

## Control de un sistema multivariable no lineal y en fase no mínima empleando un controlador predictivo neuronal

Elmer Calle Chojeda, José Oviden Semino, William Ipanaque Alama\*

Laboratorio de Sistemas Automáticos de Control, Departamento de Ingeniería Mecánico-Eléctrica, Universidad de Piura, Av. Ramón Mugica, n°131, 20001, Piura, Perú.

**To cite this article:** Calle-Chojeda, E., Oviden-Semino, J., Ipanaque-Alama, W. 2023. Control of a non-linear and non-minimum phase multivariable system using a neural predictive controller. Revista Iberoamericana de Automática e Informática Industrial 20, 32-43. <https://doi.org/10.4995/riai.2022.17375>

### Resumen

En este artículo se propone un Controlador Predictivo Neuronal (ANN-MPC) para controlar un sistema no lineal de tanque cuádruple, el cual es complejo de controlar debido a la no linealidad de sus válvulas y a la interacción entre sus variables controladas. Además, el problema se agrava ya que el proceso presenta una respuesta transitoria con dinámica inversa por ser de fase no mínima. El ANN-MPC emplea una estructura modular de red neuronal artificial y el algoritmo de entrenamiento Levenberg-Marquardt para estimar con mayor precisión y rapidez las salidas del proceso no lineal y evitar el sobreajuste del modelo. Se generaron datos operativos a partir de la planta para entrenar la red neuronal empleando Matlab. Se probó el rendimiento del ANN-MPC ante cambios de referencia y se comparó con un MPC lineal y un MPC no lineal. Los resultados de simulación mostraron que el ANN-MPC produjo un menor tiempo de establecimiento que el MPC lineal y generó valores RMSE de las salidas similares a los del NMPC. Además, se redujo el tiempo de cómputo requerido para calcular la variable de control óptima comparado con el NMPC.

*Palabras clave:* Control predictivo basado en modelo, Redes neuronales artificiales, Sistema MIMO, Sistema de tanque cuádruple.

### Control of a non-linear and non-minimum phase multivariable system using a neural predictive controller

#### Abstract

In this paper, a Neural Predictive Controller (ANN-MPC) is proposed to control a nonlinear quadruple tank system, which is complex to control due to the nonlinearity of its valves and the interaction between its controlled variables. In addition, the problem is aggravated since the process presents a transient response with inverse dynamics due to being in a non-minimum phase. The ANN-MPC employs a modular artificial neural network framework and the Levenberg-Marquardt training algorithm to more accurately and quickly estimate nonlinear process outputs and avoid model overfitting. Operational data was generated from the plant to train the neural network using Matlab. The performance of the ANN-MPC under reference changes was tested and compared with a linear MPC and a non-linear MPC. The simulation results showed that the ANN-MPC produced a shorter settling time than the linear MPC and generated RMSE values of the outputs similar to those of the NMPC. In addition, the computation time required to calculate the optimal control variable was reduced compared to the NMPC.

*Keywords:* Model based predictive control, Artificial neural networks, MIMO system, Quadruple-tank system.

## 1. Introducción

En la actualidad, globalmente, una gran cantidad de industrias cuentan con procesos donde se requieren sistemas de con-

trol. Si bien, por lo general, un sistema de control suele ser sencillo, en diversas ocasiones surgen algunas complicaciones, como cuando el proceso es de fase no mínima, lo cual hace que el

\*Autor para correspondencia: [william.ipanaque@udep.edu.pe](mailto:william.ipanaque@udep.edu.pe) Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

empleo de técnicas clásicas de control no sea tan eficiente. Esto se debe principalmente a que un sistema de fase no mínima muestra comportamientos transitorios de dinámica inversa, provocando que la variable controlada actúe en sentido contrario al esperado cuando se aplica la variable de control. Esta situación se torna aún más compleja para sistemas MIMO (multiple input multiple output), ya que mientras más fuerte sea la interacción entre las variables del proceso, menos eficiente será la aplicación de técnicas clásicas de control, necesitando así el empleo de técnicas avanzadas de control MIMO. Finalmente, una complicación que se presenta en la mayoría de procesos en la práctica es cuando el proceso posee un comportamiento altamente no lineal, lo cual imposibilita el uso de técnicas de control basadas en modelos lineales, pues su desempeño no sería el óptimo.

Un proceso que cuenta con todos los inconvenientes mencionados líneas arriba, y que ha sido ampliamente empleado para ilustrar diversas estrategias de control, es el proceso de tanque cuádruple, estudiado en algunos trabajos como (Johansson, 2000). Este proceso incluso se usa como una herramienta educativa para enseñar técnicas avanzadas de control MIMO (Alvarado et al., 2011) y tiene la ventaja de poseer ceros ajustables según lo cual puede mostrar características de fase mínima y fase no mínima. Este sistema, en particular, es más complejo de controlar cuando se encuentra en configuración de fase no mínima, ya que en esta configuración los tanques ubicados en la parte superior de la planta son alimentados con un mayor caudal de agua, mientras que los tanques inferiores reciben un menor caudal de los actuadores, y dado que el objetivo de control es precisamente la regulación del nivel en los tanques inferiores, el control de nivel en estos últimos se vuelve tedioso.

Una técnica de control que ha mostrado grandes ventajas frente a los problemas antes mencionados es el Control Predictivo Basado en Modelo (MPC). Este controlador emplea el modelo dinámico del proceso para predecir su comportamiento futuro (Kouvaritakis and Cannon, 2016). Por lo cual, el modelo del proceso es un factor clave en el diseño del MPC. Sin embargo, en el caso de un MPC lineal, el modelo empleado en su formulación a menudo se logra aproximando el proceso con modelos lineales, los cuales solo trabajan de manera eficiente dentro de puntos de operación limitados. La linealización del modelo de un proceso genera un desajuste estructural entre el modelo y el proceso, principalmente porque los modelos lineales por lo general no son precisos ante procesos con fuertes no linealidades. Como consecuencia, un MPC basado en linealización tendrá un desempeño deficiente (Pan and Wang, 2012). Una solución alternativa podría ser el uso de un MPC no lineal (NMPC), que emplea modelos no lineales y considera restricciones no lineales en su formulación (Dong et al., 2019). Sin embargo, el NMPC requiere emplear un optimizador en línea para obtener los estados y entradas estacionarios óptimos, y esto hace que el sistema de control requiera más tiempo de cómputo para obtener las variables de control. Este inconveniente hace que el NMPC sea inutilizable con ciertos procesos que poseen periodos de muestreo demasiado cortos.

Ante esta problemática, las Redes Neuronales Artificiales (ANN) han surgido como una posible y prometedora solución. En los últimos años, se han usado diversas redes neuronales junto con MPC, y su uso se ha clasificado en dos categorías: para identificación de modelos (Vieira et al., 2005), (Tang and

Daoutidis, 2019); y para resolver el problema de optimización (Gao et al., 2011), (Pan and Wang, 2010). En el caso de identificación de modelos, se han empleado las ANN como modelos sustitutos en lugar de los modelos lineales usados para predecir las respuestas futuras de los sistemas, como en (Carvalho et al., 2020), donde se entrenó una ANN *Feed-forward* que se usó para aproximar la dinámica de un intercambiador de calor y con la cual se obtuvo el modelo para el diseño de un MPC. Los resultados de aquel trabajo demostraron el buen desempeño del enfoque presentado al conseguir menor sobre-impulso y mejor rechazo a las perturbaciones en comparación con un PID clásico. Sin embargo, el sistema en el cual se probó el controlador era SISO (Single-input Single-output).

Por otra parte, otros trabajos han utilizado las ANN para aproximar la dinámica del MPC y así obtener la variable de control a partir de datos en lazo cerrado. Esta estrategia se enfoca en el mapeo de los estados del sistema y las entradas de control óptimas. Entre los trabajos que han empleado el enfoque mencionado, encontramos a (Cseko et al., 2015), donde se diseñó un MPC basado en funciones de base radial (RBF) para el control de un sistema de suspensión de un vehículo, o el trabajo realizado por (Lucia and Karg, 2018), donde se emplea una estructura *Deep Learning* para modelar la ley de control de un NMPC robusto y así controlar un reactor de polimerización. Por su lado, (Sarali et al., 2019) hace básicamente lo mismo, pero lo aplica a un inversor de fuente de voltaje. Asimismo, en un trabajo anterior, (Calle and Oviden, 2021a), implementaron la antes mencionada estrategia para identificar el MPC empleado para controlar un sistema MIMO de cuatro tanques acoplados. Se realizó una comparación de diferentes estructuras para identificar el controlador, y los mejores resultados los obtuvieron una red *feed-forward* y una red recurrente, mientras que una red LSTM obtuvo el resultado más bajo. Sin embargo, a pesar de los buenos resultados de estos trabajos, es necesario mencionar que este enfoque limita los escenarios de operación del controlador para los cuales fue entrenado, pues se restringe a las situaciones en donde se produjeron los datos de entrenamiento.

En tanto, otros estudios han empleado técnicas de aprendizaje automático para sintonizar los parámetros del mejor MPC que satisfaga las restricciones de entrada y salida. Como en (Yan and Xu, 2019), donde se emplean las ANN para encontrar los parámetros del MPC usado para controlar un sistema de micro-posicionamiento. Mientras que (Piga et al., 2019), a partir de datos en lazo cerrado, ajusta un modelo que proporcione el mejor desempeño a lazo cerrado del sistema. No obstante, para emplear esta estrategia se necesita realizar identificación en línea, lo cual, evidentemente, es demasiado costoso, computacionalmente hablando, y dificulta su aplicación en sistemas con un elevado número de variables. Por su parte, (Xu et al., 2016) presenta una ANN recurrente en tiempo discreto para resolver un problema de programación convexo en el sistema de cuatro tanques. En tanto, (Nguyen, 2020) presenta un enfoque ANN que implica una red Elman recurrente para capturar la dinámica inversa del sistema de cuatro tanques. Sin embargo, los controladores basados en dinámica inversa podrían presentar problemas para controlar sistemas de fase no mínima.

