2011 9th IEEE International Conference on
Control and Automation (ICCA)
Santiago, Chile, December 19-21, 2011

A Systematic Tunning Method of PID Controller for Robot Manipulators

Xiaoou Li, Wen Yu

Abstract— **This paper addresses the iterative tuning method of PID control for the robot manipulator based on the responses of the closed loop system. Several properties of the robot control are used, such as any PD control can stabilize a robot in regulation case, the colsed-loop system of PID control can be approximated by a linear system, and the control torque to the robot manipulator is linearly independent of the robot dynamic. By using these properties, a novel systematic turning method for the PID control is proposed. Simulations and experimental results of an upper limb exoskeleton give validation of this PID tuning method.**

I. INTRODUCTION

The proportinal-integral-derivative (PID) control has simple structure and clear physical meanings for its three gains. The control performances are acceptable in the most of industrial processes. It has been used in more than 90% of various practical control systems [1][2]. Three parameters of PID controller are tuned such that the performances at transient, including rise-time, overshoot, and settling time, steady-state error, are satisfied, meanwhile the closed-loop system is stable and robust against plant modeling uncertainty and disturbances. The study on tuning methods of PID controller mainly focused on linear systems [19]. The tuning methods for PID controllers can be grouped according to their nature and usage:

- Analytical methods: PID parameters are calculated from analytical or algebraic relations between a plant model and an objective [5][6][16].
- Heuristic methods: These are evolved from practical experience in manual tuning [25][4][2], and from artificial intelligence techniques [22][14][11].
- Frequency response methods: frequency characteristics of the controlled process are used to tune the PID controller [20].
- Optimization methods: These can be regarded as a special type of optimal control, where PID parameters are obtained ad hoc using an offline numerical optimization [12].
- Adaptive tuning methods: These are for automated online tuning, using one or a combination of the previous methods based on real-time identification [24].

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

Most robot manipulators employed in industrial operations are controlled by PID algorithms independently at each joint [23]. Compared with the above linear systems-based tuning algorithms, there are some difficulties to design a systematic tuning method for robot PID control

- Robot manipulators are strong nonlinear systems, and the torque of one joint affects the other and vice versa.
- If the gains are tuned heuristically [25], Cohen-Coon method [4] and optimization [12] methods. There are too many gains to tune simultaneously for robot. A 6-degrees-of-freedom robot manipulator has 18 gains to be tuned. When one gain is tuned, it requires to tune the other 17 gains in turn because of dynamics coupling in robot.
- Based on stability analysis, the upper bounds of PD gains and lower bound of derivative gain can be derived. However, these bounds cannot guarantee desired performances.

There are few research regarding PID gains tuning for robot manipulators. PID tuning algorithms cannot be used straight because the responses are nonlinear. The intelligent techniques have been applied for PID gains tuning, for example fuzzy logic [22], neural networks [14] and genetic method [11], but the final controllers are no longer linear PID, they become complete intelligent control systems. Another PID tuning method for robots is impedance control [13], which first uses inverse dynamics to transfer the robot into a linear system. Then some mechanical impedance ideas are applied to tune PID gains. In [5] discrete-time approximation of inverse dynamics was calculated such that PID parameters could be adjusted. Lyapunov approach was used in [7] to adjust PID controller such that it follows linearization control. All above methods need the models of robot manipulators, and their PID controllers do not have clear physical meaning.

In this paper, three important properties of PID control of robot manipulator are applied for PID gains turning.

1) Any PD controller can stabilize a robot in regulation case when its gains are positive
2) The behavior of the colsed-loop system of PID control is simple, and it can be approximated by a linear system
3) The control torque to the robot manipulator is independent of the other robot dynamic.

