

Implementación de un Multimodelo Neuronal Jerárquico para Identificación y Control de Sistemas Mecánicos

Implementation of a Neural Hierarchical Multimodel for Identification and Control of Mechanical Systems

Ieroham Baruch y José Luis Olivares Guzmán

CINVESTAV-IPN

Departamento de Control Automático, Av. IPN 2508

Col. San Pedro Zacatenco, A.P. 14-740

C.P. 07360, México D.F., México

{baruch, jolivares}@ctrl.cinvestav.mx

Artículo recibido en octubre 17, 2003; aceptado en abril 01, 2005

Resumen

En este artículo se propone la implementación de un Multimodelo Neuronal Jerárquico (MNJ) basándose en la similitud con el modelo difuso de Takagi-Sugeno. El modelo MNJ tiene tres partes: 1) fuzificación; 2) inferencia en el nivel bajo usando Redes Neuronales Recurrentes, RNR; 3) defuzificación en el nivel jerárquico alto usando una RNR que es en realidad un filtro-sumador ponderado de las salidas de las RNR del nivel bajo. El aprendizaje y el funcionamiento de ambos niveles jerárquicos son independientes. El modelo MNJ es implementado como identificador y controlador (feedforward, y feedback) en dos esquemas de control directo adaptable. Ambos esquemas de control son aplicados con una planta mecánica con fricción y comparados con otros esquemas de control neuronal y difuso, mostrando mejores resultados. **Palabras Clave:** Control adaptable neuronal con modelo inverso, control neuronal directo adaptable, identificación de sistemas, Multimodelo Neuronal Jerárquico, Red Neuronal Recurrente Entrenable, sistema mecánico con fricción.

Abstract

The present paper proposed to implement a Neural Hierarchical Multi-Model (MNJ) based on the similarity with the fuzzy model of Takagi-Sugeno. The MNJ has three parts: 1) fuzzyfication; 2) inference engine in the lower hierarchical level, using Recurrent Neural Networks, RNR; 3) defuzzyfication in the upper hierarchical level, using one RNR doing a filtered weighted summation of the outputs of the lower level RNRs. The learning and functioning of both hierarchical levels is independent. The MNJ is implemented in two schemes of direct adaptive control as an identifier and as a feedforward/feedback controller, as well. Both control schemes are applied for control of a mechanical plant with friction and compared with other neural and fuzzy control schemes, exhibiting better results.

Keywords: Inverse model adaptive neural control, direct adaptive neural control, systems identification, Neural Hierarchical Multimodel, Recurrent Trainable Neural Network, mechanical system with friction.

1 Introducción

La fricción es una resistencia natural relativa al movimiento entre dos cuerpos en contacto, [1], [2], [3]. Las fuerzas de fricción no lineales son inevitables en sistemas de control de movimiento de alta ejecución. Para evitar esto, es apropiado el uso del enfoque multimodelo neuronal. Algunos trabajos en este campo, relacionados a la identificación de objetos dinámicos no lineales complejos, utilizando el multimodelo neuronal, donde el espacio de las variables de salida de la planta se divide en dos – positivo y negativo sin recubrimiento, han sido publicados por Baruch y otros, [4], [5], [6], [7], [8], [9]. En los artículos [10], [11], [12], se ha utilizado un enfoque neuronal para implementar un control indirecto y directo adaptable. En el artículo [13] se aplica el multimodelo neuronal para realizar las dichas esquemas de identificación y control de sistemas no lineales. La unión de las técnicas de lógica difusa y las redes neuronales poseen la novedosa idea de la transformación de los límites del diseño del control de la lógica difusa y la decisión de sistemas hacia el entrenamiento y aprendizaje de las redes neuronales, [14], [15], [16]. Esto es, las redes neuronales proveen una estructura de conexión (tolerancia al error y propiedades de representación distribuida) y las habilidades de aprendizaje de los sistemas de control de lógica difusa que proveen a las redes neuronales como un marco estructural con un alto nivel de raciocinio y pensamiento de la regla IF-THEN. En los artículos [10], [11], [12], se habló acerca de la Red Neuronal Recurrente Entrenable (RNRE), la cual tiene características importantes con respecto al diseño de leyes de control, debido a que estas RNRE permiten el ajuste de los pesos y umbrales, además de que la arquitectura de la red neuronal es completamente paralela y posee un número mínimo

de parámetros, de tal manera que nos permite un aprendizaje mas rápido. Con la restricción en los pesos de retroalimentación se asegura la estabilidad de la red, y con solo la información de entrada y salida, será suficiente para el entrenamiento de la red.

En el trabajo presente se propone un nuevo esquema de Multimodelo Neuronal Jerárquico (MNJ) para identificación y control, el cual se aplica a un sistema mecánico con fricción en la carga. Este sistema presenta un comportamiento altamente no lineal debido a la naturaleza de la fuerza de fricción que actúa sobre la carga. Este trabajo tiene como objetivo a desarrollar un multimodelo neuronal jerárquico usando tres intervalos con recubrimiento (positivo, cero y negativo), que otorga las ventajas propias de las redes neuronales artificiales en combinación con la lógica difusa, y dada su construcción con RNREs, otorga también las ventajas dichas de las RNREs.

2 Modelos de Lógica Difusa y RNRE

Uno de los modelos comunes en la lógica difusa es el modelo de Takagi –Sugeno, [17]. Dicho modelo es una mezcla entre el modelo lingüístico de Mamdani, [14], y el modelo de regresión matemática. La regla antecedente describe las regiones difusas en el espacio de entrada. La regla consecuente es una función matemática no difusa de la entrada. La estructura de las reglas está descrita por:

$$R_i: \text{ IF } x \text{ is } A_i \text{ THEN } y_i = f_i(x), \quad i=1,2,\dots, P \quad (1)$$

Donde f_i son las funciones matemáticas no difusas, las cuales conservan la misma estructura, variando los parámetros en cada una de las reglas. El uso más simple y práctico en la parametrización es la forma lineal, ilustrada en Figura 1, donde se definen las reglas de forma:

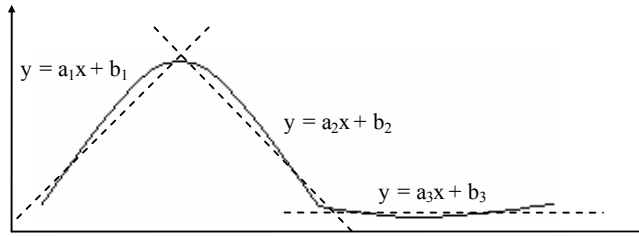


Fig.1. Aproximación lineal de Takagi-Sugeno

$$R_i: \text{ IF } x \text{ is } A_i \text{ THEN } y_i = \mathbf{a}_i^T \mathbf{x} + b_i, \quad i=1,2,\dots, P \quad (2)$$

Donde \mathbf{a}_i es un vector de parámetros y b_i es un escalar (umbral).

