A precise control of AC servo motor using neural network PID controller

Geum-Bae Cho and Pyoung-Ho Kim

A new control technique based on a neural network, is proposed here for control of AC servo motors. The PID control is widely used in servo systems as it has simple structure, safety and reliability. However, it has certain problems in a complex system, resulting in imperfect action in the presence of uncertain parameters. To solve these problems, a new hybrid control algorithm of the PID controller is proposed, which could prove the adequacy of the proposed control algorithm through simulation and experiments after driving the AC servo motor system using neural network PID controller.

Advanced developments in microcomputers based on microprocessors are making them cheaper. AC controllers can be as cheap as DC controllers in terms of control. An AC controller has easy maintainability. The use of DC servo motors for the purpose of high-speed and high-torque applications is limited due to commutation problem. The brush in the DC servo motor is subject to wear; hence installation of DC servo motor is difficult in places where checking or swapping is needed. As an AC servo motor does not have any brush, it is highly reliable. Besides, the AC servo motor is usually used in the analogue form due to rapid reaction response and precise control is achieved because of digitalization[1]. In the industry, the PI or PID controller is widely used by means of servo system control. These controllers enable excellent ability if a simple control algorithm be implemented. However, these have low reliability because these control results are sensitive to change in system parameters and do not react rapidly to parameter changes. To solve these problems, the neural network controller that adjusts itself to control circumstances is studied[2,3].

Here we describe implementation of a PID controller for an AC servo motor, based on TMS320C31. One could obtain superior and highly precise control response than a conventional PID controller, which could resolve the disadvantages of parameter changes in an inferior environment[4]. Contrary to the processing method of a digital computer, which carries out a single computation at a time in an ALU because of parallel and distributed processes, information stored in neurons is processed in parallel and does not affect the overall output in case several process elements are perturbed or if immeasurable disturbances occur[5]. Therefore, online-type neural network PID controller, using past data as well as current inputs and outputs in order to control the AC servo motor, can be implemented easily; it is considered to be robust to immeasurable disturbances of load[6]. With results from both experiments and digital simulation, the proposed method could be verified and hence utilized for high-performance servo-control.

Design of AC servo motor

Neural network controller

The neural network makes process speed faster because of hardware implementation. Also it can be implemented when the structure of the circuit net is large[7]. It is closer to a nonlinear control due to accessibility of actual problem, such as function mapping in nonlinear theory. The neural network enables parallel and distributed processes because of its parallel structure[8], hence it can be used for control in real time. It makes synthesis of sensor data easy to interface with different data, because it is robust to noise compared to a conventional control and utilizes both qualitative and quantitative data simultaneously. Since it has several inputs and outputs, it is adequate for multi-input and multi-output systems, and it can improve the control through learning. Moreover it is not required to model the plant and circumstance because of learning ability. Due to the above advantages and characteristics, a robot control having the neural network controller can solve problems of a conventional robot control[9].

Structure of PID controller using neural network control

Figure 1 shows the block diagram of PID controller using neural network.

The proposed controller can tune the conventional PID controller using indirect neural network, which can be controlled by only inputs and outputs even in Jacobian of unknown control objects. Indirect neural network is composed of an emulator supervising control object and the neural network controller controlling

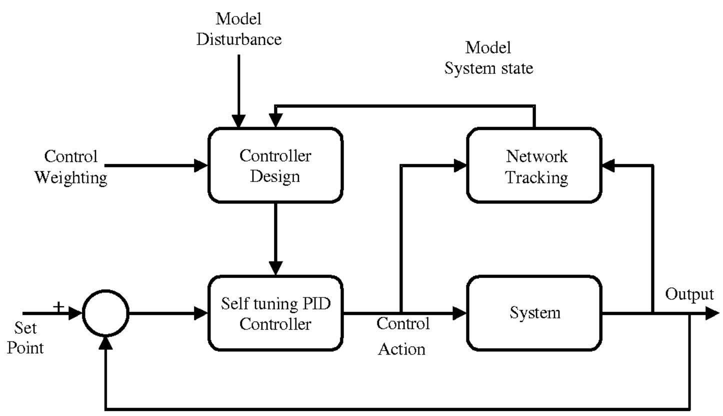

Figure 1. PID controller of neural network.