978-1-4577-1474-0/11/$26.00 ©2011 IEEE

By using these properties, we propose a new systematic tuning method for PID control. The turning steps are as follows

1) a) Stabilize the robot with a PD control
   b) Add a step input to the closed-loop system in (a), and save the step response.
   c) Search a linear time-invariant model, which has a similar step response with (b).
   d) Tune PD/PID gains similar with the linear system in (c)
   e) Refine PID gains in (d) by prior knowledge.

Finally, we apply this method on an upper limb exoskeleton. The experimental results show this PID tuning method is effective for robot manipulator

II. PID TUNING FOR ROBOT MANIPULATORS

The dynamics of robots are derived from Euler-Lagrange equation. It can be written as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + d(q) = u \qquad (1)$$

where $q \in R^n$ represents the link positions. n is joint number, $M(q)$ is the inertia matrix, $C(q,\dot{q}) = \{c_{kj}\} \in R^{n\times n}$ represents centrifugal force, $g(q)$ is vector of gravity torques, $d(q)$ is unknown disturbance. $u \in R^n$ is control input.

Classical linear PID law is

$$u = K_p e + K_i \int_0^t e(\tau)\, d\tau + K_d \dot{e} \qquad (2)$$

where $e = q^d - q$, q^d is desired joint angle, K_p, K_i and K_d are proportional, integral and derivative gains of the PID controller, respectively. This PID control law can be expressed via the following equations

$$\begin{aligned} u &= K_p e + K_d \dot{e} + \xi \\ \dot{\xi} &= K_i e, \quad \xi(0) = \xi_0 \end{aligned} \qquad (3)$$

It is known that in regulation case, any positive gains of the PD controller

$$u = K_p e + K_d \dot{e} \qquad (4)$$

can guarantee stability of the closed-loop system, see Spong and Vidyasagar (1989). PD control does not guarantee the achievement of the position control objective because manipulators dynamics contain the gravitational torques vector, unless gravity compensation is applied. The integrator is the most effective tool to eliminate steady-state error, in this way PD control (4) becomes PID control (2). However, integrator gain has to be increased when the robot is heavy. This causes big overshoot, long settling time, and less robust. An approximation model compensation can decrease integrator gain, see Kelly et al. (2005).

It is known that if the PD control law (4) is applied to each joint, the position tracking error is bounded within a ball whose radius decreases approximately $\frac{1}{\sqrt{\lambda_{\min}(K_p)}}$, see Lewis et al. (2004). Theoretically, PD control is sufficient for robot control. However, in order to decrease steady-state error caused by gravity and friction, derived gain K_p has to be increased. The closed-loop system become slow. Usually, the big settling time does not allow us to increase K_p as we want.

Although adding an integrator can extraordinarily decrease steady-state error, the overshoot of the closed-loop system becomes larger and robustness property deteriorates.

A. Tuning in closed-loop

Since it is danger to send a step command to the joints of the exoskeleton robot. We use closed-loop identification and tuning method. Here we use two properties of the robot dynamics:

- The control torque of the robot is dependent of the other terms;
- PID control is linear.

It is well known the robot (1) is open-loop unstable, and positive gains of a PD controller can guarantee closed-loop stability (bounded) in regulation case , see Spong and Vidyasagar (1989). We first use a PD control (2) with $K_i = 0$, and small K_p and K_d, to stabilize the robot. When the desired position is constant, the closed-loop system is stable,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + d(q) = PD_0$$

Considering gravity compensation, the closed-loop system is

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tilde{g}(q) + d(q) = PD_0 - \hat{g}(q) \quad (5)$$

where $g(q) = \hat{g}(q) + \tilde{g}(q)$, $\hat{g}(q)$ is estimated gravity.

