Applied Soft Computing 25 (2014) 139–148

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Neuro-fuzzy self-tuning of PID control for semiglobal exponential tracking of robot arms

Jorge Armendariz [a,*], Vicente Parra-Vega [a], Rodolfo García-Rodríguez [b], Sergio Rosales [a]

[a] Department of Robotics and Advanced Manufacturing, Research Center for Advanced Studies (CINVESTAV), Mexico
[b] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Chile

ARTICLE INFO

Article history:
Received 5 September 2013
Received in revised form 8 June 2014
Accepted 15 August 2014
Available online 26 September 2014

Keywords:
Self-tuning PID
Neuro-fuzzy
Robot arms
Model-free tracking

ABSTRACT

The PID controller with constant feedback gains has withstood as the preferred choice for control of linear plants or linearized plants, and under certain conditions for non-linear ones, where the control of robotic arms excels. In this paper a model-free self-tuning PID controller is proposed for tracking tasks. The key idea is to exploit the passivity-based formulation for robotic arms in order to shape the damping injection to enforce dissipativity and to guarantee semiglobal exponential convergence in the sense of Lyapunov. It is shown that a neuro-fuzzy network can be used to tune dissipation rate gain through a self-tuning policy of a single gain. Experimental studies are presented to confirm the viability of the proposed approach.

1. Introduction

Effectiveness and simple model-free control structure are the principal characteristics that make PID controller be useful in several applications. Based only on state feedback, the PID control has been developed for regulation tasks principally, [1–5]. The regulation of robotic arms, without linearization scheme, stands out from the overwhelming contributions in the literature on PID control. In particular, it has become the preferred option in the industrial floor for control of robot manipulators, [6]. To overcome the intrinsic uncertainties of the model commonly encountered in the robotic tasks, [7], some studies have provided the fundamentals for regulation without using the regressor nor any robot parameter, [8]. However, *stability* for the *tracking case* with a pure PID structure of non-linear plants, in particular for robotic arms, remains largely elusive in the literature. Additionally, to simplify the complex procedure of tuning constant feedback gains in the time domain for robotic arms, explicit rules rely on advanced stability grounds [9], or in view of the lack of simple procedures *knowledge-based* schemes have been proposed as an alternative to tuning feedback gains on-line [10]. In this paper, we explore a novel idea to design a self-tuning PID controller for robotic arms to obtain exponentially semiglobal tracking, in the sense of Lyapuonov [11], even under affine unmodeled bounded exogenous disturbance and smooth joint friction.

Using a typical *passivity-based* robot control approach [12,13], it is shown that the error equation can be stabilized in the extended error manifold in the sense of Lyapunov, where the nonlinear integral control term yields a smaller stable domain of attraction. In order to drive the error manifold to zero to ensure the *tracking*, a neuro-fuzzy network is proposed to extract the knowledge and tune a gain, avoiding *any knowledge* of robot dynamics, in contrast to [14,15], nor attempting to approximate inverse dynamics, [12]. This gain is used together a PID controller of constant gains, resulting in a self-tuning PID control based on a single feedback gain using a neuro-fuzzy scheme. This self-tuning gain in fact expands or contracts the domain of attraction where passivity is enforced to yield Global Uniformly Ultimately Bounded, GUUB. Once tracking errors are trapped in this domain of attraction, the tuning scheme is synthesized and an exponential convergence to the desired trajectory is enforced, as long as initial conditions belong to the compact set in the neighbourhood of the time-varying desired trajectory. In this sense our proposal extends, by simplifying, the involved neuro-fuzzy structure of [16–18] such that the neuro-fuzzy tuning scheme enforces a dissipative mapping in terms of the error manifold by shaping the dissipation rate function to dominate the robot dynamics. This shows that a *simple* and *intuitive* fuzzy-based tuning rules keeps a model-free PID control structure to get tracking.

This paper is organized as follows. A short background on self-tuning PID is presented in Section 2 to contextualize our

* Corresponding author. Tel.: +528444389600.
E-mail addresses: jorgearm.ryma@gmail.com (J. Armendariz), vparra@cinvestav.mx (V. Parra-Vega), rgarcia1@miuandes.cl (R. García-Rodríguez), wildsarm@gmail.com (S. Rosales).

http://dx.doi.org/10.1016/j.asoc.2014.08.037
1568-4946/

contribution. Then, the proposal is given in Section 3, with the self-tuning policy presented in Section 4. Section 5 shows the main result and its stability analysis, with discussions and remarks in Section 6. Comparative experimental results are presented in Section 7, which highlights the viability and potential of the proposed approach in real robotic arms applications. Final conclusions are provided in Section 8.

2. Background on self-tuning PID

The seminal work of Ref. [20] established the fundamentals for regulation of robotic arms with PID control using constant gains, and [21] sets the grounds for its applicability to tracking tasks. Afterwards, various schemes of non-constant self-tuning feedback gains have been studied to provide better stability properties, [22]. Schemes such as linearizing robot dynamics, regressor-free or non-constant feedback gains have been introduced for improved performance, e.g. gain scheduling [23], iterative learning control [24], regressor-free [8], neuro-fuzzy [25], adaptive fuzzy [26], or state-dependant tuning gains [27].

Among these schemes, in contrast to the classical neural network plus a PID-like stabilizer, see [28,29], there appear other schemes whose PID structure in fact hides a classical function approximation scheme, see [30] for neural networks, with wavelets in [31], recurrent neural networks in [32], and fuzzy neural networks in [33]. Although several schemes claim tracking capabilities with PID control, a close scrutiny shows the lack of *stability results* or in fact turns out BIBO stability even for LTI plants, at best, similar to [21] for the non-linear robot arm case. From the instrumental work of [34], PID fuzzy-based control for linear plants has been applied indistinctly for a wide family of systems and regimes, including for semiglobal regulation of non-linear robot dynamics [14], however unsuccessfully for tracking regime. Although there are impressive application studies on PID control of complex plants, the vast majority of these contributions share the common characteristic that *no stability proof* is presented, [35]. Recently, in [36] an adaptive fuzzy controller for tracking of a single-link robot manipulator is proposed based only in the measurement of position while the velocity signals are estimated using an adaptive fuzzy filter observer. The boundedness of the closed loop signals is guaranteed under assumption that the non-linear uncertainties are unknown. Within the large body of literature on PID control, our proposal is aligned to the schemes that preserve only a PID control structure without resorting on any type of inverse dynamics compensation while the feedback gains are time-varying. The proposed self-tuning mechanism, based on dissipation rate using a *PID error manifold*, facilitates the design controller because the non-autonomous representation of the open-loop error equation is parametrized by such PID error manifold, which is used as the output of the passivity inequality. In this realm, we claim that our proposal stands for the first one that enforces and proves semiglobal exponential tracking for robotic arms using a self-tuning PID, or PD.

3. Model-based control and our proposal

3.1. Dynamic model in error coordinates

Consider the Euler–Lagrange formalism to model the energy balance of articulated mechanical systems, then let the well-known non-linear dynamic model of a rigid serial n-link robot manipulator be represented as follows:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - \tau_f + \tau_d, \tag{1}$$

where $q \in \Re^n$, $\dot{q} \in \Re^n$ are the generalized position and velocity joint coordinates, respectively, $H(q) \in \Re^{n\times n}$ denotes a symmetric positive definite inertial matrix, $C(q,\dot{q}) \in \Re^{n\times n}$ represents the Coriolis and centripetal forces, $g(q) \in \Re^n$ models the gravity forces, and $\tau \in \Re^n$ stands for the torque input. Term $\tau_f = B\dot{q} + C\tanh(D\dot{q})$ stands for joint friction where B and C are positive definite diagonal $n \times n$ matrices modeling viscous damping and kinetic friction, respectively, and bounded $D \in \Re_+^{n\times n}$ gives rise to a reasonable approximation of the non-smooth Coulomb friction model. Disturbance torque τ_d is assumed bounded and possibly slowly time-varying.

Let the following linear parametrization of (1), [40],

$$H(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + g(q) + B\dot{q}_r + C\tanh(D\dot{q}_r) - \tau_d = Y_r\Theta, \tag{2}$$

where the regressor $Y_r = Y_r(q,\dot{q},\dot{q}_r,\ddot{q}_r) \in \Re^{n\times p}$ represents the matrix of known terms and $\Theta \in \Re^p$ is a p vector of unknown parameters, [13], and $\dot{q}_r \in \Re^n$ stands for the nominal reference, being $\ddot{q}_r$ its derivative. Adding and subtracting (2) into (1), there appears the open-loop error equation:

$$H(q)\dot{S} + C(q,\dot{q})S = \tau - Y_r\Theta, \tag{3}$$

where the error manifold S is defined as:

$$S = \dot{q} - \dot{q}_r. \tag{4}$$

Let the nominal reference given by Ref. [19],

$$\dot{q}_r = \dot{q}_d - \alpha\Delta q - K_i\int_{t_0}^{t_f}\tanh(\lambda S_q)dt, \tag{5}$$

where S_q defines the PD error manifold by:

$$S_q = \Delta\dot{q} + \alpha\Delta q, \tag{6}$$

for α, λ, and K_i as positive-definite diagonal feedback gain matrices, $\Delta q(t) = q(t) - q_d(t)$ denotes the position tracking error for $q_d(t) \in C^2$ (the dependency on t will be dropped subsequently to avoid any confusion and simplify the notation) the bounded differentiable continuous desired trajectory, with $\dot{q}_d$ its derivative as the desired velocity, and $\tanh(\lambda x)$ is the continuous hyperbolic tangent function of $x \in \Re^n$. Substituting (5) into (4), it renders the extended error manifold:

$$S = S_q + K_i\sigma, \tag{7}$$

$$\dot{\sigma} = \tanh(\lambda S_q), \tag{8}$$

notice that (7) and (8) stands for a PI error manifold homogeneous in S_q, whose unforced derivative $\dot{S}_q = -K_i\tanh(\lambda S_q)$ has a unique solution at $S_q = 0$. Then, if there exists a controller τ that renders asymptotically $S \to 0$, we can show that the unforced solution of (7) and (8) attains an unique asymptotic equilibrium at $(\Delta q, \Delta\dot{q}) = (0, 0)$ for a storage function $V_0 = 1/2S_q^TS_q$. Finally, at this point, observe that $\sigma = 0$ yields a linear PD error manifold with similar results.

