Neural PID Control of Robot Manipulators With Application to an Upper Limb Exoskeleton

Wen Yu, *Senior Member, IEEE*, and Jacob Rosen

***Abstract*—In order to minimize steady-state error with respect to uncertainties in robot control, proportional–integral–derivative (PID) control needs a big integral gain, or a neural compensator is added to the classical proportional–derivative (PD) control with a large derivative gain. Both of them deteriorate transient performances of the robot control. In this paper, we extend the popular neural PD control into neural PID control. This novel control is a natural combination of industrial linear PID control and neural compensation. The main contributions of this paper are semiglobal asymptotic stability of the neural PID control and local asymptotic stability of the neural PID control with a velocity observer which are proved with standard weight training algorithms. These conditions give explicit selection methods for the gains of the linear PID control. An experimental study on an upper limb exoskeleton with this neural PID control is addressed.**

***Index Terms*—Exoskeleton, neural networks, PID, robot.**

I. Introduction

PROPORTIONAL–integral-derivative (PID) control is widely used in industrial robot manipulators [1]. In the absence of robot knowledge, a PID controller may be the best controller because it is model free and its parameters can be adjusted easily and separately [2]. However, an integrator in a PID controller reduces the bandwidth of the closed-loop system. In order to remove steady-state error caused by uncertainties and noise, the integrator gain has to be increased. This leads to worse transient performance and even destroys the stability. Therefore, many robot manipulators use pure proportional–derivative (PD) control or PD control with a small integral gain [3].

It is known that a PD controller can guarantee stability (bounded) of a robot manipulator in regulation case. However, asymptotic stability is not achieved when the manipulator dynamics contain gravitational torque vector and friction. From control viewpoint, this steady-state error can be removed by introducing an integral component to the PD control. It is PID control. Besides the transient performance and stability problems of the integrator, theory analysis is also difficult for industrial linear PID control. In order to ensure asymptotic stability of the PID control, a popular method is to modify the linear PID into nonlinear one. For example, the position error was modified into nonlinear form in [4]. The integral term was saturated by a nonlinear function in [5]. The input was saturated in [6]. An extra integral term in the filtered position was added in [7]. The variable structure control and neural control were combined with the classic PID control in [8] and [9]. Only a few researchers worked on the linear PID. The stability (not asymptotic stability) of the linear PID control was proved in [10], where the robot dynamic was rewritten in a decoupled linear system and a bounded nonlinear system. In [11], asymptotic stability of linear PID was proved; however, conditions for linear PID gains are not explicit.

Model-based compensation with PD control is an alternative method for PID control [2], such as adaptive gravity compensation [12], Lyapunov-based compensation [14], desired gravity compensation [11], and PD+ with position measurement [13]. They all needed structure information of the robot gravity. Some nonlinear PD controllers can also achieve asymptotic stability, for example, PD control with time-varying gains [15], PD control with nonlinear gains [16], and PD control with sliding-mode compensation [8]. However, these controllers are complex, and many good properties of the linear PID control do not exist.

Intelligent compensation for PD control does not need mathematical model since it is a model-free compensator. It can be classified into fuzzy compensator [17], fuzzy PID [18], neural compensator [19], and fuzzy–neural compensator [20], [21]. The basic idea behind these controllers is to use a filtered tracking error in the Lyapunov-based analysis [3]. By proper weight tuning algorithms, which are similar with robust adaptive control methods [22], the derivative of the Lyapunov function is negative, as long as the filtered tracking error is outside of the ball with radius B/K_v; here, B is the upper bound of all unknown uncertainties, and K_v is the derivative gain in PD control. These neural PD controllers are uniformly ultimately bounded, and tracking errors decrease with increasing the gain K_v. The cost of large K_v is that the transient performance becomes slow. Only when $K_v \to \infty$ that the tracking error converges to zero [31].

It is well known that the simplest method to decrease the tracking error is to add an integral action, i.e., changing the neural PD control into neural PID control. A natural question is as follows: Why do we not add an integrator instead of increasing derivative gain in the neural PD control?

Manuscript received June 23, 2011; revised May 24, 2012; accepted August 7, 2012. Date of publication September 28, 2012; date of current version April 16, 2013. This paper was recommended by Associate Editor R. Selmic.

W. Yu is with the Departamento de Control Automatico, CINVESTAV-IPN, Mexico City 07360, Mexico (e-mail: yuw@ctrl.cinvestav.mx).

J. Rosen is with the Department of Computer Engineering, University of California—Santa Cruz, Santa Cruz, CA 95064, USA (e-mail: jrosen1@ucsc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2012.2214381

2168-2267/$31.00 © 2012 IEEE

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

There are two different approaches to combine PID control with the intelligent control, such as neural control. The first one is that neural networks are formed into PID structure [23]–[25]. By proper updating laws, the parameters of PID controllers are changed such that the closed-loop systems are stable. They are not real industrial PID controllers, because the PID gains (weights of the neural networks) are time varying. The second method is that intelligent techniques are used to tune the parameters of PID controllers, such as fuzzy tuning [26], neural tuning [27], [28], and expert tuning [29]. The controllers are still industrial linear PID; however, the stability of closed-loop system is not guaranteed. The neural PID control of this paper overcomes the aforementioned disadvantages. It is an industrial linear PID controller adding a neural compensator. The main obstacle of this neural PID is theoretical difficulty in analyzing the stability. Even for linear PID, it is not easy to prove asymptotic stability [11]. Without theoretical guarantee for this neural PID control, industrial applications cannot be carried out safely. From the best of our knowledge, theory analysis for this neural PID control is still not published.

In this paper, the well-known neural PD control of robot manipulators is extended to the neural PID control. The semiglobal asymptotic stability of this novel neural control is proved. Explicit conditions for choosing PID gains are given. When the measurement of velocities it is not available, local asymptotic stability is also proved with a velocity observer. Unlike the other neural controllers of robot manipulators, our neural PID does not need big derivative and integral gains to assure asymptotic stability. We apply this new neural control to a 7-DOF exoskeleton robot at the University of California—Santa Cruz (UCSC). Experimental results show that this neural PID control has many advantages over classical PD/PID control, the neural PD control, and the other neural PID control.

II. Semiglobal Asymptotic Stability of Neural PID Control

Many industrial rigid robots (without flexible links and high-frequency joint dynamics) can be expressed in the Lagrangian form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = u \tag{1}$$

where $q \in R^n$ represents the link positions, $M(q)$ is the inertia matrix, $C(q,\dot{q}) = \{c_{kj}\}$ represents centrifugal force, $G(q)$ is a vector of gravity torques, and $F(\dot{q})$ is friction. All terms $M(q), C(q,\dot{q}), G(q)$, and $F(\dot{q})$ are unknown. $u \in R^n$ is control input. The friction $F(\dot{q})$ is represented by the Coulomb friction model

$$F(\dot{q}) = K_{f1}\dot{q} + K_{f2}\tanh(k_{f3}\dot{q}) \tag{2}$$

where k_{f3} is a large positive constant, such that $\tanh(k_{f3}\dot{q})$ can approximate $\mathrm{sign}(\dot{q})$, and K_{f1} and K_{f2} are positive coefficients. In this paper, we use a simple model for the friction as in [3] and [11]

$$F(\dot{q}) = K_{f1}\dot{q}. \tag{3}$$

When $G(q)$ and $F(\dot{q})$ are unknown, we may use a neural network to approximate them as

$$\begin{aligned} f(q,\dot{q}) &= G(q) + F(\dot{q}) \\ \hat{f}(q,\dot{q}) &= \widehat{W}\sigma(q,\dot{q}) \quad f(q,\dot{q}) = W^*\sigma(q,\dot{q}) + \phi(q) \end{aligned} \tag{4}$$

where W^* is unknown constant weight, $\widehat{W}$ is estimated weight, $\phi(q,\dot{q})$ is the neural approximation error, and σ is a neural activation function; here, we use Gaussian function such that $\sigma(q,\dot{q}) \geq 0$.

Since the joint velocity $\dot{q}$ is not always available, we may use a velocity observer which will be discussed in Section III to approximate it. This linear-in-the-parameter net is the simplest neural network. According to the universal function approximation theory, the smooth function $f(q,\dot{q})$ can be approximated by a multilayer neural network with one hidden layer in any desired accuracy provided proper weights and hidden neurons

$$\hat{f}(q,\dot{q}) = \widehat{W}\sigma\left(\hat{V}[q\dot{q}]\right) \quad f(q) = W^*\sigma\left(V^*[q\dot{q}]\right) + \phi(q,\dot{q}) \tag{5}$$

where $\widehat{W} \in R^{n\times m}$, $\hat{V} \in R^{m\times n}$, m is hidden node number, and $\hat{V}$ is the weight in hidden layer. In order to simplify the theory analysis, we first use linear-in-the-parameter net (4); then, we will show that the multilayer neural network (5) can also be used for the neural control of robot manipulators. The robot dynamics (1) have the following standard properties [2] which will be used to prove stability.

P1. The inertia matrix $M(q)$ is symmetric positive definite, and

$$0 < \lambda_m\{M(q)\} \leq \|M\| \leq \lambda_M\{M(q)\} \leq \beta, \qquad \beta > 0 \tag{6}$$

where $\lambda_M\{M\}$ and $\lambda_m\{M\}$ are the maximum and minimum eigenvalues of the matrix M.

P2. For the centrifugal and Coriolis matrix $C(q,\dot{q})$, there exists a number $k_c > 0$ such that

$$\|C(q,\dot{q})\dot{q}\| \leq k_c\|\dot{q}\|^2, \qquad k_c > 0 \tag{7}$$

and $\dot{M}(q) - 2C(q,\dot{q})$ is skew symmetric, i.e.,

$$x^{\mathrm{T}}\left[\dot{M}(q) - 2C(q,\dot{q})\right]x = 0 \tag{8}$$

also

$$\dot{M}(q) = C(q,\dot{q}) + C(q,\dot{q})^{\mathrm{T}}. \tag{9}$$

P3. The neural approximation error $\phi(q,\dot{q})$ is Lipschitz over q and $\dot{q}$

$$\|\phi(x) - \phi(y)\| \leq k_\phi\|x - y\|. \tag{10}$$

From (4), we know that

$$G(q) + F(\dot{q}) = W^*\sigma(q,\dot{q}) + \phi(q,\dot{q}). \tag{11}$$

Because $G(q)$ and $F(\dot{q})$ satisfy Lipschitz condition, **P3** is established.