El mecanismo de defuzificación, se muestra en la siguiente ecuación:

$$y = \frac{\sum_{i=1}^K \mu_i(x) y_i}{\sum_{i=1}^K \mu_i(x)} \quad (3)$$

Este método es comúnmente conocido como defuzificación por medio de centros, [14]. El modelo de Takagi-Sugeno, [17], admite el uso en la parte consecuente de una función no difusa, la cual puede ser un modelo estático o dinámico (espacio de estados), donde la validación es determinada por la función de pertenencia. El uso de RNR en la parte antecedente de la regla puede disminuir drásticamente el número de intervalos y reglas difusas necesarias para una buena aproximación. Mastorocostas y Thecharis, [16], hacen un amplio estudio de distintos trabajos en la aérea y proponen como función antecedente usar la RNR de Frasconi-Gori-Soda, [18]. Esta RNR tiene la desventaja de ser secuencial, lo cual introduce un retardo variable en el modelo, dependiente de su orden relativo. Baruch y Gortcheva, [4], proponen como una función con-

secuente no difusa, usar un modelo de RNRE implantado en un multimodelo difuso-neuronal con dos intervalos - positivo y negativo, sin recubrimientos. La red RNRE tiene la ventaja de ser paralela con un retardo fijo (que es uno), independiente de la dimensión del vector de su estado. En el presente trabajo, nosotros estamos proponiendo usar un multimodelo difuso-neuronal con tres intervalos – positivo, negativo y cero, con recubrimientos. La regla difusa de identificación, propuesta, es dada por la siguiente representación:

$$R_i: \text{ IF } x \text{ is } A_i \text{ THEN } y_i(k+1)=N_i[x(k),u(k)], i=1,2,3 \quad (4)$$

Donde: $N_i(\cdot)$ denota la función de la RNRE; i es el número de la función RNRE; el número total de modelos RNRE es 3 (esto corresponde a las reglas difusas R_i). En el caso de control, el multimodelo difuso-neuronal está dado por la siguiente regla:

$$R_i: \text{ IF } x \text{ is } A_i \text{ THEN } u_i(k) = F_i[X(k), r(k), Y(k)], i=1,2,3 \quad (5)$$

Donde $u_i(k) = F_i[X(k), r(k), Y(k)]$, denota la función de control correspondiente; i es el número de la función RNRE; el número total de modelos RNRE es 3.

En [10], [11], [12], se expone la forma discreta de la RNRE con una estructura canónica de Jordán. El modelo está descrito por las siguientes ecuaciones:

$$X(k+1) = J X(k) + B U(k) \quad (6)$$

$$J = \text{diag}(J_i) ; |J_i| < 1 \quad (7)$$

$$Z(k) = S[X(k)] \quad (8)$$

$$Y(k) = S[C Z(k)] \quad (9)$$

Donde: $Y(\cdot)$, $X(\cdot)$, $U(\cdot)$ son vectores de salida, estado y entrada, con dimensiones L , N , M , respectivamente; $Z(\cdot)$ es un vector de la salida de la capa oculta con dimensión N ; J es una matriz diagonal ($N \times N$), esto es, $J = \text{diag}(J_i)$. La ecuación (7) es una condición de estabilidad, impuesta sobre todos los elementos J_i de J , que en realidad son los valores propios del modelo discreto de la RNRE. Las matrices B y C son de dimensión ($N \times M$) y ($L \times N$), respectivamente. El vector $S(\cdot)$ es un vector de funciones de activación (tangente hiperbólico) con una dimensión apropiada. Los parámetros del modelo son las matrices de los pesos J , B , C y el vector de estado $X(\cdot)$. Para ajustar los pesos de la RNRE se utiliza la ley de aprendizaje - retropropagación del error (Backpropagation - BP), descrita por la siguiente ecuación en una forma general:

$$W_{ij}(k+1) = W_{ij}(k) + \eta \Delta W_{ij}(k) + \gamma \Delta W_{ij}(k-1) \quad (10)$$

Donde η es la tasa de aprendizaje; γ es la tasa de momento; W_{ij} es el peso ij ; ΔW_{ij} es el gradiente de la función de costo con respecto al peso ajustable. Las actualizaciones ΔC_{ij} , ΔJ_{ij} , ΔB_{ij} de los pesos C_{ij} , J_{ij} , B_{ij} están dados por:

$$\Delta C_{ij}(k) = [T_j(k) - Y_j(k)] [1 - Y_j^2(k)] Z_i(k) \quad (11)$$

$$R1 = C_i(k) [T(k) - Y(k)] [1 - Z_j^2(k)] \quad (12)$$

$$\Delta J_{ij}(k) = R X_i(k-1) \quad (13)$$

$$\Delta B_{ij}(k) = R U_i(k) \quad (14)$$

Donde: T es el vector de la salida deseada con dimensión L (para el caso de identificación, T es la salida de la planta, y para el caso de control, T es la señal de referencia); $[T-Y]$ es el vector del error con dimensión L ; $R1$ es una variable auxiliar.

3 Identificación de Sistemas Usando MNJ

Un diagrama general de bloques del esquema de identificación dinámica neuronal usando una regla del MNJ de Identificación (MNI) está dado en la Figura 2.

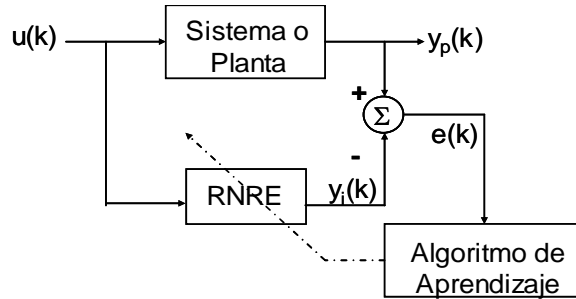


Fig.2. Diagrama general de bloques para la identificación de una planta usando una RNRE

