

# APLICACIÓN DE LA MATRIZ JACOBIANA DE UN SISTEMA DE CONTROL BORROSO A LA OBTENCIÓN DE SUS ESTADOS DE EQUILIBRIO

José Manuel Andújar Márquez  
andujar@diesia.uhu.es

Antonio Javier Barragán Piña  
antonio.barragan@diesia.uhu.es

Manuel Emilio Gegúndez Arias  
gegundez@dmate.uhu.es

Francisco J. Toro Negro  
ftoro@diesia.uhu.es

Dpto. Ingeniería Electrónica, Sistemas Informáticos y Automática, Universidad de Huelva, E.P.S. La Rábida,  
Carretera Huelva-La Rábida s/n, 21071 Palos de la Frontera (Huelva)

## Resumen

*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. Para el caso de los sistemas borrosos, 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. En este artículo se presenta un modelo matemático equivalente simplificado de un sistema de control multivariable no lineal basado en la teoría de la lógica borrosa. Posteriormente se utiliza este modelo para calcular los estados de equilibrio del sistema. Para ello se utilizarán algoritmos que requieren el cálculo de la matriz Jacobiana. Como se demuestra en este trabajo, la incorporación del algoritmo desarrollado para calcular la matriz Jacobiana mejora tanto la velocidad de convergencia como la precisión de los algoritmos numéricos de uso habitual en la literatura.*

**Palabras Clave:** Algoritmo, Control Borroso, Estados de Equilibrio, Matriz Jacobiana.

## 1 INTRODUCCIÓN

Una de las críticas usuales al uso de la lógica borrosa en el diseño de sistemas de control ha sido su escasa formalización matemática –las soluciones son a menudo empíricas y requieren del conocimiento de un experto- y la ausencia de herramientas estándares formalizadas que eviten el que cada problema sea abordado con una metodología diferente.

Los autores de este trabajo están interesados en la formalización de los sistemas de control borroso [1,2,

3, 4], de modo que las herramientas y metodologías de análisis y síntesis de sistemas de control aceptadas por la comunidad científica, sean de aplicación a este tipo de sistemas.

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.

**Definición.** Un vector  $\mathbf{x}$  es un estado de equilibrio de un sistema dinámico, si una vez que el vector de estados del sistema es igual a  $\mathbf{x}$ , se mantiene en esa posición por tiempo indefinido mientras el sistema no sea perturbado.

Casi la totalidad de los métodos numéricos de minimización de funciones, emplean la matriz Jacobiana de la función a minimizar para mejorar la convergencia del algoritmo y la precisión de su resultados [7, 7, 10, 11, 12, 19]. Si no se dispone de este valor se suelen emplear aproximaciones, pero con la consiguiente pérdida de eficiencia y precisión del algoritmo.

En este trabajo se desarrolla matemáticamente un modelo de planta identificada mediante técnicas de lógica borrosa a la que se le aplica un controlador borroso, obteniendo explícitamente el modelo del sistema realimentado. Una vez obtenido este modelo, es necesaria la obtención de los estados de equilibrio del sistema, para, según los casos, estudiar la estabilidad en torno a dichos estados mediante resultados derivados del teorema de Lyapunov, como el teorema de Krasovskii [13].

Para el cálculo de los estados de equilibrio utilizamos en este trabajo dos métodos clásicos de mínimos cuadrados [8, 14, 15, 16] derivados del método de

Newton, como son el método *Line-Search* y el método de Región de Confianza. Ambos métodos precisan de la matriz Jacobiana del sistema o en su defecto de una aproximación de la misma, por lo que se hace aconsejable el cálculo explícito de la matriz Jacobiana del modelo controlado. Con objeto de estudiar el efecto que produce en la eficiencia de los algoritmos de análisis numérico mencionados, el disponer o no de la matriz Jacobiana del sistema a analizar, se comparan en dos ejemplos sus funcionamientos en función del uso o bien de la matriz Jacobiana o bien de una aproximación de la misma.

El trabajo está organizado en las siguientes secciones: la primera, *Modelo borroso no lineal multivariable de un sistema de control*, describe el desarrollo matemático hasta su expresión final de un modelo de planta identificada mediante técnicas de lógica borrosa, así como la aplicación a la misma de un controlador borroso. Conjuntamente se presenta una simplificación de las ecuaciones basada en la ampliación del vector de estados, que proporciona mayor claridad a las expresiones finales obtenidas y facilita la programación de éstas. A continuación se trata el problema de la *Obtención de los estados de equilibrio de un sistema borroso no lineal*, comentando los distintos algoritmos que se emplearán más tarde para la resolución de los *Ejemplos* propuestos. Finalmente, en el apartado de *Conclusiones*, se analizan y comparan los resultados obtenidos con los distintos métodos de análisis numéricos empleados.

## 2 MODELO BORROSO MULTIVARIABLE NO LINEAL DE UN SISTEMA DE CONTROL

Considérese una planta multidimensional descrita por:

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m) \\ \dot{x}_2 &= f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m)\end{aligned}\quad (1)$$

donde el controlador multivariable se define como:

$$\begin{aligned}u_1 &= \phi_1(x_1, x_2, \dots, x_n); \\ u_2 &= \phi_2(x_1, x_2, \dots, x_n); \\ &\vdots \\ u_m &= \phi_m(x_1, x_2, \dots, x_n)\end{aligned}\quad (2)$$

Las expresiones (1) y (2) representan un modelo totalmente general tanto para la planta como para el

controlador, sin ningún tipo de restricciones, por lo que pueden ser no lineales. El sistema puede ser escrito de forma simplificada como en (3).