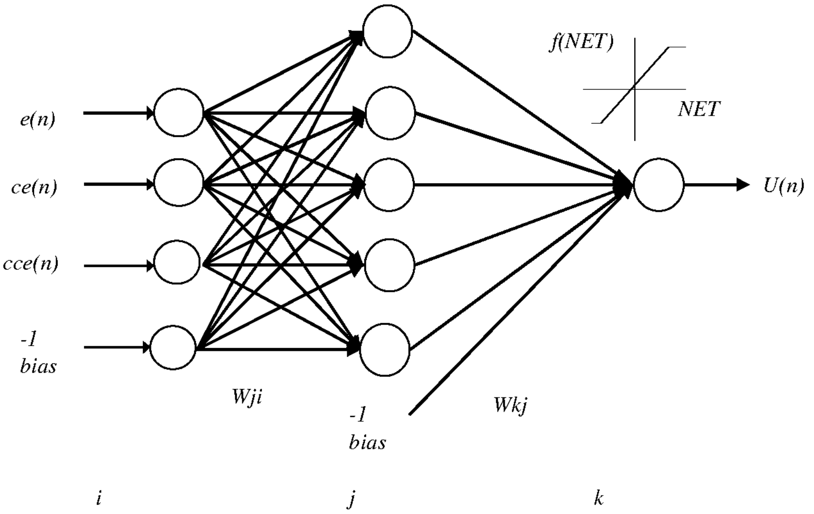

Figure 2. Internal structure of proposed neuro controller.

Table 1. Ratings of 11 kW servo motor

Rated voltage	220 V	Rotor resistance R_r	5 Ù
Rated current	35 A	Stator resistance R_S	4.5 Ù
Rated speed	3000 rpm	Rotor inductance L_r	0.243 H
Stator inductance L_S	0.244 H	Moment of inertia J_m	0.082 kg m^2
Mutual inductance L_m	0.239 H		

control objects. Here, all variables and structure of emulator and controller are considered to be the same in order to easily design the controller. The disadvantage that the parameters of conventional PID controller must be adjusted to the system is avoided, as the controller adjusts to the system through learning of neural network.

Figure 2 shows the internal structure of the proposed neural controller, which explains the number of neurons of each stage and threshold function of activation output of neuron. Arguments to neural controller are composed of time-delay component of error between set-up value and actual output value, $e(n)$, error variation according to time change, $ce(n)$, amount of error variation $cce(n)$. $e(n)$ is the time-delay component of $e(n+1)$, which is expressed as in eq. (1) and $ce(n)$ and $cce(n)$ are given by eqs (2) and (3) respectively.

$$e(n) = TD[e(n+1)] = TD[R(n+1) - Y(n+1)], \quad (1)$$

$$ce(n) = e(n) - e(n+1), \quad (2)$$

$$cce(n) = ce(n) - ce(n+1). \quad (3)$$

Stable neuron output is obtained by inputting fourth argument to middle-stage neuron as bias.

The middle stage of neural network controller is a single stage and the number of middle-stage neurons is five. Bias is applied as in the input stage and the threshold function of each neuron is the linear function whose slope is unity. The number of output-stage neurons, $u(n)$, is one. The threshold function of the output-stage neuron is linear as in the middle stage, but its slope can be arbitrarily adjusted to control the output gain in order to obtain optimum output during learning.

Simulation

A digital simulation is made in order to consider the feasibility of AC servo motor control algorithm using PID controller and neural network algorithm. The power rating of the servo motor for simulation is 11 kW and its rated velocity is 3000 rpm. The detailed parameters shown in Table 1 are used with proportional and integral velocity controller, whose gain is self-tuned by reverse neural network. Velocity-response characteristics are compared in servo motor according to self-tuned proportional and integral controller, and proportional and integral controller.

Assuming that relative velocity of axis to sinusoidal AC flow in the motor is zero, the servo motor current i_d is given by eq. (4)

$$i_a = -\frac{R_a}{L_a} i_a - \frac{\omega_m \Phi}{L_a} + \frac{V_a}{L_a}. \quad (4)$$

Figure 3 shows the block diagram of eq. (4) using integrator.

