Neural Comput & Applic
DOI 10.1007/s00521-016-2242-7

ORIGINAL ARTICLE

Discrete-time sliding mode neuro-adaptive controller for SCARA robot arm

F. G. Rossomando[1] · **C. M. Soria**[1]

Received: 11 August 2015 / Accepted: 16 February 2016

Abstract This work presents a discrete-time sliding mode neuro-adaptive control (DTSMNAC) method for robot manipulators. Due to the dynamics variations and uncertainties in the robot model, the trajectory tracking of robot manipulators has been one of the research areas for the last years. The proposed control structure is a practical design that combines a discrete-time neuro-adaptation technique with sliding mode control to compensate the dynamics variations in the robot. Using an online adaptation technique, a DTSMNAC controller is used to approximate the equivalent control in the neighborhood of the sliding surface. A sliding control is included to guarantee that the discrete-time neural sliding mode control can improve a stable closed-loop system for the trajectory tracking control of the robot with dynamics variations. The proposed technique simultaneously ensures the stability of the adaptation of the neural networks and can be obtained a suitable equivalent control when the parameters of the robot dynamics are unknown in advance. This neural adaptive system is applied to a SCARA robot manipulator and shows to be able to ensure that the output tracking error will converge to zero. Finally, experiments on a SCARA robot have been developed to show the performance of the proposed technique, including the comparison with a PID controller.

Keywords MIMO system · Neural networks · Nonlinear control · Adaptive control · SCARA robot

✉ F. G. Rossomando
frosoma@inaut.unsj.edu.ar

[1] INAUT, UNSJ-CONICET, Capital, San Juan, Argentina

1 Introduction

The control of robot manipulators has been a research area for years and has developed various control strategies [1–3]. Due to the robot manipulators being composed of several joints bonded together, the joints have highly nonlinear dynamics with a strong link between them. This complicates the control task, especially with model uncertainties or external disturbances.

Some techniques have been proposed with control systems that take the model into account, as a computed torque control [4, 5]. The work provides satisfactory results in terms of tracking errors and robustness. The uncertainties in the model due to bad estimates or model parameters are difficult to design an efficient algorithm based on a precise mathematical model.

The work of Jiang and Ishida [6] proposes a dynamic trajectory tracking control of industrial robot manipulators using a PD controller and a neural network controller. The neural network is a three-layer feed-forward network. The learning law of neural network weights was derived using a simplified dynamic model of the robot and a back-propagation theory. With this system, trajectory tracking control simulations and experiments were carried out using an industrial manipulator AdeptOne XL robot. The results of their work have shown the effectiveness and usefulness of the proposed control method, and it was shown that the learning effect of the neural network affects the trajectory tracking accuracy.

In [7], the problem of designing robust variable structure control and sliding mode planes was considered for the robot manipulator SCARA RP41. The simulation results show the robustness of the extension variable structure control, a sliding mode opposite the transported load, parametric variations, an imprecise model, and external perturbation signals.

Published online: 24 March 2016

On the other hand [8], an ANN-based robust adaptive tracking control scheme for SCARA robot was implemented. The ANNs are not directly used to adapt the system uncertainties, but they are used to adjust the bounds of dynamics variations in a compact set. The output signals of the ANNs then adjust the gain of the sliding mode controller so that the undesired effects of system variations can be eliminated. And the output tracking error between the robot output and the desired reference signal can asymptotically converge to zero. The system performance was demonstrated in a simulation model.

The work of Benjanarasuth et al. [9] showed via simulations that a DDR-type SCARA robot can be successfully controlled by the two-degree-of-freedom simple adaptive control. The control system structure is based on a linear model, and its implementation is relatively simple. The joint's angle can track the reference path properly, and the effect of constant input disturbance can be suppressed by means of the disturbance rejection property being altered independently by adjusting the integral gain.

In [10], a PID controller design based on Internal Model Control (IMC) for a two-link SCARA robot is presented. The suitable IMC low-pass filter is proposed so that the PID controller can be derived by applying *Maclaurin* series expansion to the IMC controller in a general feedback control loop. The simulation results showed in this work confirm that the PID controllers designed by the proposed method can control the angular positions of the SCARA robot precisely without steady-state error. And the study of [11] shows a comprehensive modeling and identification of an industrial SCARA robot developed to include servo actuator dynamics. The kinematics model of the manipulator was studied. The authors used *Lagrangian* mechanics to derive equations of motion. Conventional PD controller is compared to a neural network controller to achieve precise control of position and motion characteristics. Each joint is treated individually to reach optimal positioning of the end effector. The neural network model is trained to achieve accurate positioning and minimize joint displacements. This work shows simulation results that verify the proposed control method; however, it does not provide a stability analysis.

The work of Thanok [12] presents two results derived from experimentation. In the first scheme, the friction model that contains the Coulomb and viscous effect is static. However, the dynamics model is known, but the value of its parameter is not; therefore, the initial is set to zero. The scheme is identified online during trajectory tracking control. The second scheme used a dynamic friction *LuGre* model, which is a known dynamic model, but has an unknown parameter and its initial set to zero. The experiment results demonstrate that parameters in the adaptive PD controller using dynamic friction can decrease errors at a steadier state than parameters in a PD controller using static friction.

In another work, Escobar et al. [13] describe the simulation of movement control of a one-degree-of-freedom articulated SCARA robot arm actuated by a pair of pneumatic actuators. The pneumatic actuator emulates the behavior of biological muscles; due to its nonlinear behavior, was needed to implement control strategies for robot arms using this type of actuator. In this work, a PID controller is used to the linear transfer function and generates the necessary information to train the multilayer perceptron artificial neural network (RNAPM). The simulation results show that the RNAPM has proved to outperform the PID control's response time, minimize the angular error, and avoid the oscillation problem due to its continuous, constant behavior.

The study of [14] describes a position control scheme for robots using high-dimensional neural networks which learn inverse kinematics. A complex-valued neural network and quaternion neural network using the simultaneous perturbation optimization method are used to control the SCARA robot and the three-dimensional robot whose characteristics are unknown. In the mentioned work, experiment results demonstrate the effectiveness of the proposed control method but do not provide a stability analysis.

The work of Lin et al. [15] has proposed a discrete-time robust adaptive fuzzy strategy to design the sliding mode controller. This method could be used for nonlinear systems, but do not present applications on robotics systems.

Meanwhile, the study of [16] presented a neural adaptive-based sliding mode controller for nonlinear systems, which was applied to trajectory tracking for mobile robots. The results shown are based on experiments and use a neural method to adapt the dynamic model to be controlled. In addition, the complete control system is designed in the continuous domain.

In this paper, the design of an adaptive trajectory tracking controller based on a nominal robot dynamics and neural controller is developed on a SCARA robot Bosch SR-800. The control system is designed with a neural sliding mode dynamic control in the discrete-time domain. The dynamic neuro-controller is designed based on the "discrete-time sliding mode neuro-adaptive control" (DTSMNAC), where an online adaptation law is used to adjust the weights of the radial basis functions (RBF). Such law is conditioned by the sliding surface which has been specified. The dynamic neuro-controller uses a neural network based on the RBF functions, which is the main controller in charge of the inverse dynamics of the SCARA robot, where the compensation by sliding surface is designed to delete the approximation error introduced by the neuronal controller. Furthermore, the adaptation laws

of the control system are obtained from the Lyapunov stability criterion. Therefore, the control system stability can be ensured, which results as an asymptotic stability property. The proposal of a "discrete-time sliding mode neuro-adaptive control"-based controller has the following advantages:

1. This control technique can be applied to a nonlinear MIMO system, which is the case of the SCARA robot manipulator and other nonlinear dynamics.
2. The complete analysis was done in discrete time using the Lyapunov's discrete method.
3. It can control most of the robot manipulator systems without knowing their exact mathematical models.
4. The main advantage of this DTSMNAC technique over the model-based ones is that it does not require previous knowledge of the robot dynamics; moreover, it can be tuned online adjusting the weights.
5. The proposed control scheme adjusts the main part of the robot dynamics effects, being a robust system. Besides, the controller integrates the PI control with the DTSMNAC. With this control technique, the "chattering" effect can be reduced to small values.

This work is organized as in the following way: Section II presents an overview of the system and shows the mathematical representation of the SCARA robot dynamics. The neural adaptive RBF compensator and its stability analysis are studied in Sects. 3, 4, and 5. Experimental results are shown in Sect. 6, showing the efficiency of the controllers. Finally, the conclusions are shown in Sect. 7.

2 SCARA robot model

2.1 Review of the robot manipulator system

Taking into account that DTSMNAC controller will command the robot on the horizontal motion plane, the dynamic model for the first two joints of the robot (Fig. 1) is considered.

The dynamic model in the joint space for a robot manipulator with n degrees of freedom was showed in [17]. In this case, $n = 2$:

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

where $\mathbf{q}$ is the vector (2×1) of generalized coordinates (joint positions), $\mathbf{q} = [q_1, q_2]^T$, $\mathbf{M}(\mathbf{q})$ is a (2×2) matrix, usually referred to as manipulator mass matrix containing the kinetic energy functions of the manipulator. $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$: (2×2) matrix represents torques arising from centrifugal and Coriolis forces. $\mathbf{f}(\dot{\mathbf{q}})$: (2×1) represents viscosity friction effects when the manipulator is moving in its work space. $\boldsymbol{\tau}$: Vector of joint actuator torques (2×1).