Por lo antes visto, en este artículo se diseña un MPC no lineal, cuyo modelo de predicción se basa en una red neuronal tipo NARX con arquitectura *Feed-forward*. Lo novedoso de es-

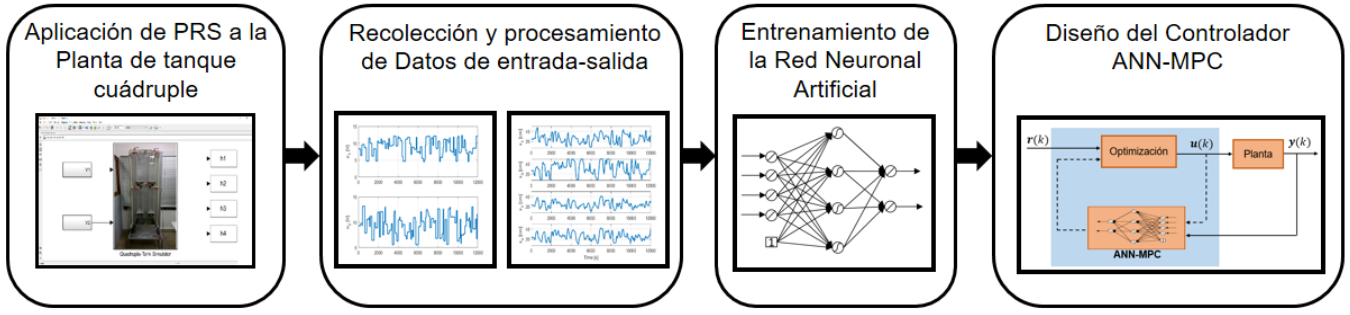


Figura 1: Diagrama esquemático del controlador ANN-MPC. Aplicación de las señales de excitación a la planta de tanque cuádruple. Recolección y procesamiento de los datos de entrenamiento. Modelamiento y entrenamiento de la red neuronal artificial. Diseño del controlador ANN-MPC.

te enfoque es que el inconveniente del control de sistemas no lineales de fase no mínima se supera mejorando la precisión de predicción del modelo a lo largo del horizonte de predicción, lo cual se consigue empleando una red neuronal modular junto con el algoritmo de entrenamiento *Levenberg-Marquardt* (LM) para obtener el modelo del proceso. Este enfoque evita emplear técnicas de regularización para superar el sobreajuste de parámetros y agiliza la etapa de modelado, ya que el entrenamiento de una red *Feed-forward* es menos complejo comparado con las redes neuronales recurrentes. Luego, el modelo se integra con el algoritmo de optimización del MPC para calcular la variable de control óptima. El modelo ANN se entrena, valida y prueba con datos reales de entrada-salida de la planta de tanque cuádruple, mientras que las pruebas del controlador se realizan usando un simulador basado en un modelo de primeros principios de la planta.

El resto de este artículo está organizado de la siguiente forma: la Sección 2 introduce brevemente la teoría de Control Predictivo Basado en Modelo (MPC) y la identificación con Redes Neuronales Artificiales (ANN). Luego, se presenta la metodología propuesta en este trabajo, la cual engloba la integración de estos dos conceptos. Por su parte, en la Sección 3, se presenta el caso de estudio, el cual es un proceso hidráulico no lineal de tanque cuádruple, que posee una alta interacción entre sus variables y se encuentra en configuración de fase no mínima. Primero, usando la planta de tanque cuádruple, se generan los datos de entrenamiento. Posteriormente se realiza el diseño y entrenamiento del modelo ANN, y en la Sección 4 se presenta un análisis de los resultados obtenidos de la aplicación del enfoque propuesto sobre el sistema de tanque cuádruple. Finalmente, la Sección 5 presenta algunas conclusiones obtenidas de la experimentación realizada.

## 2. Metodología

Esta sección proporciona una breve introducción a los dos conceptos más importantes discutidos en este trabajo, como el control MPC y las ANN para identificación. Asimismo, se muestra cómo se pueden integrar estas dos técnicas para diseñar un sistema de control robusto y preciso, cuyo diagrama esquemático se puede ver en la Figura 1.

### 2.1. Control predictivo basado en modelo (MPC)

El MPC es una estrategia de control avanzado e iterativo que emplea un modelo dinámico del proceso para predecir su

comportamiento futuro y calcular una secuencia de control óptima que, cuando se aplica, permite lograr ese comportamiento. El MPC es capaz de considerar los requisitos ambientales, económicos, y de seguridad requeridos para la operación de un proceso dentro de un índice de desempeño, el cual se minimiza resolviendo un problema de optimización. Esto le permite mejorar el desempeño de los sistemas de control MIMO (Rawlings et al., 2017).

Dado un sistema discreto en espacio de estados, lineal e invariante en el tiempo (LTI), definido por la siguiente expresión

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (1a)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (1b)$$

donde  $k \in \mathbb{Z}$  se refiere al instante de tiempo actual,  $\mathbf{x}(k) \in \mathbb{R}^n$ ,  $\mathbf{u}(k) \in \mathbb{R}^p$ , y  $\mathbf{y}(k) \in \mathbb{R}^q$  son el vector de estados, vector de entradas, y vector de salidas, respectivamente, con  $n$ ,  $p$  y  $q$ , como el número de estados, el número de entradas, y el número de salidas. Por otro lado,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times p}$  y  $\mathbf{C} \in \mathbb{R}^{q \times n}$  son las matrices de estados, de entradas y de salidas, respectivamente, y describen la dinámica de la planta.

En cada periodo de muestreo, el MPC minimiza una función de costo,  $J(k)$ , conformada por dos términos, uno referido al error entre la salida,  $\mathbf{y}$ , y la referencia  $\mathbf{r}$ , y el otro referido al incremento de control,  $\Delta \mathbf{u}$ :

$$J(k) = \sum_{j=1}^{N_p} \|\mathbf{y}(k+j|k) - \mathbf{r}(k+j|k)\|_Q^2 + \sum_{j=0}^{N_c-1} \|\Delta \mathbf{u}(k+j|k)\|_R^2 \quad (2)$$

donde  $\|\mathbf{z}\|_Q^2$  denota la norma euclidiana del vector  $\mathbf{z}$  y es equivalente a  $\sqrt{\mathbf{z}^T \mathbf{Q} \mathbf{z}}$ , con  $\mathbf{z}^T$  como la traspuesta del vector  $\mathbf{z}$ . Además,  $N_p$  se conoce como el horizonte de predicción, y es el número de pasos de tiempo hacia el futuro en el cual queremos minimizar  $J(k)$ . Asimismo,  $N_c$  es el horizonte de control e indica el número de movimientos de la variable de control que tenemos que aplicar para llegar a la referencia dada en el tramo indicado por  $N_c$ . Por otro lado, se denota por  $\mathbf{y}(k+j|k)$ , con  $j = 1, \dots, N_p$ , a la predicción de la salida,  $\mathbf{y}$ ,  $j$  pasos hacia el futuro, estimada a partir del instante  $k$ . Por otro lado, definimos el incremento de control como  $\Delta \mathbf{u}(k+j|k) = \mathbf{u}(k+j|k) - \mathbf{u}(k+j-1|k)$ . Las matrices  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  y  $\mathbf{R} \in \mathbb{R}^{p \times p}$  son matrices de ponderación que sirven para dar mayor relevancia a la minimización de cualquiera de los términos que conforman la función de costo, con la condición que  $\mathbf{Q} \geq 0$  sea positiva semi-definida, y  $\mathbf{R} > 0$  sea positiva definida.

La función de costo en (2) está sujeta a las siguientes restricciones

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0 \\ \mathbf{x}(k+j+1|k) &= \mathbf{A}\mathbf{x}(k+j|k) + \mathbf{B}\mathbf{u}(k+j|k), & j &= 0, \dots, N_P - 1 \\ \mathbf{y}(k+j|k) &= \mathbf{C}\mathbf{x}(k+j|k), & j &= 1, \dots, N_P \\ \mathbf{x}_{min} &\leq \mathbf{x}(k+j|k) \leq \mathbf{x}_{max}, & j &= 1, \dots, N_P \\ \mathbf{y}_{min} &\leq \mathbf{y}(k+j|k) \leq \mathbf{y}_{max}, & j &= 1, \dots, N_P \\ \Delta \mathbf{u}_{min} &\leq \Delta \mathbf{u}(k+j|k) \leq \Delta \mathbf{u}_{max}, & j &= 0, \dots, N_C - 1 \\ \mathbf{u}_{min} &\leq \mathbf{u}(k+j|k) \leq \mathbf{u}_{max}, & j &= 0, \dots, N_C - 1 \end{aligned}$$

Las ecuaciones a las que se sujeta el problema de optimización son conjuntos poliédricos de restricciones de estado, de entrada, y de salida. La implementación de las restricciones asegura que la variable de control se comporte adecuadamente sin exceder los límites físicos de los actuadores o sin realizar cambios bruscos que causen un deterioro en su vida útil. Estas restricciones también mantienen las salidas del sistema dentro de los límites de operación y de calidad impuestos por el proceso.

Resolviendo el problema de optimización, obtenemos la secuencia de control óptima ( $\mathbf{U} = \mathbf{u}(0), \dots, \mathbf{u}(N_C - 1)$ ) a lo largo del horizonte de control tal que se minimice  $J(k)$ . El primer término de la secuencia de control ( $\mathbf{u}(0)$ ) se aplica a la planta y el problema de optimización completo se vuelve a resolver en cada periodo de muestreo.

## 2.2. Identificación con redes neuronales artificiales

Una red neuronal artificial es un aproximador universal de funciones dinámicas (Hornik et al., 1989), es decir, es un modelo empírico no lineal que representa datos de entrada-salida. Una ANN es capaz de imitar el modelo de un proceso aprendiendo patrones de un conjunto de datos suficientemente grande, requiriendo así poco conocimiento a priori de la estructura (Bahar et al., 2004). Su capacidad para predecir con precisión respuestas futuras se ha demostrado en una diversidad de trabajos y el campo de sus aplicaciones está aumentando rápidamente (Brigham et al., 2018).

