Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Can a Two-Link Manipulator Learn how to Write?

Basel Alali [*] **Kentaro Hirata** [*] **Kenji Sugimoto** [*]

[*] *Nara Institute of Science and Technology (NAIST), 8916-5 Takayama-cho, Ikoma City, Nara 630-0192, Japan. Fax: +81 743 72 5359; emails: basel-a@is.naist.jp, kent@is.naist.jp and kenji@is.naist.jp*

Abstract: This paper considers a version of the problem of how to teach robots to write characters in an actual environment. In particular, a feedforward controller is designed for two-link manipulators to improve tracking performance despite limited knowledge of the surroundings. An adaptive scheme, called MIMO-FEL (Multi-input Multi-output Feedback Error Learning), is employed to achieve the objective. After convergence, the feedforward controllers are switched depending on the target character to be written. This switching strategy is a clear contrast with the precise identification approach, which uses a single general-purpose controller.

Keywords: Feedback Error Learning, MIMO Systems, Two-link Manipulator.

1. INTRODUCTION

High tracking performance is one of the most important requirements for robotics applications. To design a model-based feedforward controller with good tracking performance, an accurate model of the process is needed. However, factors such as uncertainty, nonlinearity or time-varying behavior make modeling and identification more difficult or expensive. To overcome this challenge, several adaptive and learning control techniques have been proposed; see Tao (2003), de Vries (2000), Colbaugh *et al.* (1994), Dawson (1991), Li and Slotine (1989). There are in general two distinct adaptive control approaches. The first approach is called indirect adaptive control because the adaptive laws provide explicit estimates of the dynamics of the model parameter, which is then used in controller design ; Li and Slotine (1989). The second is called direct adaptive control, as the adaptive laws adjust the control gains directly without parameter estimation; Colbaugh *et al.* (1994). These approaches have also been used to adjust the feedforward controller to obtain an accurate inverse model of the plant as in Sun and Tsao (1999). Adaptive inversion, which first needs to estimate a plant model, is less sensitive to plant uncertainties and variations but also adjusts itself to plant parameter changes.

However, two powerful model-free learning control methods, iterative learning control (ILC) and feedback error learning (FEL), have attracted much attention in the last two decades (see, *e.g.*, Bein and Xu (1998), Sugie and Ono (1991), Sugie (2004), Kawato *et al.* (1987), Miyamura and Kimura (2002), Muramatsu and Watanabe (2004)). ILC deals with repeating tracking tasks in a finite time interval. Thus, it yields the desired input through the iteration of trials with the reset action of initial conditions. FEL, proposed by Kawato *et al.* (1987), achieves an inverse model of the plant without extensive modeling by utilizing the error signal during continuous-time closed-loop operation. The key point of FEL is to use a learning law which depends on the feedback error in order to tune the feedforward controller parameters.

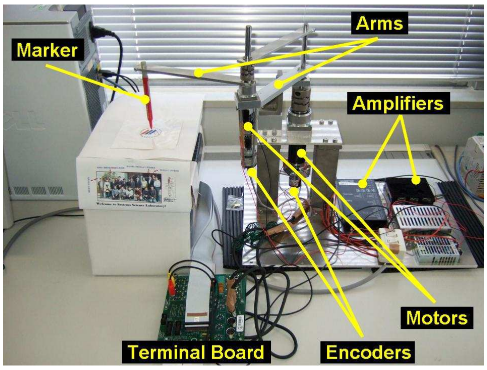

Fig. 1. Experimental Hardware

Miyamura and Kimura (2002) have established a control theoretical validity of the FEL method in the frame of adaptive control for the SISO case, proving its stability based on strictly positive realness, whereas Muramatsu and Watanabe (2004) have relaxed the positive realness condition of FEL. Following Miyamura and Kimura (2002), Alali *et al.* (2006a,b) have studied some generalizations of the FEL scheme.