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

In order to simplify calculation, we use the simple model for the friction as in (3), and the lower bound of $\int \phi(q)dq$ can be estimated as

$$\int_0^t \phi(q,\dot{q})dq = \int_0^t G(q)dq + \int_0^t F(\dot{q})dq - \int_0^t W^*\sigma(q)dq \tag{12}$$

where $U(q_t)$ is the potential energy of the robot and $\partial U/\partial q = G(q)$. Since $\sigma(\cdot)$ is a Gaussian function, $W^*\sigma(q) > 0$. By $U(q_t) > 0$

$$\int_0^t \phi(q,\dot{q})dq > K_{f1}q_t - K_{f1}q_0 - \frac{1}{2}\sqrt{\pi}W^*$$

where $\int_0^t \sigma(q) = (1/2)\sqrt{\pi}\mathrm{erf}(q)$. Since the workspace of a manipulator (the entire set of points reachable by the manipulator) is known, $\min\{q_t\}$ can be estimated. We define the lower bound of $\int_0^t \phi(q)dq$ as

$$k_\phi = K_{f1}\min\{q_t\} - K_{f1}q_0 - \frac{1}{2}\sqrt{\pi}W^*. \tag{13}$$

Given a desired constant position $q^d \in R^n$, the objective of robot control is to design the input torque u in (1) such that the regulation error

$$\tilde{q} = q^d - q \tag{14}$$

$\tilde{q} \to 0$ and $\dot{\tilde{q}} \to 0$ when initial conditions are in arbitrarily large domain of attraction.

The classical industrial PID law is

$$u = K_p\tilde{q} + K_i\int_0^t \tilde{q}(\tau)d\tau + K_d\dot{\tilde{q}} \tag{15}$$

where K_p, K_i, and K_d are proportional, integral, and derivative gains of the PID controller, respectively.

When the unknown dynamic $\|f(q,\dot{q})\|$ in (4) is big, in order to assure asymptotic stability, the integral gain K_i has to be increased. This may cause big overshoot, bad stability, and integrator windup. Model-free compensation is an alternative solution, where $f(q)$ is estimated by a neural network as in (4). Normal neural PD control is [3]

$$u = K_p\tilde{q} + K_d\dot{\tilde{q}} + \hat{f} \tag{16}$$

where $\hat{f}(q,\dot{q}) = \widehat{W}\sigma(q,\dot{q})$. With the filtered error $r = \tilde{q} + \Lambda\dot{\tilde{q}}$, (16) becomes

$$u = K_v r + \hat{f}. \tag{17}$$

The control (17) avoids integrator problems in (15). Unlike industrial PID control, they cannot reach asymptotic stability. The stability condition of the neural PD control (16) is $\|r\| > B/K_v$, where B is a constant [32]. In order to decrease $\|r\|$, K_d has to be increased. This causes long settling time problem. The asymptotic stability ($r \to 0$) requires $K_v \to \infty$.

In this paper, an integrator is added into the normal neural PD control (16), and it has a similar form as the industrial PID in (15)

$$u = K_p\tilde{q} + K_d\dot{\tilde{q}} + K_i\int_0^t \tilde{q}(\tau)d\tau + \hat{f}. \tag{18}$$

Because, in regulation case, $\dot{q}^d = 0$, $\dot{\tilde{q}} = -\dot{q}$, the PID control law can be expressed via the following equations:

$$\begin{aligned} u &= K_p\tilde{q} - K_d\dot{q} + \xi + \widehat{W}\sigma(q,\dot{q}) \\ \dot{\xi} &= K_i\tilde{q} \quad \xi(0) = \xi_0. \end{aligned} \tag{19}$$

We require that the PID control part of (19) is decoupled, i.e., K_p, K_i, and K_d are positive definite diagonal matrices. The closed-loop system of the robot (1) is

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tilde{f}(q,\dot{q}) &= K_p\tilde{q} - K_d\dot{q} + \xi \\ \dot{\xi} &= K_i\tilde{q} \end{aligned} \tag{20}$$

where $\tilde{f} = f - \hat{f}$

$$\tilde{f} = W^*\sigma(q) + \phi(q) - \widehat{W}\sigma(q) = \tilde{W}\sigma(q) + \phi(q) \tag{21}$$

where $\tilde{W} = W^* - \widehat{W}$. In matrix form, the closed-loop system is

$$\frac{d}{dt}\begin{bmatrix} \xi \\ \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} K_i\tilde{q} \\ -\dot{q} \\ \ddot{q}^d + M^{-1}\begin{pmatrix} C\dot{q} + \tilde{W}\sigma(q) + \phi(q,\dot{q}) \\ -K_p\tilde{q} + K_d\dot{q} - \xi \end{pmatrix} \end{bmatrix}. \tag{22}$$

The equilibrium of (22) is $[\xi, \tilde{q}, \dot{\tilde{q}}] = [\xi^*, 0, 0]$. Since at equilibrium point $q = q^d$ and $\dot{q}^d = 0$, the equilibrium is $[\phi(q^d), 0, 0]$. We simplify $\phi(q^d, 0)$ as $\phi(q^d)$.

In order to move the equilibrium to origin, we define

$$\tilde{\xi} = \xi - \phi(q^d). \tag{23}$$

The final closed-loop equation becomes

$$\begin{aligned} &M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tilde{W}\sigma(q,\dot{q}) + \phi(q,\dot{q}) \\ &\quad = K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} + \phi(q^d) \\ &\dot{\tilde{\xi}} = K_i\tilde{q}. \end{aligned} \tag{24}$$

The following theorem gives the stability analysis of the neural PID control. From this theorem, we can see how to choose the PID gains and how to train the weight of the neural compensator in (19). Another important conclusion is that the neural PID control (19) can force the error $\tilde{q}$ to zero.

Theorem 1: Consider robot dynamic (1) controlled by the neural PID control (19); the closed-loop system (24) is

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

semiglobally asymptotically stable at the equilibrium $x = [\xi - \phi(q^d), \tilde{q}, \dot{\tilde{q}}]^{\mathrm{T}} = 0$, provided that control gains satisfy

$$\begin{aligned}\lambda_m(K_p) &\geq \frac{3}{2}k_\phi \\ \lambda_M(K_i) &\leq \beta\frac{\lambda_m(K_p)}{\lambda_M(M)} \\ \lambda_m(K_d) &\geq \beta + \lambda_M(M)\end{aligned} \tag{25}$$

where $\beta = \sqrt{\lambda_m(M)\lambda_m(K_p)/3}$, k_ϕ satisfies (10), and the weight of the neural networks (4) is tuned by

$$\dot{\hat{W}} = -K_w\sigma(q,\dot{q})(\dot{q} + \alpha\tilde{q})^{\mathrm{T}} \tag{26}$$

where α is positive design constant, it satisfies

$$\frac{\sqrt{\frac{1}{3}\lambda_m(M)\lambda_m(K_p)}}{\lambda_M(M)} \geq \alpha \geq \frac{3}{\lambda_m\left(K_i^{-1}\right)\lambda_m(K_p)}. \tag{27}$$

Proof: See the Appendix. ■

Remark 1: From the aforementioned stability analysis, we see that the gain matrices of the neural PID control (19) can be chosen directly from the conditions (25). The tuning procedure of the PID parameters is more simple than that in [4], [6], [11], [10], and [16]. No modeling information is needed. The upper or lower bounds of PID gains need the maximum eigenvalue of M in (25), and it can be estimated without calculating M. For a robot with only revolute joints [2]

$$\lambda_M(M) \leq \beta, \qquad \beta \geq n\left(\max_{i,j}|m_{ij}|\right) \tag{28}$$

where m_{ij} stands for the ijth element of M $M \in R^{n\times n}$. A β can be selected such that it is much bigger than all elements.

Remark 2: The main difference between our neural PID control and the other neural PD controllers is that the stability condition is changed and we require the regulation error

$$\|\tilde{q}\| < k_1\lambda_M(K_i)\lambda_m(K_p). \tag{29}$$

The other neural PD controllers need

$$\|\tilde{q}\| > \frac{k_2}{K_v} \tag{30}$$

where k_1 and k_2 are positive constants. Obviously, if the initial condition is not worse and satisfies (29), (29) is always satisfied, and $\|\tilde{q}\|$ will decrease to zero. However, (30) cannot be satisfied when $\|\tilde{q}\|$ becomes small, so K_v has to be increased.

Remark 3: If the unknown $f(q)$ is estimated by the multilayer neural network (5), the modeling error (21) becomes

$$\begin{aligned}\tilde{f} =& f - \hat{f} = W^*\sigma\left(V^*[q\dot{q}]\right) + \phi(q,\dot{q}) - \widehat{W}\sigma\left(\hat{V}[q\dot{q}]\right) \\ =& \tilde{W}\sigma\left(\hat{V}[q\dot{q}]\right) - W^*\sigma\left(\hat{V}[q\dot{q}]\right) + W^*\sigma\left(V^*[q\dot{q}]\right) + \phi(q,\dot{q}) \\ =& \tilde{W}\sigma\left(\hat{V}[q\dot{q}]\right) + W^*\sigma'\tilde{V}[q\dot{q}] + \epsilon_1 + \phi(q,\dot{q}) \\ =& \tilde{W}\left[\sigma\left(\hat{V}[q\dot{q}]\right) + \sigma'\tilde{V}[q\dot{q}]\right] + \widehat{W}\sigma'\tilde{V}[q\dot{q}] + \phi_1(q,\dot{q})\end{aligned} \tag{31}$$

where $\phi_1(q) = \epsilon_1 + \phi(q,\dot{q})$ and ϵ_1 is Taylor approximation error. The closed-loop equation (24) becomes

$$\begin{aligned}&M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tilde{W}\left\{\sigma\left(\hat{V}[q\dot{q}]\right) + \sigma'\tilde{V}[q\dot{q}]\right\} \\ &\quad + \widehat{W}\sigma'\tilde{V}[q\dot{q}] + \phi_1(q,\dot{q}) \\ &\quad = K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} + \phi(q^d) \\ &\dot{\tilde{\xi}} = K_i\tilde{q}.\end{aligned} \tag{32}$$

If the Lyapunov function in (54) is changed as

$$V_m = V + \frac{1}{2}\mathrm{tr}\left(\widetilde{V}^{\mathrm{T}}K_v^{-1}\widetilde{V}\right) \tag{33}$$

then the derivative of (33) is

$$\begin{aligned}\dot{V}_m =& \dot{V} - \dot{q}^{\mathrm{T}}\tilde{W}\sigma\left(q[q\dot{q}]\right) \\ &+\dot{q}^{\mathrm{T}}\tilde{W}\left[\sigma\left(\hat{V}[q\dot{q}]\right) + \sigma'\tilde{V}[q\dot{q}]\right] + \mathrm{tr}\left(\widetilde{V}^{\mathrm{T}}K_v^{-1}\dot{\widetilde{V}}\right).\end{aligned} \tag{34}$$

If the training rule (26) is changed as

$$\begin{aligned}\dot{\widehat{W}} =& -K_w\left\{\sigma\left(\hat{V}[q\dot{q}]\right) + \sigma'\tilde{V}[q\dot{q}]\right\}(\dot{q} + \alpha\tilde{q})^{\mathrm{T}} \\ \dot{\hat{V}} =& -K_v\widehat{W}\sigma' q(\dot{q} + \alpha\tilde{q})^{\mathrm{T}}\end{aligned} \tag{35}$$

Theorem 1 is also established.