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (3)$$

siendo  $\dot{\mathbf{x}} = (\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n)^T$ ,  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  el vector de estado y  $\mathbf{u} = (u_1, u_2, \dots, u_m)^T$  el vector de control.

Un modelo borroso equivalente múltiple-entrada-múltiple-salida (MIMO) del proceso puede ser representado mediante el siguiente conjunto de reglas [5, 6, 18, 21]:

$$\begin{aligned}R^{(l)}: & \text{SI } x_1 \text{ es } A_{1i}^l \text{ y } x_2 \text{ es } A_{2i}^l, \dots, \text{ y } x_n \text{ es } A_{ni}^l \text{ y} \\ & u_1 \text{ es } B_{1i}^l \text{ y } u_2 \text{ es } B_{2i}^l, \dots, \text{ y } u_m \text{ es } B_{mi}^l \\ & \text{ENTONCES } \dot{x}_i \text{ es } g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l)\end{aligned}\quad (4)$$

Donde  $l = 1 \dots M$  es el número de reglas del modelo borroso de la planta;  $A_{ki}^l$  el conjunto borroso definido en el universo de discurso de la variable de estado  $x_k$ ,  $k = 1 \dots n$ , y la  $i$ -ésima ecuación diferencial del proceso; y  $B_{ji}^l$  el conjunto borroso definido en el universo de discurso de la  $j$ -ésima señal de control  $u_j$ ,  $j = 1 \dots m$ , y la  $i$ -ésima ecuación diferencial del proceso.

Considerando en la expresión (4) que el consecuente es TSK con término afín:

$$\begin{aligned}g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l) &= a_{0i}^l + a_{1i}^l x_1 + \dots + a_{ni}^l x_n + \\ &+ b_{1i}^l u_1 + b_{2i}^l u_2 + \dots + b_{mi}^l u_m\end{aligned}\quad (5)$$

Donde  $a_{ki}^l$  y  $b_{ji}^l$  representan, respectivamente, el coeficiente constante para la variable de estado  $x_k$  y para la  $j$ -ésima señal de control de la ecuación diferencial  $i$  que representa el proceso. El vector  $\boldsymbol{\theta}_i^l$  representa el conjunto de parámetros adaptables:

$$\boldsymbol{\theta}_i^l = (a_{0i}^l, a_{1i}^l, \dots, a_{ni}^l; b_{1i}^l, b_{2i}^l, \dots, b_{mi}^l)^T \quad (6)$$

Considerando un sistema borroso TSK con desborrosificador centro-promedio y método de inferencia producto, la expresión (3) puede ser escrita como [22]:

$$\dot{x}_i = \frac{\sum_{l=1}^M w_i^l g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l)}{\sum_{l=1}^M w_i^l} \quad (7)$$

Donde  $w_i^l$  es el grado de cumplimiento de la regla  $l$ :

$$w_i^l = \prod_{k=1}^n \mu_{F_k}^l(x_k, \theta_i^l) \prod_{j=1}^m \mu_{F_j}^l(u_j, \theta_i^l), \quad (8)$$

siendo  $\mu_{F_k}^l(x_k, \theta_i^l)$  y  $\mu_{F_j}^l(u_j, \theta_i^l)$  las funciones de pertenencia definidas en sus correspondientes universos de discurso  $X$  y  $U$  para el conjunto borroso  $F \in (X \cup U)$  representado por un conjunto de pares ordenados  $\left\{ (x_k, \mu_{F_k}^l(x_k, \theta_i^l)) : x_k \in X \right\}$  y  $\left\{ (u_j, \mu_{F_j}^l(u_j, \theta_i^l)) : u_j \in U \right\}$ .

Si el consecuente de la regla tiene la forma definida en (5), entonces se puede escribir (7) como en (9) o (10):

$$\dot{x}_i = \frac{1}{\sum_{l=1}^M w_i^l} \left[ \sum_{l=1}^M w_i^l a_{0i}^l + \cdots + \sum_{l=1}^M w_i^l a_{ni}^l x_n + \sum_{l=1}^M w_i^l b_{1i}^l u_1 + \cdots + \sum_{l=1}^M w_i^l b_{mi}^l u_m \right] \quad (9)$$

$$\begin{aligned} \dot{x}_i &= a_{0i}^l + a_{1i}^l x_1 + a_{2i}^l x_2 + \cdots + a_{ni}^l x_n + \\ &+ b_{1i}^l u_1 + b_{2i}^l u_2 + \cdots + b_{mi}^l u_m = \\ &= a_{0i}^l + \sum_{k=1}^n a_{ki}^l x_k + \sum_{j=1}^m b_{ji}^l u_j \end{aligned} \quad (10)$$

Donde  $a_{0i}^l$ ,  $a_{ki}^l$  y  $b_{ji}^l$  son coeficientes variables [23] definidos por las expresiones siguientes:

$$a_{0i}^l = \frac{\sum_{l=1}^M w_i^l a_{0i}^l}{\sum_{l=1}^M w_i^l}, \quad a_{ki}^l = \frac{\sum_{l=1}^M w_i^l a_{ki}^l}{\sum_{l=1}^M w_i^l}; \quad k=1, \dots, n \quad (11)$$

$$b_{ji}^l = \frac{\sum_{l=1}^M w_i^l b_{ji}^l}{\sum_{l=1}^M w_i^l}; \quad j=1, \dots, m \quad (12)$$

Con objeto de simplificar la ecuación (10), el vector de estados se extiende en una coordenada, de manera que éste adopta la forma:

$$\tilde{\mathbf{x}} = \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} = (x_0, x_1, \dots, x_n)^T; \quad x_0 = 1 \quad (13)$$

