Stable PID Control for Robot Manipulators with Neural Compensation

Wen Yu, Xiaoou Li

Abstract—In order to minimize steady-state error with respect to uncertainties in robot control, the integral gain of PID control should be increased. Another method is to add a compensator to PD control, such as neural compensator, but the derivative gain of this PD control should be large enough. These two approaches deteriorate transient performances. In this paper, the popular neural PD is extended to neural PID control. The semiglobal asymptotic stability of the neural PID control is proven. The 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. 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 [1]. 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 [2]. Therefore, the integral gain of PID controller cannot be set too big in industrial practice.

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 [3]; The integral term was saturated by a nonlinear function in [4]; The input was saturated in [5]; An extra integral term in the filtered position was added in [6]; Variable structure controller was combined with PID control in [7]. Only a few researchers worked on the linear PID. The stability (not asymptotic stability) of the linear PID control was proven in [8], where the robot dynamic was re-written in a decoupled linear system and a bounded nonlinear system. In [9], 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 [1], such as adaptive gravity compensation [10], desired gravity compensation [9], and PD with position measurement [11]. 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 [12], PD control with nonlinear gains [13], and PD control with sliding mode compensation [7]. 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 [14], neural compensator [15] and fuzzy-neural compensator [16],[17]. The basic idea behind these controllers is to use a filtered tracking error in the Lyapunov-based analysis [2]. By proper weight tuning algorithms, which are similar with robust adaptive control methods [18], 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 $K_v$. However, increasing $K_d$ causes long settling time. Only when $K_v \rightarrow \infty$, the static tracking error converges to zero [15]. The simplest method to decrease the static 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 neural control. First, the neural networks are formed into PID structure [19],[20],[21]. 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. Second, neural networks are used to tune the parameters of PID controllers [22],[23]. 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 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 [9]. Although PID control has been used in industrial robots for a long time, there is few explicit stability analysis on it. From the best of our knowledge, theory analysis for this neural PID control is still not published.

Wen Yu is with the Departamento de Control Automatico, CINVESTAV-IPN, Mexico City, Mexico. Xiaoou Li is with the Departamento de Computacion, CINVESTAV-IPN, Mexico City, Mexico.
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. 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

\[ M(q)\ddot{q} + C(\dot{q}, \dot{q}) \dot{q} + G(q) + F(\dot{q}) = u \]  

where \( q \in \mathbb{R}^n \) represents the link positions. \( M(q) \) is the inertia matrix, \( C(q, \dot{q}) = \{c_{ik}\} \) 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. \( u \in \mathbb{R}^n \) is control input. The friction \( F(\dot{q}) \) is represented by the Coulomb friction model

\[ F(\dot{q}) = K_f \dot{q} + K_{f2} \tanh (k_{f3} \dot{q}) \]  

where \( k_f \) is a large positive constant, such that \( \tanh (k_{f3} \dot{q}) \) can approximate \( \text{sign}(\dot{q}) \), \( k_f \) and \( K_{f2} \) are positive coefficients. In this paper we use a simple model for the friction as in [2] and [9].

\[ F(\dot{q}) = K_f \dot{q} \]  

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

\[ f(q, \dot{q}) = G(q) + F(\dot{q}) \]  

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

where \( W^* \) is unknown constant weight, \( \hat{W} \) is estimated weight, \( \phi(q, \dot{q}) \) is the approximation error of \( f(q, \dot{q}) \), \( \sigma(\cdot) \) is sigmoid vector function as

\[ \sigma_i(x) = a_i / (1 + e^{-b_i x}) - c_i \]

where \( a_i, b_i \) and \( c_i \) are known positive constants. 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

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

where \( \hat{W} \in \mathbb{R}^{n \times m}, \hat{V} \in \mathbb{R}^{m \times n}, m \) is hidden node number, \( \hat{V} \) 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 [1] which will be used to prove stability.

