Available online at www.sciencedirect.com
ScienceDirect
International Journal of Machine Tools & Manufacture 48 (2008) 851–857
INTERNATIONAL JOURNAL OF MACHINE TOOLS & MANUFACTURE DESIGN, RESEARCH AND APPLICATION
www.elsevier.com/locate/ijmactool

Parameter identification when designing a solid-modeling-based grinding-machine controller

Shu-Hung Liu, Yi-Chih Lai, Jia-Yush Yen*

Department of Mechanical Engineering, National Taiwan University, Taipei 10617, Taiwan, ROC

Received 19 June 2007; received in revised form 30 November 2007; accepted 4 December 2007
Available online 30 January 2008

Abstract

Modern computer-aided design and dynamics simulation tools allow design and testing to be performed in a virtual-reality environment. The integration of controller synthesis programs also makes it possible to perform servo tuning without actually having to build a prototype. However, most servo controllers are based on integer processors for machine tools, and the integer numbers entering into a servo tuning interface often do not directly reflect the magnitudes of the actual controller parameters, impairing the virtual tuning. This study addressed this problem by employing a gain parameter to each control parameter, and developing an identification algorithm for these parameter gains. The identification results were used to tune the servo, which was first carried out within the solid-modeling simulation environment. This procedure makes it easy to implement the controller parameters in the actual machine. The actual implementation of the resultant controller verified that the proposed method improved the servo performance, including eliminating the overshoot in both position and velocity responses of the grinding machine.

Keywords: Solid-modeling-based design; Controller parameter identification; Integer processor controller

1. Introduction

The new trend for designing high-end precision machines relies heavily on computer-aided design (CAD) tools to minimize the design effort and to speed up the design process. A very realistic representation of a machine and an examination of its dynamics behavior can be implemented within a software environment [1]. Recent advancements in software technology have also enabled servo tuning to be included within a virtual-reality environment [2,3].

Conventional approaches to machine design rely heavily on the designer's experience, and prototyping is expensive, which means that the ability to successfully develop prototypes in the virtual world provides many advantages for manufacturers. There are many research results on the design of servos for precision machine tools [4–13]. These efforts have aimed at establishing suitable mathematical models for the underlining machining process, with some actual controllers being based on these models. Basic tuning rules, based on the system responses, have been analyzed [14,15], but there have been very few reports on performing servo tuning within a virtual environment. An FEM model has been used for volumetric error estimation [16], and a design and tuning process has been attempted within a virtual solid-modeling environment [17]. However, no research effort has addressed the actual implementation of a controller interface.

In this study we addressed the issue of interfacing between the actual controller and the virtual environment. The CAD/CAM environment allows engineers to perform mechanical design and servo tuning with a virtual-reality solid model. However, attempts to tune the solid-modeling process depend on the ability to convert the virtual environment parameters into the actual hardware parameters, and hence we also present a validation of our simulation results. A coefficient vector was used to represent the differences in the numerical representations, and an algorithm for identifying controller parameters has been developed. The target machine in this study

*Corresponding author. Tel./fax: +886 2 33662688.
E-mail address: jyen@ntu.edu.tw (J.-Y. Yen).

0890-6955/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijmachtools.2007.12.005

was a Chevalier FGP-608LM CNC grinding machine. We operated the test machine under different conditions, and compared the obtained signals with the simulation responses. These comparisons could also be used to validate assumptions made regarding hard-to-measure machine parameters, such as the friction coefficient and acceleration time constants. The validated results show that this proposed procedure can provide a very detailed solid model for the design of a machine servo, with engineers being able to perform servo tuning within the virtual environment and compute the actual numerical numbers needed for the settings of an actual machine controller.

2. Solid modeling

2.1. Grinding machine

The Chevalier FGP-608LM CNC grinding machine (Fig. 1(a)) used in this study comprises three major components: the spindle, the moving worktable, and the base of the machine. The grinding wheel mounted on the spindle can move linearly in the y-direction and can rotate $\pm 5^{\circ}$ about the x-axis. The grinding machine uses two sets of high-thrust linear motors to drive the worktable rapidly back and forth in the x-direction for high-speed grinding (Fig. 1(b)).