Al sustituir (13) en (10), se obtiene la siguiente expresión simplificada:

$$\dot{x}_i = \sum_{k=0}^n a_{ki}^l \tilde{x}_k + \sum_{j=1}^m b_{ji}^l u_j \quad (14)$$

Obtenido el modelo equivalente borroso de la planta, se procederá de forma similar con el controlador. Para ello se considera que su diseño es completamente independiente del proceso de identificación del modelo de la planta, de forma que la partición del universo de discurso de los estados para la planta y controlador no tienen porqué coincidir. Por lo mismo, sus modelos borrosos no tienen porqué tener el mismo número de reglas. En consecuencia, el controlador borroso puede ser representado por el siguiente conjunto de reglas:

$$\begin{aligned} R^{(r)}: & \text{SI } x_1 \text{ es } C_{1j}^r \text{ y } x_2 \text{ es } C_{2j}^r, \dots, \text{ y } x_n \text{ es } C_{nj}^r \\ & \text{ENTONCES } u_j = c_{0j}^r + c_{1j}^r x_1 + \cdots + c_{nj}^r x_n \end{aligned} \quad (15)$$

Donde  $C_{kj}^r$  representa el conjunto borroso de las  $k = 1 \dots n$  variables de estado, con  $j = 1 \dots m$  acciones de control y  $r = 1 \dots N$  reglas.

Procediendo de forma similar a la mostrada en (8), (9) y (10), las señales de control pueden expresarse por:

$$u_j = \frac{1}{\sum_{r=1}^N \omega_j^r} \left[ \sum_{r=1}^N \omega_j^r c_{0j}^r + \cdots + \sum_{r=1}^N \omega_j^r c_{nj}^r x_n \right] \quad (16)$$

o de forma más simplificada:

$$u_j = c_{0j}^l + c_{1j}^l x_1 + c_{2j}^l x_2 + \cdots + c_{nj}^l x_n = c_{0j}^l + \sum_{k=1}^n c_{kj}^l x_k \quad (17)$$

donde los coeficientes variables vienen determinados por:

$$c_{0j}^l = \frac{\sum_{r=1}^N \omega_j^r c_{0j}^r}{\sum_{r=1}^N \omega_j^r}; \quad c_{kj}^l = \frac{\sum_{r=1}^N \omega_j^r c_{kj}^r}{\sum_{r=1}^N \omega_j^r}; \quad k=1, \dots, n \quad (18)$$

y el grado de activación de la regla  $r$  del controlador, por:

$$\omega_j^r = \prod_{k=1}^n \mu_{F_k}^r(x_k, \varphi_j^r). \quad (19)$$

$\varphi_j^r$  es el vector de parámetros adaptables del controlador:

$$\varphi_j^r = (c_{0j}^r, c_{1j}^r, \dots, c_{nj}^r) \quad (20)$$

Sustituyendo la expresión (17) en (10), se obtiene la ecuación del sistema no lineal en lazo cerrado:

$$\dot{x}_i = a_{0i}^t + \sum_{j=1}^m b_{ji}^t c_{0j}^t + \sum_{k=1}^n \left( a_{ki}^t + \sum_{j=1}^m b_{ji}^t c_{kj}^t \right) x_k \quad (21)$$

$i, k = 1, \dots, n$  y  $j = 1, \dots, m$

Aplicando la extensión del vector de estados a (21), se obtiene la siguiente expresión simplificada:

$$\dot{\tilde{x}}_i = \sum_{k=0}^n \left( a_{ki}^t + \sum_{j=1}^m b_{ji}^t c_{kj}^t \right) \tilde{x}_k \quad (22)$$

$k = 0, 1, \dots, n, i = 1, \dots, n$  y  $j = 1, \dots, m$

### 3 OBTENCIÓN DE LOS ESTADOS DE EQUILIBRIO DE UN SISTEMA BORROSO NO LINEAL

El problema de hallar los estados de equilibrio de un sistema, puede ser uno de los primeros pasos que deba resolver el ingeniero de control previo a la síntesis de un controlador. La búsqueda de los estados de equilibrio de un sistema puede tener varios objetivos. Como análisis previo al diseño del sistema de control, la obtención de los estados de equilibrio proporciona una valiosa información sobre el comportamiento de éste, 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, será necesaria la obtención de los estados de equilibrio del sistema realimentado, para así poder aplicar análisis de estabilidad a dichos estados.

Para obtener los estados de equilibrio de un sistema borroso de tiempo continuo hay que resolver el sistema de ecuaciones planteado en (14) y desglosado aquí:

$$\begin{aligned} \sum_{k=0}^n a_{k1}^t \tilde{x}_k + \sum_{j=1}^m b_{j1}^t u_j &= 0, \\ \sum_{k=0}^n a_{k2}^t \tilde{x}_k + \sum_{j=1}^m b_{j2}^t u_j &= 0, \\ &\vdots \\ \sum_{k=0}^n a_{kn}^t \tilde{x}_k + \sum_{j=1}^m b_{jn}^t u_j &= 0. \end{aligned} \quad (23)$$

Si se realimenta el sistema con un controlador borroso formado por un conjunto de reglas de la forma dada en (15), la obtención de los estados de equilibrio pasa por resolver el sistema de ecuaciones planteado en (22) y desglosado aquí:

$$\begin{aligned} \sum_{k=0}^n \left( a_{k1}^t + \sum_{j=1}^m b_{j1}^t c_{kj}^t \right) \tilde{x}_k &= 0, \\ \sum_{k=0}^n \left( a_{k2}^t + \sum_{j=1}^m b_{j2}^t c_{kj}^t \right) \tilde{x}_k &= 0, \\ &\vdots \\ \sum_{k=0}^n \left( a_{kn}^t + \sum_{j=1}^m b_{jn}^t c_{kj}^t \right) \tilde{x}_k &= 0. \end{aligned} \quad (24)$$