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

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

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

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

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

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

also

\[ M(q) = C(q, \dot{q}) + C(q, \dot{q})^T \]  

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

\[ \|\phi(x) - \phi(y)\| \leq k_\phi \|x - y\| \]

From (4) we know

\[ G(q) + F(\dot{q}) = W^* \sigma(q, \dot{q}) + \phi(q, \dot{q}) \]

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 \( \int \phi(q) dq \) can be estimated as

\[ \int_0^t \phi(q) dq = \int_0^t G(q) dq + \int_0^t F(\dot{q}) dq - \int_0^t W^* \sigma(q) dq = U(q_t) + K_{f2}q_t - K_{f1}q_0 - W^* \ln \left( e^{2\omega t} + 1 \right) - q_t - \ln \left( e^{2\omega t} + 1 \right) - q_0 \]

where \( U(q_t) \) is the potential energy of the robot. \( \frac{\partial U}{\partial q} = G(q) \), the active function \( \sigma(\cdot) \) is hyperbolic tangent. Since each term of the above equation has a constant lower bound, we define the lower bound of \( \int_0^t \phi(q) dq \) as

\[ k_{\phi} = \min_q \left\{ \int_0^t \phi(q, \dot{q}) d\tau \right\} \]

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

\[ \dot{q} \to 0 \]  

\[ \ddot{q} \to 0 \]  

when initial conditions are in arbitrary large domain of attraction.
The classical industrial PID law is
\[
\frac{d\tilde{q}}{dt} + M^{-1}\left( C\tilde{q} + \tilde{W}\sigma(q, \dot{q}) + \phi(q, \dot{q}) \right) = K_i\tilde{q} + K_d\tilde{q} - K_p\tilde{q}
\]
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 [2]
\[
u = K_pq + K_d\dot{q} + \tilde{f}
\]
(16)

where \( \tilde{f} = \tilde{W}\sigma(q, \dot{q}) \). With the filtered error \( r = \dot{q} + \Delta\dot{q} \), (16) becomes
\[
u = K_vr + \tilde{f}
\]
(17)

The control (17) avoids integrator problems in (15). Unlike industrial PID control, they cannot reach asymptotic stability. [24] proved that the filtered tracking error satisfied
\[\lim_{K_v \to \infty} r = 0.\]
In order to decrease \( r, K_d \) has to be increased, and this causes long settling time. 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),
\[
u = K_p\tilde{q} + K_d\tilde{q} + K_i\int_0^t \tilde{f}(\tau) d\tau + \tilde{f}
\]
(18)

Because in regulation case \( q^d = 0, \dot{q} = -\dot{q} \), the PID control law can be expressed via the following equations
\[
u = K_p\tilde{q} - K_d\dot{q} + \xi + \tilde{W}\sigma(q, \dot{q}) \quad \xi = K_i\tilde{q}, \quad \xi(0) = \xi_0
\]
(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
\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \dot{\tilde{f}}(q, \dot{q}) = K_p\tilde{q} - K_d\dot{q} + \xi
\]
(20)

where \( \dot{\tilde{f}} = f - \tilde{f} \)
\[
\dot{\tilde{f}} = W^*\sigma(q) + \phi(q) - \tilde{W}\sigma(q) = W\sigma(q) + \phi(q)
\]
(21)

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

In order to move the equilibrium to origin, we define
\[
\tilde{\xi} = \xi - \phi(q^d)
\]
(23)

The final closed-loop equation becomes
\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \dot{\tilde{f}}(q, \dot{q}) = K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} + \tilde{\phi}(q^d)
\]
(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 \( \dot{q} \) 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 point \([\xi, \dot{q}, \dot{\dot{q}}]^T = 0 \), provided that control gains satisfy
\[
\lambda_m(K_p) \geq \frac{3}{2}k_\phi, \quad \lambda_M(K_d) \leq \beta \frac{\lambda_m(K_p)}{\lambda_M(M)} \quad \lambda_m(K_d) \geq \beta \lambda_M(M)
\]
(25)

where \( \beta = \sqrt{\frac{\lambda_m(M)\lambda_m(K_p)}{3}} \), \( k_\phi \) satisfies (10), and the weight of the neural networks (4) is tuned by
\[
\dot{W} = -K_w\sigma(q, \dot{q})(q + \alpha_1\dot{q})^T
\]
(26)

where \( \alpha_1 > 0 \) is the positive design constant, it satisfies
\[
\sqrt{\frac{\lambda_m(M)\lambda_m(K_p)}{3}} \geq \alpha_1 \geq \frac{3}{\lambda_m(K_p)}\lambda_m(K_p)
\]
(27)