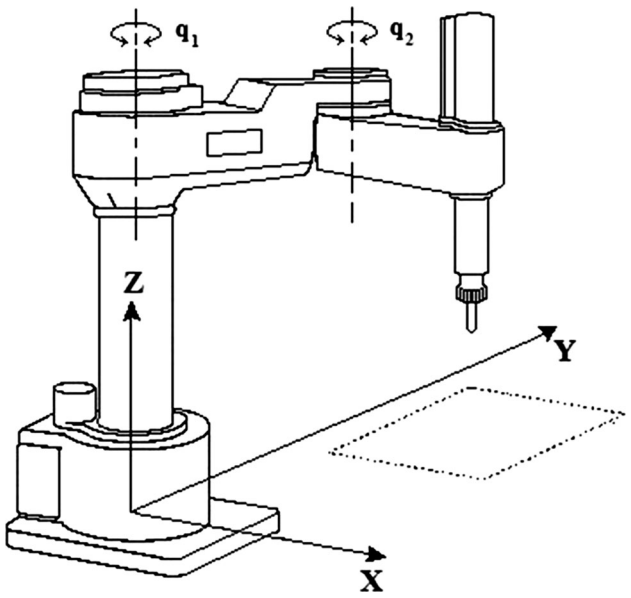

Fig. 1 SCARA robot manipulator

The parameter of the robot manipulator Bosch SR-800 is as follows:

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

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

$$\hat{\mathbf{f}}(\dot{q}) = \begin{bmatrix} 1.0256 sign(\dot{q}_1) \\ 1.7842 sign(\dot{q}_2) \end{bmatrix} \tag{2}$$

The nonlinear dynamic model of a manipulator (1) is used to compute the control torque inputs,

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

where the quantities $\hat{\mathbf{M}}(\mathbf{q}), \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}}), \hat{\mathbf{f}}(\dot{\mathbf{q}})$ are estimates of the true parameters and $\boldsymbol{\Delta}(\mathbf{q})$ is the unknown time-dependent uncertainties. From (2), $M(q)$ is a positive definite matrix, perhaps it can be demonstrated that $det(M(q)) \neq 0$, also is an invertible matrix.

Assumption 1 The uncertainty function vector $\boldsymbol{\Delta}(\mathbf{q})$ is bounded by a constant $\|\boldsymbol{\Delta}(\mathbf{q})\| \leq \Delta_{\text{Max}}$.

A direct discretization with $\dot{q}_i = \frac{q_i(k) - q_i(k-1)}{T_o}$ and $\ddot{q}_i = \frac{q_i(k+1) - 2q_i(k) + q_i(k-1)}{T_o^2}$ to obtain a discrete-time dynamic model was used, where $T_0 = 1$ ms is the sampling time and k is the discrete-time index.

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

where the matrix parameters are defined as:

$$\begin{aligned} \boldsymbol{\beta}(\mathbf{q}(k), \mathbf{q}(k-1)) &= -2\mathbf{I} + \hat{\mathbf{M}}^{-1}(\mathbf{q}(k))\hat{\mathbf{C}}(\mathbf{q}(k), \quad \mathbf{q}(k-1))T_0 \\ \boldsymbol{\alpha}(\mathbf{q}(k), \mathbf{q}(k-1)) &= \mathbf{I} - \hat{\mathbf{M}}^{-1}(\mathbf{q}(k))\hat{\mathbf{C}}(\mathbf{q}(k), \quad \mathbf{q}(k-1))T_0 \\ \boldsymbol{\psi}(\mathbf{q}(k)) &= \hat{\mathbf{M}}^{-1}(\mathbf{q}(k))T_0^2 \\ \boldsymbol{\delta}(k) &= \boldsymbol{\psi}(\mathbf{q}(k))\boldsymbol{\Delta}(k) \end{aligned} \tag{5}$$

Rearranging (4), it does obtain:

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

where:

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

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

3 Dynamic controller

The dynamic controller receives the difference between desired references and angular output positions (q_1 and q_2), which are sent to the robot servos, as it is shown in Fig. 2. Now, this error vector of output angular positions is defined in (12)

The robot dynamics system (6) can be written in the following form

$$\begin{aligned} \mathbf{q}(k+1) &= \mathbf{F}(\mathbf{q}(k)) + \mathbf{G}(\mathbf{q}(k))\boldsymbol{\tau}(k) + \boldsymbol{\delta}(k) \\ &= \begin{pmatrix} \mathbf{F}_1(\mathbf{q}(k)) \\ \mathbf{F}_2(\mathbf{q}(k)) \end{pmatrix} + \begin{pmatrix} \mathbf{G}_1(\mathbf{q}(k)) \\ \mathbf{G}_2(\mathbf{q}(k)) \end{pmatrix} \boldsymbol{\tau}(k) + \begin{pmatrix} \delta_1(k) \\ \delta_2(k) \end{pmatrix} \end{aligned} \tag{9}$$

In order for (6) to be controllable, it is required that $g \neq 0$, and without the loss of generality, it is assumed that the robot dynamics represent a bounded input bounded output (BIBO) system; G(k) is a positive definite positive matrix ($M(q)T_0^{-2}$), and $\delta(k)$ is the unknown time-dependent uncertainties vector, with δ_{Max} being its upper bound,

$$\delta_{\text{Max}} = \sup_{t \in \Re^+} |\delta(k)| \tag{10}$$

also,

$$\begin{aligned} \mathbf{q}(k) &= (q_1(k) \quad q_2(k))^T \\ \boldsymbol{\tau}(k) &= (\tau_1(k) \quad \tau_2(k))^T \\ \boldsymbol{\delta}(k) &= (\delta_1(k) \quad \delta_2(k))^T \end{aligned} \tag{11}$$

are the state variables' output vector, input vector, and uncertainty parameter vectors, respectively, and the state tracking error is defined as:

$$e(k) = q(k) - q_{ref}(k) = \left(q_1(k) - q_{1ref}(k), q_2 - q_{2ref}(k)\right)^T \tag{12}$$

4 Discrete-time sliding mode control

The main objective was to implement an adaptive neural controller which guarantees the boundedness of all variables for the closed-loop system and tracking of a given bounded reference signal $q_{ref}(k)$.

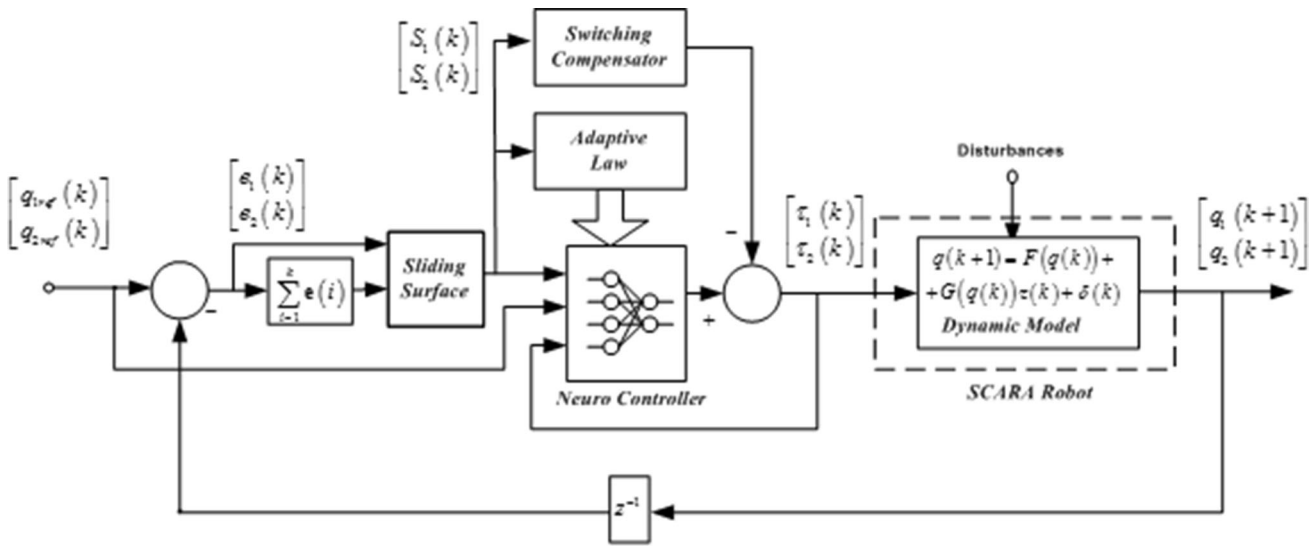

Fig. 2 Control system structure including the DTSMNAC and the kinematic controllers

The neural feedback linearization method which is based on NN-RBF model can solve this kind of control problem [12, 13].

The state tracking error is defined as $e(k) = q(k) - q_{ref}(k)$. In this work, the control objective was to find a control action such that the state q of the closed-loop system will follow the desired state q_{ref}; in other words, the tracking error should converge to zero. The goal of the control law obtained by sliding mode technique was to track the trajectory of the system to a defined surface (calculated by the designer) in the state space and maintain it on the entire surface for all subsequent time.