This work studies an application of MIMO-FEL technique developed in Alali *et al.* (2006a) to a practical problem in terms of two-link manipulators. The basic idea of this work is to achieve an approximated inverse of the plant adaptively, using linear parameterization instead of Artificial Neural Networks (ANN) used in Kawato *et al.* (1988) and Ruan (2007), to improve tracking performance for each specific desired trajectory and also the speed of parameter convergence by means of MIMO-FEL. This also contrasts with achieving an exact inverse via precise system identification, which requires a huge amount of data and richness of the excitation input. Thus, by FEL, one can obtain an

978-1-1234-7890-2/08/$20.00 © 2008 IFAC 10.3182/20080706-5-KR-1001.0217

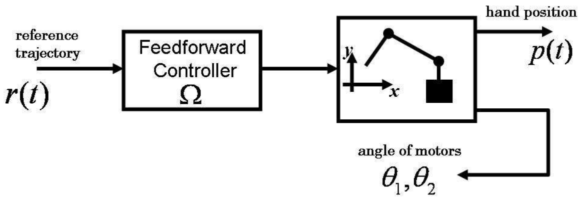

Fig. 2. Feedforward Controller Design

inverse model for a specific reference signal with a limited amount of data and a limited range of frequency components. In practice, the feedforward controllers are switched depending on the target character to be written. This is a clear contrast with the precise identification approach, which uses a single general-purpose controller.

The basic assumption made in Alali *et al.* (2006a) to prove the convergence of the proposed algorithm is that the plant is linear. Since the two-link manipulator is a typical nonlinear system, we have to fill this gap. There are basically two sources of the nonlinearity: the first one is caused by the friction force generated by the reduction gear of the motor and the time-varying inertia. The second one is caused by the trigonometric dependency of the angle of the motors to the X-Y coordinates of the hand position. We overcome the first one by means of high-gain local feedback, the second by restricting the working area of the hand position to within a small neighborhood around the equilibrium point.

2. DYNAMICS OF TWO-LINK MANIPULATOR

We consider a two-link manipulator in Fig. 1. Each arm is driven by a DC motor with a reduction gear. The derivers apply the current to each motor in proportion to their input voltages. The objective here is to design a feedforward controller Ω for this manipulator system which achieves a good tracking performance, namely $\|p(t) - r(t)\| \to 0$ as $t \to \infty$ as illustrated in Fig. 2. As mentioned above, local angular velocity and angular feedback based on the encoder signal with relatively high gains are applied, as in Fig. 3. Thus, the nonlinearities due to friction or time-varying inertia are compensated. Assume that the behavior of the dashed line block in Fig. 3 is much improved by the inner PI loop, *i.e.*, $G(s) \simeq 1$. Then, the dynamics from input voltage v_i to angle θ_i can be approximated as

