

ANÁLISIS FORMAL DE LA DINÁMICA DE SISTEMAS NO LINEALES MEDIANTE REDES NEURONALES

Eloy Irigoyen, Mikel Larrea

DISA, EIB, UPV/EHU, Alda. Urquijo, S/N, 48013, Bilbao; {eloy.irigoyen; m.larrea}@ehu.eus

A.Javier Barragán, Miguel Ángel Martínez, José Manuel Andújar
DIESIA, Universidad de Huelva, Campus de La Rábida, 21819, Huelva;
{antonio.barragan; bohorquez; andujar}@diesia.uhu.es

Resumen

A la hora de estudiar un sistema (sistema industrial, biológico, económico, etc.), llegar a enfocar su análisis desde un punto de vista dinámico puede ser de gran utilidad e interés, en función de los requerimientos que condicionen dicho análisis. Mediante un análisis dinámico se puede conocer qué respuesta tendrá un sistema ante determinados estímulos de entrada, cuál es su comportamiento, si es estable en lazo abierto (local y globalmente), incluso si viene afectado por fenómenos no lineales, como bifurcaciones y ciclos límites, por ejemplo. En ocasiones donde el sistema es desconocido o su dinámica es lo suficientemente compleja como para formalizar su análisis dinámico a través de modelos matemáticos, las herramientas basadas en técnicas de Control Inteligente, como son las Redes Neuronales, cobran vital importancia ya que permiten abordar el problema a través de modelos del sistema. La literatura ha demostrado que los modelos neuronales son aproximadores universales tanto de una función como de su derivada, por lo que en base a datos de entrada/salida permiten modelar sistemas no lineales. Considerando que un modelo neuronal es un modelo matemático formalmente hablando, en base al mismo se pueden estudiar aspectos de la dinámica del sistema modelado real, del mismo modo a como se desarrolla en la teoría de control no lineal. Este trabajo presenta una metodología de estudio de los estados de equilibrio de un sistema no lineal, la linealización exacta de su modelo de estado neuronal y el estudio de la estabilidad local de los equilibrios a partir de dicha linealización. A partir de dicha información, es posible estudiar la estabilidad local de los estados de equilibrio, así como la dinámica del sistema en su entorno y la presencia de oscilaciones, obteniéndose una valiosa información del comportamiento dinámico del sistema.

Palabras clave: Análisis dinámico, estabilidad, estado de equilibrio, linealización, modelado neuronal, sistemas dinámicos, red neuronal MLP.

1. INTRODUCCIÓN

Mediante el análisis dinámico de un sistema se puede conocer su comportamiento y qué respuesta presentará ante diferentes variaciones en su entrada. El análisis formal de dicho sistema se convierte en tarea inalcanzable si no es posible determinar un modelo matemático, bien por ser un sistema desconocido, o parcialmente desconocido, bien porque su dinámica es lo suficientemente compleja [1]. Al no conocerse dicho modelo matemático se puede proceder generando un modelo en base a datos de entrada-salida, y con dicho modelo, realizar un estudio de la dinámica del sistema original.

Tradicionalmente, los métodos de ingeniería de control utilizados en muchas aplicaciones se han basado en enfoques lineales. Esto es debido a que los problemas han podido ser estudiados y resueltos en un ámbito local. Pero sin embargo, hoy en día muchos de dichos problemas se enfocan desde un punto de vista global, lo que lleva al uso de métodos no lineales.

Una de las razones principales que recomienda la utilización de los modelos no lineales es que los modelos lineales no tienen una dinámica suficientemente rica como para presentar una serie de fenómenos que se dan comúnmente en la vida real [2, 3, 4]. El comportamiento dinámico de los sistemas lineales, sin tener en cuenta su orden, está gobernado fundamentalmente por los autovalores de su matriz de estado. Por el contrario, los sistemas no lineales muestran un comportamiento mucho más rico, pudiendo producirse oscilaciones autoexcitadas, conocidas como ciclos límite, un comportamiento aperiódico y críticamente sensible a las condiciones iniciales [5], el caos [6], así como otros fenómenos dinámicos exclusivos de los sistemas no lineales, como la existencia de múltiples estados de equilibrio y bifurcaciones [7], entre otros.