In this paper, we exploit the model-free non-linear PID controller with constant gains $\tau = -K_dS$ and show that it yields GUUB. Moreover, we show that if K_d is chosen on-line using a neuro-fuzzy scheme, then there arises exponential convergence of S. To motivate the structure of this controller, and similar to [21], we show that the well-known result for local stability of robots using the PI error manifold (7) and (8) is valid. This fact will empower our self-tuning proposal to include input knowledge of the user by means of setting bounded multi-valued criteria to tune online K_d based on a neuro-fuzzy mechanism. In the sequel, we will review some well-known techniques to contextualize our proposal.

3.2. Model-based robot control

Before to present the control-system stabilization of unknown regressor and parameters (Section 3.3), it is necessary to briefly explain the passive computed torque with and without adaptation, in order to remark the importance of adaptation by using only one varying parameter, named $\hat{K}_d$.

3.2.1. Passive computed torque: Y_r and Θ are known

Aiming at preserving passivity in closed-loop when Y_r and Θ are available, it is easy to see that a dissipative mapping is globally valid for a storage (Lyapunov) function $V_1 = 1/2S_q^T H(q)S_q$ for $\tau = -K_d S_q + Y_r\Theta$ where $K_d = K_d^T > 0 \in \Re_+^{n\times n}$, [38]. To see this, notice that the following closed loop equation appears:

$$H(q)\dot{S}_q + C(q,\dot{q})S_q = -K_d S_q - \nu_1, \tag{9}$$

where $\nu_1 = 0$ is a virtual input defined for analysis purpose. The passivity balance leads to:

$$\int_{t_0}^{t} {S_q}^T \nu_1\, dt = \int_{t_0}^{t} \frac{d}{dt} V_1(S_q) dt + \xi \tag{10}$$

in virtue of the antisymmetry of $1/2\dot{H}(q) - \dot{C}(q,\dot{q})$, [42], and $\xi = \int_{t_0}^{t} {S_q}^T K_d S_q dt > 0$ stands for the dissipation function, with K_d its dissipation rate gain. In this case, any constant K_d endows the system with dissipative behavior and a global domain of stability $\mathcal{D}'_0$ containing the open set $\mathcal{R}_0$ of initial conditions $X(t_0) = X_0$ is established. More precisely, (10) leads to $\dot{V}_1(S_q) = -{S_q}^T K_d S_q$ which suggest the GUUB of the origin. Now, let us review the case of unknown Θ in light of the adaptive approach of Ref. [13].

3.2.2. Adaptive passive computed torque: Y_r is known but Θ is unknown

When Y_r is known but Θ is unknown, the well-known adaptive passive computed torque, or simply adaptive control of [13] is defined. Considers $\tau = -K_d S_q + Y_r\hat{\Theta}$, where $K_d = K_d^T > 0 \in \Re_+^{n\times n}$, and the vector of parametric uncertainty $\Delta\Theta$ is compensated adaptively by $\dot{\hat{\Theta}} = -\Gamma Y_r^T S_q$. This gives rise, by Barbalath Lemma, to the local asymptotic stability using the Lyapunov function $V_2 = 1/2S_q^T H(q)S_q + 1/2\Delta\Theta^T\Gamma^{-1}\Delta\Theta$. In this case, a dissipative mapping can be obtained locally to guarantee tracking with smooth control effort under an over-parameterization of the closed-loop dynamics in virtue of $p \gg n$. Interestingly, this can be re-interpreted as if $\hat{\Theta}(t)$ were time-varying feedback gains that contributes to drive error $S_q \to 0$ at the expense of full knowledge of the robot model encoded through Y_r. Finally, notice that even though online adaptation of vector $\hat{\Theta}(t)$ is not constant obviously, it does not contribute to the dissipation rate because it targets to compensate for parametric uncertainty $\Delta\Theta$, that is, ξ is the same as previous case. Now, let us analyze the case when both Y_r and Θ are unknowns.

3.3. Stabilization with a PID of constant gains under unknown Y_r and Θ

In this case, for unknown Y_r and Θ let

$$\begin{aligned}\tau &= -K_d S = -K_d S_q - K_d K_i \sigma \\ &\equiv -K_p\Delta q - K_v\Delta\dot{q} - K_\iota\sigma,\end{aligned} \tag{11}$$

be a PID controller parametrized by the PI error manifold, where $K_d = K_d^T > 0 \in \Re_+^{n\times n}$, $K_p = K_d\alpha$, $K_v = K_d$, and $K_\iota = K_d K_i$. The closed-loop system, using (11) and (3), leads to the very same (9):

$$H(q)\dot{S} + C(q,\dot{q})S = -K_d S - \nu_1, \tag{12}$$

however in this case $\nu_1 = Y_r\Theta$. Clearly, we obtain (10), that is $\int_{t_0}^{t} S^T\nu_1\, d\tau = \int_{t_0}^{t} \frac{d}{dt}V_3(S)d\tau + \xi$ for $V_3(S) = 1/2S^T H(q)S$ and $\xi = \int_{t_0}^{t} S^T K_d S d\tau > 0$. This is not passive because $\nu_1 \neq 0$, however, for large enough K_d a bounded local domain of stability $\mathcal{D}_0$ can be established to guarantee stability of tracking errors. To see this, consider that there exists positive scalars ρ_i ($i = 0, \ldots, 5$) that bounds $Y_r\Theta$, [19,21], as follows:

$$\begin{aligned}
\|H(q)\| &\geq \lambda_m(H(q)) > \rho_0 > 0,\\
\|H(q)\| &\leq \lambda_M(H(q)) < \rho_1 < \infty,\\
\|C(q,\dot{q})\| &\leq \rho_2\|\dot{q}\|,\\
\|g(q)\| &\leq \rho_3,\\
\|\dot{q}_r\| &\leq \rho_4 + \alpha\|\Delta q\| + K_i\|\sigma\|,\\
\|\ddot{q}_r\| &\leq \rho_5 + \alpha\|\Delta\dot{q}\|,
\end{aligned} \tag{13}$$

where $\lambda_m(A)$, $\lambda_M(A)$ stand the minimum and maximum eigenvalues of matrix $A \in \Re^{n\times n}$, respectively, and ρ_4, ρ_5 sets the upper bounds for desired velocity and acceleration, respectively. Then, we have that:

$$\begin{aligned}
Y_r\Theta &\leq \|H(q)\|\|\ddot{q}_r\| + \{\|B\| + \|C(q,\dot{q})\|\}\|\dot{q}_r\| + \|g(q)\| + \|C\|\|\tanh(\Gamma\dot{q}_r)\|\\
&\leq \rho_1\alpha\|\Delta\dot{q}\| + \{\lambda_M(B) + \rho_2\|\dot{q}\|\}\cdot(\alpha\|\Delta q\| + K_i\|\sigma\| + \rho_4) + \lambda_M(C) + \overline{\rho}_3\\
&\leq \eta(t),
\end{aligned} \tag{14}$$

where $\overline{\rho}_3 = \rho_1\rho_5 + \rho_3$, and $\eta(t) = f(\Delta q, \Delta\dot{q}, \sigma, \rho_i, t)$. According to (14), we have that $\dot{V}_3 \leq -\|S\|(\lambda_m(K_d)\|S\| - \eta(t))$. Thus, there exists a finite time $t_1 > t_0$ such that $\|S\| > (c/\lambda_m(K_d))$ for $c > suparg_{t\geq 0}\eta(t)$ for which $\dot{V}_3 < 0$. Then, for small initial conditions $X_0 \leq c_0$, for $c_0 > 0$, $\eta(t_0)$ belongs to a neighbourhood with radius $r > 0$ centered in the origin $S = 0$. This shows the existence of bounded domain of stability $\mathcal{D}_0$ for a large enough feedback gain K_d, implying that S converges into a set ε_1 of radius $r_1 > 0$ bounded by $S \leq c/\lambda_m(K_d)\,\forall t \geq t_2$. Therefore, $\dot{S} \leq \bar{c}$, for some real $\bar{c}$. Namely,

$$S \to \varepsilon_1 \Rightarrow S_q \to \varepsilon_1 \quad \text{and} \quad (S, S_q) \subset \mathcal{D}_0 \quad \forall t \geq t_2, \tag{15}$$

this stands for local stability of S and the boundedness of all closed-loop signals, in the $\mathcal{L}_\infty$ sense, which leads to the existence of bounded neighbourhoods $\varepsilon_2 > 0$, and $\varepsilon_3 > 0$ such that $\|S\| < \varepsilon_2$, $\|\dot{S}\| < \varepsilon_3$. Finally, in virtue of (7), the boundedness of tracking errors is established, that is $\|\Delta q\| < \varepsilon_4$, $\|\Delta\dot{q}\| < \varepsilon_5$, $\|\sigma\| < \varepsilon_6$, for a PID controller $\tau = -K_d S = -K_d S_q - K_d K_i\sigma \equiv -K_p\Delta q - K_v\Delta\dot{q} - K_\iota\sigma$ with *constant* K_d. Thus, we have proved the following result.

