

# Design and Implementation of a High Precision Servo System

Hyun Sik Ahn\*, Sang-Rok Oh\*, Ick Choy\*, Kwang Bae Kim\* and Myoung Sam Ko\*\*

\* Control Systems Laboratory, Korea Institute of Science and Technology, Seoul, Korea

\*\* Department of Control and Instrumentation, Seoul National University, Seoul, Korea

## ABSTRACT

*A novel type of a play-back servo system with high precision is designed using an iterative learning control method by employing the model algorithmic control concept together with an inverse model. A sufficient condition is also provided for the convergency. It is shown by simulation that the proposed control algorithm yields a good performance even in the presence of a periodic load disturbance and proved by experiments using microprocessor-based play-back servo system.*

## 1. Introduction

When mechanical systems such as industrial robots and numerically controlled machines are used for path-dependent jobs such as arc-welding or sealing, the systems should be capable of tracking as precisely as possible the given continuous-path. However, since the mechanical system model often suffers from large parameter uncertainties due to loading effects and difficulty of modeling non-linearity, conventional control algorithm such as PI or PID control may be unsatisfactory.

In fact, continuous-path control of non-linear robot manipulators with uncertainty is a difficult task, and as an efficient means of control, there have been proposed a class of novel methods called iterative learning control.[1],[2] The main idea of iterative learning control method is to apply a simple algorithm repetitively to an unknown plant until perfect tracking is achieved. The algorithms show uniform tracking performance and the possibility of application to a play-back servo system for industrial robots and/or NC machines.

In this paper, a novel type of a high precision play-back servo system is designed by using a conventional PI controller together with an iterative learning controller which generates a virtual reference input for the PI-controlled system. And it is shown by experimental results that the output trajectory converges to the desired trajectory as iteration increase.

## 2. A play-back servo system using an iterative learning control method

Consider a play-back servo system using PMSM (Permanent Magnet Synchronous Motor) in which "field-orientation" control is performed. In other words, both magnitude and phase of stator currents are controlled instantaneously through some current regulated inverter, according to the values generated by rotor-to-stator transformation using the absolute position of permanent magnet. Then, the electrical dynamics of PMSM can be ignored, and PMSM dynamics is approximated as in

$$J\dot{\omega} + D\omega + T_L = K_T i_q \quad (1)$$

where  $J$  : moment of inertia       $D$  : friction coefficient  
 $T_L$  : load torque                   $K_T$  : torque constant  
 $\omega$  : mechanical speed of the rotor  
 $i_q$  : torque component of current.

Suppose we wish to determine a control input such that the output velocity  $\omega(t)$  tracks a given desired velocity  $\omega^d(t)$ ,  $t \in [0, T]$  within a specified  $\epsilon$ -bound, i.e.,

$$\|\omega^d(t) - \omega(t)\|_{\infty} \leq \epsilon, \quad t \in [0, T] \quad (2)$$

where  $\|\cdot\|_{\infty}$  denotes a sup-norm. As a means of designing a controller for the system in equation (1) to satisfy the constraint in equation (2), the conventional PI(Proportional and Integral) controller is very often used. But if there exist modelling errors and unmodelled load disturbances, it is very difficult to find the proper gain constants, furthermore the PI-controlled PMSM drive system can obtain only asymptotic convergency but cannot guarantee that the system output tracks the desired output satisfying the given tolerance error bound all along the trajectory.

To overcome the shortcomings, an iterative learning controller together with a conventional PI controller is proposed for the PMSM drive system as shown in Fig. 1.

