Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 14, No. 2, June 2014, pp. 136-144
http://dx.doi.org/10.5391/IJFIS.2014.14.2.136

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Implementation of Fuzzy Self-Tuning PID and Feed-Forward Design for High-Performance Motion Control System

Ngo Ha Quang Thinh[1] **and Won-Ho Kim**[2]
[1]Department of Intelligent System Engineering, Graduate School of Dong-Eui University, Busan, Korea
[2]Department of Mechatronics Engineering, Dong-Eui University, Busan, Korea

Ijfis

Abstract

The existing conventional motion controller does not perform well in the presence of nonlinear properties, uncertain factors, and servo lag phenomena of industrial actuators. Hence, a feasible and effective fuzzy self-tuning proportional integral derivative (PID) and feed-forward control scheme is introduced to overcome these problems. In this design, a fuzzy tuner is used to tune the PID parameters resulting in the rejection of the disturbance, which achieves better performance. Then, both velocity and acceleration feed-forward units are added to considerably reduce the tracking error due to servo lag. To verify the capability and effectiveness of the proposed control scheme, the hardware configuration includes digital signal processing (DSP) which plays the main role, dual-port RAM (DPRAM) to guarantee rapid and reliable communication with the host, field-programmable gate array (FPGA) to handle the task of the address decoder and receive the feed-back encoder signal, and several peripheral logic circuits. The results from the experiments show that the proposed motion controller has a smooth profile, with high tracking precision and real-time performance, which are applicable in various manufacturing fields.

Keywords: Fuzzy control, Proportional integral derivative control, Motion system, Feed-forward control, Digital signal processing, AC servo

1. Introduction

The development of industrial motion control systems has expanded considerably even though it just emerged over the last several decades. The most significant component of a motion control system is the motion controller, with the control strategy being the main technique. Generally, industrial motion control schemes must solve two major problems. The first problem deals with the tracking command. In most of industrial motion controllers, the motion trajectory, which is pre-defined by the input parameters, is generated as a tracking command. Hence, the motion controller assesses the performance of how well the actual motion follows what was determined in advance. Traditional solutions utilize the proportional integral derivative (PID) algorithm, which is applied in various manufacturing systems. Qu and Wang [1] implemented a PID controller in the programmable multi-access controller (PMAC) card to control the AC servo motor. However, this design, which is based on a complex debugging process, depends on the specific control device. In [2], the authors introduced a least square parameter estimation algorithm for PID control, which made movement smoother.

Received: Jan. 13, 2014
Revised : Mar. 25, 2014
Accepted: Apr. 7, 2014

Correspondence to: Won-Ho Kim
(kwh@dau.ac.kr)
©The Korean Institute of Intelligent Systems

However, the parameter estimator algorithm increases the computational cost and decreases the speed of serial communication, which does not satisfy the high performance requirements in industry. Zhao and Zhang [3] presented a fractional-order PID scheme that continuously alters the frequency response of the system. This method is not trivial to implement in practical uses and does not guarantee satisfactory communication speed. The second problem tackles the disturbance rejection characteristics of the system. On the motor shaft, torque disturbance is common in the manufacturing environment. Thus, there is a need to design a controller for compensating for these problems. Tran et al. [4] studied motion control on the electronic-centralized aircraft monitor (ECAM) system, introducing the auto-tuning PID velocity controller to overcome the uncertainties. The gains realized from this controller are updated constantly to guarantee the consistency of system performance under varying working conditions. However, this controller depends on only one specific motion system and cannot be widely implemented. Several additional control methods have been introduced, such as optimal control [5], adaptive robust control [6], adaptive robust neural network [7] [8], variable structure control [9] and linear-quadratic regulator (LQR) [10]. However, these solutions do not incorporate existing human knowledge into their algorithm. Owing to the system complexity, nonlinearity and uncertainty, it is difficult to obtain good performance in a variety of manufacturing machines based on the exact mathematical models.

An intelligent control technique, especially the fuzzy control technique (FCT), can overcome the difficulties created by conventional methodologies. This is a knowledge-based controller that is focused on the reasoning mechanism rather than concentrating uniquely on the model of the system. In [11], researchers investigated a control diagram for high speed control of an interior permanent magnet synchronous motor by computing the operational point. Then in the fuzzy logic controller, these values are input and the parameters of the controller are tuned by a genetic algorithm. Conversely, this method only presents analysis and development theory. However, it lacks practical implementation in manufacturing conditions. Other researchers [12] designed a two-input normalized linear fuzzy logic controller incorporated with a traditional derivative controller to control both the linear and nonlinear systems. However, this controller is still not suitable for most manufacturing machines because of the high computational burden and complex use. Authors in [13] researched a switching mechanism between traditional PID-based and fuzzy-PID-based controllers using a generic technique for optimizing the scaling values. However, the results have only obtained in a laboratory environment where practical programming and a debugging control language are absent.