Proposition 1. *Consider the closed-loop system (12), then the nonlinear PID controller (11) yields global uniform ultimate boundedness (GUUB) with constant K_d assuming small errors on initial conditions.* ■

Remark 1. Notice that the size of ε_i is reduced as K_d increases, however the saturation level of actuators sets a finite upper limit for K_d. An alternative approach is online tuning of K_d depending on the size of position tracking errors. We pursue this idea based on neuro-fuzzy arguments to show that for bounded S, a bounded $K_d = f(S)$ gives rise to local exponential stability, which can be expanded in the semi-global sense, [15]. Now, we review the case of finite $K_d(t)$ under controller $\tau = -K_d(t)S$.

3.4. Proposal: varying K_d under unknown $Y_r\Theta$

In the sequel and provided that Proposition 1 holds, there remains the question on what sort of improvement can be obtained with time-varying K_d, and more important, how to obtain the corresponding *stability* proof. This question is pertinent because experience indicates that some tasks can be executed more

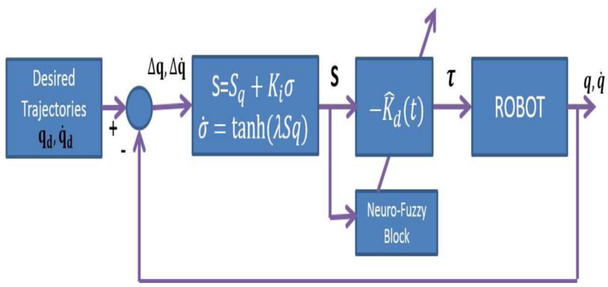

Fig. 1. Block diagram of the proposed controller.

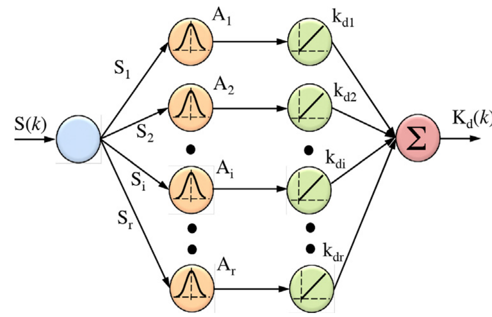

Fig. 2. The fuzzy-neural network is driven by input joint error manifold S to produce a $\hat{K}_d$ output, at instant k. The network structure is decomposed into four layers: (1) S-input-vector layer decomposed on S_i elements, (2) membership function layer, (3) linear consequence layer, and (4) $\hat{K}_d$-output layer.

accurately and smoothly when feedback gains are time-varying according to the error and initial conditions rather than using constant gains. Now, consider the following controller:

$$\tau = -\hat{K}_d S \equiv -\hat{K}_P \Delta q - \hat{K}_D \Delta\dot{q} - \hat{K}_I \sigma, \tag{16}$$

for $\hat{K}_P = \hat{K}_d\alpha$, $\hat{K}_D = \hat{K}_d$, and $\hat{K}_I = \hat{K}_d K_i$, for constant positive definite feedback gains α, and K_i. Intuition suggests that a time-varying gain policy can simply consider that if error increases, then increases the gain that affects most such error, and viceversa, and so forth, which eventually will result on $(\varepsilon_4, \varepsilon_5) \rightarrow 0$ accordingly. This motivates to argue that since $\hat{K}_d$ shapes the dissipation rate ξ in the passivity inequality (10), it seems reasonable to embrace tuning of $\hat{K}_d$ based on the deviation of position errors, essentially a greater dissipation rate to greater deviation is required and vice-versa. Clearly, that under unknown $Y_r\Theta$, an analytical algorithm to tune $\hat{K}_d$ is incomplete, (pag. 26 of [34]), but exploiting user experience and knowledge on system performance, it seems feasible to resort in neuro-fuzzy schemes with the purpose to design a self-tuning algorithm for $\hat{K}_d$ without any attempt to approximate inverse dynamics.

4. Self-tuning design of $\hat{K}_d$

4.1. Design of $\hat{K}_d$

Let a neuro-fuzzy scheme tune the gain $\hat{K}_d$ as follows:

$$\hat{K}_d \equiv \sum_{i=1}^{r} k_{di}(k) = \sum_{i=1}^{r} \beta_i(k)\,\mu_{A_i}(k), \tag{17}$$

where $\beta_i(k)$ is a parameter to minimize a quadratic cost function of $S(k)$, and $\mu_{A_i}(k)$ stands for the membership function. In this way, and based on user experience, it is assumed the user is able to tune them to obtain tracking for a given bounded and smooth time-varying trajectory, resulting in bounded $\hat{K}_d$, see Fig. 1. The tuning process will be explained later. Let the universe of discourses of S and k_d be defined as $S \subset \bar{S}$ and $k_d \subset \hat{K}_d$, respectively, [34]. Then for each S input, the fuzzy **IF-THEN** rule, similar to a Sugeno-type model, is defined as:

Rule i: **IF** S IS A_i **THEN** $K_d = k_{di}(k) = \beta_i(k)\,\mu_{A_i}(k)$, where A_i denotes a linguistic-classified subset linked with the membership function (*MF*) of the fuzzy state, for i=1, 2,..., r-fuzzy states and $k_{di}(k) \subset \hat{K}_d$.

We now shows firstly the self-tuning fuzzy logic scheme to obtain $\beta_i(k)$, and then we describe the rationale behind the design of $\mu_{A_i}(k)$. In fact, the gain $\hat{K}_d > 0$ is tuned using a two-stages neuro-fuzzy algorithm. In the first stage, fuzzy logic rules are designed based on S vector, while in the second stage, it is used the output of the first stage as inputs of a neural network (with unity weighting between layers) to obtain the corresponding summation to build up the output $\hat{K}_d$, see Fig. 2.

4.2. Design of membership functions

The *Rule i* indicates that **if** S belongs to the fuzzy set A_i, with corresponding membership value $\mu_{A_i}(k)$, **then** the fuzzy value of the output of this rule, denoted by $k_{di}(k)$, is equal to the linear consequence (*LC*) defined by the product $\beta_i(k)\,\mu_{A_i}(k)$; where $\beta_i(k)$ and $\mu_{A_i}(k)$ represent the slope of the *LC* and the membership value of the *MF*, respectively. Once the rules are processed, the output $\hat{K}_d$ at instant k is computed from the summation of all $k_{di}(k)$ values by using defuzzification.

Each fuzzy rule is transferred through a fuzzy-neural network composed by four layers. This particular configuration has a simple structure able to transfer knowledge about S based on fuzzy rules injected into the network, [16–18]. It can be seen that the input layer simply processes S into the second layer, where each node has a Gaussian or sigmoid *MF*, given by $A_i = \mu_{A_i}(k)$ corresponding to one linguistic level (e.g. negative, positive, etc.). The *MFs* are processed to the corresponding *i-th* node of the next (*LC* layer), which is computed as (17), that is:

$$k_{di}(k) = \beta_i(k)\,\mu_{A_i}(k). \tag{18}$$

Finally at the output layer gives rise to (17). This structure allows to tune a suitable $\hat{K}_d$ value according to user experience on system performance encoded in terms of tracking errors or error manifold S, which is useful to modulate the domains of attraction $\mathcal{D}_i$. Now, an adaptive algorithm for *LC* $\beta_i(k)$ is proposed to minimize a quadratic index of extended error manifold S.