El funcionamiento de un modelo ANN se basa en el comportamiento del cerebro humano y está conformado por un conjunto de neuronas artificiales las cuales están ubicadas en una capa de entrada, una o más capas ocultas o intermedias, y una capa de salida. Las neuronas de una capa se hallan interconectadas a las neuronas de otra capa a través de pesos sinápticos para producir una salida a partir de ciertas entradas. Estas conexiones le permiten a la ANN lidiar con problemas de no linealidad.

En este trabajo se adoptará una red neuronal dinámica debido a que las variables del sistema están representadas por datos en series de tiempo. Las ANN dinámicas se diferencian de las ANN estáticas debido a que cuentan con elementos de retroalimentación con retardos y, gracias a estos, son más precisas. Entre las ANN dinámicas destacan los modelos no lineal autoregresivos exógenos (NARX), que son considerados buenos predictores del comportamiento de sistemas dinámicos no lineales gracias a su rápida convergencia y su mejor capacidad de generalización, en comparación con otras redes (Manonmani et al., 2018). Las NARX's se han empleado en varias aplicaciones y han mostraron un rendimiento confiable (Saadon et al.,

2020). La arquitectura de un modelo NARX se muestra en (3).

$$\hat{x}(t+1) = f_{NN}(x(t), \dots, x(t-n_x), u(t), \dots, u(t-n_u)) \quad (3)$$

donde  $f_{NN}$  es una función no lineal que representa al modelo,  $u(t)$  y  $x(t)$  son la entrada y salidas del modelo en el instante  $t$ , respectivamente.  $\hat{x}(t+1)$  es salida estimada por el modelo NARX en el instante  $t$  para el tiempo  $t+1$ . Finalmente,  $n_x$  y  $n_u$  son el número de retardos de la salida y de la entrada, respectivamente.

El valor futuro de la salida,  $\hat{x}(t+1)$ , se calcula a partir de los valores de entrada actuales y pasados de  $u(t-i)$ ,  $i = 0, 1, \dots, n_u$ , junto con los valores de salida actuales y previos de  $x(t-i)$ ,  $i = 0, 1, \dots, n_x$ .

En cada capa, la señal de salida se expresa por

$$y_i = f_a \left( \sum_{j=1}^n x_j \cdot w_{ij} + bias \right) \quad (4)$$

donde  $f_a$  es la función de activación,  $i$  es la ubicación en la capa actual,  $j$  es la ubicación en la capa previa, y  $n$  es el número de neuronas en dicha capa. Asimismo,  $y_i$  es la señal de salida,  $x_j$  = señal de entrada,  $w_{ij}$  es el peso sináptico entre  $i$  y  $j$ . Muchas veces para obtener mejores resultados es necesario usar un valor de sesgo, también conocido como bias.

La función de activación es la acción que ejecuta una neurona para obtener una salida a partir de las entradas que le llegan. Esta función se utiliza para limitar el rango de valores de salida de la neurona. Generalmente, sus valores de salida suelen estar entre (0,1) ó (-1,1), dependiendo de la función con la que se esté trabajando, o se podrían adoptar otros valores según la aplicación. Existen muchas funciones de activación y se debe elegir una de acuerdo al problema que se requiera resolver. La función de activación debe ser continua; diferenciable en cada punto; y monótona no decreciente. Hay muchas funciones candidatas, incluyendo la función lineal, escalonada, sigmoidea, tanh, etc.

El proceso de aprendizaje de una ANN se realiza modificando sus pesos sinápticos, lo que se conoce como entrenamiento, y suele realizarse mediante el algoritmo *Back-Propagation* (BP) (Amari, 1993). El procedimiento consiste en aprender un conjunto de patrones de entrada-salida, que sirven de estímulo para activar las neuronas de la primera capa de la ANN, que generan una salida que se propaga a través de las neuronas de las capas siguientes hasta generar una salida global. La salida global predicha por la ANN se compara con la salida deseada y se calcula el error. El objetivo es minimizar una función de pérdida formada por estos errores. Generalmente, se puede elegir como función de pérdida el error medio cuadrático (MSE), o la raíz del error medio cuadrático (RMSE), entre otros (Popoola et al., 2019). Esto depende de factores como: la aplicación, la capacidad de cálculo de los equipos, etc. En particular, en este trabajo se usará el MSE. Una vez calculados los errores de salida, se propagarán hacia atrás para modificar los pesos sinápticos de la red de forma que se minimice la función de pérdida.

## 2.3. Control predictivo basado en un modelo de redes neuronales artificiales (ANN-MPC)

El enfoque ANN-MPC empleado en este artículo consta básicamente de dos etapas. La Figura 2 muestra el procedimiento operativo del sistema ANN-MPC. En la primera parte,

se identifica el proceso empleando un modelo basado en redes neuronales con estructura NARX. Este paso es de mucha importancia y es el que mayor tiempo toma para su elaboración.

Para identificar el modelo ANN, es importante evitar el efecto de sobreajuste, también conocido como *overfitting*, que hace que la red neuronal entrenada pierda su capacidad para generalizar, es decir, la ANN aprende los patrones de entrenamiento con gran precisión, pero no obtiene buenos resultados de predicción ante nuevos datos. Una posible causa del *overfitting* es tener una gran cantidad de neuronas en la capa oculta. Esto sería sencillo de resolver disminuyendo el número de neuronas. Sin embargo, al reducir el número de neuronas en la capa oculta, corremos el riesgo de perder precisión en la predicción.

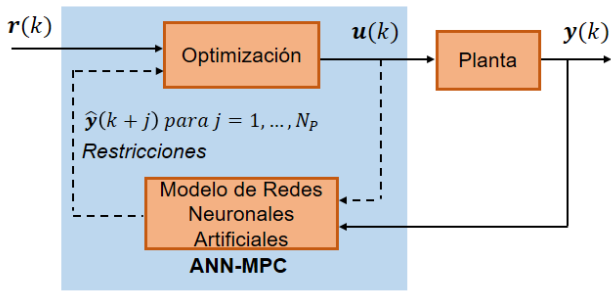


Figura 2: Diagrama de bloques del enfoque ANN-MPC empleado. Las predicciones obtenidas con el modelo ANN son usadas por el algoritmo de optimización del MPC para calcular la variable de control óptima.

Dado el inconveniente antes mencionado, una posible solución sería la regularización, que consiste en modificar la función de pérdida de la ANN agregando un término ponderado que se basa en la media de la suma cuadrada de los pesos y el bias de la red. La ponderación de este término se logra modificando un parámetro de tasa de desempeño. De esta manera, los pesos y el bias se vuelven más pequeños, lo cual suaviza la predicción de la red y evita el sobreajuste. Sin embargo, es posible caer en el mismo problema anterior, puesto que si elegimos un valor demasiado grande para el parámetro de tasa de desempeño, podemos generar *overfitting*. Por lo contrario, si dicho parámetro es demasiado pequeño, la red neuronal tendría problemas para ajustar su salida a los datos de entrenamiento. Una solución para terminar con este problema es el empleo de técnicas estadísticas, como la regularización Bayesiana, que suele usarse en el entrenamiento de ANN junto con el algoritmo de aprendizaje LM, como en (Shin et al., 2020).

En este trabajo, sin embargo, para evitar el sobreajuste, se propone emplear una estructura de red neuronal modular junto con el algoritmo LM. Es decir, se empleará un número de ANN igual al número de subsistemas que conforman el sistema a controlar que, en este caso, es un sistema hidráulico de cuatro tanques. Con esto se pretende disminuir el número de neuronas en la capa oculta de cada red neuronal, y así evitar el sobreajuste, pero además, esto permitirá reducir el tiempo de cómputo que emplea cada red para entregar la estimación de sus variables de estado. De esta manera, cada ANN se encargará de estimar el estado que le corresponde a cada una de ellas, sin usar regularización.

Una vez obtenido el modelo ANN, este será integrado dentro del algoritmo de optimización de un controlador MPC no

lineal. El esquema del controlador ANN-MPC consta de una función de costo modificada, basada en el trabajo de (Alvarado et al., 2011), donde se considera la adición de los estados y entradas estacionarios ( $\mathbf{x}_s, \mathbf{u}_s$ ) como variables de decisión. Además, se considera una restricción terminal extendida. La función de costo en este enfoque se puede escribir como en (5), donde además se incluyen las restricciones de estados, de entradas y de salidas.

$$J(k) = \sum_{j=1}^{N_p} \|\hat{\mathbf{y}}(k+j) - \mathbf{r}(k+j)\|_Q^2 + \sum_{j=1}^{N_p-1} \|\hat{\mathbf{x}}(k+j) - \mathbf{x}_s(k+j)\|_S^2 + \|\hat{\mathbf{x}}(N_p) - \mathbf{x}_s(N_p)\|_G^2 + \sum_{j=0}^{N_C-1} \|\mathbf{u}(k+j) - \mathbf{u}_s(k+j)\|_R^2 \quad (5)$$

sujeta a

$$\begin{aligned} \hat{\mathbf{x}}(0) &= \mathbf{x}_0 \\ \hat{\mathbf{x}}(k+j+1|k) &= f_{NN}(\hat{\mathbf{x}}(k+j|k), \mathbf{u}(k+j|k)), \quad j = 0, \dots, N_p - 1 \\ \hat{\mathbf{y}}(k+j|k) &= g(\hat{\mathbf{x}}(k+j|k), \mathbf{u}(k+j|k)), \quad j = 1, \dots, N_p \\ \mathbf{x}_{min} &\leq \hat{\mathbf{x}}(k+j|k) \leq \mathbf{x}_{max}, \quad j = 1, \dots, N_p \\ \mathbf{y}_{min} &\leq \hat{\mathbf{y}}(k+j|k) \leq \hat{\mathbf{y}}_{max}, \quad j = 1, \dots, N_p \\ \Delta \mathbf{u}_{min} &\leq \Delta \mathbf{u}(k+j|k) \leq \Delta \mathbf{u}_{max}, \quad j = 0, \dots, N_C - 1 \\ \mathbf{u}_{min} &\leq \mathbf{u}(k+j|k) \leq \mathbf{u}_{max}, \quad j = 0, \dots, N_C - 1 \end{aligned}$$

donde la matriz de ponderación  $S$  nos garantiza que los estados del proceso se aproximen a los estados estacionarios del punto de operación generado por la referencia  $r$ . Mientras que la matriz de ponderación  $G$  nos asegura que la restricción terminal sea alcanzada. Si los estados y entradas estacionarios no son alcanzables, el controlador calculará la solución alcanzable más cercana a ese punto.