En [10], [11], [12], se ha comentado acerca de la identificación de sistemas no lineales con RNRE. En este trabajo, se propone la implementación de un multimodelo neuronal jerárquico basándose en la similitud con el modelo de Takagi-Sugeno, [17]. El modelo MNJ tiene tres partes nombradas como: 1) fuzificación (o categorización), donde la señal de salida se divide en intervalos μ (aquí son tres – positivo [1, -0.5], negativo [-1, 0.5] y cero [-0.5, 0.5]); 2) inferencia, que es el Nivel Jerárquico Bajo (NJB) y contiene un cierto numero de reglas correspondientes a un cierto numero de redes neuronales (en este caso son tres) que operan sobre los intervalos μ ; 3) defuzificación, que es Nivel Jerárquico Alto (NJA), cual constituye de una red neuronal que es en realidad un sumador de las salidas de las RNREs del nivel bajo. El aprendizaje y el funcionamiento de ambos niveles jerárquicos son independientes. El objetivo principal del identificador neuronal es que la salida del MNJ de identificación tiene que seguir la salida de la planta y_p con un mínimo error global de identificación $e_i(k)$:

$$e_i(k) = y_p(k) - y_i(k) \quad (15)$$

El error local, usado para el ajuste de cada una RNRE que entra en el nivel bajo del MNJ, estará descrito como:

$$e_{ii}(k) = y_{pi}(k) - y_{ii}(k) \quad (16)$$

Figura 2 representa el diagrama general de bloques para la identificación de una planta usando una RNRE. El cambio de RNRE con un MNJ de identificación (con estructura, dada en Figura 3) nos da el diagrama general de bloques para este caso. Es claro denotar que la parte antecedente está relacionada directamente con la salida de la planta y_p , para que y_p , sea fuzificada en y_{pi} o $\mu_i \in y_p$ funciones de inclusión. Por su parte, para la defuzificación se propone sustituir la suma ponderada, denotada por la ecuación (3), con una RNRE, que representa el nivel alto del MNI y que tiene una simple arquitectura. La dicha RNRE requiere de las salidas y_{ii} como entradas y del error e_i , que permite el ajuste de los pesos en el algoritmo de aprendizaje (10).

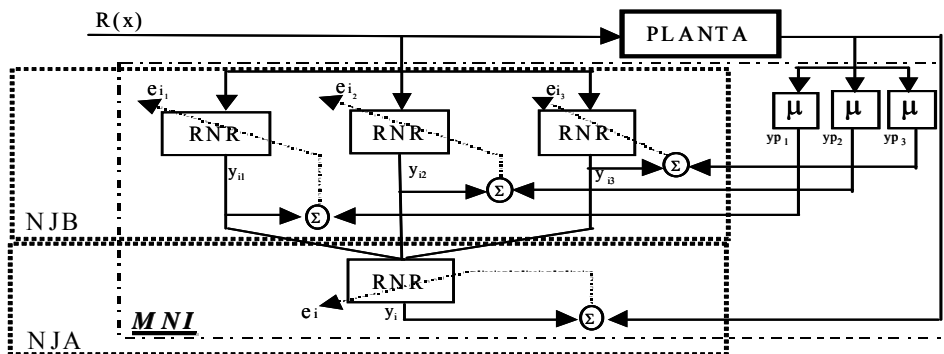


Fig.3. Diagrama de bloques de un MNJ de Identificación (MNI)

4 Esquemas de Control Adaptable

El control de sistemas adaptables requiere el diseño de un sistema que modifique el comportamiento natural de la planta con el objetivo de seguir una referencia deseada. En este esquema de control se supone que el diseñador tiene el suficiente conocimiento del sistema como para definir el comportamiento deseado. El primer trabajo de aplicación de redes neuronales para el control adaptable de sistemas es de Narendra y Parthasarathy, [19]. A continuación se presentan un par de esquemas de control adaptable para sistemas no lineales en tiempo real, en donde la identificación y el control se realizan por medio de MNJs.

4.1 Control Adaptable con Modelo Inverso

El control adaptable de seguimiento de trayectoria con modelo inverso de la planta es un control feedforward, donde se construye un modelo inverso en serie de la planta, capaz de compensar su dinámica en forma tal que la salida de la planta sigue la señal de referencia. El aprendizaje del controlador requiere minimizar una función de costo del error de seguimiento de la trayectoria deseada con respecto a los parámetros (pesos) del controlador neuronal. Bajo este principio se propone el uso de un Multimodelo Neuronal de Control (MNC), que es un MNJ y contiene las tres partes ya descritas (fuzificación, inferencia y defuzificación), y es usado como un modelo inverso de la planta e ilustrado en las Figuras 4 y 5. Aquí el error en la salida de la planta se transmite hacia la entrada usando un MNI. El error de control local y el error de control global, están dados por:

$$e_i(k) = R_i(k) - y_{pi}(k); e(k) = R(k) - y_p(k) \tag{17}$$

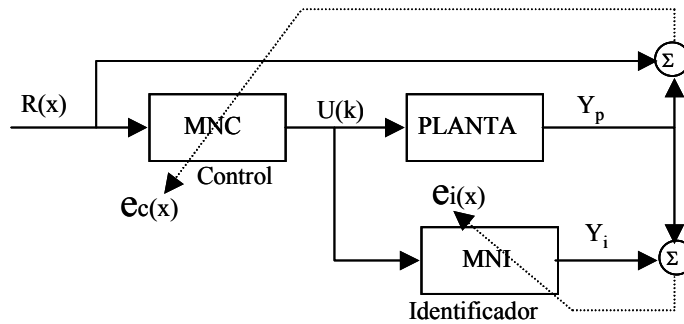


Fig. 4. Diagrama general de bloques para un control adaptable usando MNJ inverso

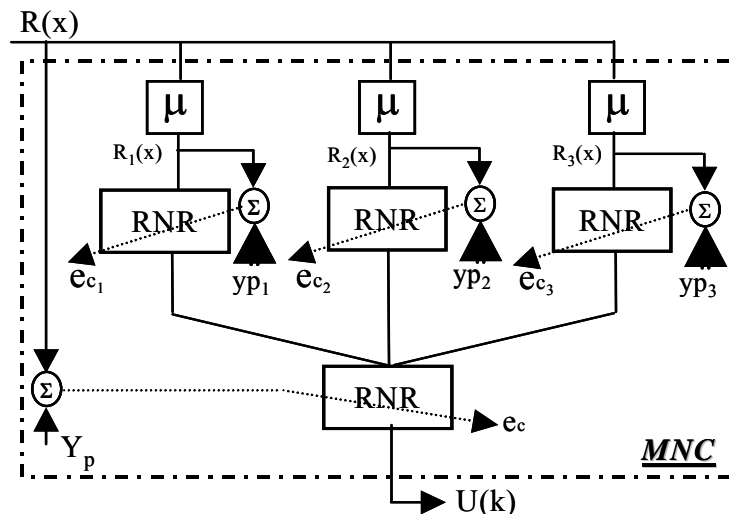


Fig. 5. Diagrama de bloques de un controlador adaptable (MNC) con MNJ inverso

La parte de la inferencia difusa del MNC está compuesta por un conjunto de i reglas, formadas por i RNREs (nivel bajo del controlador), entrenados con los errores propios de control para el ajuste de los pesos en cada uno de los algoritmos de aprendizaje, de tal modo que el ajuste de los pesos de la i RNRE, requiere del error local, formado por la R_i señal de referencia y la y_{pi} señal de salida de la planta, entregados por las μ_i de una fuzzificación respectiva (ver la ecuación (17)) de R y y_p . Referente a la defuzzificación, utilizamos una RNRE (nivel alto del controlador), la cual es ajustada directamente por el error global entre la señal de referencia R y la salida de la planta y_p , entregando así la señal de control U en la entrada de la planta, situación ilustrada en las Figuras 4 y 5.