2.2. Software integration

This study used several software packages to achieve realistic simulations. The CAD model with detailed structural and dimensional data for the machine was first built in Pro-Engineer. The moving part of the CAD model (i.e., the worktable with two guide rails that regulate the sliding motion) was then imported into the software MSC.visualNastran Desktop (Fig. 2(a)) for dynamic simulation. The worktable contacts the guide rails at four bearing sliders indicated by A–D in Fig. 2(b). The dynamic model was then integrated into the MATLAB Simulink environment (Fig. 3) for system analysis and controller synthesis. Because the simulation was based on the underlining CAD solid model, the servo design was very realistic, and the resultant servo design was directly applicable.

2.3. Machine base

The base of the machine not only supports the weight of the machine but also receives the reaction force from the linear motor. This study incorporated base resonance effects by artificially adding the dominant resonance modes to the MATLAB Simulink environment. We employed a series of second-order mass–spring systems to represent the resonance modes, with dominating resonance modes of

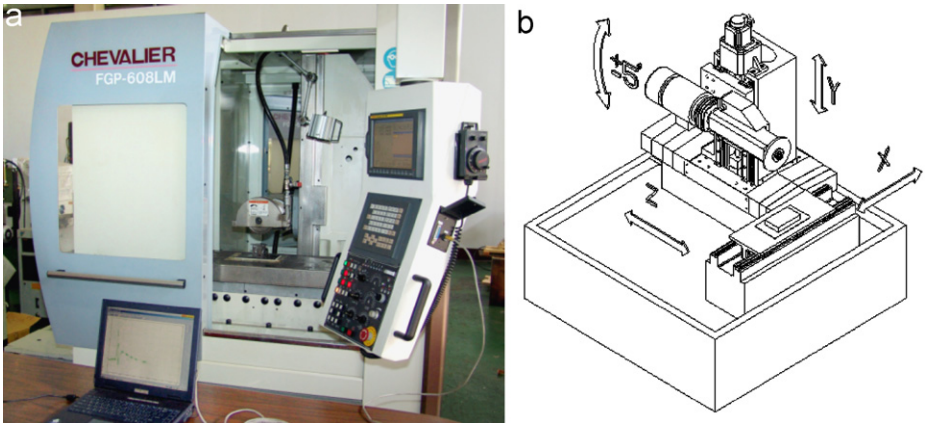

Fig. 1. (a) FGP-608LM grinding machine. (b) Design concept and coordination.

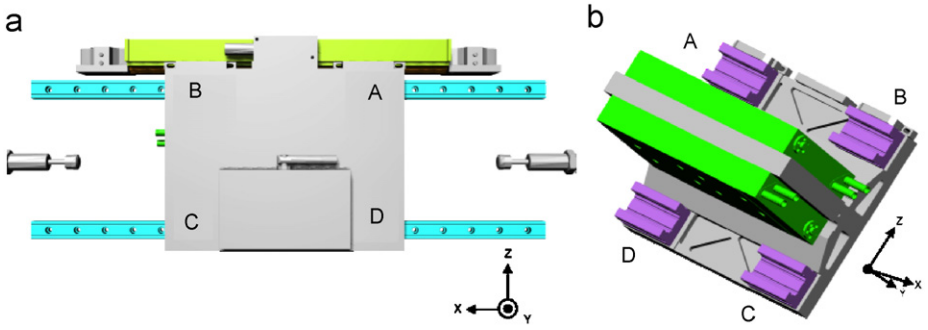

Fig. 2. Top (a) and bottom (b) views of the worktable in MSC.visualNastran Desktop.