### 3.1 MÉTODOS NUMÉRICOS PARA LA RESOLUCIÓN DE SISTEMAS DE ECUACIONES NO LINEALES

Los métodos de Newton proporcionan una importante clase de algoritmos de optimización local que requieren el cálculo del vector gradiente,  $\mathbf{G}(x_1, \dots, x_n)$ , y la matriz Hessiana,  $\mathbf{H}(x_1, \dots, x_n)$ . En la forma básica del método de Newton, la siguiente iteración se obtiene desde la minimización de  $\mathbf{f}_x^i$ , siendo

$$\mathbf{f}_x^i(\mathbf{s}) = \mathbf{f}(\mathbf{x}^i) + \mathbf{G}(\mathbf{x}^i)^T \mathbf{s} + 1/2 \mathbf{s}^T \mathbf{H}(\mathbf{x}^i) \mathbf{s} \quad (25)$$

un modelo cuadrático de la función objetivo en torno al valor de  $\mathbf{x}$  en la  $i$ -ésima iteración, de forma que  $\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{s}^i$ . Cuando la matriz Hessiana,  $\mathbf{H}(\mathbf{x}^i)$ , es definida positiva, el modelo cuadrático tiene una única solución que puede obtenerse mediante la resolución del sistema simétrico lineal

$$\mathbf{H}(\mathbf{x}^i) \mathbf{s}^i = -\mathbf{G}(\mathbf{x}^i). \quad (26)$$

La convergencia del método de Newton está garantizada siempre que el punto inicial esté lo suficientemente cerca de algún mínimo local  $\mathbf{x}^*$  en el cual la matriz Hessiana sea definida positiva. Además la razón de convergencia es cuadrática, esto es

$$\|\mathbf{x}^{i+1} - \mathbf{x}^*\| \leq \beta \|\mathbf{x}^i - \mathbf{x}^*\|^2, \quad (27)$$

para alguna constante positiva  $\beta$ .

Si el punto de partida no está lo suficientemente cerca de un mínimo local, entonces la convergencia puede conseguirse mediante los métodos *Line-Search* y de Región de Confianza. El método *Line-Search* modifica la dirección de búsqueda para conseguir una nueva dirección de descenso para  $\mathbf{f}(\mathbf{x})$ . El método de Región de Confianza utiliza el modelo cuadrático inicial de la función original pero restringiendo la

búsqueda de la siguiente iteración a un entorno de la iteración actual donde el modelo cuadrático es lo suficientemente fiable. Ambos métodos son eficientes sólo si el número  $n$  de variables no es demasiado grande ya que el costo por iteración es del orden de  $n^3$ .

Hasta el momento hemos dado por sentado que disponemos tanto de los vectores gradientes y matriz Jacobiana como de la matriz Hessiana, aunque por lo general este no suele ser el caso. Los métodos anteriores, en caso de no disponer de los gradientes y de la matriz Jacobiana, suelen hacer uso de aproximaciones en diferencias para calcular las derivadas, con el consiguiente error de cálculo y retraso en la convergencia. Además, como veremos en lo que sigue, estos elementos son utilizados a su vez para el cálculo de aproximaciones de la matriz Hessiana. Por lo que los errores en las aproximaciones iniciales de la matriz Jacobiana se incrementan ineludiblemente en los cálculos de la matriz Hessiana.

Existen dos tipos de métodos que utilizan aproximaciones de la matriz Hessiana. El primero de ellos utiliza aproximaciones en diferencias para calcular las columnas de la matriz Hessiana mediante la diferencia de dos evaluaciones del vector gradiente en puntos cercanos. El segundo, los métodos de Quasi-Newton [20], va construyendo nuevas aproximaciones de la matriz Hessiana en función del valor del gradiente y de la matriz Hessiana de la iteración anterior.

Finalmente existen otras aproximaciones que no están relacionadas, al menos tan directamente, con el método de Newton. Dos de los más interesantes son el método de Gradiente-Conjugado que combina la dirección de máximo descenso con otra dirección no previamente explorada y el método del Simplex de Nelder y Mead [17] que no requiere evaluaciones ni del gradiente ni de la matriz Hessiana de la función. Puede ser útil en casos muy extremos en los que  $\mathbf{f}(\mathbf{x})$  es muy no lineal, la función tiene muchas discontinuidades y/o cuando las derivadas no existen o no pueden calcularse.

En lo que sigue utilizaremos dos métodos de mínimos cuadrados para aproximar el estado de equilibrio, minimizando el valor de la derivada: el método de Gauss-Newton y el método de Levenberg-Marquardt [16], donde la función a minimizar  $\mathbf{f}(\mathbf{x})$  se obtiene según la siguiente expresión cuadrática:

$$\mathbf{f}(\mathbf{x}) = \frac{1}{2} \|\mathbf{F}(\mathbf{x})\|^2 = \frac{1}{2} \sum_{k=1}^n (F_k(\mathbf{x}))^2 \quad (28)$$

siendo  $F_k(\mathbf{x})$  las ecuaciones dadas en (14) o (22), según el problema a resolver.

Desde un punto de vista algorítmico, la característica que distingue los problemas de mínimos cuadrados de los problemas de optimización no lineales es la estructura de la matriz Hessiana.