4.2 Control Directo Adaptable

En la Figura 6, se muestra el esquema del control directo adaptable usando MNJs. Este control directo adaptable extiende las posibilidades del control similar, usando tres RNREs, [10]. El esquema de control consta de tres MNJs. Dos MNJs forman la parte del control, y la tercera es un identificador (estimator) de los estados locales de la planta, usados en el control. La parte del control está compuesta por un Multimodelo Neuronal de Control feedforward (MNCff), dependiente de la trayectoria de referencia, y también por otro Multimodelo Neuronal de Control feedback (MNCfb), dependiente de los estados que genera el Multimodelo Neuronal de Identificación (MNI). La estructura del controlador MNCfb está dada en la Figura 7.

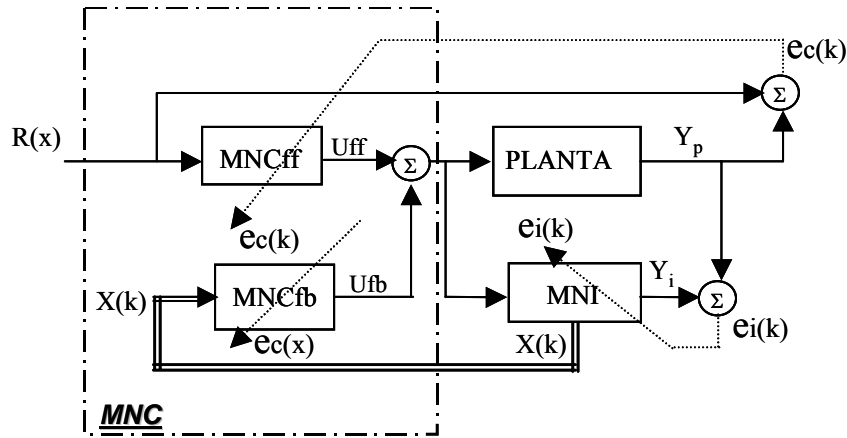


Fig. 6. Diagrama general de bloques de un control directo adaptable usando MNC

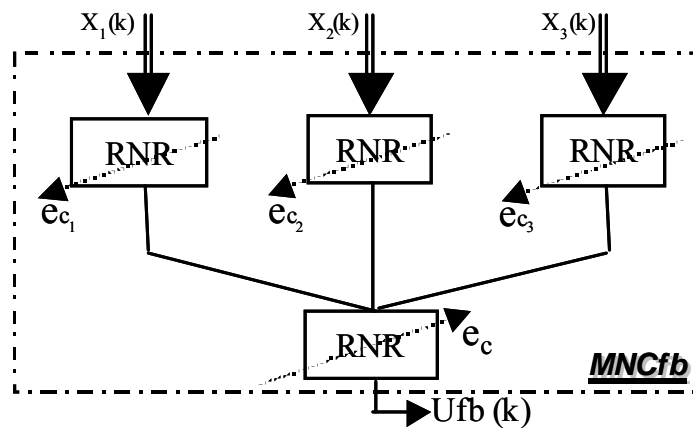


Fig. 7. Diagrama de bloques del controlador MNC feedback

El control entonces está formado por la suma de U_{ff} (control feedforward) y U_{fb} (control feedback). El controlador MNCff es similar al controlador MNC con modelo inverso, dado en la Figura 5, en donde el ajuste de los pesos de la i -RNRE requiere el error local de control (ver ecuación (17)), formado por la y_{pi} señal fuzzificada de la salida de la planta, y la

La señal de referencia, entregada por la μ_i de fuzzificación. Similarmente, la red neuronal de defuzzificación es ajustada directamente por el error global de control (ver la ecuación (17)), entregando así la parte feedforward de la señal de control.

El controlador MNC feedback tiene la misma estructura que el controlador MNC feedforward con la única diferencia que el tiene como entradas los vectores de estado estimados por MNI. Como se puede ver en Figura 7, el control feedback U_{fb} es directamente dependiente del número de los estados generados por el identificador - estimador como entradas, lo que está directamente ligado con el diseño de las capas ocultas de las RNRE del MNI, lo cual determina el número de entradas de las RNRE de control en el MNCfb.

5 Resultados de Simulación

En esta parte, las dos propuestas esquemas de control serán ilustradas usando el modelo discreto de una planta mecánica de un grado de libertad con fricción. El modelo de la fuerza fricción es basado al desarrollo, dado en [1], [2], [3].

5.1 Modelo General de la Planta con Fricción

La ecuación general de un sistema mecánico de un grado de libertad con fricción está dada por la siguiente ecuación:

$$m \ddot{\theta}(t) + fr(v,t) + d(t) = K_o u(t) \tag{18}$$

Donde: m es la masa; $v(t) = \dot{\theta}(t)$ es la velocidad; $\theta(t)$ es el desplazamiento relativo; $fr(v,t)$ es la fuerza de fricción; $u(t)$ es la fuerza de control; K_o es la ganancia del sistema; $d(t)$ es una perturbación externa acotada, causada por la fuerza de la carga. Se supondrá que la perturbación externa es acotada por una constante $d > 0$, y tiene la forma, dada por la ecuación siguiente:

$$d(t) = d_1 \theta(t) + d_2 \dot{\theta}(t) ; |d(t)| \leq d ; t > 0 \tag{19}$$

La fuerza-fricción, como se puede ver en la Figura 2, es comúnmente modelada como una combinación de la fricción estática, fricción de Coulomb, fricción viscosa y efecto Stribeck, [1], [2], [3].