4.3. Online adaptation of parameter $\beta_i(k)$

Due to $\hat{K}_d = f(S(k))$ is an state-dependent non-linear function of $S(k) = \Delta\dot{q}(k) + \alpha\Delta q(k) + K_i\sigma(k)$ with $\Delta q(k) = q(k) - q_d(k)$, $\Delta\dot{q}(k) = \dot{q}(k) - \dot{q}_d(k)$, and $\sigma(k)$, representing position and velocity errors, and the integral term, at instant k, respectively. The convex quadratic cost function $\mathcal{F}(k)$ defined as:

$$\mathcal{F}(k) = \frac{1}{2} S^T(k) S(k), \tag{19}$$

is minimized as $k \rightarrow \infty$ when:

$$\mathcal{F}(k+1) = \frac{1}{2} S^T(k+1) S(k+1) \rightarrow 0, \tag{20}$$

with $S(k+1) = \Delta\dot{q}(k+1) + \alpha\Delta q(k+1) + K_i\sigma(k+1)$. Thus, the adjustment of $\Delta\beta_i(k)$ can be defined along the negative gradient of (20) as follows:

$$\Delta\beta_i(k) = -\eta_i(k) \left[\frac{\partial \mathcal{F}(k+1)}{\partial \beta_i(k)} \right], \tag{21}$$

for $\eta_i(k) > 0$ the learning rate at instant k for the i-th LC. Applying the chain rule, the term $\partial \mathcal{F}(k+1)/\partial \beta_i(k)$ is obtained as follows:

$$\frac{\partial \mathcal{F}(k+1)}{\partial \beta_i(k)} = \frac{\partial \mathcal{F}(k+1)}{\partial S(k+1)} \cdot \frac{\partial S(k+1)}{\partial \Delta q(k+1)} \cdot \frac{\partial \Delta q(k+1)}{\partial \hat{K}_d(k)} \cdot \frac{\partial \hat{K}_d(k)}{\partial \beta_i(k)}. \quad (22)$$

Clearly,

$$\frac{\partial \mathcal{F}(k+1)}{\partial S(k+1)} = S(k+1), \quad (23)$$

$$\frac{\partial S(k+1)}{\partial \Delta q(k+1)} = \alpha, \quad (24)$$

$$\frac{\partial \hat{K}_d(k)}{\partial \beta_i(k)} = \mu_{A_i}(k), \quad (25)$$

but the term $\partial \Delta q(k+1)/\partial \hat{K}_d(k)$ cannot be computed analytically, yet, for a fast enough sampling rate (relative to the constant time of the electromechanical robot) it can be approximated reasonably by:

$$\frac{\partial \Delta q(k+1)}{\partial \hat{K}_d(k)} \approx \frac{\Delta q(k+1) - \Delta q(k)}{\hat{K}_d(k) - \hat{K}_d(k-1)} = \frac{\Delta \Delta q(k+1)}{\Delta \hat{K}_d(k)}. \quad (26)$$

Now, by design, it is desirable to make the ratio (26) sensible to parameter $\hat{K}_d(k)$ so as to $\hat{K}_d(k)$ reacts faster than $\Delta q(k+1)$ does, *i.e.* the ratio (26) is such that:

$$\frac{\partial \Delta q(k+1)}{\partial \hat{K}_d(k)} = \Upsilon_p \ll 1, \quad (27)$$

is small and positive, by design. Finally, substituting (23)–(27) into (22), (21) becomes:

$$\beta_i(k+1) = \beta_i(k) - \eta_i(k)\Upsilon_p S(k+1)\alpha \mu_{A_i}(k), \quad (28)$$

given that $\beta_i(k+1) = \beta_i(k) + \Delta\beta_i(k)$. Note that $\{\eta_i(k)\Upsilon_p\} < \eta_i(k) \in (0, 1]$ in virtue of (28), which can be interpreted as a new learning rate.

Remark 2. *On the no dependency of $\beta(k+1)$ on velocity tracking errors.* We underline that state $\dot{q}(k)$, explicit in $\Delta\dot{q}(k)$, is omitted purposely in (22) because $S(k)$ is homogeneous in $\Delta q(k)$, then when $\Delta q(k) \to 0 \Rightarrow \Delta \dot{q}(k) \to 0$, but the converse is obviously not always true. In this way, this algorithm avoids that $\hat{K}_d$ reacts to typical noise measurements of velocity tracking errors.

5. Main result

We are now in the position to state the main result with its stability analysis in the following theorem.

Theorem. *Consider the robot dynamics (1) in closed-loop with the controller (16) with a varying $\hat{K}_d$, (17). If $\lambda_m(\hat{K}_d)$ satisfies Proposition 1 and $\beta_i(k)$ is computed according to (28) then, the extended error manifold S is driven exponentially to zero. Thus, a semiglobal exponential stability of tracking errors is guaranteed, $(\Delta q, \Delta\dot{q}) \to 0$.*

Proof. Consider a Lyapunov function candidate as follows:

$$V_4(k) = \frac{1}{2} S^T(k) S(k), \quad (29)$$

whose discrete time difference yields:

$$\begin{aligned} \Delta V_4(k) &= V_4(k+1) - V_4(k) \\ &= \frac{1}{2}\{S^T(k+1)S(k+1) - S^T(k)S(k)\} \\ &= \frac{1}{2}\{[S(k) + \Delta S(k)]^T [S(k) + \Delta S(k)] \ldots - S(k)^T S(k)\} \\ &= \Delta S^T(k)[S(k) + \frac{1}{2}\Delta S(k)], \end{aligned} \quad (30)$$

for $\Delta S(k) = S(k+1) - S(k)$; owing to (26), $\Delta S(k)$ can be approximated along small variations of $\Delta\beta_i(k)$ as follows:

$$\Delta S(k) = \frac{\Delta S(k)}{\Delta \beta_i(k)} \Delta\beta_i(k) \approx \frac{\partial S(k+1)}{\partial \beta_i(k)} \Delta\beta_i(k). \quad (31)$$

Similar to (22), $\partial S(k+1)/\partial\beta_i(k)$ becomes:

$$\frac{\partial S(k+1)}{\partial \beta_i(k)} = \Upsilon_p \, \alpha \, \mu_{A_i}(k). \quad (32)$$

Finally, from (28), $\Delta\beta_i(k)$ becomes:

$$\Delta\beta_i(k) = -\eta_i(k)\Upsilon_p S(k+1)\alpha \mu_{A_i}(k). \quad (33)$$

Substituting (32) and (33), (31) can be written as follows, using $\alpha^2 = \alpha^T\alpha$ and $S(k+1) = S(k) + \Delta S(k)$,

$$\begin{aligned} \Delta S(k) &= -\eta_i(k)\Upsilon_p^2 S(k+1)\alpha^2 \mu_{A_i}^2(k), \\ &= -\eta_i(k)\Upsilon_p^2 \alpha^2 \mu_{A_i}^2(k)[S(k) + \Delta S(k)], \\ &= -\eta_i(k)\Upsilon_p^2 \alpha^2 \mu_{A_i}^2(k) S(k) - \ldots \eta_i(k)\Upsilon_p^2 \alpha^2 \mu_{A_i}^2(k)\Delta S(k), \\ &= -\overline{\mathcal{B}} S(k), \end{aligned} \quad (34)$$

where $\overline{\mathcal{B}} = \mathcal{B}/1 + \mathcal{B}$, and $\mathcal{B} = \eta_i(k)\Upsilon_p^2 \alpha^2 \mu_{A_i}^2(k)$, then $\overline{\mathcal{B}}$ is well posed since $0 < \overline{\mathcal{B}} < 1$, $\forall\ \mathcal{B} > 0$. Finally, from (30) and (34) we obtain:

$$\begin{aligned} \Delta V_4(k) &= -\overline{\mathcal{B}} S^T(k)S(k) + \frac{1}{2}\overline{\mathcal{B}}^2 S^T(k)S(k), \\ &\le -\frac{1}{2}\gamma S^T(k)S(k), \\ &\le -\gamma V_4(k), \end{aligned} \quad (35)$$

for rate $\gamma > 0$, concluding the exponential convergence of the quadratic convex function $V_4(k)$, with the unique equilibrium at $S(k) = 0$, since $\overline{\mathcal{B}} > \overline{\mathcal{B}}^2$. Resorting on definition of S in (4) and (7) and (8), it guarantees that $S(k) \to 0 \Rightarrow (\Delta q, \Delta\dot{q}) \to (0, 0)$ exponentially. ■

6. Discussions

6.1. Fuzzy reasoning and regions of attraction

Let us explain intuitively how the fuzzy reasoning is connected with the physical meaning of the attractors (also see [38,39]). An interpretation of the interplay of these gains in the workspace is established by identifying two regions of attraction, see Fig. 3, by arguing the following:

- It is clear that α and $\hat{K}_d$ have a particular impact on the applied torque $\tau = -\hat{K}_d S$ since $S = \Delta\dot{q} + \alpha\Delta q + K_i\sigma$. For large α and large Δq, S is large then $\hat{K}_d$ can be small, otherwise τ may attain impractical large values for real applications.
- Conversely, large α and small $\hat{K}_d$, for a large Δq, may result in a small applied torque, then the domain of attraction may be too large, which means large tracking errors, consequently low performance, again impractical for real applications.
- In order to modulate adequately the expansion and contraction of the domain of attractions, we have to define regions in which $\hat{K}_d$ should be automatically adjusted.

Therefore, as depicted in Fig. 3, two regions of attraction are considered: Region Ω_α based on the constant α, and region Ω_{K_d}, for a region where $\hat{K}_d$ is more influential. We have then that:

Region Ω_α: It covers the workspace far from the desired trajectory, then we design α by considering the largest admissible Δq. Notice that it enlarges the local domain of attraction to semiglobal

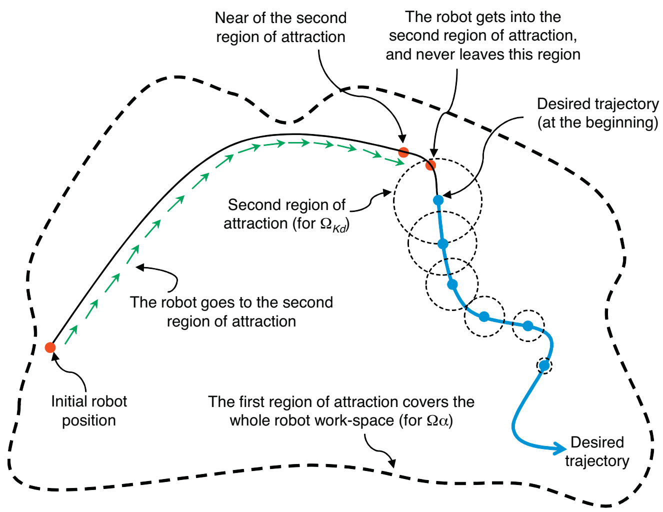

Fig. 3. Two regions of attraction splits the domain of attraction in the whole robot workspace: the outer region Ω_α, and Ω_{K_d}, which contracts the stability domain based on a time-varying $\hat{K}_d$.

