

Adaptive Neural Sliding Mode Control in Discrete Time for a SCARA Robot Arm

F. G. Rossomando and C. M. Soria

Abstract— This work presents a neuro-adaptive control method in sliding mode control designed in discrete time for SCARA robot arm. The proposed control structure is based on neuro-adaptive sliding mode control to adjust dynamics variations of the SCARA robot. The sliding control is included to ensure that the discrete-time neural control can improve the stable closed loop system to be immune to the parameters variations. The proposed technique simultaneously ensures the stability of the adaptation of neural networks and can obtain adequate control when the robot's parameters are inexactly or unknown. This adaptive neural system was applied on a SCARA type robot arm where the trajectory tracking error converges to zero. Finally, experiments have been developed on SCARA robot arm to demonstrate the efficiency of the proposed technique, including comparison with a classical controller.

Keywords— Sliding Mode Control, MIMO system, Adaptive control, Neural Networks.

I. INTRODUCCIÓN

EL CONTROL de robots manipuladores ha sido un área de investigación desde hace años y ha desarrollado diversas estrategias de control ([1], [1],[3]).

Debido a que los robots manipuladores se componen de varias articulaciones unidas entre sí, las juntas tienen una dinámica altamente no lineales con un fuerte vínculo entre ellos. Esto complica la tarea de control, sobre todo cuando el robot es sometido a diferentes dinámicas.

Se han propuesto algunas técnicas con los sistemas de control que tienen en cuenta el modelo de la dinámica, como un control de torque computarizada ([4] y [5]). El trabajo proporciona resultados satisfactorios en términos de errores de seguimiento y robustez. Las incertidumbres en el modelo debido a malas estimaciones de los parámetros dificultan el diseño de un algoritmo de control eficiente basado en un modelo matemático preciso.

En [6], se proponen un control de seguimiento de trayectoria dinámica de robots manipuladores industriales que utilizan un controlador PD junto con un controlador neuronal. La red neuronal es una red tipo feedforward de tres capas. La ley de aprendizaje de los pesos de la red neural se obtuvo utilizando un modelo dinámico simplificado del modelo del robot y aplicando la teoría de retropropagación. Con este sistema, se realizaron las simulaciones de control de seguimiento de trayectoria y posteriormente los experimentos fueron llevados a cabo usando un robot manipulador industrial AdeptOne XL. Los resultados de su trabajo han demostrado la eficacia y la utilidad del método de control propuesto, y se demostró que el efecto de aprendizaje de la red neuronal afecta la precisión a la trayectoria de seguimiento. En [7], el problema de diseñar robusta estructura de control de variables y arrastrando

aviones modo se consideró para la RP41 SCARA robot manipulador. Los resultados de simulación muestran la robustez de la estructura de la variable de control de extensión, un modo deslizante frente a la carga transportada, las variaciones paramétricas, un modelo impreciso y con señales de perturbación externas.

En [8] Raafat et al., 2009, se implementó un esquema de control robusto adaptable basado en redes neuronales seguimiento para el robot SCARA. Las redes neuronales no se utilizan directamente para adaptar las incertidumbres del sistema, pero si se utilizan para ajustar los límites de las variaciones dinámicas dentro de un conjunto compacto. Las señales de salidas de las redes neuronales ajustan la ganancia del controlador de modo deslizante, de modo que los efectos no deseados producidos por las variaciones del sistema puedan ser eliminados. Y donde el error de seguimiento entre la salida robot y la señal de referencia deseada puede converger asintóticamente a cero. Basándose en un modelo de simulación se demostró la eficiencia del sistema. En el trabajo [9], se mostró a través de simulaciones que un robot SCARA de dos grados de libertad del tipo DDR se puede controlar con éxito por medio de control adaptativo sencillo. La estructura del sistema de control se basa en un modelo lineal y su aplicación es relativamente simple. El ángulo de la articulación puede seguir la trayectoria de referencia correctamente, y la perturbación de entrada constante puede ser suprimida por medio del ajuste de la ganancia integral. En [10], se presenta un diseño del controlador PID basado en Modelo de Control Interno (IMC) para un robot SCARA de dos grados de libertad. Se propone el filtro de paso bajo IMC adecuado para que el controlador PID se pueda obtener mediante la aplicación de expansión en serie de Maclaurin al controlador IMC en un lazo de control de realimentado. Los resultados simulación mostraron en este trabajo confirman que los controladores PID diseñado por el método propuesto pueden controlar las posiciones angulares a las posiciones deseadas del robot SCARA sin error de estado de estado estacionario. En [11], muestran una modelación integral y la identificación de un robot SCARA industrial desarrollado para incluir la dinámica del servo actuador. Se estudió el modelo de la cinemática del manipulador. Los autores utilizaron la mecánica de Lagrange para obtener las ecuaciones de movimiento. El controlador PD convencional se compara con un controlador neuronal para lograr un control preciso de las características de posición y movimiento. Cada articulación es tratada de forma individual para alcanzar un posicionamiento óptimo. El modelo de red neuronal es entrenado para lograr un posicionamiento preciso y minimizar los desplazamientos conjuntos. Este trabajo muestra los resultados de simulación que verifican el método de control propuesto; sin embargo, no proporciona un análisis de estabilidad. En [12] se presenta dos resultados derivados de la experimentación. En el primer esquema, el modelo que contiene la fricción de

F. G. Rossomando, Instituto de Automática, Conicet-UNSJ CCT-San Juan, frosoma@inaut.unsj.edu.ar

C. M. Soria, Instituto de Automática, Conicet-UNSJ CCT-San Juan, csoria@inaut.unsj.edu.ar

Coulomb y el efecto viscoso es estática. Sin embargo, el modelo dinámico es conocido, pero los valores de sus parámetros no lo son, por lo tanto el valor inicial se establece en cero. El esquema se identifica en línea durante el control de seguimiento de trayectoria. El segundo esquema es utilizado un modelo de fricción dinámica de Luge, siendo un modelo dinámico conocido, pero tiene un parámetro desconocido y su conjunto inicial es seteado a cero. Los resultados del experimento demuestran que los parámetros en el controlador PD adaptable utilizando la fricción dinámica pueden disminuir los errores en estado estable que ajustando un controlador PD utilizando la fricción estática. En otro trabajo [13], describen la simulación del control de movimiento de un brazo de robot de un grado de libertad SCARA articulado, siendo accionado por un par de actuadores neumáticos. El actuador neumático emula el comportamiento de los músculos biológicos; debido a su comportamiento no lineal, que utilizan este tipo de actuador, el cual se necesitaba para poner en práctica estrategias de control para los brazos de robot que utilizan este tipo de actuador. En este trabajo, un controlador PID se utiliza para la función de transferencia lineal y genera la información necesaria para entrenar a la red neuronal artificial del tipo perceptrón multicapa (RNAPM). Los resultados de simulación muestran que la RNAPM ha demostrado que superar el tiempo de respuesta del control PID, minimizando el error angular y evitando el problema de oscilación debido a su comportamiento continuo. En [14] describe un esquema de control de posición para robots utilizando redes neuronales de orden elevado, que aprenden cinemática inversa. En el trabajo mencionado, los resultados experimentales demuestran la eficacia del método de control propuesto, pero no proporcionan un análisis de estabilidad. En [15], propuso una estrategia fuzzy adaptable robusta de tiempo discreto para el diseño del controlador en modo deslizante. Este método podría ser aplicado para controlar sistemas no lineales, pero no muestra aplicaciones en sistemas de robóticos. En [16], presenta un controlador adaptable neuronal en modo deslizante, que fue diseñado para el seguimiento de la trayectoria para robots móviles. Los resultados mostrados se basan en experimentos y utilizar un método neuronal para adaptarse al modelo dinámico a ser controlado. Además, el sistema de control completo está diseñado en el dominio continuo.

En este trabajo, se presenta el diseño de un controlador de trayectoria seguimiento con capacidad neuronal adaptable basado en la dinámica nominal del robot que se aplica en un robot SCARA Bosch SR-800.. El neuro-controlador dinámico está diseñado basado en el "control de modo deslizante de control neuro-adaptable de tiempo discreto" (DTSMNAC), donde se utiliza una ley de adaptación en línea para ajustar los pesos de las funciones de base radial (RBF). Dicha ley está condicionada por la superficie de deslizamiento que se ha especificado. El neuro-controlador dinámico utiliza una red neural basada en funciones RBF básicos, que es el controlador principal a cargo de la dinámica inversa del robot SCARA, donde la compensación por superficie de deslizamiento está diseñado para eliminar el error de aproximación introducida por el controlador neuronal. Donde las leyes de adaptación del sistema de

control se obtienen a partir del criterio de estabilidad de Lyapunov. Por lo tanto, la estabilidad del sistema de control se puede asegurar, lo que se traduce como una propiedad de estabilidad convergente. La propuesta de un controlador basado en un "control de modo deslizante neuro adaptable de tiempo discreto" tiene las siguientes ventajas:

- (i). Esta técnica de control se puede aplicar a un sistema MIMO no lineal, que es el caso del manipulador de robot SCARA y otras dinámicas no lineales.
- (ii). El análisis completo se hizo en tiempo discreto utilizando el método discreto del Lyapunov.
- (iii). Se puede controlar la mayor parte de los sistemas de robot manipulador sin conocer sus modelos matemáticos exactos.
- (iv). La principal ventaja de esta técnica DTSMNAC sobre los basados en modelos es que no requiere un conocimiento previo de la dinámica del robot; Por otra parte, se puede ajustar los parámetros del controlador en la línea.
- (v). El esquema de control propuesto se ajusta a la parte principal de la dinámica del robot, que es un sistema robusto. Además, el controlador integra un control PI con el DTSMNAC. Con esta técnica de control, el efecto "chattering" puede reducirse a valores pequeños

Este trabajo está organizado de la siguiente manera: La sección 2 presenta una visión general del sistema y muestra la representación matemática de la dinámica del robot SCARA. El controlador neuronal adaptable en modo deslizante y su análisis de estabilidad se estudian en la Sección 3. Los resultados experimentales se muestran en la sección 4, que muestra la eficiencia de los controladores. Por último, las conclusiones se presentaron en la Sección 5.