Si denotamos por  $\mathbf{J}(\mathbf{x})$  la matriz Jacobiana de  $\mathbf{F}(\mathbf{x})$ , el vector gradiente de  $\mathbf{f}(\mathbf{x})$  por  $\mathbf{G}(\mathbf{x})$ , la matriz Hessiana de  $\mathbf{f}(\mathbf{x})$  por  $\mathbf{H}(\mathbf{x})$  y la matriz Hessiana de cada  $F_i(\mathbf{x})$  por  $H_i(\mathbf{x})$  tenemos que

$$\begin{aligned} \mathbf{G}(\mathbf{x}) &= 2\mathbf{J}(\mathbf{x})^T \mathbf{F}(\mathbf{x}) \\ \mathbf{H}(\mathbf{x}) &= 2\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + 2\mathbf{Q}(\mathbf{x}) \end{aligned} \quad (29)$$

Donde la matriz

$$\mathbf{Q}(\mathbf{x}) = \sum_{i=1}^n F_i(\mathbf{x}) H_i(\mathbf{x}) \quad (30)$$

tiende a cero cuando  $\|\mathbf{F}(\mathbf{x})\|$  tiende a cero si  $\mathbf{x}^i$  se aproxima a la solución.

En el método de Gauss-Newton, la dirección de búsqueda de la nueva iteración,  $\mathbf{d}^i$ , es la solución del problema de mínimos cuadrados

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{J}(\mathbf{x}^i) \mathbf{d}^i + \mathbf{F}(\mathbf{x}^i)\|^2 \quad (31)$$

La dirección de búsqueda  $\mathbf{d}^i$  puede a su vez utilizarse en una estrategia de minimización para garantizar que la función  $\mathbf{f}(\mathbf{x})$  decrece en cada iteración.

El método de Gauss-Newton puede no ser efectivo cuando  $\|\mathbf{F}(\mathbf{x})\|$  no es pequeño al acercarse  $\mathbf{x}^i$  a la solución y ser el término de segundo orden  $\mathbf{Q}(\mathbf{x})$  significativo.

Por otro lado el método de Levenberg-Marquardt salva este problema utilizando una dirección de búsqueda que es la solución del siguiente conjunto de ecuaciones lineales

$$(\mathbf{J}(\mathbf{x}^i)^T \mathbf{J}(\mathbf{x}^i) + \lambda^i \mathbf{I}) \mathbf{d}^i = -\mathbf{J}(\mathbf{x}^i)^T \mathbf{F}(\mathbf{x}^i) \quad (32)$$

Donde  $\lambda^i$  controla la dirección y magnitud de  $\mathbf{d}^i$ ; cuando  $\lambda^i$  es cero,  $\mathbf{d}^i$  es la misma que el método de Gauss-Newton y según  $\lambda^i$  tiende a infinito  $\mathbf{d}^i$  tiende a la dirección de máximo descenso. De modo que el método de Levenberg-Marquardt utiliza una dirección de búsqueda que está entre la dirección del

método de Gauss-Newton y la dirección de máximo descenso.

Como se ha visto hasta ahora, los métodos numéricos para la resolución de sistemas de ecuaciones no lineales, y especialmente, los métodos de mínimos cuadrados, necesitan disponer del cálculo de la matriz Jacobiana del sistema. En la sección siguiente, se desarrollará, a partir de trabajos previos de los autores, el cálculo de la matriz Jacobiana del sistema de control borroso de forma explícita, para utilizarlo posteriormente en el estudio comparativo del final del presente trabajo.

### 3.2 CÁLCULO DE LA MATRIZ JACOBIANA DEL SISTEMA DE CONTROL BORROSO

Dada la mejoría que puede suponer obtener la matriz Jacobiana de las expresiones (14) y (22) para la resolución de los estados de equilibrio de la planta o del sistema de control realimentado, se hace necesario obtener la derivada de estas expresiones con respecto a cada una de las variables de estado.

El cálculo de la matriz Jacobiana del sistema de control borroso dado en (22) fue resuelto en [1, 2]. Para obtener la matriz Jacobiana de la planta expresada en la ecuación (14), se considerará que las señales de control son nulas en el estado de equilibrio, de esta forma se han de calcular las derivadas con respecto a cada una de las variables de estado de la siguiente expresión:

$$\dot{x}_i = \sum_{k=0}^n a_{ki}^l \tilde{x}_k \quad (33)$$

La derivada de la ecuación (33) está resuelta en [1] como paso previo a la obtención de la matriz Jacobiana del sistema de control borroso, salvo que en este caso se deben reformular las derivadas del grado de activación de las reglas de la planta.

Así se obtiene que:

$$\frac{\partial \left( \sum_{k=0}^n a_{ki}^l \tilde{x}_k \right)}{\partial x_q} = a_{qi}^l + \sum_{k=0}^n \left( \frac{\sum_{l,p=1}^M \left( \frac{\partial w_i^l}{\partial x_q} w_i^p (a_{ki}^l - a_{ki}^p) \right)}{\left( \sum_{l=1}^M w_i^l \right)^2} \right) \tilde{x}_k \quad (34)$$

siendo en este caso

$$\frac{\partial w_i^l}{\partial x_q} = \frac{\partial \mu_{F_q}^l(x_q, \theta_i^l)}{\partial x_q} \prod_{\substack{k=1 \\ k \neq q}}^n \mu_{F_k}^l(x_k, \theta_i^l) \prod_{j=1}^m \mu_{F_j}^l(u_j, \theta_i^l), \quad (35)$$

y

$$\frac{\partial w_i^l}{\partial u_t} = \frac{\partial \mu_{F_t}^l(u_t, \theta_i^l)}{\partial u_t} \prod_{k=1}^n \mu_{F_k}^l(x_k, \theta_i^l) \prod_{\substack{j=1 \\ j \neq t}}^m \mu_{F_j}^l(u_j, \theta_i^l). \quad (36)$$