One common problem of the linear PID control (18) is integral windup, where the rate of integration is larger than the actual speed of the system. The integrator's output may exceed the saturation limit of the actuator. The actuator will then operate at its limit no matter what the process outputs. This means that the system runs with an open loop instead of a constant feedback loop. The solutions of antiwindup schemes are mainly classified into two types [34]: conditional integration and back calculation. It has been shown that none of the existing methods are able to provide good performance over a wide range of processes [35]. In this paper, we use the conditional integration algorithm. The the integral term is limited to a selected value

$$u = K_p\tilde{q} + K_d\dot{\tilde{q}} + sat\left[K_i\int_0^t \tilde{q}(\tau)d\tau, \nu_{\max}\right] + \hat{f} \tag{36}$$

where $sat[x,\nu_{\max}] = \begin{cases} x & \text{if } \|x\| < \nu_{\max} \\ \nu_{\max} & \text{if } \|x\| \geq \nu_{\max}\end{cases}$. $\nu_{\max}$ is a prescribed value to the integral term when the controller saturates. This approach is also called prelodding [36]. Now, the linear PID controller becomes nonlinear PID. The semiglobal asymptotic stability has been analyzed by [6]. When $\nu_{\max}$ is the maximum torque of all joint actuators, $\nu_{\max} = k_s\max_i(|u_i^{\max}|)$, $u_i^{\max} = \max(|u_i|)$, $k_s \leq 1$. A necessary condition is

$$\nu_{\max} \geq 3\bar{G}, \quad \|G(q)\| \leq \bar{G}$$

where $G(q)$ is the gravity torque of the robot (1) and $\bar{G}$ is the upper bound of $G(q)$. k_s is a design factor in the case where not all PID terms are subjected to saturation. For the controller (36), k_s can selected as $k_s = 1/4$.

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

Following the process from (4)–(13), the neural PID with antiwindup controller (36) requires

$$\begin{aligned}\nu_{\max} &\geq 3\bar{\phi}\,\|W^*\sigma(q,\dot{q})+\phi(q,\dot{q})\| \\ &\leq \|W^*\sigma(q,\dot{q})\|+\|\phi(q,\dot{q})\| \\ &\leq \bar{\phi}\end{aligned} \tag{37}$$

where $\bar{\phi}$ is the upper bound of the neural estimator and W^*, $\sigma(q,\dot{q})$, and $\phi(q,\dot{q})$ are defined in (4).

We can see that the first additional condition for the neural PID with antiwindup is that the neural estimator must be bounded, while the linear neural PID only requires that the neural estimation error satisfies Lipschitz condition (10).

Since $u_i^{\max}$ (or $\nu_{\max}$) is a physical requirement for the actuator, it is not a design parameter. In order to satisfy the condition (37), we should force $\bar{\phi}$ as small as possible. A good structure of the neural estimator may make the term $\|W^*\sigma(q,\dot{q})\|$ smaller, such as multilayer neural network in Remark 2. There are several methods that can be used to find a good neural network, such as the genetic algorithm [37] and pruning [38]. Besides structure optimization, initial condition for the gradient training algorithm (26) also affects $\bar{\phi}$. Since the initial conditions for $\hat{W}$ and $\hat{V}$ in (35) do not affect the stability property, we design an offline method to find a better value for $\hat{W}(0)$ and $\hat{V}(0)$. If we let $\hat{W}(0)=W_0$ and $V(0)=V_0$, the algorithm (35) can make the identification error convergent, i.e., $\hat{W}(t)$ and $\hat{V}(t)$ will make the identification error smaller than that of W_0 and V_0. $\hat{W}(0)$ and $\hat{V}(0)$ are selected by the following steps.

1) Start from any initial value for $\hat{W}(0)=W_0$ and $\hat{V}(0)=V_0$.
2) Do training with (35) until T_0.
3) If $\|\tilde{q}(T_0)\|<\|\tilde{q}(0)\|$, let $\hat{W}(T_0)$ and $\hat{V}(T_0)$ as new $\hat{W}(0)$ and $\hat{V}(0)^0$, i.e., $\hat{W}(0)=\hat{W}(T_0)$ and $\hat{V}(0)=\hat{V}(T_0)$, go to 2) to repeat the training process.
4) If $\|\tilde{q}(T_0)\|\geq\|\tilde{q}(0)\|$, stop this offline identification; now, $\hat{W}(T_0)$ and $\hat{V}(T_0)$ are the final values for $\hat{W}(0)$ and $\hat{V}(0)$.

III. Neural PID Control With Unmeasurable Velocities

The neural PID control (19) uses the joint velocities $\dot{q}$. In contrast to the high precision of the position measurements by the optical encoders, the measurement of velocities by tachometers may be quite mediocre in accuracy, specifically for certain intervals of velocity. The common idea in the design of PID controllers, which requires velocity measurements, has been to propose state observers to estimate the velocity. The simplest observer may be the first-order and zero-relative position filter [2]

$$\upsilon_i(s)=\frac{b_i s}{s+a_i}q_i(s),\qquad i=1,\ldots,n \tag{38}$$

where $\upsilon_i(s)$ is an estimation of $\dot{q}_i$, a_i and b_i are the elements of diagonal matrices A and B, $A=\mathrm{diag}\{a_i\}$, $B=\mathrm{diag}\{b_i\}$, $a_i>0$, and $b_i>0$. The transfer function (38) can be realized by

$$\begin{cases}\dot{x}=-A(x+Bq)\\ \tilde{q}=x+Bq\end{cases}. \tag{39}$$

The linear PID control (19) becomes

$$\begin{aligned}u&=K_p\tilde{q}-K_d\upsilon+\xi+\widehat{W}\sigma(q)\\ \dot{\xi}&=K_i\tilde{q}\quad \xi(0)=\xi_0\\ \dot{x}&=-A(x+Bq)\\ \upsilon&=x+Bq\end{aligned} \tag{40}$$

where K_p, K_i, and K_d are positive definite diagonal matrices and a_i and b_i in (38) are positive constants.

The closed-loop system of the robot (1) is

$$\frac{d}{dt}\begin{bmatrix}\xi\\ \upsilon\\ \dot{q}\end{bmatrix}=\begin{bmatrix}K_i\tilde{q}\\ -A\upsilon+B\dot{q}\\ M^{-1}\begin{bmatrix}-C(q,\dot{q})\dot{q}-\tilde{W}\sigma(q)-\phi(q)\\ +K_p\tilde{q}-K_d\upsilon+\tilde{\xi}+\phi(q^d)\end{bmatrix}\end{bmatrix}. \tag{41}$$

The equilibrium of (41) is $[\tilde{\xi},\upsilon,\dot{q}]=[0,0,0]$.

The following theorem gives the asymptotic stability of the neural PID control with the velocity observer (38). This theorem also provides a training algorithm for neural weights and explicit selection method of PID gains.

Since the velocities are not available, the input of the neural networks becomes

$$\begin{aligned}&\hat{f}(q,\dot{q})=\widehat{W}\sigma(q,\upsilon)\\ \text{or}\quad&\hat{f}(q,\dot{q})=\widehat{W}\sigma(\hat{V}[q,\upsilon].\end{aligned} \tag{42}$$

Theorem 2: Consider robot dynamic (1) controlled by neural PID controller (40); if A and B of the velocity observer (38) satisfy

$$\begin{aligned}\frac{\lambda_M(A)}{\lambda_m^2(A)}&\leq\lambda_m(K_d)\frac{\lambda_m(B)}{\lambda_M(B)}\\ \lambda_M(B)&\leq\frac{1}{4}\lambda_m(K_d)\frac{\lambda_m(M)}{\lambda_M^2(M)}\\ \lambda_m(B-\alpha I)&\geq\frac{1}{2}\lambda_m(A)\end{aligned} \tag{43}$$

where α is positive design constant, provided that the PID control gains of (40) satisfy (44), shown at the bottom of the page,

$$\begin{aligned}\lambda_m(K_p)-\frac{1}{2}\lambda_M(K_p)\geq&\frac{1}{\alpha}\begin{bmatrix}\lambda_M(K_i)+\lambda_M(A^{-1}BK_i)\\ +\frac{1+2\alpha}{2}k_\phi+\frac{\alpha^2}{2}\lambda_M(K_d)+\frac{\alpha}{2}\lambda_M(A^{-1}K_i)\end{bmatrix}\\ \lambda_m(K_d)\geq&\frac{k_g+\frac{1}{2\alpha}\lambda_M(A^{-1}K_i)+\frac{1}{2\alpha}\lambda_M(K_p)+\kappa(M)\lambda_M(M)\lambda_M(A)}{2\lambda_m(AB^{-1}-I)-1}\\ \lambda_M(K_i)\leq&\frac{\alpha}{3}\lambda_m(K_p)\end{aligned} \tag{44}$$