II. MODELO ROBOTICO

Descripción del modelo.

Teniendo en cuenta que el controlador DTSMNAC comandará al robot en el movimiento plano horizontal, el modelo dinámico para las dos primeras articulaciones del robot se muestra como la Fig.1.

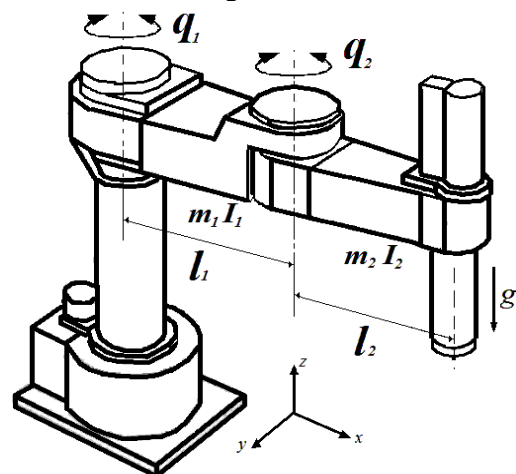


Figura 1. Articulaciones del robot tipo SCARA.

El modelo dinámico en el espacio de la articulación para un robot manipulador con n grados de libertad se mostró en [17]. En nuestro caso $n = 2$:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}(\dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (1)$$

donde

\mathbf{q} es el vector (2×1) de la coordenadas articulares (posicion de las articulaciones), $\mathbf{q} = [q_1, q_2]^T$

$\mathbf{M}(\mathbf{q})$ (2×2) matriz, normalmente se conoce como matriz de masa manipulador que contiene las funciones de la energía cinética del manipulador.

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ (2×2) matriz, representa pares resultantes de fuerzas centrífugas y de Coriolis.

$\mathbf{f}(\dot{\mathbf{q}})$ (2×1) representa los efectos de fricción de viscosidad cuando el manipulador se mueve en su espacio de trabajo.

$\boldsymbol{\tau}$: Vector de torques de los actuadores articulares (2×1)

Parámetros del Robot manipulador Bosch SR-800:

$$\hat{\mathbf{M}}(\mathbf{q}) = \begin{bmatrix} 1.7277 + 0.1908\cos(q_2) & 0.0918 + 0.0954\cos(q_2) \\ 0.3340 + 0.3418\cos(q_2) & 0.9184 \end{bmatrix};$$

$$\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 31.8192 - 0.0954\sin(q_2)(\dot{q}_2) & -0.0954\sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ 0.3418\sin(q_2)(\dot{q}_1) & 12.5783 \end{bmatrix} \quad (2)$$

$$\hat{\mathbf{f}}(\dot{\mathbf{q}}) = \begin{bmatrix} 1.0256\text{sign}(\dot{q}_1) \\ 1.7842\text{sign}(\dot{q}_2) \end{bmatrix}$$

El modelo dinámico no lineal de un robot SCARA (1) se utiliza para calcular las entradas de torque de control,

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{f}}(\dot{\mathbf{q}}) = \boldsymbol{\tau} + \boldsymbol{\Delta}(\mathbf{q}) \quad (3)$$

Donde las cantidades $\hat{\mathbf{M}}(\mathbf{q})$, $\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$, $\hat{\mathbf{f}}(\dot{\mathbf{q}})$ son estimaciones de los parámetros verdaderos y $\boldsymbol{\Delta}(\mathbf{q})$ es la incertidumbre desconocida que dependen del tiempo. A partir de (2) $\mathbf{M}(\mathbf{q})$ es una matriz definida positiva, por lo tanto se puede demostrar que $\det(\mathbf{M}(\mathbf{q})) \neq 0$, también es una matriz invertible.

Consideración 1: Es el vector función de incertidumbres $\boldsymbol{\Delta}(\mathbf{q})$ que es acotado por una constante $\|\boldsymbol{\Delta}(\mathbf{q})\| \leq \Delta_{Max}$

Para obtener un tiempo discreto modelo dinámico se utilizó una discretización directa con $\dot{q}_i = (q_i(k) - q_i(k-1))/T_0$ y $\ddot{q}_i = (q_i(k+1) - 2q_i(k) + q_i(k-1))/T_0^2$ Donde $T_0 = 1$ ms es el tiempo de muestreo y k es el índice de tiempo discreto.

$$\begin{pmatrix} \mathbf{q}(k) \\ \mathbf{q}(k+1) \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{\psi}(k) \end{pmatrix} \hat{\mathbf{f}}(\mathbf{q}(k)) + \begin{pmatrix} 0 & \mathbf{I} \\ -\boldsymbol{\alpha}(k) & -\boldsymbol{\beta}(k) \end{pmatrix} \begin{pmatrix} \mathbf{q}(k-1) \\ \mathbf{q}(k) \end{pmatrix} + \dots \quad (4)$$

$$+ \begin{pmatrix} 0 \\ \boldsymbol{\psi}(k) \end{pmatrix} \boldsymbol{\tau}(k) + \begin{pmatrix} 0 \\ \boldsymbol{\psi}(k) \end{pmatrix} \boldsymbol{\Delta}(k)$$

donde los parámetros de la matriz se definen como:

$$\boldsymbol{\beta}(\mathbf{q}(k), \mathbf{q}(k-1)) = -2\mathbf{I} + \hat{\mathbf{M}}^{-1}(\mathbf{q}(k))\hat{\mathbf{C}}(\mathbf{q}(k), \mathbf{q}(k-1))T_0$$

$$\boldsymbol{\alpha}(\mathbf{q}(k), \mathbf{q}(k-1)) = \mathbf{I} - \hat{\mathbf{M}}^{-1}(\mathbf{q}(k))\hat{\mathbf{C}}(\mathbf{q}(k), \mathbf{q}(k-1))T_0 \quad (5)$$

$$\boldsymbol{\psi}(\mathbf{q}(k)) = \hat{\mathbf{M}}^{-1}(\mathbf{q}(k))T_0^2$$

$$\boldsymbol{\delta}(k) = \boldsymbol{\psi}(\mathbf{q}(k))\boldsymbol{\Delta}(k)$$

Reorganizando la ec. (4), se obtiene:

$$\mathbf{q}(k+1) = \mathbf{F}(\mathbf{q}(k)) + \mathbf{G}(\mathbf{q}(k))\boldsymbol{\tau}(k) + \boldsymbol{\delta}(k) \quad (6)$$

donde

$$\mathbf{F}(\mathbf{q}(k)) = \boldsymbol{\psi}(k)\hat{\mathbf{f}}(\mathbf{q}(k)) - \boldsymbol{\alpha}(k)\mathbf{q}(k-1) - \boldsymbol{\beta}(k)\mathbf{q}(k) \quad (7)$$

$$\mathbf{G}(\mathbf{q}(k)) = \boldsymbol{\psi}(k) \quad (8)$$

III. CONTROLADOR DINAMICO

Formulación del Problema.

El controlador dinámico recibe la diferencia entre las referencias deseadas y posiciones angulares de salida (q_1 y q_2), que se envían a los servos del robot, como se muestra en la Fig. 2. Ahora, este vector de error de salida de posiciones articulares se define como:

$$\mathbf{e}(k) = \mathbf{q}(k) - \mathbf{q}_{ref}(k) = \begin{pmatrix} q_1(k) - q_{1ref}(k), q_2(k) - q_{2ref}(k) \end{pmatrix}^T \quad (9)$$

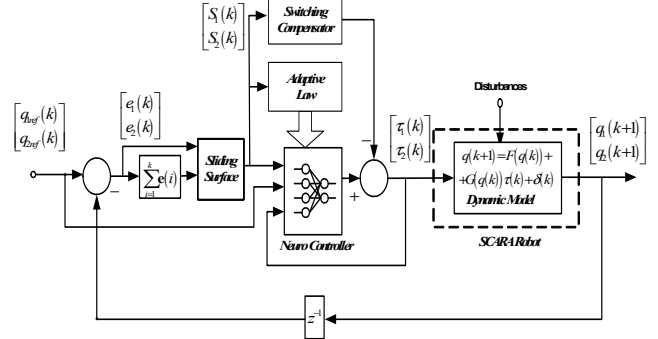


Figura 2. Sistema de control que incluye la estructura del controlador DTSMNAC.