$$\theta_i = \frac{k_i}{\tau_i s + 1} v_i, \tag{1}$$

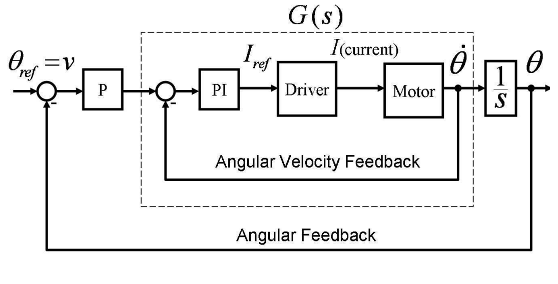

Fig. 3. Local Feedback Structure

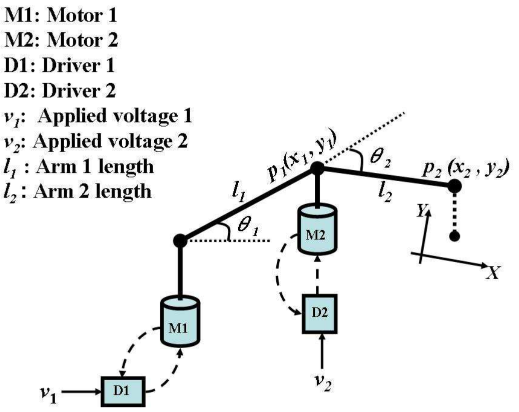

Fig. 4. Two-Link Manipulator Model

where τ_i and k_i are parameters that need to be identified. The x-y coordinates of the two points p_1 and p_2 in Fig. 4 are given as follows:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} l_1 \cos\theta_1 \\ l_1 \sin\theta_1 \end{bmatrix}, \tag{2}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 \cos\theta_1 + l_2 \cos(\theta_2 - \theta_1) \\ l_1 \sin\theta_1 - l_2 \sin(\theta_2 - \theta_1) \end{bmatrix}. \tag{3}$$

Let

$$\begin{aligned} \theta_1 &= \theta_1^* + \Delta\theta_1, \\ \theta_2 &= \theta_2^* + \Delta\theta_2, \end{aligned} \tag{4}$$

where θ_1^* and θ_2^* are initial angles of the motor corresponding to the equilibrium point of the hand position. To overcome the nonlinearity effects caused by the trigonometric functions in (3), we restrict the working area of the hand position to within a small neighborhood around the equilibrium point. Then, the motion around (θ_1^*, θ_2^*) can be approximated by linear dynamics using Taylor series expansion. Therefore, the resulting hand position $p_2(x_2, y_2)$ is given as follows:

$$\begin{aligned} x_2 &= x_2^* + \left\{ l_2 \sin(\theta_2^* - \theta_1^*) - l_1 \sin\theta_1^* \right\} \Delta\theta_1 \\ &\qquad - l_2 \sin(\theta_2^* - \theta_1^*) \Delta\theta_2, \\ y_2 &= y_2^* + \left\{ l_2 \cos(\theta_2^* - \theta_1^*) - l_1 \cos\theta_1^* \right\} \Delta\theta_1 \\ &\qquad - l_2 \cos(\theta_2^* - \theta_1^*) \Delta\theta_2, \end{aligned} \tag{5}$$

The coordinates x_2^* and y_2^* are given as

$$\begin{aligned} x_2^* &= l_1 \cos\theta_1^* + l_2 \cos(\theta_2^* - \theta_1^*), \\ y_2^* &= l_1 \sin\theta_1^* - l_2 \sin(\theta_2^* - \theta_1^*). \end{aligned} \tag{6}$$

Note that (x_2^*, y_2^*) is the equilibrium point of linearization. Further, the following relationship is obtained from (1)

$$\Delta\theta_i = \frac{k_i}{\tau_i s + 1} \Delta v_i, \tag{7}$$

where Δv_i denotes the deviation from the initial voltage. As a result, if we consider the motion in a small neighborhood of (x_2^*, y_2^*), then we can approximate the hand motion with the following linearized model as

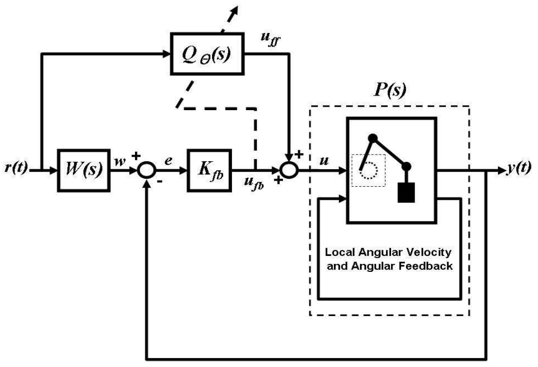

Fig. 5. Control System Scheme

$$\begin{aligned}\begin{bmatrix}\Delta x_2\\ \Delta y_2\end{bmatrix} &= M_1(\theta_1^*, \theta_2^*)M_2(s)\begin{bmatrix}\Delta v_1\\ \Delta v_2\end{bmatrix},\\ &=: P(s)\begin{bmatrix}\Delta v_1\\ \Delta v_2\end{bmatrix}, \end{aligned} \tag{8}$$