A sliding surface for MIMO system can be defined in the error state $\boldsymbol{S}(k)$ from (14).

$$S(k) = \begin{pmatrix} (t_d + \lambda_1 z^{-1}) & 0 \\ 0 & (t_d + \lambda_2 z^{-1}) \end{pmatrix}$$

$$\sum_{i=1}^{k} e(i)T_0 = \begin{pmatrix} e_1(k) + \lambda_1 \sum_{i=1}^{k-1} e_1(i)T_0 \\ e_2(k) + \lambda_2 \sum_{i=1}^{k-1} e_2(i)T_0 \end{pmatrix} \tag{13}$$

where T_0 is the sampling time and $t_d = (1 - z^{-1})/T_0$.

The discrete difference in sliding surface $\boldsymbol{S}(k)$ is:

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

where λ_i is a strictly positive constant. Let us define $\boldsymbol{\rho} = diag(\lambda_1 T_0 - 1,\ \lambda_2 T_0 - 1)$.

In designing the sliding mode control system, first it is defined as the ideal equivalent control law $\boldsymbol{\tau}^*$, which determines the dynamic of the system on the sliding surface. The ideal equivalent control law is derived by recognizing

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

Substituting (14) into (15), is obtained

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

Now, let us consider the problem of controlling the uncertain nonlinear system (6) as treated in [17]. Defining a control law $\boldsymbol{\tau}^*$ that guarantees the sliding condition of (16), which is composed of an equivalent control,

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

A condition to ensure that the trajectory of the error vector $\boldsymbol{e}(k)$ will evolve from the initial phase to the sliding phase is to select the control strategy such that:

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

where $\boldsymbol{Kd} = [K_{d1}\ 0;\ 0\ K_{d2}]^{\mathrm{T}}$, where K_{d1} and K_{d2} are small positive real numbers, $\boldsymbol{\eta} = [\eta_1\ 0;\ 0\ \eta_2]^{\mathrm{T}}$, and the function sign is defined by:

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

From (16)

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

From (20), the ideal equivalent control law $\boldsymbol{\tau}^*$

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

is obtained.

5 Neural system adjustment laws

In real systems, $F(q(k))$, $G(q(k))$, $\delta(k)$, and η may be unknown, and the function sign(S) is not continuous. Thus, it is impossible to generate the control law (21). To overcome these difficulties, it is used a neural system for estimating $\hat{F}\big(\mathbf{q}(k), \boldsymbol{\theta}_F^*\big)$, $\hat{G}\big(\mathbf{q}(k), \boldsymbol{\theta}_G^*\big)$ and for $\hat{\delta}\big(\mathbf{S}(k), \boldsymbol{\theta}_\delta^*\big)$ to approximate, respectively, $F(q(k))$, $G(q(k))$, $\delta(k)$.

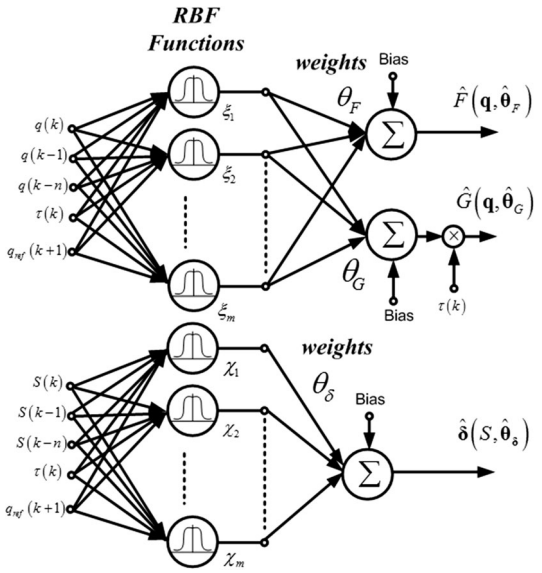

Fig. 3 Radial basis function network

The optimal parameter vectors are defined as:

$$\begin{cases} \boldsymbol{\theta}_F^* = \arg\min\limits_{\theta_F \in \Omega_\theta} \left\{ \sup\limits_{q \in \Omega_q} \left| F(\mathbf{q}(k)) - F\left(\mathbf{q}(k)|\boldsymbol{\theta}_G\right) \right| \right\} \\ \boldsymbol{\theta}_G^* = \arg\min\limits_{\theta_G \in \Omega_\theta} \left\{ \sup\limits_{q \in \Omega_q} \left| G(\mathbf{q}(k)) - G\left(\mathbf{q}(k)|\boldsymbol{\theta}_G\right) \right| \right\} \\ \boldsymbol{\theta}_\delta^* = \arg\min\limits_{\theta_\delta \in \Omega_\theta} \left\{ \sup\limits_{q \in \Omega_q} \left| \delta(k) - \delta\left(k|\boldsymbol{\theta}_\delta\right) \right| \right\} \end{cases} \tag{22}$$

defining $\Omega_\theta = \{\theta / \theta \|\theta_f\| \le M_f \wedge \|\theta_g\| \le M_g \wedge \|\theta_{\delta\text{-}} \|\theta_f\| \le M_f \wedge \|\theta_g\| \le M_g \wedge \|\theta_\delta\| \le M_\delta . \|\theta_f\| \le M_f \wedge \|\theta_{g\text{-}} \|\theta_g\| \le M_g \wedge \|\theta_\delta\| \le M_\delta . \|\theta_f\| \le M_f \wedge \|\theta_g\| \le M_g \wedge \|\theta_\delta \| \le M_\delta\}$ and $\Omega_q = \{q / qq \le M_q . q \le M_q\}$ where M_F, M_G, M_δ, and M_q are positive constants, and Ω_θ and Ω_q are compact sets of suitable bounds on $\boldsymbol{\theta}_F \boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_\delta$ and $\mathbf{q}$, respectively.

The Gaussian function is used as the activation function of each neuron in the hidden layer (23).

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

where i is the ith neuron of the hidden layer, c_i is the central position of the ith neuron, and σ_i is the spread factor of the Gaussian function, and the regressors of the Gaussian function are $\zeta(k) = [q(k),\ q(k-1), \ldots\ q(k-n),\ \tau(k),\ q_{ref}(k+1)]^{\mathrm{T}}$ and $\upsilon(k) = [S(k), S(k-1), \ldots S(k-n),\ \tau(k),\ q_{ref}(k+1)]^{\mathrm{T}}$.

The structure of RBF-NN is shown in Fig. 3.

The control $\boldsymbol{\tau}(k)$ can be approximated by a RBF-NN through online learning,

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

It was added a robust control action $\boldsymbol{\tau}_\Delta(k)$ to attenuate the external disturbance, which may be defined as (Δ sign(S)). The variables $\boldsymbol{\theta}_F \boldsymbol{\theta}_G$, and $\boldsymbol{\theta}_\delta$ are neural weights of the approximating adaptive system $\hat{\mathbf{F}}\left(\mathbf{q}(k), \hat{\theta}_F\right)$, $\hat{\mathbf{G}}\left(\mathbf{q}(k), \hat{\theta}_G\right)$, $\hat{\delta}\left(\mathbf{S}(k), \hat{\theta}_\delta\right)$, respectively, and can be expressed by:

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

It was added a robust control action $\boldsymbol{\tau}_\Delta(k)$ to attenuate the external disturbance, which may be defined as (Δ sign(S)). The variables $\boldsymbol{\theta}_F \boldsymbol{\theta}_G$, and $\boldsymbol{\theta}_\delta$ are neural weights of the approximating adaptive system $\hat{\mathbf{F}}\left(\mathbf{q}(k), \hat{\theta}_F\right)$, $\hat{\mathbf{G}}\left(\mathbf{q}(k), \hat{\theta}_G\right)$, $\hat{\delta}\left(\mathbf{S}(k), \hat{\theta}_\delta\right)$, respectively, and can be expressed by:

$$\hat{\mathbf{F}}\left(\mathbf{q}(k), \hat{\theta}_F\right) = \hat{\theta}_F^T \hat{\xi}(\boldsymbol{\zeta}(k)))$$
$$= \begin{pmatrix} \sum\limits_{i=1}^{m} \hat{\theta}_{F1i}^T \exp\left(-\left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Fi}\right)^T \left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Fi}\right) \Big/ \sigma_{Fi}^2 \right) \\ \sum\limits_{i=1}^{m} \hat{\theta}_{F2i}^T \exp\left(-\left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Fi}\right)^T \left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Fi}\right) \Big/ \sigma_{Fi}^2 \right) \end{pmatrix} \tag{26}$$

and

$$\hat{\mathbf{G}}\left(\mathbf{q}(k), \hat{\theta}_G\right) = \hat{\theta}_G^T \hat{\xi}(\boldsymbol{\zeta}(k))$$
$$= \begin{pmatrix} \sum\limits_{i=1}^{m} \hat{\theta}_{G1i}^T \exp\left(-\left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Fi}\right)^T \left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Gi}\right) \Big/ \sigma_{Gi}^2 \right) \\ \sum\limits_{i=1}^{m} \hat{\theta}_{G2i}^T \exp\left(-\left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Gi}\right)^T \left(\boldsymbol{\zeta}(k) - \mathbf{c}_{Gi}\right) \Big/ \sigma_{Gi}^2 \right) \end{pmatrix} \tag{27}$$

where $\hat{\theta}_{F1,2}^T$ $(m = 5)$ and $\hat{\theta}_{G1,2}^T \in \mathbb{R}^{1xm}$.