El sistema de dinámico del robot (6) se puede escribir de la siguiente forma

$$\mathbf{q}(k+1) = \mathbf{F}(\mathbf{q}(k)) + \mathbf{G}(\mathbf{q}(k))\boldsymbol{\tau}(k) + \boldsymbol{\delta}(k) = \dots \quad (10)$$

$$= \begin{pmatrix} \mathbf{F}_1(\mathbf{q}(k)) \\ \mathbf{F}_2(\mathbf{q}(k)) \end{pmatrix} + \begin{pmatrix} \mathbf{G}_1(\mathbf{q}(k)) \\ \mathbf{G}_2(\mathbf{q}(k)) \end{pmatrix} \boldsymbol{\tau}(k) + \begin{pmatrix} \boldsymbol{\delta}_1(k) \\ \boldsymbol{\delta}_2(k) \end{pmatrix}$$

A fin de que (6) para ser controlable, se requiere que $\mathbf{G} \neq 0$, y sin pérdida de la generalidad, se supone que la dinámica del robot representan un sistema MIMO. $\mathbf{G}(k)$ es una matriz definida positiva ($\mathbf{M}(\mathbf{q})T_0^{-2}$) y $\boldsymbol{\delta}(k)$ es el vector de incertidumbres desconocido dependiente del tiempo, con δ_{Max} siendo su límite superior,

$$\delta_{Max} = \sup_{t \in \mathbb{N}^+} \|\boldsymbol{\delta}(k)\| \quad (11)$$

También definimos

$$\mathbf{q}(k) = \begin{pmatrix} q_1(k) & q_2(k) \end{pmatrix}^T$$

$$\boldsymbol{\tau}(k) = \begin{pmatrix} \tau_1(k) & \tau_2(k) \end{pmatrix}^T \quad (12)$$

$$\boldsymbol{\delta}(k) = \begin{pmatrix} \delta_1(k) & \delta_2(k) \end{pmatrix}^T$$

Son el vector de posiciones de salida del robot, el vector de torques (acciones de control) y el vector de incertidumbres paramétricas respectivamente.

Control de modo deslizante.

El principal objetivo es implementar un controlador neuronal adaptable que garantice la acotación de todas las variables para el sistema de lazo cerrado de control y el seguimiento de una señal de referencia acotada $q_{ref}(k)$.

El método de linealización por retroalimentación neuronal que se basó en el modelo NN-RBF puede resolver este tipo de problema de control ([12] y [13]).

El error de seguimiento de estado se define como $e(k) = \mathbf{q}(k) - \mathbf{q}_{ref}(k)$. En este trabajo el objetivo de control es encontrar una acción de control tal que el vector \mathbf{q} estado de salida del sistema en lazo cerrado seguirá al vector q_{ref} estado deseado, en otras palabras, el error de seguimiento debe converger a cero. El objetivo de la ley de control obtenida por modo deslizante es la de seguir la trayectoria del sistema en una superficie definida (calculado por el

diseñador) dentro del espacio de estado. Y mantenerse en toda la superficie para todos los tiempos posteriores.

Una superficie de deslizamiento para el sistema MIMO se puede definir en el estado de error $S(k)$ a partir de la siguiente ecuación:

$$S(k) = \begin{pmatrix} (t_d + \lambda_1 z^{-1}) & 0 \\ 0 & (t_d + \lambda_2 z^{-1}) \end{pmatrix} \sum_{i=1}^k e(i)T_0 = \begin{pmatrix} e_1(k) + \lambda_1 \sum_{i=1}^{k-1} e_1(i)T_0 \\ e_2(k) + \lambda_2 \sum_{i=1}^{k-1} e_2(i)T_0 \end{pmatrix} \quad (13)$$

donde T_0 es el tiempo de muestreo y $t_d = (1 - z^{-1}) / T_0$.

La diferencia discreta de la superficie deslizante $S(k)$ es:

$$\begin{aligned} S(k+1) - S(k) &= \dots \\ &= \begin{pmatrix} e_1(k+1) + \lambda_1 \sum_{i=1}^k e_1(i)T_0 \\ e_2(k+1) + \lambda_2 \sum_{i=1}^k e_2(i)T_0 \end{pmatrix} - \begin{pmatrix} e_1(k) + \lambda_1 \sum_{i=1}^{k-1} e_1(i)T_0 \\ e_2(k) + \lambda_2 \sum_{i=1}^{k-1} e_2(i)T_0 \end{pmatrix} = \dots \\ &= \begin{pmatrix} e_1(k+1) - e_1(k) + \lambda_1 e_1(k)T_0 \\ e_2(k+1) - e_2(k) + \lambda_2 e_2(k)T_0 \end{pmatrix} = \dots \\ &= \begin{pmatrix} e_1(k+1) + (\lambda_1 T_0 - 1)e_1(k) \\ e_2(k+1) + (\lambda_2 T_0 - 1)e_2(k) \end{pmatrix} \end{aligned} \quad (14)$$

Donde λ_i es una constante estrictamente positiva. Definimos $\rho = \text{diag}(\lambda_1 T_0 - 1, \lambda_2 T_0 - 1)$.

En el diseño del sistema de control de modo deslizante, primero se define la ley de control ideal equivalente τ^* , que determina la dinámica del sistema en la superficie de deslizamiento. La ley control ideal equivalente se obtiene reconociendo

$$\Delta S(k+1) = S(k+1) - S(k) \Big|_{\tau(k) = \tau^*(k)} = 0 \quad (15)$$

Substituyendo (14) en (15), se obtiene

$$\begin{aligned} \Delta S(k+1) &= \begin{pmatrix} e_1(k+1) + \rho_1 e_1(k) \\ e_2(k+1) - \rho_2 e_2(k) \end{pmatrix} = \mathbf{e}(k+1) + \boldsymbol{\rho}\mathbf{e}(k) = \dots \\ &= \left(\mathbf{F}(\mathbf{q}(k)) + \mathbf{G}(\mathbf{q}(k))\boldsymbol{\tau}^*(k) + \boldsymbol{\delta}(k) - \mathbf{q}_{ref}(k+1) \right) + \boldsymbol{\rho}\mathbf{e}(k) = 0 \end{aligned} \quad (16)$$

Ahora, vamos a considerar el problema de controlar el sistema no lineal incierto (6) como el tratado [12]. Definiendo una ley de control $\boldsymbol{\tau}^*$ que garantice la condición de deslizamiento (16), que se compone de un control equivalente

$$\boldsymbol{\tau}^*(k) = \mathbf{G}(\mathbf{q}(k))^{-1} \left[-\mathbf{F}(\mathbf{q}(k)) - \boldsymbol{\rho}\mathbf{e}(k) - \boldsymbol{\delta}(k) + \mathbf{q}_{ref}(k+1) \right] \quad (17)$$

Una condición para asegurar que la trayectoria del vector de error $e(k)$ va a evolucionar desde la fase inicial de la fase de deslizamiento, es seleccionar la estrategia de control de tal manera que:

$$(S(k+1) - S(k)) = -\mathbf{K}_d S(k)T_0 - \boldsymbol{\eta}T_0 \text{sign}(S(k)) \quad (18)$$

donde $\mathbf{K}_d = [K_{d1} \ 0; \ 0 \ K_{d2}]^T$, donde K_{d1} y K_{d2} son pequeños números reales positivos, $\boldsymbol{\eta} = [\eta_1 \ 0; \ 0 \ \eta_2]^T$ y el signo de la función se define por:

$$\text{sgn}(S_i) = \begin{cases} 1 & \text{for } S_i > 0 \\ 0 & \text{for } S_i = 0 \\ -1 & \text{for } S_i < 0 \end{cases} \quad (19)$$

De (16)

$$\begin{aligned} S(k+1) - S(k) &= \mathbf{e}(k+1) + \boldsymbol{\rho}\mathbf{e}(k) = \mathbf{e}(k+1) + \boldsymbol{\rho}\mathbf{e}(k) = \dots \\ &= \left(\mathbf{F}(\mathbf{q}(k)) + \mathbf{G}(\mathbf{q}(k))\boldsymbol{\tau}^*(k) + \boldsymbol{\delta}(k) - \mathbf{q}_{ref}(k+1) \right) + \boldsymbol{\rho}\mathbf{e}(k) = \dots \\ &= \mathbf{K}_d S(k)T_0 - \boldsymbol{\eta}T_0 \text{sign}(S(k)) \end{aligned} \quad (20)$$

De (20) se obtiene la ley de control ideal equivalente $\boldsymbol{\tau}^*$

$$\begin{aligned} \boldsymbol{\tau}^*(k) &= \mathbf{G}(\mathbf{q}(k))^{-1} \left[-\mathbf{F}(\mathbf{q}(k)) + \mathbf{q}_{ref}(k+1) - \dots \right. \\ &\quad \left. - \boldsymbol{\rho}\mathbf{e}(k) - \mathbf{K}_d S(k)T_0 - \boldsymbol{\eta}T_0 \text{sign}(S(k)) - \boldsymbol{\delta}(k) \right] \end{aligned} \quad (21)$$

Leyes de Ajuste del Sistema Neuronal.