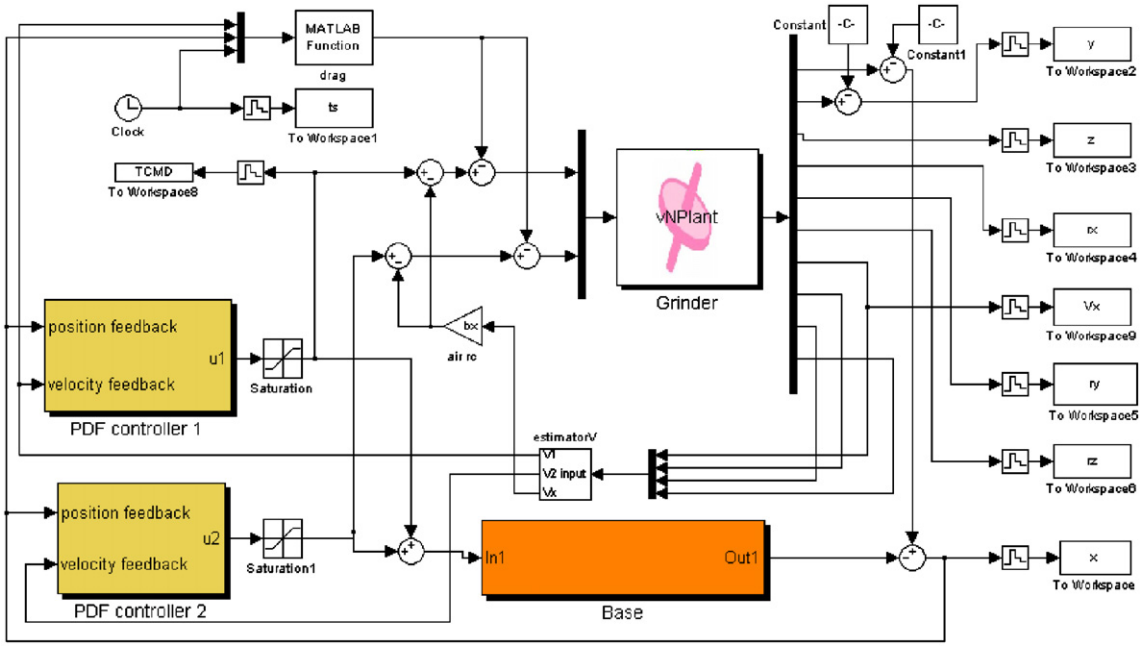

Fig. 3. Integration of the solid model in the MATLAB Simulink environment.

86.1, 88.69, 134.19, 140.92, and 162.86 Hz, as reported by the Chevalier machinery.

3. Identification of controller parameters

3.1. Estimation of the controller parameters

The servo loop in the machine-tool controller was based on the pseudo-derivative-feedback (PDF) algorithm. The PDF control is very popular in machine-tool applications because it has been shown to be robust against variations in control parameters.

Referring to the servo system block diagram in Fig. 4, let

$$C_1(z) = \frac{U_1(z)}{E(z)} = \frac{d_0 + d_1 z^{-1} + \cdots + d_m z^{-m}}{1 + c_1 z^{-1} + \cdots + c_m z^{-m}}, \quad (1)$$

$$C_2(z) = \frac{U_2(z)}{Y(z)} = \frac{h_0 + h_1 z^{-1} + \cdots + h_p z^{-p}}{1 + g_1 z^{-1} + \cdots + g_p z^{-p}}, \quad (2)$$

and let the machine plant be

$$P(z) = \frac{Y(z)}{U(z)} = \frac{b_1 z^{-1} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}},$$

where

$$U(z) = U_1(z) + U_2(z), \quad E(z) = R(z) - Y(z).$$

The closed-loop system can be written as

$$\begin{aligned} u(k) = {} & d_0 e(k) + d_1 e(k-1) + \cdots + d_m e(k-m) \\ & - c_1 u_1(k-1) - \cdots - c_m u_1(k-m) \\ & + h_0 y(k) + h_1 y(k-1) + \cdots + h_p y(k-p) \\ & - g_1 u_2(k-1) - \cdots - g_p u_2(k-p), \end{aligned} \quad (3)$$

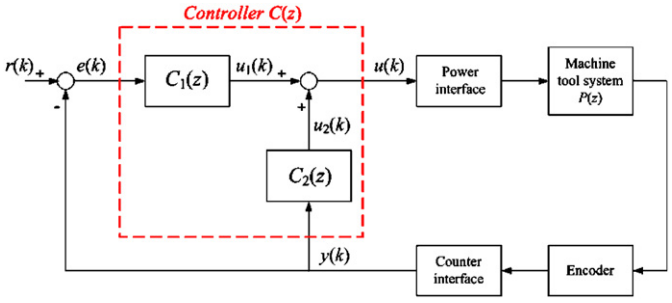

Fig. 4. PDF-controller block diagram of the grinding-machine servo system.

and

$$\begin{aligned} y(k) = {} & b_1 u(k-1) + \cdots + b_n u(k-n) - a_1 y(k-1) \\ & - \cdots - a_n y(k-n), \end{aligned}$$

where

$$e(k) = r(k) - y(k).$$

Control effort $u(k)$ is the interfacing signal from the controller D/A, and system output $y(k)$ is the position or velocity measurement signal that is sent back to the A/D converter. These signals are the interfacing signals between the digital control algorithm and the actual machine tool, and therefore are the most convenient signals for external data processing or system identification. The servo controller is usually implemented with integer microprocessors. Unlike the usual floating-point programs, the variables in an integer program usually do not equal the floating-point numbers in the designed controller, which makes it necessary to consider the numbers entered into the

controller setting interface for each control parameter. The easiest way to address this is to add a scaling factor to each of the control parameters to relate the actual controller parameter to the numbers set in the program.