where

$$\begin{aligned}\Delta x_2 &= x_2 - x_2^*,\\ \Delta y_2 &= y_2 - y_2^*,\end{aligned} \tag{9}$$

$$M_1(\theta_1^*, \theta_2^*) = \begin{bmatrix} l_2\sin(\theta_2^*-\theta_1^*) - l_1\sin\theta_1^* - l_2\sin(\theta_2^*-\theta_1^*)\\ l_2\cos(\theta_2^*-\theta_1^*) + l_1\cos\theta_1^* - l_2\cos(\theta_2^*-\theta_1^*)\end{bmatrix}, \tag{10}$$

$$M_2(s) = \begin{bmatrix}\dfrac{k_1}{\tau_1 s+1} & 0\\ 0 & \dfrac{k_2}{\tau_2 s+1}\end{bmatrix}. \tag{11}$$

The theoretical contribution in Alali *et al.* (2006a) is twofold: extension of FEL to MIMO and to strictly proper systems. The target system (8) is exactly what should be considered in this framework and we may apply the linear MIMO-FEL technique to this plant.

3. CONTROLLER DESIGN

In this section, we briefly summarize our MIMO-FEL algorithm developed in Alali *et al.* (2006a). The block diagram of the control system is shown in Fig. 5. The objective of our controller design is to minimize the error signal between $w(t)$ and the plant output $y(t)$. The following equations represent the interconnection in the figure:

$$\begin{aligned} y(t) &= P(s)u(t)\\ u(t) &= u_{\mathrm{fb}}(t) + u_{\mathrm{ff}}(t)\\ u_{\mathrm{fb}}(t) &= K_{\mathrm{fb}}e(t)\\ e(t) &= w(t) - y(t)\\ w(t) &= W(s)r(t)\\ u_{\mathrm{ff}}(t) &= Q_\Theta(s)r(t). \end{aligned} \tag{12}$$

Let $P(s)$ be a MIMO strictly proper plant. K_{fb} is a feedback gain to stabilize the plant, and Θ is a tunable parameter. The first equation implies

$$y(t) = \pounds^{-1}[P(s)](t) * u(t),$$

where $*$ denotes the time-domain convolution. For simplicity, we will adopt this kind of slight abuse of the notation throughout this paper. We first note that, if the system is invertible with properness as in Miyamura and Kimura (2002); Muramatsu and Watanabe (2004), then we can readily take $Q_\Theta = P^{-1}$ and obtain a perfect tracking. This is not the case, however, since $P(s)$ is assumed to be strictly proper, and hence $P^{-1}(s)$ becomes improper. To overcome this difficulty, pre-filter $W(s)$ is introduced. Under a mild assumption on the plant, one can set the pre-filter to a diagonal form

$$W(s) = \begin{bmatrix}\dfrac{1}{(s+a_1)^{\mu_1}} & & 0\\ & \ddots & \\ 0 & & \dfrac{1}{(s+a_m)^{\mu_m}}\end{bmatrix}, \tag{13}$$

where a_k is an arbitrarily chosen positive real number and the integers μ_1, μ_2, $\cdots$, μ_m can be regarded as a generalization of the relative degree to MIMO case.

We now construct a tunable feedforward controller. To generate $u_{\mathrm{ff}}(t)$, we consider the following dynamical system:

$$\begin{aligned} \dot{\xi_1}(t) &= A_f\xi_1(t) + B_f r(t)\\ \dot{\xi_2}(t) &= A_f\xi_2(t) + B_f u_{\mathrm{ff}}(t)\\ u_{\mathrm{ff}}(t) &= F(t)\xi_1(t) + G(t)\xi_2(t) + H(t)r(t)\\ &= \Theta(t)\xi(t), \end{aligned} \tag{14}$$

where

$$\Theta(t) = [F(t)\ \ G(t)\ \ H(t)],\ \ \xi(t) = \begin{bmatrix}\xi_1(t)\\ \xi_2(t)\\ r(t)\end{bmatrix}. \tag{15}$$