En los sistemas reales, $F(q(k))$, $G(q(k))$, $\delta(k)$ y η puede ser desconocido y la función signo (S) es no continua. Por lo tanto, es imposible generar la ley de control (21). Para superar estas dificultades, se utiliza un sistema neural para la estimación, y para la aproximación, de $F(q(k))$, $G(q(k))$, $\delta(k)$, respectivamente

Los vectores de parámetros óptimos se definen como:

$$\begin{cases} \boldsymbol{\theta}_F^* = \arg \min_{\boldsymbol{\theta}_F \in \Omega_F} \left\{ \sup_{q \in \Omega_q} |F(\mathbf{q}(k)) - F(\mathbf{q}(k)|\boldsymbol{\theta}_F)| \right\} \\ \boldsymbol{\theta}_G^* = \arg \min_{\boldsymbol{\theta}_G \in \Omega_G} \left\{ \sup_{q \in \Omega_q} |G(\mathbf{q}(k)) - G(\mathbf{q}(k)|\boldsymbol{\theta}_G)| \right\} \\ \boldsymbol{\theta}_\delta^* = \arg \min_{\boldsymbol{\theta}_\delta \in \Omega_\delta} \left\{ \sup_{q \in \Omega_q} |\delta(k) - \delta(k|\boldsymbol{\theta}_\delta)| \right\} \end{cases} \quad (22)$$

donde Ω_F y Ω_G son conjuntos compactos con límites acotados en $\boldsymbol{\theta}_F$, $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_\delta$ y \mathbf{q} , respectivamente, que se define como

$$\begin{aligned} \Omega_F &= \{ \boldsymbol{\theta} \mid |\theta_F| \leq M_F \wedge |\theta_G| \leq M_G \wedge |\theta_\delta| \leq M_\delta \} \quad \text{y} \quad \Omega_q \\ &= \{ q \mid |q| \leq M_q \} \quad \text{donde } M_F, M_G, M_\delta \text{ y } M_q \text{ son constantes positivas.} \end{aligned}$$

Ahora utilizando la función gaussiana que se define como función de activación de cada neurona en la capa oculta (23).

$$\hat{\xi}(\mathbf{q}(k)) = \exp\left(-(\zeta(k) - \mathbf{c}_i)^T (\zeta(k) - \mathbf{c}_i) / \sigma_i^2\right) \quad (23)$$

donde i es la i -ésimo neurona de la capa oculta, c_i es la posición central de la neurona i -ésima, y σ_i es el factor de apertura de la función de Gauss, y los regresores de la función Gaussiana son:

$$\begin{aligned} \zeta(k) &= [q(k), q(k-1), \dots, q(k-n), \tau(k), q_{ref}(k+1)]^T \\ \mathbf{v}(k) &= [S(k), S(k-1), \dots, S(k-n), \tau(k), q_{ref}(k+1)]^T \end{aligned}$$

La estructura de RBF-NN se muestra en la Fig. 3.

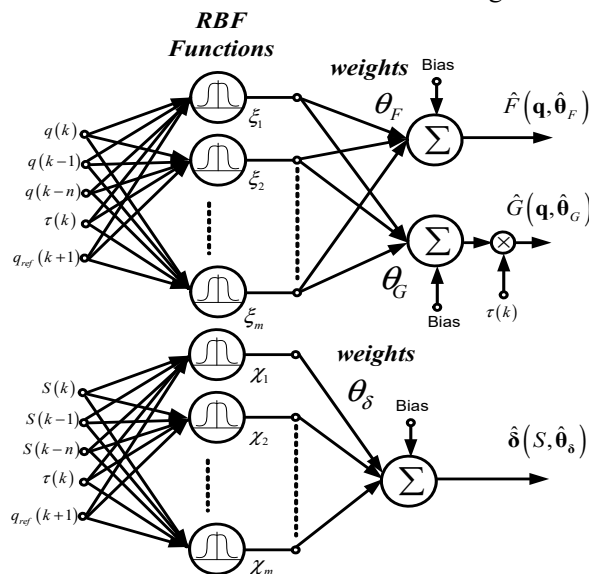


Figura 3. Red Neuronal de Funciones de Base Radial utilizadas en el control DTSMNAC.

El control $\tau(k)$ puede aproximarse por un RBF-NN través de un aprendizaje en línea,

$$\begin{aligned} \boldsymbol{\tau}^* = \hat{\mathbf{G}}^{-1}(\mathbf{q}(k), \boldsymbol{\theta}_G^*) & \left[-\hat{\mathbf{F}}(\mathbf{q}(k), \boldsymbol{\theta}_F^*) - \hat{\boldsymbol{\delta}}^*(\mathbf{S}(k), \boldsymbol{\theta}_\delta^*) + \mathbf{q}_{ref}(k+1) - \dots \right. \\ & \left. - \boldsymbol{\rho}\mathbf{e}(k) - \mathbf{K}_d \mathbf{S}(k) T_0 - \boldsymbol{\eta} T_0 \text{sign}(\mathbf{S}(k)) \right] \end{aligned} \quad (24)$$

Donde $\boldsymbol{\theta}_F^*(m \times j)$, $\boldsymbol{\theta}_G^*(j \times j)$ and $\boldsymbol{\theta}_\delta^*(m \times j)$ son los vectores de parámetros óptimos de pesos de $\boldsymbol{\theta}_F$, $\boldsymbol{\theta}_G$ y $\boldsymbol{\theta}_\delta$ respectivamente. Las funciones $\xi(m \times 1)$, $\chi(m \times 1)$ son las funciones de base radial vinculados a los pesos $\boldsymbol{\theta}_F$, $\boldsymbol{\theta}_G$ y $\boldsymbol{\theta}_\delta$ respectivamente.

Sin embargo, los vectores de parámetros óptimos son desconocidos, por lo que es necesario estimar sus valores. Definir una función de control estimada.

$$\begin{aligned} \boldsymbol{\tau} = \hat{\mathbf{G}}^{-1}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_G) & \left[-\hat{\mathbf{F}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_F) - \hat{\boldsymbol{\delta}}(\mathbf{S}(k), \hat{\boldsymbol{\theta}}_\delta) + \mathbf{q}_{ref}(k+1) - \dots \right. \\ & \left. - \boldsymbol{\rho}\mathbf{e}(k) - \mathbf{K}_d \mathbf{S}(k) T_0 - \boldsymbol{\eta} T_0 \text{sign}(\mathbf{S}(k)) + \boldsymbol{\tau}_\Delta \right] \end{aligned} \quad (25)$$

Se añadió una acción de control robusto $\boldsymbol{\tau}_\Delta(k)$ para atenuar la perturbación externa, que puede ser definido como ($\Delta \text{sign}(\mathbf{S})$). Las variables $\boldsymbol{\theta}_F$, $\boldsymbol{\theta}_G$ y $\boldsymbol{\theta}_\delta$ son pesos neuronales del sistema adaptativo de aproximación, $\hat{\mathbf{F}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_F)$, $\hat{\mathbf{G}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_G)$, $\hat{\boldsymbol{\delta}}(\mathbf{S}(k), \hat{\boldsymbol{\theta}}_\delta)$ respectivamente, y pueden expresarse por:

$$\hat{\mathbf{F}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_F) = \hat{\boldsymbol{\theta}}_F^T \xi(\zeta(k)) = \left(\sum_{i=1}^m \hat{\theta}_{F1i}^T \exp(-(\zeta(k) - \mathbf{c}_{F1})^T (\zeta(k) - \mathbf{c}_{F1}) / \sigma_{F1}^2) \right) \quad (26)$$

y

$$\hat{\mathbf{G}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_G) = \hat{\boldsymbol{\theta}}_G^T \xi(\zeta(k)) = \left(\sum_{i=1}^m \hat{\theta}_{G1i}^T \exp(-(\zeta(k) - \mathbf{c}_{G1})^T (\zeta(k) - \mathbf{c}_{G1}) / \sigma_{G1}^2) \right) \quad (27)$$

donde $\hat{\boldsymbol{\theta}}_{F1,2}^T$ ($m=5$) y $\hat{\boldsymbol{\theta}}_{G1,2}^T \in \mathbb{R}^{1 \times m}$

Es empleado otra término en la ley de control neuronal para atenuar la incertidumbre y perturbaciones externas. El termino de control MIMO es de la forma de:

$$\begin{aligned} \hat{\boldsymbol{\delta}}(\mathbf{S}(k), \hat{\boldsymbol{\theta}}_\delta) & = \hat{\boldsymbol{\theta}}_\delta^T \chi(\mathbf{v}(k)) = \dots \\ & = \left(\sum_{i=1}^m \hat{\theta}_{\delta 1i}^T \exp(-(\mathbf{v}(k) - \mathbf{c}_{\delta 1})^T (\mathbf{v}(k) - \mathbf{c}_{\delta 1}) / \sigma_{\delta 1}^2) \right) \\ & = \left(\sum_{i=1}^m \hat{\theta}_{\delta 2i}^T \exp(-(\mathbf{v}(k) - \mathbf{c}_{\delta 2})^T (\mathbf{v}(k) - \mathbf{c}_{\delta 2}) / \sigma_{\delta 2}^2) \right) = \dots \\ & = \left(\hat{\boldsymbol{\theta}}_{\delta 1}^T \chi(\mathbf{v}(k)) \right) \\ & = \left(\hat{\boldsymbol{\theta}}_{\delta 2}^T \chi(\mathbf{v}(k)) \right) \end{aligned} \quad (28)$$

donde $\hat{\boldsymbol{\theta}}_{\delta 1}^T$ y $\hat{\boldsymbol{\theta}}_{\delta 2}^T$ son parámetros ajustables, y la función vector $\chi(\mathbf{v}(k))$ es una función de la superficie de deslizamiento.