In Fig. 1, the control input  $u_k$  generated by learning controller at the  $k$ -th trial can be thought of as a virtual reference input for perfect tracking, and from the view point of configuration,

Now we propose an iterative learning control algorithm for the learning controller in Fig. 1 as follows. From equation (1), the dynamics of the PI-controlled PMSM drive system can be written as

where  $x = [\omega, z]^T$ ,  $z(t) = \int_0^t (\omega^d(s) - \omega(s)) ds$

$$C = [1, 0], \quad w(t) = \begin{bmatrix} -\frac{T_L(t)}{J} & 0 \end{bmatrix}^T$$

$K_I$  : Integral gain constant of the PI-controller.

$$\dot{x}^M(t) = A^M x^M(t) + B^M u(t) \quad (4)$$
$$e_k(t) = \omega^d(t) - \omega(t). \quad (5)$$

the solution input, otherwise, generate the next trial input  $u_{k+1}(t)$  by the formula

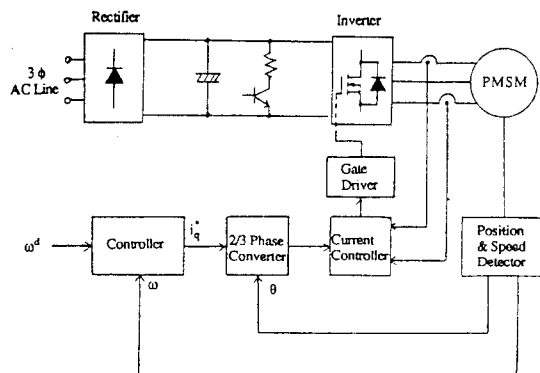
where

and  $S$  is an  $n \times p$  weighting matrix and  $B^{M+}$  is the generalized inverse of  $B^M$ . Then, for the proposed play-back servo system, the following theorem holds :

$$x_k(0) = x^d(0) = x(0) = \zeta^0, \quad k = 0, 1, 2, \dots \quad (8)$$
$$\|I - B^{M^+} S C B\|_\infty < 1 \quad (9)$$
$$\lim_{k \rightarrow \infty} \|e_k\|_{\infty} = 0 \quad (10)$$

### 3. Simulation and Experimental Results

The block diagram of the system configuration is shown in Fig.2. It consists of a 4-pole IHP PMSM, a current regulated PWM MOSFET inverter, and a microprocessor-based control system.



1416

The PMSM is outfitted with a resolver to provide the absolute rotor position feedback, and Hall-effect current sensors to provide phase current feedback to close the phase current regulation loops using a 3-independent hysteresis current controller.

The principal control functions are performed by a real-time, interrupt-driven control algorithm, and implemented by using a 32-bit single board microcomputer (MVME 133A-20). Fig. 3 shows the configuration of the experimental system. The microcomputer-based control system inputs the desired speed trajectory, the control parameters such as the poles of PI-controlled system, weighting matrix, sampling time, and generates the q-axis current reference by the proposed scheme. The d-axis current reference is forced to zero as is often the case in vector control of PMSM.

All of the programs are written by C language in IBM PC/AT and compiled by Microtec/Paragon MCC68K C compiler and down-loaded into MVME 133A-20 microcomputer via RS-232C serial port.

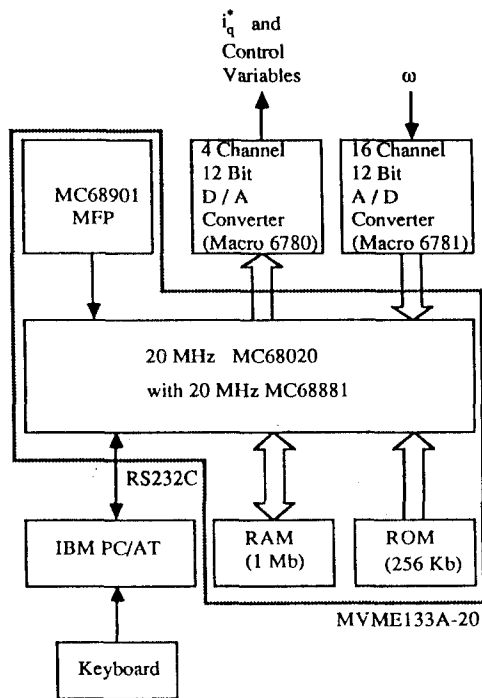


Fig.3 Configuration of microcomputer system

Table 1 and Table 2 show the specifications of the PMSM and the conditions of simulation and experiment, respectively.