Another neural net control term is employed in order to attenuate the external disturbances. The MIMO control term is in the form of:

$$\hat{\delta}\left(\mathbf{S}(k),\hat{\theta}_{\delta}\right)=\hat{\theta}_{\delta}^{T}\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))$$
$$=\begin{pmatrix}\sum_{i=1}^{m}\hat{\theta}_{\delta 1}^{T}\exp\left(-\left(\boldsymbol{\upsilon}(k)-\mathbf{c}_{\delta i}\right)^{T}\left(\boldsymbol{\upsilon}(k)-\mathbf{c}_{\delta i}\right)\Big/\sigma_{\delta i}^{2}\right)\\ \sum_{i=1}^{m}\hat{\theta}_{\delta 2}^{T}\exp\left(-\left(\boldsymbol{\upsilon}(k)-\mathbf{c}_{\delta i}\right)^{T}\left(\boldsymbol{\upsilon}(k)-\mathbf{c}_{\delta i}\right)\Big/\sigma_{\delta i}^{2}\right)\end{pmatrix}$$
$$=\begin{pmatrix}\hat{\theta}_{\delta 1}^{T}\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\\ \hat{\theta}_{\delta 2}^{T}\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\end{pmatrix}\tag{28}$$

where $\hat{\theta}_{\delta 1}^{T}$ and $\hat{\theta}_{\delta 2}^{T}$ are adjustable parameters, and the function vector $\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))$ is a function of the sliding surface.

The global control law is given by (25), and it is defined as the minimum approximation error as:

$$\boldsymbol{\varepsilon}(k)=\mathbf{F}(\mathbf{q})-\hat{\mathbf{F}}\left(\mathbf{q}(k),\boldsymbol{\theta}_{F}\right)+\left[\mathbf{G}(\mathbf{q}(k))-\mathbf{G}\left(\mathbf{q}(k),\boldsymbol{\theta}_{G}\right)\right]\boldsymbol{\tau}+\boldsymbol{\delta}(k)-\hat{\delta}\left(\mathbf{S}(k),\hat{\theta}_{\delta}^{*}\right)\tag{29}$$

Now, using (16) and considering the robot dynamic model (6), it can be written as:

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

Replacing the proposed control action in (25) in (30)

$$\begin{aligned}\Delta\mathbf{S}(k+1)=&\rho\mathbf{e}(k)+\left(\mathbf{F}(\mathbf{q}(k))-\hat{\mathbf{F}}\left(\mathbf{q}(k),\hat{\boldsymbol{\theta}}_{f}\right)\right)+\mathbf{q}_{ref}(k+1)+\left(\mathbf{G}-\hat{\mathbf{G}}\left(\mathbf{q}(k),\hat{\boldsymbol{\theta}}_{g}\right)\right)\boldsymbol{\tau}(k)\\&-\rho\mathbf{e}(k)+\left[\boldsymbol{\delta}(k)-\hat{\delta}\left(\mathbf{S}(k),\hat{\boldsymbol{\theta}}_{\delta}\right)\right]-\mathbf{q}_{ref}(k+1)\\&-\mathbf{K}_{d}\mathbf{S}(k)T_{0}-\boldsymbol{\eta}T_{0}sign(\mathbf{S}(k))+\boldsymbol{\tau}_{\Delta}\\=&\left[\hat{\mathbf{F}}\left(\mathbf{q}(k),\boldsymbol{\theta}_{f}^{*}\right)-\hat{\mathbf{F}}\left(\mathbf{q}(k),\hat{\boldsymbol{\theta}}_{f}\right)\right]+\left(\hat{\mathbf{G}}\left(\boldsymbol{\theta}_{g}^{*}\right)-\hat{\mathbf{G}}\left(\hat{\boldsymbol{\theta}}_{g}\right)\right)\boldsymbol{\tau}(k)\\&+\left[\hat{\delta}\left(\mathbf{S}(k),\hat{\boldsymbol{\theta}}_{\delta}^{*}\right)-\hat{\delta}\left(\mathbf{S}(k),\hat{\boldsymbol{\theta}}_{\delta}\right)\right]+\boldsymbol{\varepsilon}-\mathbf{K}_{d}\mathbf{S}(k)T_{0}-\boldsymbol{\eta}T_{0}sign(\mathbf{S}(k))+\boldsymbol{\tau}_{\Delta}\end{aligned}\tag{31}$$

Considering that

$$\hat{\mathbf{F}}\left(\mathbf{q}(k),\boldsymbol{\theta}_{F}^{*}\right)-\hat{\mathbf{F}}\left(\mathbf{q}(k),\hat{\boldsymbol{\theta}}_{F}\right)=\boldsymbol{\theta}_{F}^{*T}\xi(\boldsymbol{\zeta}(k))-\hat{\boldsymbol{\theta}}_{F}^{T}\xi(\boldsymbol{\zeta}(k))=\tilde{\boldsymbol{\theta}}_{F}^{T}\xi(\boldsymbol{\zeta}(k))\tag{32}$$

$$\hat{\mathbf{G}}\left(\mathbf{q}(k),\boldsymbol{\theta}_{G}^{*}\right)-\hat{\mathbf{G}}\left(\mathbf{q}(k),\hat{\boldsymbol{\theta}}_{G}\right)=\boldsymbol{\theta}_{G}^{*T}\xi(\boldsymbol{\zeta}(k))-\hat{\boldsymbol{\theta}}_{G}^{T}\xi(\boldsymbol{\zeta}(k))=\tilde{\boldsymbol{\theta}}_{F}^{T}\xi(\boldsymbol{\zeta}(k))\tag{33}$$

$$\hat{\delta}\left(\mathbf{S}(k),\boldsymbol{\theta}_{\delta}^{*}\right)-\hat{\delta}\left(\mathbf{S}(k),\hat{\boldsymbol{\theta}}_{\delta}\right)=\left(\boldsymbol{\theta}_{\delta}^{*T}-\hat{\boldsymbol{\theta}}_{\delta}^{T}\right)\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))=\tilde{\boldsymbol{\theta}}_{\delta}^{T}\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\tag{34}$$

where $\tilde{\boldsymbol{\theta}}_{F}$, $\tilde{\boldsymbol{\theta}}_{G}$ and $\tilde{\boldsymbol{\theta}}_{\delta}$ are defined as

$$\tilde{\boldsymbol{\theta}}_{F}^{T}=\boldsymbol{\theta}_{F}^{*T}-\hat{\boldsymbol{\theta}}_{F}^{T}\tag{35}$$

$$\tilde{\boldsymbol{\theta}}_{G}^{T}=\boldsymbol{\theta}_{G}^{*T}-\hat{\boldsymbol{\theta}}_{G}^{T}\tag{36}$$

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

Being $\boldsymbol{\theta}_{F}^{*}$, $\boldsymbol{\theta}_{G}^{*}$, and $\boldsymbol{\theta}_{\delta}^{*}$ are optimal constant values, $\Delta\mathbf{S}$ can be approximated by

$$\Delta\mathbf{S}(k+1)=-\mathbf{K}_{d}T_{0}\mathbf{S}(k)+\tilde{\boldsymbol{\theta}}_{G}^{T}\xi(\boldsymbol{\zeta}(k))\boldsymbol{\tau}+\tilde{\boldsymbol{\theta}}_{F}^{T}\xi(\boldsymbol{\zeta}(k))+\tilde{\boldsymbol{\theta}}_{\delta}^{T}\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))+\boldsymbol{\varepsilon}-\boldsymbol{\eta}T_{0}sign(\mathbf{S}(k))+\boldsymbol{\tau}_{\Delta}\tag{38}$$

Remark 1 The uncertain ε is assumed to be bounded by $\|\boldsymbol{\varepsilon}\|\leq\varepsilon_{Max}$.

Remark 2 The constant $\varepsilon_{\mathrm{Max}}$ is equal to $\|\boldsymbol{\eta}\||T_{0}|$.