La ley de control global está dada por (25), y se define el mínimo error aproximación como:

$$\begin{aligned} \boldsymbol{\varepsilon}(k) & = \mathbf{F}(\mathbf{q}) - \hat{\mathbf{F}}(\mathbf{q}(k), \boldsymbol{\theta}_F^*) + \dots \\ & + \left[\mathbf{G}(\mathbf{q}) - \hat{\mathbf{G}}(\mathbf{q}(k), \boldsymbol{\theta}_G^*) \right] \boldsymbol{\tau} + \boldsymbol{\delta}(k) - \hat{\boldsymbol{\delta}}(\mathbf{S}(k), \boldsymbol{\theta}_\delta^*) \end{aligned} \quad (29)$$

Ahora, usando (16) y teniendo en cuenta el modelo dinámico de robot (6), se puede escribir como:

$$\begin{aligned} \Delta \mathbf{S}(k+1) & = \mathbf{S}(k+1) - \mathbf{S}(k) = \left[\mathbf{q}(k+1) - \mathbf{q}_{ref}(k+1) \right] + \boldsymbol{\rho}\mathbf{e}(k) = \dots \\ & = \boldsymbol{\rho}\mathbf{e}(k) + \left[\mathbf{F}(\mathbf{q}(k)) + \mathbf{G}(\mathbf{q}(k)) \boldsymbol{\tau}(k) + \boldsymbol{\delta}(k) - \mathbf{q}_{ref}(k+1) \right] \end{aligned} \quad (30)$$

Reemplazando la acción de control propuesta en (25) en (30)

$$\begin{aligned} \Delta \mathbf{S}(k+1) & = \boldsymbol{\rho}\mathbf{e}(k) + \left(\mathbf{F}(\mathbf{q}(k)) - \hat{\mathbf{F}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_F) \right) + \dots \\ & + \mathbf{q}_{ref}(k+1) + \left(\mathbf{G} - \hat{\mathbf{G}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_G) \right) \boldsymbol{\tau}(k) - \dots \\ & - \boldsymbol{\rho}\mathbf{e}(k) + \left[\boldsymbol{\delta}(k) - \hat{\boldsymbol{\delta}}(\mathbf{S}(k), \hat{\boldsymbol{\theta}}_\delta) \right] - \mathbf{q}_{ref}(k+1) - \dots \\ & - \mathbf{K}_d \mathbf{S}(k) T_0 - \boldsymbol{\eta} T_0 \text{sign}(\mathbf{S}(k)) + \boldsymbol{\tau}_\Delta = \dots \\ & = \left[\hat{\mathbf{F}}(\mathbf{q}(k), \boldsymbol{\theta}_F^*) - \hat{\mathbf{F}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_F) \right] + \left(\hat{\mathbf{G}}(\boldsymbol{\theta}_G^*) - \hat{\mathbf{G}}(\hat{\boldsymbol{\theta}}_G) \right) \boldsymbol{\tau}(k) \dots \\ & + \left[\hat{\boldsymbol{\delta}}(\mathbf{S}(k), \boldsymbol{\theta}_\delta^*) - \hat{\boldsymbol{\delta}}(\mathbf{S}(k), \hat{\boldsymbol{\theta}}_\delta) \right] + \boldsymbol{\varepsilon} - \dots \\ & - \mathbf{K}_d \mathbf{S}(k) T_0 - \boldsymbol{\eta} T_0 \text{sign}(\mathbf{S}(k)) + \boldsymbol{\tau}_\Delta \end{aligned} \quad (31)$$

considerando que

$$\hat{\mathbf{F}}(\mathbf{q}(k), \boldsymbol{\theta}_F^*) - \hat{\mathbf{F}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_F) = \boldsymbol{\theta}_F^{*T} \xi(\zeta(k)) - \hat{\boldsymbol{\theta}}_F^T \xi(\zeta(k)) = \tilde{\boldsymbol{\theta}}_F^T \xi(\zeta(k)) \quad (32)$$

$$\hat{\mathbf{G}}(\mathbf{q}(k), \boldsymbol{\theta}_G^*) - \hat{\mathbf{G}}(\mathbf{q}(k), \hat{\boldsymbol{\theta}}_G) = \boldsymbol{\theta}_G^{*T} \xi(\zeta(k)) - \hat{\boldsymbol{\theta}}_G^T \xi(\zeta(k)) = \tilde{\boldsymbol{\theta}}_G^T \xi(\zeta(k)) \quad (33)$$

$$\hat{\boldsymbol{\delta}}(\mathbf{S}(k), \boldsymbol{\theta}_\delta^*) - \hat{\boldsymbol{\delta}}(\mathbf{S}(k), \hat{\boldsymbol{\theta}}_\delta) = (\boldsymbol{\theta}_\delta^{*T} - \hat{\boldsymbol{\theta}}_\delta^T) \chi(\mathbf{v}(k)) = \tilde{\boldsymbol{\theta}}_\delta^T \chi(\mathbf{v}(k)) \quad (34)$$

donde $\tilde{\boldsymbol{\theta}}_F$, $\tilde{\boldsymbol{\theta}}_G$ y $\tilde{\boldsymbol{\theta}}_\delta$ son definidos como:

$$\tilde{\boldsymbol{\theta}}_F^T = \boldsymbol{\theta}_F^{*T} - \hat{\boldsymbol{\theta}}_F^T \quad (35)$$

$$\tilde{\boldsymbol{\theta}}_G^T = \boldsymbol{\theta}_G^{*T} - \hat{\boldsymbol{\theta}}_G^T \quad (36)$$

$$\tilde{\boldsymbol{\theta}}_\delta^T = \boldsymbol{\theta}_\delta^{*T} - \hat{\boldsymbol{\theta}}_\delta^T \quad (37)$$

Siendo, $\boldsymbol{\theta}_F^*$, $\boldsymbol{\theta}_G^*$ y $\boldsymbol{\theta}_\delta^*$ valores constantes óptimas, $\Delta \mathbf{S}$ se puede aproximar por

$$\begin{aligned} \Delta \mathbf{S}(k+1) & = -\mathbf{K}_d T_0 \mathbf{S}(k) + \tilde{\boldsymbol{\theta}}_F^T \xi(\zeta(k)) \boldsymbol{\tau} + \tilde{\boldsymbol{\theta}}_G^T \xi(\zeta(k)) + \dots \\ & + \tilde{\boldsymbol{\theta}}_\delta^T \chi(\mathbf{v}(k)) + \boldsymbol{\varepsilon} - \boldsymbol{\eta} T_0 \text{sign}(\mathbf{S}(k)) + \boldsymbol{\tau}_\Delta \end{aligned} \quad (38)$$

Consideración 1: La incertidumbre $\boldsymbol{\varepsilon}$ se supone que está delimitada por $\|\boldsymbol{\varepsilon}\| \leq \varepsilon_{Max}$

Consideración 2: La constante de ε_{Max} es igual a $\|\boldsymbol{\eta}\| T_0$

Teorema: Considere el sistema no lineal con incertidumbres definido por (6). Entonces, el controlador propuesto por (25) asegura la convergencia de seguimiento de error a cero, cuando se utilizan las siguientes leyes parámetros de adaptación:

$$\Delta \boldsymbol{\theta}_{F1} = -\gamma_1 S_1 \xi(\zeta(k)) \quad (39)$$

$$\Delta \boldsymbol{\theta}_{G1} = -\gamma_2 S_1 \xi(\zeta(k)) \boldsymbol{\tau}_1 \quad (40)$$

$$\Delta \boldsymbol{\theta}_{\delta 1} = -\gamma_3 S_1 \chi(\mathbf{v}(k)) \quad (41)$$

Demostración: Definimos una función candidata de Lyapunov $V(k)$ definida positiva que se define como

$$\begin{aligned} V(k) & = \frac{1}{2} \sum_{i=1}^2 \left[S_i^2(k) + \gamma_1^{-1} (\tilde{\boldsymbol{\theta}}_{F1}^T(k) - \tilde{\boldsymbol{\theta}}_{F1}^T(k-1)) + \dots \right. \\ & \left. + \gamma_2^{-1} (\tilde{\boldsymbol{\theta}}_{G1}^T(k) - \tilde{\boldsymbol{\theta}}_{G1}^T(k-1)) + \gamma_3^{-1} (\tilde{\boldsymbol{\theta}}_{\delta 1}^T(k) - \tilde{\boldsymbol{\theta}}_{\delta 1}^T(k-1)) \right] \end{aligned} \quad (42)$$

Ahora, haciendo la diferencia discreto de $V(k)$

$$\begin{aligned} \Delta V(k) & = \sum_{i=1}^2 \left[(S_i^2(k+1) - S_i^2(k)) + \dots \right. \\ & + \gamma_1^{-1} (\tilde{\boldsymbol{\theta}}_{F1}^T(k) \tilde{\boldsymbol{\theta}}_{F1}(k) - \tilde{\boldsymbol{\theta}}_{F1}^T(k-1) \tilde{\boldsymbol{\theta}}_{F1}(k-1)) + \dots \\ & + \gamma_2^{-1} (\tilde{\boldsymbol{\theta}}_{G1}^T(k) \tilde{\boldsymbol{\theta}}_{G1}(k) - \tilde{\boldsymbol{\theta}}_{G1}^T(k-1) \tilde{\boldsymbol{\theta}}_{G1}(k-1)) + \dots \\ & \left. + \gamma_3^{-1} (\tilde{\boldsymbol{\theta}}_{\delta 1}^T(k) \tilde{\boldsymbol{\theta}}_{\delta 1}(k) - \tilde{\boldsymbol{\theta}}_{\delta 1}^T(k-1) \tilde{\boldsymbol{\theta}}_{\delta 1}(k-1)) \right] \end{aligned} \quad (43)$$