La fuerza de fricción $fr(\cdot)$ está descrita como sigue:

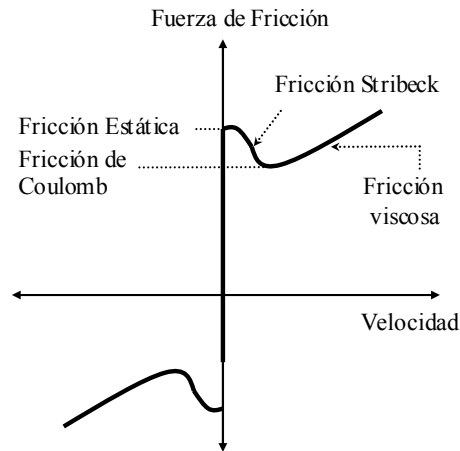


Fig.8. Comportamiento de la fuerza-fricción

$$fr(v,t) = F_{slip}(v) a(v) + F_{stick}(u)[1 - a(v)] \tag{20}$$

$$a(v) = \begin{cases} 1, & |v(t)| > \alpha \\ 0, & |v(t)| \leq \alpha \end{cases} \tag{21}$$

Donde: $\alpha > 0$; $F_{slip}(v)$ es la fuerza fricción de deslizamiento; $F_{stick}(u)$ es la fuerza de fricción a velocidad cero. Este término es usado para determinar si la masa permanecerá sin movimiento o se liberará de la fuerza de fricción estática (stick). Los límites positivo y negativo de la fuerza de fricción estática están dados por F_s^+ y F_s^- , respectivamente, que generalmente no son iguales en magnitud. El modelo de la fuerza $F_{stick}(u)$ en su forma siguiente considera esas asimetrías:

$$F_{stick}(u) = \begin{cases} F_s^+, & u(k) \geq F_s^+ \\ u(k), & F_s^- < u(k) < F_s^+ \\ 0, & u(k) \leq F_s^- \end{cases} \quad (22)$$

La masa no se puede mover hasta que la fuerza aplicada sea más grande en magnitud que la respectiva fuerza de fricción estática.

La fricción viscosa (slip) $F_{slip}(v)$, proporciona valores de la fuerza de fricción a velocidades diferentes de cero, y es representada como sigue:

$$F_{slip}(v) = F_d^+(v) b(v) + F_d^-(v) b(-v) \quad (23)$$

$$b(v) = \begin{cases} 1, & v(t) > 0 \\ 0, & v(t) \leq 0 \end{cases} \quad (24)$$

$$F_d^+(v) = F_s^+ - \Delta F^+ [1 - \exp(-v/v_{cr}^+)] + \beta^+ v \quad (25)$$

$$F_d^-(v) = F_s^- - \Delta F^- [1 - \exp(-v/v_{cr}^-)] + \beta^- v \quad (26)$$

Donde: ΔF^+ y ΔF^- son las caídas respectivas desde el nivel de fuerza estática hasta la fuerza de nivel cinética; v_{cr}^+ y v_{cr}^- son las velocidades críticas de Stribeck; β^+ y β^- son los coeficientes de fricción viscosa. La fuerza de fricción es modelada como una suma de fricción estática, fricción viscosa y el efecto Stribeck. En este caso, el nivel de la fricción de Coulomb (ver Figura 8) es presentado como $F_c^{+(-)} = F_s^{+(-)} - \Delta F^{+(-)}$. El efecto Stribeck significa que la fuerza-fricción decrece con el aumento de la velocidad, [2]. Con el fin de observar el comportamiento de dichas simulaciones, se realizara un cambio de la fuerza-fricción a los diez segundos del inicio de la simulación que tiene una duración de veinte segundos.

Los valores de los parámetros de la fricción están dados en Tabla 1. Los parámetros dados presentan una asimetría en la fuerza-fricción para valores positivos y negativos de la velocidad.

Por lo tanto el modelo en tiempo discreto de un sistema con un grado de libertad con fricción, es obtenido en la forma:

$$X_1(k+1) = X_2(k) \quad (27)$$

$$X_2(k+1) = 0.025X_1(k) - 0.3X_2(k) + 0.8U(k) - 0.1Fr(k) \quad (28)$$

$$V(k) = X_2(k) - X_1(k) \quad (29)$$

$$\theta(k) = 0.1X_1(k) \quad (30)$$

Donde $X_1(k)$, $X_2(k)$ son los estados del sistema; $V(k)$ es la velocidad angular y $\theta(k)$ es la posición del sistema; k es la variable de tiempo discreto, y la fuerza de fricción $Fr(k)$ está gobernada por las ecuaciones (20) hasta (26). Los parámetros utilizados para las simulaciones pertinentes están dados en Tabla 2.

Tabla 1. Parámetros de fricción

Parámetro	Fricción1	Fricción 2	Unidad
F_s^+	4.2	5.4	N
ΔF^+	1.8	2.5	N
v_{cr}^+	0.1	0.2	m/s
β^+	0.5	0.4	N s/m
F_s^-	-4.0	-5.0	N
ΔF^-	-1.7	-2.0	N
v_{cr}^-	0.1	0.2	m/s
β^-	0.5	0.4	N s/m
α	0.001	0.001	m/s
m	1.0	1.2	kg

Tabla 2. Parámetros utilizados en la simulación

Parámetro	Valor
Tasa-Momento, γ	0.01
Tasa-Aprendizaje, η	0.9
Periodo de Muestreo, T_o	0.1 seg.

Uno de ellos es T_o , el cual debe ser lo suficientemente pequeño para capturar los efectos de las no linealidades. La señal de referencia está dada por:

$$R(k) = \text{Sat} [0.5 \text{ Sen} (\pi k) + 0.5 \text{ Sen} (\pi k/2)] \tag{31}$$

Donde el nivel de saturación es ± 0.8 .

5.2 Análisis de los Resultados Gráficos Obtenidos

A continuación se presentan los resultados de simulación bajo la lógica difusa. El sistema de control contiene tres reglas (similar al MNJ) de tipo Takagi-Sugeno, compuestas por funciones de membresía de tipo campana de Gauss. La defuzificación está compuesta por la suma ponderada, descrita con la ecuación (3). Para la simulación se utilizó la herramienta “Fis Editor” de MATLAB (ver Figura 9).

Los resultados gráficos de simulación, obtenidos con un sistema difuso de control están presentados en la Figura 10 a,b.