In modern and practical production environments, a feed-forward scheme promises to improve the tracking performance. Tsai et al. [14] indicated a complete feed-forward compensator in a computer numerical control (CNC) machine. The structure of the feed-forward control can also be applied in the motion planning algorithm, where it maintains the characteristics such as optimizing time, reliability and precision. For the other implementation issues, authors [15, 16] suggested a high order feed-forward planning strategy resulting in a significant performance improvement. The details of the fourth-order trajectory planning algorithm proved a practical, reliable and accurate outcome. In [17, 18], the stabilizing design method of a motion controller in the presence of uncertainty and disturbances is discussed by using the robust internal loop compensator. This design provides a systematic approach to the problem of robust stability and the desired performance in the presence of uncertainties.

In this paper, we presented a feasible and effective design of a fuzzy self-tuning PID with a feed-forward controller, which offers great precision and high performance applicable in the machining industry. This knowledge-based control scheme is robust enough to handle systems with external disturbances to the motor shaft, model free systems, and systems for which accurate information is not required. The remainder of this paper is structured as follows. In Section 2, the proposed controller design is presented. Section 3 describes the whole motion control system. Experimental results are shown in Section 4. Finally, conclusions are discussed in Section 5.

2. Proposed Controller Design

2.1 Conventional PID Controller

Although the PID controller was well-known in various industrial areas due to its simple implementation and easy control, it revealed several drawbacks in modern industry. Basically, there were two methods to achieve the required performance gains - trial-and-error or analysis. It can be seen that the trial-and-error approach relies noticeably on the technician's personal experience with a specific servo system. Once the significant practice becomes low or intuitive performance is unclear, it is difficult to know whether you have reached the optimal performance gains. Through analysis of the servo system, Ziegler and Nichols [19] introduced a method to tune three parameters of the PID controller. However, all three performance gains are inter-related, since altering one affects any of the previous adjustments. Moreover, the conventional PID controller does not yield reasonable performance over a wide range of operating conditions due to the fixed gain. Therefore, there is a critical need to design a high performance controller for modern industrial motion control systems.

2.2 Fuzzy Self Tuning PID and Feed-Forward Controller

The coefficients of the conventional PID controller are not often adapted for nonlinear models with unpredictable parameter variations. To overcome these uncertainties, the PID controller was engineered to automatically tune the parameters, called fuzzy self-tuning PID (FPID), while maintaining the simple control characteristic and eliminating the steady state error. Nevertheless, with only a feedback control structure, the tracking errors due to servo lag phenomena still occur. One existing solution is to add the feed-forward control scheme into the main controller to enhance tracking performance. For this kind of design, the feed-forward control scheme is available to change the values of parameters in the position loop controller. In consequence, the overall controller is presented in Figure 1, which is composed of the feed-forward control unit, a fuzzy inference mechanism, and a self-adjusting PID controller. As can be seen from Figure 1, each of three gains K$_P$, K$_I$ and K$_D$ is tuned using the fuzzy inference tuner.

The output of the proposed controller is presented by:

$$u = u_{FF} + u_{FPID} \tag{1}$$

where u_{FF}, u_{FPID}: the control efforts of the feed-forward control unit and the fuzzy self-tuning PID controller respectively.

$e(t), \dfrac{de}{dt}$: the error and deviation of error.

$K_P,\ K_I,\ K_D$: proportional, integral and derivative gains of the PID controller respectively.

$K_{vff},\ K_{aff}$: velocity and acceleration of the feed-forward gains.

Position, velocity and acceleration references are inputs from the motion profile generator.

The first step in designing the fuzzy inference process is to establish the linguistic fuzzy variables for the inputs and outputs. In Figure 2, the inputs of fuzzy inference are error and deviation of error, while the outputs are used for adjusting the PID controller. The membership functions of these fuzzy sets are triangles and trapezoids because of simple and effective programming. In this research, the linguistic variables are assigned as negative big (NB), negative small (NS), zero (Z), positive small (PS) and positive big (PB) for both input sets. As a result, I_i and I_j $(i, j = 1 \ldots 5)$ represent a fuzzy set for input variables, then, $\mu_{I_i}(e)$ and $\mu_{I_j}(\Delta e)$ $(i, j = 1 \ldots 5)$ indicates the respective membership function. Similarly, the linguistic levels for the outputs consist of small (S), medium small (MS), medium (M), medium big (MB) and big (B). Additionally, in order to guarantee the real-time performance by cutting the computational cost, the membership functions for the output fuzzy sets are fuzzy singletons. Consequently, the membership functions for these sets are symbolized as μ_{O_k} $(k = 1 \ldots 5)$.