When the sampling interval of control input is maintained constant, the transfer function is expressed by eq. (5).

$$G(z) = Z\left[\frac{1-e^{-st}}{s} G(s)\right]^{-}. \quad (5)$$

Since $G(s)$ is the transfer function, it can be expanded in partial fractions and Z-transformed, and the result is shown in Figure 4.

In Figure 4, $G(s) = \omega_m(s)/I_0(s)$ is given by

$$G(s) = \frac{\dfrac{1.5K_e K_a}{JL_a}}{S^2 + \dfrac{R + K_{vK}}{L_a} + \dfrac{1.5K + K_v}{JL_a}}. \quad (6)$$

The algorithm is implemented using Borland C++. The roots of the fifth-order nonlinear simultaneous equation are obtained by Runge–Kutta method. The period of current control and velocity control is set to 25 and 250 ms, respectively. The learning rate, inertia constant and each initial weight are near the optimum value by trial and error. The initial connection strength is obtained off-line, and later obtained on-line using reverse algorithm from NNE learning. Figure 5 *a* shows velocity-response waveform using PID controller of neural network as the velocity is changed to 500, 1500 and 0 rpm. The servo motor is well followed by self-tuned proportional integrator. Figure 5 *b* shows velocity-response waveform using PID controller under similar condition as in Figure 5 *a*. The PI controller gives good performance when the system parameters are constant. However, it is difficult to tune in the presence of immeasurable disturbances in the system parameters.

Figure 6 *a* shows results for slow control by PID velocity controller. Figure 6 *b* shows the change in performance for a 15% change in the stator and rotor inductance. Figure 7 *a* shows results for slow control by PID controller of neural network. Figure 7 *b* shows steady-state oscillation when inductance of stator and rotor

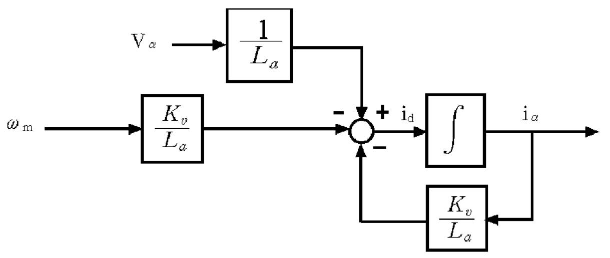

Figure 3. Block diagram of *d*-axis current control.

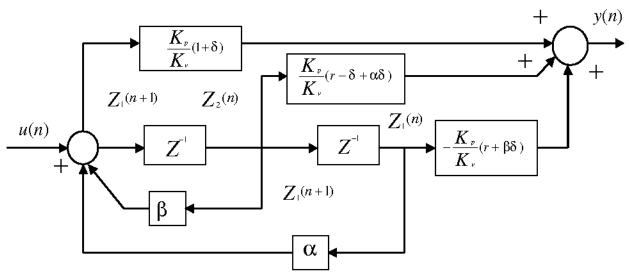

Figure 4. Discrete model of AC servo motor.

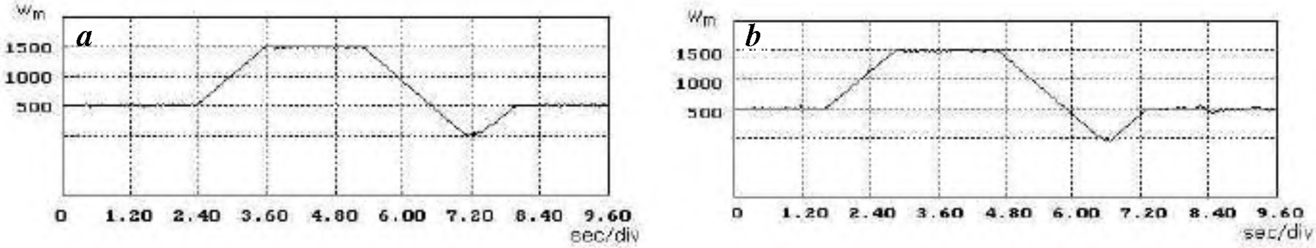

Figure 5. Speed characteristics with (***a***) PID controller using neural network and (***b***) PID controller.