Para el análisis de un sistema no lineal, su representación mediante un modelo matemático, generalmente mediante un modelo de estado, es el planteamiento inicial idóneo. Esto es posible si se dispone de suficiente conocimiento del sistema como para generar sus ecuaciones de estado, siempre y cuando su dinámica no sea excesivamente com-

pleja. Pero la realidad es que en múltiples ocasiones el sistema objeto de estudio puede ser demasiado complejo, ya sea por su dinámica o por su dimensión, o simplemente porque se desconocen las ecuaciones que gobiernan su funcionamiento. Las técnicas procedentes del área del Control Inteligente pueden ayudar a mejorar estos estudios, como es el caso de la Lógica Difusa [8, 9]. De igual modo, en estas situaciones las redes neuronales (RN) son una poderosa herramienta ya que permite modelar sistemas altamente no lineales a partir de datos de entrada-salida. Esta modelización puede ser cualitativa [10, 11, 12, 4] o estrictamente analítica, aprovechando el hecho de que las RN son aproximadores universales tanto de la función [13, 14, 15] como de su derivada [16, 17]. De este modo, aunque el sistema sea desconocido, es posible obtener un modelo neuronal del mismo. Formalmente hablando, un modelo neuronal es un modelo matemático. Por lo tanto, a partir de este modelo es posible estudiar aspectos de la dinámica del sistema real siempre que su precisión sea suficientemente alta. Esta forma de abordar el problema permite enfrentarse a sistemas no lineales, cuya modelización por técnicas matemáticas tradicionales puede resultar realmente difícil.

A la hora de analizar o diseñar una solución de control, el conocimiento de los estados de equilibrio que posee un sistema, así como la estabilidad de dichos estados, son aspectos que resultan de máximo interés. En el caso en el que el sistema es completamente desconocido, esta información puede ayudar a aclarar el funcionamiento del mismo, así como facilitar el diseño de un control adecuado. Es de destacar que, si bien existen trabajos previos que presentan una novedosa metodología de análisis basada en lógica difusa [9, 8], no se ha encontrado ningún trabajo que lo haga desde el punto de vista de las RN.

Este trabajo presenta una metodología para analizar un sistema no lineal bajo un enfoque generalista, el cual es inicialmente desconocido, a partir de un modelo neuronal. En el apartado 2 se introduce la formulación del problema y se ofrecen las ecuaciones que se utilizarán para la representación del modelo del sistema. En el apartado 3 se presenta la linealización exacta de un modelo neuronal general. A continuación, en el apartado 4 se plantea una propuesta de procedimiento para el análisis dinámico del sistema. Finalmente se aplica dicho procedimiento a dos ejemplos y se presentan las correspondientes conclusiones.

2. FORMULACIÓN DEL PROBLEMA

Sea n el orden del sistema y m su número de entradas, un modelo neuronal MLP equivalente de un sistema continuo, que estima tanto sus variables de estado, como salida, puede venir representado por [18, 19, 20, 21]:

$$\begin{aligned}\hat{x}(t+1|\theta) &= g[\varphi(t, \theta), \theta] \\ \hat{y}(t|\theta) &= C(\theta)\hat{x}(t|\theta)\end{aligned}\quad (1)$$

$$\text{siendo } \varphi(t, \theta) = \begin{bmatrix} \hat{x}(t|\theta) \\ u(t) \end{bmatrix},$$

donde $\varphi(t, \theta)$ es el vector de regresión, $u(t)$ las entradas al sistema y θ el vector de parámetros (ω, W) de la RN, los pesos y bias de sus capas.

De esta forma, la salida de una RN con una capa oculta y funciones de activación sigmoideal, junto a una capa de salida lineal, puede calcularse a través de la expresión siguiente, que representa el modelo neuronal de estado del sistema [20]:

$$\begin{aligned}\dot{x}_i &= f_i(\mathbf{x}, \mathbf{u}) \equiv \hat{x}(t+1|\theta) = \\ &= \sum_{j=1}^{n_h} W_{i,j} F_j \left[\sum_{l=1}^{n+m} \omega_{j,l} \varphi_l(t) + \omega_{j,0} \right] + W_{i,0},\end{aligned}\quad (2)$$