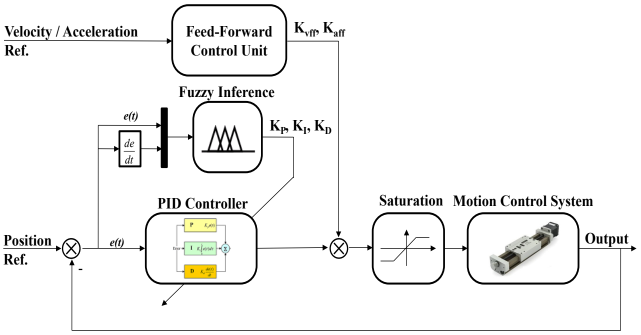

Figure 1. Control scheme of proposed controller.

Table 1. Fuzzy rules of fuzzy tuner

Error	Deviation of error				
	NB	NS	Z	PS	PB
NB	S	S	MS	MS	M
NS	S	MS	MS	M	MB
Z	MS	MS	M	MB	MB
PS	MS	M	MB	MB	B
PB	M	MB	MB	B	B

NB, negative big; NS, negative small; Z, zero; PS, positive small; PB, positive big; S, small; MS, medium small; M, medium; MB, medium big; B, big.

The rule base includes a set of linguistic IF-THEN rules containing two antecedents and one consequence, as expressed in the following form:

$$R_{x,y,z}: \text{ IF } e = A_x \text{ AND } \Delta e = B_y \text{ THEN Output } = C_z \tag{2}$$

where $1 \leq x, y, z \leq 5$.

The conjunction of the rule's antecedents is determined by the fuzzy operation intersection which is implemented by the minimum operator. The total number of IF-THEN rules is equal

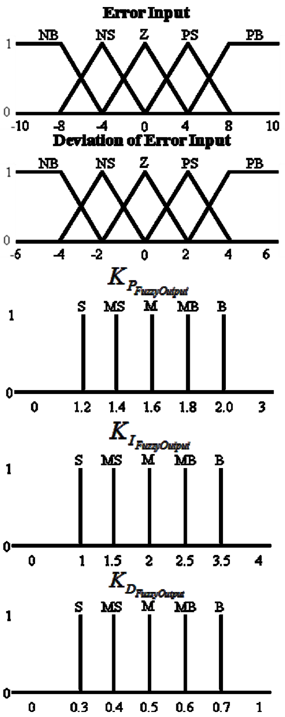

Figure 2. Membership functions of inputs and outputs. NB, negative big; NS, negative small; Z, zero; PS, positive small; PB, positive big; S, small; MS, medium small; M, medium; MB, medium big; B, big.

to the product of the number of fuzzy sets that make up each of the two fuzzy input variables. For that reason, a total of 25 fuzzy rules are required to represent each possible combination of the two fuzzy input variables used to create the output membership fuzzy sets. The fuzzy rules ensure that the output of the fuzzy controller improves the overall tracking performance.

The fuzzy inference engine uses the appropriately designed knowledge base to evaluate the fuzzy rules and produce an output for each rule. Table 1 shows the possible combinations of fuzzy values that are experienced by the drive system and the corresponding output membership function. Then the multiple outputs are transformed to a crisp output by defuzzification.

$$\text{Fuzzy}_{\text{output}} = \frac{\sum_{l=1}^{25} \mu_{O_l}\left(\min\left(\mu_{I_i}^{l}, \mu_{I_j}^{l}\right)\right)}{\sum_{l=1}^{25}\left(\min\left(\mu_{I_i}^{l}, \mu_{I_j}^{l}\right)\right)} \tag{3}$$

Then, the three outputs are substituted into the following equations to compute the tuning gains in (1).

$$\begin{cases} K_{P_{Tuning}} = \frac{K_{P_{FuzzyOutput}} - K_{P_{Min}}}{K_{P_{Max}} - K_{P_{Min}}} \\ K_{I_{Tuning}} = \frac{K_{I_{FuzzyOutput}} - K_{I_{Min}}}{K_{I_{Max}} - K_{I_{Min}}} \\ K_{D_{Tuning}} = \frac{K_{D_{FuzzyOutput}} - K_{D_{Min}}}{K_{D_{Max}} - K_{D_{Min}}} \end{cases} \tag{4}$$

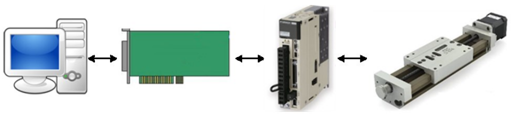

Figure 3. Experimental motion control system.

