

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

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

Abstract— In order to minimize steady-state error with respect to uncertainties in robot control, PID control needs a big integral gain, or a neural compensator is added to the classical 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 are proven with standard weights training algorithms. These conditions give explicit selection methods for the gains of the linear PID control. A experimental study on an upper limb exoskeleton with this neural PID control is addressed.

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, 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 torques 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 proven in [10], where the robot dynamic was re-written in a decoupled linear system and a bounded nonlinear system. In [11], asymptotic stability of linear PID was proven, 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]. But these controllers are complex, many good properties of the linear PID control do not exist.

Intelligent compensation for PD control does not need mathematical model, 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 $\frac{B}{K_v}$, here B is the upper bound of all unknown uncertainties, K_v is the derivative gain in PD control. These neural PD controllers are uniformly ultimate boundedness (UUB), and tracking errors go to smaller with increasing the gain K_v . The cost of large K_v is the transient performance becomes slow. Only when $K_v \rightarrow \infty$, the tracking error converges to zero [19].

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

There are two different approaches to combine PID control with the intelligent control, such as neural control. The first one is neural networks are formed into PID structure [23][24][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 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 above disadvantages. It is an industrial

Wen Yu is with the Departamento de Control Automatico, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, 07360, Mexico. Jacob Rosen is with Department of Computer Engineering, University of California - Santa Cruz, Santa Cruz, California, USA

linear PID controller adding a neural compensator. The main obstacle of this neural PID is theoretical difficult 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 proven. Explicit conditions for choosing PID gains are given. When the measurement of velocities it is not available, local asymptotic stability is also proven 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 in 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

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

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, $F(\dot{q})$ 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

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

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

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

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

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

where W^* is unknown constant weight, \widehat{W} is estimated weight, $\phi(q, \dot{q})$ is the neural approximation error, 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

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

where $\widehat{W} \in R^{n \times m}$, $\widehat{V} \in R^{m \times n}$, m is hidden node number, 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

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

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

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

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

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

also

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

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

$$\|\phi(x) - \phi(y)\| \leq k_\phi \|x - y\| \quad (10)$$

From (4) we know

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

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

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

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

where $U(q_t)$ is the potential energy of the robot, $\frac{\partial U}{\partial q} = G(q)$. 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 $\int_0^t \sigma(q) dq = \frac{1}{2}\sqrt{\pi} \text{erf}(q)$. 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 $\int_0^t \phi(q, \dot{q}) dq$ as

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

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

$$\tilde{q} = q^d - q \quad (14)$$

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

The classical industrial PID law is

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

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]

$$u = K_p \tilde{q} + K_d \dot{\tilde{q}} + \hat{f} \quad (16)$$

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

$$u = K_v r + \hat{f} \quad (17)$$

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 $\|r\| > \frac{B}{K_v}$, 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 ($r \rightarrow 0$) requires $K_v \rightarrow \infty$.

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

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

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

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

We require 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

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

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

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

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

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

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

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

The final closed-loop equation becomes

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

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

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

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

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

where α is positive design constant, it satisfies

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

Proof: See Appendix. ■

Remark 2: From above 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 [6][4][11][16][10]. No modeling information is needed. The upper or lower bounds of PID gains need the maximum eigenvalue of M in (25), it can be estimated without calculating M . For a robot with only revolute joints [2]

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

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

Remark 3: The main difference between our neural PID control with the other neural PD controllers is the stability condition is changed, we require the regulation error

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

The other neural PD controllers need

$$\|\tilde{q}\| > \frac{k_2}{K_v} \quad (30)$$

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. But (30) cannot be satisfied when $\|\tilde{q}\|$ becomes small, so K_v has to be increased.

Remark 4: If the unknown $f(q)$ is estimated by the multi-layer neural network (5). The modeling error (21) becomes

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

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

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

If the Lyapunov function in (54) is changed as

$$V_m = V + \frac{1}{2} \text{tr} \left(\tilde{V}^T K_v^{-1} \tilde{V} \right) \quad (33)$$

then the derivative of (33) is

$$\begin{aligned} \dot{V}_m &= \dot{V} - \dot{q}^T \tilde{W} \sigma(q [q\dot{q}]) \\ &+ \dot{q}^T \tilde{W} [\sigma(\widehat{V} [q\dot{q}]) + \sigma' \widehat{V} [q\dot{q}]] + \text{tr} \left(\tilde{V}^T K_v^{-1} \dot{\tilde{V}} \right) \end{aligned} \quad (34)$$