donde $W_{i,j}$ son los pesos y bias de la capa de salida, ω_j son los pesos y bias de la capa oculta, n_h el número de neuronas de la capa oculta, $\varphi_l(t)$ es el vector de entrada a la RN en cada instante t y $n+m$ su dimensión. Considerando $neta$ el resultado de la combinación de entradas y pesos a cada neurona y que F_j es la función de activación de las neuronas de la capa oculta, se tiene:

$$neta = \sum_{l=1}^{n+m} \omega_{j,l} \varphi_l(t) + \omega_{j,0} \quad (3)$$

$$F_j(neta) = \frac{e^{neta} - e^{-neta}}{e^{neta} + e^{-neta}}, \quad (4)$$

donde F_j representa la función tangente hiperbólica.

3. LINEALIZACIÓN DE UN MODELO NEURONAL DE ESTADO

La linealización es una de las técnicas más utilizadas en el diseño de sistemas de control no lineales. Aunque es una técnica que no se debe emplear en muchos casos, ya que se desprecian los efectos de las no linealidades de los sistemas controlados, sí

que puede ser válida para controlar sistemas no muy complejos, o cuya dinámica se conozca de forma aproximada en regiones donde su comportamiento se acerque mucho a uno lineal.

Aparte de ser utilizada como método de control, la linealización puede emplearse para obtener información de un sistema no lineal. Se puede considerar que, con algunas salvedades, el comportamiento de un sistema no lineal en torno a un estado de equilibrio es muy similar al del sistema linealizado en torno a dicho estado [22, 7, 23]; por lo tanto, la obtención del sistema lineal equivalente de un modelo neuronal no lineal puede ser una herramienta muy útil para obtener información del sistema original.

Sea el modelo de estado genérico de un sistema no lineal dado por:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (5)$$

el desarrollo en serie de Taylor hasta orden 1 de dicho sistema en torno a un punto $(\mathbf{x}_0, \mathbf{u}_0)$, determina que este se puede aproximar por:

$$\dot{\mathbf{x}}(t) \approx \mathbf{A}_0 + \mathbf{A}\bar{\mathbf{x}}(t) + \mathbf{B}\bar{\mathbf{u}}(t) \quad (6)$$

siendo $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$ y $\bar{\mathbf{u}} = \mathbf{u} - \mathbf{u}_0$, y donde las matrices del sistema lineal se calculan mediante las expresiones (7), (8) y (9).

$$\mathbf{A}_0 = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) \quad (7)$$

$$\dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left. \frac{\partial f_i}{\partial x_q} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q + \sum_{v=n+1}^{n+m} \left. \frac{\partial f_i}{\partial u_v} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v, \quad (10)$$

$$\begin{aligned} \dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left[\left. \frac{\partial}{\partial x_q} \left(\sum_{j=1}^{n_h} W_{i,j} F_j \left[\sum_{l=1}^n \omega_{j,l} \varphi_l(t) + \omega_{j,0} \right] + W_{i,0} \right) \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q \right] \\ + \sum_{v=n+1}^{n+m} \left[\left. \frac{\partial}{\partial u_v} \left(\sum_{j=1}^{n_h} W_{i,j} F_j \left[\sum_{l=n+1}^{n+m} \omega_{j,l} \varphi_l(t) + \omega_{j,0} \right] + W_{i,0} \right) \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v \right], \quad (11) \end{aligned}$$

$$\begin{aligned} \dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left[\left. \frac{\partial \left(\sum_{j=1}^{n_h} W_{i,j} F_j \left[\sum_{l=1}^n \omega_{j,l} \varphi_l(t) + \omega_{j,0} \right] \right)}{\partial x_q} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q + \right. \\ \left. + \sum_{v=n+1}^{n+m} \left[\left. \frac{\partial \left(\sum_{j=1}^{n_h} W_{i,j} F_j \left[\sum_{l=n+1}^{n+m} \omega_{j,l} \varphi_l(t) + \omega_{j,0} \right] \right)}{\partial u_v} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v \right] \right] \quad (12) \end{aligned}$$

Si el punto $(\mathbf{x}_0, \mathbf{u}_0)$ es un estado de equilibrio del sistema, la matriz \mathbf{A}_0 será nula, ya que por definición un estado de equilibrio hace nula la ecuación de estado.