El proceso estudiado en este artículo consta de cuatro variables de estado y dos variables de salida. Para obtener las salidas, primero debemos obtener los estados, por ello, en el instante de muestreo actual ( $k$ ), el modelo ANN, representado por  $f_{NN}$ , estima los estados del sistema a lo largo del horizonte de predicción,  $\hat{\mathbf{x}}(k+j)$ ,  $j = 1, \dots, N_p$ . Los valores futuros de los cuatro estados,  $\hat{\mathbf{x}}(k+j+1)$ , se obtendrán empleando los estados estimados en el paso anterior,  $\hat{\mathbf{x}}(k+j)$ , es decir, se empleará una estimación recurrente hasta  $\hat{\mathbf{x}}(k+N_p)$ . Luego, se estiman las salidas utilizando la función  $g$ , que en este caso será una función identidad. Después, se calcula la secuencia de control óptima, ( $\mathbf{U} = \mathbf{u}(0), \dots, \mathbf{u}(N_C - 1)$ ), tratando de que las salidas estimadas  $N_p$  pasos hacia adelante alcancen a las referencias  $\mathbf{r}$ . Solo el primer valor de  $\mathbf{U}$  es aplicado a la planta. Finalmente, se avanza al siguiente instante y el procedimiento de optimización se repite para encontrar nuevamente  $\mathbf{U}$ , para lo cual se empleará la función *fmincon* del toolbox de optimización de *Matlab*®, donde se usará el método de optimización de punto interior.

### 3. Caso de estudio: Sistema de tanque cuádruple

#### 3.1. Descripción del proceso de tanque cuádruple

El diagrama del proceso de tanque cuádruple se muestra en la Figura 3. Este sistema está compuesto por cuatro tanques que son alimentados por dos electrobombas que impulsan agua desde un reservorio ubicado en la parte inferior del módulo. Los

dos tanques superiores, 1 y 3, se descargan directamente sobre los tanques inferiores correspondientes, 2 y 4. El objetivo de control de este sistema es regular el nivel de los tanques inferiores ( $h_2$  y  $h_4$ ) por medio de la manipulación del voltaje de alimentación de las electrobombas ( $u_1$  y  $u_2$ ). Por otro lado, para modificar la configuración de la planta existen 6 válvulas de bola que proveen una proporción del caudal total de acuerdo a su porcentaje de apertura  $mv_j$ , con  $j \in a, \dots, f$ . Cuando las válvulas  $mv_b$  y  $mv_e$  permanecen totalmente cerradas, y las válvulas superiores,  $mv_c$  y  $mv_f$ , se abren en mayor proporción que las válvulas inferiores,  $mv_a$  y  $mv_d$ , tal que proporcionen mayor flujo de agua, se obtiene un sistema en configuración de fase no mínima.

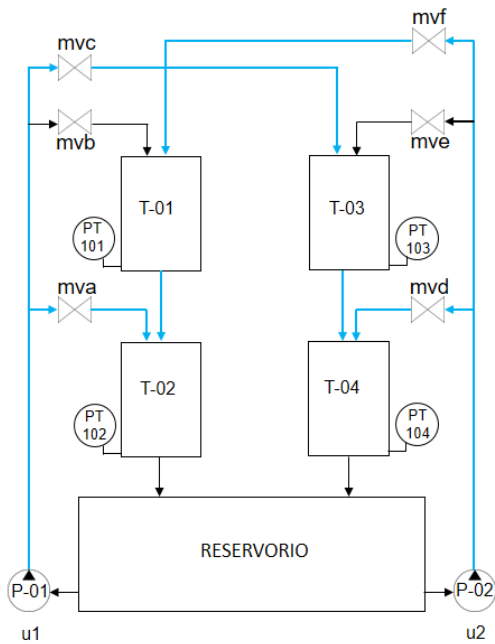


Figura 3: Diagrama del proceso de tanque cuádruple. Las variables de control son los voltajes aplicados a las electrobombas,  $v_1$  y  $v_2$ , y las variables controladas son los niveles de los tanques inferiores,  $h_2$  y  $h_4$ .

Para obtener los datos de entrenamiento de la red neuronal se empleó la planta piloto de tanque cuádruple del Laboratorio de Sistemas Automáticos de Control de la Universidad de Piura (Calle and Oliden, 2021b), la cual se puede apreciar en la Figura 4, y cuyos parámetros se muestran en la Tabla 1. Esta planta posee 4 tanques acrílicos con una altura ( $H_{max}$ ) igual a 50 cm y una superficie transversal de 100 cm<sup>2</sup>. Los actuadores son dos electrobombas marca Johnson, modelo CM30P7-1, las cuales trabajan con un voltaje máximo de 24 VDC y proporcionan un caudal máximo de 22.5 L/min. Asimismo, para obtener el nivel de los tanques se cuenta con 4 transmisores de presión (PT-101, PT-102, PT-103 y PT-104) marca WIKA, modelo S-10, con un rango de operación entre 0-100 mbar, los mismos que están conectados a un PLC marca SIEMENS S7-1200 CPU 1214C. Con la presión,  $P$  (mbar), indicada por los transmisores se obtiene la altura,  $h$  (cm), del agua en los tanques empleando la siguiente ecuación:

$$h = 0,981P \quad (6)$$

Tabla 1: Parámetros de la planta (Calle and Oliden, 2021b)

Parámetro	Valor	Unidad	Descripción
$a_1$	0.4	cm <sup>2</sup>	Área descarga tanque 1
$a_2$	0.6	cm <sup>2</sup>	Área descarga tanque 2
$a_3$	0.4	cm <sup>2</sup>	Área descarga tanque 3
$a_4$	0.7	cm <sup>2</sup>	Área descarga tanque 4
$mv_a$	30	%	Apertura de la válvula $a$
$mv_b$	0	%	Apertura de la válvula $b$
$mv_c$	70	%	Apertura de la válvula $c$
$mv_d$	40	%	Apertura de la válvula $d$
$mv_e$	0	%	Apertura de la válvula $e$
$mv_f$	60	%	Apertura de la válvula $f$
$H_{max}$	50	cm	Altura máx. del tanque
$H_{min}$	7	cm	Altura mín. del tanque
$S$	100	cm <sup>2</sup>	Área transversal tanque
$k$	15.625	cm <sup>3</sup> /(s.V)	Factor de conversión
$g$	981	cm/s <sup>2</sup>	Aceleración gravedad



Figura 4: Planta de tanque cuádruple instalada en el laboratorio de Sistemas Automáticos de Control, de la Facultad de Ingeniería de la Universidad de Piura, Perú.

Para realizar las pruebas del enfoque propuesto se empleará un simulador de la planta de tanque cuádruple implementado en *Matlab*<sup>®</sup>. Este simulador usa un modelo basado en primeros principios, el cual a su vez se basa en el balance de masa, la ecuación de Bernoulli, y la ley de conservación de la velocidad. La dinámica de este sistema es descrita por las siguientes ecuaciones no lineales:

$$\frac{dh_1}{dt} = \frac{k(mv_b)}{S}u_1 + \frac{k(mv_f)}{S}u_2 - \frac{a_1}{S}\sqrt{2gh_1} \quad (7a)$$

$$\frac{dh_2}{dt} = \frac{k(mv_a)}{S}u_1 + \frac{a_1}{S}\sqrt{2gh_1} - \frac{a_2}{S}\sqrt{2gh_2} \quad (7b)$$

$$\frac{dh_3}{dt} = \frac{k(mv_c)}{S}u_1 + \frac{k(mv_e)}{S}u_2 - \frac{a_3}{S}\sqrt{2gh_3} \quad (7c)$$

$$\frac{dh_4}{dt} = \frac{k(mv_d)}{S}u_2 + \frac{a_3}{S}\sqrt{2gh_3} - \frac{a_4}{S}\sqrt{2gh_4} \quad (7d)$$

### 3.2. Diseño y entrenamiento del modelo ANN

Para diseñar el controlador ANN-MPC, primeramente, se consiguieron datos en series de tiempo para entrenar el modelo ANN. Para esto, se empleó la planta de tanque cuádruple, descrita antes, con la cual se obtuvieron los pares de datos de entrada-salida del funcionamiento del proceso.

En la planta de tanque cuádruple, las entradas al sistema son los voltajes aplicados a las electrobombas ( $u_1$  y  $u_2$ ), y las variables de estado son los niveles de los cuatro tanques ( $h_1, h_2, h_3$  y  $h_4$ ), los cuales dependen de las variables de entrada. Para las entradas se diseñaron dos Señales Pseudo Aleatorias (PRS) que fueron aplicadas con el objetivo de tener una excitación que fuera rica en frecuencias y no estacionaria. Se decidió primero estabilizar la planta en un punto de operación, que aparece en la Tabla 2, y a partir del cual se realizaron variaciones aleatorias de valores entre  $\pm 4V$  para cada entrada, como se aprecia en la Figura 5. El comportamiento de los niveles de los tanques ante estas variaciones se puede apreciar en la Figura 6. Los pares de datos de entrada-salida se generaron durante una prueba de 12000 s y se guardaron con un periodo de muestreo de  $T_s = 5$  s, con lo cual se recolectaron  $M = 2400$  muestras.