If the training rule (26) is changed as

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

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 anti-windup schemes are mainly classified into two types [34]: conditional integration and back-calculation. It has been shown that none of existed methods is 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:

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

where $\text{sat}[x, \nu_{\max}] = \begin{cases} x & \text{if } \|x\| < \nu_{\max} \\ \nu_{\max} & \text{if } \|x\| \geq \nu_{\max} \end{cases}$. ν_{\max} is a prescribed value to the integral term when the controller saturates. This approach is also called preloading [36]. Now the linear PID controller becomes nonlinear PID. The semi-global asymptotic stability has been analyzed by [6]. When ν_{\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), \bar{G} is the upper bound of $G(q)$. k_s is a design factor in the case of not all PID terms are subjected to saturation. For the controller (36), k_s can selected as $k_s = \frac{1}{4}$.

Following the process from (4) to (13), the neural PID with anti-windup controller (36) requires

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

where $\bar{\phi}$ is the upper bound of the neural estimator, 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 anti-windup is the neural estimator must be bounded. While the linear neural PID only requires the neural estimation error satisfy Lipschitz condition (10).

Since u_i^{\max} (or ν_{\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 $\left\| W^* \sigma(q, \dot{q}) \right\|$ smaller, such as multilayer neural network in Remark 3. There are several methods 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 \widehat{W} and \widehat{V} in (35) do not effect the stability property, we design an off-line method to find a better value for $\widehat{W}(0)$ and $\widehat{V}(0)$. If we let $\widehat{W}(0) = W_0$, $V(0) = V_0$, the algorithm (35) can make the identification error convergent, *i.e.*, $\widehat{W}(t)$ and $\widehat{V}(t)$ will make the identification error smaller than that of W_0 and V_0 . $\widehat{W}(0)$ and $\widehat{V}(0)$ are selected by following steps:

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

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]

$$v_i(s) = \frac{b_i s}{s + a_i} q_i(s), \quad i = 1 \cdots n \quad (38)$$

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

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

The linear PID control (19) becomes

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

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

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

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

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

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

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

Theorem 5: Consider robot dynamic (1) controlled by neural PID controller (40), if A and B of the velocity observer (38) satisfy

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

where α is positive design constant, provided that the PID control gains of (40) satisfy

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

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

$$\dot{\widehat{W}} = -K_w \sigma(q, v) [\alpha \tilde{q} + v + B^{-1}(\dot{v} + Av)]^T \quad (45)$$

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

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

in the domain of attraction

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

Proof: See Appendix. ■

Remark 6: The conditions (43) and (44) decide how to choose the PID gains. The first condition of (44) is

$$\begin{aligned} \lambda_m(K_p) &\geq \frac{1}{\alpha} \lambda_M(K_i) + \Omega \\ \Omega &= \frac{1}{\alpha} \left[\frac{\lambda_M(A^{-1}BK_i) + \frac{1+2\alpha}{2} k_g}{+ \frac{\alpha}{2} \lambda_M(K_d) + \frac{1}{2} \lambda_M(A^{-1}K_i)} \right] + \frac{1}{2\alpha} \lambda_M(K_p) \end{aligned} \quad (48)$$

the third conditions of (44) is $\lambda_m(K_p) \geq \frac{3}{\alpha} \lambda_M(K_i)$, 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. So K_i can not be big. These requirements are reasonable for our real control. If we select $B = \beta A + \alpha I$, form the third condition of (43), $\beta \geq \frac{1}{2}$. The second condition of (44) requires $\lambda_m(AB^{-1} - I) > \frac{1}{2}$, there exists $1 > \beta \geq \frac{1}{2}$ and a small α such that $\lambda_m[A(\beta A + \alpha I)^{-1} - I] > \frac{1}{2}$. 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 Figure 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 Figure 2. It is composed of a 3-DOF shoulder