A saturation unit is also incorporated in the overall controller structure. As such, the final output of the proposed controller is,

$$u_{sat} = \begin{cases} u_{\max} & u_{\max} < u \\ u & u_{\min} < u < u_{\max} \\ u_{\min} & u < u_{\min} \end{cases} \tag{5}$$

where $u_{\max}$, $u_{\min}$: the permitted maximum and minimum inputs to the servo drive system.
u_{sat}: the output of saturation unit.

3. Motion System Descriptions

In Figure 3, a digital signal processing (DSP)-based motion control system is shown. The host is personal computer with the Windows 7 operating system. The proposed controller can communicate with the host through a peripheral component interconnect (PCI) bus. The motion controller outputs analog signals to the servo driver in order to provide control. In this research, the servo driver and servo motor are manufactured by YASKAWA. The servo motor, SGMJV-02ADA21 200W, has a 20-bit incremental encoder driven by a high performance servo driver, SGDV-1R6A01B.

In this system, the proposed DSP-based motion controller is designed with the CPU as a key component. This microprocessor handles multi-tasking operations such as, motion planning, motion detection, and arithmetic computation, based on the powerful resources available on-chip.

3.1 Hardware Development

In Figure 4, the hardware design of the proposed controller consists of DSP, DPRAM, static RAM (SRAM), Flash, FPGA and other peripheral components. the user controls the full S-curve movement mode (acceleration mode) on the host computer. This controller communicates with the host through the PCI-bus at 33MHz, thus, it receives the motion command from the host. The high speed and reliable data communication is guaranteed by DPRAM which allows two directional operations simultaneously. The DSP reads the motion parameters in the shared memory area and calculates the smooth profile. Several

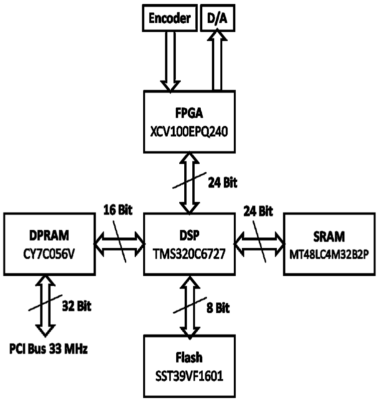

Figure 4. Hardware block diagram of proposed controller.

motion coefficients are saved in SRAM in order to obtain them easily when needed by the CPU, and the data in Flash is used for the initial motion information and some boot functions. The DSP generates the profile based on polynomial equations. The command position is sent to the main controller as a reference signal. Then, the result of the computation is transferred to a digital-analog conversion. The servo driver receives analog control value and sends the feedback value to the motion controller. In this design, FPGA acts as the address decoder and receives the encoder signal.

3.2 Software Development

The components of the software design include a graphical user interface (GUI), C++ and C programs. In the Windows OS, GUI allows a user to interact with the motion control system using imaging demonstrations rather than textual commands. Intuitively, parameters are input and displayed while the motion is executed. A motion library, programmed in Visual Studio 2008, helps to classify and synchronize the data flow between the host and the motion controller. At a lower level, the C program, mainly operating in Code Composer Studio version 3.3, manages most of the tasks of the motion controller.

To organize the program systematically and steadily, varieties of channels and commands are put forward. When motion is accomplished, the channel is ready. Otherwise, the channel is busy. The request status of the command first arrives to set the parameters in the C code program. The command executes when the preparation for motion is finished. During the process of motion, the command returns an error type if any error occurs. If not, the status of the command is ready for the next operation.

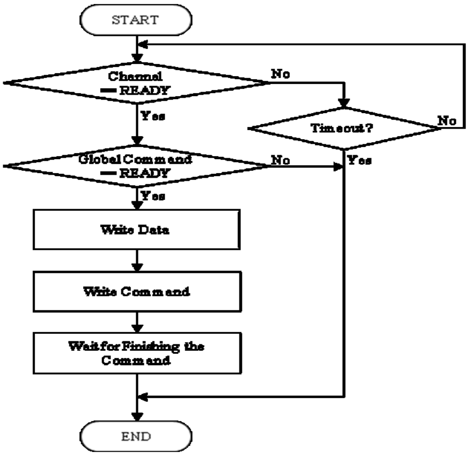

Figure 5. Flowchart of global command.

As shown in Figure 5, the global command that handles the communication of the host and the controller is presented. After the confirmation of the channel and global command being ready, data is written into the memory area correspondingly. The host sends the motion command which takes into account the acceleration mode, blending mode and step mode to start the motion.