Theorem *Consider the uncertain nonlinear system defined by (6). Then, the controller proposed by (25) ensures the convergence of tracking error to zero, when using the following parameters adaptation laws:*

$$\Delta\tilde{\boldsymbol{\theta}}_{Fi}=-\gamma_{1}S_{\mathrm{i}}\xi(\boldsymbol{\zeta}(k))\tag{39}$$

$$\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}=-\gamma_{2}S_{\mathrm{i}}\xi(\boldsymbol{\zeta}(k))\boldsymbol{\tau}_{\mathrm{i}}\tag{40}$$

$$\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}=-\gamma_{3}S_{\mathrm{i}}\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\tag{41}$$

Proof Let a positive definite Lyapunov function candidate $V(k)$ be defined as

$$V(k)=\frac{1}{2}\sum_{\mathrm{i}=1}^{2}\left[S_{\mathrm{i}}^{2}(k)+\gamma_{1}^{-1}\left(\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k-1)\right)+\gamma_{2}^{-1}\left(\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k-1)\right)+\gamma_{3}^{-1}\left(\tilde{\boldsymbol{\theta}}_{\delta i}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k-1)\right)\right]\tag{42}$$

Now, doing the discrete difference in $V(k)$

$$\begin{aligned}\Delta V(k)=\sum_{i=1}^{2}\Bigg[&\left(S_{\mathrm{i}}^{2}(k+1)-S_{\mathrm{i}}^{2}(k)\right)+\gamma_{1}^{-1}\left(\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^{T}(k)\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)-\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k-1)\right)\\&+\gamma_{2}^{-1}\left(\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^{T}(k)\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k)-\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k-1)\right)\\&+\gamma_{3}^{-1}\left(\tilde{\boldsymbol{\theta}}_{\delta i}^{T}(k)\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k)-\tilde{\boldsymbol{\theta}}_{\delta i}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k-1)\right)\Bigg]\end{aligned}\tag{43}$$

Defining $\Delta\boldsymbol{\theta}_{Fi}$, $\Delta\boldsymbol{\theta}_{Gi}$, and $\Delta\boldsymbol{\theta}_{\delta i}$ as:

$$\begin{aligned}\Delta\boldsymbol{\theta}_{Fi}&=\gamma_{1}^{-1}\left(\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^{T}(k)\tilde{\boldsymbol{\theta}}_{Fi}(k)-\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{Fi}(k-1)\right)\\\Delta\boldsymbol{\theta}_{Gi}&=\gamma_{2}^{-1}\left(\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^{T}(k)\tilde{\boldsymbol{\theta}}_{Gi}(k)-\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{Gi}(k-1)\right)\\\Delta\boldsymbol{\theta}_{\delta i}&=\gamma_{3}^{-1}\left(\tilde{\boldsymbol{\theta}}_{\delta i}^{T}(k)\tilde{\boldsymbol{\theta}}_{\delta i}(k)-\tilde{\boldsymbol{\theta}}_{\delta i}^{T}(k-1)\tilde{\boldsymbol{\theta}}_{\delta i}(k-1)\right)\end{aligned}\tag{44}$$

and rearranging Eq. (43).

$$\begin{aligned}\Delta V &= \sum_{i=1}^{2}\left[\left(S_{\mathrm{i}}^2(k+1)-S_{\mathrm{i}}^2(k)\right)+\Delta\boldsymbol{\theta}_{F\mathrm{i}}+\Delta\boldsymbol{\theta}_{G\mathrm{i}}+\Delta\boldsymbol{\theta}_{\delta\mathrm{i}}\right]\\ &= \sum_{i=1}^{2}\left[\left(\left(S_{\mathrm{i}}(k)+\Delta S_{\mathrm{i}}(k+1)\right)^2-S_{\mathrm{i}}^2(k)\right)\right.\\ &\quad\left.+\Delta\boldsymbol{\theta}_{F\mathrm{i}}+\Delta\boldsymbol{\theta}_{G\mathrm{i}}+\Delta\boldsymbol{\theta}_{\delta\mathrm{i}}\right]\\ &= \sum_{i=1}^{2}\left[\left(2S_{\mathrm{i}}(k)\Delta S_{\mathrm{i}}(k+1)+\Delta S_{\mathrm{i}}^2(k+1)\right)\right.\\ &\quad\left.+\Delta\boldsymbol{\theta}_{F\mathrm{i}}+\Delta\boldsymbol{\theta}_{G\mathrm{i}}+\Delta\boldsymbol{\theta}_{\delta\mathrm{i}}\right]\end{aligned} \tag{45}$$

Replacing (38) in (45)

$$\begin{aligned}\Delta V = \sum_{i=1}^{2}\left[2\left(-k_{di}T_0S_{\mathrm{i}}^2(k)+S_{\mathrm{i}}(k)\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^T\boldsymbol{\xi}(\boldsymbol{\zeta}(k))\tau_{\mathrm{i}}\right.\right.\\ +S_{\mathrm{i}}\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^T\boldsymbol{\xi}(\boldsymbol{\zeta}(k))+S_{\mathrm{i}}\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\\ +S_{\mathrm{i}}\varepsilon_{\mathrm{i}}-\boldsymbol{\eta}_{\mathrm{i}}T_0S_{\mathrm{i}}sign(S_{\mathrm{i}}(k)))+\Delta S_{\mathrm{i}}^2(k+1)\\ \left.\left.+\Delta\boldsymbol{\theta}_{F\mathrm{i}}+\Delta\boldsymbol{\theta}_{G\mathrm{i}}+\Delta\boldsymbol{\theta}_{\delta\mathrm{i}}{}_{G\mathrm{i}}^T\right]\right.\end{aligned} \tag{46}$$

From (44), rearranging $\Delta\boldsymbol{\theta}_{Fi}$ as:

$$\begin{aligned}\Delta\boldsymbol{\theta}_{F\mathrm{i}} &= \gamma_1^{-1}\left(\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^T(k)\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)-\left[\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)-\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)\right]^T\right.\\ &\quad\left.\times\left[\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)-\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)\right]\right)\\ &= 2\gamma_1^{-1}\left(\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)\right)-\left(\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)\right)\end{aligned} \tag{47}$$

Making the same analysis for $\Delta\boldsymbol{\theta}_{Gi}$ and $\Delta\boldsymbol{\theta}_{\delta i}$, these variables can be expressed as:

$$\Delta\boldsymbol{\theta}_{G\mathrm{i}} = 2\gamma_2^{-1}\left(\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k)\right)-\left(\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k)\right) \tag{48}$$

$$\Delta\boldsymbol{\theta}_{\delta\mathrm{i}} = 2\gamma_3^{-1}\left(\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k)\right)-\left(\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k)\right) \tag{49}$$

Replacing (39), (40), and (41) in (46) and rearranging

$$\begin{aligned}\Delta V(k) = \sum_{i=1}^{2}\left[-2k_{di}T_0S_{\mathrm{i}}^2(k)+\Delta S_{\mathrm{i}}^2(k+1)+2S_i\boldsymbol{\varepsilon}\right.\\ +2\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^T(k)\left(S_i\boldsymbol{\xi}(\boldsymbol{\zeta}(k))+2\gamma_1^{-1}\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)\right)\\ -2\gamma_1^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)\right)\\ +2\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^T(k)\left(S_{\mathrm{i}}(k)\boldsymbol{\xi}(\boldsymbol{\zeta}(k))\tau_{\mathrm{i}}+\gamma_2^{-1}\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k)\right)\\ -2\gamma_2^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k)\right)+2\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T(k)(S_{\mathrm{i}}\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\\ +\gamma_3^{-1}\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k))-2\gamma_3^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k)\right)\\ \left.-2\boldsymbol{\eta}_{\mathrm{i}}T_0S_{\mathrm{i}}sign(S_{\mathrm{i}}(k))\right]\end{aligned} \tag{50}$$

Replacing the adjustment laws (39), (40), and (41) in $\Delta\tilde{\boldsymbol{\theta}}_{Fi}$, $\Delta\tilde{\boldsymbol{\theta}}_{Gi}$ and $\Delta\tilde{\boldsymbol{\theta}}_{\delta i}$ of (50).

$$\begin{aligned}\Delta V(k) = \sum_{i=1}^{2}\left[-2k_{di}T_0S_{\mathrm{i}}^2(k)+\Delta S_{\mathrm{i}}^2(k+1)\right.\\ +2S_{\mathrm{i}}\varepsilon_{\mathrm{i}}-2\gamma_1^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{F\mathrm{i}}(k)\right)\\ -2\gamma_2^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{G\mathrm{i}}(k)\right)\\ \left.-2\gamma_3^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}(k)\right)-2\boldsymbol{\eta}_{\mathrm{i}}T_0|S_{\mathrm{i}}(k)|\right]\end{aligned} \tag{51}$$

According to (38)

$$\begin{aligned}|\Delta S_i(k+1)| \le |K_{di}T_0S_i(k)|+\left\|\tilde{\boldsymbol{\theta}}_{gi}^T\right\|\|\boldsymbol{\xi}(\boldsymbol{\zeta}(k))\||\tau_i|+\|\boldsymbol{\eta}\||T_0|\\ +\left\|\tilde{\boldsymbol{\theta}}_{fi}^T\right\|\|\boldsymbol{\xi}(\boldsymbol{\zeta}(k))\|+\left\|\tilde{\boldsymbol{\theta}}_{\delta i}^T\right\|\|\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\|+|\varepsilon_i|+|\tau_{\Delta i}|\end{aligned} \tag{52}$$

All terms in (52) are bounded, and considering $\alpha_0 = \left\|\tilde{\boldsymbol{\theta}}_{Gi}^T\right\|\|\boldsymbol{\xi}(\boldsymbol{\zeta}(k))\||\tau_i|+\|\boldsymbol{\eta}\||T_0|+\left\|\tilde{\boldsymbol{\theta}}_{Fi}^T\right\|\|\boldsymbol{\xi}(\boldsymbol{\zeta}(k))\|+\left\|\tilde{\boldsymbol{\theta}}_{\delta i}^T\right\|\|\boldsymbol{\chi}(\boldsymbol{\upsilon}(k))\|+|\varepsilon_i|$, (52) can be expressed as:

$$|\Delta S_i(k+1)| \le |K_{di}T_0S_i(k)|+|\alpha_0|+|\tau_{\Delta i}| \tag{53}$$