Now we will use a tuning rule to find another PID controller PID_1 for this closed-loop system. If we define the final control torque as

$$u = PID_1 + PD_0 - \hat{g}(q)$$

Obviously, the closed-loop system is

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tilde{g}(q) + d(q) - PD_0 + \hat{g}(q) = u_c \quad (6)$$

The control for the closed-loop system is

$$u_c = PID_1$$

Since the PID control is linear, this idea can be extended to general case,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tilde{g}(q) + d(q) = \sum_{j=1}^{m} PID_j - \hat{g}(q)$$

where m is tuning times, and

$$\sum_{j=1}^{m} PID_j = \sum_{j=1}^{m} K_{p,j} e + \sum_{j=1}^{m} K_{i,j} \int_0^t e(\tau)\, d\tau + \sum_{j=1}^{m} K_{d,j} \dot{e}$$

This means we can start from small PID gains to stabilize the robot first, then tuning the other PID controllers independently. The final PID control is the summarization of all these controllers.

B. Linearization of the colsed-loop system

There are several methods to linearize robot models. If the velocity and gravity are neglected, the terms $C(q,\dot{q})\dot{q}$ and $g(q)$ in the nonlinear dynamics (1) are zero, resulting in a linear model of the form,m see Goldenberg and Bazerghi (1986)

$$M(q)\ddot{q} = u \tag{7}$$

It is an oversimplified model and is impossible for PID tuning, because velocity and gravity are main control issues of robots. Most of robot, the gravity loading is a dominant component of the dynamics.

The velocity dependent term $C(q,\dot{q})\dot{q}$ representing Coriolis-centrifugal forces, can be assumed to be negligible for small joint velocities. This is a rate linearization scheme, see Golla et al. (1981), which results in a linear model of the form

$$A\ddot{q} + Bq = u \tag{8}$$

where $A = M(q)\mid_{q=q_0}$, $B = \frac{\partial g(q)}{\partial q}\mid_{q=q_0}$, q_0 is operating point. But many experiments, see Swarup and Gopal (1993), showed that even at low speeds $C(q,\dot{q})$ should be accounted for.

When the robot model is completely known, Taylor series expansion can be applied, see Li (1989). At the operating point q_0,the nonlinear model (1) can be approximated by

$$A\ddot{q} + D\dot{q} + Bq = u \tag{9}$$

where $A = M(q)\mid_{q=q_0}$, $B = \frac{\partial[g(q)+C(q,\dot{q})]}{\partial q}\mid_{q=q_0}$, $D = \frac{\partial C(q,\dot{q})}{\partial \dot{q}}\mid_{q=q_0}$.

Although the physical and mathematical structure of the complete dynamic robot model is analytically coupled and nonlinear, the observed transient response of robot dynamics appears to resemble the transient response of the linear systems. Consequently, each joint of the robot can be characterized as a single input-single output (SISO) system. In this paper, we use this identification-based linearization method. For each joint, typical linear model is a first order system with transportation delay as

$$G_p = \frac{K_m}{1+T_m s}e^{-\tau_m s} \tag{10}$$

The response is characterized by three parameters, the plant gain K_m, the delay time τ_m, and the time constant T_m. These are found by drawing a tangent to the step response at its point of inflection and noting its intersections with the time axis and the steady state value.

Sometimes the first model cannot describe the complete nonlinear dynamic of robot. A reasonable linear model of robot is Taylor series model as in (9). The model can be written in frequency domain

$$\frac{q_i(s)}{u_i(s)} = \frac{K_m}{T_m^2 s^2 + 2\xi_m T_m s + 1}e^{-\tau_m s} \tag{11}$$

or

$$\frac{q_i(s)}{u_i(s)} = \frac{K_m}{(1+T_{m1}s)(1+T_{m2}s)}e^{-\tau_m s}$$

The responses of this second order model are similar with mechanical motions. If there exists a big overshoot, a negative zero is added in (11)

$$\frac{q_i(s)}{u_i(s)} = \frac{K_m(1+T_{m3}s)}{(1+T_{m1}s)(1+T_{m2}s)}e^{-\tau_m s} \tag{12}$$

The normal input signals for PID tuning are step and repeat inputs.