The main interrupt service routine (ISR) is described in Figure 6. Initially, hardware data is read from the encoder to update the feedback signal. In this stage, positive/negative limit sensors and home sensors are also renewed. Later, the digital-to-analog converter (DAC) data of the previous computation is output. From the encoder values, actual positions can be obtained. Consequently, a motion profile is generated referencing its real position and velocity in the generator function. Afterward, the difference between the command or target values and the actual values is calculated and transmitted into the fuzzy self-tuning PID controller. The final DAC data is recorded in the register before the ISR ends.

In the C program, there is a main loop which continuously monitors the execution of the motion controller. The loop detects the signal from the sensors, the error in the amplifier and the encoder or a large gap between the command value and the actual value. It also verifies whether motion is in-position or not.

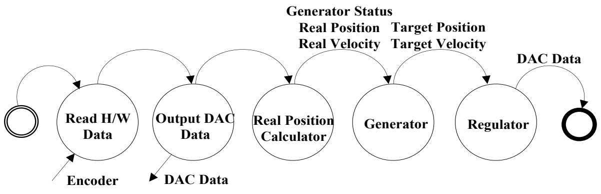

Figure 6. Flowchart of interrupt service routine. DAC, digital-to-analog converter.

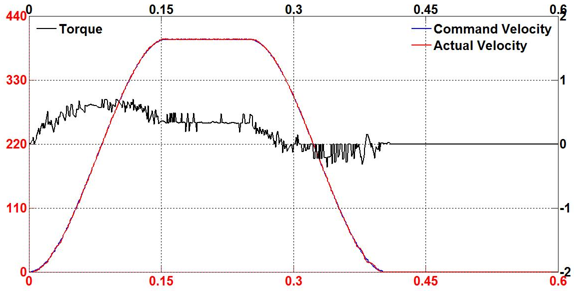

Figure 7. Tracking performance of proposed controller in acceleration motion mode.

4. Experimental Results

In order to evaluate the performance of the proposed fuzzy self-tuning PID and feed-forward controller, several experiments were conducted in a variety of machining conditions. The tests were also conducted using the conventional PID controller under the same conditions. The overall hardware of the motion system was operated and observed in three common modes: acceleration motion, blending motion and step motion. Also, three monitoring parameters were collected to clarify the status of the motion system. Command velocity plays a role as a reference signal from the motion profile generator while the actual velocity makes an effort to track the pre-determined path. The torque value is the output control torque of the proposed controller. Figure 7-9 show the tracking results of the proposed control scheme in all three modes. It can be seen that the fuzzy self-tuning PID and feed-forward controller brings the actual speed to the specified desired speed smoothly.

In comparison with the proposed controller, Figure 10-12 show the results of the experiments using the conventional PID controller. In both cases, it is also observed that the tracking responses swing when motion is in progress.

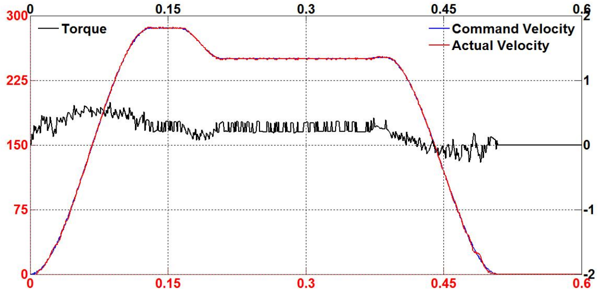

Figure 8. Tracking performance of proposed controller in blending motion mode.

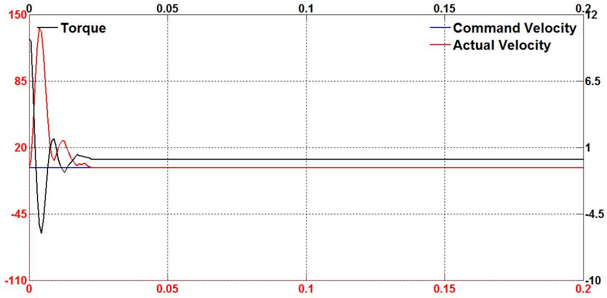

Figure 9. Tracking performance of proposed controller in step motion mode.

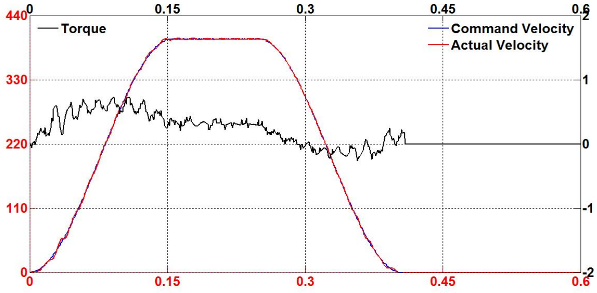

Figure 10. Tracking performance of PID controller in acceleration motion mode.