Now consider the parameters set in the integer microprocessor controller as $d_{m0}, d_{m1}, \ldots, d_{mm}$, $c_{m1}, \ldots, c_{mm}$, $h_{p0}, h_{p1}, \ldots, h_{pp}$, $g_{p1}, \ldots, g_{pp}$. Replacing the actual machine-tool system $P(z)$ with the simulation plant $P^*(z)$ results in unknown gains $k_{d0}, k_{d1}, \ldots, k_{dm}, k_{c1}, \ldots, k_{cm}, k_{h0}, k_{h1}, \ldots, k_{hp}, k_{g1}, \ldots, k_{gp}$ within each controller coefficient. Therefore, (3) can be rewritten as

$$\begin{aligned} u^*(k) = {} & k_{d0}d_{m0}e^*(k) + k_{d1}d_{m1}e^*(k-1) + \cdots \\ & + k_{dm}d_{mm}e^*(k-m) - k_{c1}c_{m1}u_1^*(k-1) \\ & - \cdots - k_{cm}c_{mm}u_1^*(k-m) + k_{h0}h_{p0}y^*(k) \\ & + k_{h1}h_{p1}y^*(k-1) + \cdots + k_{hp}h_{pp}y^*(k-p) \\ & - k_{g1}g_{p1}u_2^*(k-1) - \cdots - k_{gp}g_{pp}u_2^*(k-p), \end{aligned} \tag{4}$$

where $u^*(k)$, $e^*(k)$, and $y^*(k)$ are the control effort, control error, and system output under the simulation plant $P^*(z)$, respectively. It is possible to define a diagonal matrix

$$\begin{aligned} \Theta = {} & diag[d_{m0}, d_{m1}, \ldots, d_{mm}, C_{m1}, \ldots, C_{mm}, \\ & h_{p0}, h_{p1}, \ldots, h_{pp}, g_{p1}, \ldots, g_{pp}], \end{aligned}$$

such that

$$u^*(k) = \phi^{*\mathrm{T}}(k)\Theta\psi. \tag{5}$$

In (5), the $2(m+p+1)\times 1$ measurement vector is defined as

$$\begin{aligned} \phi^*(k) = {} & [e^*(k), e^*(k-1), \ldots, e^*(k-m), \\ & -u_1^*(k-1), \ldots, -u_1^*(k-m), y^*(k), \\ & y^*(k-1), \ldots, y^*(k-p), -u_2^*(k-1), \ldots, \\ & -u_2^*(k-p)]^{\mathrm{T}}. \end{aligned}$$

One can now define a $2(m+p+1)\times 1$ gain-parameter vector ψ:

$$\begin{aligned} \psi = {} & [k_{d0}, k_{d1}, \ldots, k_{dm}, k_{c1}, \ldots, k_{cm}, k_{h0}, k_{h1}, \ldots, \\ & k_{hp}, k_{g1}, \ldots, k_{gp}]^{\mathrm{T}}, \end{aligned} \tag{6}$$

that is to be identified. Notice that

$$\begin{aligned} \phi^{*\mathrm{T}}(k)\Theta = {} & [d_{m0}e^*(k), d_{m1}e^*(k-1), \ldots, d_{mm}e^*(k-m), \\ & -c_{m1}u_1^*(k-1), \ldots, -c_{mm}u_1^*(k-m), \\ & h_{p0}y^*(k), h_{p1}y^*(k-1), \ldots, h_{pp}y^*(k-p), \\ & -g_{p1}u_2^*(k-1), \ldots, -g_{pp}u_2^*(k-p)]. \end{aligned}$$

It is now possible to process the signal using various parameter-identification algorithms to compute the gain-parameter vector ψ.