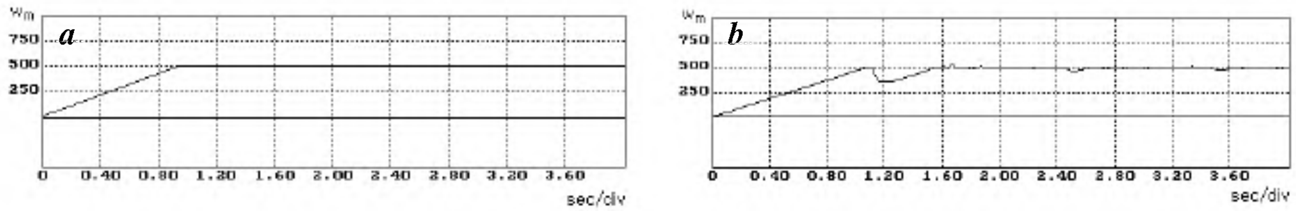

Figure 6. Low-speed characteristics with (***a***) PID controller and (***b***) as in (***a***) with 15% variation in inductance of stator and rotor.

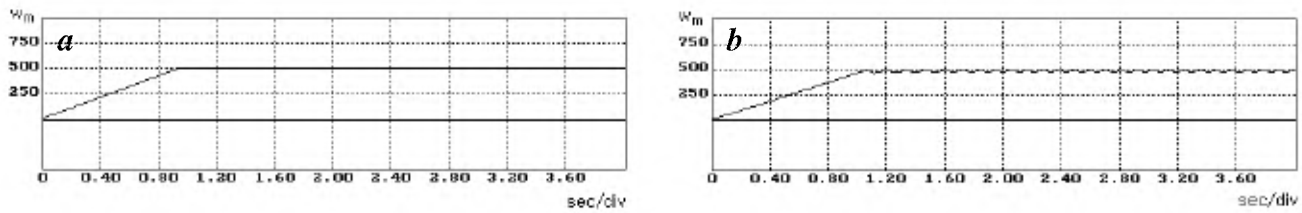

Figure 7. Speed characteristics with (***a***) self-tuning PID controller using neural network and (***b***) as in (***a***) with 15% variation in inductance of stator and rotor.

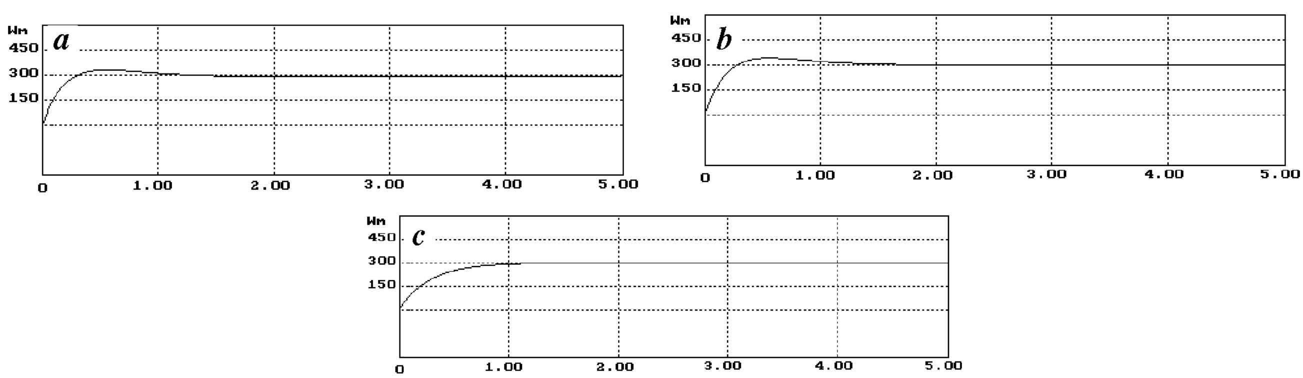

Figure 8. Starting characteristics by (***a***) PI controller, (***b***) PID controller and (***c***) PID control by neural network.