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

where k_ϕ satisfies (10), $\kappa(M)$ is the condition number of M, and the weight of neural networks is tuned by

$$\dot{\hat{W}} = -K_w \sigma(q, v) \left[\alpha\tilde{q} + v + B^{-1}(\dot{v} + Av)\right]^{\mathrm{T}} \tag{45}$$

then the closed-loop system (41) is locally asymptotically stable at the equilibrium

$$x = \left[\xi - \phi(q^d), \tilde{q}, \dot{\tilde{q}}\right]^{\mathrm{T}} = 0 \tag{46}$$

in the domain of attraction

$$\|\tilde{q}\| \le \frac{\lambda_m(M)}{\alpha k_c}\left[\lambda_m(B - \alpha I) - \frac{1}{2}\lambda_m(A)\right] + \frac{1}{\alpha}\|v\|. \tag{47}$$

Proof: See the Appendix. ■

Remark 4: The conditions of (43) and (44) decide how to choose the PID gains. The first condition of (44) is

$$\begin{aligned}\lambda_m(K_p) &\ge \frac{1}{\alpha}\lambda_M(K_i) + \Omega \\ \Omega &= \frac{1}{\alpha}\begin{bmatrix}\lambda_M(A^{-1}BK_i) + \frac{1+2\alpha}{2}k_g \\ +\frac{\alpha}{2}\lambda_M(K_d) + \frac{1}{2}\lambda_M(A^{-1}K_i)\end{bmatrix} \\ &\quad + \frac{1}{2\alpha}\lambda_M(K_p)\end{aligned} \tag{48}$$

and the third condition of (44) is $\lambda_m(K_p) \ge (3/\alpha)\lambda_M(K_i)$; they are compatible. When K_i is not big, these conditions can be established. The second condition of (44) and the third condition of (43) are not directly compatible. We first let α as small as possible and K_p as big as possible. Thus, K_i cannot be big. These requirements are reasonable for our real control. If we select $B = \beta A + \alpha I$, from the third condition of (43), $\beta \ge 1/2$. The second condition of (44) requires $\lambda_m(AB^{-1} - I) > 1/2$, and there exist $1 > \beta \ge 1/2$ and a small α such that $\lambda_m[A(\beta A + \alpha I)^{-1} - I] > 1/2$. After A and B are decided, we use the second condition of (44) to select K_d.

IV. Experimental Results of the Neural PID Control for an Exoskeleton Robot

Recently, a research group in UCSC has successfully constructed a 7-DOF exoskeleton robot, see Fig. 1. In this paper, we apply our neural PID control in this exoskeleton. The computer control platform of the UCSC 7-DOF exoskeleton robot is a PC104 with an Intel Pentium4@2.4-GHz processor and 512-Mb RAM. The motors for the first four joints are mounted in the base such that large mass of the motors can be removed. Torque transmission from the motors to the joints is achieved using a cable system. The other three small motors are mounted in link five.

Fortunately, this upper limb exoskeleton is fixed on the human arm; the behavior of the exoskeleton is the same as the human arm, see Fig. 2. It is composed of a 3-DOF shoulder [Joint 1 (J1)–Joint 3 (J3)], a 1-DOF elbow [Joint 4 (J4)], and a 3-DOF wrist [Joint 5 (J5)–Joint 7 (J7)]. J1–J3 are responsible for shoulder flexion–extension, adduction, and internal–external rotation; J4 creates elbow flexion–extension; and J5–J7 are responsible for wrist flexion-extension, pronation-supination, and radial-ulnar deviation. We regard J1, J2, and J3 in Fig. 5 as three spherical joints of the human shoulder, see Fig. 2. Also, J5, J6, and J7 in Fig. 5 are considered as three spherical joints of the human wrist.

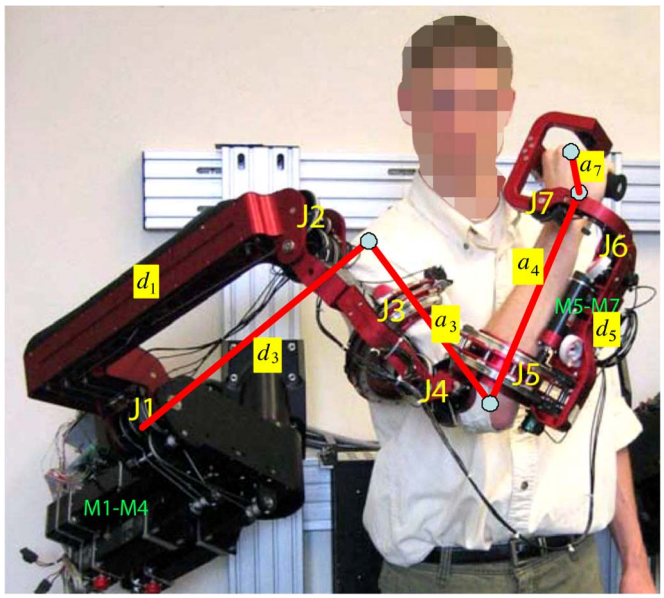


Fig. 1. UCSC 7-DOF exoskeleton robot.

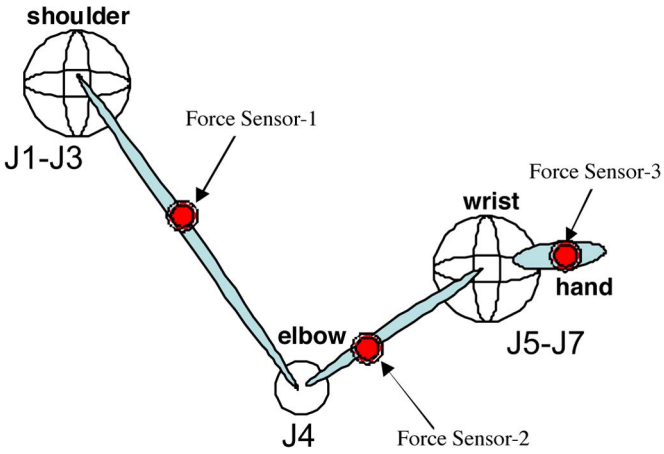


Fig. 2. Human arm versus exoskeleton.

The exoskeleton's height was adjusted for each user in a seated position, see Fig. 3. The user's left hand is an enable button which released the brakes on the device and engaged the motor. The objective of the admittance control is to move the end-effector of the exoskeleton robot from an initial position into six slotted holes, see Fig. 4.

The reference signals are generated by admittance control in task space. These references are sent to joint space. The robot in joint space can be regarded as free motion without human constraints. The whole control system is shown in Fig. 5. The objective of neural PID control is to make the transient performance faster and less overshoot, such that human feels comfortable. In Fig. 2, Force Sensor-1 and Force Sensor-2 are used to detect the human feeling, while Force Sensor-3 is used to generate the control command.

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

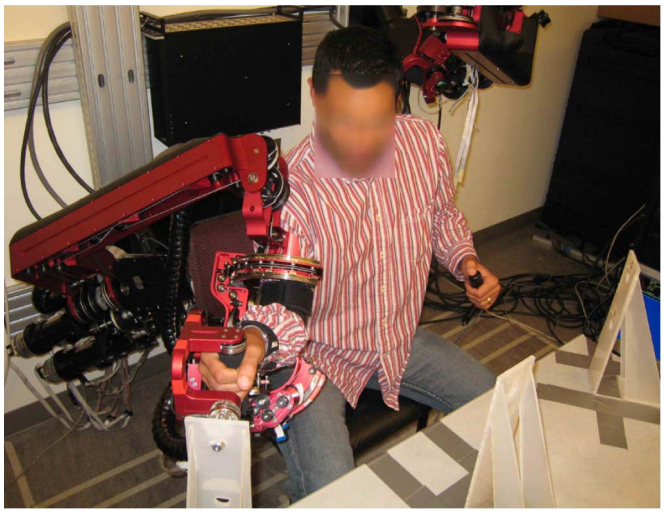


Fig. 3. Admittance control of the UCSC exoskeleton.

Fig. 4. End-effector of the exoskeleton.

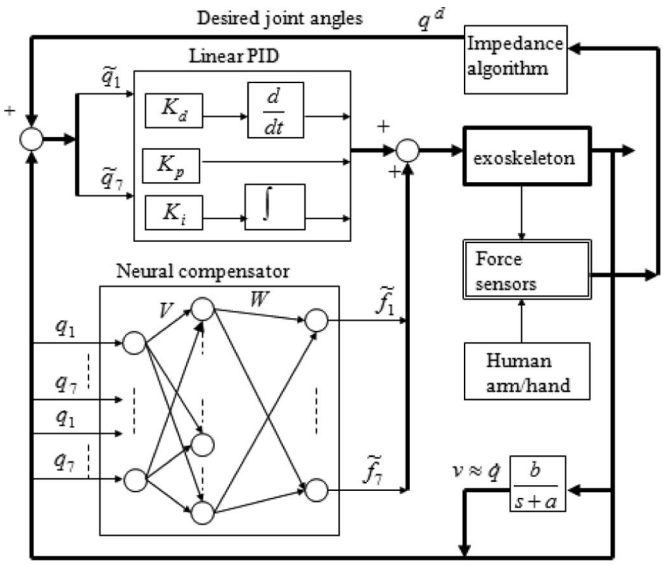


Fig. 5. Neural PID control of the exoskeleton.

TABLE I
PARAMETERS OF THE EXOSKELETON

Joint	Mass (Kg)	Center of Mass (m)	Link Length (m)	Joint Offset (m)
1	3.4	0.3	0.7	0.3
2	1.7	0.05	0.1	0
3	0.7	0.1	0.2	0.1
4	1.2	.02	0.05	0
5	1.8	.02	0.05	0.1
6	0.2	0.04	0.1	0
7	0.5	0.02	0.05	0.1

The real-time control program is operated in Windows XP with Matlab 7.1, Windows Real-Time Target, and C++. All of the controllers employed a sampling frequency of 1 kHz. The properties of the exoskeleton with respect to base frame are shown in Table I.