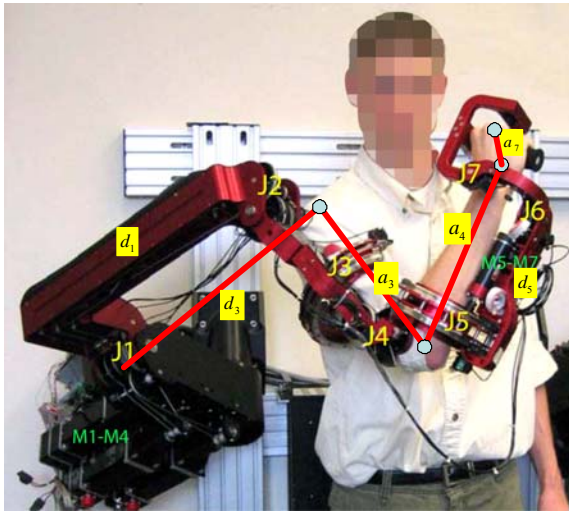


Fig. 1. The UCSC 7-DOF exoskeleton robot.



Fig. 3. The admittance control of the UCSC exoskeleton

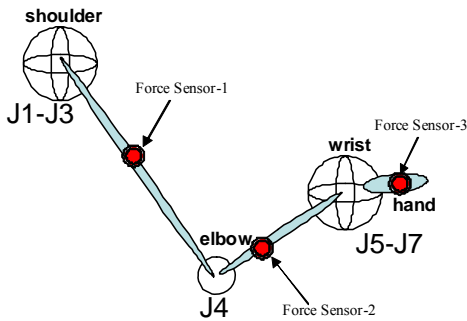


Fig. 2. Human arm vs. exoskeleton

(J1-J3), a 1-DOF elbow (J4) and a 3-DOF on wrist (J5-J7). J1-J3 are responsible for shoulder flexion-extension, adduction and internal-external rotation, J4 create elbow flexion-extension, J5-J7 are responsible for wrist flexion-extension, pronation-supination and radial-ulnar deviation. We regard J1, J2 and J3 in Figure 5 as three spherical joints of the human shoulder, see Figure 2. Also J5, J6 and J7 in Figure 5 are considered as three spherical joints of the human wrist.

The exoskeleton's height was adjusted for each user in a seated position, see Figure 3. The users 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 Figure 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 Figure 5. The objective of neural PID control is make the transient performance faster and less overshoot, such that human feel comfortable. In Figure 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.

The real-time control program operated in Windows XP



Fig. 4. The end-effector of the exoskeleton

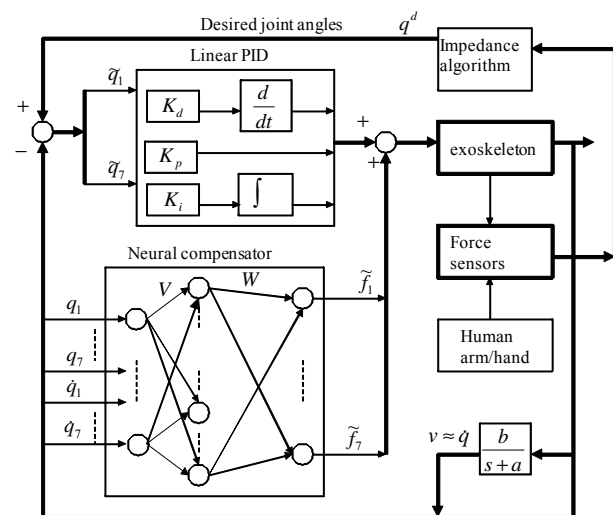


Fig. 5. The neural PID control of the exoskeleton.

with Matlab 7.1, Windows Real-Time Target and C++. All of the controllers employed a sampling frequency of $1kHz$. The properties of the exoskeleton with respect to base frame are shown in Table 1.

Table 1. 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 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 1 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 $\alpha = \frac{4\lambda_M(K_i)}{\lambda_m(K_p)}$ such that $\lambda_M(K_i) \leq \frac{\alpha}{3}\lambda_m(K_p)$ is satisfied. $\alpha = 0.08$, A is chosen as $A = \text{diag}(30)$, $\beta = \frac{7}{12}$, so $B = \text{diag}(17.58)$. The joint velocities are estimated by the standard filters

$$\tilde{q}(s) = \frac{bs}{s+a}q(s) = \frac{18s}{s+30}q(s) \quad (49)$$

The PID gains are chosen as

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

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

$$\sigma = \exp\left\{-\frac{(q_i - m_i)^2}{100}\right\}, \quad i = 1 \dots 7 \quad (51)$$

where m_i is selected randomly from 0 to 2 . The weights are updated by (45) with $K_w = 10$.