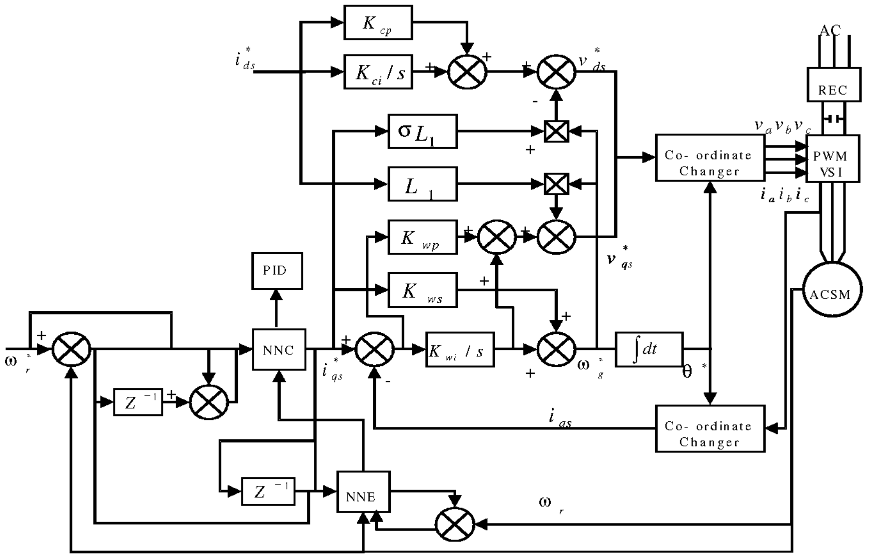

Figure 9. Block diagram of servo system.

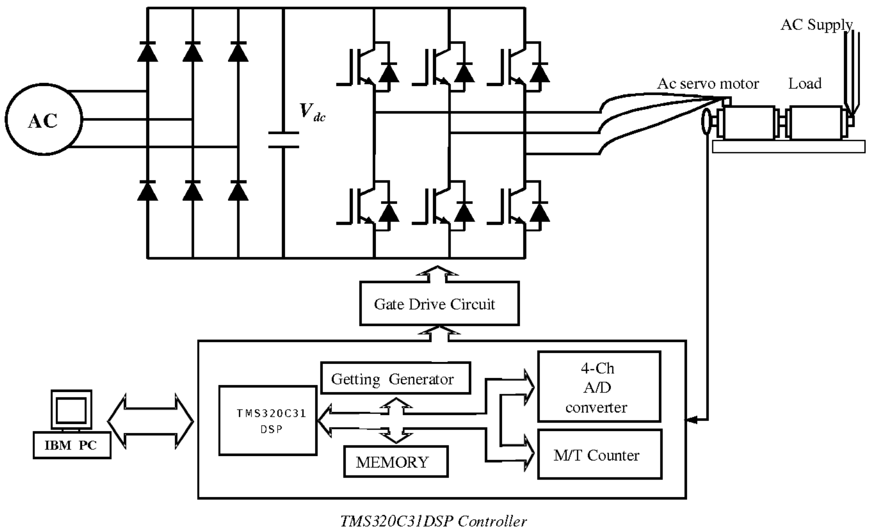

Figure 10. Configuration of AC servo system.

is changed by 15%. From Figure 7 *a* and *b*, it is clear that the PID controller using neural network is more robust to change in environment than a PI controller. Delay in simulation results occurs when inertia coefficient, friction coefficient and integral coefficient *I* are not proper. Figure 8 *a–c* compares the systems implemented by PI, PID controller and PID controller with neural network. It can be seen that the rising time of the general controller is faster than the PID controller with neural network, wherever overshoot of the former occurs. Although gain tuning of the controller can eliminate overshoot, it is difficult to tune automatically the gain coefficients on-line according to the change in work process and surrounding environment. The PID controller using neural network has slow rising time, but can approach steady state without overshoot. Since the torque is not constant, if an overshoot occurs in a servo system used in an industrial robot, a PID control using neural network is better than conventional P, PI, PID controls.