## 4 EJEMPLOS

Para poder comparar los resultados obtenidos con la inclusión del cálculo exacto de la matriz Jacobiana en los algoritmos numéricos, se compararán las soluciones obtenidas, el número de llamadas que requiere cada algoritmo a la función objetivo o a minimizar, dada por (28), y el tiempo empleado para la resolución del cálculo. La importancia de las soluciones obtenidas es obvia, ya que éstas se corresponden con los estados de equilibrio que se desean calcular. El número de llamadas a la función objetivo da una idea de la eficiencia del algoritmo (en los casos en los que se aplica la matriz Jacobiana, se indicará el número de veces que se requiere su cálculo junto con el número de llamadas a la función objetivo, ya que su coste computacional es superior al de la aproximación utilizada cuando ésta no se incluye). Por último, dado que todos los casos se van a resolver en el mismo computador, dedicado en exclusiva y con el mismo software, MATLAB<sup>®</sup>, se puede emplear el tiempo de cómputo como una medida exacta de la eficiencia de cada uno de los algoritmos. La tolerancia establecida para el resultado es igual en todos los casos a  $10^{-12}$ .

### 4.1 EJEMPLO 1

Dado un modelo borroso obtenido mediante datos de entrada/salida de la planta siguiente:

$$\begin{aligned} \dot{x}_1 &= x_1 + 2x_2 + u_1 \\ \dot{x}_2 &= x_1 - 2x_2^3 + u_2 \end{aligned} \quad (37)$$

Se podría realizar un primer análisis de la dinámica de la misma mediante la obtención de sus estados de equilibrio. Para ello se emplearán los métodos numéricos presentados en el apartado 3.1 aplicados a la ecuación (14).

Los resultados medios obtenidos con los distintos métodos se reflejan en la tabla 1. Para obtener estos valores, se han ejecutado 5 veces los algoritmos con las mismas condiciones iniciales, obtenidas en cada caso de forma aleatoria dentro del rango de definición de los modelos. Teniendo en cuenta la

tolerancia establecida, el estado de equilibrio alcanzado siempre ha sido el mismo:  $x_1 = 0.1704$  y  $x_2 = -0.0852$ .

Casos	Llamadas a la Función Objetivo	Tiempo (ms)
Gauss-Newton	31.2	340.8
GN con Jacobiano	10.4 + 10.4	134.2
Levenberg-Marquardt	31.2	156
LM con Jacobiano	10.4 + 10.4	132.4

Tabla 1: Resultados medios obtenidos para la búsqueda de los estados de equilibrio de la planta.

Si se diseña un controlador borroso para esta planta en la forma:

SI  $x_1$  es Gauss(-10,12) y  $x_2$  es Gauss(-10,12)  
 ENTONCES  $u_1 = -7 * x_1$   
 SI  $x_1$  es Gauss(-10,12) y  $x_2$  es Gauss(10,12)  
 ENTONCES  $u_1 = -7 * x_1$   
 SI  $x_1$  es Gauss(10,12) y  $x_2$  es Gauss(-10,12)  
 ENTONCES  $u_1 = -7 * x_1$   
 SI  $x_1$  es Gauss(10,12) y  $x_2$  es Gauss(10,12)  
 ENTONCES  $u_1 = -7 * x_1$

SI  $x_1$  es Gauss(-10,12) y  $x_2$  es Gauss(-10,12)  
 ENTONCES  $u_2 = x_1 - 6 * x_2$   
 SI  $x_1$  es Gauss(-10,12) y  $x_2$  es Gauss(10,12)  
 ENTONCES  $u_2 = x_1 - 6 * x_2$   
 SI  $x_1$  es Gauss(10,12) y  $x_2$  es Gauss(-10,12)  
 ENTONCES  $u_2 = x_1 - 6 * x_2$   
 SI  $x_1$  es Gauss(10,12) y  $x_2$  es Gauss(10,12)  
 ENTONCES  $u_2 = x_1 - 6 * x_2$

donde los parámetros de la función gaussiana son el centro y el ancho respectivamente, se pueden obtener los estados de equilibrio del sistema realimentado mediante la utilización del equivalente dado en (22).

Resolviendo los estados de equilibrio del sistema borroso realimentado, se obtienen los resultados reflejados en la tabla 2. Al igual que en el caso anterior, se han ejecutado 5 veces los algoritmos con condiciones iniciales, obteniendo éstas de forma aleatoria. En todos los casos se ha obtenido el estado de equilibrio  $x_1 = -0.0111$  y  $x_2 = -0.3333$ .

Las tablas 1 y 2 muestran de forma clara que la reducción del tiempo de cómputo es sustancial, independientemente del algoritmo empleado y pese al coste computacional que supone la obtención de la matriz Jacobiana. En el primer caso se ha conseguido una reducción del 60.62% del tiempo para el algoritmo de Gauss-Newton y un 15.13% para el

algoritmo de Levenberg-Marquardt; mientras que en el segundo caso, se ha obtenido una reducción media del 37.8% como consecuencia de la inclusión explícita de la matriz Jacobiana en los algoritmos de minimización.

Casos	Llamadas a la Función Objetivo	Tiempo (ms)
Gauss-Newton	42.8	380.8
GN con Jacobiano	15.2 + 15.2	238.2
Levenberg-Marquardt	42.8	366.2
LM con Jacobiano	15.2 + 15.2	226

Tabla 2: Resultados medios obtenidos para la búsqueda de los estados de equilibrio del sistema de control realimentado.