C. PD/PID tuning

The linear PID law in time domain (2) can be transformed into frequency domain

$$U(s) = K_c\left(1 + \frac{1}{T_i s} + T_d s\right)E(s) = G_c(s)E(s)$$

where $K_c = K_p$ is proportional gain, $T_i = \frac{K_c}{K_i}$ is integral time constant and $T_d = \frac{K_d}{K_c}$ is derivative time constant.

Because the robot can be approximated by a linear system. Some tuning rules for linear systems can be applied for the colsed-loop system tuning. We first give PD tuning rules. When each joint can be approximated by a first-order system,

$$G_p = \frac{K_m}{1+T_m s}e^{-\tau_m s}$$

The PD gains are tuned as in Table 1, here Model 1 is from Huang et al. (2005), Model 2 is from Chien and Fruehauf (1990).

Table 1. PD turning for the first-order model

	K_c	T_i	T_d
Ziegler-Nichols tuning	$a\frac{T_m}{K_m\tau_m}$	$0.5\tau_m$	
Cohen-Coon method	$\frac{T_m}{K_m\tau_m}\left(\frac{4}{3}+\frac{\tau_m}{4T_m}\right)$	$\frac{4T_m\tau_m}{11T_m+2\tau_m}$	
Our Method	$\frac{T_{m2}}{K_m}$		T_{m1}

Here K_m, T_m and τ_m are obtained from Figure 1.

In Table 1 we list Ziegler-Nichols and Cohen-Coon methods, they are PI controllers. From the best of our knowledge, PD turning rules are still not published and applied. If each joint is approximated by a second-order system,

$$\frac{q_i(s)}{u_i(s)} = \frac{K_m}{T_m^2 s^2 + 2\xi_m T_m s + 1}$$

The PD gains are tuned as in Table 2.

Table 2. PD tuning for the second-order model

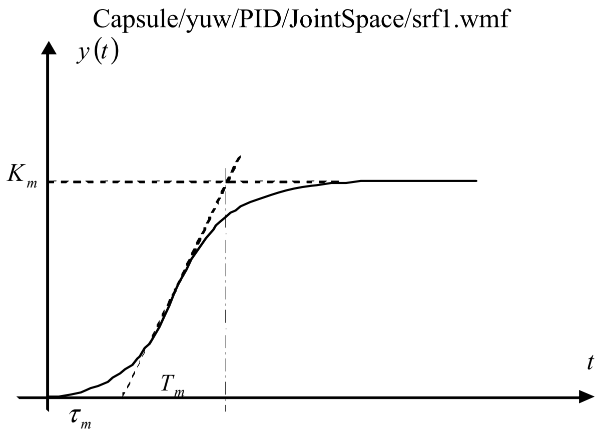

Fig. 1. Step response of a linear system

	K_c	T_i	T_d
Method 1	$\frac{5T_{m1}\xi_m}{K_m T_{m3}}$	$\frac{T_{m1}+0.1\xi_m}{0.8T_{m1}\xi_m}$	
Method 2	$\frac{T_{m2}}{K_m}$	T_{m1}	
Our Method	$\frac{T_{m2}}{K_m}$		T_{m1}

When PD control cannot provide good performances, PID control should be used. The PID gains for the first-order model is decided by Table 3.

Table 3. PID tuning for the first-order model

	K_c	T_i	T_d
Ziegler-Nichols tuning	$a\frac{T_m}{K_m\tau_m},$	$2\tau_m$	$0.5\tau_m$
Cohen-Coon method	$\frac{T_m}{K_m\tau_m}\frac{\tau_m}{4T_m}$	$\frac{\tau_m(32T_m+6\tau_m)}{13T_m+8\tau_m}$	$\frac{4T_m\tau_m}{11T_m+2\tau_m}$
Our Method	$\frac{T_{m2}}{K_m}$	T_{m2}	T_{m1}

The PID gains for the second-order model is decided by Table 4.