Experiments

Experimental equipment

Figure 9 shows the block diagram of a servo system using PID tuning of indirect neural network. In the digital servo structure, the instruction given by the program is processed by the controller to position each axis. This instruction is transferred to the high-performance microprocessor. The microprocessor batch-processes the control of position, velocity and current based on the information and outputs the PWM control signal. This signal inputs the servo amplifier equipped separately, which amplifies the PWM control signal and supplies power to the AC servo motor. The driving current of the motor passes the servo amplifier and its position and velocity are fed back by the controller through the pulse coder, and so on. As mentioned above, after the microprocessor

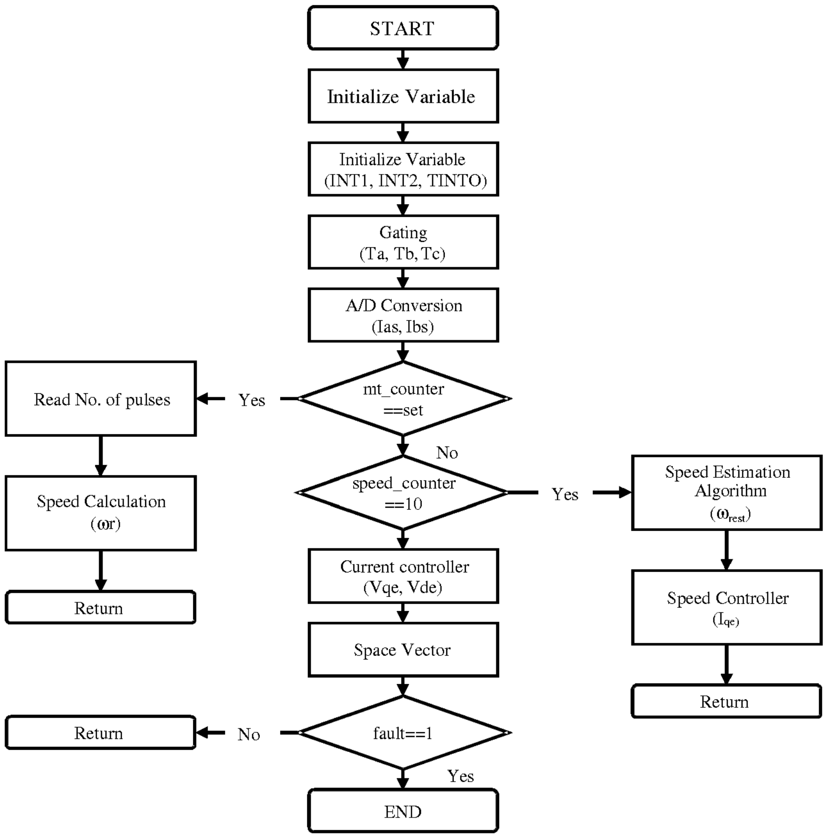

Figure 11. Flow chart of system.

detects the position and velocity it selects and performs the operation.

Figure 10 shows the system structure. The AC servo motor, which is the control object, is a synchronous motor and the load is an induction motor. In order to supply power to the control object motor, the inverter and controller are composed and the load is supplied by the voltage transposed in the three-phase transformer and is let contrary to rotate motor. The power inverter has Insulated Gate Bipolar Transistor (IGBT) in order to reduce the current ripple of Pulse Width Modulation (PWM). The control algorithm of velocity and position, neural network control algorithm and PID control algorithm are processed digitally by the microprocessor. DSP TMS320C31 which operates at 33 MHz and is capable of 32 bit floating-point, is used.

Control software

The software has an initializing program and interrupt routine carrying out the control algorithm in the constant period. Figure 11 is the overall flowchart of control software. The initializing program decides the initial global variables and the controller gain, and initializes the elements which belong to the control board. Because the routine for control keeps the period of sampling constant time, it is composed of the interrupt routine. The interrupt can be divided into the timer interrupt (TINTO) for current control, the external interrupt (INTO) for the M/T and external interrupt (INT1) for velocity control.

The sampling period for the current control is set to 90 μs because of the DSP speed. Sampling for velocity control occurs ten times that of the current control loop. The current control interrupt makes the gating for the inverter and executes the modulation routine of spatial voltage vector. The interrupt to calculate the actual velocity is M/T-type and the sampling period of velocity measurement is processed after Ä*T* later than sampling period of current sampling. The velocity control interrupt estimates velocity using intelligence control algorithm and then executes the indirect vector control after calculating the current instruction value in the velocity controller. The slope of learning rate and output-stage neuron is set to 0.059 and 0.0039 respectively, using experimental data.