Tabla 2: Punto de operación

Variable	Valor	Unidad
$u_{10}$	8.50	V
$u_{20}$	9.50	V
$h_{10}$	25.26	cm
$h_{20}$	23.52	cm
$h_{30}$	27.53	cm
$h_{40}$	24.14	cm

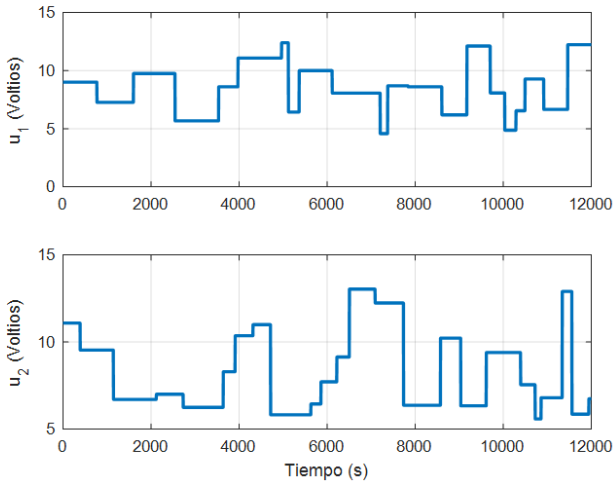


Figura 5: Señales PRS aplicadas a las electrobombas de la planta.

El proceso de modelado de la ANN se realizó empleando el software *Matlab*<sup>®</sup>. Para obtener mejores resultados de predicción del comportamiento de la planta de tanque cuádruple, en este artículo se propone el empleo de una estructura modular de red neuronal. Dicha estructura, a diferencia de una estructura compacta, diferenciará de forma individual la importancia de cada entrada para cada salida (estado estimado de la planta). La estructura modular está compuesta por cuatro ANN tipo NARX

que representan a los cuatro tanques del proceso (Figura 7). La ANN 1 recibe como entradas el valor actual ( $t$ ) y previo ( $t - 1$ ) del estado  $x_1$  y de la variable de control  $u_2$ , además considera un bias de entrada. La ANN 2 recibe como entradas el valor actual y previo de los estados  $x_2$  y  $x_1$ , y de la variable de control  $u_1$ , además del bias de entrada. Situaciones similares ocurren con las ANN 3 y 4, respectivamente. El número de neuronas en la capa intermedia de cada ANN se determinó variando dicho número y evaluando el error medio cuadrático (MSE) de la suma de los cuatro estados estimados en cada caso (ver Figura 8). Finalmente, se estableció un número igual 5 neuronas, ya que para un mayor número se obtuvieron MSE parecidos pero con mayor carga computacional. El MSE se define en (8) (Popoola et al., 2019)

$$MSE = \frac{1}{M} \sum_{i=1}^M (x_i - \hat{x}_i)^2 \quad (8)$$

donde  $x_i$  son los estados reales,  $\hat{x}_i$  son los estados estimados por la ANN, y  $M$  es el número de muestras. Asimismo, en la capa de salida se estableció 1 neurona para cada ANN que representa el valor futuro ( $t + 1$ ) de los estados estimados de la planta ( $\hat{x}_1, \hat{x}_2, \hat{x}_3$  y  $\hat{x}_4$ ), que se representan por

$$\hat{x}_1(t+1) = f_{NN1}(x_1(t), x_1(t-1), u_2(t), u_2(t-1)) \quad (9a)$$

$$\hat{x}_2(t+1) = f_{NN2}(x_1(t), x_1(t-1), x_2(t), x_2(t-1), u_1(t), u_1(t-1)) \quad (9b)$$

$$\hat{x}_3(t+1) = f_{NN3}(x_3(t), x_3(t-1), u_1(t), u_1(t-1)) \quad (9c)$$

$$\hat{x}_4(t+1) = f_{NN4}(x_3(t), x_3(t-1), x_4(t), x_4(t-1), u_2(t), u_2(t-1)) \quad (9d)$$

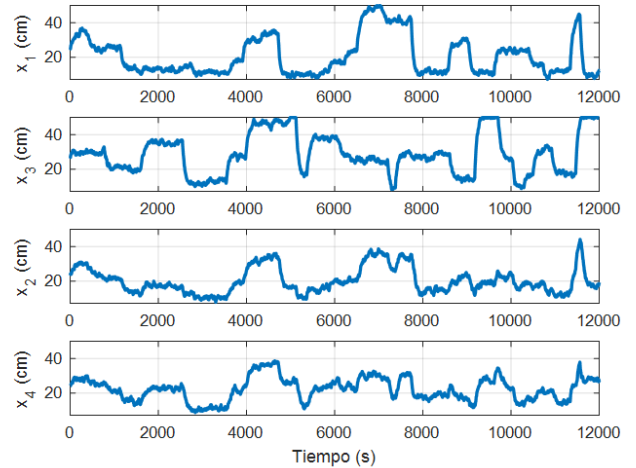


Figura 6: Respuesta de los niveles de los cuatro tanques de la planta ante las entradas PRS aplicadas.

Para el entrenamiento del modelo ANN, como se mencionó antes, se empleó el algoritmo LM, que a pesar de necesitar bastante memoria para calcular las matrices Jacobianas, es uno de los métodos más empleados y eficientes que existe para el entrenamiento de ANN. La ecuación de este algoritmo puede representarse como una combinación del algoritmo del Gradiente Descendente y el algoritmo de Newton, y se expresa como

$$\mathbf{W}(k+1) = \mathbf{W}(k) - (\mathbf{J}^T(k)\mathbf{J}(k) + \mu\mathbf{I})^{-1}\mathbf{J}(k)\mathbf{e}(k) \quad (10)$$

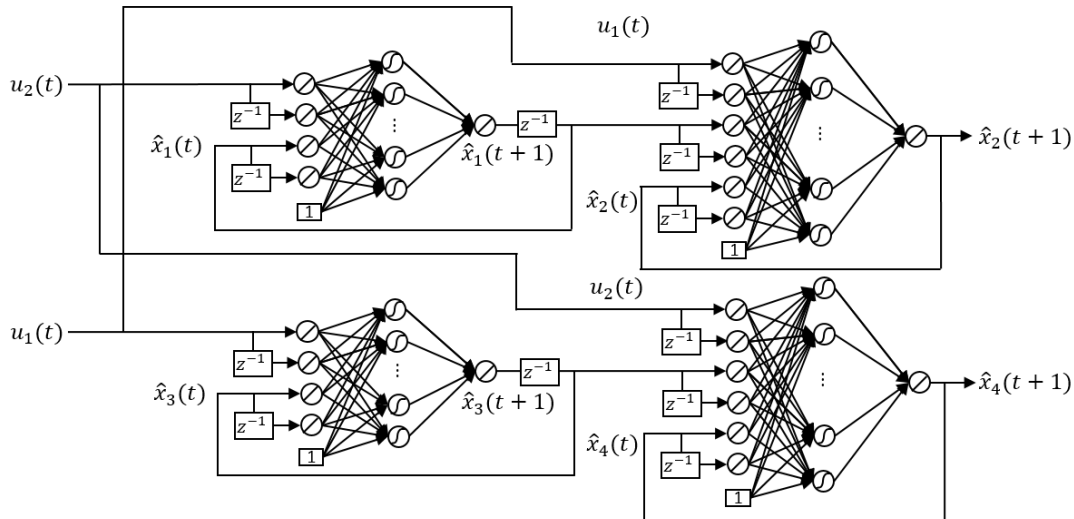


Figura 7: Estructura modular del modelo ANN compuesto por 4 redes neuronales representando cada estado de la planta. Los estados estimados son realimentados para estimar los próximos valores en la siguiente iteración.

donde  $\mu$  es un factor de combinación,  $\mathbf{J}$  es la matriz Jacobiana,  $\mathbf{I}$  es la matriz identidad y  $\mathbf{e}$  es el vector error.

De (10) podemos ver que, cuando  $\mu$  es cero, el algoritmo se comporta como el algoritmo de Newton. Así mismo, cuando  $\mu$  es grande, el algoritmo se comporta como el algoritmo del Gradiente Descendente con un tamaño de paso pequeño.

prueba. Se realizaron 1000 iteraciones para el entrenamiento de la red neuronal. El resumen de todos estos hiper-parámetros se muestra en la Tabla 3.

Tabla 3: Hiper-parámetros del modelo ANN

Variable	Valor
Estandarización	Sí
Normalización	Sí (-1,1)
Función de activación	tanh
Función de pérdida	MSE
Optimizador	Levenberg-Marquardt
Iteraciones	1000

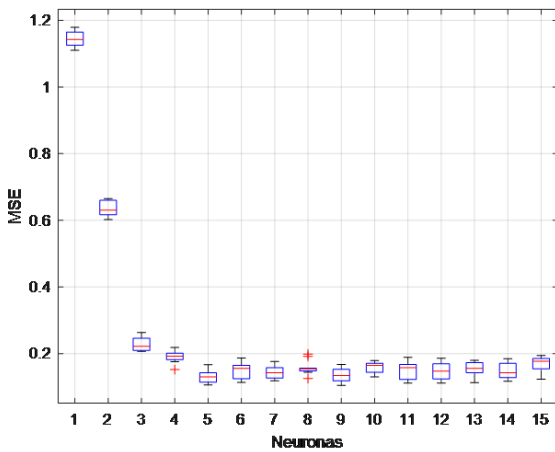


Figura 8: Valor del MSE para distintos números de neuronas en la capa intermedia.

Por otro lado, se eligió como función de activación la función *tanh* y, por las características de aquella función de activación, se realizó la normalización de los datos de entrenamiento, validación, y prueba, empleando la siguiente expresión

$$u_{nom} = \frac{2u - (u_{max} + u_{min})}{u_{max} - u_{min}} \quad (11)$$

con la cual, conseguimos valores en un rango entre (-1,1).