Table 4. PID tuning for the second-order model

	K_c	T_i	T_d
Method 1	$\frac{5T_{m1}\xi_m}{K_m T_{m3}}$	$2T_{m1}\xi_m$	$\frac{T_{m1}+0.1\xi_m}{0.8T_{m1}\xi_m}$
Method 2	$\frac{T_{m2}}{K_m}$	T_{m2}	T_{m1}
Our Method	$\frac{20\xi_m T_m}{K_m}$	$15\xi_m T_m$	$\frac{T_m^2}{10}$

If the above four tables cannot give us good performances, we use Table 5 to refine PID gains as PID_2.

Table 5. Effects of PID gains

	Rise	Overshoot	Settling	Steady Error	Stability
P↑	Decrease	Increase	Small Increase	Decrease	Degrade
I↑	Small Decrease	Increase	Increase	Large Decrease	Degrade
D↑	Small Decrease	Decrease	Decrease	Minor Decrease	Improve

The procedure of PD/PID tuning for robot control is described as follows

Step 1 Gravity modeling $\hat{g}(q)$: the objective of this step is to decrease integrator gain, such that overshoot is small

Step 2 PD control PD_0: use small PD gain to generate a stable closed-loop system.

Step 3 Obtain the step responses of the closed-loop system for each joint. Now the robot has been compensated by gravity model, i.e.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tilde{g}(q) = PD_0 - \hat{g}(q)$$

Step 4 Use the first-order or the second-order linear models to approximate the step responses of the closed-loop systems. PID gains are obtained by Table 1-Table 4, it is PID_1

Step 5 Refine PID gains with Table , it is PID_2

Step 6 The final control for the robot is

$$U = PD_0 + PID_1 + PID_2 - \hat{g}(q)$$

III. Application to an exoskeleton

Exoskeletons are wearable robots, which are worn by the human operators as orthotic devices. The exoskeleton links, joints and work space correspond to those of the human body. The system may be used as a human input device for tele operation, human-amplifier, and physical therapy modality as part of the rehabilitation process [10]. Although great progress has been made in a century-long effort to design and implement robotic exoskeletons, many design challenges continue to limit the performance of the system. One of the limiting factors is the lack of simple and effective control systems for the exoskeleton. The PID/PD control is the simplest scheme that can be used to control robot manipulators. The exoskeletons are usually heavy, it is not easy to obtain an ideal PID for an exoskeleton robot. The 7-DOF upper limb exoskeleton shown in Figure 2 is composed of a 3-DOF shoulder (J1-J3), a 1-DOF elbow (J4) and a 3-DOF wrist (J5-J7). J1-J3 are responsible for shoulder flexion-extension, abductionadduction and internal–external rotation, J4 create elbow flexion-extension, J5-J7 are responsible for wrist flexion–extension, pronation-supination and radial–ulnar deviation.

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. The real-time control program operated in Windows XP 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 8.

Table 8. Parameters of the exoskeleton

Capsule/yuw/PID/Exoskeleton/exosf1.wmf

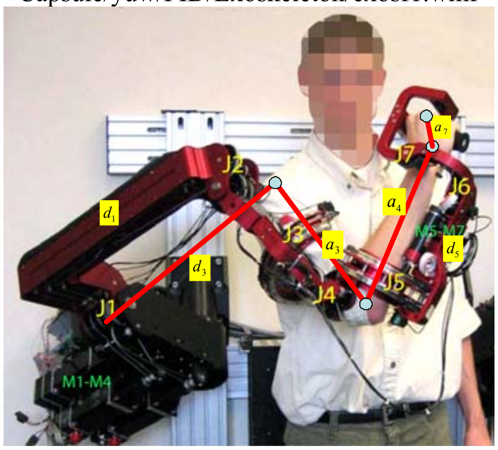

Fig. 2. The UCSC 7-DOF exoskeleton robot.