The two theorems in this paper give sufficient conditions for the minimal values of proportional and derivative gains and maximal values of integral gains. We use the parameters in Table I and (28) to estimate the upper and the lower bounds of the eigenvalues of the inertia matrix $M(q)$ and k_g in (10). We select $\lambda_M(M) < 3$, $\lambda_m(M) > 1$, $k_g = 10$. We choose $\alpha = 4\lambda_M(K_i)/\lambda_m(K_p)$ such that $\lambda_M(K_i) \leq (\alpha/3)\lambda_m(K_p)$ is satisfied. $\alpha = 0.08$, A is chosen as $A = \mathrm{diag}(30)$, $\beta = 7/12$, so $B = \mathrm{diag}(17.58)$. The joint velocities are estimated by the standard filters

$$\widetilde{\dot{q}}(s) = \frac{bs}{s+a}q(s) = \frac{18s}{s+30}q(s). \tag{49}$$

The PID gains are chosen as

$$\begin{aligned} K_p &= \mathrm{diag}[150, 150, 100, 150, 100, 100, 100] \\ K_i &= \mathrm{diag}[2, 1, 2, 2, 0.2, 0.1, 0.1] \\ K_d &= \mathrm{diag}[330, 330, 300, 320, 320, 300, 300] \end{aligned} \tag{50}$$

such that the conditions of Theorem 2 are satisfied. The initial elements of the weight matrix $W \in R^{7\times 7}$ are selected randomly from -1 to 1. The active function in (26) is Gaussian function

$$\sigma = \exp\left\{-(q_i - m_i)^2/100\right\}, \qquad i = 1, \ldots, 7 \tag{51}$$

where m_i is selected randomly from zero to two. The weights are updated by (45) with $K_w = 10$.

The control results of J1 with neural PID control are shown in Fig. 6, marked "Neural PID." We compare our neural PID control with the other popular robot controllers. First, we use the linear PID (15); the PID gains are the same as (50), and the control result is shown in Fig. 6, marked "Linear PID-2." Because the steady-state error is so big, the integral gains are increased as

$$K_i = \mathrm{diag}[50, 20, 30, 30, 10, 10, 10]. \tag{52}$$

The control result is shown in Fig. 6, marked "Linear PID-1," and the transient performance is poor. There still exists

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

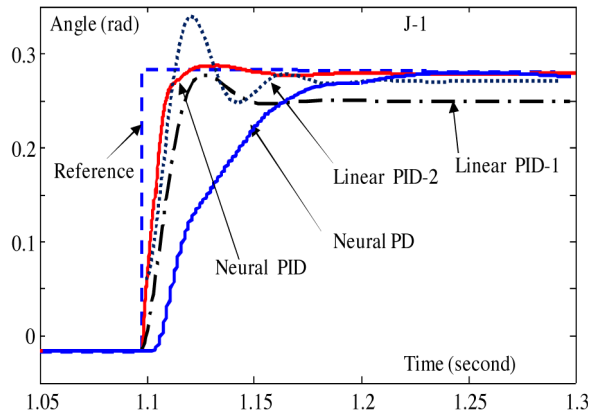


Fig. 6. Comparison of several PID controllers for J1.

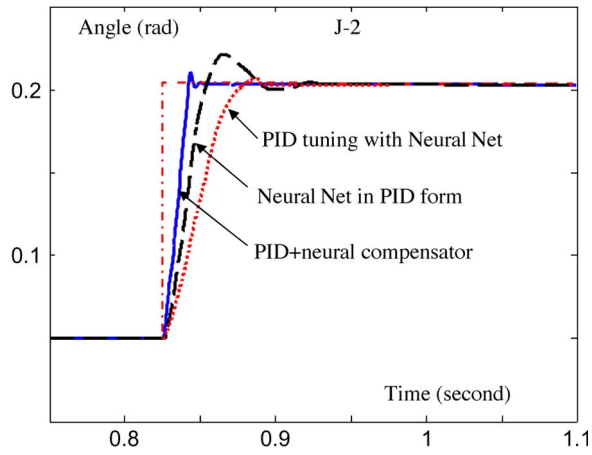


Fig. 7. Comparison of three neural PID controllers for J2.

regulation error. Further increasing K_i causes the closed-loop system unstable. Then, we use a neural compensator to replace the integrator; it is normal neural PD control (16). In order to decrease steady-state error, the derivative gains are increased as

$$K_d = \text{diag}[970, 900, 970, 970, 970, 800, 800] \qquad (53)$$

and the control result is shown in Fig. 6, marked "Neural PD." The response becomes very slow.

Now, we use J2 to compare our neural PID control with the other two types of neural PID control. The control results of these three neural PID controllers are shown in Fig. 7. Here, we use a three-layer neural network with three nodes which have integral, proportional, and derivative properties. A backpropagation-like training algorithm is used to ensure closed-loop stability [23]; it is marked "Neural Net in PID form." Then, we use a one-hidden-layer neural network to tune the linear PID gains as in [27]; it is marked "PID tuning via neural net." It can be found that the "Neural net in PID form" can assure stability, but the transient performance is not good. The "PID tuning via neural net" is acceptable except for its slow response.

Finally, we use the other joints, J3–J7, to compare our neural PID with the other popular robot controllers. The results are shown in Figs. 8–12.

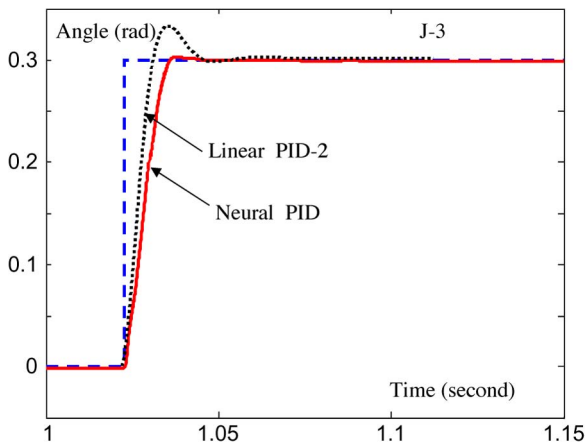


Fig. 8. Linear PID via neural PID for J3.

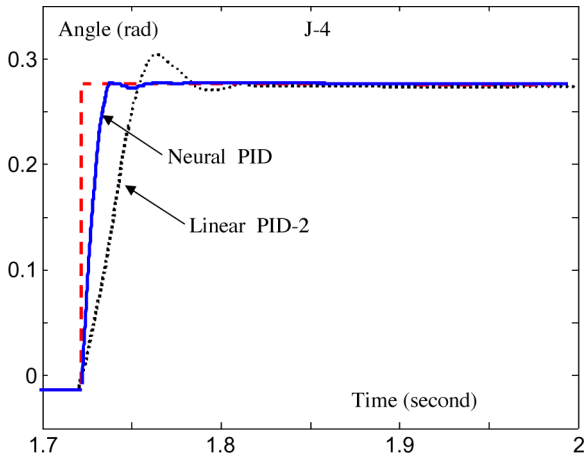


Fig. 9. Linear PID-2 via neural PID for J4.

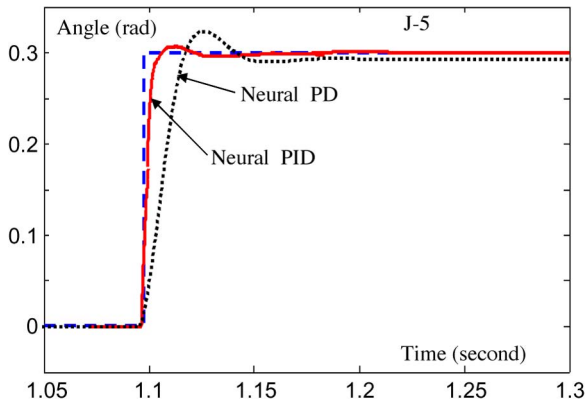


Fig. 10. Neural PD via neural PID for J5.

Clearly, neural PID control can successfully compensate the uncertainties such as friction, gravity, and the other uncertainties of the robot. Because the linear PID controller has no compensator, it has to increase its integral gain to cancel the uncertainties. The neural PD control does not apply an integrator; its derivative gain is big.

The structure of neural compensator is very important. The number of hidden nodes m in (5) constitutes a structural

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

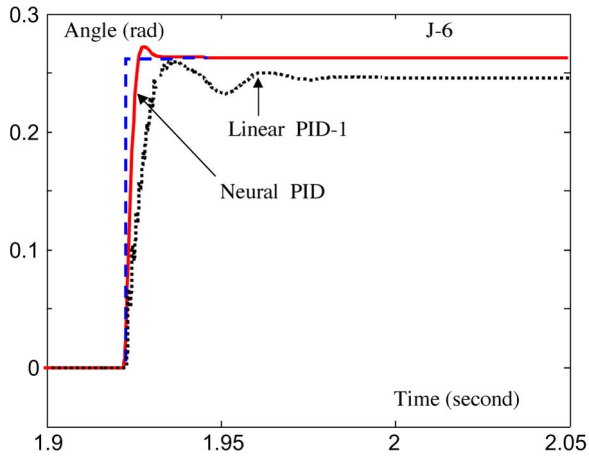


Fig. 11. Linear PID via neural PID for J6.

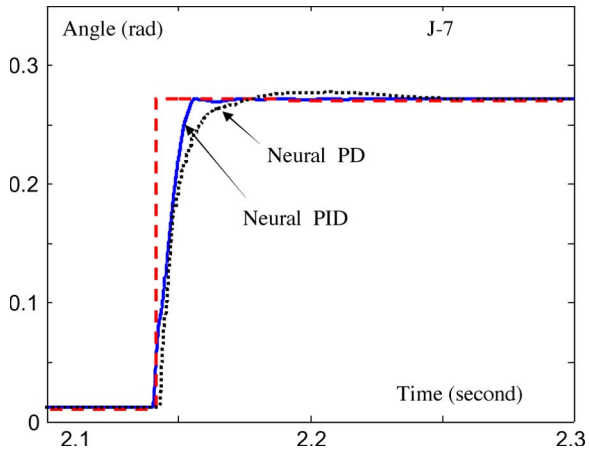


Fig. 12. Neural PD via neural PID for J7.

problem for neural systems. It is well known that increasing the dimension of the hidden layer can cause the "overlap" problem and add to the computational burden. The best dimension to use is still an open problem for the neural control research community. In this application, we did not use hidden layer, and the control results are satisfied. The learning gain K_w in (45) will influence the learning speed, so a very large gain can cause unstable learning, while a very small gain produces slow learning process.