$$\mathbf{A} = \left(\begin{array}{ccc} \left. \frac{\partial f_1(\mathbf{x}, \mathbf{u})}{\partial x_1} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} & \cdots & \left. \frac{\partial f_1(\mathbf{x}, \mathbf{u})}{\partial x_n} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \\ \vdots & \ddots & \vdots \\ \left. \frac{\partial f_n(\mathbf{x}, \mathbf{u})}{\partial x_1} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} & \cdots & \left. \frac{\partial f_n(\mathbf{x}, \mathbf{u})}{\partial x_n} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \end{array} \right) \quad (8)$$

$$\mathbf{B} = \left(\begin{array}{ccc} \left. \frac{\partial f_1(\mathbf{x}, \mathbf{u})}{\partial u_1} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} & \cdots & \left. \frac{\partial f_1(\mathbf{x}, \mathbf{u})}{\partial u_m} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \\ \vdots & \ddots & \vdots \\ \left. \frac{\partial f_n(\mathbf{x}, \mathbf{u})}{\partial u_1} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} & \cdots & \left. \frac{\partial f_n(\mathbf{x}, \mathbf{u})}{\partial u_m} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \end{array} \right) \quad (9)$$

Si el sistema (5) se representa mediante un modelo neuronal, su equivalente matemático viene dado por (2). Si se linealiza esta expresión en torno al punto $(\mathbf{x}_0, \mathbf{u}_0)$, se puede representar el modelo matemático equivalente de un sistema neuronal linealizado (10), donde \bar{x}_q y \bar{u}_v son las componentes de los vectores $\bar{\mathbf{x}}_0$ y $\bar{\mathbf{u}}_0$, respectivamente, los cuales a su vez componen el conjunto de entradas $(n + m)$ a la RN. Operando con la expresión anterior se obtiene (11), que desarrollando se puede expresar como (12).

A continuación, en base a los desarrollos de Pirabakaran y Becerra [10] y su extensión general a problemas de identificación de Larrea [24], junto a la función de activación $F_j(neta)$ de la RN descrita en la ecuación (4), se resolverán independientemente cada una de las derivadas parciales anteriores. En primer lugar, la primera derivada de (12) queda representada por:

$$\frac{\partial \left(\sum_{j=1}^{n_h} W_{i,j} F_j \left[\sum_{l=1}^n \omega_{j,l} \varphi_l(t) + \omega_{j,0} \right] \right)}{\partial x_q} = \sum_{j=1}^{n_h} W_j o_j (1 - o_j) \omega_{j,q} \quad (13)$$

donde o_j es la salida de la neurona j de la capa oculta y q toma valores enteros en el intervalo $[1, n]$.

Para el cálculo de la segunda derivada se procede del mismo modo, obteniéndose la siguiente expresión:

$$\frac{\partial \left(\sum_{j=1}^{n_h} W_{i,j} F_j \left[\sum_{l=n+1}^{n+m} \omega_{j,l} \varphi_l(t) + \omega_{j,0} \right] \right)}{\partial u_v} = \sum_{j=1}^{n_h} W_j o_j (1 - o_j) \omega_{j,v} \quad (14)$$

donde v toma valores enteros en el intervalo $[n + 1, n + m]$.

Sustituyendo las expresiones (13) y (14) en (12), el modelo matemático equivalente de un sistema neuronal linealizado en torno a un punto $(\mathbf{x}_0, \mathbf{u}_0)$ viene dado por la siguiente expresión:

$$\dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left[\sum_{j=1}^{n_h} W_j o_j (1 - o_j) \omega_{j,q} \right] \bar{x}_q + \sum_{v=n+1}^{n+m} \left[\sum_{j=1}^{n_h} W_j o_j (1 - o_j) \omega_{j,v} \right] \bar{u}_v. \quad (15)$$