Joint	Mass (kg)	Center (m)	Length (m)
1	3.4	.3	.7
2	1.7	.05	.1
3	.7	.1	0.2
4	1.2	.02	.05
5	1.8	.02	.05
6	.2	.04	.1
7	.5	.02	.05

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 first use the following PD control to stabilize the robot

$$\begin{array}{l} K_p = diag\left[150, 150, 100, 150, 100, 100, 100\right] \\ K_d = diag\left[330, 330, 300, 320, 320, 300, 300\right] \end{array} \quad (13)$$

The joint velocities are estimated by the standard filters

$$\widetilde{\dot{q}}\left(s\right) = \frac{bs}{s+a} q\left(s\right) = \frac{18s}{s+30} q\left(s\right)$$

The PD regulation of the first four joints are shown in the sold lines of Figure 3. Then we use open-loop step responses of linear systems to approximate the closed-loop responses of the robot.

$$\begin{array}{c} G_1 = \frac{0.93}{60s^2+9s+1} \\ G_2 = \frac{1}{20s^2+3s+1} \\ G_3 = \frac{0.9}{5.5s^2+4s+1} \\ G_4 = \frac{0.85}{30s^2+8s+1} \end{array} \quad (14)$$

The step responses of the following four linear system are shown in the dash lines of Figure 3.

Here the main weight of the exoskeleton is in the first four joints. The potential energy is

$$\begin{array}{c} U = m_1 g l_1 s_1 + m_2 g \left(l_1 s_1 + c_2 l_2 s_1\right) + m_3 g a_3 s_1 s_2 \\ + m_4 g \left[a_4 c_4 \left(c_1 s_3 + c_2 c_3 s_1\right) + a_4 s_4 \left(c_1 c_3 - c_2 s_1 s_3\right) + a_3 s_1 s_2\right] \end{array}$$

Capsule/yuw/PID/Exoskeleton/autof4.wmf

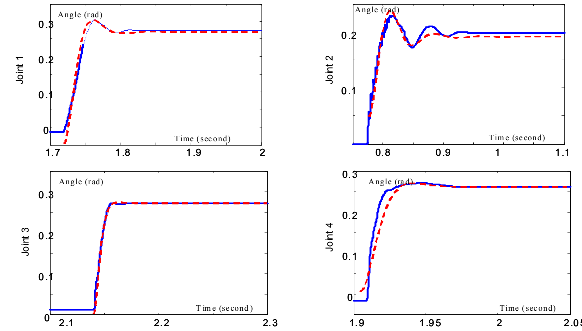

Fig. 3. PD control of the exoskeleton and step responses of linear models

The gravity compensation in (5) is calculated by $\hat{g}\left(q\right) = \frac{\partial}{\partial q} U\left(q\right).$

We will design a PID tuning rule for these linear systems and apply the tuned PID controllers to the robot. In order to tuning PID gains for the linear systems (14), we rewrite the PID (2) as

$$PID_t = K_c \left(\tilde{q} + \frac{1}{T_i} \int_0^t \tilde{q}\left(\tau\right) d\tau + T_d \dot{\tilde{q}}\right)$$

where $K_c = K_p$ is proportional gain, $T_i = \frac{K_c}{K_i}$ is integral time constant and $T_d = \frac{K_d}{K_c}$ is derivative time constant.

We use the following tuning rule

$$K_c = \frac{20\xi_m T_m}{K_m}, \quad T_i = 15\xi_m T_m, \quad T_d = \frac{T_m^2}{10} \quad (15)$$

to tune the PID parameters. This rule is similar with Huang et al. (2005), and Chien and Fruehauf (1990), in their case $K_c = \frac{5T_{m1}\xi_m}{K_m T_{m3}}$, $T_i = 2T_{m1}\xi_m$, $T_d = \frac{T_{m1}+0.1\xi_m}{0.8T_{m1}\xi_m}$. It is different with the other two famous rules, Ziegler-Nichols and Cohen-Coon methods, where $K_c = a\frac{T_m}{K_m \tau_m}$, $T_i = 2\tau_m$, $T_d = 0.5\tau_m$ or $K_c = \frac{T_m}{K_m \tau_m}\left(\frac{4}{3} + \frac{\tau_m}{4T_m}\right)$, $T_i = \frac{\tau_m(32T_m+6\tau_m)}{13T_m+8\tau_m}$, $T_d = \frac{4T_m \tau_m}{11T_m+2\tau_m}$. Because their rules are suitable for the process control, our rule is for mechanical systems.