Asimismo, los datos generados se dividieron en 1600 muestras para entrenamiento, 800 para validación, y se tomaron 800 muestras de una nueva señal PRS diseñada para la etapa de

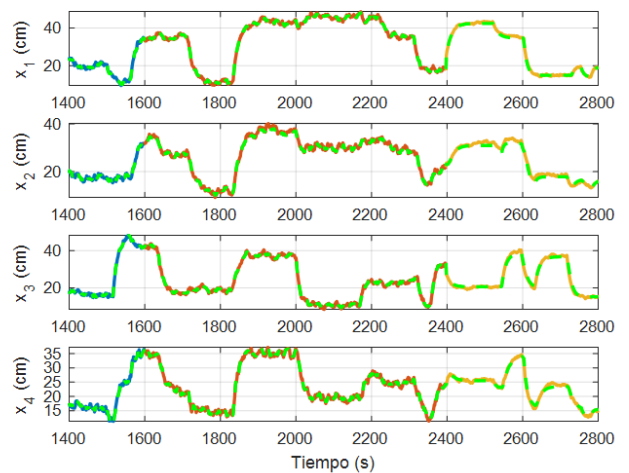


Figura 9: Predicción de estados empleando el modelo ANN (línea discontinua verde) en comparación con los datos de entrenamiento (línea sólida azul), validación (línea sólida roja), y prueba (línea sólida naranja).

Finalmente, el desempeño de predicción del modelo ANN entrenado se puede observar en la Figura 9, donde apreciamos



una excelente aproximación entre los estados estimados por la ANN y los estados reales de la planta. Esto se cumple en todas las etapas del procedimiento, es decir en el entrenamiento, en la validación y finalmente en la prueba. Con esto podemos concluir que el modelo ANN es adecuado.

Con el objetivo de evaluar el desempeño de predicción del modelo ANN entrenado, para cada estado se calculó el índice *Fitness* definido como

$$FIT = 100 \left( 1 - \frac{\sum_{i=1}^M (x_i - \hat{x}_i)^2}{\sum_{i=1}^M (x_i - \mu)^2} \right) \quad (12)$$

donde  $x_i$  son los estados reales,  $\hat{x}_i$  son los estados estimados, y  $\mu$  es la media de los estados reales. El resumen de los FIT's alcanzados lo podemos ver en la Tabla 4, donde apreciamos muy buenos resultados de predicción para los cuatro estados.

Tabla 4: Estadísticas de entrenamiento

Etapas	Muestras	FIT			
		$x_1$	$x_2$	$x_3$	$x_4$
Entrenamiento	1600	95.90	94.14	95.25	93.06
Validación	800	94.01	90.52	96.60	90.65
Prueba	800	93.04	86.09	92.85	91.19

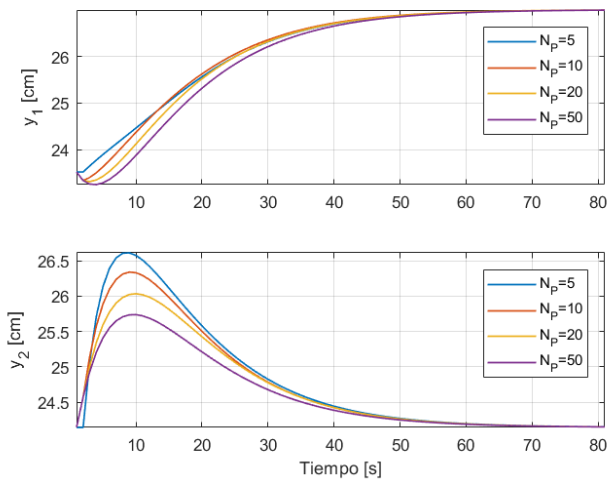


Figura 10: Influencia del horizonte de predicción sobre el desempeño del controlador ANN-MPC.

Una vez entrenado el modelo ANN, este es integrado con el algoritmo de optimización del controlador ANN-MPC

### 3.3. Diseño del controlador ANN-MPC

Para conseguir un buen desempeño del ANN-MPC, se debe sintonizar correctamente sus parámetros. Para ello, se probó el sistema ante cambios de referencia. La prueba consiste en variar  $r_1$  a partir del punto de operación (Tabla 2) y observar el efecto de trabajar con diferentes horizontes de predicción,  $N_p$ , mientras el horizonte de control,  $N_c$ , se mantiene igual a  $N_p$ . El resultado de esta prueba se aprecia en la Figura 10, donde para la salida  $y_1$ , se ha establecido  $r_1 = 27$  cm, y se aprecia cómo el sistema tiene un comportamiento de fase no mínima más marcado mientras más grande es  $N_p$ . Mientras que para  $y_2$ , si bien

no se ha impuesto ningún cambio de referencia en dicha variable, esta sufre la interacción de  $y_1$ , así como lo demuestra el sobreimpulso generado en  $y_2$  mientras transcurre la respuesta transitoria en  $y_1$ . A diferencia de lo ocurrido para  $y_1$ , en este segundo caso, el sobreimpulso en  $y_2$  es menor mientras más grande es  $N_p$ . Asimismo, también se probaron diferentes pesos de ponderación para la variable de control. Para medir la influencia de los parámetros de sintonización del ANN-MPC, se calculará la raíz del error medio cuadrático (RMSE), definido en (13) (Popoola et al., 2019).

$$RMSE = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (r_i - y_i)^2} \quad (13)$$

donde  $r_i$  es la referencia deseada para cada salida ( $y_1$  y  $y_2$ ), y  $N_{test}$  es el número de muestras de la simulación. Todos los resultados de estas pruebas se pueden ver en la Tabla 5.

Tabla 5: RMSE para los parámetros del ANN-MPC

$R = 1, N_C = N_P$	$N_P = 5$	$N_P = 10$	$N_P = 20$	$N_P = 50$
$y_1$	1.3335	1.3765	1.4605	1.5577
$y_2$	1.0342	0.9471	0.7153	0.7153
$N_P = N_C = 10$	R=0.1	R=1	R=10	R=100
$y_1$	1.3970	1.3755	1.3777	1.3980
$y_2$	0.9723	0.9482	0.9457	0.9220

Con esto se concluye que, para el diseño del ANN-MPC,  $T_s$  será igual a 5 s, y tanto  $N_p$  como  $N_c$  serán igual a 10. Asimismo, se ha establecido como matrices de ponderación  $Q = 1$  y  $R = 10$ . Para comparar el desempeño del enfoque propuesto se diseñó un MPC lineal, basado en (2), y un MPC no lineal (NMPC), basado en la estructura de control centralizado (Alvarado et al., 2011). El modelo de predicción del MPC no lineal es el modelo visto en (7), y el del MPC lineal se obtuvo al linealizar (7) usando series de Taylor (Colosi and Abrudean, 2013). En el caso del NMPC, este tendrá los mismos parámetros de sintonización que el ANN-MPC, y en el caso del MPC lineal, se estableció  $N_p = 100$  debido a que es el que mejores resultados arrojó en las pruebas de sintonización. Los parámetros de estos tres controladores se ven en la Tabla 6.

Por otro lado, como restricciones se han establecido valores entre (7, 50) cm para el nivel de los tanques, y valores entre (0,17) voltios para el caso de la variable de control  $u_1$ , y valores entre (0,19) voltios para  $u_2$ .

Tabla 6: Parámetros de los controladores

Parámetro	MPC	NMPC	ANN-MPC
Periodo de muestreo ( $T_s$ )	5 s	5 s	5 s
Horizonte de predicción ( $N_p$ )	100	10	10
Horizonte de control ( $N_c$ )	100	10	10
Ponderación de $y$ ( $Q$ )	1	1	1
Ponderación de $u$ ( $R$ )	10	10	10

## 4. Resultados numéricos

Para probar el desempeño de los controladores, se realizó un experimento de seguimiento de referencia variando el *set-point*

para el nivel de los tanques 2 y 4 ( $y_1$  y  $y_2$ ). Inicialmente, las referencias se establecen en  $r_1 = r_2 = 23.68 \text{ cm}$  con el objetivo de llevar la planta a un punto de operación donde se garanticen las mismas condiciones iniciales para cada controlador. Estas referencias se mantienen así durante los primeros 1000 s. Después de ese tiempo, se aplica un escalón negativo donde las referencias se cambian a  $r_1 = r_2 = 10.71 \text{ cm}$ , y se mantienen así por otros 1000 s. Luego, las referencias se cambian a  $r_1 = 18.12 \text{ cm}$  y  $r_2 = 27.39 \text{ cm}$ , y se mantienen durante los siguientes 1000 s. Finalmente, se cambia a  $r_1 = 32.95 \text{ cm}$  y  $r_2$  se conserva igual durante los 1000 s finales.

Para evaluar el desempeño del enfoque propuesto, se plantea usar un índice de performance definido en (14), y a la vez, para cada salida del sistema, se calculará los RMSE.

$$J_{Indice} = \sum_{k=1}^{N_{test}} (r_i - y_i)^2 + 0,01(u_i)^2 \quad (14)$$

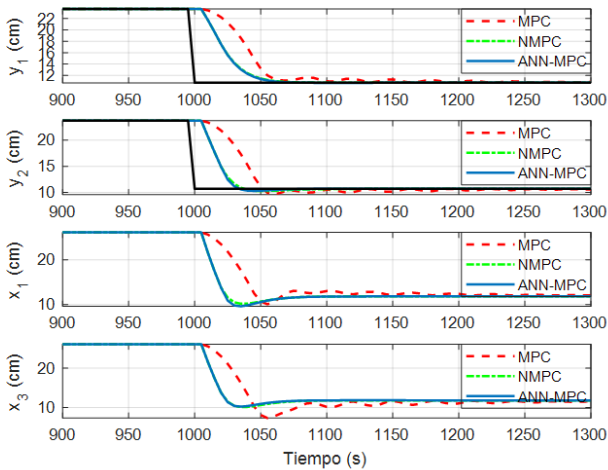


Figura 11: Desempeño del control ANN-MPC (línea sólida azul) en comparación con el MPC lineal (línea punteada roja) y el MPC no lineal (línea punteada verde) ante cambios de referencia (línea sólida negra) en el instante  $t = 1000 \text{ s}$ .