**Proof:** We construct a Lyapunov function as
\[
V = \frac{1}{2}q^T M\tilde{q} + \frac{1}{2}q^T K_p\tilde{q} + \int_0^t \phi(q, \dot{q}) dq - k_\phi + \dot{\tilde{f}}^T\phi(q^d) + \frac{1}{2}\phi(q^d)^T K_p^{-1}\phi(q^d) + \frac{1}{2}\tilde{\xi}^TK_i^{-1}\tilde{\xi}
\]
where \( k_\phi \) is defined in (13) such that \( V(0) = 0 \). \( \alpha_1 + \alpha_2 / 2 \) is a design positive constant.

1) We first prove \( V \) is a Lyapunov function, \( V \geq 0 \). The term \( \frac{1}{2}q^T K_p\tilde{q} \) is separated into three parts, and \( V = \sum_{i=1}^4 V_i \)
\[
V_1 = \frac{1}{2}q^T K_p\tilde{q} + \frac{1}{2}\dot{q}^T\phi(q^d) + \frac{1}{2}\phi(q^d)^T K_p^{-1}\phi(q^d)
\]
\[
V_2 = \frac{1}{2}q^T K_p\tilde{q} + \frac{1}{2}q^T K_i^{-1}\xi + \frac{1}{2}\xi^T K_i^{-1}\tilde{\xi}
\]
\[
V_3 = \frac{1}{2}q^T K_p\tilde{q} - \alpha_1q^TM\tilde{q} + \frac{1}{2}\dot{q}^TM\tilde{q}
\]
\[
V_4 = \int_0^t \phi(q) dq - k_\phi + \frac{1}{2}\dot{q}^T K_d\tilde{q} + \frac{1}{2}\dot{q}^T (\tilde{W}^TW)^{-1}\tilde{W}
\]
(28)

From (13) we know \( V_4 \geq 0 \). It is easy to find
\[
V_1 = \frac{1}{2}\left[ \frac{1}{3}K_p I + \frac{I}{3K_p} \right] \left[ \frac{1}{3}K_p I + \frac{I}{3K_p} \right]
\]
(29)
Since \( K_p \geq 0 \), \( V_1 \) is a semi positive definite matrix, \( V_1 \geq 0 \). When \( \alpha \geq \frac{3}{\lambda_m(K_i^{-1}) \lambda_m(K_p)} \),

\[
V_2 \geq \frac{1}{6} \lambda_m(K_p) \| \dot{q} \|^2 - \frac{1}{2} \lambda_m(K_p) \| \ddot{q} \|^2 + \frac{1}{2} \lambda_m(K_p) \| q \|^2
\geq \frac{1}{2} \left( \frac{1}{3} \lambda_m(K_p) \| q \|^2 - \| q \| \left\| \dot{q} \right\| \right) + \frac{3}{\lambda_m(K_p)} \| \dot{q} \|^2
\geq \frac{1}{2} \left( \sqrt{\frac{3}{\lambda_m(K_p)}} \| q \| \right)^2 \geq 0
\]

(31)

Because

\[
y^T Ax \leq \| y \| \| A x \| \leq \| y \| \| A \| \| x \| \leq |\lambda_M(A)| \| y \| \| x \|
\]

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

\[
V_3 \geq \frac{1}{6} \lambda_m(K_p) \| \dot{q} \|^2 - \alpha \lambda_m(M) \| q \|^2 + \frac{1}{2} \lambda_m(K_p) \| \dot{q} \|^2
\geq \frac{1}{2} \left\{ \lambda_m(M) \| q \|^2 - 2 \alpha \lambda_m(M) \| q \|^2 \right\} + \frac{3}{\lambda_m(K_p)} \| \dot{q} \|^2
\]

\[
= \frac{1}{2} \left( \sqrt{\frac{3}{\lambda_m(K_p)}} \| q \| \right)^2 \geq 0
\]