The control results of Joint-1 with neural PID control is shown in Figure 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), the control result is shown in Figure 6, marked "Linear PID-2". Because the steady-state error is so big, the integral gains are increased as

$$K_i = \text{diag}[50, 20, 30, 30, 10, 10, 10] \quad (52)$$

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

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

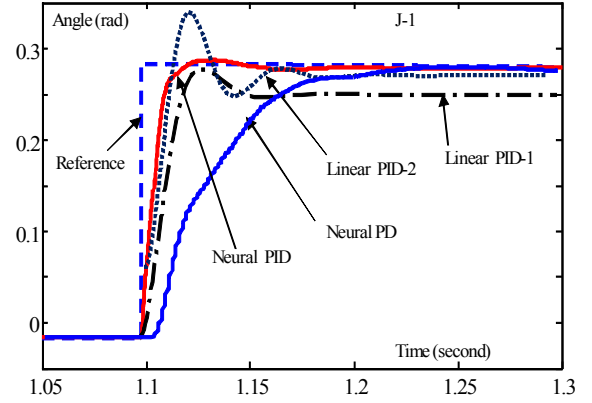


Fig. 6. Comparison of several PID controllers for Joint 1.

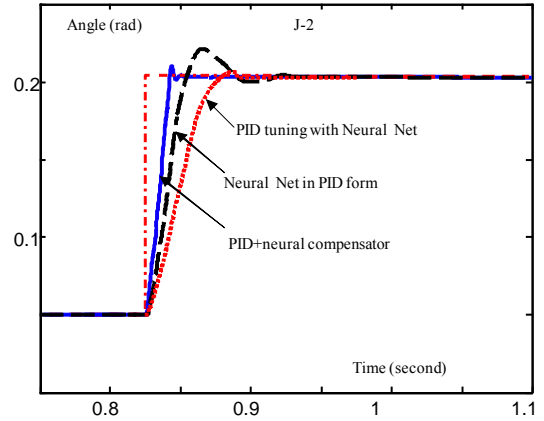


Fig. 7. Comparison of three neural PID controllers for Joint 2.

the control result is shown in Figure 6, marked "Neural PID". The response becomes very slow.

Now we use Joint-2 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 Figure 7. Here we use a three-layer neural network with 3 nodes which have integral, proportional and derivative properties. A backprogration-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 [?], 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 its slow response.

Finally, we use the other joints, Joint 3 to Joint 7, to compare our neural PID with the other popular robot controllers. The results are shown in Figure 8-12.

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

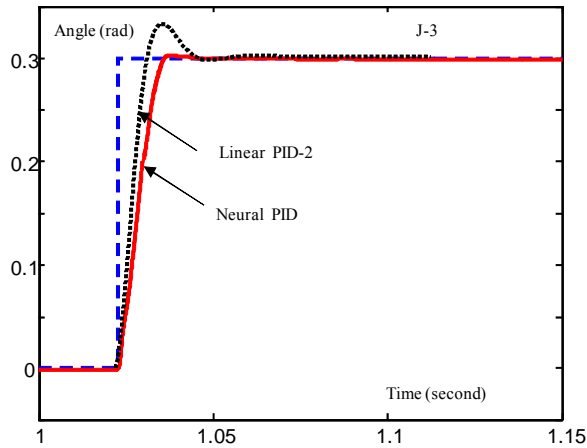


Fig. 8. Linear PID via neural PID for Joint 3.

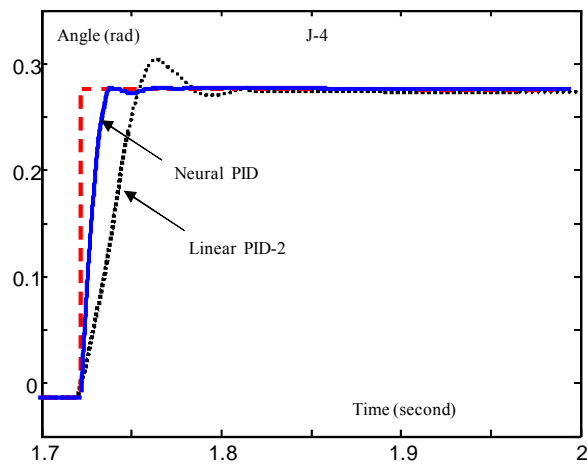


Fig. 9. Linear PID-2 via neural PID for Joint 4.