And taking square from both sides of (53),

$$\begin{aligned}|\Delta S_i(k+1)|^2 &\le \left[|K_{di}T_0S_i(k)|+|\alpha_0|+|u_{\Delta i}|\right]^2\\ |\Delta S_i(k+1)|^2 &\le |K_{di}T_0S_i(k)|^2+2|K_{di}T_0S_i(k)||\alpha_0|+\ldots\\ &\quad+2|K_{di}T_0S_i(k)||u_{\Delta i}|+2|\alpha_0||u_{\Delta i}|+|\alpha_0|^2+|u_{\Delta i}|^2\end{aligned} \tag{54}$$

Now adding and subtracting the same term $2|u_\Delta||S_i(k)|$ and rewriting (54).

$$\begin{aligned}&\le \left[|K_{di}T_0S_i(k)|+|\alpha_0|\right]^2+2|K_{di}T_0S_i(k)||\tau_{\Delta i}|\\ &\quad+2|\alpha_0||\tau_{\Delta i}|+2|S_i(k)||\tau_{\Delta i}|-2|S_i(k)||\tau_{\Delta i}|+|\tau_{\Delta i}|^2\end{aligned} \tag{55}$$

Rewriting and rearranging (55)

$$\begin{aligned}|\Delta S_i(k+1)|^2 &\le \left(\beta_2(k)+\beta_1(k)|\tau_{\Delta i}|+|\tau_{\Delta i}|^2\right)\\ &\quad+2|\tau_{\Delta i}||S_i(k)|\end{aligned} \tag{56}$$

where

$$\begin{aligned}\beta_2 &= \left[|K_{di}T_0S_i(k)|+|\alpha_0|\right]^2\\ \beta_1 &= 2|K_{di}T_0S_i(k)|+2|\alpha_0|-2|S_i(k)|\end{aligned} \tag{57}$$

From (58), $\tau_{\Delta i}$ is selected as

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

with the last considerations, and it is easily demonstrated that:

$$\Delta V(k+1) = \sum_{i=1}^{2} \left[-2k_{di}T_0 S_i^2(k) + 2S_i\varepsilon_i \right.$$
$$-2\gamma_1^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{Fi}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{Fi}(k)\right)$$
$$-2\gamma_2^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{Gi}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{Gi}(k)\right)$$
$$\left.-2\gamma_3^{-1}\left(\Delta\tilde{\boldsymbol{\theta}}_{\delta i}^T(k)\Delta\tilde{\boldsymbol{\theta}}_{\delta i}(k)\right) - 2\boldsymbol{\eta}_i T_0 |S_i(k)|\right] < 0 \tag{59}$$

This result produces a better control of the SCARA robot since the error velocities of the dynamic controller converge to zero

6 Experimental results

This section describes the industrial robot used for implementing the control algorithms proposed in Sects 4 and 5. To this aim, the BOSCH SR-800 robot with 2 DOF SCARA structure is considered. The proposed robot manipulator has a CPU control Intel Dual Core on board, running at a frequency of 2.6 GHz with 4 Gb of RAM memory. The computer has an Operating System Linux Debian with RTAI (Real-Time Application Interface). The designed control scheme is applied to the robot manipulator by the power unit control that introduces joints torques as the control actions. In order to perform experiments by applying DTSMAC and static PID control algorithms on this system, the same robot is used (Fig. 4).

For the experiment, the DTMSNAC controller was initiated with random weights and adjusted using different trajectories. In the experiment shown in this paper, the neural parameters are adjusted online by the learning algorithm, evaluating its evolution as a function of time.

The reference trajectory (60) to test the two controllers is the eight-shaped:

$$\begin{cases} X_{\text{ref}} = 0.30\sin\left(\dfrac{2\pi kT_0}{10}\right) & [m] \\ Y_{\text{ref}} = 0.55 + 0.20\cos\left(\dfrac{2\pi kT_0}{10}\right) & [m] \end{cases} \tag{60}$$

In Fig. 5, the control actions of the DTMSNAC controller are shown. Figure 6 shows the reference signals of joint position and instantaneous SCARA robot joint position. Figure 7 shows the trajectories followed by the SCARA robot, using each one of the controllers. In Fig. 8, the square norm of the control errors of both controllers is shown. The highest error was obtained by the classical PID controller, which does not have any online adaptation. In this case, the effect of the uncertainties on the error is clearly observed. But, the lowest error was obtained by the DTSMNAC proposed in this work, which decreases the error caused by the nonmodeled structure and external disturbance. In addition, the evolutions of the tuning neural weights of the DTMSNAC controller are shown in Figs. 9, 10, and 11.

The DTSMNAC controller was designed to be robust with respect to modeling errors. It is also more effective in rejecting disturbances and does not produce constant error caused by any of the uncertainties of parameters and external disturbances. Furthermore, the static PID controller is vulnerable to changes in the dynamic model and the uncertainties, due to the static PID control being designed for a linearized model of the robot arm dynamics.