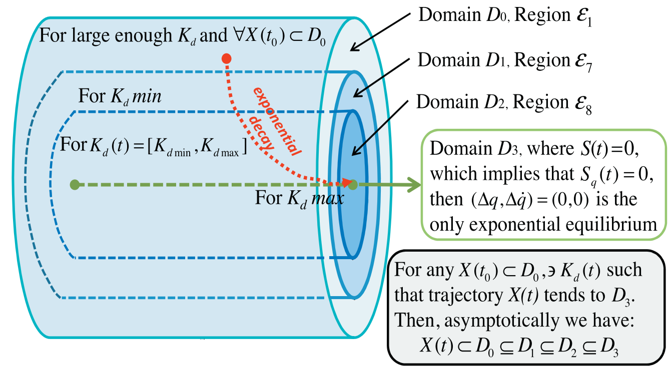

Fig. 4. The chain of implications on attraction domains shows that for any initial conditions $X(t_0)$ contained within domain $\mathcal{D}_0$, there exists a large enough K_d bounded within the range $[K_{d\min}, K_{d\max}]$ that contracts the domains $\mathcal{D}_0 \to \mathcal{D}_1 \to \mathcal{D}_2 \to \mathcal{D}_3$, which leads to $S \to 0$ exponentially, that is, enforcing semiglobal exponential tracking.

one, [15]. The constant α is defined big enough to generate a suitable torque in order to guarantee the robot converges to region Ω_{K_d}. Thus, a constant α is proposed taking also into account the maximum torque of each joint-motor pair.

Region Ω_{K_d}: This region is defined near the desired trajectory, that is, for small S. Since $S \to 0$ when $t \to \infty$ and based on **user-experience** or **fuzzy-knowledge**, then *we define* μ_{A_i} *when* $|S| < 1$ with a suitable torque mainly affected by α, and once into Ω_{K_d}, gain $\hat{K}_d$ is increased in order to reach and retain the desired trajectory in the vicinity of origin $S = 0$. Once in Ω_{K_d}, then $\hat{K}_d$ varies from $\hat{K}_{d_{\min}}$ to $\hat{K}_{d_{\max}}$ to produce exponential attraction, as established in the main theorem.

In order to represent the above transition, from Ω_α to Ω_{K_d}, we define three MFs: the first one, *before the transition* (big), the second one, *during the transition* (medium), and the third one, *after the transition* (small); as defined in Section 7.2.

Therefore, for a particular robotic arm, it is only necessary to adjust the parameter α intuitively according to the maximum torque specified by the motor manufacturer, and to design the fuzzy inference mechanism for $|S| < 1$ in order to discriminate between regions Ω_α and Ω_{K_d} by varying the value of $\hat{K}_d$, see Sections 4 and 7.2. Although $\hat{K}_d$ is bounded, strictly speaking, it does not deliver bounded torques because S is not saturated, but its contribution through the fuzzy procedure including the selection of α has the purpose of avoiding such saturation in order to produce suitable smooth torques/controllers within actuator limits, as shown in experiments of Section 7.3.

6.2. On the exponential convergence

It is shown that tuning of $\beta_i(k)$ drives bounded $S(k)$ to zero exponentially by injecting a controlled dissipation rate $\xi = S^T \hat{K}_d S$ into $\dot{V}_3$. This eventually leads to enforcing an exponential attractor at $S(k) = 0$, arising *semiglobal* stability at $(\Delta q, \Delta\dot{q}) = (0, 0)$ in the sense of [15]. This is due to the role that plays $\hat{K}_d$ on the stability criterion, in the sense of fuzzy reasoning [41]. Then by design, we can select the minimum and maximum bounds of $\hat{K}_d$, such that, $\dot{V}_3(S) = -S^T \hat{K}_d S - S^T Y_r \Theta$ leads to the following condition:

$$S^T \hat{K}_d S > |S^T Y_r \Theta|, \tag{36}$$

with a dissipation rate $\dot{\xi} = S^T \hat{K}_d S$, for a particular non-linear Euler–Lagrange dynamical plant. Thus, there exists a finite time t_i such that for initial conditions within an open set, tracking errors of the manifold S are trapped into a set-bounded $\varepsilon_7 < \varepsilon_1$, consequently enforcing the exponential contraction of domains $\mathcal{D}_0 \to \mathcal{D}_1 \to \mathcal{D}_2 \to \mathcal{D}_3$, through $\hat{K}_d$, then:

$$S \to \varepsilon_7, \quad \dot{S} \to \varepsilon_8, \quad \|S\| < \varepsilon_9, \quad \|\dot{S}\| < \varepsilon_{10}.$$

as shown in Fig. 4.

6.3. On the stability properties of the closed-loop system

Notice that when the state dependent friction, bounded smooth exogenous, or endogenous, disturbances, are not present, it implies simply that domains ε_4 and ε_5 are smaller but bounded and the result stated in the theorem applies. However, for hard non-linearities such as backlash or more stringent dynamic terms like dynamic friction that includes the Stribeck effect [37], limit cycles may be induced which are advised to take into account to avoid high frequency of $\hat{K}_d$. Notice that the sizes of neighbourhoods ε_7 and ε_8, defined around the origin $S = 0$, are based on conditions (36), which by design represent a fine tuning of the $\hat{K}_d$ based on the experience of the operator through the fuzzy-neural network.

6.4. Self-tuning with a PD controller

If there is not integral term or if $K_i = 0$, then a PD controller appears, that is, $\tau = -\hat{K}_P \Delta q - \hat{K}_D \Delta\dot{q}$. In this case, notice that domain $\mathcal{D}_2$ does not arise, in which case, larger ε_4 and ε_5 appears, which in turns demand large gains to react faster for tracking errors changes. But yet, this PD controller with the same self-tuning scheme applies, which guarantees also exponential tracking. This was explored recently in our paper [38]. It shows that the integral term induces a smaller stability domain, $\mathcal{D}_2$, with smoother $\hat{K}_d$, in part because of the contribution of the nonlinear integral term of the manifold S.

6.5. Extension to others PID-like schemes

Some others non-linear functions can be used as the error manifold S, in particular, in substitution of tanh(*) and still preserving the result of the theorem. This allows to introduce certain stability properties and requirements. Consider the family of bounded differentiable functions, possibly almost everywhere, defined in [42] as follows:

Definition 1. Consider the function $\Gamma(\mu, \rho, x)$ with $1 \geq \mu > 0$, $\rho > 0$, and $x \in \Re^n$ denotes the set of all continuous differentiable increasing functions $f(x) = [f(x_1), f(x_2), \ldots f(x_n)]^T$ such that:

- $|x| \geq |f(x)| \geq \mu|x|, \quad \forall\, x \in \Re^n : |x| < \rho$
- $\rho \geq |f(x)| \geq \mu\rho, \quad \forall\, x \in \Re^n : |x| \geq \rho$
- $1 \geq (d/dx)f(x) \geq 0,$

where $|\cdot|$ stands for the absolute value. Function $\Gamma(\mu, \rho, x)$ attains two properties needed for stability analysis in terms of the metric, those are:

1. The Euclidean norm of $f(x)$ satisfies for all $x \in \Re^n$:

$$\|f(x)\| \geq \begin{cases} \mu\|x\|, & \text{if } \|x\| < \rho \\ \mu\rho, & \text{if } \|x\| \geq \rho, \end{cases}$$

and

$$\|f(x)\| \leq \begin{cases} \|x\|, & \text{if } \|x\| < \rho \\ \sqrt{n}\rho, & \text{if } \|x\| \geq \rho. \end{cases}$$

2. The function $f(x)^T x$ satisfies for all $x \in \Re^n$:

$$f(x)^T x > \begin{cases} \mu\|x\|^2, & \text{if } \|x\| < \rho \\ \mu\rho\|x\|, & \text{if } \|x\| \geq \rho. \end{cases}$$

Some non-linear functions $f(x)$ that satisfies these properties are the following:

- Sine function,

$$f(x) = \begin{cases} \sin(x), & \text{if } |x| < \dfrac{\pi}{2} \\ 1, & \text{if } x \geq \dfrac{\pi}{2} \\ -1, & \text{if } x \leq -\dfrac{\pi}{2}. \end{cases}$$

- Saturation function,

$$f(x) = \begin{cases} +1, & \text{if } x > 1 \\ sat(x), & \text{if } x \in [-1, 1] \\ -1, & \text{if } x < 1. \end{cases}$$

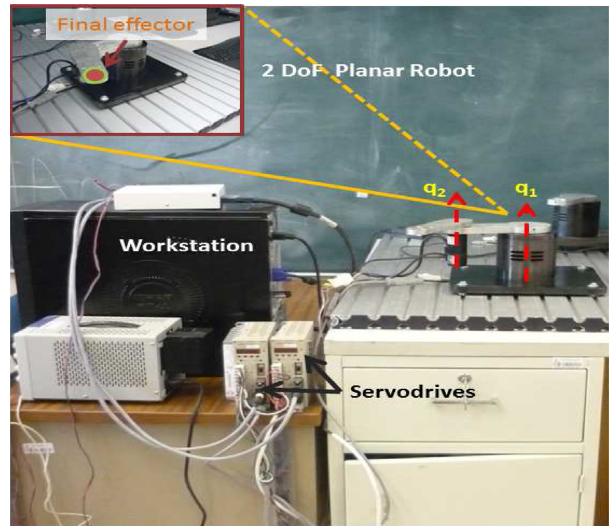

Fig. 5. Experimental setup (2-DOF robot arm).