Definiendo $\Delta \boldsymbol{\theta}_{F1}$, $\Delta \boldsymbol{\theta}_{G1}$ y $\Delta \boldsymbol{\theta}_{\delta 1}$ como:

$$\begin{aligned} \Delta \boldsymbol{\theta}_{F1} & = \gamma_1^{-1} (\tilde{\boldsymbol{\theta}}_{F1}^T(k) \tilde{\boldsymbol{\theta}}_{F1}(k) - \tilde{\boldsymbol{\theta}}_{F1}^T(k-1) \tilde{\boldsymbol{\theta}}_{F1}(k-1)) \\ \Delta \boldsymbol{\theta}_{G1} & = \gamma_2^{-1} (\tilde{\boldsymbol{\theta}}_{G1}^T(k) \tilde{\boldsymbol{\theta}}_{G1}(k) - \tilde{\boldsymbol{\theta}}_{G1}^T(k-1) \tilde{\boldsymbol{\theta}}_{G1}(k-1)) \end{aligned} \quad (44)$$

$$\Delta \boldsymbol{\theta}_{\delta 1} = \gamma_3^{-1} (\tilde{\boldsymbol{\theta}}_{\delta 1}^T(k) \tilde{\boldsymbol{\theta}}_{\delta 1}(k) - \tilde{\boldsymbol{\theta}}_{\delta 1}^T(k-1) \tilde{\boldsymbol{\theta}}_{\delta 1}(k-1))$$

y reordenando la ecuación (43).

$$\begin{aligned} \Delta V &= \sum_{i=1}^2 \left[(S_i^2(k+1) - S_i^2(k)) + \Delta \theta_{F_i} + \Delta \theta_{G_i} + \Delta \theta_{\delta_i} \right] = \dots \\ &= \sum_{i=1}^2 \left[\left((S_i(k) + \Delta S_i(k+1))^2 - S_i^2(k) \right) + \Delta \theta_{F_i} + \Delta \theta_{G_i} + \Delta \theta_{\delta_i} \right] = \dots \quad (45) \\ &= \sum_{i=1}^2 \left[2S_i(k) \Delta S_i(k+1) + \Delta S_i^2(k+1) + \Delta \theta_{F_i} + \Delta \theta_{G_i} + \Delta \theta_{\delta_i} \right] \end{aligned}$$

Reemplazando (38) en (45)

$$\begin{aligned} \Delta V &= \sum_{i=1}^2 \left[2 \left(-k_{di} T_0 S_i^2(k) + S_i(k) \tilde{\theta}_{G_i}^T \xi(\zeta(k)) \tau_i + \dots \right. \right. \\ &\quad \left. \left. + S_i \tilde{\theta}_{F_i}^T \xi(\zeta(k)) + S_i \tilde{\theta}_{\delta_i}^T \chi(\mathbf{v}(k)) + \dots \right. \right. \\ &\quad \left. \left. + S_i \varepsilon_i - \boldsymbol{\eta}_i T_0 S_i \operatorname{sign}(S_i(k)) \right) + \dots \right. \\ &\quad \left. + \Delta S_i^2(k+1) + \Delta \theta_{F_i} + \Delta \theta_{G_i} + \Delta \theta_{\delta_i} \right] \quad (46) \end{aligned}$$

de (44), reordenando $\Delta \theta_{F_i}$ como:

$$\begin{aligned} \Delta \theta_{F_i} &= \gamma_1^{-1} \left(\tilde{\theta}_{F_i}^T(k) \tilde{\theta}_{F_i}(k) - [\tilde{\theta}_{F_i}(k) - \Delta \tilde{\theta}_{F_i}(k)]^T [\tilde{\theta}_{F_i}(k) - \Delta \tilde{\theta}_{F_i}(k)] \right) = \dots \\ &= 2\gamma_1^{-1} \left(\tilde{\theta}_{F_i}^T(k) \Delta \tilde{\theta}_{F_i}(k) - (\Delta \tilde{\theta}_{F_i}^T(k) \Delta \tilde{\theta}_{F_i}(k)) \right) \quad (47) \end{aligned}$$

Haciendo el mismo análisis para $\Delta \theta_{G_i}$ y $\Delta \theta_{\delta_i}$, estas variables pueden ser expresadas como:

$$\Delta \theta_{G_i} = 2\gamma_2^{-1} \left(\tilde{\theta}_{G_i}^T(k) \Delta \tilde{\theta}_{G_i}(k) - (\Delta \tilde{\theta}_{G_i}^T(k) \Delta \tilde{\theta}_{G_i}(k)) \right) \quad (48)$$

$$\Delta \theta_{\delta_i} = 2\gamma_3^{-1} \left(\tilde{\theta}_{\delta_i}^T(k) \Delta \tilde{\theta}_{\delta_i}(k) - (\Delta \tilde{\theta}_{\delta_i}^T(k) \Delta \tilde{\theta}_{\delta_i}(k)) \right) \quad (49)$$

Reemplazando (39), (40) y (41) en (46) y reordenando

$$\begin{aligned} \Delta V(k) &= \sum_{i=1}^2 \left[-2k_{di} T_0 S_i^2(k) + \Delta S_i^2(k+1) + 2S_i \varepsilon_i + \dots \right. \\ &\quad \left. + 2\tilde{\theta}_{F_i}^T(k) (S_i \xi(\zeta(k)) + 2\gamma_1^{-1} \Delta \tilde{\theta}_{F_i}(k)) - \dots \right. \\ &\quad \left. - 2\gamma_1^{-1} (\Delta \tilde{\theta}_{F_i}^T(k) \Delta \tilde{\theta}_{F_i}(k)) + 2\tilde{\theta}_{G_i}^T(k) (S_i \xi(\zeta(k)) \tau_i + \gamma_2^{-1} \Delta \tilde{\theta}_{G_i}(k)) - \dots \right. \\ &\quad \left. - 2\gamma_2^{-1} (\Delta \tilde{\theta}_{G_i}^T(k) \Delta \tilde{\theta}_{G_i}(k)) + 2\tilde{\theta}_{\delta_i}^T(k) (S_i \chi(\mathbf{v}(k)) + \gamma_3^{-1} \Delta \tilde{\theta}_{\delta_i}(k)) - \dots \right. \\ &\quad \left. - 2\gamma_3^{-1} (\Delta \tilde{\theta}_{\delta_i}^T(k) \Delta \tilde{\theta}_{\delta_i}(k)) - 2\boldsymbol{\eta}_i T_0 S_i \operatorname{sign}(S_i(k)) \right] \quad (50) \end{aligned}$$

Reemplazando las leyes de ajuste (39), (40) y (41) en $\Delta \tilde{\theta}_{F_i}$, $\Delta \tilde{\theta}_{G_i}$ y $\Delta \tilde{\theta}_{\delta_i}$ de (50).

$$\begin{aligned} \Delta V(k) &= \sum_{i=1}^2 \left[-2k_{di} T_0 S_i^2(k) + \Delta S_i^2(k+1) + \dots \right. \\ &\quad \left. + 2S_i \varepsilon_i - 2\gamma_1^{-1} (\Delta \tilde{\theta}_{F_i}^T(k) \Delta \tilde{\theta}_{F_i}(k)) - \dots \right. \\ &\quad \left. - 2\gamma_2^{-1} (\Delta \tilde{\theta}_{G_i}^T(k) \Delta \tilde{\theta}_{G_i}(k)) - \dots \right. \\ &\quad \left. - 2\gamma_3^{-1} (\Delta \tilde{\theta}_{\delta_i}^T(k) \Delta \tilde{\theta}_{\delta_i}(k)) - 2\boldsymbol{\eta}_i T_0 |S_i(k)| \right] \quad (51) \end{aligned}$$

De acuerdo a (38)

$$\begin{aligned} |\Delta S_i(k+1)| &\leq |K_{di} T_0 S_i(k)| + |\alpha_0| + |\tau_{\Delta i}| + \dots \\ &\quad + \|\tilde{\theta}_{F_i}^T\| \|\xi(\zeta(k))\| + \|\tilde{\theta}_{G_i}^T\| \|\xi(\zeta(k))\| \tau_i + \|\boldsymbol{\eta}_i\| T_0 + \dots \\ &\quad + \|\tilde{\theta}_{\delta_i}^T\| \|\chi(\mathbf{v}(k))\| + |\varepsilon_i| + |\tau_{\Delta i}| \quad (52) \end{aligned}$$

Todos los términos en (52) son acotados, y considerando $\alpha_0 = \|\tilde{\theta}_{G_i}^T\| \|\xi(\zeta(k))\| \tau_i + \|\boldsymbol{\eta}_i\| T_0 + \|\tilde{\theta}_{F_i}^T\| \|\xi(\zeta(k))\| + \|\tilde{\theta}_{\delta_i}^T\| \|\chi(\mathbf{v}(k))\| + |\varepsilon_i|$, (52) pueden ser expresados como:

$$|\Delta S_i(k+1)| \leq |K_{di} T_0 S_i(k)| + |\alpha_0| + |\tau_{\Delta i}| \quad (53)$$

Y aplicando la función cuadrado ambos lados de (53),

$$\begin{aligned} |\Delta S_i(k+1)|^2 &\leq \left[|K_{di} T_0 S_i(k)| + |\alpha_0| + |\tau_{\Delta i}| \right]^2 \leq \dots \\ &\leq |K_{di} T_0 S_i(k)|^2 + 2|K_{di} T_0 S_i(k)| |\alpha_0| + \dots \quad (54) \end{aligned}$$

$$+ 2|K_{di} T_0 S_i(k)| |\tau_{\Delta i}| + 2|\alpha_0| |\tau_{\Delta i}| + |\alpha_0|^2 + |\tau_{\Delta i}|^2$$

Ahora adicionando y substrayendo el termino $2|\tau_{\Delta i}| |S_i(k)|$ y reescribiendo la ecuación (54).