## 4.2 EJEMPLO 2

Sea el péndulo de la figura 1, cuya dinámica ha sido modelada mediante un sistema borroso partiendo de datos de entrada/salida.

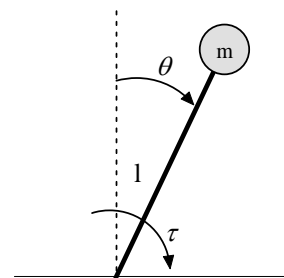


Figura 1. Péndulo Invertido sobre base fija.

Al igual que en el ejemplo anterior, se estudiará el sistema en dos supuestos: en el primero de ellos se buscarán los estados de equilibrio del sistema autónomo formado por la planta libre de acciones de control; en el segundo caso, se realimentará el sistema con un controlador borroso y se buscarán los estados de equilibrio del conjunto.

Tras ejecutar los algoritmos de minimización 5 veces seleccionando las condiciones iniciales en cada iteración de forma aleatoria, se han obtenido los resultados cuya media aritmética se presenta en la tabla 3.

Se han obtenido dos estados de equilibrio, dependiendo de las condiciones iniciales introducidas al del algoritmo, que se corresponden con las posiciones del ángulo  $\theta$  en 0 y  $\pi$ , y su derivada en 0.

Casos	Llamadas a la Función Objetivo	Tiempo (ms)
<b>Gauss-Newton</b>	14.4	244.4
<b>GN con Jacobiano</b>	4.8 + 4.8	62
<b>Levenberg-Marquardt</b>	14.4	66
<b>LM con Jacobiano</b>	4.8 + 4.8	60

Tabla 3: Resultados medios obtenidos para la búsqueda de los estados de equilibrio de la planta.

La reducción promedio de tiempo obtenida al emplear el cálculo explícito de la matriz Jacobiana en este ejemplo, ha sido del 74.63% para el algoritmo de Gauss-Newton y del 9.1% para el algoritmo de Levenberg-Marquardt.

A continuación se procederá con el estudio del sistema realimentado. Para ello se empleará el siguiente controlador diseñado según el conocimiento de un experto:

SI X1 es Trap(-1.422,-1.222,-0.7236,-0.524) y X2 es Trap(-172,-108,-92,-28)  
 ENTONCES Tau = -1300\*X1-90\*X2  
 SI X1 es Trap(-1.422,-1.222,-0.7236,-0.524) y X2 es Trap(-72,-8,8,72)  
 ENTONCES Tau = -1300\*X1-90\*X2  
 SI X1 es Trap(-1.422,-1.222,-0.7236,-0.524) y X2 es Trap(28,92,108,172)  
 ENTONCES Tau = -1300\*X1-90\*X2  
 SI X1 es Trap(-0.7236,-0.5236,0.5236,0.7236) y X2 es Trap(-172,-108,-92,-28)  
 ENTONCES Tau = 28.714-1000\*X1-60\*X2  
 SI X1 es Trap(-0.7235,-0.5236,0.5236,0.7236) y X2 es Trap(-72,-8,8,72)  
 ENTONCES Tau = 28.714-1000\*X1-60\*X2  
 SI X1 es Trap(-0.7236,-0.5236,0.5236,0.7236) y X2 es Trap(28,92,108,172)  
 ENTONCES Tau = 28.714-1000\*X1-60\*X2  
 SI X1 es Trap(0.524,0.7236,1.222,1.422) y X2 es Trap(-172,-108,-92,-28)  
 ENTONCES Tau = -1300\*X1-90\*X2  
 SI X1 es Trap(0.524,0.7236,1.222,1.422) y X2 es Trap(-72,-8,8,72)  
 ENTONCES Tau = -1300\*X1-90\*X2  
 SI X1 es Trap(0.524,0.7236,1.222,1.422) y X2 es Trap(28,92,108,172)  
 ENTONCES Tau = -1300\*X1-90\*X2  
 SI X1 es Trap(-6.4,-6.4,-1.422,-1.222) y X2 es Trap(-172,-108,-92,-28)  
 ENTONCES Tau = -1700\*X1-120\*X2  
 SI X1 es Trap(-6.4,-6.4,-1.422,-1.222) y X2 es Trap(-72,-8,8,72)  
 ENTONCES Tau = -1700\*X1-120\*X2

SI X1 es Trap(-6.4,-6.4,-1.422,-1.222) y X2 es Trap(28,92,108,172)  
 ENTONCES Tau = -1700\*X1-120\*X2  
 SI X1 es Trap(1.222,1.422,6.4,6.4) y X2 es Trap(-172,-108,-92,-28)  
 ENTONCES Tau = -1700\*X1-120\*X2  
 SI X1 es Trap(1.222,1.422,6.4,6.4) y X2 es Trap(-72,-8,8,72)  
 ENTONCES Tau = -1700\*X1-120\*X2  
 SI X1 es Trap(1.222,1.422,6.4,6.4) y X2 es Trap(28,92,108,172)  
 ENTONCES Tau = -1700\*X1-120\*X2

Donde *Trap* es la función de pertenencia trapezoidal definida por sus cuatro puntos característicos.

Los resultados promediados de las 5 ejecuciones se muestran en la tabla 4, donde se puede apreciar que la reducción de tiempo para el algoritmo de Gauss-Newton ha sido del 70.71%, y para el algoritmo de Levenberg-Marquardt, del 16.9%.

Casos	Llamadas a la Función Objetivo	Tiempo (ms)
<b>Gauss-Newton</b>	16.8	232.2
<b>GN con Jacobiano</b>	5.6 + 5.6	68
<b>Levenberg-Marquardt</b>	16.8	72.2
<b>LM con Jacobiano</b>	5.6 + 5.6	60

Tabla 4: Resultados medios obtenidos para la búsqueda de los estados de equilibrio del sistema de control realimentado.