3.2. Actual machine parameters

Fig. 5 shows the PDF-controller structure built in the MATLAB Simulink environment for solid-modeling simulation. Parameters 19 and 260 correspond to the integer parameters set in the microprocessor controller in the actual machine; k_1 and k_2 are unknown gains to be

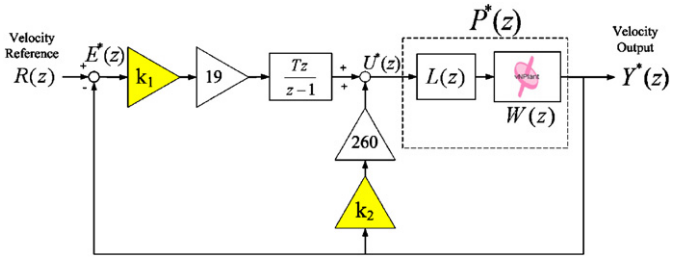

Fig. 5. Representation of the unknown gains in the machine-tool servo controller.

calculated when replacing the actual machine-tool system $P(z)$ with simulation plant $P^*(z)$. $P^*(z)$ is represented using the dynamic equation of linear motor $L(z)$ and the transfer function representation of worktable $W(z)$.

The discrete dynamic equation of the linear motor is

$$L(z) = \frac{K_{\mathrm{t}}Tz}{(L_{\mathrm{a}} - R_{\mathrm{a}}T)z - L_{\mathrm{a}}}, \tag{7}$$

where $L_{\mathrm{a}} = 0.008\,\mathrm{H}$ is the armature inductance, $R_{\mathrm{a}} = 1.3\,\Omega$ the armature resistance, $K_{\mathrm{t}} = 83.1$ the torque (or force) constant, and $T = 250\,\mu\mathrm{s}$ the sampling time of the discrete system. These values correspond to those in the user manual of the linear motor. Applying the autoregressive exogenous (ARX) model identification to the MSC.visualNastran Desktop model yields the transfer function

$$W(z) = \frac{0.45}{z - 0.85}.$$

From the parameter-identification procedure described above, (1) is rewritten as

$$C_1(z) = \frac{4.75\times 10^{-3}k_1}{1 - z^{-1}},$$

and (2) is rewritten as

$$C_2(z) = 260k_2.$$

Control effort $u^*(k)$ from (4) now becomes

$$u^*(k) = 4.75\times 10^{-3}k_1e^*(k) + u_1^*(k-1) + 260k_2y^*(k). \tag{8}$$

Substituting $u_1^*(k-1)$ by $u^*(k-1) - u_2^*(k-1)$ into (8) yields

$$\begin{aligned} u^*(k) = {} & 4.75\times 10^{-3}k_1e^*(k) + u^*(k-1) \\ & - 260k_2[y^*(k-1) - y^*(k)]. \end{aligned} \tag{9}$$

From (9), (5) can be rewritten as

$$\begin{aligned} u^*(k) = {} & \begin{bmatrix} e^*(k) & u^*(k-1) & y^*(k) - y^*(k-1) \end{bmatrix} \\ & \times \begin{bmatrix} 4.75\times 10^{-3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 260 \end{bmatrix} \begin{bmatrix} k_1 \\ 1 \\ k_2 \end{bmatrix}. \end{aligned} \tag{10}$$

The system output $y^*(k)$ can now be obtained using

$$\begin{aligned} Y^*(z) = P^*(z)U^*(z) &= [W(z)L(z)]U^*(z) \\ &= \frac{1.218z^{-1}}{1 - 1.892z^{-1} + 0.886z^{-2}}U^*(z), \end{aligned}$$

$$y^*(k) = 1.218u^*(k-1) + 1.892y^*(k-1) - 0.886y^*(k-2). \quad (11)$$

Substituting (11) into (10) yields

$$u^*(k) = \begin{bmatrix} e^*(k) \\ u^*(k-1) \\ 1.218u^*(k-1) + 0.892y^*(k-1) - 0.886y^*(k-2) \end{bmatrix}^{\mathrm{T}} \times \begin{bmatrix} 4.75\times10^{-3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 260 \end{bmatrix} \begin{bmatrix} k_1 \\ 1 \\ k_2 \end{bmatrix}. \quad (12)$$

It is now easy to substitute the actual machine data $u(k)$ and $y(k)$ for $u^*(k)$ and $y^*(k)$ in (12) to identify the gain-parameter vector ψ. Note that the data used for the identification process must still satisfy the persistence excitation condition to ensure that $\Phi(N)$ is a full-rank matrix.