Experimental results

Figure 12 shows results of the experiment to change in load using PID control with neural network. Figure 12 *a* shows speed response characteristics of increasing and decreasing speed for change in reference speed from 500 to 1500 rpm. It responds to 1500 rpm after 10 s and decreases to 500 rpm after 7 s. Response speed of the motor converges within 100 ms when reference speed is 500 rpm. It converges to the steady state without overshooting within 200 ms when increased to 1500 rpm after 0 s, and within 200 ms, when decreased to 500 rpm after 500 ms. The response characteristics of increasing and decreasing speed are nearly the same, which is similar to the simulation waveform. Figure 12 *b* shows speed response results for change in reference speed from 1500 to 0 rpm. Figure 12 *c* shows speed response characteristics of 85% rating load and 6% changing load injection at

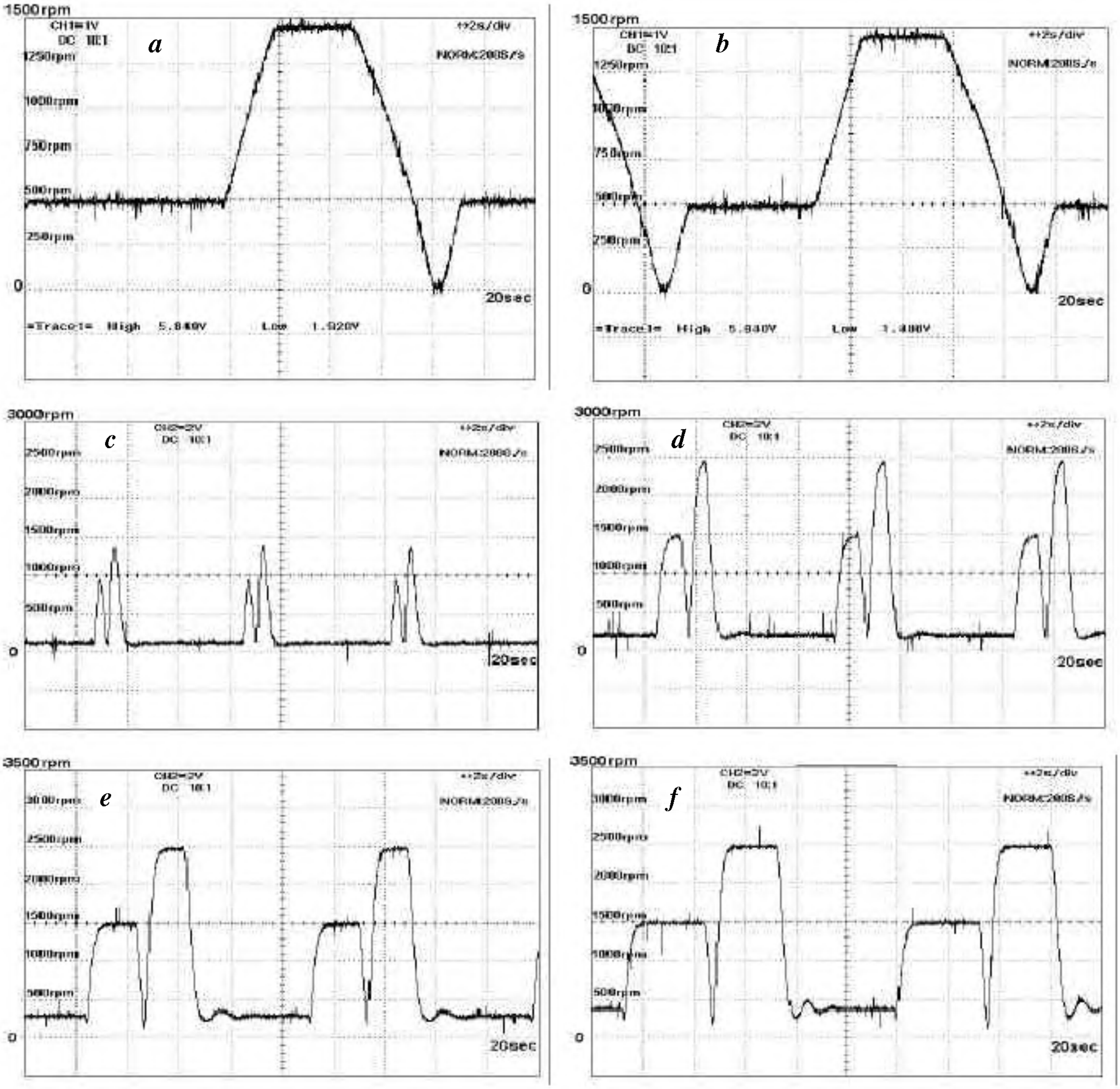

Figure 12. Speed response characteristics for change in reference speed from (***a***) 500 to 1500 rpm and (***b***) 1500 to 0 rpm; (***c***) Speed response characteristics of 85% rating load and 6% changing load injection at 100 rpm. Speed response characteristics for change in reference speed from (***d***) 0 to 500 rpm with 140% load and 500 rpm with 6% load; (***e***) 0 to 1000 rpm with 141% load and 1000 rpm with 17% load; and (***f***) 0 to 1500 rpm with 141% and 1500 rpm with 23% load.

100 rpm. Figure 12*d* shows speed response characteristics for change in reference speed from 0 to 500 rpm with 140% load and 500 rpm with 6% load. Figure 12*e* shows speed response characteristics for change in reference speed from 0 to 1000 rpm, with 141% load and 1000 rpm with 17% load. Figure 12*f* shows speed response characteristics for change in reference speed from 0 to 1500 rpm with 141% and 1500 rpm with 23% load. This system has a little overshoot and delay occurs in changing speed, but it approaches near to instruction value.

Conclusions

In this note, the PID controller using neural network is implemented by the proposed algorithm. The speed characteristics to reference speed are studied by PI control and PID control with neural network. It is concluded that the PID control using neural network is superior. The AC servo motor controller without information on model structure is implemented by TMS320C31 and compared with conventional PID controller. The optimum control value according to error detection value can be obtained by PID controller using neural network differently than from conventional control method. Since the