$$\begin{aligned} &\leq \left[|K_{di} T_0 S_i(k)| + |\alpha_0| \right]^2 + 2|K_{di} T_0 S_i(k)| |\tau_{\Delta i}| + \dots \\ &\quad + 2|\alpha_0| |\tau_{\Delta i}| + 2|S_i(k)| |\tau_{\Delta i}| - 2|S_i(k)| |\tau_{\Delta i}| + |\tau_{\Delta i}|^2 \quad (55) \end{aligned}$$

Reescribiendo y reordenando (55)

$$|\Delta S_i(k+1)|^2 \leq \left(\beta_2(k) + \beta_1(k) |\tau_{\Delta i}| + |\tau_{\Delta i}|^2 \right) + 2|\tau_{\Delta i}| |S_i(k)| \quad (56)$$

donde

$$\begin{aligned} \beta_2 &= \left[|K_{di} T_0 S_i(k)| + |\alpha_0| \right]^2 \\ \beta_1 &= 2|K_{di} T_0 S_i(k)| + 2|\alpha_0| - 2|S_i(k)| \quad (57) \end{aligned}$$

De (56) $\tau_{\Delta i}$ es seleccionada como:

$$\begin{aligned} 2S_i(k) u_{\Delta i} = 2S_i(k) \Delta_i \operatorname{sign}(S_i(k)) = \dots \\ = - \left[-\beta_1(k) + \sqrt{\beta_1^2(k) - 4\beta_2(k)} \right] |S_i(k)| \quad (58) \end{aligned}$$

con las últimas consideraciones, se demuestra fácilmente que:

$$\begin{aligned} \Delta V(k+1) &= \sum_{i=1}^2 \left[-2k_{di} T_0 S_i^2(k) + 2S_i \varepsilon_i - \dots \right. \\ &\quad \left. - 2\gamma_1^{-1} (\Delta \tilde{\theta}_{F_i}^T(k) \Delta \tilde{\theta}_{F_i}(k)) - \dots \right. \\ &\quad \left. - 2\gamma_2^{-1} (\Delta \tilde{\theta}_{G_i}^T(k) \Delta \tilde{\theta}_{G_i}(k)) - \dots \right. \\ &\quad \left. - 2\gamma_3^{-1} (\Delta \tilde{\theta}_{\delta_i}^T(k) \Delta \tilde{\theta}_{\delta_i}(k)) - 2\boldsymbol{\eta}_i T_0 |S_i(k)| \right] < 0 \quad (59) \end{aligned}$$

Este resultado produce un mejor control del robot SCARA ya que las velocidades de error del controlador dinámico tiende a cero.

IV. RESULTADOS EXPERIMENTALES

En esta sección se describe el robot industrial utilizado para la aplicación de los algoritmos de control que se proponen en las secciones 3.1 y 3.2. Para ello, se considera el robot SCARA BOSCH SR-800 (Fig.4) con 2 grados de libertad. El robot manipulador propuesto tiene un control de la CPU Intel Dual Core a bordo, que funciona a una frecuencia de 2,6 GHz con 4 GB de memoria RAM. El equipo tiene sistema operativo un Linux Debian con RTAI (Real Time Application Interface). El esquema de control diseñado se aplica al manipulador del robot a través del control de la unidad de potencia, la cual que introduce las acciones de control a las articulaciones en forma de torques. Para llevar a cabo experimentos mediante los algoritmos de control DTSMNAC y PID estáticos se aplican al mismo robot.

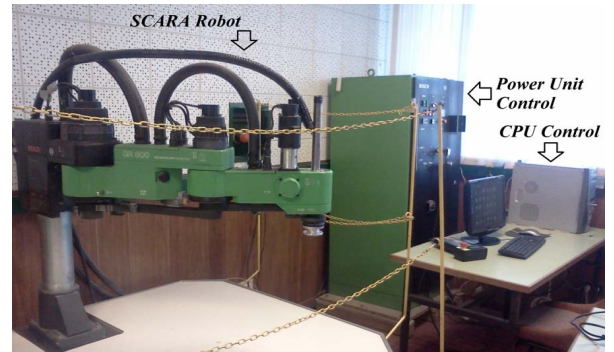


Figura 4. SCARA Robot and Power Unit Control.

Para el experimento, el controlador DTSMNAC se inició con pesos aleatorios y se ajustó utilizando diferentes trayectorias. En el experimento que se muestra en este documento, los parámetros neuronales se ajustan en línea por el algoritmo de aprendizaje, evaluando su evolución en función del tiempo.

La trayectoria elegida de referencia para poner a prueba a ambos controladores posee la forma de ocho. En la Fig. 5, se

muestran las acciones de control del controlador DTSMNAC. Además, las evoluciones de los pesos de ajuste neurales del controlador DTSMNAC que se muestran en la Fig. 9, Fig. 10 y Fig. 11 respectivamente. La figura 6 muestra las señales de referencia de posición y la posición instantánea del robot SCARA. La Figura 7 muestra las trayectorias seguidas por el robot SCARA, utilizando cada uno de los controladores. En la Fig. 8, se muestra la norma cuadrada de los errores de control de ambos controladores. El mayor error corresponde al controlador PID clásico, que no posee ningún ajuste en línea. En este caso se observa claramente el efecto de las incertidumbres en el error. Pero, el menor error se obtuvo por el controlador DTSMNAC propuesto en este trabajo, siendo el menos afectado por la estructura no modelada y las posibles perturbaciones externa.

El controlador DTSMNAC fue diseñado para ser robusto con respecto a errores de modelado. También es más eficaz en el rechazo de perturbaciones y no produce error constante causado debido a las incertidumbres en los parámetros o perturbaciones externas. Además, el controlador PID estático es vulnerable a los cambios en el modelo dinámico y a las posibles incertidumbres; siendo que al control PID estático está diseñado para un modelo linealizado de la dinámica del robot SCARA. La ley de control desarrollada en este trabajo, para un sistema no lineal, no se necesita conocer el modelo dinámico del robot SCARA. Además, se sabe que una representación matemática de un modelo dinámico no describe con precisión el comportamiento real.

Los efectos de estas no linealidades e incertidumbres del modelo, junto con las variaciones en la dinámica del robot demuestran la robustez del controlador DTSMNAC. La estabilidad del sistema de control propuesto se demostró analíticamente a través de la teoría de la estabilidad de Lyapunov discreta (Sección III).

Esta propuesta de control inteligente puede ser considerada como una solución general para el control de sistemas no lineales y, en particular, para el caso de sistemas robóticos o cuando la dinámica es variable o posee incertidumbres en el modelo.

Este campo de investigación continúa abierta a las cuestiones de modelado, la estabilidad matemática, la convergencia y el análisis de robustez de control de sistemas continúan avanzando para diseñar controladores cada vez más precisos

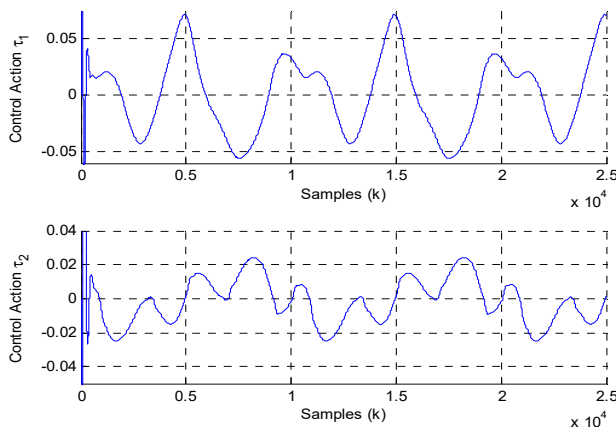


Figura 5. Acciones de control del controlador DTSMNAC.

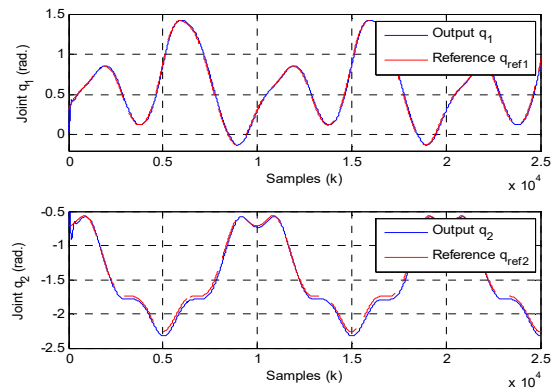


Figura 6: Señales de salida y de Referencia de las posiciones articulares del robot SCARA.

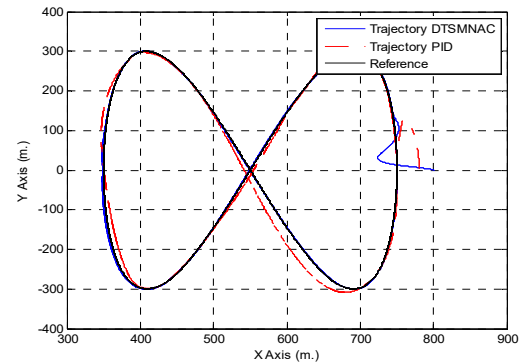


Figura 7. Trayectoria seguida por el robot SCARA con controlador DTSMNAC (línea continua) comparado con la trayectoria seguida por el robot con un controlador PID (línea de puntos).