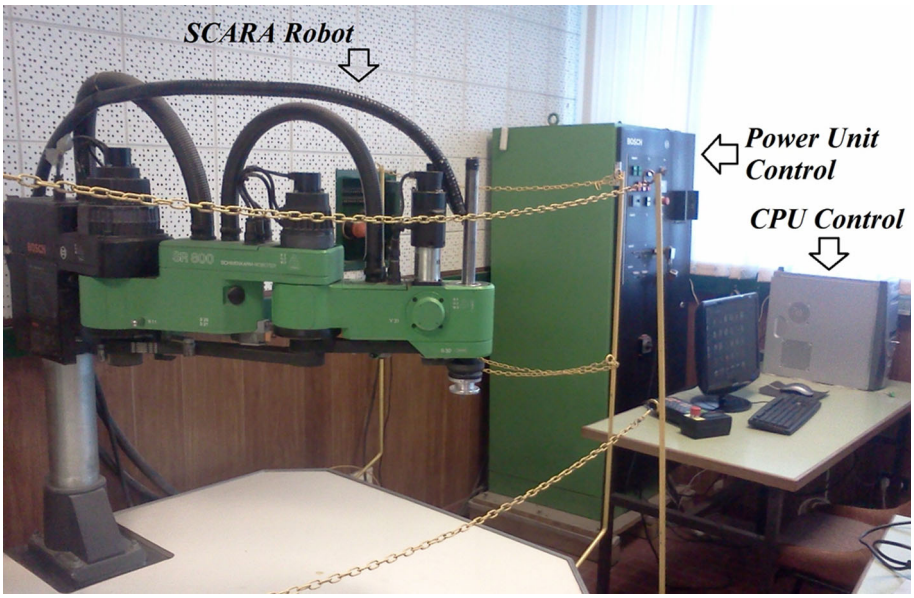

Fig. 4 SCARA robot and power unit control

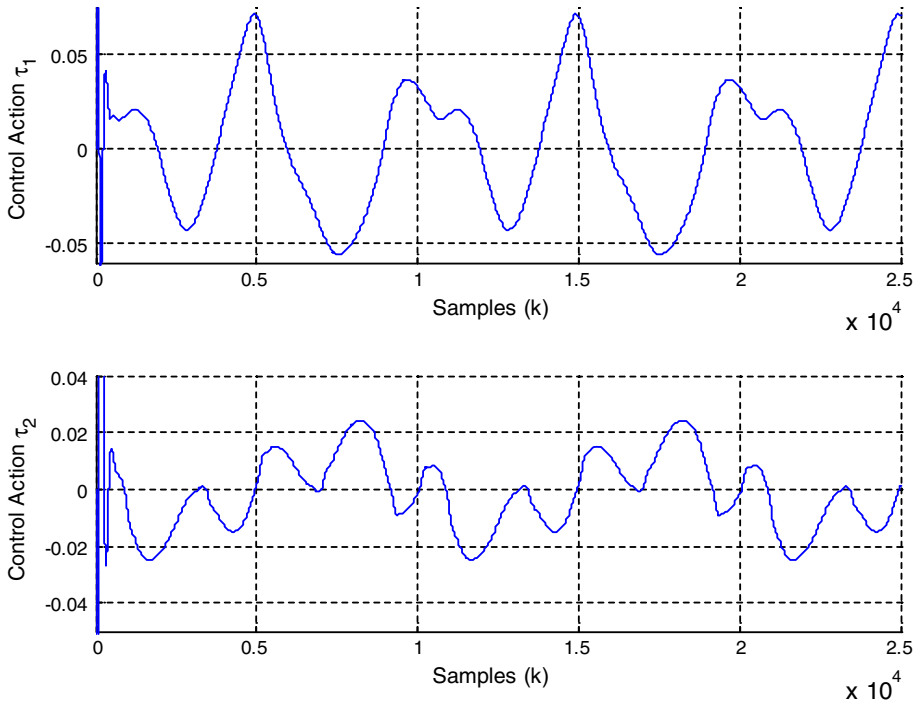

Fig. 5 Control actions, outputs of the SCARA robot

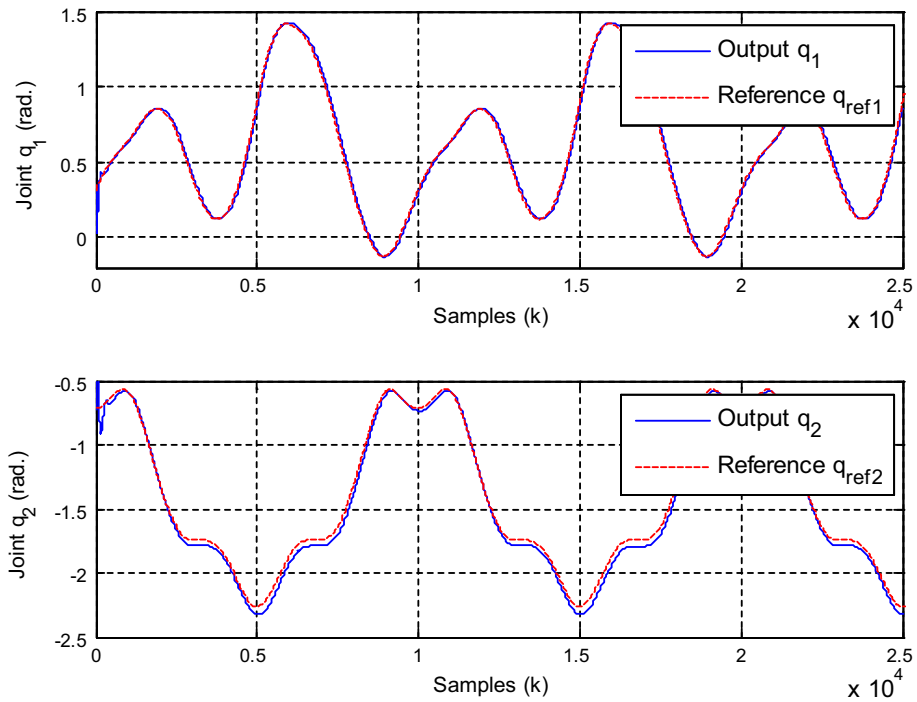

Fig. 6 Outputs and reference signals of joint position and SCARA robot joint position

The control law developed in this work for a nonlinear system does not need to know the dynamic model of the robot arm. However, it is known that the mathematical representation of a dynamic model does not accurately describe the actual behavior.

These nonlinearities and uncertainties of the model along with variations in robot dynamics demonstrate the robustness of the DTSMNAC controller. The stability of the proposed control system was demonstrated analytically via discrete Lyapunov's stability theory (Sects. 4, 5).

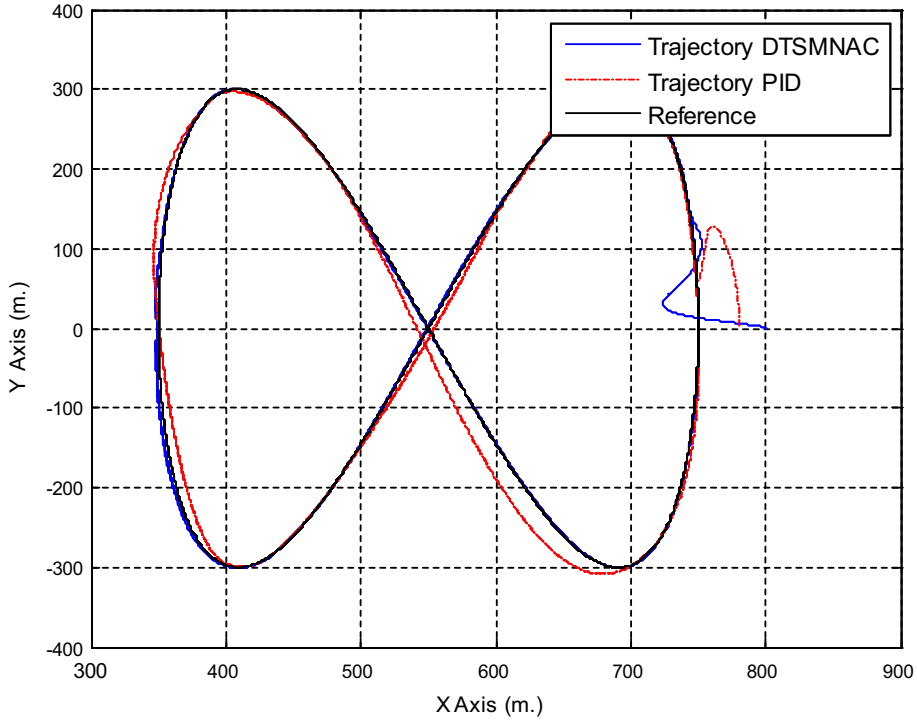

Fig. 7 Trajectory followed by the SCARA robot including a DTSMNAC controller (*solid line*) compared with the trajectory made by the robot including a conventional static PID controller (*dashed line*)

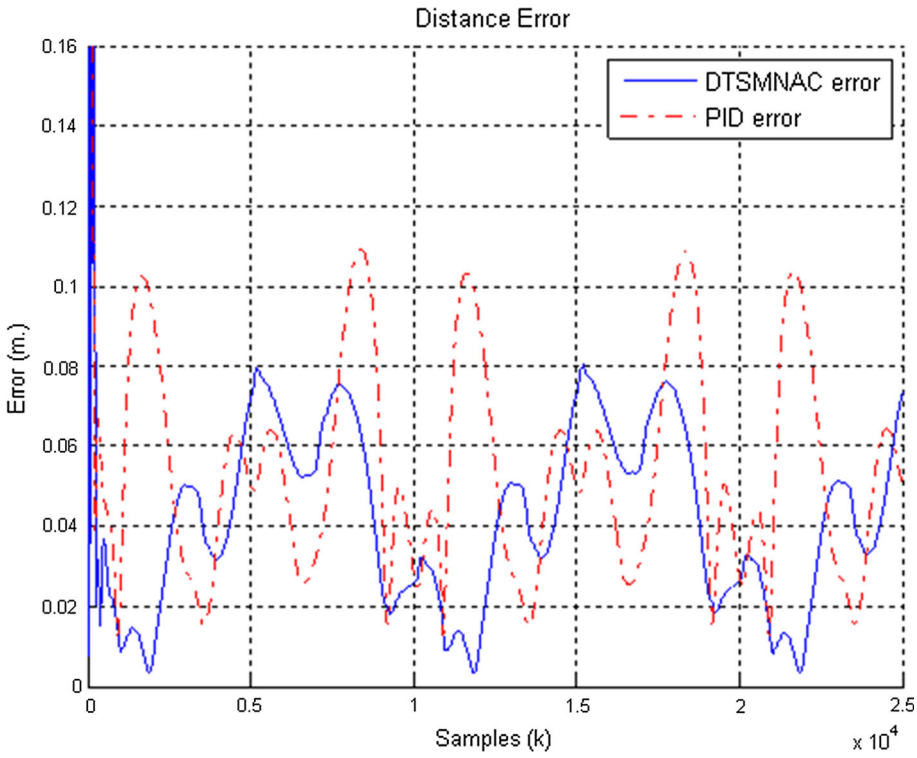

Fig. 8 Instantaneous quadratic error of the robot position. Robot with DTSMNAC (*solid line*) and robot with classical PID (*dashed line*)

This proposal of intelligent control can be considered as a general solution for the control of nonlinear systems and in particular for the case of robotic systems or when the dynamics is variable or has uncertainties in the model.

This field of research is wide open to the issues of modeling, mathematical stability, convergence, and robustness analysis of control systems which continue to advance to design increasingly accurate controllers.

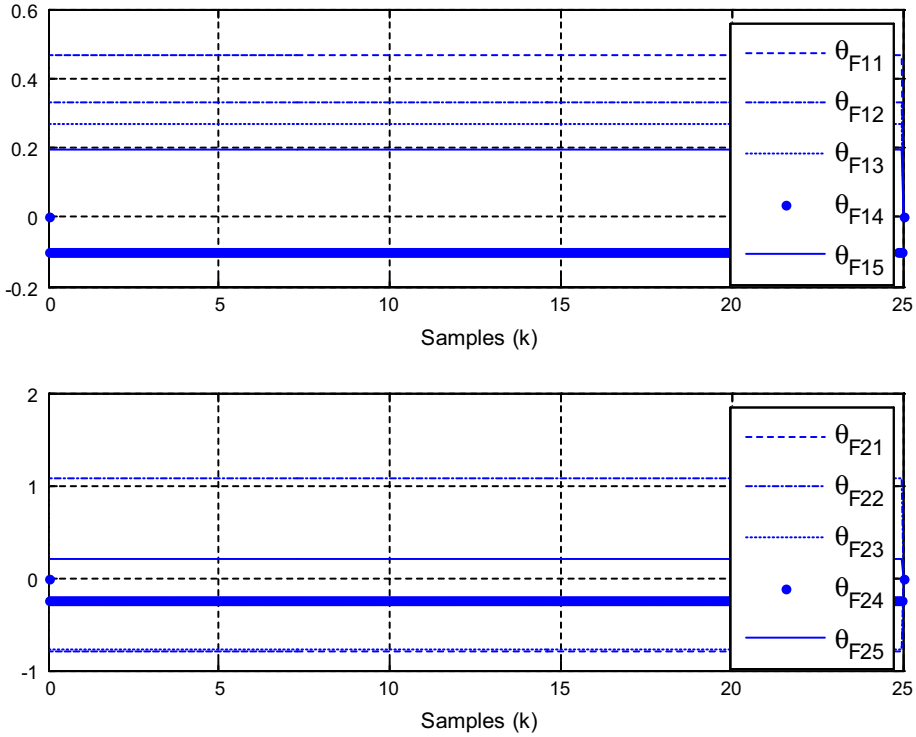

Fig. 9 Evolution of the θ_F weight parameters of the neural DTSMNAC network during the experiment