Take A_f and B_f in a controllable canonical form as shown in (16) and (17),

$$A_f = \begin{bmatrix} 0 & 1 & & 0 & & & & \\ \vdots & & \ddots & & & & 0 & \\ & & \cdots & 1 & & & & \\ -f_\mu & & & -f_1 & & & & \\ & & & & \ddots & & & \\ & & & & & 0 & 1 & 0\\ & & 0 & & & \vdots & \ddots & \\ & & & & & & \cdots & 1\\ & & & & & -f_\mu & & -f_1 \end{bmatrix}, \tag{16}$$

$$B_f = \begin{bmatrix} 0 & \\ \vdots & 0\\ 1 & \\ & \ddots\\ & 0\\ 0 & \vdots\\ & 1 \end{bmatrix}, \tag{17}$$

where A_f is a stable matrix. Then, we have

$$\begin{aligned} u_{\mathrm{ff}}(t) &= \Big[I - G(t)(sI - A_f)^{-1}B_f\Big]^{-1} \\ &\quad \cdot\Big\{H(t) + F(t)(sI - A_f)^{-1}B_f\Big\}r(t) \\ &=: Q_\Theta(s)r(t). \end{aligned} \tag{18}$$

Note that the unknown parametric matrices $F(t)$, $G(t)$, $H(t)$ enter linearly into (14). The matrix $\Theta(t)$ is tuned using the following learning law

$$\frac{d\Theta}{dt} = \alpha u_{\mathrm{fb}}(t)\xi^T(t). \tag{19}$$

The derivation and stability analysis of the learning law have been studied in detail in Alali *et al.* (2006a). If $\Theta(t)$ converges to some constant value Θ_c which corresponds to F_c, G_c, H_c and $e(t) \to 0$ as $t \to \infty$, then we can obtain from (18) the following learning feedforward controller

$$\begin{aligned} u_{\mathrm{ff}_{\mathrm{Learning}}}(t) &= \Big[I - G_c(sI - A_f)^{-1}B_f\Big]^{-1} \\ &\quad \cdot\Big\{H_c + F_c(sI - A_f)^{-1}B_f\Big\}r(t) \\ &=: Q_{\Theta_c}(s)r(t). \end{aligned} \tag{20}$$

Clearly, the learning process for $Q_{\Theta_c}(s)$ depends on the reference signal $r(t)$. Thus, we employ a switching strategy in writing different characters since it means tracking to different references.

4. SIMULATION AND EXPERIMENTAL RESULTS

Using the on-line output $y(t)$ for learning may be the ultimate goal to demonstrate the practical usefulness of the proposed method. Toward this end, we first try a numerical simulation with nonlinear model (3). Since the nonlinear model includes the motor dynamics (1), we have to identify their parameters to perform the simulation under a realistic situation.

We use the following parameters for the length of the links and the equilibrium points in both simulation and experiment: $l_1 = 0.2$ [m], $l_2 = 0.2$ [m], $\theta_1^* = 30$ [deg], $\theta_2^* = 45$ [deg]. The working space is within 0.02 [m].

4.1 Identification of Motor Dynamics

As mentioned earlier, due to the double local feedback loops, the dynamics of each motor can be approximated as the first order systems. we then identified the parameters for each motor via curve fitting method as

$$\theta_1 = \frac{0.9982}{0.2422s + 1}v_1, \ \theta_2 = \frac{0.9956}{0.1458s + 1}v_1.$$

The responses are well approximated as shown in Figs. 6 and 7.