## 5 CONCLUSIONES

En este trabajo se ha presentado un modelo matemático equivalente de un sistema de control borroso multivariable no lineal. A continuación, se ha abordado la problemática de buscar sus estados de equilibrio, empleándose para ello métodos clásicos de análisis numérico de sistemas no lineales con la mejora que supone disponer del cálculo explícito de la matriz Jacobiana. Se ha presentado la expresión de dicha matriz para un sistema de control borroso en lazo cerrado, y ha sido utilizada en los ejemplos para demostrar la mejora sustancial que aporta cuando se emplea de forma conjunta con los algoritmos basados en los métodos clásicos.

Es de notar que los algoritmos de cálculo de la matriz Jacobiana desarrollados por los autores de este trabajo (el software que los implementa contiene más de 3.500 líneas de código C++), pueden ser aplicados a cualquier sistema de control borroso sin restricciones (dados como en (14) o (22)) ni en el orden del vector de entrada ni tampoco en el de



control. También, es destacable que el algoritmo es independiente del tipo de función de pertenencia utilizada. De hecho, el primer ejemplo se ha implementado con funciones de pertenencia Gaussianas y el segundo con trapezoidales.

Este trabajo es un paso más dentro de un objetivo mucho más amplio que constituye una de las líneas de investigación de sus autores: la síntesis de sistemas de control borroso estables por diseño.

## Referencias

- [1] J. M. Andújar, A. J. Barragán, Matriz Jacobiana de un Sistema de Control Borroso en Lazo Cerrado. Aplicación a la Síntesis de Sistemas de Control Borroso. Proc. XXIV Jornadas de Automática, León, España, (2003).
- [2] J. M. Andújar, A. J. Barragán, Algoritmo para el Cálculo de la Matriz Jacobiana de un Sistema de Control Borroso en Lazo Cerrado. Proc. ESTYLF'04, Jaén, España, (2004).
- [3] J.M. Andújar, J. M Bravo, Multivariable fuzzy control applied to the physical-chemical treatment facility of a cellulose factory, disponible en línea desde el 22 de abril de 2004 en <http://dx.doi.org/10.1016/j.fss.2004.03.023> .
- [4] J. M. Andújar, J. M. Bravo and A. Peregrín, Stability analysis and synthesis of multivariable fuzzy systems using interval arithmetic, Fuzzy Sets and Systems, disponible en línea desde el 24 de febrero de 2004 en <http://dx.doi.org/10.1016/j.fss.2004.01.008> .
- [5] R. Babuška, Fuzzy modeling a control engineering perspective, Proc. FUZZ-IEEE/IFES'95, Yokohama, Japan, (1995) 1897-1902.
- [6] R. Babuška, H. B. Verbruggen, A new identification method for linguistic fuzzy models, Proc. FUZZ-IEEE/ IFES'95, Yokohama, Japan, (1995) 905-912.
- [7] Conn, N.R., N.I.M. Gould, and Ph.L. Toint, Trust-Region Methods, MPS/SIAM Series on Optimization, SIAM and MPS, 2000.
- [8] Dennis, J.E., Jr., "Nonlinear least-squares," State of the Art in Numerical Analysis ed. D. Jacobs, Academic Press, pp 269-312, 1977.
- [9] Dennis, J.E., Jr. and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall Series in Computational Mathematics, Prentice-Hall, 1983.
- [10] Fletcher, R., "Practical Methods of Optimization," Vol. 1, Unconstrained Optimization, and Vol. 2, Constrained Optimization, John Wiley and Sons, 1980.
- [11] Forsythe, G.F., M.A. Malcolm, and C.B. Moler, Computer Methods for Mathematical Computations, Prentice Hall, 1976.
- [12] Gill, P.E., W. Murray, and M.H. Wright, Numerical Linear Algebra and Optimization, Vol. 1, Addison Wesley, 1991.
- [13] Jean-Jacques E. Slotine and Weiping Li, Applied Nonlinear Control, Prentice Hall, 1991.
- [14] Levenberg, K., "A Method for the Solution of Certain Problems in Least Squares," Quart. Appl. Math. Vol. 2, pp 164-168, 1944.
- [15] Marquardt, D., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM J. Appl. Math. Vol. 11, pp 431-441, 1963.
- [16] Moré, J.J., "The Levenberg-Marquardt Algorithm: Implementation and Theory," Numerical Analysis, ed. G. A. Watson, Lecture Notes in Mathematics 630, Springer Verlag, pp 105-116, 1977.
- [17] Nelder, J.A. and R. Mead, "A Simplex Method for Function Minimization," Computer J., Vol.7, pp 308-313, 1965.
- [18] T. Nguyen, M. Sugeno, R. Tong, R. R. Yager, Theoretical aspects of fuzzy control, John Wiley Sons, (1995).
- [19] Nocedal, J. and S.J. Wright, Numerical Optimization, Springer Series in Operations Research, Springer Verlag, 1999.
- [20] Shanno, D.F., "Conditioning of Quasi-Newton Methods for Function Minimization," Mathematics of Computing, Vol. 24, pp 647-656, 1970.
- [21] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transaction on Systems, Man and Cybernetics 15 (1985) 116-132.

- [22] L. X. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall, New Jersey, USA, (1997).
- [23] L. Wong, F. Leung, P. Tam, Satability design of TS model based fuzzy systems, Proc. Six IEEE international conference on fuzzy systems, Barcelona, Spain, (1997) 83-86.