V. Conclusion

The neural PID proposed in this paper solves the problems of large integral and derivative gains in the linear PID control and the neural PD control. It keeps good properties of the industrial PID control and neural compensator. Semiglobal asymptotic stability of this neural PID control is proved. When the joint velocities of robot manipulators are not available, local asymptotic stability is assured with filtered positions. The stability conditions give explicit methods to select PID gains. We also apply our neural PID to the UCSC 7-DOF exoskeleton robot. Theory analysis and experimental study show the validity of the neural PID control.

Appendix

Proof: [Proof of Theorem 1] We construct a Lyapunov function as

$$V = \frac{1}{2}\dot{q}^{\mathrm{T}} M \dot{q} + \frac{1}{2}\tilde{q}^{\mathrm{T}} K_p \tilde{q} + \int_0^t \phi(q,\dot{q}) dq - k_\phi + \tilde{q}^{\mathrm{T}} \phi(q^d) + \frac{3}{2}\phi(q^d)^{\mathrm{T}} K_p^{-1} \phi(q^d) + \frac{\alpha}{2}\tilde{\xi}^{\mathrm{T}} K_i^{-1} \tilde{\xi} + \tilde{q}^{\mathrm{T}} \tilde{\xi} - \alpha \tilde{q}^{\mathrm{T}} M \dot{q} + \frac{\alpha}{2}\tilde{q}^{\mathrm{T}} K_d \tilde{q} + \frac{1}{2}\mathrm{tr}\left(\widetilde{W}^{\mathrm{T}} K_w^{-1} \widetilde{W}\right) \quad (54)$$

where k_ϕ is defined in (13) such that $V(0) = 0$. α is a design positive constant. We first prove that$V$ is a Lyapunov function, $V \geq 0$. The term $(1/2)\tilde{q}^{\mathrm{T}} K_p \tilde{q}$ is separated into three parts, and $V = \sum_{i=1}^{4} V_i$

$$V_1 = \frac{1}{6}\tilde{q}^{\mathrm{T}} K_p \tilde{q} + \tilde{q}^{\mathrm{T}} \phi(q^d) + \frac{3}{2}\phi(q^d)^{\mathrm{T}} K_p^{-1} \phi(q^d)$$

$$V_2 = \frac{1}{6}\tilde{q}^{\mathrm{T}} K_p \tilde{q} + \tilde{q}^{\mathrm{T}} \tilde{\xi} + \frac{\alpha}{2}\tilde{\xi}^{\mathrm{T}} K_i^{-1} \tilde{\xi}$$

$$V_3 = \frac{1}{6}\tilde{q}^{\mathrm{T}} K_p \tilde{q} - \alpha \tilde{q}^{\mathrm{T}} M \dot{q} + \frac{1}{2}\dot{q}^{\mathrm{T}} M \dot{q}$$

$$V_4 = \int_0^t \phi(q) d\tau - k_\phi + \frac{\alpha}{2}\tilde{q}^{\mathrm{T}} K_d \tilde{q} + \frac{1}{2}\mathrm{tr}\left(\widetilde{W}^{\mathrm{T}} K_w^{-1} \widetilde{W}\right) \geq 0. \quad (55)$$

It is easy to find

$$V_1 = \frac{1}{2}\begin{bmatrix} \tilde{q} \\ \phi(q^d) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{3}K_p & I \\ I & 3K_p^{-1} \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \phi(q^d) \end{bmatrix}. \quad (56)$$

Since $K_p \geq 0$, V_1 is a semipositive definite matrix, $V_1 \geq 0$. When $\alpha \geq 3/\lambda_m(K_i^{-1})\lambda_m(K_p)$

$$V_2 \geq \frac{1}{2}\left(\sqrt{\frac{1}{3}\lambda_m(K_p)}\|\tilde{q}\| - \sqrt{\frac{3}{\lambda_m(K_p)}}\|\tilde{\xi}\|\right)^2 \geq 0. \quad (57)$$

Because

$$y^{\mathrm{T}} A x \leq \|y\|\|Ax\| \leq \|y\|\|A\|\|x\| \leq |\lambda_M(A)|\|y\|\|x\| \quad (58)$$

when $\alpha \leq \sqrt{(1/3)\lambda_m(M)\lambda_m(K_p)}/\lambda_M(M)$

$$V_3 \geq \frac{1}{2}\left(\sqrt{\lambda_m(M)}\|\dot{q}\| - \sqrt{\frac{1}{3}\lambda_m(K_p)}\|\tilde{q}\|\right)^2 \geq 0. \quad (59)$$

Obviously, if

$$\sqrt{\frac{1}{3}}\lambda_m\left(K_i^{-1}\right)\lambda_m^{\frac{3}{2}}(K_p)\lambda_m^{\frac{1}{2}}(M) \geq \lambda_M(M) \quad (60)$$

there exists

$$\frac{\sqrt{\frac{1}{3}\lambda_m(M)\lambda_m(K_p)}}{\lambda_M(M)} \geq \alpha \geq \frac{3}{\lambda_m\left(K_i^{-1}\right)\lambda_m\left(K_p\right)}. \quad (61)$$

This means that if K_p is sufficiently large or K_i is sufficiently small, (60) is established, and $V(\dot{q}, \tilde{q}, \tilde{\xi})$ is

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

globally positive definite. Using $(d/dt)\int_0^t \phi(q,\dot{q})dq = (\partial \int_0^t \phi(q,\dot{q})dq/\partial q)(\partial q/\partial t) = \dot{q}^{\mathrm{T}}\phi(q,\dot{q})$, $(d/dt)\phi(q^d) = 0$ and $(d/dt)[\tilde{q}^{\mathrm{T}}\phi(q^d)] = \dot{\tilde{q}}^{\mathrm{T}}\phi(q^d)$, the derivative of V is

$$\begin{aligned}\dot{V} =& \dot{q}^{\mathrm{T}}M\ddot{q} + \frac{1}{2}\dot{q}^{\mathrm{T}}\dot{M}\dot{q} + \dot{\tilde{q}}^{\mathrm{T}}K_p\tilde{q} + \phi(q,\dot{q})^{\mathrm{T}}\dot{q} \\ &+ \dot{\tilde{q}}^{\mathrm{T}}\phi(q^d) + \mathrm{tr}\left(\widetilde{W}^{\mathrm{T}}K_w^{-1}\dot{\widetilde{W}}\right) \\ &+ \alpha\dot{\tilde{\xi}}^{\mathrm{T}}K_i^{-1}\tilde{\xi} + \dot{\tilde{q}}^{\mathrm{T}}\tilde{\xi} + \tilde{q}^{\mathrm{T}}\dot{\tilde{\xi}} \\ &- \alpha(\dot{\tilde{q}}^{\mathrm{T}}M\dot{q} + \tilde{q}^{\mathrm{T}}\dot{M}\dot{q} + \tilde{q}^{\mathrm{T}}M\ddot{q}) \\ &+ \alpha\tilde{q}^{\mathrm{T}}K_d\dot{\tilde{q}}. \end{aligned} \tag{62}$$

Using (8), the first three terms of (62) become

$$-\dot{q}^{\mathrm{T}}\phi(q) - \dot{q}^{\mathrm{T}}K_d\dot{q} + \dot{q}^{\mathrm{T}}\tilde{\xi} + \dot{q}^{\mathrm{T}}\phi(q^d) + \dot{q}^{\mathrm{T}}\tilde{W}\sigma(q,\dot{q}) \tag{63}$$

$$\begin{aligned}\dot{V} \le& -[\lambda_m(K_d) - \alpha\lambda_M(M) - \alpha k_c\|\tilde{q}\|]\,\|\dot{q}\|^2 \\ &- [\alpha\lambda_m(K_p) - \lambda_M(K_i) - \alpha k_g]\,\|\tilde{q}\|^2. \end{aligned} \tag{64}$$

If

$$\|\tilde{q}\| \le \frac{\lambda_M(M)}{\alpha k_c} \tag{65}$$

$$\begin{aligned}\lambda_m(K_d) \ge& (1+\alpha)\lambda_M(M) \\ \lambda_m(K_p) \ge& \frac{1}{\alpha}\lambda_M(K_i) + k_g \end{aligned} \tag{66}$$

then $\dot{V} \le 0$ and $\|\tilde{q}\|$ decreases. Then, (66) is established. Using (60) and $\lambda_m(K_i^{-1}) = 1/\lambda_M(K_i)$, (66) is (25).

$\dot{V}$ is negative semidefinite. Define a ball Σ of radius $\sigma > 0$ centered at the origin of the state space, which satisfies this condition

$$\Sigma = \left\{\tilde{q} : \|\tilde{q}\| \le \frac{\lambda_M(M)}{\alpha k_c} = \sigma\right\} \tag{67}$$

$\dot{V}$ is negative semidefinite on the ball Σ. There exists a ball Σ of radius $\sigma > 0$ centered at the origin of the state space on which $\dot{V} \le 0$. The origin of the closed-loop equation (24) is a stable equilibrium. Since the closed-loop equation is autonomous, we use La Salle's theorem. Define Ω as

$$\begin{aligned}\Omega =& \left\{x(t) = [\tilde{q},\dot{q},\tilde{\xi}] \in R^{3n} : \dot{V} = 0\right\} \\ =& \left\{\tilde{\xi} \in R^n : \tilde{q} = 0 \in R^n, \dot{q} = 0 \in R^n\right\}. \end{aligned} \tag{68}$$

From (62), $\dot{V} = 0$ if and only if $\tilde{q} = \dot{q} = 0$. For a solution $x(t)$ to belong to Ω for all $t \ge 0$, it is necessary and sufficient that $\tilde{q} = \dot{q} = 0$ for all $t \ge 0$. Therefore, it must also hold that $\ddot{q} = 0$ for all $t \ge 0$. We conclude that, from the closed-loop system (24), if $x(t) \in \Omega$ for all $t \ge 0$, then

$$\begin{aligned}\phi(q,\dot{q}) =& \phi(q^d,0) = \tilde{\xi} + \phi(q^d,0) \\ \dot{\tilde{\xi}} =& 0 \end{aligned} \tag{69}$$

implies that $\tilde{\xi} = 0$ for all $t \ge 0$. Thus, $x(t) = [\tilde{q},\dot{q},\tilde{\xi}] = 0 \in R^{3n}$ is the only initial condition in Ω for which $x(t) \in \Omega$ for all $t \ge 0$.