(32)

Obviously, if

\[
\sqrt{\frac{3}{\lambda_m(K_p)}} \lambda_m(K_p) \leq \lambda_M(M)
\]

there exists

\[
\frac{3}{\lambda_m(K_p)} \geq \alpha \geq \frac{3}{\lambda_m(K_i^{-1}) \lambda_m(K_p)}
\]

(33)

This means if \( K_p \) is sufficiently large or \( K_i \) is sufficiently small, (34) is established, and \( V \left( \dot{q}, \ddot{q}, \dddot{q} \right) \) is globally positive definite.

2) We now prove \( \dot{V} \leq 0 \). Using \( \frac{d}{dt} \int_0^t \phi(q, \dot{q}) \, dq = \int_0^t \frac{\partial}{\partial q} \phi(q, \dot{q}) \, dq + \frac{\partial}{\partial \dot{q}} \phi(q, \dot{q}) \, dq \), \( \frac{\partial}{\partial \dot{q}} \phi(q, \dot{q}) = 0 \) and \( \frac{\partial}{\partial \dot{q}} \left[ \dot{q} \phi(q, \dot{q}) \right] = \dot{q} \phi(q, \dot{q}) \), the derivative of \( V \) is

\[
\dot{V} = \dot{q}^T M \ddot{q} + \frac{1}{2} \dot{q}^T T \dot{q}_p \ddot{q} + \phi(q, \dot{q}) \ddot{q} + \dddot{q} \phi(q, \dot{q})
\quad + \dot{q} \dot{q}^T \dot{K}_p \dot{q} + \alpha \xi^T \dot{q} \dot{q} + \alpha \xi^T \dot{q} + \alpha \xi^T \dot{q} + \alpha \xi^T \dot{q} + \alpha \xi^T \dot{q} + \alpha \xi^T \dot{K}_p \dot{q}
\]

(36)

Using (8), the first three terms of (36) become

\[
-\dot{q}^T \phi(q) - \dot{q}^T K_d \ddot{q} + \dddot{q} \phi(q) + \dddot{q}^T \dot{W} \sigma(q, \dot{q})
\]

(37)

Because \( \dddot{q}^T \phi(q) = -\dot{q}^T \phi(q) \) and \( \dddot{q} = K_i \dddot{q} \), the first ten terms of (36) are

\[
-\dot{q}^T K_d \ddot{q} + \alpha \dddot{q}^T \dddot{q} + \alpha \dddot{q}^T \dddot{q} + \alpha \dddot{q}^T \dddot{q} + \alpha \dddot{q}^T \dddot{q} + \alpha \dddot{q}^T \dddot{q} + \alpha \dddot{q}^T \dddot{q}
\]

(38)

Now we discuss the last two terms of (36). From (9), we have

\[
\dot{q}^T M \ddot{q} = \dot{q}^T C \dot{q} + \dot{q}^T C \dot{q}
\]

(39)

From (24),

\[
\dot{q}^T M \ddot{q} = -\dot{q}^T C \dot{q} - \dot{q}^T \phi(q) + \dot{q}^T K_p \ddot{q} - \dot{q}^T K_d \ddot{q}
\quad + \dot{q}^T \dot{W} \sigma(q, \dot{q})
\]

(40)

Since \( \dddot{q} \ddot{q} \), using (7) and (10) the last two terms of (36) are

\[
-\alpha \dot{q}^T K_p \ddot{q} + \dot{q}^T C \dot{q} + \dot{q}^T T \dot{q} + \dot{q}^T \dot{W} \sigma(q, \dot{q})
\quad + \dot{q}^T \dot{q} + \dot{q}^T \phi(q) + \dot{q}^T \dot{W} \sigma(q, \dot{q})
\]

\[
\leq \alpha \dot{q}^T M \ddot{q} - \dot{q}^T K_p \ddot{q} + \alpha k_c \| q \|^2 + \alpha k_g \| q \|^2
\]

(41)