4. ESTADOS DE EQUILIBRIO Y ESTABILIDAD LOCAL

Para el procedimiento de análisis, el primer paso a dar va enfocado en la obtención de un modelo neuronal de estado del sistema a analizar, como el dado por (1), mediante alguna de las muchas técnicas de modelado existentes [20], ya sea en línea [25, 26], o fuera de línea [4, 27]. Seguidamente es importante la localización de los estados de

equilibrio del sistema. La búsqueda de los estados de equilibrio de un sistema de control es uno de los primeros problemas que se deben resolver para proceder con el estudio de su dinámica. Como análisis previo al diseño del sistema de control, la obtención de los estados de equilibrio proporciona una valiosa información sobre el comportamiento de este, especialmente si se trata de un sistema no lineal, ya que estos puntos son candidatos ideales para el estudio de estos sistemas mediante técnicas de linealización. Una vez diseñado el controlador, la obtención de los estados de equilibrio del sistema realimentado puede ser útil para comprobar que el diseño se ha realizado adecuadamente, así como para poder aplicar análisis de estabilidad a dichos estados.

Para el caso de los sistemas neuronales, y dada la inherente capacidad de este tipo de modelos para representar dinámicas no lineales, se hace necesaria la utilización de métodos numéricos capaces de resolver el sistema de ecuaciones no lineales que se plantea:

$$\begin{aligned} \sum_{j=1}^{n_h} W_{1,j} F_j(neta_j) + W_{1,0} &= 0 \\ \sum_{j=1}^{n_h} W_{2,j} F_j(neta_j) + W_{2,0} &= 0 \\ &\vdots \\ \sum_{j=1}^{n_h} W_{n+m,j} F_j(neta_j) + W_{n+m,0} &= 0 \end{aligned} \quad (16)$$

En el análisis de los sistemas dinámicos no lineales, el cálculo de los estados de equilibrio del sistema puede convertirse en un problema de envergadura. A diferencia de los sistemas lineales que tienen uno o infinitos estados de equilibrio, un sistema no lineal puede tener uno, ninguno, cualquier número finito o infinitos estados de equilibrio. En este trabajo se propone emplear métodos numéricos para resolver el sistema de ecuaciones no lineales dado en (16), concretamente el método de Levenberg-Marquardt [28], extendido en el trabajo de Moré [29]. Este algoritmo requiere de un punto inicial para comenzar la búsqueda. Con objeto de maximizar las posibilidades de encontrar todos los estados de equilibrio existentes, se procede a inicializar el algoritmo mediante un barrido de puntos a lo largo de todo el universo de discurso, tanto de las variables de estado, como de las variables de entrada del sistema. Existen otras alternativas tanto dentro del ámbito de los algoritmos numéricos como en el de los bioinspirados para poder realizar esta búsqueda, pero los autores se han decantado por el algoritmo de Levenberg-Marquardt por ofrecer una rápida convergencia y precisión en los resultados [30, 31]. Este algoritmo inicializado en una malla de puntos es capaz de encontrar los

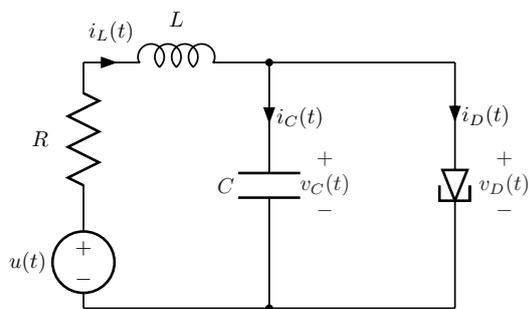


Figura 1: Circuito diodo túnel.

múltiples estados de equilibrio que pueden presentarse en un sistema no lineal con mayor rapidez que los algoritmos genéticos.

El funcionamiento del algoritmo de Levenberg-Marquardt requiere del uso de la matriz Jacobiana del sistema, la cual puede aproximarse o calcularse de forma explícita para mejorar la convergencia. El cálculo de la matriz Jacobiana de un modelo neuronal se ha resuelto de forma general en la sección anterior, por lo que puede incluirse en el algoritmo numérico para mejorar su precisión y velocidad de convergencia.

La misma matriz Jacobiana empleada para la resolución de los estados de equilibrio puede emplearse para linealizar el sistema en cada una de las soluciones obtenidas. De esta forma también es posible estudiar las características de los estados de equilibrio encontrados a partir de los autovalores de la matriz dinámica del sistema linealizado. Este análisis puede ayudar a comprender la dinámica del sistema, su estabilidad local, y a los ojos de un experto, incluso se pueden observar comportamientos más complejos como bifurcaciones o ciclos límite.