Los resultados comparativos de identificación de un modelo mecánico con fricción en la carga, aplicando una sola RNRE y el MNJ de identificación, se ilustran en Figuras 11, a,b,c,d. La arquitectura de las RNRE en la parte consecuente de las reglas difusas es de (1, 5, 1- una entrada, una salida y cinco neuronas en la capa oculta). Para la parte de defuzificación y para esquemas de identificación con una RNRE se usa la topología (1, 3, 1). Figura 11,a ilustra los resultados gráficos de la identificación de la planta para el caso que usa una sola RNRE. Los resultados usando un MNI están dados en Figura 11,b. Los resultados del Error Medio Cuadrático (EMC) de identificación para ambos casos están dados en Figuras 11,c,d, respectivamente. Nosotros encontramos que el índice de desempeño en etapa final de la identificación (los 20 seg.) para el modelo con una RNRE es de 1.5% y para el MNI es de 0.28%.

Los resultados comparativos de seguimiento de trayectoria, aplicando un control con modelo inverso usando una sola red y usando MNC inverso, están dados en Figuras 12,a,b,c,d. Figuras 12,c,d ilustran el EMC de control, para el caso de control con una sola red y para el MNC inverso, donde el índice de desempeño está por debajo del 0.8% y 0.5% a los 20 seg., respectivamente.

Referentes a un controlador difuso, Figura 10,b ilustra el EMC de control, el cual muestra un índice de desempeño de 12% en 20 seg. Este valor es mayor que el resultado de simulación con redes neuronales, además de presentar mayor susceptibilidad a perturbaciones externas. Por ejemplo, en el cambio de fricción a los 10 seg después del inicio de la simulación, la señal de seguimiento de control se ve claramente afectada, como se puede ver en Figuras 10,a,b.

Los resultados de simulación utilizando el control directo adaptable para el modelo mecánico con fricción en la carga, usando tres RNREs y el MNCff-fb se muestran en las Figuras 13, a,b,c,d, respectivamente. Como se mencionó en el Parágrafo

4.2 de este artículo, el MNJ-control directo adaptable está formado por la suma de un control de tipo feedforward U_{ff} y un control de tipo feedback U_{fb} . La topología de RNRE en la i regla de la parte consecuente del control U_{ff} es (1, 5, 1) y la topología de RNRE en la i regla de la parte consecuente del control U_{fb} es (5, 5, 1), esto debido a que el control U_{ff} depende directamente de los estados estimados por el MNJ-identificador. Por lo tanto cada una de las i RNRE en nivel bajo de inferencia del MNI otorgará los estados $X(k)$ como entradas de la i RNRE en nivel bajo de inferencia del MNC. La topología para la RNRE en el nivel alto de defuzificación es de (1, 3, 1) así como para el control con tres RNREs.

Los resultados comparativos de seguimiento de trayectoria, usando un control directo adaptable con tres RNREs y con MNCff-fb, están dados en Figuras 13,a,b. Figuras 13,c,d ilustran el EMC de control para el caso de control tres RNREs y para el uso de MNCff-fb, donde el índice de desempeño está por debajo de 0.25% y 0.15% a los 20 segundos, respectivamente, superando el control con modelo inverso.

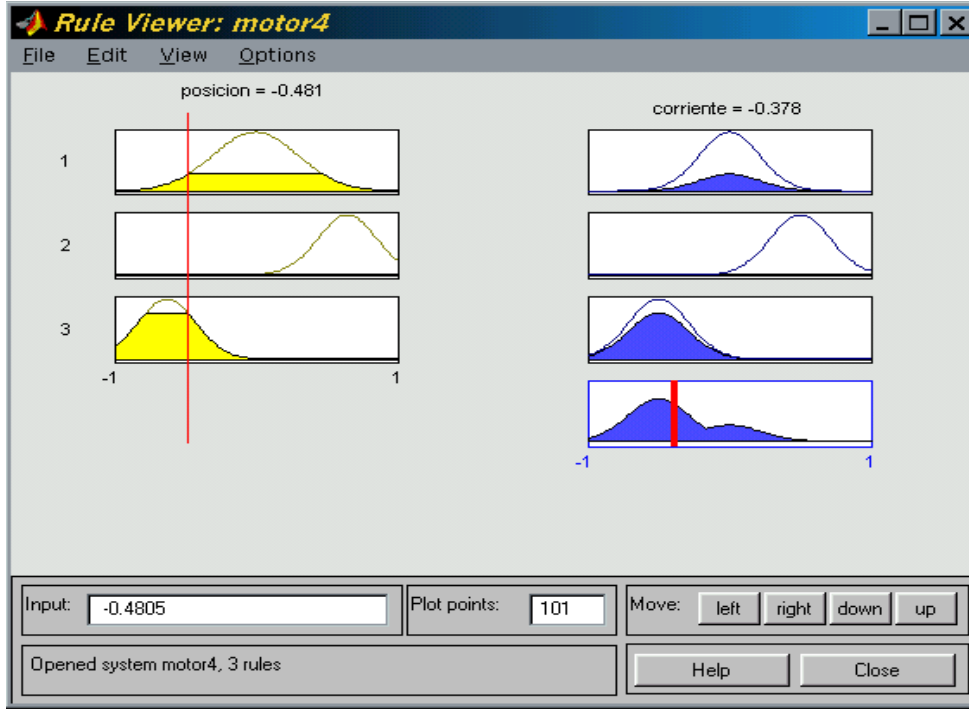


Figura 9. Simulador de sistemas difusas “FIS Editor”

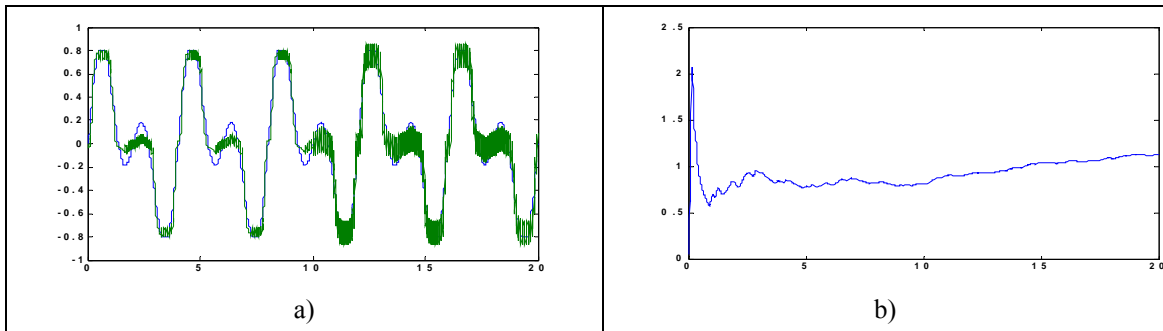


Fig. 10. Resultados de simulación usando un control difuso; a) Seguimiento de la señal de referencia por la salida de la planta; b) EMC de control