All these function, as argument of the integral term into S, provides slightly different stability properties of $\mathcal{D}_2$. Thus, the general joint error manifold for these kind of non-linear functions is given as:

$$S = S_q + K_i \int_{t_0}^{t_f} f(\lambda S_q) d\tau,$$

where $S_q = \Delta\dot{q} + \alpha\Delta q$. Clearly, the inclusion of the integral term contributes to a fast response, even for small λ.

7. Experimental results

7.1. Experimental platform

A real-time validation is carried out on a 2-DOF dynamic robot actuated with direct-drive motors under a digital controlled current servo-loop at 2 kHz, see Fig. 5. An Intel processor 2 GHz PC-based platform in OS Windows 7 is programmed to run Labview 7.1, with 4 Gb of RAM memory. Real-time threads are implemented, including for the acquisition card *NI PCI-7041/6040E RT* to enforce

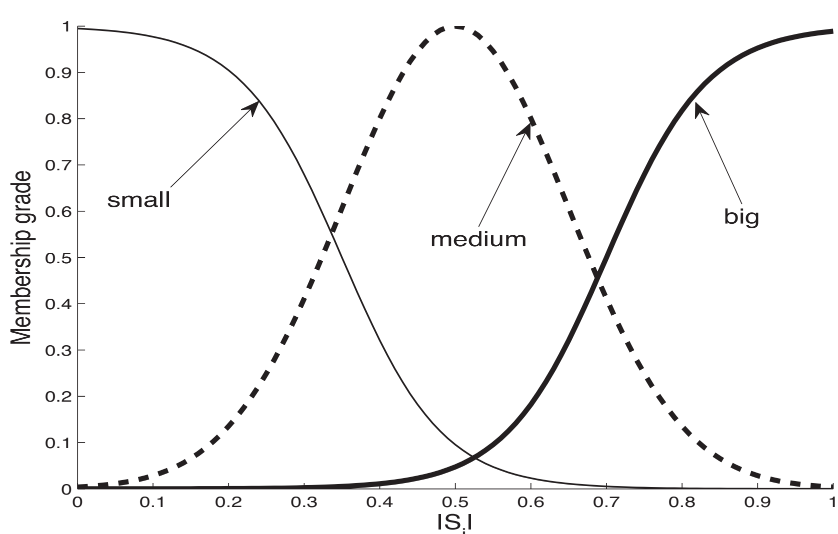

Fig. 6. Membership functions for each joint error manifold S_j corresponding to link $j = 1, 2$, with $|S_j| < 1$.

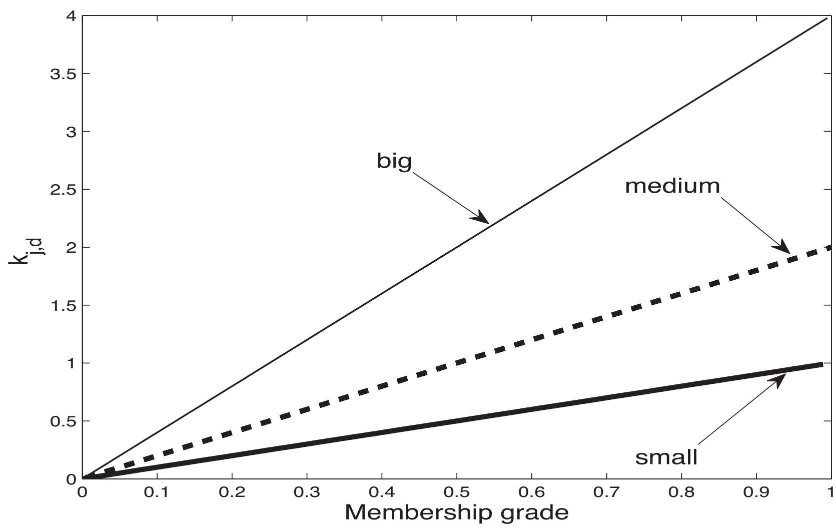

Fig. 7. Setting of linear consequences for each joint error manifold S_j corresponding to link j = 1, 2, with $(K_{d_{\min}}, K_{d_{\max}}) = (1, 4)$.

a constant 1*ms* sampling. The system is tested at first run, that is, results are presented for experiments that are congruent with theory at first trial, without an additional effort to tune it for certainly best performance, but those presented here are compliant to Proposition 1 and Theorem.

7.2. Fuzzy engine and parameters

Considering the above discussion, specifically Section 6, let us now to define a set of suitable IF-THEN rules for a robot arm of two degrees of freedom (2-DOF),

RULE 1:	IF S_j IS b	THEN $k_{j,d1} = \beta_{j,1}\, \mu_{A_{j,1}}$,
	(*Big joint error manifold*)	(*Small K_d value*)
RULE 2:	IF S_j IS m	THEN $k_{j,d2} = \beta_{j,2}\, \mu_{A_{j,2}}$,
	(*Medium joint error manifold*)	(*Medium K_d value*)
RULE 3:	IF S_j IS s	THEN $k_{j,d3} = \beta_{j,3}\, \mu_{A_{j,3}}$,
	(*Small joint error manifold*)	(*Big K_d value*)

for the link j = 1, 2. Terms b, m, and s denote big, medium, and small linguistic levels, respectively. According to the consideration about *small error surface when* $|S| < 1$, in Section 6.1, membership functions depends on three linguistic levels for each joint, as depicted in Fig. 6. After all rules have been processed, the crisp output $\hat{K_d} = diag(\hat{K}_{d1}, \hat{K}_{d2})$ is computed (where the labels d_1 and d_2 corresponds to link 1 and link 2, respectively) based on the corresponding linear consequences for each link by using defuzzification (as described in Sections 4 and 6). The setting of linear parameters $\beta_{j,i} > 0$ provide the upper bound of $\hat{K_d}$.

In order to tackle an intuitive (fuzzy) process to select suitable parameters, it is worth to point out a guide on how to do it in terms of PID gains. Considering that $\hat{K}_P(t) = \hat{K}_d(t)\alpha$, $\hat{K}_D(t) = \hat{K}_d(t)$, and $\hat{K}_I(t) = \hat{K}_d(t)K_i$ are the proportional, derivative, and integral terms, respectively; we choose the gains as follows:

- The **parameters** β are considered as the incremental product about how many times the parameters α and K_i should be incremented within the region Ω_{K_d}, providing at the same time, the lower and upper bounds of $\hat{K}_d$.
- The **integral term** is used to reduce the steady state errors, therefore, its effect must be more influential on the vicinity and within Ω_{K_d}. In order to avoid overshoots when Δq is large, based on user experience, this term should be selected small in the range $0 < K_i \leq 0.5$, for minor impact when Δq is large and major one when Δq is small, because this term increases for small errors due to the product $\hat{K}_I = \hat{K}_d(t)K_i$ through the increment of $\hat{K}_d$.
- The **proportional term** is selected big enough in accordance to the maximum torque, as established in Section 6.1. Notice that by increasing α, raising time decreases (from region Ω_α towards Ω_{K_d}) but at the expense of increasing overshoot.
- The **parameter** λ is related to the smoothness of the response of the integral part, but increasing it may causes wind-up effects, then care is advised on the numerical integrator.

Therefore, the feedback parameters for each link are defined as $\alpha = diag(5, 7.5)$, $\lambda = diag(10, 10)$, and $K_i = diag(0.2, 0.2)$. The $\beta_{j,i}$ parameters of the linear consequences for the j-*th* link are $\beta_{j,1} = 1$, $\beta_{j,2} = 2$, $\beta_{j,3} = 4$, as depicted in Fig 7.