To demonstrate the capability and effectiveness of the proposed controller, external disturbance was randomly applied to the torque control on the motor shaft in the servo drive system while the experiment was on track. Figure 13-15 show the performance of the proposed controller design for both motion modes in the existence of external and sudden disturbance. It is noticed that the proposed controller effectively eliminates the disturbance and continues to track the desired velocity under these conditions. When the disturbance occurs, the motion system tends to oscillate or overshoot. Instantly, the control scheme adjusts to drive the servo motor to the desired profile.

Figure 16-18 show the test results for the conventional PID

Table 2. Comparison of root mean square tracking errors in test cases

Mode control scheme	Acceleration	Blending	Step
PID	$1.2121e-5$	$1.4903e-5$	$1.3036e-4$
FPID-FF	$1.1608e-5$	$1.2979e-5$	$1.1015e-4$
PID with disturbance	$1.6434e-5$	$1.9640e-5$	$1.4736e-4$
FPID-FF with disturbance	$1.2363e-5$	$1.3584e-5$	$1.2869e-4$

PID, proportional integral derivative.

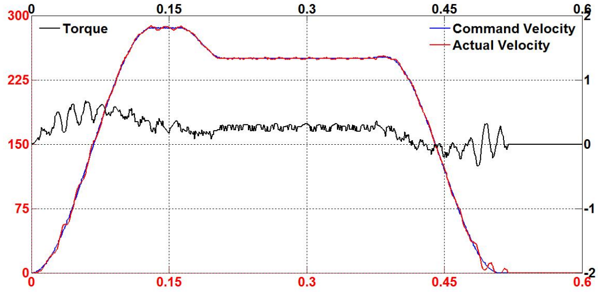

Figure 11. Tracking performance of proportional integral derivative controller in blending motion mode.

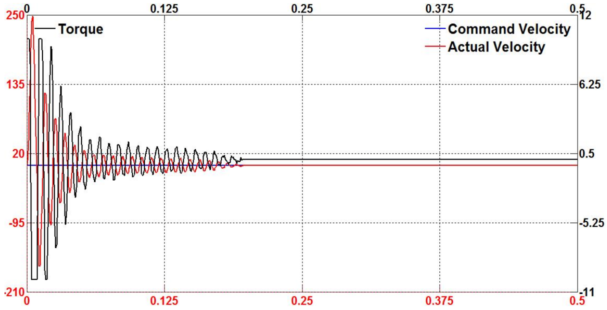

Figure 12. Tracking performance of proportional integral derivative controller in step motion mode.

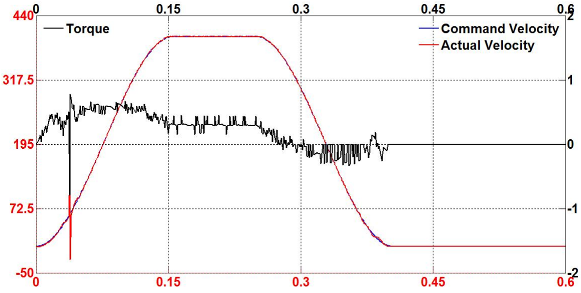

Figure 13. Tracking performance of proposed controller in acceleration motion mode under disturbance.

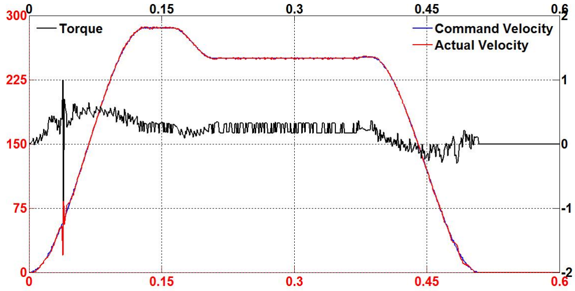

Figure 14. Tracking performance of proposed controller in blending motion mode under disturbance.

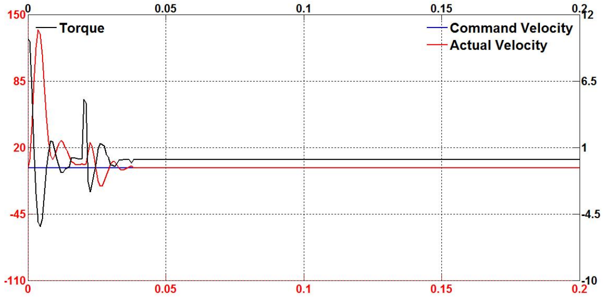

Figure 15. Tracking performance of proposed controller in step motion mode under disturbance.

