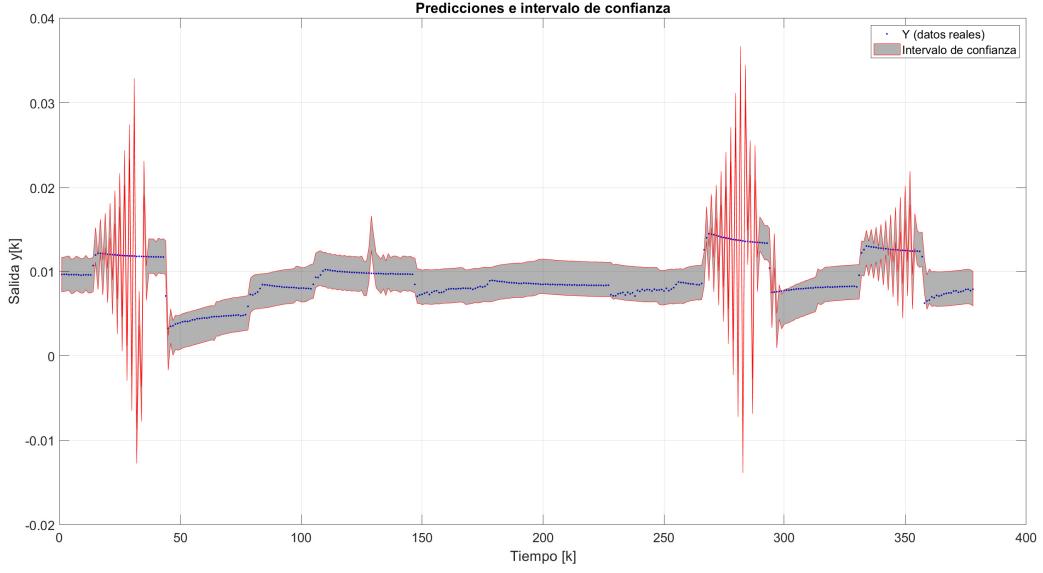


Figura 33: Intervalos de covarianza para 20 pasos

Finalmente se tiene la comparación con la data real del sistema y un muestreo del error por zonas en la Figura 34. Dados los valores del error se pueden extraer varias observaciones de las predicciones del sistema.

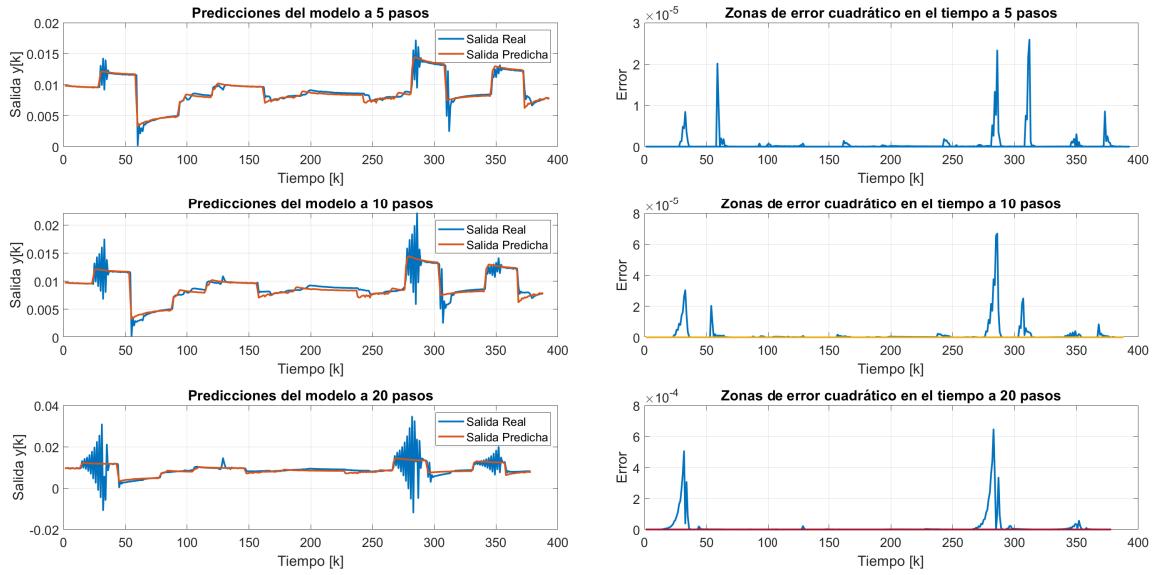


Figura 34: Comparación de predicciones y errores por zona

Primeramente se puede notar que las zonas de error son siempre las mismas, correspondiendo a aquellas muestras donde la APRBS presenta cambios bruscos de amplitud. Por otro lado, se nota que en estas variaciones de amplitud el sistema comienza a subir y bajar crecientemente desde el punto de cambio, dando a entender la presencia de incertidumbre producto de predecir a mayor cantidad de pasos la salida. Esta misma salida posteriormente se ajusta al obtener nueva información del cambio de amplitud pasada la cantidad de pasos configurado.