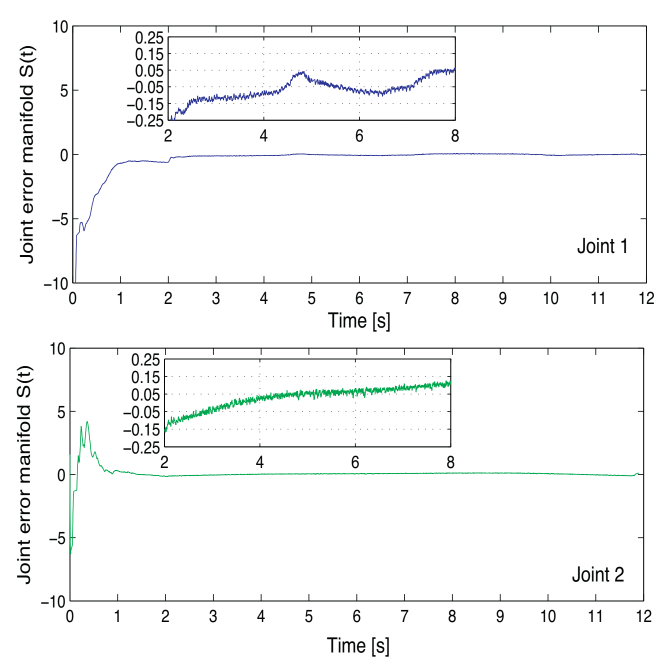

Fig. 8. Exponential convergence of manifold S for each joint. Time window shows a zooming. Exponential convergence is enforced to domain $\mathcal{D}_2 \to \mathcal{D}_3$.

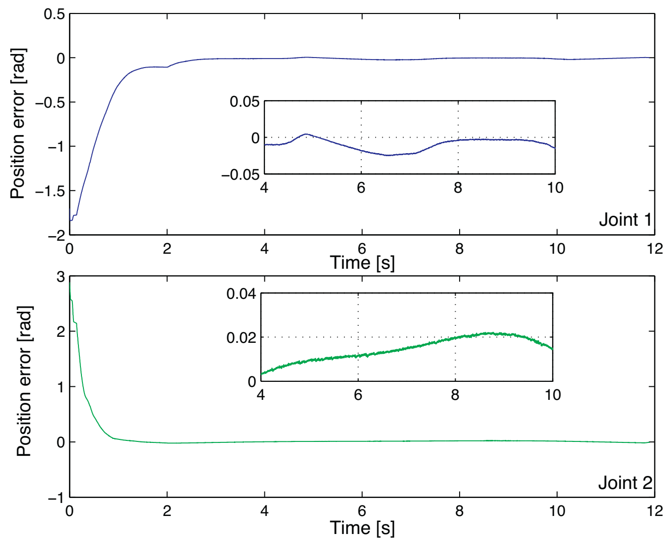

Fig. 9. Exponential position tracking errors for first (Δq_1) and second (Δq_2) joints, respectively. The time window displays a zooming, showing small tracking errors within the minimum resolution of the experimental platform.

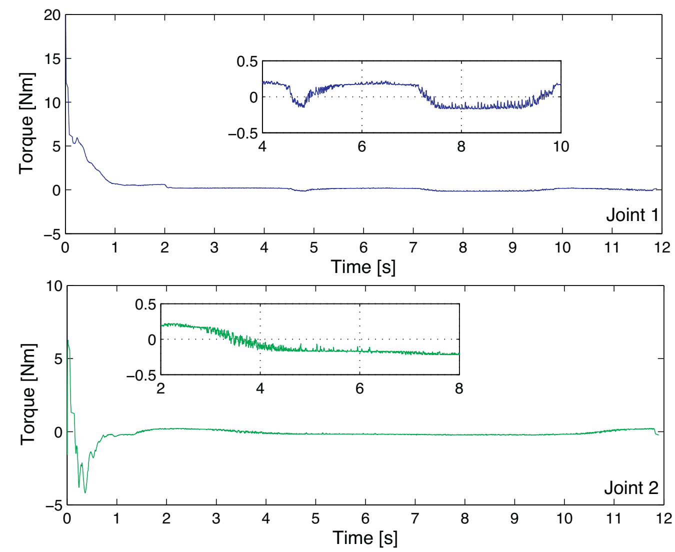

Fig. 11. Evolution of the applied torques/controllers for each joint. The time window shows a zooming, where suitable torques are maintained within small values.

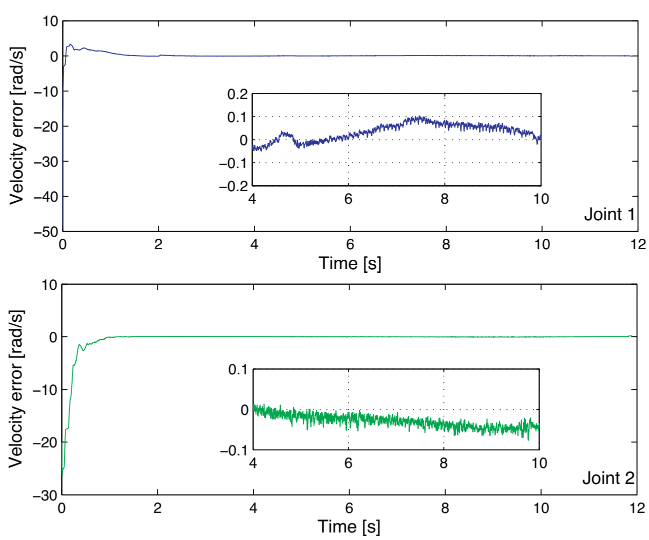

Fig. 10. Exponential velocity tracking errors for first ($\Delta \dot{q}_1$) and second ($\Delta \dot{q}_2$) joints, respectively. The time window shows a zooming.

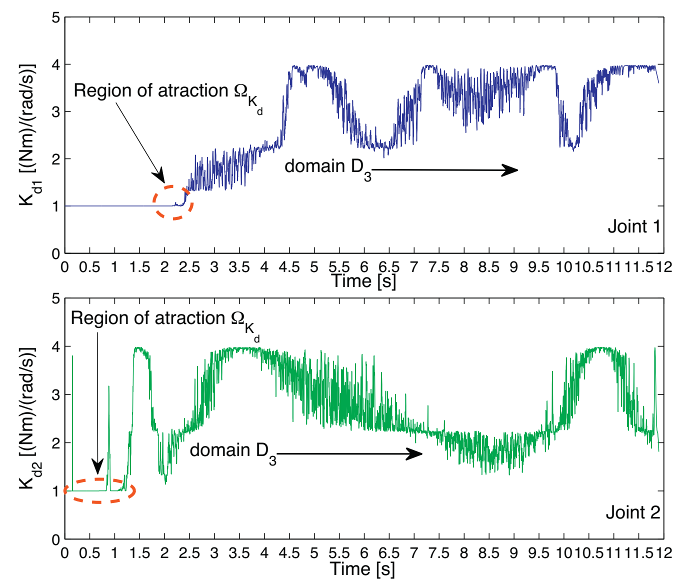

Fig. 12. Online tuning of $\hat{K}_d(t)$ for $(K_{d_{\min}}, K_{d_{\max}}) = (1, 4)$. Surprisingly, a good agreement with theory is found, once the robot gets into domain $\mathcal{D}_3$, it never leaves this region.

7.3. Results

Let us now to consider the experimental results, where data shows the expected smooth exponential tracking within the minimum resolution of the system. From Fig. 8 is shown the exponential convergence of the joint error manifold S, which is enforced to the exponential stability of domain $\mathcal{D}_2$ and subsequently to domain $\mathcal{D}_3$. Consequently, Fig. 9 shows the exponential position tracking errors, where it can be observed that tracking errors converge to zero within the minimum resolution of the experimental platform. The exponential velocity tracking errors are shown in Fig. 10. Signals are bounded within the best minimum physical precision range of the 12 bits digital encoders and bounded within the physical precision range considering the Levant velocity estimation. Fig. 11 shows the smooth evolution of the applied torques/controllers for each joint, remaining within a suitable value according to each motor limits. Finally, in Fig. 12 is shown the time-varying feedback gains $\hat{K}_d(t)$ for each joint, in which can be observed the change of domains of attraction and that once the robot gets into the second region of attraction Ω_{K_d}, it never leaves this region. Therefore, experimental results prove semiglobal exponential tracking, demonstrating consistency with proposed theoretical framework.

7.4. Discussions

The underlying assumption of this self-tuning PID scheme is that the expert user knows bounds of the particular set of gains for different operational performance, a typical assumption on fuzzy-based control. Then, setting this as the minimum and maximum appropriate gains, the system automatically tunes the appropriate $\hat{K}_d(t)$ gain to guarantee exponential tracking. When this is not the case, that is, when tuning is deficient for the maximum gain, the system will be only GUUB (Proposition 1), since by assumption of minimum gain yields $\mathcal{D}_0$ at least, but will not attain tracking (Theorem). In practice, such tuning in fact tunes on-line the dissipation rate to reinforce a dissipative mapping from torque input to manifold S, and when S_q is present, integral gain creates a smaller stability domain $\mathcal{D}_3$, which in turns requires a smaller maximum K_d. Trial-and-error is useful to improve the training user, and to satisfy with conditions of Theorem, to guarantee a semiglobal practical exponential stability and an alternative for tracking in practical applications. Comparative studies of the proposed approach, through simulations and real-time experiments, with respect to traditional PD controller and considering different PID-like structures are presented in [38,39].