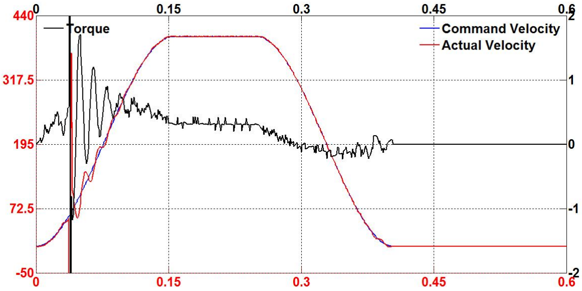

Figure 16. Tracking performance of proportional integral derivative controller in acceleration motion mode under disturbance.

controller under external disturbances for comparison with the proposed fuzzy self-tuning PID and feed-forward controller. It can also be seen that the conventional controller suffers from the disturbance with significant overshoot and undesirable oscillatory response. Table 2 shows the root mean square (RMS) error values during practical tests. From these results, it is suggested that the proposed motion controller generates high performance for industrial motion applications.

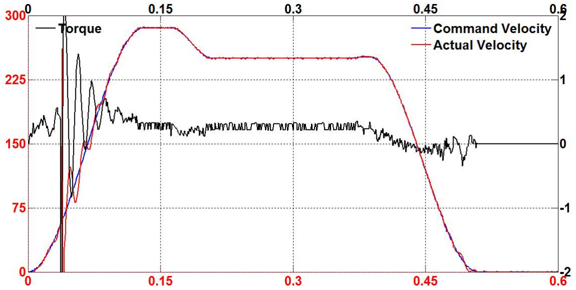

Figure 17. Tracking performance of proportional integral derivative controller in blending motion mode under disturbance.

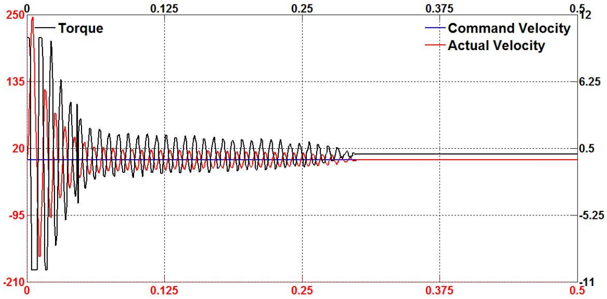

Figure 18. Tracking performance of proportional integral derivative controller in step motion mode under disturbance.

5. Conclusions

A feasible and effective fuzzy self-tuning PID and feed-forward motion controller has been introduced in this paper. The resulting experiments have shown great tracking performance of the proposed controller, a smooth profile and real-time response. The practical implementation was accomplished on a DSP-based motion controller and a Windows-based software environment. Under these conditions, motion control engineers have the advantages of programming control and debugging methods to express the interest in accuracy manufacturing system. Therefore, the proposed controller applies to a wide range of industrial motion control systems.

In our future work, other algorithms such as linear/circular interpolation and a homing method will be implemented into the proposed controller. Additionally, although the current performance satisfies most of the requirements, motion synchronization should be considered in a multiple axes system.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This work was supported by Dong-Eui University Foundation Grant (grant number 2013AA138).

References

[1] B. Qu and H. Wang, "Study of PID simulation based on PMAC servo system," in *Proceedings of the International Conference on Electronic and Mechanical Engineering and Information Technology*, Harbin, China, August 12-14, 2011, pp. 440-443. http://dx.doi.org/10.1109/EMEIT.2011.6022913

[2] H. Xiao and S. Wang, "Auto-tuning PID module of robot motion system," in *Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications*, Beijing, China, June 21-23, 2011, pp. 668-673. http://dx.doi.org/10.1109/ICIEA.2011.5975670

[3] C. Zhao and X. Zhang, "The application of fractional order PID controller to position servomechanism," in *Proceedings of the 7th World Congress on Intelligent Control and Automation*, Chongqing, China, June 25-27, 2008, pp. 3380-3383. http://dx.doi.org/10.1109/WCICA.2008.4593462

[4] Q. V. Tran, W. H. Kim, J. H. Shin, and W. B. Baek, "ECAM control system based on auto-tuning PID velocity controller with disturbance observer and velocity compensator," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 10, no. 2, pp. 113-118, Jun. 2010.