4. Data validation

Table 1 lists the PDF-controller parameter-identification results calculated from four mutually independent sets of sufficiently rich data. The obtained values of gains k_1 and k_2 in the PDF controller showed no apparent differences. Therefore, we used the averaged values of $k_1 = 6.942\times10^{-2}$ and $k_2 = 2.359\times10^{-4}$ for the combined experimental and solid-modeling identification procedure. The identification results were then verified by comparing the solid-modeling simulations with the experimental data under two independent operating conditions, as shown in Figs. 6 and 7. The operation condition for Fig. 6 was a position reference of 0.02 m and a feed rate of 8 m/min. The position and velocity responses of the actual machine and the simulation were very close, with only slight differences appearing in the thrust response. The operation condition for Fig. 7 was a position reference of 0.03 m and a feed rate of 16 m/min. The responses of the actual machine and the simulation were again very close. These results indicate that the controller gain identified using the method described in Section 3 can be used for different feed rates and position commands.

Table 1
PDF-controller parameter-identification results

Stroke		Feed rate (m/min)		
		4	8	16
0.02 m	k_1	6.858×10^{-2}		
	k_2	2.423×10^{-4}		
0.03 m	k_1	6.858×10^{-2}	6.937×10^{-2}	
	k_2	2.346×10^{-4}	2.308×10^{-4}	
0.04 m	k_1		6.884×10^{-2}	7.037×10^{-2}
	k_2		2.269×10^{-4}	2.423×10^{-4}
0.05 m	k_1			7.079×10^{-2}
	k_2			2.385×10^{-4}

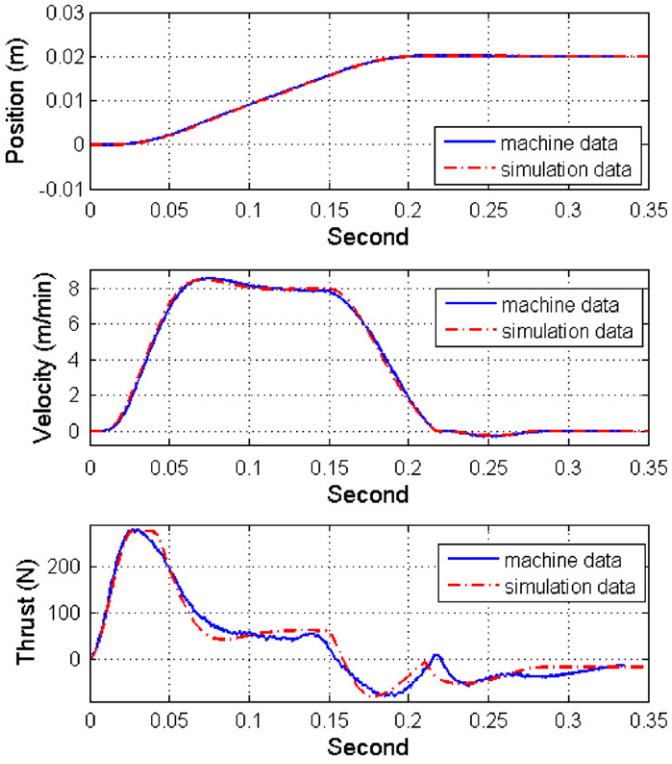

Fig. 6. Comparisons of machine responses for a position reference of 0.02 m and a feed rate of 8 m/min.