5. EJEMPLO

En esta sección se presenta un ejemplo de análisis a partir de su modelo neuronal, para un sistema no lineal que inicialmente se considerará desconocido. El sistema elegido es un caso de estudio ampliamente utilizado en estudios y análisis de sistemas no lineales [9, 8]

Sea el circuito diodo-túnel mostrado en la figura 1, donde $R = 1,5 \text{ k}\Omega$, $C = 2 \text{ pF}$ y $L = 5 \text{ }\mu\text{H}$, $x_1(t) = v_C(t)$, $x_2(t) = i_L(t)$ son las variables de estado del sistema, y $h(v)$ la característica $v_D - i_D$ no lineal del diodo-túnel mostrada en la figura 2 [32].

Suponiendo desconocida la dinámica del sistema, el procedimiento a seguir es obtener un modelo neuronal del sistema a partir de datos de entrada-

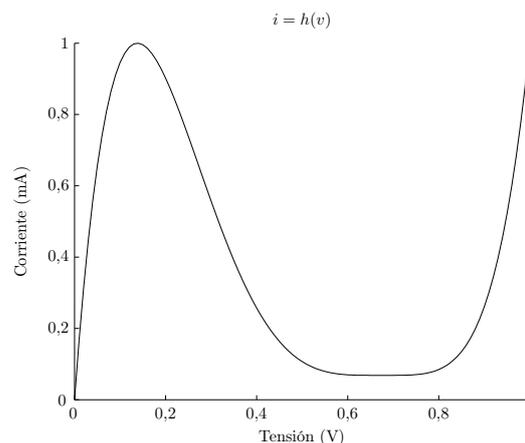


Figura 2: Característica V-I del diodo túnel.

salida, y analizar a continuación dicho modelo con objeto de extraer conclusiones sobre la dinámica del sistema.

Para modelar el sistema se ha empleado un conjunto de 3000 datos distribuidos uniformemente en el universo de discurso $x_1(t) \in [0, 1]$, $x_2(t) \in [0, 2]$ y $u(t) \in [0, 2]$. Realizando el ajuste de los parámetros de la red neuronal MLP, de estructura 3-3-1, mediante un procedimiento de aprendizaje basado en el algoritmo de entrenamiento de segundo orden de Levenberg-Marquardt [28, 33] se han obtenido unos errores cuadráticos medios (MSE) durante la validación con otro conjunto de datos de $0,1147 \text{ V/ns}$ y $3,03 \times 10^{-5} \text{ mA/ns}$, respectivamente.

A continuación se ha empleado el algoritmo de minimización de Levenberg-Marquardt [28, 29] para buscar las soluciones al sistemas de ecuaciones planteado en (16). Para ello se ha partido de una malla de puntos iniciales cuyos límites coinciden con el universo de discurso de cada una de las variables del sistema, empleando un incremento de 0,05 en todas ellas. En la figura 3 se muestran las soluciones del sistema real junto con las soluciones obtenidas por el algoritmo numérico sobre el modelo neuronal, descartando aquellas que se sitúan fuera del universo de discurso por carecer de validez.

Empleando la linealización exacta desarrollada en este artículo se han obtenido los modelos linealizados en cada uno de los estados de equilibrio mostrados en la figura, y se han caracterizado según su comportamiento (según los tipos de autovalores de la matriz dinámica del sistema linealizado). Como se puede apreciar en la figura, los estados de equilibrios del sistema real han sido hallados con bastante exactitud, incluso su tipo se corresponde con el obtenido a partir del modelo neuronal. Cabe mencionar que el algoritmo numérico ha localizado