[5] A. Mannava, S. N. Balakrishnan, L. Tang, and R. G. Landers, "Optimal tracking control of motion systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1548-1558, Nov. 2012. http://dx.doi.org/10.1109/TCST.2011.2168608

[6] B. Yao, "Advanced motion control: from classical PID to nonlinear adaptive robust control," in *11th IEEE International Workshop on Advanced Motion Control*, Nagaoka, Japan, March 21-24, 2010, pp. 815-829. http://dx.doi.org/10.1109/AMC.2010.5464026

[7] W. H. Kim, “Implementation of an adaptive robust neural network based motion controller for position tracking of AC servo drives,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 9, no. 4, pp. 294-300, Dec. 2009.

[8] M. J. Lee and T. S. Jin, “An adaptive tracking control for robotic manipulators based on RBFN,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 7, no. 2, pp. 96-101, Jun. 2007.

[9] K. Kondak and G. Hommel, “Design of a robust high gain PID motion controller using sliding mode theory,” in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Niagara Falls, Canada, July 29-August 1, 2005, pp. 26-31. http://dx.doi.org/10.1109/ICMA.2005.1626517

[10] M. T. Leines and J. S. Yang, “LQR control of an under actuated planar biped robot,” in *Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications*, Beijing, China, June 21-23, 2011, pp. 1684-1689. http://dx.doi.org/10.1109/ICIEA.2011.5975861

[11] M. N. Uddin and M. A. Rahman, “High-speed control of IPMSM drives using improved fuzzy logic algorithms,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 190-199, Feb. 2007. http://dx.doi.org/10.1109/TIE.2006.888781

[12] M. J. Er and Y. L. Sun, “Hybrid fuzzy proportional-integral plus conventional derivative control of linear and nonlinear systems,” *IEEE Transactions on Industrial Electronics*, vol. 48, no. 6, pp. 1109-1117, Dec. 2001. http://dx.doi.org/10.1109/41.969389

[13] A. Rubaai, M. J. Castro-Sitiriche, and A. R. Ofoli, “DSP-based laboratory implementation of hybrid fuzzy-PID controller using genetic optimization for high-performance motor drives,” *IEEE Transactions on Industry Applications*, vol. 44, no. 6, Nov-Dec. 2008. http://dx.doi.org/10.1109/TIA.2008.2006347

[14] M. C. Tsai, I. F. Chiu, and M. Y. Cheng, “Design and implementation of command and friction feedforward control for CNC motion controllers,” *IEE Proceedings: Control Theory and Applications*, vol. 151, no. 1, pp. 13-20, Jan. 2004. http://dx.doi.org/10.1049/ip-cta:20040100

[15] P. Lambrechts, M. Boerlage, and M. Steinbuch, “Trajectory planning and feedforward design for electromechanical motion systems,” *Control Engineering Practice*, vol. 13, no. 2, pp. 145-157, Feb. 2005. http://dx.doi.org/10.1016/j.conengprac.2004.02.010

[16] M. Boerlage, R. Tousain, and M. Steinbuch, “Jerk derivative feedforward control for motion systems,” *Proceedings of American Control Conference*, Boston, MA, June 30-July 2, 2004, pp. 4843-4848.

[17] B. K. Kim and W. K. Chung, “Performance tuning of robust motion controllers for high-accuracy positioning systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, Dec. 2002. http://dx.doi.org/10.1109/TMECH.2002.805626

[18] S. Seo, “Adaptive fuzzy sliding mode control for uncertain nonlinear systems,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 11, no. 1, pp. 12-18, Mar. 2011. http://dx.doi.org/10.591/IJFIS.2011.11.1.012

[19] J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” *Transactions of the American Society of Mechanical Engineers*, vol. 64, pp. 759-768, 1942.

Ngo Ha Quang Thinh received the B.S. degree in School of Mechatronics Engineering from University of Technology, Vietnam, and the M.S. degree in Department of Intelligent System Engineering from Dong-Eui University, Korea, in 2006 and 2009 respectively. He is currently Ph.D. student in Department of Intelligent System Engineering, Dong-eui University and researcher of AJINEXTEK Co. Ltd., Korea. His research interests include motion control, real-time systems.

E-mail: hqtngo@ajinextek.com

Won-Ho Kim received the B.S., M.S. and Ph.D. degree in School of Electronics Engineering from Kyungpook National University, Daegu, Korea, in 1985, 1988 and 1999 respectively. From 1988 to 1993, he was a researcher of Electronics and Telecommunications Research Institute (ETRI). He is currently a professor in the Department of Mechatronics Engineering, Dong-Eui University, Busan, Korea. His research interests include robotics and embedded control system.

E-mail: kwh@deu.ac.kr