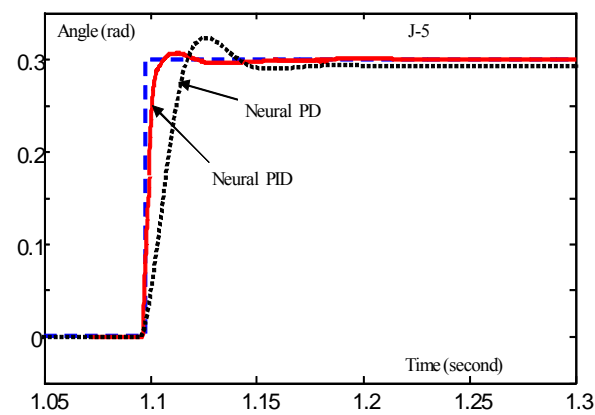


Fig. 10. Neural PD via neural PID for Joint 5.

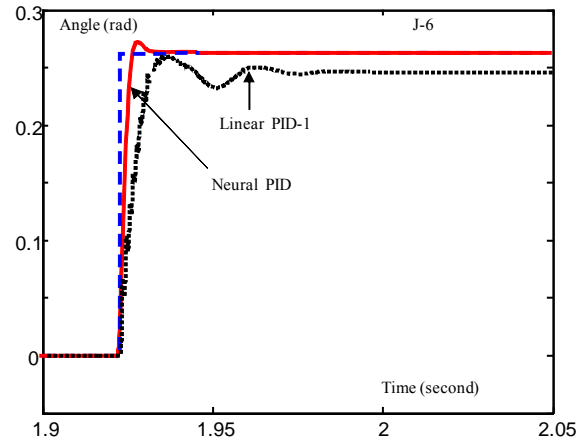


Fig. 11. Linear PID via neural PID for Joint 6.

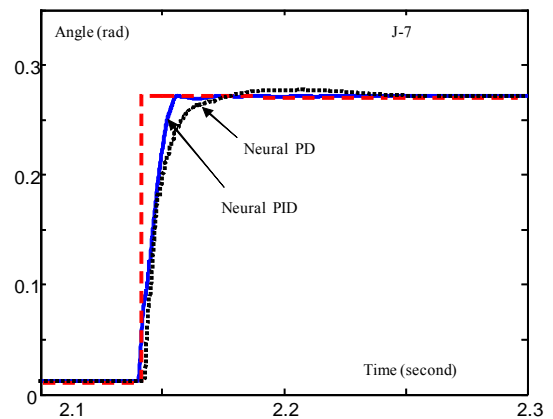


Fig. 12. Neural PD via neural PID for Joint 7.

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 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 produce slow learning process.

V. CONCLUSIONS

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 proven. 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.

VI. ACKNOWLEDGMENTS

The authors would like to thanks Levi Makaïo Miller for his help to complete the experiments.

VII. APPENDIX

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

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

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

$$\begin{aligned} V_1 = & \frac{1}{6} \tilde{q}^T K_p \tilde{q} + \tilde{q}^T \phi(q^d) + \frac{3}{2} \phi(q^d)^T K_p^{-1} \phi(q^d) \\ V_2 = & \frac{1}{6} \tilde{q}^T K_p \tilde{q} + \tilde{q}^T \tilde{\xi} + \frac{\alpha}{2} \tilde{\xi}^T K_i^{-1} \tilde{\xi} \\ V_3 = & \frac{1}{6} \tilde{q}^T K_p \tilde{q} - \alpha \tilde{q}^T M \dot{q} + \frac{1}{2} \tilde{q}^T M \dot{q} \\ V_4 = & \int_0^t \phi(q) d\tau - k_\phi + \frac{\alpha}{2} \tilde{q}^T K_d \tilde{q} + \frac{1}{2} \text{tr} \left(\tilde{W}^T K_w^{-1} \tilde{W} \right) \geq 0 \end{aligned} \quad (55)$$

It is easy to find

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

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

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

Because

$$y^T A x \leq \|y\| \|A x\| \leq \|y\| \|A\| \|x\| \leq |\lambda_M(A)| \|y\| \|x\| \quad (58)$$