PID controller using neural network has learning ability of nonlinear function without mathematical modelling and is robust to changeable parameters, it can identify complex systems such as nonlinear systems by learning. If the neural network follower by conventional reverse learning method is applied to the proposed algorithm, the speed control characteristics are excellent. However, the speed estimate is not stable, and is improved when the back-propagation neural network and load observer applied machine constant are added. Each coefficient of machine tool is applied to the servo system, and hence the characteristics of PID controller using neural network can be improved.

1. Ruangsheng, W., Pingyang, W., Song, Y. H. and Johns, A. T., Coordinated system of fuzzy logic control and neural network. ICEE'96, 1996, vol. 2, pp. 1178–1183.
2. Qingding, G., Limei, W. and Ruifu, L., Completely digital AC servo system using neural network controller. ICEPE'95, 1995, pp. 51–55.
3. Ha, Q. P. and Negnevistsky, M., A neural network-based controller for servo drives. IEEE'96, 1996, pp. 904–909.
4. Fu, K. S., Gonzalez, R. C. and Lee, C. S. G., *Robotics*, McGraw-Hill, 1988, pp. 149–265.
5. Galvan, E., Barrero, F., Aguirre, M. A., Torralba, A. and Franquelo, L. G., A robust speed control of AC motor drives based on fuzzy reasoning, IAS'93, Part III, 1993, pp. 2055–2058.
6. Kim, H-W., Choi, J-W. and Sul, S-K., Accurate position control for servo motor using novel speed estimator, IECON'95, 1995, vol. 1, pp. 627–632.
7. Patel, D. M., High speed floating-point robot controller, CAMC'87, 1987, pp. 13–22.
8. Takakura, S., Murakami, T. and Ohnishi, K., An approach to collision detection and recovery motion in industrial robot, IEEE, 1989, pp. 421–426.
9. Ohm, D. Y. and Mazurkiewicz, J., Control of AC motors for servo applications, PCIM'91, 1991, pp. 288–298.

ACKNOWLEDGEMENT. This study was supported by research funds from Chosun University 2003.

Received 1 September 2004; revised accepted 11 April 2005

Geum-Bae Cho is in the Department of Electrical Engineering, College of Engineering, Chosun University, 375 Seosuk-dong, Dong-gu, Gwangju, Republic of Korea; Pyoung-Ho Kim is in the Department of Information and Communication, Seokang College 789-1 woon am-dong, Pukku, Gwang ju, Republic of Korea.*
**For correspondence.*
e-mail: gbcho@chosun.ac.kr