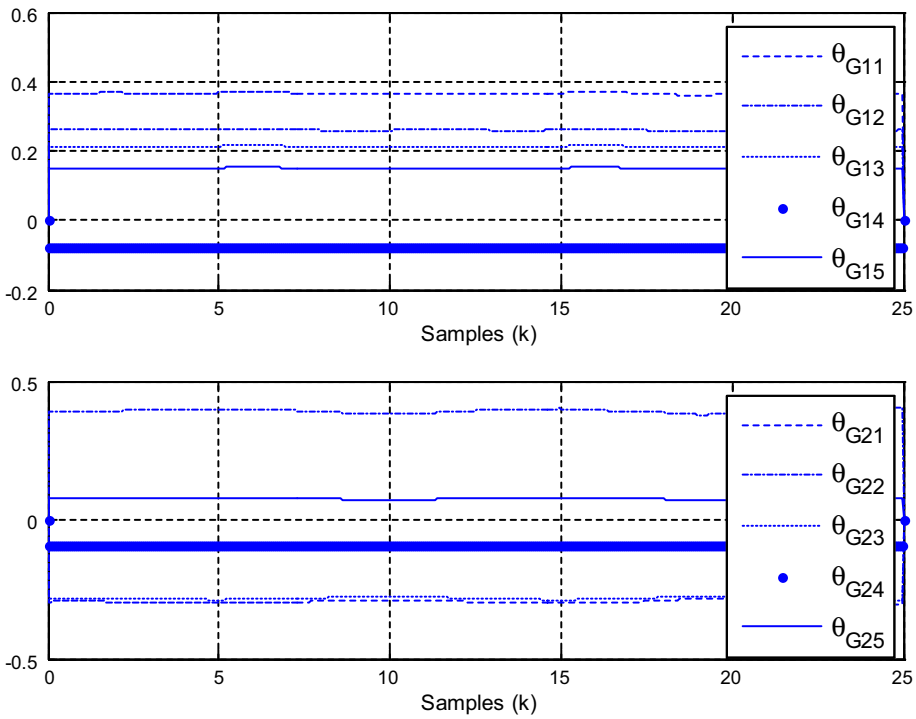

Fig. 10 Evolution of the θ_G weights of the neural DTSMNAC network during the experiment

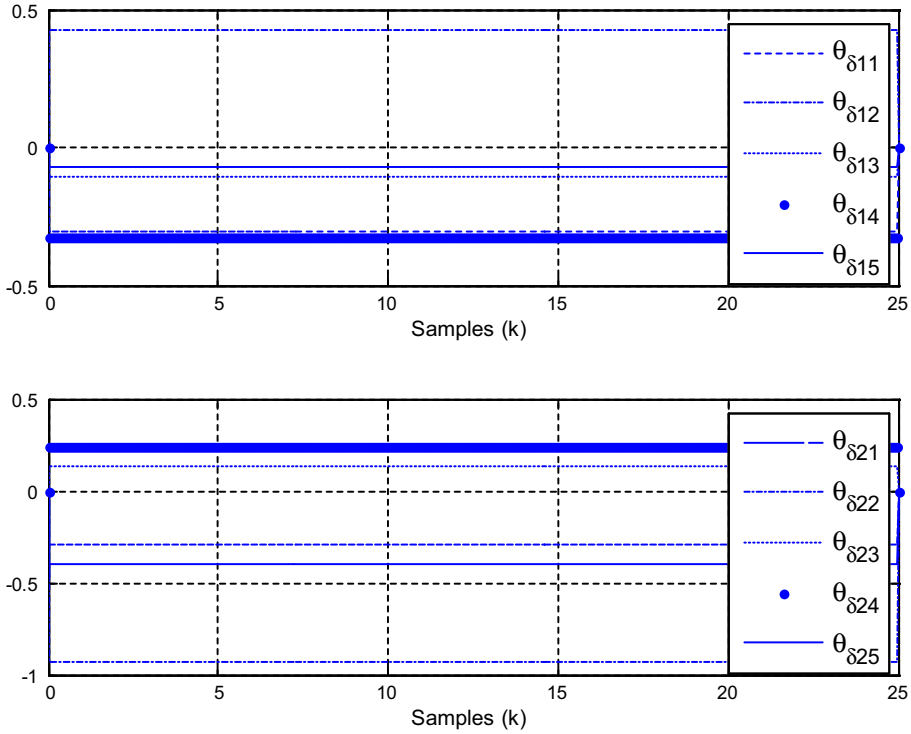

Fig. 11 Evolution of the θ_δ weights of the neural DTSMNAC network during the experiment

7 Conclusion

An adaptive neural controller-based sliding mode control has been proposed for the robust trajectory tracking of SCARA robot manipulator with unknown nonlinear dynamics. The core of this structure does not require knowledge of the system dynamics and parameters to compute the equivalent control, and an adaptive neural system is developed to compensate further the system uncertainty and knowledge incompleteness. This scheme obtains robustness in the sense that the self-tuning mechanism can automatically adjust the neural sliding mode controller by using a learning algorithm. And the global asymptotic stability of the algorithm is established via the Lyapunov discrete stability conditions. When matching with the predefined model occurs, the whole control system becomes equivalent to a stable dynamic system in the discrete-time domain.

The design can achieve the objective of discrete-time adaptive sliding mode control and also ensure that the output tracking error converges to zero.

References

1. de Wit CC, Siciliano B, Bastin G (1996) Theory of robot control. Springer, New York
2. Lewis F, Abdallah C, Dawson D (1993) Control of robot manipulators. MacMillan Publishing Co., New York
3. Samson C, Le Borgne M, Espinau B (1991) Robot control. Oxford University Press, Oxford
4. Spong M (1996) Motion control of robot manipulator. In: Levine W (ed) Handbook control. CRC Press, pp 1339–1350
5. Khalil W, Dombre E (2002) Modeling identification and control of robots. Hermes Penton Science, London
6. Jiang Z-H, Ishida T (2007) Trajectory tracking control of industrial robot manipulators using a neural network controller. In: IEEE international conference on systems, man and cybernetics, 2007. ISIC, pp 2390–2395
7. Khalal O, Mellit A, Rahim M, Salhi H, Guessoum A (2007) Robust control of manipulator robot by using the variable structure control with sliding mode. In: Mediterranean conference on control & automation, 2007. MED '07, pp 1–6
8. Raafat SM, Said WK, Akmeliawati R, Tariq NM (2009) Improving trajectory tracking of a three-axis SCARA robot using neural networks. In: IEEE symposium on industrial electronics & applications, 2009. ISIEA 2009, pp 283–288
9. Benjanarasuth T, Sowannee N, Naksuk N (2010) Two-degree-of-freedom simple servo adaptive control for SCARA robot. In: 2010 international conference on control automation and systems (ICCAS), p 480–484
10. Suvilath S, Khongsomboun K, Benjanarasuth T, Kominet N (2011) IMC-based PID controllers design for a two-links SCARA robot. In: TENCON 2011–2011 IEEE region 10 conference, p 1030–1034
11. Al-Khedher MA, Alshamasin MS (2012) SCARA robot control using neural networks. In: 2012 4th international conference on intelligent and advanced systems (ICIAS), vol 1, pp 126–130
12. Thanok S (2014) Design of an adaptive PD controller with dynamic friction compensation for direct-drive SCARA robot. In: Electrical engineering congress (iEECON), 2014 international. IEEE, 2014
13. Escobar F, Díaz S, Gutiérrez C, Ledeneva Y, Hernández C, Rodríguez D, Lemus R (2014) Simulation of control of a Scara robot actuated by pneumatic artificial muscles using RNAPM. J Appl Res Technol 12(5):939–946

14. Maeda Y, Fujiwara T, Ito H (2014) Robot control using high-dimensional neural networks. In: 2014 Proceedings of the SICE annual conference (SICE), p 738–743
15. Lin TC, Chang SW, Hsu CH (2012) Robust adaptive fuzzy sliding mode control for a class of uncertain discrete-time nonlinear systems. Int J Innov Comput Inf Control 8(1):347–359
16. Rossomando FG, Soria C, Carelli R (2014) Sliding mode neuro adaptive control in trajectory tracking for mobile robots. J Intell Robot Syst 74(3):931–944. ISSN: 0921-0296
17. Asada H, Slotine JJE (1986) Robot analysis and control. Wiley, New York