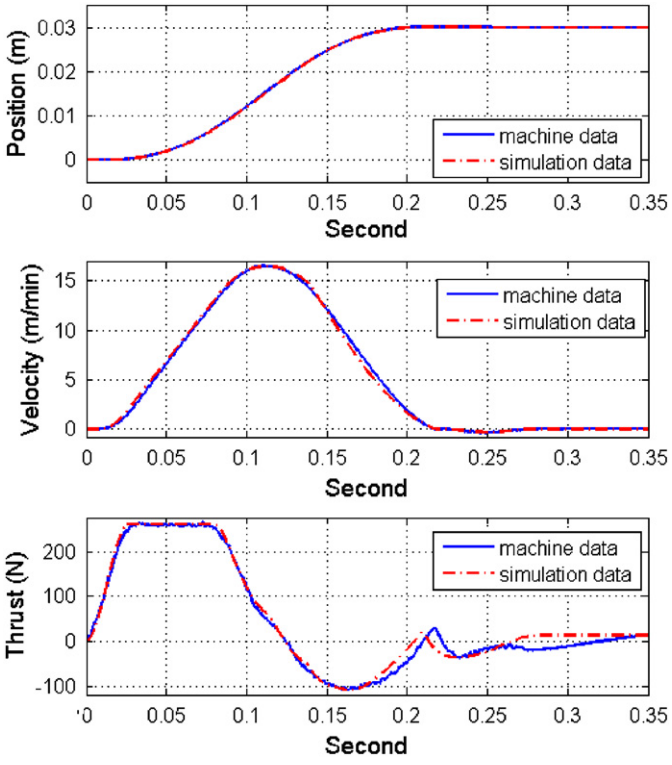

Fig. 7. Comparisons of machine responses for a position reference of 0.03 m and a feed rate of 16 m/min.

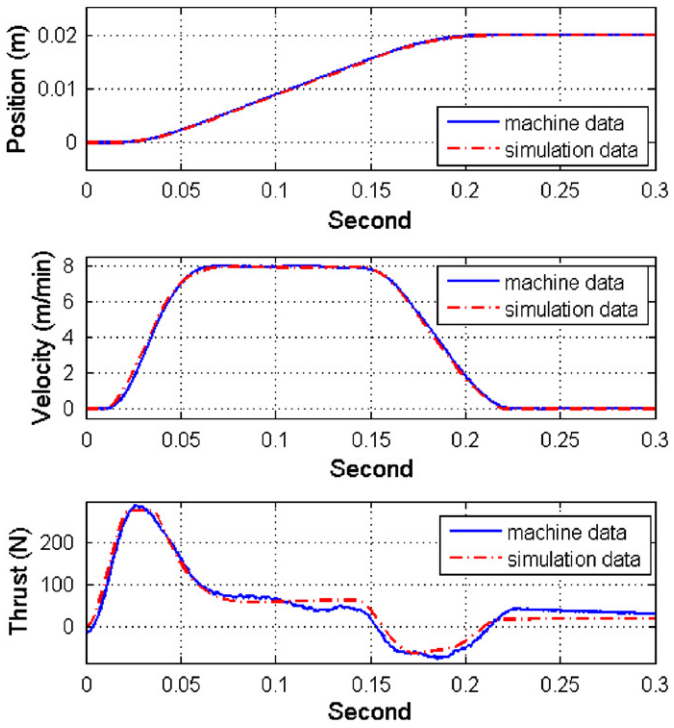

Fig. 8. Comparisons of machine responses with modified control parameters for a position reference of 0.02 m and a feed rate of 8 m/min.

5. Servo performance improvement

After including the base vibration modes discussed above and recalculating the appropriate gains for the controller, the solid-modeling simulation closely represented the actual machine behavior. To improve the servo performance, the controller was tuned within the solid-modeling simulation environment before its actual implementation. Fig. 6 shows that there was a velocity overshoot in the accelerating phase in both the simulation and experimental data. There was also an overshoot in the position response that caused the worktable to move in the opposite direction. This also resulted in a small negative velocity appearing at the end of the velocity response. We first eliminated the overshoots in the velocity and position responses in the simulation by adjusting the controller parameters from 19 and 260 to 22 and 220, respectively, with these modified parameters implemented in the actual machine. Fig. 8 compares the simulations and the actual machine data after the parameter modifications, and comparison with Fig. 6 clearly indicates that the position and velocity overshoots were absent in the actual machine.

6. Conclusions

This paper addresses the practical implementation of a servo design from within a virtual simulation environment. The study first derived a gain-parameter vector to represent the difference between the numerical values in the virtual environment and the actual implementation parameters, which made it straightforward to identify the unknown gain-parameter vector using popular parameter-identification algorithms. Comparisons between simulation data and actual measurements were used to validate the estimations of the gain parameters. The results indicated that this is a very accurate solid-modeling tool that enables direct servo tuning within a virtual environment, with the directly implemented virtual tuning results improving the performance of the actual machine. The future use of a realistic dynamic model would also make it possible to accurately estimate nonlinear forces such as friction and dynamic-imbalance forces. The proposed controller parameter-identification process and solid-modeling procedure could also be used on other types of machines based on a PDF controller or other types of controller.