En la Figura 11 vemos el desempeño del enfoque propuesto en comparación con los otros dos controladores, es decir, el MPC lineal y el MPC no lineal. Se inicia desde un punto de operación igual para los tres controladores. En el instante  $t = 1000 \text{ s}$ , se aplicó una variación negativa para ambas referencias, y podemos notar que, tanto el ANN-MPC como el NMPC, alcanzan las referencias de forma veloz y suavizada, mientras que el MPC lineal lo hace un poco más lento y con pequeñas oscilaciones. Asimismo, en la Figura 12 vemos que las variables de control trabajan cerca de sus límites inferiores y, nuevamente, la respuesta más rápida la obtuvieron el ANN-MPC y el NMPC.

Por otro lado, en la Figura 13 vemos que, para el instante  $t = 2000 \text{ s}$ , se aplica una variación positiva de ambas referencias. Logramos notar que los controladores ANN-MPC y NMPC consiguen respuestas muy parecidas, ambas con un sobre-impulso en  $y_1$  similar. Esta vez el MPC lineal tiene una respuesta parecida con un sobre-impulso ligeramente más pronunciado. También es posible notar que el estado  $x_3$  está cerca de su límite superior. Esto hace que sea más complicado encontrar una variable de control óptima alcanzable, y que se fuerce al sistema a violar las restricciones interpuestas.

trar una variable de control óptima alcanzable, y que se fuerce al sistema a violar las restricciones interpuestas.

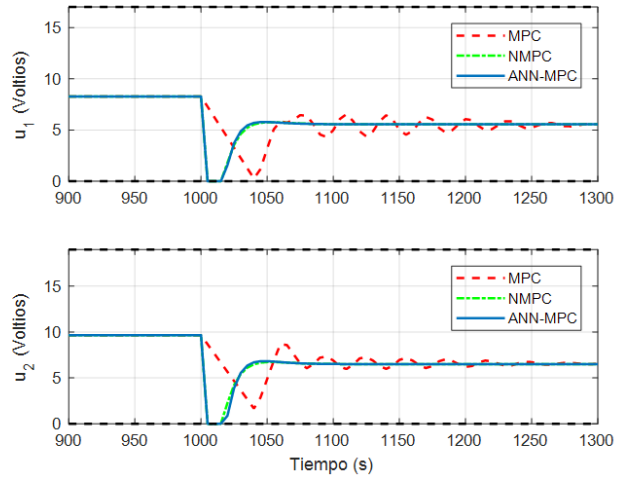


Figura 12: Variable de control del ANN-MPC (línea sólida azul) en comparación con el MPC lineal (línea punteada roja) y el MPC no lineal (línea punteada verde) ante cambios de referencia en el instante  $t = 1000 \text{ s}$ .

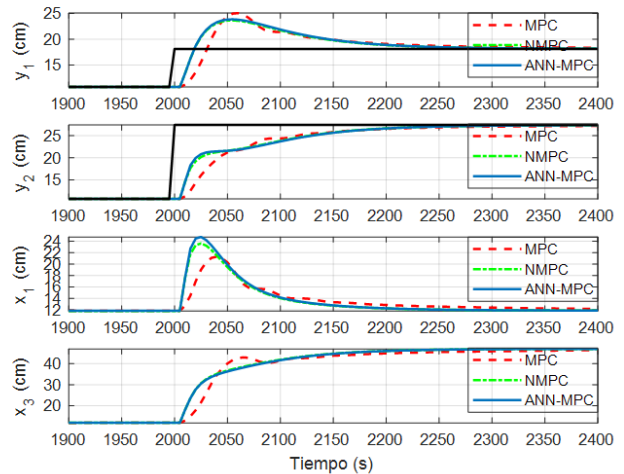


Figura 13: Desempeño del control ANN-MPC (línea sólida azul) en comparación con el MPC lineal (línea punteada roja) y el MPC no lineal (línea punteada verde) ante cambios de referencia (línea sólida negra) en el instante  $t = 2000 \text{ s}$ .

En la Figura 14 vemos que la variable de control  $u_2$ , en este caso, es más pronunciada para el ANN-MPC y el NMPC, mientras que para el MPC es menor. Sin embargo, para el MPC la reacción de  $u_1$  tiene una dinámica más lenta. Además, en esta situación, ambas variables de control deben trabajar, por un instante, cerca de sus límites superiores.

Finalmente, en la Figura 15 vemos que el cambio de referencia solo se da para  $r_1$ . En esta situación, los tres controladores logran una respuesta suavizada para  $y_1$ , sin sobre-impulsos ni oscilaciones. Sin embargo, debido a que las variables controladas están acopladas por la dinámica de la planta, vemos que  $y_2$  sufre un cambio por su interacción con  $y_1$ , y en este caso, el ANN-MPC y el NMPC logran un menor sobre-impulso para  $y_2$ . Por otro lado, en la Figura 16 vemos que la variable de con-

tol  $u_1$  del MPC se comporta un poco oscilante pero con menos sobre-impulso.

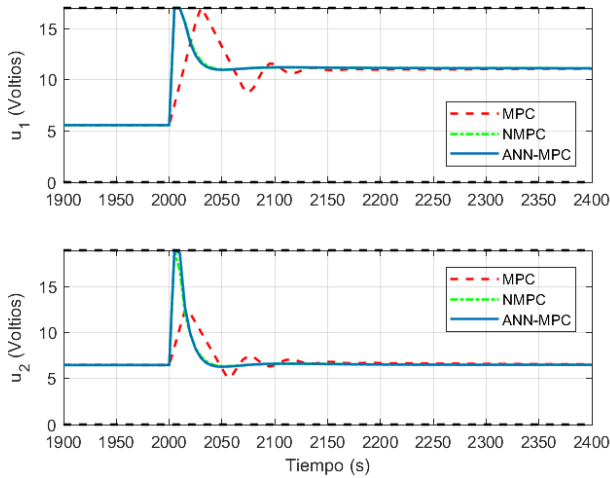


Figura 14: Variable de control del ANN-MPC (línea sólida azul) en comparación con el MPC lineal (línea punteada roja) y el MPC no lineal (línea punteada verde) ante cambios de referencia en el instante  $t=2000$  s.

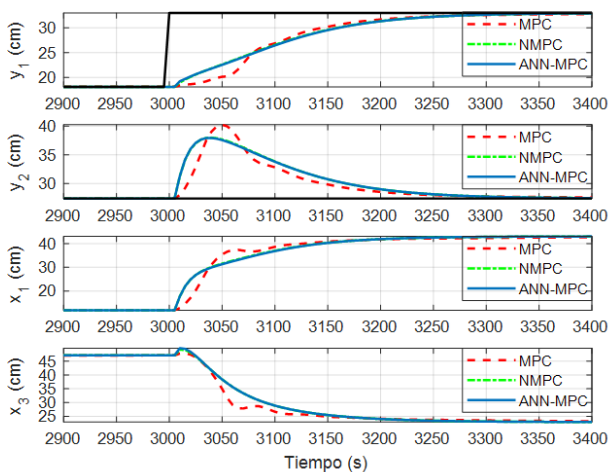


Figura 15: Desempeño del control ANN-MPC (línea sólida azul) en comparación con el MPC lineal (línea punteada roja) y el MPC no lineal (línea punteada verde) ante cambios de referencia (línea sólida negra) en el instante  $t=3000$  s.

Asimismo, para toda la prueba, el valor del índice de performance fue igual a 11916 para el NMPC, siendo este el menor de los tres, seguido por el ANN-MPC, con un valor igual a 11940, mientras que el MPC lineal obtuvo un valor igual a 20420, el que equivale a un 71 % más que el valor del NMPC. Por otro lado, el menor valor RMSE para la salida  $y_1$  la obtuvo el NMPC, con un valor igual a 2.066, mientras que el ANN-MPC arrojó un valor muy cercano a este, con 2.092, el cual representa solo un 1 % más del valor del NMPC. En tanto, el menor valor RMSE para  $y_2$  esta la vez lo obtuvo el ANN-MPC, con un valor de 2,124, seguido por el NMPC, con un valor de 2.149, que representa solo el 1 % más que el valor del ANN-MPC. Mientras que el MPC obtuvo un valor RMSE para  $y_2$  igual a 2.348 que representa un 10 % más que el valor del ANN-MPC.

De los resultados vemos que el ANN-MPC y el NMPC tienen un mejor desempeño. Sin embargo, hay que tener muy en cuenta que el NMPC requiere un mayor tiempo de cálculo, a diferencia del MPC lineal, que requiere menor tiempo que todos. Sin embargo, vemos que si bien, el tiempo de cálculo del ANN-MPC ha sido mayor al del MPC lineal, este se ha reducido considerablemente comparado con el NMPC, y sumado a su buen desempeño, podemos decir que el ANN-MPC ha logrado óptimos resultados para trabajar con sistemas MIMO de fase no mínima y no lineales. El resumen de todos estos resultados se muestra en la Tabla 7.

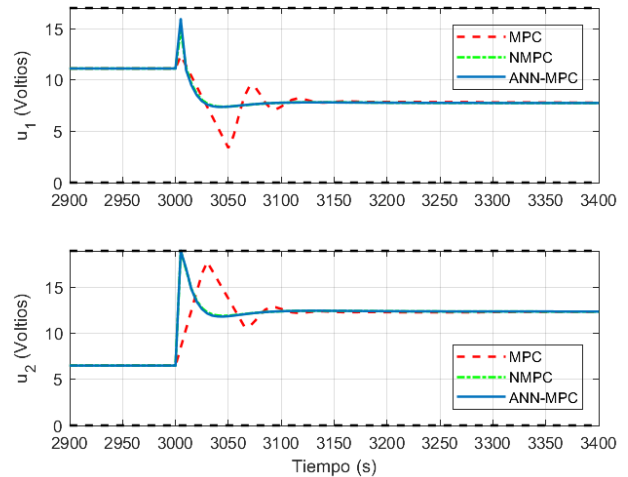


Figura 16: Variable de control del ANN-MPC (línea sólida azul) en comparación con el MPC lineal (línea punteada roja) y el MPC no lineal (línea punteada verde) ante cambios de referencia en el instante  $t=3000$  s.