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

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

Obviously, if

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

there exists

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

This means if K_p is sufficiently large or K_i is sufficiently small, (60) is established, and $V(\tilde{q}, \dot{q}, \tilde{\xi})$ is globally positive definite. Using $\frac{d}{dt} \int_0^t \phi(q, \dot{q}) dq = \frac{\partial \int_0^t \phi(q, \dot{q}) dq}{\partial q} \frac{\partial q}{\partial t} = \dot{q}^T \phi(q, \dot{q})$, $\frac{d}{dt} \phi(q^d) = 0$ and $\frac{d}{dt} [\tilde{q}^T \phi(q^d)] = \tilde{q}^T \dot{\phi}(q^d)$, the derivative of V is

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

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

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

And

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

If

$$\|\tilde{q}\| \leq \frac{\lambda_M(M)}{\alpha k_c} \quad (65)$$

and

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

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

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

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

\dot{V} is negative semi-definite on the ball Σ . There exists a ball Σ of radius $\sigma > 0$ centered at the origin of the state space on which $\dot{V} \leq 0$. 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

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

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 $t \geq 0$, it is necessary and sufficient that $\tilde{q} = \dot{q} = 0$ for all $t \geq 0$. Therefore it must also hold that $\tilde{\xi} = 0$ for all $t \geq 0$. We conclude that from the closed-loop system (24), if $x(t) \in \Omega$ for all $t \geq 0$, then

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

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

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

$$\|\tilde{q}\| \leq \frac{\lambda_M(M)}{k_c} \lambda_M(K_i) \lambda_m(K_p) \quad (70)$$

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

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

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

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

Here V_1 and V_2 are the same as (55), i.e.

$$\lambda_M(K_i) \leq \frac{\alpha}{3} \lambda_m(K_p) \quad (73)$$

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

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

Because $\lambda_m(AB) \leq \lambda_m(B^{-1}) \lambda_M(A)$ and $\lambda_m(B^{-1}) = \frac{1}{\lambda_M(B)}$, it is easy to find that,

$$\text{if } \lambda_M(A^{-1}) \leq \sqrt{\lambda_m(B^{-1} K_d) \lambda_m((A^{-1} B))} \text{ or } \frac{\lambda_M(A)}{\lambda_m^2(A)} \leq \lambda_m(K_d) \frac{\lambda_m(B)}{\lambda_M(B)}$$

$$V_4 \geq \frac{1}{2} \left(\frac{1}{2} \lambda_m(B^{-1} K_d) \|v\|^2 - 2 \lambda_M(A^{-1}) \|v\| \|\tilde{\xi}\| + 2 \lambda_m((A^{-1} B)) \|\tilde{\xi}\|^2 \right) \geq 0 \quad (75)$$

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

$$V_5 = \frac{1}{2} \left[\frac{1}{2} v^T K_d B^{-1} v + 2 v^T M \dot{q} + \frac{1}{2} \tilde{q}^T M \dot{q} \right] \geq 0 \quad (76)$$

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

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

This means 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

$$\begin{aligned} \dot{V} &\leq -\dot{q}^T B M \dot{q} - v^T A B^{-1} K_d v - \alpha \tilde{q}^T K_p \tilde{q} + \alpha k_g \|\tilde{q}\|^2 \\ &+ \alpha \tilde{q}^T M \dot{q} + k_c \|\alpha \tilde{q} - v\| \|\tilde{q}\|^2 \\ &+ \tilde{q}^T K_i \tilde{q} + \tilde{q}^T A^{-1} B K_i \tilde{q} + v^T K_d v + \frac{1}{2} k_g \|v\|^2 + \frac{1}{2} k_g \|\tilde{q}\|^2 \\ &+ \tilde{q}^T (\alpha K_d - K_p - A^{-1} K_i) v + \tilde{q}^T A M v \\ &+ \text{tr} \left[\tilde{W}^T \left(K_w^{-1} \tilde{W} + \sigma(q, v) \dot{q}^T + \alpha \sigma(q, v) \tilde{q}^T + \sigma(q, v) v^T \right) \right] \end{aligned} \quad (78)$$