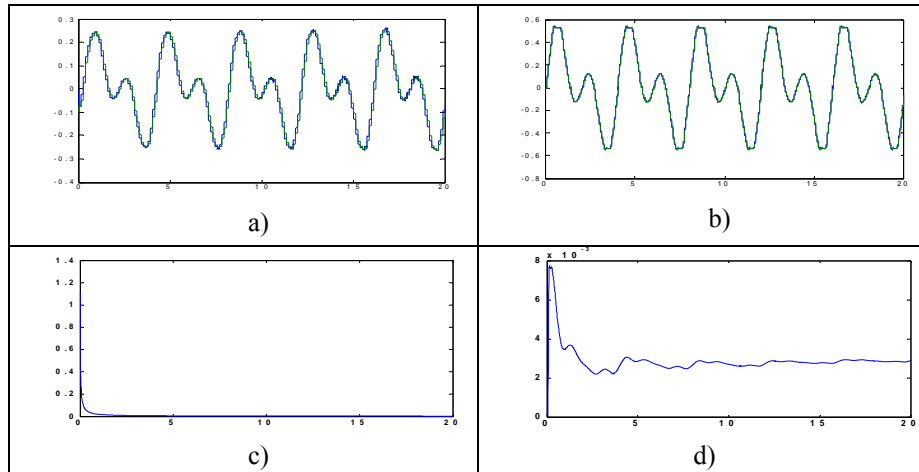


Fig. 11. Resultados comparativos de identificación con una sola RNRE y con MNI; a) Seguimiento de la salida de la planta por la salida de la RNRE de identificación; b) Seguimiento de la salida de la planta por la salida del MNJ de identificación; c) EMC de identificación con una RNRE; d) EMC de identificación con un MNI

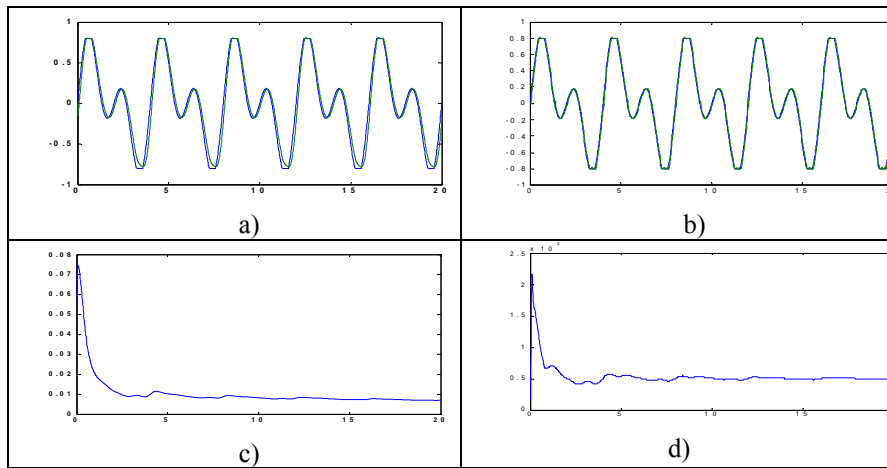


Fig. 12. Resultados comparativos de control con modelo inverso, usando RNRE y usando MNJ de control; a) Seguimiento de la señal de referencia por la salida de la planta usando RNRE de control; b) Seguimiento de la señal de referencia por la salida de la planta usando MNC; c) EMC de control con RNRE; d) EMC de control con un MNC

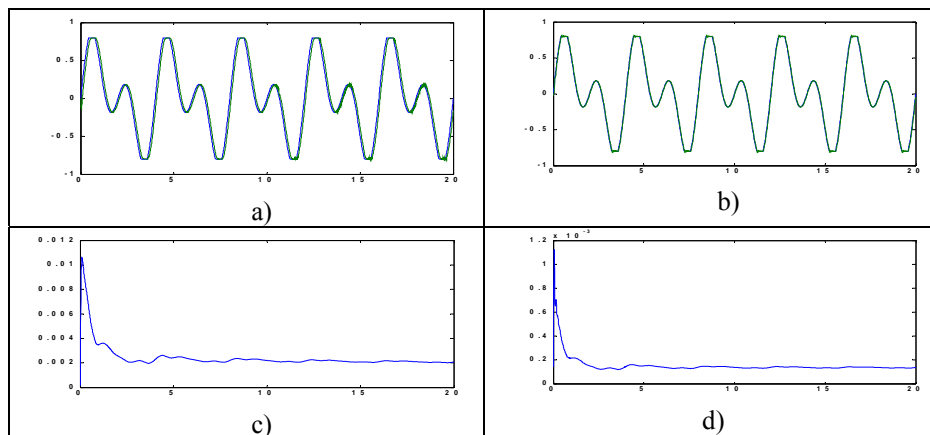


Fig. 13. Resultados comparativos de control directo adaptable, usando RNREs y usando MNJs de control; a) Seguimiento de la señal de referencia por la salida de la planta usando RNREs de control; b) Seguimiento de la señal de referencia por la salida de la planta usando MNCs; c) EMC de control con Rores; d) EMC de control con un MNCs

6 Conclusiones

En este artículo se propone la implementación de un Multimodelo Neuronal Jerárquico (MNJ), basándose en la similitud con el modelo difuso de Takagi-Sugeno. El modelo MNJ tiene tres partes: 1) fuzificación; 2) inferencia en el nivel bajo usando RNREs; 3) defuzificación en el nivel jerárquico alto usando una RNRE que es en realidad un filtro-sumador ponderado de las salidas de las RNRE del nivel bajo. El aprendizaje y el funcionamiento de ambos niveles jerárquicos son independientes. El modelo MNJ es implementado como identificador y controlador (feedforward, y feedback) en dos esquemas de control directo adaptable. Ambos esquemas de control son aplicados con una planta mecánica con fricción y comparados con otros esquemas de control neuronal y difuso, mostrando mejores resultados.

Al final, podemos concluir que el MNJ tanto de identificación como de control, es una herramienta poderosa, ya que guarda las virtudes de las redes neuronales recurrentes y la lógica difusa. Dichas virtudes son observadas en las simulaciones pertinentes de control, donde la lógica difusa no presenta buenos resultados y se desestabiliza cuando los parámetros de la fricción cambian, ya que una sola RNRE logra superar el dicho control. El controlador difuso logra tener un buen control al aumentar el número de reglas y por consecuencia el número de funciones de pertenencia, pero con un alto precio algorítmico-computacional. El MNJ propuesto realiza el control mejor que una sola RNRE y también al respecto de un controlador multimodelo neuronal sin recubrimiento, aun bajo los efectos de la fricción variable.