By the rule (15), the PID_1 gains are

$$\begin{array}{l} K_{p1} = 90, K_{i1} = 1, K_{d1} = 540, \\ K_{p2} = 30, K_{i2} = 2, K_{d2} = 60 \\ K_{p3} = 40, K_{i3} = 20, K_{d3} = 20, \\ K_{p4} = 90, K_{i4} = 1.5, K_{d4} = 270 \end{array} \quad (16)$$

We apply these PID controllers PID_1 to the robot, the new closed-loop system

$$M\left(q\right)\ddot{q} + C\left(q,\dot{q}\right)\dot{q} + \tilde{g}\left(q\right) - PID_0 + \hat{g}\left(q\right) = PID_1$$

The control torque becomes

$$u = PID_1 + PD_0 - \hat{g}\left(q\right) \quad (17)$$

Capsule/yuw/PID/Exoskeleton/autof5.wmf

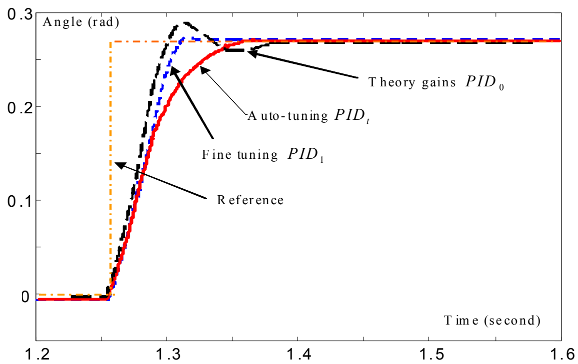

Fig. 4. PID tuning for joint 1 (J1)

Capsule/yuw/PID/Exoskeleton/autof6.wmf

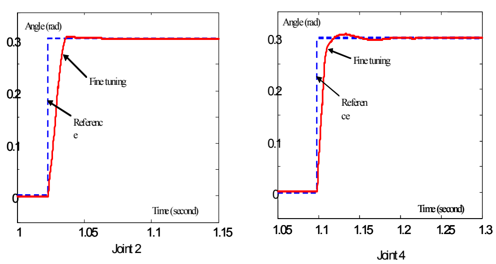

Fig. 5. Final PID control.

Since we use linear PID control, the gains of PID_t and PID_0 can be added together. The control results of Joint 1 are shown in Figure 4.

After this refine turning, the final PID control is PID_2, their gains are

$$\begin{aligned} K_i &= diag\,[5,4,5,6,3,4,2] \\ K_p &= diag\,[320,280,210,250,210,210,220] \\ K_d &= diag\,[410,400,420,430,410,410,410] \end{aligned}$$

The final control is

$$U = PD_0 + PID_1 + PID_2 - \hat{g}(q)$$

The control results are shown in Figure 4 and Figure 5.

IV. Conclusions

In this paper, a new systematic tuning method for PID control is proposed. This method can be applied to any robot manipulator. By using several properties of robot manipulators, the tuning process becomes simple and is easily applied in real applications. Some concepts for PID tuning are novel, such as step responses for the closed-loop systems under any PD control, and the joint torque is separated into several independent PIDs. We finally apply this method on an upper limb exoskeleton, real experiment results give validation of our PID tuning method.