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

$$\begin{aligned} \dot{V} &\leq -\dot{q}^T \left\{ \begin{array}{l} \lambda_m(BM - \alpha M) - k_c \|\alpha \tilde{q} - v\| \\ -\frac{1}{2\kappa(M)} \lambda_M(AM) \end{array} \right\} \dot{q} \\ &- v^T \left(\begin{array}{l} \lambda_m(AB^{-1}K_d - K_d) - \frac{1}{2} k_g - \frac{1}{2\alpha} \lambda_M(K_e) \\ -\frac{\kappa(M)}{2} \lambda_M(AM) \end{array} \right) v \\ &- \tilde{q}^T \left(\begin{array}{l} \lambda_m(\alpha K_p - K_i - A^{-1} B K_i) - \alpha k_g \\ -\frac{1}{2} k_g - \frac{\alpha}{2} \lambda_M(K_e) \end{array} \right) \tilde{q} \end{aligned} \quad (79)$$

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

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

It is the attraction area (47).

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

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

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

$$\lambda_m(\alpha K_p) \geq \lambda_M(K_i) + \lambda_M(A^{-1} B K_i) + \frac{1 + 2\alpha}{2} k_g + \frac{\alpha}{2} \lambda_M(K_e)$$

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

REFERENCES

- [1] Y.Jin, Decentralized Adaptive Fuzzy Control of Robot Manipulators, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.28, No.1, 47-57, 1998
- [2] M.W.Spong and M.Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons Inc., Canada, 1989.
- [3] F.L.Lewis, D.M.Dawson, C.T.Abdallah, *Robot Manipulator Control: Theory and Practice*, 2nd Edition, Marcel Dekker Inc, New York, NY 10016, 2004
- [4] S. Arimoto, Fundamental problems of robot control: Part I, Innovations in the realm of robot servo-loops, *Robotica*, vol.13, No.1, 19-27, 1995.
- [5] D.Sun, S.Hu, X.Shao, and C.Liu, Global Stability of a Saturated Nonlinear PID Controller for Robot Manipulators, *IEEE Transactions on Control Systems Technology*, Vol.17, No.4, 892-899, 2009
- [6] J.Alvarez-Ramirez, R.Kelly, I.Cervantes, Semiglobal stability of saturated linear PID control for robot manipulators, *Automatica*, vol.39, 989-995, 2003
- [7] R.Ortega, A.Loria, R.Kelly, A Semiglobally Stable Output Feedback PI²D Regulator for Robot Manipulators, *IEEE Transactions on Automatic Control*, Vol.40, No.0, 1432-1436, 1985
- [8] 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 Transactions on Robotics and Automation*, Vol.19, No.6, 967-976, 2003