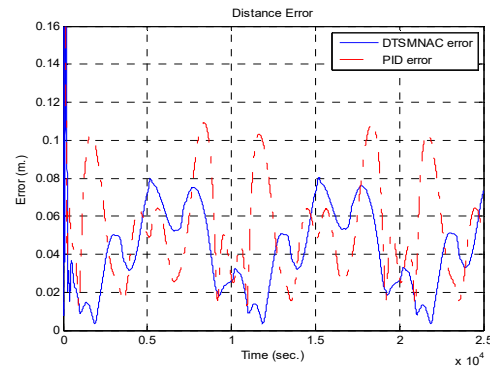


Figura 8. Error cuadrático instantáneo de la posición del robot. Con cont. DTSMNAC (línea continua) y con PID estático (línea punteada).

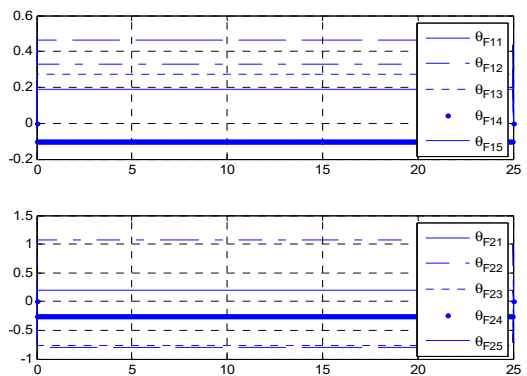


Figura 9. Evolución de los pesos θ_F del controlador neural DTSMNAC durante el experimento.

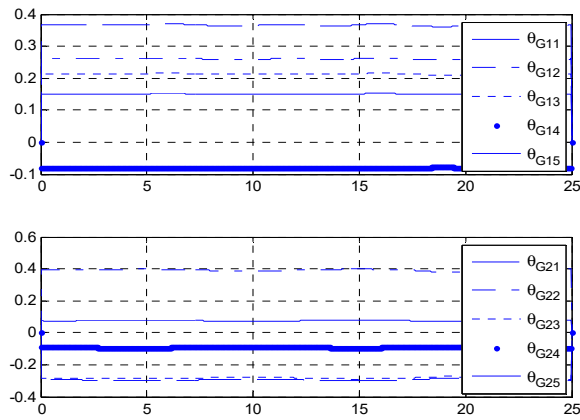


Figura 10. Evolución de los pesos θ_G del controlador neural DTSMNAC durante el experimento.

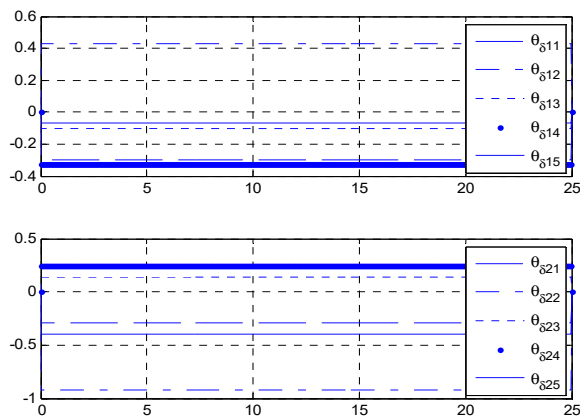


Figura 11. Evolución de los pesos θ_δ del controlador neural DTSMNAC durante el experimento.

CONCLUSIONES

Se ha propuesto un controlador neuronal adaptable de modo deslizante robusto para el seguimiento de la trayectoria de un brazo robot tipo SCARA, donde la dinámica no lineal es desconocida. El núcleo de esta estructura no requiere el conocimiento previo de la dinámica del sistema y de los parámetros del mismo para calcular el control equivalente, donde un sistema de adaptable neural en modo deslizante se desarrolla para compensar aún más la incertidumbre del sistema. De este esquema se obtiene robustez en el sentido de que el mecanismo de auto-sintonización puede ajustar automáticamente el controlador de modo deslizante neural mediante el uso de un algoritmo de aprendizaje. Y la estabilidad asintótica global del algoritmo se establece a través de las condiciones de estabilidad de Lyapunov discreta.

Cuando se produce el mapeo con el modelo predefinido, todo el sistema de control se vuelve equivalente a un sistema dinámico estable en el dominio de tiempo discreto.

El diseño se puede lograr el objetivo de control de modo adaptable en modo deslizante en tiempo discreto y también asegurarse de que el error de seguimiento de salida converge a cero.

REFERENCIAS

- [1] Canudas de Wit, C., Siciliano, B. and Bastin, G. *Theory of robot control*. Springer Verlag, (1996).
- [2] Lewis, F., Abdallah, C. and Dawson, D. *Control of robot manipulators*, MacMillan Publishing Co., New York, (1993).
- [3] Samson, C., Le Borgne, M. and Espinau, B. *Robot control*. Oxford University Press, (1991).
- [4] Spong, M. *Motion control of robot manipulator*. *Handbook Control*, W. Levine Editor, 1339–1350, (1996).
- [5] Khalil, W and Dombre, E. *Modeling identification and control of robots*. Hermes Penton Science, (2002).
- [6] Zhao-Hui Jiang; Ishida, T., "Trajectory tracking control of industrial robot manipulators using a neural network controller," *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, vol., no., pp.2390,2395, 7-10 Oct. (2007).
- [7] Khalal, O.; Mellit, A.; Rahim, M.; Salhi, H.; Guessoum, A., "Robust control of manipulator robot by using the variable structure control with sliding mode," *Control & Automation, 2007. MED '07. Mediterranean Conference on*, vol., no., pp.1,6, 27-29 June (2007).
- [8] Raafat, S.M.; Said, W.K.; Akmeliawati, R.; Tariq, N.M., "Improving trajectory tracking of a three-axis SCARA robot using neural networks," *Industrial Electronics & Applications, 2009. ISIEA 2009. IEEE Symposium on*, vol.1, no., pp.283,288, 4-6 Oct. (2009).
- [9] Benjanarasuth, T.; Sowanee, N.; Naksuk, N., "Two-degree-of-freedom simple servo adaptive control for SCARA robot," *Control Automation and Systems (ICCAS), 2010 International Conference on*, vol., no., pp.480,484, 27-30 Oct. (2010).
- [10] Suvilath, S.; Khongsomboun, K.; Benjanarasuth, T.; Kominet, N., "IMC-based PID controllers design for a two-links SCARA robot," *TENCON 2011 - 2011 IEEE Region 10 Conference*, vol., no., pp.1030,1034, 21-24 Nov. (2011).
- [11] Al-Khedher, M.A.; Alshamasin, M.S., "SCARA robot control using neural networks," *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on*, vol.1, no., pp.126,130, 12-14 June (2012).
- [12] Thanok, Sunphong, "Design of an Adaptive PD Controller with Dynamic Friction Compensation for Direct-Drive SCARA Robot." *Electrical Engineering Congress (iEECON), 2014 International. IEEE*, (2014).
- [13] Escobar, F., Díaz, S., Gutiérrez, C., Ledeneva, Y., Hernández, C., Rodríguez, D., & Lemus, R. Simulation of Control of a Scara Robot Actuated by Pneumatic Artificial Muscles Using RNAPM. *Journal of Applied Research and Technology*, 12(5), 939-946, (2014).
- [14] Maeda, Y.; Fujiwara, T.; Ito, H., "Robot control using high-dimensional neural networks," *SICE Annual Conference (SICE), 2014 Proceedings of the*, vol., no., pp.738,743, 9-12 Sept. (2014)
- [15] T. C. Lin, S. W. Chang, and C. H. Hsu, "Robust adaptive fuzzy sliding mode control for a class of uncertain discrete-time nonlinear systems", *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 347-359, (2012).
- [16] Rossomando F.G., Soria C. and Carelli R., "Sliding Mode Neuro Adaptive Control in Trajectory Tracking for Mobile Robots", *Journal of Intelligent and Robotic Systems*, ed.Springer Volume 74, Issue 3, Page 931-944 ISSN: 0921-0296, (2014).
- [17] Asada, H., and Slotine, J.J.E., *Robot Analysis and Control*, John Wiley & Sons, New York, (1986).



Francisco Guido Rossomando, nació en San Juan, Argentina en 1969. se graduó en ingeniería electrónica y magister en ingeniería de sistemas de control en la Universidad Nacional de San Juan (UNSJ), Argentina, en 1997 y 2002 respectivamente. Obtuvo el grado de Doctor en ingeniería eléctrica en la Universidad Federal de Espirito Santo, ES, Brasil, en 2006. Realizo su post-doctorado en la UNSJ en 2008, a través de una beca posdoctoral del Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET, Argentina); Fue director de Ciencia y Técnica, y Subsecretario de Promoción Científica del gobierno de la provincia de San Juan, actualmente es investigador adjunto de CONICET. Sus áreas de investigación son control de robots, control adaptable, identificación y control de procesos industriales.



Carlos Miguel Soria, nació en Tucumán, Argentina en el 27 de Noviembre de 1970. Se graduó como ingeniero eléctrico en la Facultad de Ciencias de la Universidad Nacional de Tucumán (UNT) in 1996. En 2000 se graduó como magister en Ingeniería de sistemas de Control en la facultad de Ingeniería de la Universidad Nacional de San Juan (UNSJ). Obtuvo el grado de doctor en Ingeniería en Sistemas de Control en la misma Universidad. Durante el periodo de 1997 a 2000 fue becario del programa FOMEC (Fondo para la Mejora en la Educación) para concluir los estudios de Maestría y durante el periodo 2001 a 2004 fue becario del Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET). Actualmente es Profesor en la Universidad Nacional de San Juan e Investigador Adjunto del Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET, Argentina).