Table 1 Specifications of the PMSM

|   |                            |
|---|----------------------------|
| Rated torque                                | 2.6 [Nm]                   |
| Rated current                               | 3.6 [A]                    |
| Rated speed                                 | 3000 [rpm]                 |
| Stator resistance                           | 2.14 [ $\Omega$ ]          |
| Stator inductance                           | 4.2 [mH]                   |
| Stator flux linkage due to permanent magnet | 0.17 [wb]                  |
| Number of poles                             | 4 [Pole]                   |
| Moment of inertia                           | 0.00569[Nms <sup>2</sup> ] |
| Friction coefficient                        | 0.00339[Nms]               |

Table 2 Conditions of simulation and experiment

|   |   |
|---|---|
| Desired trajectory                              | Equation (11)                             |
| Modelled parameters                             | $J^M = 0.5 J$ , $B^M = B$ , $K_T^M = K_T$ |
| Desired poles locations of PI controlled system | $-30 + j 0$ , $-30 - j 0$                 |
| PI gain constants                               | $K_p = 5.007$ , $K_I = 0.3272$            |
| Initial control input                           | $u_0(t) = 0$ for $t \in [0, T]$           |
| Weighting matrix S                              | $S = [0.5 \ 0]^T$                         |
| Sampling time                                   | 500 [ $\mu$ sec]                          |
| Hysteresis band of current controller           | 0.2 [A]                                   |

To consider the effects of modelling errors,  $J^M$  which is the modelled value of  $J$  is set to be  $0.5J$  and the other parameters are assumed to be modelled accurately. The gain constants of PI-controller are calculated so that the roots of characteristic equation are to be located at  $-30$ ,  $-30$  for PI-controlled system to be critically damped, in this case  $K_p = 5.007$ ,  $K_I = 0.3272$ . The desired speed trajectory  $\omega^d(t)$ , the initial control input  $u_0(t)$  for  $t \in [0, 1.0]$  and weighting matrix  $S$  are selected as the following equations.

$$\omega^d(t) = \begin{cases} -32400 t^2 (t - \frac{1}{2}) & [\text{rpm}], 0 \leq t \leq 0.5 \\ 32400 (t - \frac{1}{2})^2 (-t + \frac{1}{2}) & [\text{rpm}], 0.5 \leq t \leq 1.0 \end{cases} \quad (11)$$

$$u_0(t) = 0, \quad t \in [0, 1.0] \quad (12)$$

$$S = [0.5 \quad 0]^T \quad (13)$$

The hysteresis band of 3-independent hysteresis current controller in CRPWM inverter and the sampling time is set to be 0.2[amp] and 500[μs], respectively.

Fig.4 and Fig.5 show the simulation and experimental results under these conditions and Table 3 shows the maximum norm of speed error according to iteration index  $k$  in both cases of simulation and experiment. In simulation the maximum norm of speed error converges within 10 % of maximum desired speed at  $k = 3$ , but in experiment it converges at  $k=5$  and there still exists a little remaining error even after several more iterations. This seems to come from the facts that friction torque varies as the function of rotor angular position due to the mechanical coupling effect, and that there exists nonperiodic torque ripple due to the unmodelled dynamics of PMSM such as slot effect and the current harmonics generated by CRPWM inverter.