Finally, we conclude from all this that the origin of the closed-loop system (24) is locally asymptotically stable. Because $1/\alpha \le \lambda_m(K_i^{-1})\lambda_m(K_p)$, the upper bound for $\|\tilde{q}\|$ can be

$$\|\tilde{q}\| \le \frac{\lambda_M(M)}{k_c}\lambda_M(K_i)\lambda_m(K_p). \tag{70}$$

It establishes the semiglobal stability of our controller, in the sense that the domain of attraction can be arbitrarily enlarged with a suitable choice of the gains. Namely, increasing K_p, the basin of attraction will grow. ■

Proof: [Proof of Theorem 2] We construct a Lyapunov function as

$$\begin{aligned}V_c =& \frac{1}{2}\dot{q}^{\mathrm{T}}M\dot{q} + \frac{1}{2}\tilde{q}^{\mathrm{T}}K_p\tilde{q} + \int_0^t \phi(q)d\tau - k_\phi + \tilde{q}^{\mathrm{T}}\phi(q^d) \\ &+ \frac{3}{2}\phi(q^d)^{\mathrm{T}}K_p^{-1}\phi(q^d) + \frac{\alpha}{2}\tilde{\xi}^{\mathrm{T}}K_i^{-1}\tilde{\xi} \\ &- \alpha\tilde{q}^{\mathrm{T}}M\dot{q} + \tilde{q}^{\mathrm{T}}(I + A^{-1}B)\tilde{\xi} + \frac{1}{2}\upsilon^{\mathrm{T}}B^{-1}K_d\upsilon - \upsilon^{\mathrm{T}}M\dot{q} \\ &+ \upsilon^{\mathrm{T}}A^{-1}\tilde{\xi} + \frac{1}{2}\mathrm{tr}\left(\widetilde{W}^{\mathrm{T}}K_w^{-1}\widetilde{W}\right) \end{aligned} \tag{71}$$

where the definition of k_ϕ is the same as that in Theorem 1. α is a design positive constant. We first prove that V is a Lyapunov function, $V \ge 0$. The term $(1/2)\tilde{q}^{\mathrm{T}}K_p\tilde{q}$ is separated into three parts, and $V = \sum_{i=1}^6 V_i$

$$\begin{aligned}V_1 =& \frac{1}{6}\tilde{q}^{\mathrm{T}}K_p\tilde{q} + \tilde{q}^{\mathrm{T}}\phi(q^d) + \frac{3}{2}\phi(q^d)^{\mathrm{T}}K_p^{-1}\phi(q^d) \\ V_2 =& \frac{1}{6}\tilde{q}^{\mathrm{T}}K_p\tilde{q} + \tilde{q}^{\mathrm{T}}\tilde{\xi} + \frac{\alpha}{2}\tilde{\xi}^{\mathrm{T}}K_i^{-1}\tilde{\xi} \\ V_3 =& \frac{1}{6}\tilde{q}^{\mathrm{T}}K_p\tilde{q} - \alpha\tilde{q}^{\mathrm{T}}M\dot{q} + \frac{1}{4}\dot{q}^{\mathrm{T}}M\dot{q} \\ V_4 =& \frac{1}{4}\upsilon^{\mathrm{T}}(B^{-1}K_d)\upsilon + \upsilon^{\mathrm{T}}A^{-1}\tilde{\xi} + \tilde{\xi}^{\mathrm{T}}(A^{-1}B)\tilde{\xi} \\ V_5 =& \frac{1}{4}\upsilon^{\mathrm{T}}(B^{-1}K_d)\upsilon - \upsilon^{\mathrm{T}}M\dot{q} + \frac{1}{4}\dot{q}^{\mathrm{T}}M\dot{q} \\ V_6 =& \int_0^t \phi(q)d\tau - k_\phi + \frac{1}{2}\mathrm{tr}\left(\widetilde{W}^{\mathrm{T}}K_w^{-1}\widetilde{W}\right) \ge 0. \end{aligned} \tag{72}$$

Here, V_1 and V_2 are the same as (5), i.e.,

$$\lambda_M(K_i) \le \frac{\alpha}{3}\lambda_m(K_p). \tag{73}$$

For V_3, if $\alpha \le \sqrt{(1/6)\lambda_m(K_p)\lambda_m(M)}/\lambda_M(M)$

$$V_3 \ge \frac{1}{2}\left(\sqrt{\frac{1}{2}\lambda_m(M)}\|\dot{q}\| - \sqrt{\frac{1}{3}\lambda_m(K_p)}\|\tilde{q}\|\right)^2 \ge 0. \tag{74}$$

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

Because $\lambda_m(AB) \le \lambda_m(B^{-1})\lambda_M(A)$ and $\lambda_m(B^{-1}) = 1/\lambda_M(B)$, it is easy to find that if $\lambda_M(A^{-1}) \le \sqrt{\lambda_m(B^{-1}K_d)\lambda_m((A^{-1}B))}$ or $\lambda_M(A)/\lambda_m^2(A) \le \lambda_m(K_d)(\lambda_m(B)/\lambda_M(B))$

$$V_4 \ge \frac{1}{2}\begin{pmatrix} \frac{1}{2}\lambda_m(B^{-1}K_d)\|v\|^2 - 2\lambda_M(A^{-1})\|v\|\|\tilde{\xi}\| \\ +2\lambda_m\left((A^{-1}B)\right)\|\tilde{\xi}\|^2 \end{pmatrix} \ge 0. \tag{75}$$

If $\lambda_M(M) \le (1/2)\sqrt{\lambda_m((B^{-1}K_d))\lambda_m(M)}$ or $\lambda_M(B) \le (1/4)\lambda_m(K_d)(\lambda_m(M)/\lambda_M^2(M))$

$$V_5 = \frac{1}{2}\left[\frac{1}{2}v^{\mathrm{T}}K_dB^{-1}v + 2v^{\mathrm{T}}M\dot{q} + \frac{1}{2}\dot{q}^{\mathrm{T}}M\dot{q}\right] \ge 0. \tag{76}$$

Because $V_6 \ge 0$, obviously, there exist α, A, and B such that

$$\begin{aligned} \alpha^2 &\le \frac{1}{6}\frac{\lambda_m(K_p)\lambda_m(M)}{\lambda_M^2(M)} \\ \frac{\lambda_M(A)}{\lambda_m^2(A)} &\le \lambda_m(K_d)\frac{\lambda_m(B)}{\lambda_M(B)} \\ \lambda_M(B) &\le \frac{1}{4}\lambda_m(K_d)\frac{\lambda_m(M)}{\lambda_M^2(M)}. \end{aligned} \tag{77}$$

This means that if K_p is sufficiently large or K_i is sufficiently small, (60) is established, and V_c is globally positive definite. Now, we compute its derivative. The derivative of V_c is

$$\begin{aligned} \dot{V} \le & -\dot{q}^{\mathrm{T}}BM\dot{q} - v^{\mathrm{T}}AB^{-1}K_dv - \alpha\tilde{q}^{\mathrm{T}}K_p\tilde{q} + \alpha k_g\|\tilde{q}\|^2 \\ & + \alpha\dot{q}^{\mathrm{T}}M\dot{q} + k_c\|\alpha\tilde{q} - v\|\|\dot{q}\|^2 \\ & + \tilde{q}^{\mathrm{T}}K_i\tilde{q} + \tilde{q}^{\mathrm{T}}A^{-1}BK_i\tilde{q} + v^{\mathrm{T}}K_dv + \frac{1}{2}k_g\|v\|^2 + \frac{1}{2}k_g\|\tilde{q}\|^2 \\ & + \tilde{q}^{\mathrm{T}}(\alpha K_d - K_p - A^{-1}K_i)v + \dot{q}^{\mathrm{T}}AMv \\ & + \mathrm{tr}\Big[\widetilde{W}^{\mathrm{T}}\Big(K_w^{-1}\dot{\widetilde{W}} + \sigma(q,v)\dot{q}^{\mathrm{T}} + \alpha\sigma(q,v)\tilde{q}^{\mathrm{T}} \\ & \quad + \sigma(q,v)v^{\mathrm{T}}\Big)\Big]. \end{aligned} \tag{78}$$

Because $\dot{v} = -Av + B\dot{q}$ and $B = \mathrm{diag}\{b_i\}$, the last term is zero if we apply the updating law (45). Using (58), (78) is

$$\begin{aligned} \dot{V} \le & -\dot{q}^{\mathrm{T}}\begin{Bmatrix} \lambda_m(BM - \alpha M) - k_c\|\alpha\tilde{q} - v\| \\ -\frac{1}{2\kappa(M)}\lambda_M(AM) \end{Bmatrix}\dot{q} \\ & - v^{\mathrm{T}}\begin{pmatrix} \lambda_m(AB^{-1}K_d - K_d) - \frac{1}{2}k_g - \frac{1}{2\alpha}\lambda_M(K_e) \\ -\frac{\kappa(M)}{2}\lambda_M(AM) \end{pmatrix}v \\ & - \tilde{q}^{\mathrm{T}}\begin{pmatrix} \lambda_m(\alpha K_p - K_i - A^{-1}BK_i) - \alpha k_g \\ -\frac{1}{2}k_g - \frac{\alpha}{2}\lambda_M(K_e) \end{pmatrix}\tilde{q}. \end{aligned} \tag{79}$$

Using $\lambda_i(A)\lambda_M(B) \ge \lambda_i(AB) \ge \lambda_i(A)\lambda_m(B)$, i can be "m" or "M," and the last condition of (77) can be replaced by

$$\begin{aligned} & \left\|\tilde{q} - \frac{1}{\alpha}v\right\| \\ & \le \frac{1}{\alpha k_c}\left[\lambda_m(B - \alpha I)\lambda_m(M) - \frac{1}{2\kappa(M)}\lambda_M(M)\lambda_m(A)\right]. \end{aligned}$$

It is the attraction area (47).