- [9] T.Dierks, S.Jagannathan, Neural Network Output Feedback Control of Robot Formations, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.40, No.4, 383-399, 2010
- [10] P.Rocco, Stability of PID control for industrial robot arms, *IEEE Transactions on Robotics and Automation*, VOL.12, NO. 4, 606-614,1996.
- [11] R.Kelly, V.Santibáñez, L.Perez, *Control of Robot Manipulators in Joint Space*, Springer-Verlag London, 2005.
- [12] P.Tomei, Adaptive PD Controller for Robot Manipulators, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, 565-570,1991.
- [13] B.Paden, R.Panja, Globally asymptotically stable PD+ controller for robot manipulators, *International Journal of Control*, Vol. 47, No. 6, 1697-1712, 1988
- [14] K.Dupree, C-H.Liang, G.Hu, W. E.Dixon, Adaptive Lyapunov-Based Control of a Robot and Mass-Spring System Undergoing, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.38, No.4, 1050-1061, 2008
- [15] Z.Qu, D.M. Dawson, S. Y. Lim, J.F. Dorsey, A New Class of Robust Control Laws for Tracking of Robots, *International Journal of Robotics Research*, Vol. 13, No.4, 355-363, 1994
- [16] E.V. L. Nunes, L.Hsu, F.Lizarralde, Arbitrarily small damping allows global output feedback tracking of a class of Euler-Lagrange systems, *2008 American Control Conference*, Seattle, USA, 378-382, 2008
- [17] H-X. Li, L.Zhang, K-Y.Cai, G.Chen, An Improved Robust Fuzzy-PID Controller With Optimal Fuzzy Reasoning, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.35, No.6, 1283-1294, 2005
- [18] E.Harinath,G.Mann, Design and Tuning of Standard Additive Model Based Fuzzy PID Controllers for Multivariable Process Systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.38, No.8, 667-674, 2008
- [19] F.L.Lewis, Nonlinear Network Structures for Feedback Control, *Asian Journal of Control*, vol. 1, no. 4, pp.205-228,1999.
- [20] C-S.Chen, Dynamic Structure Neural-Fuzzy Networks for Robust Adaptive Control of Robot Manipulators, *IEEE Trans. Industrial Electronics*, VOL. 55, NO. 9, 3402-3414, 2008
- [21] M.J.Er and Y.Gao, Robust Adaptive Control of Robot Manipulators Using Generalized Fuzzy Neural Networks, *IEEE Trans. Industrial Electronics*, VOL. 50, NO. 3, 620-628, 2003
- [22] P.A.Ioannou and J.Sun, *Robust Adaptive Control*, Prentice-Hall, Inc, Upper Saddle River: NJ, 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, 2009.
- [24] G.M. Scott, J. W.Shavlik, W. H. Ray, Refining PID Controllers Using Neural Networks, *Neural Computation*, Vol. 4, No. 5, 746-757, 1992
- [25] H. J. Uang and C. C. Lien, Mixed H₂/H_∞ PID tracking control design for uncertain spacecraft systems using a cerebellar model articulation controller, *IEE Proceedings - Control Theory and Applications*, vol. 153, no. 1, pp. 1-13, 2006.
- [26] G. K. I. Mann, B-G. Hu, R.G. Gosine, Two-Level Tuning of Fuzzy PID Controllers, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.31, No.2, 263-269, 2001
- [27] S-J. Ho, L-S.Shu, 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, 421-434, 2006
- [28] D-Li Yu, T. K. Chang, D-W.Yu, Fault Tolerant Control of Multivariable Processes Using Auto-Tuning PID Controller, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.35, No.1, 32-43, 2005
- [29] F. Karray, W. Gueaieb, S. Al-Sharhan, The Hierarchical Expert Tuning of PID Controllers Using Tools of Soft Computing, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.35, No.6, 1283-1294, 2005
- [30] G.D'Emilia, A.Marra, E.Natale, Use of neural networks for quick and accurate auto-tuning of PID controller, *Journal Robotics and Computer-Integrated Manufacturing*, Volume 23 Issue 2, 170-179, 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. Industrial Electronics*, VOL. 45, NO. 2, 307-318, 1998.No.1, 131-147, 2002.
- [32] F.L.Lewis, K.Liu, and A.Yesildirek, Neural net robot controller with guaranteed tracking performance, *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 703-715, 1995.
- [33] J. J.Slotine, W.Li, Adaptive manipulator control: A case study, *IEEE Transactions on Automatic Control*, Vol.33, No.11, 995-1003, 1988
- [34] A. Visioli, Modified anti-windup scheme for PID controllers, *IEE Proc.-Control Theory Appl.*, Vol.161, No 1., 49-54, 2003
- [35] Karl J. Astrom, T. Hagglund, *PID Controllers: Theory, Design and Tuning*, ISA Press, Research Triangle Park, North Carolina, 1995
- [36] F. Greg Shinskey, *Process-control Systems: Application, Design, Adjustment*, McGraw-Hill, New York. USA. 1996
- [37] J.Arifovica, R.Gencay, Using genetic algorithms to select architecture of a feedforward artificial neural network, *Physica A*, Vol.289, 574-594, 2001
- [38] K.Suzuki, I.Horina and N.Sugie, A Simple Neural Network Pruning Algorithm with Application to Filter Synthesis, *Neural Processing Letters*, Vol.13, 43-53, 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, both in Electrical Engineering, from Northeastern University, Shenyang, China, in 1992 and 1995, respectively. From 1995 to 1996, he served as a Lecturer in the Department of Automatic Control at Northeastern University, Shenyang, China. In 1996, he joined CINVESTAV-IPN, Mexico, where he is currently a professor in the Departamento de Control Automatico. He held research positions with the Instituto Mexicano del Petroleo from 2002 to 2003. He was a senior visiting research fellow at Queen's University Belfast from 2006 to 2007, a visiting associate professor at University of California Santa Cruz from 2009 to 2010. He also holds a visiting professorship at Northeastern University in China from 2006 until now. Dr.Wen Yu serves as an associate editor of *Neurocomputing*, and *Journal of Intelligent and Fuzzy Systems*. He is a member of Mexican Academy of Science.

Jacob Rosen ****.