References

[1] K.H.Ang, G.Chong, Y.Li, PID Control System Analysis, Design, and Technology, *IEEE Trans. Control System Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[2] K.J. Astrom, T. Hagglund, Revisiting the Ziegler–Nichols step response method for PID control, *Journal of Process Control,* 14, 635–650, 2004

[3] P. Cominos, N. Munro, PID controllers: recent tuning methods and design to specifications, *IEE Proceedings—-Control Theory and Applications,* 149 (1), 46–53, 2002

[4] G.H. Cohen, G.A. Coon, Theoretical consideration of retarded control, *Trans. ASME,* 75, 827–834, 1953

[5] P.H.Chang, and J.H.Jung, A systematic method for gain selection of robust PID control for nonlinear plants of second-order controller canonical form, *IEEE Trans. Control System Technology*, vol. 17, no. 2, pp. 473–483, 2009.

[6] I.-L.Chien, P.S.Fruehauf, Consider IMC tuning to improve controller performance, *Chemical Engineering Progress*, 33-41, 1990

[7] W.-D.Chang, R.C.Hwang, J.G.Hsieh, A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach, *Journal of Process Control,* 12, 233–242, 2002

[8] A.A.Goldenberg, A.Bazerghi, Synthesis of robot control for assembly processes, *Mech. Machine Theory,* 21(1), 43-62, 1986

[9] D.F.Golla, S.C.Garg, P.C.Hughes, Linear state-feedback control of manipulators, *Mech.Machine Theory*, 16, 93-103, 1981.

[10] H.Herr, Exoskeletons and orthoses: classification, design challenges and future directions, *Journal of NeuroEngineering and Rehabilitation*, 6(21) 2009.

[11] J-G.Juang, M-T.Huang, W-K.Liu, PID control using presearched genetic algorithms for a MIMO system, *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no.5, 716-727, 2008 .

[12] B. Kristiansson, B. Lennartsson, Robust and optimal tuning of PI and PID controllers, *IEE Proceedings—-Control Theory and Applications,* 149 (1), 17–25, 2002

[13] N.Hogan, Impedance control: An approach to manipulation, PartsI–III, *ASME J. Dynam. Syst., Meas., Contr.*, vol. 107, pp. 1–24, 1985.

[14] 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

[15] F.L.Lewis, D.M.Dawson, C.T.Abdallah, *Robot Manipulator Control: Theory and Practice*, Marcel Dekker, Inc, New York, NY 10016, 2004

[16] H.-P.Huang, J.-C.Jeng, K.-Y.Luo, Auto-tune system using single-run relay feedback test and model-based controller design, *Journal of Process Control*, 15, 713-727, 2005

[17] R.Kelly, V.Santibáñez, L.Perez, *Control of Robot Manipulators in Joint Space,* Springer-Verlag London, 2005.

[18] C.J.Li, An efficient method for linearization of dynamic models of robot manipulators, *IEEE Transactions on Robotics and Automation*, Volume: 5 Issue: 4 , 397 - 408, 1989

[19] A.O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, Imperial College Press, London, 2006

[20] F.G.Shinskey, *Process Control Systems - Application, Design and Tuning*, McGraw-Hill Inc., New York, 1996

[21] A.Swarup, M.Gopal, Comparative study on linearized robot models, *Journal of Intelligent and Robotic Systems*, Volume 7, Number 3, 287-297, 1993

[22] Y. L. Sun and M. J. Er, Hybrid fuzzy control of robotics systems, *IEEE Trans. Fuzzy Systems*, vol. 12, no. 6, pp. 755–765, Dec. 2004.

[23] M.W.Spong and M.Vidyasagar, *Robot Dynamics and Control,* John Wiley & Sons Inc., Canada, 1989.

[24] Q.-G.Wang, , Y.Zhang, X.Guo, Robust closed-loop identification with application to auto-tuning, *Journal of Process Control*, 11, pp. 519-530, 2001

[25] J. G. Ziegler and N. B. Nichols, Optimum settings for automatic controllers, *Trans. ASME*, vol. 64, pp. 759–768, 1942