From (38) and (41)

\[
\dot{V} \leq -\dot{q}^T (K_d - \alpha M - \alpha k_c \| q \|) \dot{q} - \dot{q}^T (\alpha K_p - K_i - \alpha k_g) \ddot{q}
\]

\[
\leq -\alpha \lambda_m(K_d) \alpha \lambda_M(M) \alpha \lambda_c \| \ddot{q} \|^2 + \alpha \lambda_m(K_p) \alpha \lambda_M(K_i) + k_g
\]

If

\[
\| \ddot{q} \| \leq \frac{\lambda_m(M)}{\alpha k_c}
\]

(42)

and

\[
\lambda_m(K_d) \geq (1 + \alpha) \lambda_M(M)
\]

\[
\lambda_m(K_p) \geq \frac{3}{\alpha \lambda_m(K_i)}
\]

(43)

then \( \dot{V} \leq 0 \). If \( \| \ddot{q} \| \) decreases. From (35), if

\[
\lambda_m(K_d) \geq \lambda_M(M) + \sqrt{\frac{3}{\lambda_m(M)}} \lambda_m(K_p)
\]

(44)

then (44) is established. Using (34) and \( \lambda_m(K_i^{-1}) = \lambda_m(K_i^{-1}) \lambda_m(K_p) \lambda_M(K_i) + k_g \)

(45)

(3) We prove semiglobally asymptotic stability. Define a ball \( \Sigma \) of radius \( \sigma > 0 \) centered at the origin of the state space, which satisfies this condition

\[
\Sigma = \left\{ \ddot{q} : \| \ddot{q} \| \leq \frac{\lambda_m(M)}{\alpha k_c} = \sigma \right\}
\]

(46)

\( V \) is negative semi-definite on the ball \( \Sigma \). There exists a ball \( \Sigma \) 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 \( \Omega \) as

\[
\Omega = \left\{ x(t) = \dot{\ddot{q}}, \dddot{q}, \dddot{q} \right\} \in R^3 : \dot{V} = 0 \}
\]

(47)

From (36), \( \dot{V} = 0 \) if and only if \( \dddot{q} = 0 \). For a solution \( x(t) \) to belong to \( \Omega \) for all \( t \geq 0 \), it is necessary and sufficient that \( \dddot{q} = 0 \) for all \( t \geq 0 \). Therefore it must also hold that \( \dddot{q} = 0 \) 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

\[
\phi(q, \dot{q}) = \phi(q, 0) = \dddot{q} + \dddot{q} = 0
\]

(48)
implies that \( \dot{\xi} = 0 \) for all \( t \geq 0 \). So \( x(t) = [\tilde{q}, \dot{q}, \ddot{q}] = 0 \in R^3 \) is the only initial condition in \( \Omega \) 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 \( \frac{1}{n} \leq \lambda_m(K_i^{-1}) \lambda_m(K_p) \), the upper bound for \( ||\dot{q}|| \) can be
\[
||\dot{q}|| \leq \frac{\lambda_M(M)k_e}{k_v}\lambda_M(K_i)\lambda_m(K_p)
\] (49)

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.

**Remark 1:** 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 [5][3][9][13][8]. 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 [1]
\[
\lambda_M(M) \leq \beta, \quad \beta \geq n\left(\max_{i,j} |m_{ij}|\right)
\] (50)
where \( m_{ij} \) stands the \( ij \)-th element of \( M, M \in R^{n \times n}. \) \( A \beta \) can be selected such that it is much bigger than all elements.