Acknowledgments

This work is supported by the Chung-Shan Institute of Science and Technology under Contract no. 91-SA59, and in part by the National Science Council under Project no. 95-2221-E-002-435-MY3. Thanks to Chevalier Machinery Inc., Taichung, Taiwan for supporting the design data and the test machine.

References

[1] MSC.visualNastran Desktop user's manual, MSC.visualNastran Desktop™, MSC Software, 2001.
[2] N.J. Salamon, Computer generated shapes in mechanical design with MATLAB, International Journal of Engineering Education 21 (5) (2005) 915–924.
[3] R.J. Lee, K.C. Chou, S.H. Liu, J.Y. Yen, Solid modeling based servo system design for a high speed micro grinding machine, International Journal of Machine Tools and Manufacture 46 (2) (2006) 208–217.
[4] P.N. Paraskevopoulos, G.D. Pasgianos, K.G. Arvanitis, New tuning and identification methods for unstable first order plus dead-time processes based on pseudoderivative feedback control, IEEE Transactions on Control Systems Technology 12 (3) (2004) 455–464.
[5] D.M. Alter, T.C. Tsao, Dynamic stiffness enhancement of direct linear motor feed drives for machining, in: Proceedings of the 1994 American Control Conference, 1994, vol. 3, pp. 3303–3307.
[6] B.K. Choi, C.H. Choi, H. Lim, Model-based disturbance attenuation for CNC machining centers in cutting process, IEEE/ASME Transactions on Mechatronics 4 (2) (1999) 157–168.
[7] J.F. Cuttino, A.C. Miller Jr., D.E. Schinstock, Performance optimization of a fast tool servo for single-point diamond turning machines, IEEE/ASME Transactions on Mechatronics 4 (2) (1999) 169–179.
[8] A. Arakawa, K. Miyata, Simultaneous optimization algorithm for determining both mechanical-system and controller parameters for positioning control mechanisms, in: Proceedings of the 1996 Fourth International Workshop on Advanced Motion Control, AMC'96, 18–21 March 1996, vol. 2, pp. 625–630.
[9] B.X. Xiao, M. Xia, C.M. Zhao, The main control mode and fuzzy control strategy of CNC system for gear hobbing and grinding machine, in: Proceedings of the IEEE International Conference on Industrial Technology, 1996, pp. 643–646.
[10] J. Liu, K. Yamazaki, Y. Yokoyama, Dynamic gain motion control with multi-axis trajectory monitoring for machine tool systems,

in: Proceedings of the 1998 International Workshop on Advanced Motion Control, AMC'98, 1998, pp. 316–321.

[11] R.J. Fornaro, T.A. Dow, A high-performance machine tool controller, in: Conference Record of the 1988 IEEE Industry Applications Society Annual Meeting, 1988, vol. 2, pp. 1429–1439.

[12] Y.S. Tarng, H.Y. Chuang, W.T. Hsu, Intelligent cross-coupled fuzzy feedrate controller design for CNC machine tools based on genetic algorithms, International Journal of Machine Tools and Manufacture 39 (10) (1999) 1673–1692.

[13] G. Younkin, Modeling machine tool feed servo drives using simulation techniques to predict performance, in: Conference Record of the Industry Applications Society Annual Meeting, 1989, vol. 2, pp. 1699–1706.

[14] D.Y. Ohm, Analysis of PID and PDF compensators for motion control systems, in: Conference Record—IAS Annual Meeting, 1994, vol. 3, IEEE Industry Applications Society, pp. 1923–1929.

[15] S.K. Bag, S.K. Spurgeon, C. Edwards, Output feedback sliding mode design for linear uncertain systems, IEE Proceedings—Control Theory and Applications 144 (3) (1997) 209–216.

[16] W. Wang, S.H. Kweon, I. Kim, S.H. Yang, Structural modeling and simulation of miniaturized machine tool: 3-axis vertical micro-end-milling system, International Journal of Modern Physics B 20 (25–27) (2006) 3811–3816.

[17] T.J. Cheng, A high speed machine tool servo design—a system approach, Master's Thesis, Department of Mechanical Engineering, National Taiwan University, 2000.