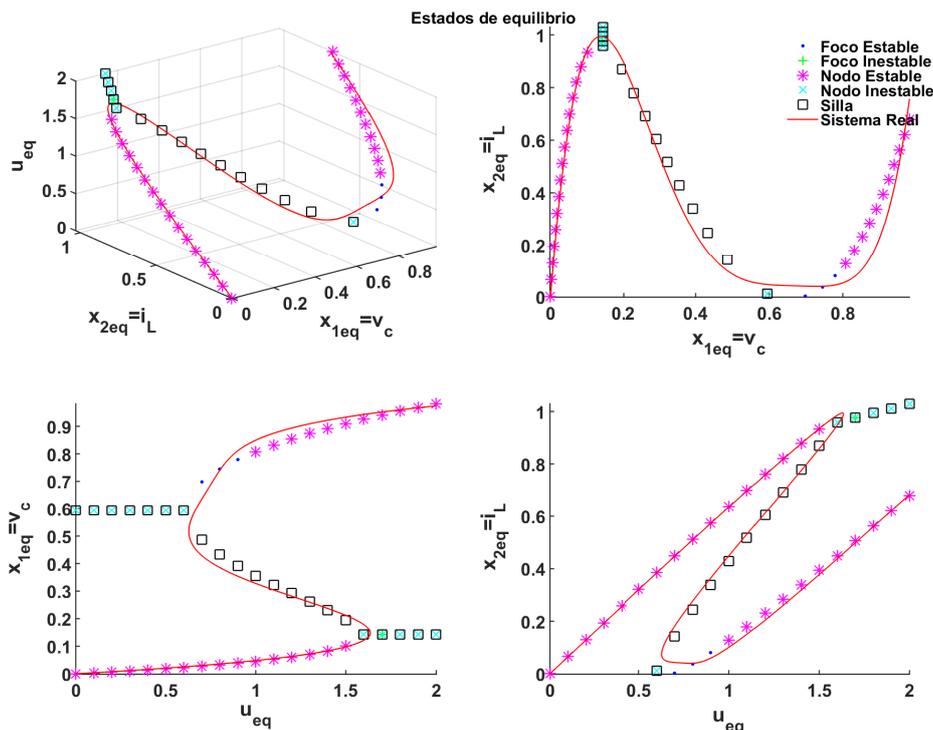


Figura 3: Estados de equilibrio localizados dentro del universo de discurso del modelo neuronal.

una serie de estados de equilibrio en el modelo neuronal que no existen en el sistema real. Este error puede deberse al propio error inherente de modelado o al algoritmo de minimización. Al menos en este caso, estos estados son fácilmente descartables porque, a diferencia del resto, no presentan un tipo de equilibrio coherente entre sí, coincidiendo prácticamente en un mismo punto distintos tipos de estados de equilibrio.

6. CONCLUSIONES

En este artículo se ha planteado el análisis dinámico de un sistema desconocido a través de la obtención de un modelo neuronal, basado en una sencilla estructura MLP, a partir de datos de entrada-salida del mismo. Se ha demostrado a través de un ejemplo ampliamente utilizado cómo se puede extraer información muy valiosa únicamente a partir del análisis del modelo neuronal, como la localización de los estados de equilibrio de un sistema y el estudio de su estabilidad local, así como la presencia y estabilidad de oscilaciones en el sistema.

Agradecimientos

Los autores quieren agradecer a la Viceconsejería de Universidades e Investigación, del Departamento de Educación, Política Lingüística y Cultura del Gobierno Vasco, por la ayuda del Programa de Movilidad del Personal Investigador concedida

al Dr.D. Eloy Irigoyen, con número de expediente MV_2017_1.0032, la cual ha dado lugar al desarrollo del presente trabajo. También desean agradecer a la Secretaría de Estado de Investigación, Desarrollo e Innovación del Ministerio de Economía y Competitividad por la financiación asignada a la Red de Excelencia, con referencia DPI2015-71320-REDT.

Referencias

- [1] S. Chen and S. A. Billings. Neural networks for nonlinear dynamic system modelling and identification. *International Journal of Control*, 56(2):319–346, 1992.
- [2] H. J. Marquez. *Nonlinear control systems. Analysis and design*. John Wiley & Sons, Inc., 2003.
- [3] José Manuel Andújar and José Manuel Bravo. Multivariable fuzzy control applied to the physical-chemical treatment facility of a cellulose factory. *Fuzzy Sets and Systems*, 150(3):475–492, March 2005.
- [4] E. Irigoyen and G. Miñano. A narx neural network model for enhancing cardiovascular rehabilitation therapies. *Neurocomputing*, 109:9 – 15, 2013. New trends on Soft Computing Models in Industrial and Environmental Applications.