Tabla 7: Resumen de resultados

Controlador	MPC	NMPC	ANN-MPC
$J_{Indice}$	20420	11916	11940
RMSE $y_1$	3.401	2.066	2.092
RMSE $y_2$	2.348	2.149	2.124
Sobre-impulso $y_1$ $t_{2000}$	38.29	30.52	25.91
Sobre-impulso $y_1$ $t_{3000}$	-	-	-
Acoplamiento $y_2$ $t_{3000}$	45.61	38.77	34.91
Tiempo de cálculo (s)	9.09	1746.0	53.73

## 5. Conclusiones

Este artículo presenta un Controlador Predictivo basado en un modelo de Redes Neuronales Artificiales (ANN-MPC) para controlar un sistema no lineal de tanque cuádruple de fase no mínima. Como paso preliminar, se generaron datos operativos de la planta de tanque cuádruple con los que se entrenó el modelo ANN, para luego realizar una prueba de predicción que mostró que el modelo ANN estimaba con bastante precisión los datos reales de la planta. Después, dicho modelo se integró al algoritmo de optimización de un MPC para predecir las salidas del sistema ante cambios de referencia.

Luego, se comparó el desempeño del controlador ANN-MPC con un MPC lineal y un MPC no lineal. Las pruebas de simulación evidenciaron que el ANN-MPC tiene un desempeño

cercano al del MPC no lineal, con respuestas más suaves y sin oscilaciones para alcanzar las referencias que forzaban al sistema a trabajar cerca de sus límites. Esto se debe a que las ANN son capaces de capturar relaciones no lineales complejas entre variables de entrada y salida. Por otro lado, el controlador ANN-MPC redujo considerablemente el tiempo de cálculo de la variable de control óptima en comparación con el NMPC. Con estos resultados se concluye que el enfoque ANN-MPC es capaz de controlar de manera óptima sistemas MIMO no lineales y de fase no mínima. Esto posibilita su aplicación en diversos sistemas no lineales de fase no mínima que posean un gran número de variables y una fuerte interacción entre ellas.

Como trabajo futuro, se comparará el desempeño de predicción de tres estructuras de ANN dinámicas: una red neuronal recurrente, una red LSTM (Long Short-Term Memory), y una ESN (Echo State Network). Asimismo, también se planteará la implementación del sistema de control visto en este artículo sobre la planta real.

### Agradecimientos

E. Calle reconoce el apoyo financiero del Proyecto Concytec- Banco Mundial "Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia Tecnología e Innovación Tecnológica" 8682-PE, a través de su unidad ejecutora Pro-Ciencia. [Contrato número 06-2018-FONDECYT/BM], para su trabajo de investigación denominado *Control de un sistema multivariable no lineal y en fase no mínima empleando un controlador predictivo neuronal*, ejecutado como parte del programa de Doctorado en Ingeniería con mención en Automatización, Control, y Optimización de Procesos, del Departamento de Ingeniería Mecánico-Eléctrica de la Universidad de Piura, Perú.

### Referencias

- Alvarado, I., Limon, D., Muñoz de la Peña, D., Maestre, J., Ridao, M., Scheu, H., Marquardt, W., Negenborn, R., De Schutter, B., Valencia, F., Espinosa, J., 2011. A comparative analysis of distributed mpc techniques applied to the hd-mpc four-tank benchmark. *Journal of Process Control* 21 (5), 800–815. DOI: 10.1016/j.jprocont.2011.03.003
- Amari, S.-i., 1993. Backpropagation and stochastic gradient descent method. *Neurocomputing* 5 (4), 185–196. DOI: 10.1016/0925-2312(93)90006-O.
- Bahar, A., Özgen, C., Leblebicioğlu, K., Halıcı, U., 2004. Artificial neural network estimator design for the inferential model predictive control of an industrial distillation column. *Industrial & Engineering Chemistry Research* 43 (19), 6102–6111. DOI: 10.1021/ie030585g.
- Brigham, K., Gupta, S., Brigham, J. C., 2018. Predicting responses to mechanical ventilation for preterm infants with acute respiratory illness using artificial neural networks. *Int J Numer Method Biomed Eng*, e3094. DOI: 10.1002/cnm.3094
- Calle, E., Oliden, J., 2021a. Iterative dual-gradient descent method for model predictive control with constraints. In: 2021 IEEE XXVIII International Conference on Electronics, Electrical Engineering and Computing (INTERCON). pp. 1–4. DOI: 10.1109/INTERCON52678.2021.9532891.
- Calle, E., Oliden, J., 2021b. Recurrent neural network based predictive control applied to 4 coupled-tank system. In: 2021 IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA). pp. 1–6. DOI: 10.1109/ICAACCA51523.2021.9465192
- Carvalho, C. B., Carvalho, E. P., Ravnani, M. A. S. S., 2020. Implementation of a neural network mpc for heat exchanger network temperature control. *Brazilian Journal of Chemical Engineering* 37 (4), 729–744. DOI: 10.1007/s43153-020-00058-2
- Colosi, T., Abrudean, M.-I., 2013. Numerical Simulation of Distributed Parameter Processes. Springer International Publishing. DOI: 10.1007/978-3-319-00014-5
- Cseko, L. H., Kvasnica, M., Lantos, B., 2015. Explicit mpc-based rbf neural network controller design with discrete-time actual kalman filter for semiactive suspension. *IEEE Transactions on Control Systems Technology* 23 (5), 1736–1753. DOI: 10.1109/TCST.2014.2382571.
- Dong, L., Yan, J., Yuan, X., He, H., Sun, C., 2019. Functional nonlinear model predictive control based on adaptive dynamic programming. *IEEE Transactions on Cybernetics* 49 (12), 4206–4218. DOI: 10.1109/TCYB.2018.2859801
- Gao, Y., Xu, X., Yang, Y., 2011. A new neural network for solving nonlinear programming problems. In: Liu, D., Zhang, H., Polycarpou, M., Alippi, C., He, H. (Eds.), *Advances in Neural Networks – ISNN 2011*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 565–571.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2 (5), 359–366. DOI: 10.1016/0893-6080(89)90020-8
- Johansson, K., 2000. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology* 8 (3), 456–465. DOI: 10.1109/87.845876
- Kouvaritakis, B., Cannon, M. (Eds.), 2016. *Model predictive control: classical, robust and stochastic*, 1st Edition. Springer Cham.
- Lucia, S., Koch, B., 2018. A deep learning-based approach to robust nonlinear model predictive control. *IFAC-PapersOnLine* 51 (20), 511–516, 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018. DOI: 10.1016/j.ifacol.2018.11.038
- Manonmani, A., Thyagarajan, T., Elango, M., Sutha, S., 2018. Modelling and control of greenhouse system using neural networks. *Transactions of the Institute of Measurement and Control* 40, 918e929. DOI: 10.1177/0142331216670235
- Nguyen, N.-S., 2020. Level control of quadruple tank system based on adaptive inverse evolutionary neural controller. *International Journal of Control, Automation and Systems* 18. DOI: 10.1007/s12555-019-0504-8.
- Pan, Y., Wang, J., 2010. A neurodynamic optimization approach to nonlinear model predictive control. In: 2010 IEEE International Conference on Systems, Man and Cybernetics. pp. 1597–1602. DOI: 10.1109/ICSMC.2010.5642367
- Pan, Y., Wang, J., 2012. Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. *IEEE Transactions on Industrial Electronics* 59 (8), 3089–3101. DOI: 10.1109/TIE.2011.2169636.
- Piga, D., Forgone, M., Formentin, S., Bemporad, A., 2019. Performance-oriented model learning for data-driven mpc design. *IEEE Control Systems Letters* 3 (3), 577–582. DOI: 10.1109/LCSYS.2019.2913347.
- Popoola, S. I., Jefia, A., Atayero, A. A., Kingsley, O., Faruk, N., Oseni, O. F., Abolade, R. O., 2019. Determination of neural network parameters for path loss prediction in very high frequency wireless channel. *IEEE Access* 7, 150462–150483. DOI: 10.1109/ACCESS.2019.2947009.
- Rawlings, J. B., Mayne, D. Q., Moritz, M. D. (Eds.), 2017. *Model predictive control: theory, computation, and design*, 2nd Edition. Nob Hill Publishing.
- Saadon, A., Abdullah, J., Muhammad, N. S., Ariffin, J., Sep 2020. Development of riverbank erosion rate predictor for natural channels using narx-qr factorization model: a case study of sg. bernam, selangor, malaysia. *Neural Computing and Applications* 32 (18), 14839–14849. DOI: 10.1007/s00521-020-04835-5
- Sarali, D. S., Agnes Idhaya Selvi, V., Pandiyan, K., 2019. An improved design for neural-network-based model predictive control of three-phase inverters. In: 2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES). pp. 1–5. DOI: 10.1109/INCCES47820.2019.9167697
- Shin, Y., Smith, R., Hwang, S., 2020. Development of model predictive control system using an artificial neural network: A case study with a distillation column. *Journal of Cleaner Production* 277, 124124. DOI: 10.1016/j.jclepro.2020.124124
- Tang, W., Daoutidis, P., 2019. Distributed control and optimization of process system networks: A review and perspective. *Chinese Journal of Chemical Engineering* 27 (7), 1461–1473. DOI: 10.1016/j.cjche.2018.08.027.
- Vieira, W., Santos, V., Carvalho, F., Pereira, J., Fileti, A., 2005. Identification and predictive control of a fcc unit using a mimo neural model. *Chemical Engineering and Processing: Process Intensification* 44 (8), 855–868. DOI: 10.1016/j.cep.2004.08.008
- Xu, J., Li, C., He, X., Huang, T., 2016. Recurrent neural network for solving model predictive control problem in application of four-tank benchmark. *Neurocomputing* 190, 172–178. DOI: 10.1016/j.neucom.2016.01.020.
- Yan, Y., Xu, Q., 2019. Neural networks-based model predictive control for precision motion control of a micropositioning system. In: 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR). pp. 252–257. DOI: 10.1109/RCAR47638.2019.9043934.