Referencias

1. **Karnopp, D.:** Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems. ASME Journal of Dynamic Systems, Measurement, and Control 107(1985) 100-103
2. **Lee, S.W., Kim, J.H.:** Robust Adaptive Stick-Slip Friction Compensation. IEEE Trans. Ind. Electr. 42 (1995) 474-479
3. **Menon, K., Krihnamurthy, K.:** Control of Low Velocity Friction and Gear Backlash in a Machine Tool Feed Drive System. Mecha-tronics 9 (1999) 33-52
4. **Baruch, I., Gortcheva, E.:** Fuzzy Neural Model for Nonlinear Systems Identification. In: Proc. of the AARTC'98 IFAC Workshop, Cancun, Mexico, 15-17 April, (1998) 283-288
5. **Baruch, I., Gortcheva, E., Thomas, F., Garrido, R.:** A Neuro-Fuzzy Model for Nonlinear Plants Identification. In: Proc. of the IASTED Int. Conf. on Modeling and Simulation, MS'99, Philadelphia, PA, USA, May 5-8, (1999) 1-6
6. **Baruch, I., Thomas, F., Garrido, R., Gortcheva, E.:** A Hybrid Multimodel Neural Network for Nonlinear Systems Identification. In: Proc. of the Int. Joint Conference on Neural Networks, Washington D.C., USA, July 10-16, 6 (1999) 4278-4283
7. **Baruch, I., Garrido, R., Mitev, A., Nenkova, B.:** A Neural Network Approach for Stick-Slip Friction Model Identification. In: Proc. of the 5-th Int. Conf. on Engineering Applications of NNs, Warsaw, Poland, Sept. 13-15, (1999) 183-188
8. **Baruch, I., Flores, J.M., Garrido, R., Gortcheva, E.:** Identificación de Sistemas No Lineales Complejos Usando un Multimodelo Neuronal Difuso. Científica, ESIME, 19 (2000) 29-40
9. **Baruch, I., Flores, J.M., Thomas, F., Gortcheva, E.:** A Multimodel Recurrent Neural Network for Systems Identification and Control. In: Proc. of the International Joint Conference on Neural Networks, Washington D.C., USA, July 14-19, (2001) 1291-1296
10. **Baruch, I., Flores, J.M., Thomas, F., Garrido, R.:** Adaptive Neural Control of Nonlinear Systems. In: Proc. of the Artificial Neural Networks Conf. –ICANN, Lecture Notes in Comp. Science, Vol. 2130, Springer-Verlag, Berlin, Heidelberg, New York (2001) 930-936
11. **Baruch, I., Flores, J.M., Nava, F., Ramirez, R., Nenkova, B.:** An Advanced Neural Network Topology and Learning, Applied for Identification and Control of a D.C. Motor. In: Proc. of the 1-st Int. IEEE Symposium on Intel. Systems, Varna, Bulgaria, Sept., (2002) 289-295
12. **Ramirez, I.R., Baruch, I., Garrido, R.:** Neuro Control Adaptable para un Motor CD. Científica, ESIME, 6, 3, Julio-Septiembre, (2002) 133-142
13. **Baruch, I., Beltran, R., del Pozo, A., Garrido, R.:** Control Multimodelo Neuronal para Sistemas Electromecánicos. Revista Ingeniería Electrónica, Automática y Comunicaciones, ISPJAE, La Habana, Cuba, ISSN 0258-5944, XXV, 1 (2004) 8-17
14. **Lin Chin-Ten, Lee C.S. George:** Neural Fuzzy Sistem, A Neuro-Fuzzy Synergism to Intelliegent Systems. Prentice - Hall PTR, New Jersey (1996)
15. **Teixeira, M., Zak, S.:** Stabilizing Controller Design for Uncertain Nonlinear Systems Using Fuzzy Models. IEEE Trans. Syst., Man, and Cyb., 7 (1999) 133-142
16. **Mastorocostas, P.A., Theocharis, J.B.:** Recurrent Fuzzy-Neural Model for Dynamic System Identification. IEEE Trans. Syst., Man, and Cyb. – Part B: Cybernetics, 32 (2002) 176-190
17. **Takagi, T., Sugeno, M.:** Fuzzy Identification of Systems and Its Applications to Modeling and Control. In: IEEE Trans. Syst., Man, and Cyb., 15 (1985) 116-132
18. **Frasconi, P., Gori, M., Soda, G.:** Local Feedback Multilayered Networks. Neural Computation, 4 (1992) 120-130
19. **Narendra, K. S., Parthasarathy, K.:** Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. Neural Networks, 1 (1990) 4-27



Ieroham Solomon Baruch. Es Profesor Titular A en el CINVESTAV-IPN (Centro de Investigación y de Estudios Avanzados del IPN), Departamento de Control Automático en la Ciudad de México, México, desde el año 1998 hasta hoy. Obtuvo su título de Ingeniero en Ingeniería Eléctrica, Facultad de Ingeniería Eléctrica, especialidad de Control Automático en 1969 en la Universidad de Ingeniería Mecánica y Eléctrica de Sofía, Bulgaria. El grado de Doctor en Ciencias Técnicas en 1974 en la Universidad de Ingeniería Mecánica y Eléctrica de Sofía, Bulgaria, Facultad de Ingeniería Eléctrica. Desde el año 1969 hasta el año 1997 el trabajó como Investigador auxiliar e Investigador Titular en el Instituto de Cibernética Técnica y Robótica en la Academia de Ciencias de Sofía, Bulgaria, ocupando distintos puestos hasta Jefe de Departamento. Sus áreas de investigación incluyen Redes Neuronales y Multimodelos Difuso-Neuronales con aplicación en sistemas de control automático de plantas mecánicas y biotecnológicas. Es autor y co autor de más de 120 artículos en revistas y memorias de congresos internacionales.



José Luis Olivares Guzmán. Nacido en la ciudad de México Distrito Federal en agosto de 1973. Realiza sus estudios de licenciatura en la Universidad Nacional Autónoma de México “UNAM”, de la cual se gradúa el 21 de Enero de 1999 con el título de Ingeniero Eléctrico-Electrónico con especialidad en Sistemas Digitales. Posteriormente, en septiembre del año 2000, ingresa al Centro de Investigaciones y Estudios Avanzados del I.P.N “CINVESTAV”, en el cual obtiene el grado de Maestro en Ciencias con especialidad en Control Automático en octubre del año 2003. Actualmente se encuentra laborando para Petróleos Mexicanos como responsable y Jefe de Mantenimiento en Terminales de Almacenamiento y Distribución de Pemex-Refinación.