**Remark 2:** The main difference between our neural PID control with the other neural PD controllers is the stability condition. Our condition is
\[
||\dot{q}|| < k_1\lambda_M(K_i)\lambda_m(K_p)
\] (51)
The condition of the other neural PD controllers is
\[
||\dot{q}|| > \frac{k_2}{k_v}
\] (52)
where \( k_1 \) and \( k_2 \) are positive constants. Obviously, if the initial condition is not worse and satisfies (51), (51) is always satisfied, and \( ||\dot{q}|| \) will decrease to zero. But (52) cannot be satisfied when \( ||\dot{q}|| \) 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
\[
\dot{f} = f - \hat{f} = W^*\sigma(V^* \left[qq\right] + \phi(q, q) - W\sigma(V \left[qq\right])
\]
\[
= \hat{W}\sigma(\hat{V} \left[qq\right]) - W^*\sigma(V^* \left[qq\right]) + W^*\sigma(V^* \left[qq\right]) + \phi(q, q)
\]
\[
= \hat{W}\sigma(\hat{V} \left[qq\right]) + W^*\sigma(V^* \left[qq\right]) + \phi(q, q)
\]
\[
= \hat{W}\sigma(\hat{V} \left[qq\right]) + W^*\sigma(V^* \left[qq\right]) + \phi(q, q)
\] (53)
where \( \phi_i(q) = \epsilon_i + \phi(q, q) \), \( \epsilon_i \) is Taylor approximation error. The closed-loop equation (24) becomes
\[
M(q) \ddot{q} + C(q, \dot{q}) \ddot{q} + \hat{W}\sigma(\hat{V} \left[qq\right]) + \phi(q, q)
\]
\[
= K_p\ddot{q} - K_d\dot{q} + \ddot{\xi} + \phi(q^d)
\]
\[
\dot{\xi} = K_i\dot{q}
\] (54)
If the Lyapunov function in (28) is changed as
\[
V_m = V + \frac{1}{2}\text{tr}
\]
\[
\left(V^T K_v^{-1} V\right)
\] (55)
the derivative of (55) is
\[
\dot{V}_m = \dot{V} - q^T \hat{W}\sigma(q \left[qq\right])
\]
\[
+ q^T W\left[\sigma(\hat{V} \left[qq\right]) + \sigma'\hat{V} \left[qq\right]\right] + \text{tr}
\]
\[
\left(V^T K_v^{-1} \hat{V}\right)
\] (56)
If the training rule (26) is changed as
\[
\ddot{W} = -K_w \left\{\sigma(\hat{V} \left[qq\right]) + \sigma'\hat{V} \left[qq\right]\right\} (q + \alpha\ddot{q})^T
\] (57)
Theorem 1 is also established.

### III. Experimental results of the neural PID control for an exoskeleton robot

In this paper, we apply our neural PID control in an exoskeleton. 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 theorem 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 (50) 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 \( \alpha = \frac{4\lambda_m(K_p)}{\lambda_m(K_p)} \) such that \( \lambda_M(K_i) < \frac{4\lambda_m(K_p)}{2} \) is satisfied. \( \alpha = 0.08, A \) is chosen as \( A = diag(30), \beta = \frac{17}{2} \). \( K_v = diag(17.58) \). The joint velocities are estimated by the standard filters
\[
\ddot{q}(s) = \frac{bs}{s + a}q(s) = \frac{18s}{s + 30}q(s)
\] (58)
The PID gains are chosen as
\[
K_p = diag[150, 150, 100, 150, 100, 100, 100] \quad K_i = diag[2, 1, 2, 2, 0.2, 0.1, 0.1] \quad K_d = diag[330, 330, 300, 320, 320, 300, 300, 300]
\] (59)
such that the conditions of Theorem 1 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 (21) is Gaussian function
\[
\sigma = \exp\left\{- (m_i - m)^2 / 100\right\}
\]
where \( m_i \) is selected randomly from \( 0 \) to \( 2 \). The weights are updated by (57) with \( K_w = 10 \).

The control results of Joint-1 with neural PID control is shown in Figure 1, 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 (59), the control result is shown in Figure 1, marked "Linear PID-2". Because the steady-state error is so big, the integral gains are increased as
\[
K_i = diag[50, 20, 30, 30, 10, 10, 10, 10]
\] (61)
The control result is shown in Figure 1, marked "Linear PID-1", the transient performance is poor. There still exists regulation error. Further increasing $K_1$ 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

$$K_d = \text{diag} \{970, 900, 970, 970, 800, 800\}$$  \hfill (62)

the control result is shown in Figure 1, marked "Neural PD". The response becomes very slow.

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 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 (57) will influence the learning speed, so a very large gain can cause unstable learning, while a very small gain produce slow learning process.

IV. 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. 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.

V. ACKNOWLEDGMENTS

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

REFERENCES