- [5] H. K. Khalil. *Nonlinear systems*. Prentice-Hall, NJ, 2000.
- [6] Stephen Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Texts in Applied Mathematics. Springer, 2nd edition, October 2003.
- [7] S. Sastry. *Nonlinear system: analysis, stability, and control*. Springer, New York, 1999.
- [8] A.J. Barragán, B.M. Al-Hadithi, A. Jiménez, and J.M. Andújar. A general methodology for online TS fuzzy modeling by the extended kalman filter. *Applied Soft Computing*, 18(0):277–289, May 2014.
- [9] A.J. Barragán, B.M. Al-Hadithi, J.M. Andújar, and A. Jiménez. Metodología formal de análisis del comportamiento dinámico de sistemas no lineales mediante lógica borrosa. *Revista Iberoamericana de Automática e Informática Industrial*, 12(4):434 – 445, 2015.
- [10] K. Pirabakaran and V.M. Becerra. Pid auto-tuning using neural networks and model reference adaptive control. *IFAC Proceedings Volumes*, 35(1):451 – 456, 2002. 15th IFAC World Congress.
- [11] Nand Kishor, R. P. Saini, and S. P. Singh. Small hydro power plant identification using nnarx structure. *Neural Computing & Applications*, 14(3):212–222, Sep 2005.
- [12] R.K. Al Seyab and Y. Cao. Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation. *Journal of Process Control*, 18(6):568 – 581, 2008.
- [13] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.
- [14] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.
- [15] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183 – 192, 1989.
- [16] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551 – 560, 1990.
- [17] A. Ronald Gallant and Halbert White. On learning the derivatives of an unknown mapping with multilayer feedforward networks. *Neural Networks*, 5(1):129 – 138, 1992.
- [18] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, Mar 1990.
- [19] S. Jagannathan F.L. Lewis and A. Yesildirek. *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor & Francis, 1999.
- [20] N.K. Poulsen M. Nørgaard, O. Ravn and L.K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook*. Springer, 2000.
- [21] S. Jagannathan. *Neural Network Control of Nonlinear Discrete-time Systems*. Francis & Taylor, 2006.
- [22] Henk Nijmeijer and Arjan van der Schaft. *Nonlinear dynamical control systems*. Springer Verlag, Berlin, 1990.
- [23] Jean-Jacques E. Slotine and Weiping Li. *Applied nonlinear control*. Prentice-Hall, NJ, 1991.
- [24] Mikel Larrea. *Estrategia Adaptativa-Predictiva Basada en Redes Neuronales para el Control de Sistemas No Lineales*. phdthesis, (UPV/EHU), December 2014. ISBN-10: 1512179191 ISBN-13: 978-1512179194 Editor: CreateSpace Independent Publishing Platform.
- [25] Qing Song, Yilei Wu, and Yeng Chai Soh. Robust adaptive gradient-descent training algorithm for recurrent neural networks in discrete time domain. *Neural Networks, IEEE Transactions on*, 19(11):1841–1853, Nov. 2008.
- [26] T. Dierks and S. Jagannathan. Output feedback control of a quadrotor uav using neural networks. *Neural Networks, IEEE Transactions on*, 21(1):50–66, 2010.
- [27] Asier Zubizarreta, Mikel Larrea, Eloy Irigoyen, Itziar Cabanes, and Eva Portillo. Real time direct kinematic problem computation of the 3prs robot using neural networks. *Neurocomputing*, pages–, 2017.
- [28] K. Levenberg. A method for the solution of certain problems in least squares. In *Quart. Appl. Math.*, volume 2, pages 164–168, 1944.

- [29] Jorge J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G.A. Watson, editor, *Numerical Analysis*, pages 105–116. Springer Verlag, Berlin, 1977.
- [30] Zsolt Ugray, Leon Lasdon, John Plummer, Fred Glover, James Kelly, and Rafael Martí. Scatter search and local nlp solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.
- [31] Fred Glover. *A template for scatter search and path relinking*, pages 1–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.
- [32] Leon O. Chua, Charles A. Desoer, and Ernest S. Kuh. *Linear and nonlinear circuits*. McGraw-Hill series in electrical and computer engineering: Circuits and systems. McGraw-Hill Book Company, New York, 1987.
- [33] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, Nov 1994.