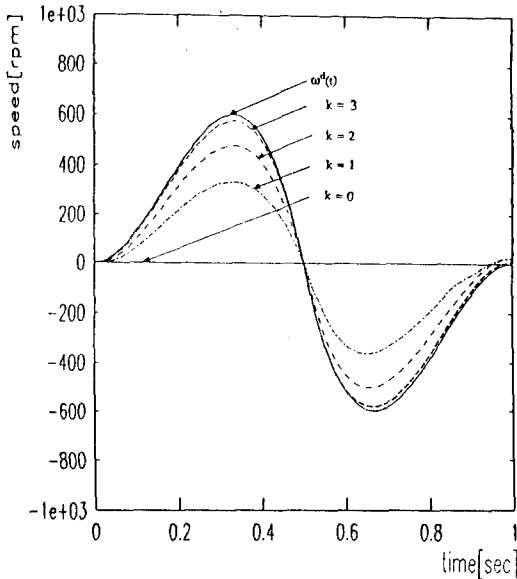
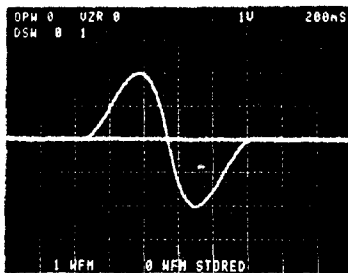
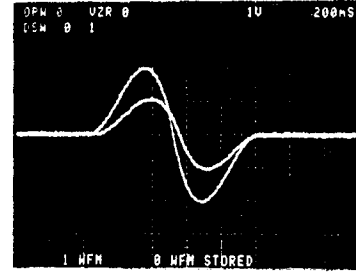


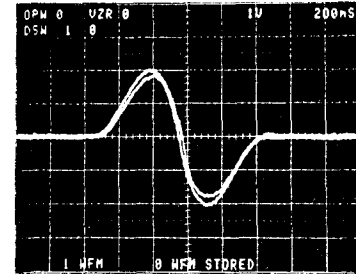
Fig. 4 Simulation results



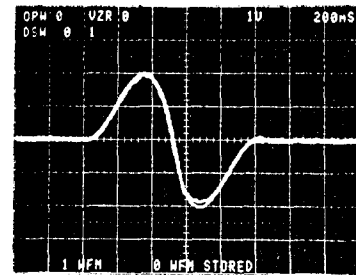
(a)  $k = 0$  ( x-axis : 200 ms/div, y-axis : 300 rpm/div )



(b)  $k = 1$  ( x-axis : 200 ms/div, y-axis : 300 rpm/div )



(c)  $k = 3$  ( x-axis : 200 ms/div, y-axis : 300 rpm/div )



(d)  $k = 5$  ( x-axis : 200 ms/div, y-axis : 300 rpm/div )

Fig. 5. Experimental results

Table 3 Maximum norm of output error

| Iteration $k$ | ( rpm )    |            |
|---------------|------------|------------|
|               | Simulation | Experiment |
| 0             | 600        | 600        |
| 1             | 269        | 303        |
| 2             | 121        | 186        |
| 3             | 54         | 130        |
| 4             | 24         | 83         |
| 5             | 11         | 55         |

#### 4. Conclusion

A novel type of a high precision play-back servo system is designed by using an iterative learning controller together with a conventional PI controller. Via computer simulations firstly, the proposed play-back servo system using a PMSM drive system has shown its tracking capability of the desired trajectory, even in the presence of a periodic load disturbance. And next, the proposed play-back servo system was implemented and tested in real-time using a microprocessor-based control system, and the experimental results may show that its applicability to real industrial tasks.

#### References

- [1] S. Arimoto, " Robustness of Learning Control for Robot Manipulators," *IEEE Conf. on Robotics and Automation*, pp. 1528-1533, 1990
- [2] S.-R. Oh, Z. Bien, and I.H. Suh," A Model Algorithmic Learning Method for Continuous-Path Control of a Robot Manipulator," *ROBOTICA*, Vol. 8, 1990
- [3] K. B. Kim, S.-R. Oh, H. S. Ahn and M. S. Ko," An Iterative Learning Control Method for Play-Back Servo System," *IEEE International Workshop on Advanced Motion Control*, Yokohama, pp. 82- 89, Mar.1990
- [4] D.W. Novotny and R.D. Lorenz, *Introduction to Field Orientation and High Performance AC Drives*, IEEE Industry Applications Society, 1985
- [5] MVME133A-20 VME module 32-bit Monoboard Micro-computer User's Manual, 1st ed., Motorola Inc., 1987