Using $\lambda_i(A) + \lambda_M(B) \ge \lambda_i(A + B) \ge \lambda_i(A) + \lambda_m(B)$, the second condition of (77) is

$$\begin{aligned} \lambda_m\left[(AB^{-1} - I)K_d\right] &\ge \lambda_m(AB^{-1} - I)\lambda_m(K_d) \\ &\ge \frac{1}{2}k_g + \frac{1}{2\alpha}\lambda_M(K_e) \\ &\quad + \frac{\kappa(M)}{2}\lambda_M(AM). \end{aligned} \tag{80}$$

It is the condition for K_d in (44). Also

$$\begin{aligned} \lambda_m(\alpha K_p) \ge & \lambda_M(K_i) + \lambda_M(A^{-1}BK_i) \\ & + \frac{1 + 2\alpha}{2}k_g + \frac{\alpha}{2}\lambda_M(K_e). \end{aligned}$$

It is the condition for K_p in (44). The condition for K_i in (44) is obtained from (73). The remaining part of the proof is the same as Theorem 1. ■

ACKNOWLEDGMENT

The authors would like to thank L. M. Miller for his help to complete the experiments.

REFERENCES

[1] Y. Jin, "Decentralized adaptive fuzzy control of robot manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 1, pp. 47–57, Feb. 1998.

[2] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. Hoboken, NJ: Wiley, 1989.

[3] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*, 2nd ed. New York: Marcel Dekker, 2004, p. 10016.

[4] S. Arimoto, "Fundamental problems of robot control: Part I. Innovations in the realm of robot servo-loops," *Robotica*, vol. 13, no. 1, pp. 19–27, Jan. 1995.

[5] D. Sun, S. Hu, X. Shao, and C. Liu, "Global stability of a saturated nonlinear PID controller for robot manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 4, pp. 892–899, Jul. 2009.

[6] J. Alvarez-Ramirez, R. Kelly, and I. Cervantes, "Semiglobal stability of saturated linear PID control for robot manipulators," *Automatica*, vol. 39, no. 6, pp. 989–995, Jun. 2003.

[7] R. Ortega, A. Loria, and R. Kelly, "A semiglobally stable output feedback PI^{2}D regulator for robot manipulators," *IEEE Trans. Autom. Control*, vol. 40, no. 8, pp. 1432–1436, Aug. 1985.

[8] V. Parra-Vega, S. Arimoto, Y.-H. Liu, G. Hirzinger, and P. Akella, "Dynamic sliding PID control for tracking of robot manipulators: Theory and experiments," *IEEE Trans. Robot. Autom.*, vol. 19, no. 6, pp. 967–976, Dec. 2003.

[9] T. Dierks and S. Jagannathan, "Neural network output feedback control of robot formations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 2, pp. 383–399, Apr. 2010.

[10] P. Rocco, "Stability of PID control for industrial robot arms," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 606–614, Aug. 1996.

[11] R. Kelly, V. Santibáñez, and L. Perez, *Control of Robot Manipulators in Joint Space*. Berlin, Germany: Springer-Verlag, 2005.

[12] P. Tomei, "Adaptive PD controller for robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 565–570, Aug. 1991.

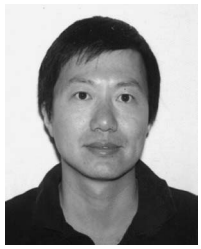
[13] B. Paden and R. Panja, "Globally asymptotically stable PD + controller for robot manipulators," *Int. J. Control*, vol. 47, no. 6, pp. 1697–1712, 1988.

[14] K. Dupree, C.-H. Liang, G. Hu, and W. E. Dixon, "Adaptive Lyapunov-based control of a robot and mass–spring system undergoing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 1050–1061, Aug. 2008.

[15] Z. Qu, D. M. Dawson, S. Y. Lim, and J. F. Dorsey, "A new class of robust control laws for tracking of robots," *Int. J. Robot. Res.*, vol. 13, no. 4, pp. 355–363, Aug. 1994.

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.

[16] E. V. L. Nunes, L. Hsu, and F. Lizarralde, "Arbitrarily small damping allows global output feedback tracking of a class of Euler–Lagrange systems," in *Proc. Amer. Control Conf.*, Seattle, WA, 2008, pp. 377–382.
[17] H.-X. Li, L. Zhang, K.-Y. Cai, and G. Chen, "An improved robust fuzzy-PID controller with optimal fuzzy reasoning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1283–1294, Dec. 2005.
[18] E. Harinath and G. Mann, "Design and tuning of standard additive model based fuzzy PID controllers for multivariable process systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 3, pp. 667–674, Jun. 2008.
[19] F. L. Lewis, "Nonlinear network structures for feedback control," *Asian J. Control*, vol. 1, no. 4, pp. 205–228, Dec. 1999.
[20] C.-S. Chen, "Dynamic structure neural–fuzzy networks for robust adaptive control of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3402–3414, Sep. 2008.
[21] M. J. Er and Y. Gao, "Robust adaptive control of robot manipulators using generalized fuzzy neural networks," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 620–628, Jun. 2003.
[22] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.
[23] S. Cong and Y. Liang, "PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3872–3879, Oct. 2009.
[24] G. M. Scott, J. W. Shavlik, and W. H. Ray, "Refining PID controllers using neural networks," *Neural Comput.*, vol. 4, no. 5, pp. 746–757, Sep. 1992.
[25] H. J. Uang and C. C. Lien, "Mixed $H2/H\infty$ PID tracking control design for uncertain spacecraft systems using a cerebellar model articulation controller," *Proc. Inst. Elect. Eng.—Control Theory Appl.*, vol. 153, no. 1, pp. 1–13, Jan. 2006.
[26] G. K. I. Mann, B.-G. Hu, and R. G. Gosine, "Two-level tuning of fuzzy PID controllers," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 2, pp. 263–269, Apr. 2001.
[27] S.-J. Ho, L.-S. Shu, and S.-Y. Ho, "Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 3, pp. 421–434, Jun. 2006.
[28] D.-L. Yu, T. K. Chang, and D.-W. Yu, "Fault tolerant control of multivariable processes using auto-tuning PID controller," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 1, pp. 32–43, Feb. 2005.
[29] F. Karray, W. Gueaieb, and S. Al-Sharhan, "The hierarchical expert tuning of PID controllers using tools of soft computing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1283–1294, Feb. 2005.
[30] G. D'Emilia, A. Marra, and E. Natale, "Use of neural networks for quick and accurate auto-tuning of PID controller," *J. Robot. Comput.-Integr. Manuf.*, vol. 23, no. 2, pp. 170–179, Apr. 2007.
[31] L. B. Gutierrez and F. L. Lewis, "Implementation of a neural net tracking controller for a single flexible link: Comparison with PD and PID controllers," *IEEE Trans. Ind. Electron.*, vol. 45, no. 2, pp. 307–318, Apr. 1998.
[32] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 703–715, May 1995.
[33] J. J. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Autom. Control*, vol. 33, no. 11, pp. 995–1003, Nov. 1988.
[34] A. Visioli, "Modified anti-windup scheme for PID controllers," *Proc. Inst. Elect. Eng.—Control Theory Appl.*, vol. 161, no. 1, pp. 49–54, Jan. 2003.
[35] K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design and Tuning*. Research Triangle Park, NC: ISA Press, 1995.
[36] F. G. Shinskey, *Process-Control Systems: Application, Design, Adjustment*. New York: McGraw-Hill, 1996.
[37] J. Arifovica and R. Gencay, "Using genetic algorithms to select architecture of a feed forward artificial neural network," *Phys. A*, vol. 289, pp. 574–594, 2001.
[38] K. Suzuki, I. Horina, and N. Sugie, "A simple neural network pruning algorithm with application to filter synthesis," *Neural Process. Lett.*, vol. 13, no. 1, pp. 43–53, Feb. 2001.



Wen Yu (M'97–SM'04) received the B.S. degree from Tsinghua University, Beijing, China, in 1990 and the M.S. and Ph.D. degrees in electrical engineering from Northeastern University, Shenyang, China, in 1992 and 1995, respectively.

From 1995 to 1996, he was a Lecturer with the Department of Automatic Control, Northeastern University, where he has been a Visiting Professor since 2006. Since 1996, he has been with the Centro de Investigación y de Estudios Avanzados, Instituto Politécnico Nacional (CINVESTAV-IPN), Mexico City, Mexico, where he is currently a Professor with the Departamento de Control Automatico. From 2002 to 2003, he held research positions with the Instituto Mexicano del Petroleo. He was a Senior Visiting Research Fellow with Queen's University Belfast, Belfast, U.K., from 2006 to 2007, and a Visiting Associate Professor with the University of California, Santa Cruz, from 2009 to 2010.

Dr. Yu serves as an Associate Editor of *Neurocomputing* and the *Journal of Intelligent and Fuzzy Systems*. He is a member of the Mexican Academy of Sciences.

Jacob Rosen received the B.Sc. degree in mechanical engineering and the M.Sc. and Ph.D. degrees in biomedical engineering from Tel-Aviv University, Tel Aviv, Israel, in 1987, 1993, and 1997, respectively.

From 1993 to 1997, he was a Research Associate developing and studying the electromyography-based powered exoskeleton with the Biomechanics Laboratory, Department of Biomedical Engineering, Tel-Aviv University. During the same period of time, he held a position in a start-up company developing innovative orthopedic spine/pelvis implants. From 1997 to 2000, he was a Postdoctoral Reseacher with the Department of Electrical Engineering and the Department of Surgery, University of Washington, Seattle, while developing surgical robotic and medical simulation systems. From 2001 to 2008, he was a Faculty Member with the Department of Electrical Engineering, University of Washington, with adjunct appointments with the Department of Surgery and the Department of Mechanical Engineering. Since 2008, he has been directing the Bionics Laboratory, University of California, Santa Cruz, where he is a Professor with the Department of Computer Engineering. He developed several key systems in the field of medical robotics such as the Blue and the Red Dragon for minimally invasive surgical skill evaluation that is commercialized by Simulab as the "Edge," Raven—a surgical robotic system for telesurgery, several generations of upper and lower limb exoskeletons, and, most recently, the Exo-UL7—a wearable robotic system. He is a coauthor of 70 manuscripts in the field of medical robotics and a coauthor and coeditor of a book entitled "Surgical Robotics—Systems, Applications, and Visions" (Springer, 2010). His research interests focus on medical robotics, biorobotics, human-centered robotics, surgical robotics, wearable robotics, rehabilitation robotics, neural control, and human–machine interface.

Authorized licensed use limited to: University of Sydney. Downloaded on January 08,2021 at 04:51:54 UTC from IEEE Xplore. Restrictions apply.