8. Conclusion

The elusive problem of tracking in robotic arms has studied in the realm of PID control with non-constants feedback gains. A self-tuning controller is proposed which preserves a unique PID structure, with the ability to track smooth references by tuning the single dissipation rate gain, using knowledge-based rules. Consequently, as it is customary in fuzzy-based controllers, it assumes implicitly that the user knows bounds of feedback gains according to specifications. The role of the self-tuning gain is to enforce dissipativeness so as to dissipative terms dominate the unknown robot dynamics and an unique exponential equilibrium is obtained at the given smooth desired trajectories. Other PID-like schemes can be used, in fact, some were discussed which may provide different performance metrics. It provides not only a flexible but also a practical control structure, in which human/user experience can be incorporated into the closed-loop of control. The experimental study shows the feasibility of the proposed scheme where the easiness of the implementation and tuning is highlighted in virtue of null-knowledge of robot dynamics. Finally, it is important to stress, that for a particular robotic arm once the tuning process have been carried out, there is no need to tune it again due to the intelligent self-tuned algorithm. For industrial applications, as robot manipulators, once the controller is tuned then a set of trajectory tasks can be carried out within the workspace of the robot. Therefore, the industrial robot could be tuned by the manufacturer directly, avoiding the tuning process by the user in the industrial floor.

References

[1] R.E. Brown, G.N. Maliotis, J.A. Gibby, PID Self-tuning Controller for aluminum rolling mill, IEEE Trans. Ind. Electron. 29 (3) (1993) 578–583.
[2] V. Mummadi, Design of robust digital PID controller for H-bridge soft-switching boost converter, IEEE Trans. Ind. Electron. 58 (7) (2011) 2883–2897.
[3] S. Skoczowski, S. Domek, K. Pietrusewicz, B. Broel-Plater, A method for improving the robustness of PID control, IEEE Trans. Ind. Electron. 52 (6) (2005) 1669–1676.
[4] K.H. Ang, G. Chon, Y. Li, PID control system analysis, design, and technology, IEEE Trans. Control Syst. Technol. 13 (4) (2005) 559–576.
[5] J. Alvarez-Ramirez, R. Kelly, I. Cervantes, Semiglobal stability of saturated linear PID control for robot manipulators, Automatica 39 (6) (2003) 989–995.
[6] Y. Li, K.-H. Ang, G.-C.-Y. Chong, PID control system, analysis and design, problems, remedies, and future directions, IEEE Control Syst. Mag. 26 (1) (2006) 32–41.
[7] A.Ch. Huang, M.Ch. Chien, Adaptive Control of Robot Manipulators: A Unified Regressor-free Approach, World Scientific Pub, Hackensack, NJ, 2010.
[8] Y. Karayiannidis, Z. Doulgeri, Regressor-free robot joint position tracking with prescribed performance guarantees, in: Proc. IEEE Int. Conf. Robot. Biomim., 2011, pp. 2312–2317.
[9] A. Loria, H. Nijmeijer, Bounded output feedback tracking control of fully-actuated Euler-Lagrange systems, Syst. Control Lett. 33 (16) (1998) 151–161.
[10] R.K. Mudi, N.R. Pal, A robust self-tuning scheme for PI- and PD-type fuzzy controllers, IEEE Trans. Fuzzy Syst. 7 (1) (1999) 2–16.
[11] A. Chaillet, A. Loria, Uniform Semiglobal Asymptotic for Time-varying Nonlinear Cascade Systems, vol. 16, Part 1, IFAC World Congress, 2005, pp. 681.
[12] E. Kwan, M. Liu, An adaptive fuzzy approach for robot manipulators tracking, in: IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1999, pp. 53–58.
[13] J.J.E. Slotine, W. Li, On the adaptive control of robot manipulators, Int. J. Robot. Res. 6 (3) (1987) 49–59.
[14] J.L. Meza, V. Santibañez, R. Soto, M.A. Llama, Fuzzy self-tuning PID semiglobal regulator for robot manipulators, IEEE Trans. Ind. Electron. 59 (6) (2012) 2709–2717.
[15] A. Loria, E. Lefeber, H. Nijmeijer, Global asymototic stability of robot manipulators with linear PID and PI2D control, Stab. Control: Theor. Appl. 3 (2) (2000) 138–149.
[16] C. Treesatayapun, Fuzzy rules emulated network and its application on nonlinear control systems, Appl. Soft Comput. 8 (2008) 996–1004.
[17] C. Treesatayapun, S. Uatrongjit, Adaptive controller with fuzzy rules emulated structure and its applications, Eng. Appl. Artif. Intell. 18 (2005) 603–615.
[18] J. Armendariz, C. Treesatayapun, A. Baltazar, Force feedback controller based on fuzzy-rules emulated networks and hertzian contact with ultrasound, Mech. Syst. Signal Process. 27 (2012) 534–550.
[19] V. Parra-Vega, S. Arimoto, Y.H. Liu, G. Hirzinger, P. Akella, Dynamic sliding PID control for tracking of robot manipulators: theory and experiments, IEEE Trans. Robot. Autom. 19 (6) (2003) 967–976.
[20] S. Arimoto, F. Miyazaki, Stability and robustness of PID feedback control for robot manipulators of sensory capability, in: Robotics Research First International Symposium, MIT press, Cambridge, MA, 1984.
[21] S. Kawamura, F. Miyazaki, S. Arimoto, Is a local linear PD feedback control law effective for trajectory tracking of robot motion? IEEE Int. Conf. Robot. Autom. (1988) 1335–1340.
[22] M.A. Llama, R. Kelly, V. Santibañez, Stable computed-torque control of robot manipulators via fuzzy self-tuning, IEEE Trans. Syst. Man Cyber. 30 (1) (2000) 143–150.
[23] Z.Y. Zhao, M. Tomizuka, S. Isaka, Fuzzy gain scheduling of PID controllers, IEEE Trans. Syst. Man Cyber. 23 (5) (1993) 1392–1398.
[24] M. Rezaei, A. Gharaveisi, A.A. Rezaei, PID parameter selection based on iterative learning control, Contemp. Eng. Sci. 4 (5) (2011) 201–220.
[25] J.C. Shen, Fuzzy neural networks for tuning PID controller for plants with underdamped responses, IEEE Trans. Fuzzy Syst. 9 (2) (2001) 333–342.
[26] K.S. Tang, K.F. Man, G. Chen, S. Kwong, An optimal fuzzy PID controller, IEEE Trans. Ind. Electron. 48 (4) (2001) 757–765.
[27] V. Santibañez, R. Kelly, A class of non-linear PID global regulators for robot manipulators, IEEE Int. Conf. Robot. Autom., Leuven, Belgium (1998).
[28] F.L. Lewis, A. Yessildirek, K. Liu, Multilayer neural-net robot controller with guaranteed tracking performance, IEEE Trans. Neural Netw. 7 (2) (1996) 388–399.
[29] W. Yu, J. Rosen, Neural PID control of robot manipulators with application to an upper limb exoskeleton, IEEE Trans. Syst. Man Cybern. 43 (2) (2013) 673–684.
[30] J. Wang, C. Zhang, Y. Jing, D. An, Study of neural network PID control in variable-frequency air-conditioning system, in: IEEE Int. Conf. Control Autom., ICCA, Guangzhou, 2007, pp. 317–322.
[31] S. Parvez, G. Zhiqiang, A wavelet-based multiresolution PID controller, IEEE Trans. Ind. Appl. 41 (2) (2005) 537–543.
[32] S. Cong, Y. Liang, PID-like neural network non-linear adaptive control for uncertain multivariable motion control systems, IEEE Trans. Ind. Electron. 56 (10) (2009) 3872–3879.
[33] Y. Gao, M.J. Er, S. Yang, Adaptive control of robot manipulators using fuzzy neural network, IEEE Trans. Ind. Electron. 48 (6) (2001) 1274–1278.
[34] W.Z. Qiao, M. Mizumoto, PID type fuzzy controller and parameters adaptive method, Fuzzy Sets Syst. 78 (1) (1996) 23–35.
[35] A.R.N. Ravari, H.D. Taghirad, A novel hybrid fuzzy-PID controller for tracking control of robot manipulators, in: IEEE Int. Conf. Robot. Biomim. ROBIO, Bangkok, 2008, pp. 1625–1630.
[36] Yongming Li, Shaocheng Tong, Tieshan Li, Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping, Nonlinear Anal.: Real World Appl. 14 (1) (2013) 483–494.
[37] C. Canudas de Wit, H. Olsson, K.J. Astrom, P. Lischinsky, A new model for control of systems with friction, IEEE Trans. Autom. Control 40 (3) (1995) 415–419.
[38] J. Armendariz, V. Parra-Vega, R. Garcia-Rodriguez, T. Hirai, Dynamic self-tuning PD control for tracking of robot manipulators, IEEE Conf. Decis. Control, Hawaii (2012).
[39] J. Armendariz, V. Parra-Vega, R. Garcia-Rodriguez, T. Hirai, S. Rosales, F.A. Machorro-Fernandez, Regressor-free tracking of robots with self-tuning PD-like control, IEEE Int. Conf. Robot. Autom., Germany (2013).
[40] F.L. Lewis, C.T. Abdallah, D.M. Dawson, Control of Robot Manipulators, Macmillan, New York, 1993.
[41] L.K. Wong, H.F. Frank, Leung, K.S. Peter, Tam, Lyapunov-function-based design of fuzzy logic controllers and its application on combining controllers, IEEE Trans. Ind. Electron. 45 (3) (1998) 502–509.
[42] R. Kelly, Global positioning of robot manipulators via PD control plus a class of non-linear integral actions, IEEE Trans. Autom. Control 43 (7) (1998) 934–938.