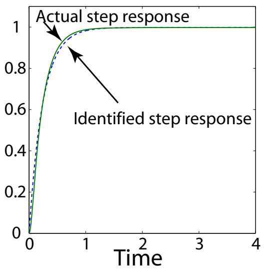

Fig. 6. Identification of Motor 1

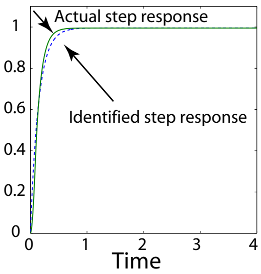

Fig. 7. Identification of Motor 2

4.2 Simulation Results

We first conduct a numerical simulation based on the nonlinear model (3) and linearized model (8). we choose the feedback gain and diagonal pre-filter as

$$K_{\mathrm{fb}} = \begin{bmatrix} -1 & 3 \\ -3 & -1 \end{bmatrix}, \quad W(s) = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix}.$$

We also need to set stable A_f and B_f which makes (A_f, B_f) controllable based on the upper bound of the relative degree $\mu = 1$, (see (16) and (17)). They are chosen as

$$A_f = \begin{bmatrix} -4 & 0 \\ 0 & -4 \end{bmatrix}, \quad B_f = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that the size of the matrix $\Theta(t)$ is 2 by 6. Thus, it includes 12 components. We consider the following two different reference trajectories to illustrate the idea clearly:

$$r_0(t) = \begin{bmatrix} \sin(0.4t) \\ \cos(0.4t) \end{bmatrix}, \quad r_8(t) = \begin{bmatrix} \sin(0.6t) \\ \sin(0.3t) \end{bmatrix}.$$

The reference trajectories represent the numerical numbers “0” and “8”, respectively. We then tune $\Theta(t)$ by using the learning rule (19).

1) Nonlinear model case:

We restrict the working area of the hand position to within a small neighborhood around the equilibrium point to guarantee the convergence of the learning law. The time evolution of $\Theta_0(t)$ and $\Theta_8(t)$ are shown in Figs. 8 and 9, respectively. Each line in the plot corresponds to the time evolution of one component of the matrix over the time period. It can be seen that the convergence has been achieved only after a long time. The simulation results of the trajectories of the hand for “0” and “8” are shown in Fig. 10. It shows that the actual trajectories are close to the reference trajectories. As a result, one can verify that the algorithm really works for a nonlinear model with a restricted moving area. However, the result also shows that it takes long time for convergence, even in the simulation. This implies that the experiment with on-line data via current algorithm is unrealistic. Thus, this direction of research will be considered in the future.

We now turn the attention toward the simulation with a linear model, as used in studies such as Miyamura and Kimura (2002) and Muramatsu and Watanabe (2004), to verify the proposed switching learning strategy in writing characters. The simulation result with the linear model is much faster than in the nonlinear case, as shown below.

2) Linear model case:

To solve the above problem of slow convergence, we use a linearized model of the plant, which can be obtained from (8) as follows:

$$P(s) = \begin{bmatrix} \dfrac{-0.04815}{0.2422s+1} & \dfrac{-0.05154}{0.1458s+1} \\ \dfrac{0.3657}{0.2422s+1} & \dfrac{-0.1923}{0.1458s+1} \end{bmatrix}. \qquad (21)$$

Based on this model, we perform another numerical simulation. The time evolution of $\Theta_0(t)$ and $\Theta_8(t)$ is shown in Figs. 11 and 12, respectively. It is clear how fast the convergence is, compared to the nonlinear case. The simulation results of the trajectories of the hand before and after learning are shown in Fig. 13. The resulting parameter matrices for each example from Figs. 11 and 12 are as follows:

a) for the first example ("0"):

$$\Theta_0 = \left[\begin{array}{cc|cc|cc} -0.22 & 0.28 & 0.88 & 1.26 & -0.77 & 1.38 \\ -0.57 & -0.27 & 0.87 & 2.86 & -2.37 & -0.97 \end{array}\right],$$

b) for the second example ("8"):

$$\Theta_8 = \left[\begin{array}{cc|cc|cc} -0.38 & 0.43 & 2.97 & 0.01 & -1.63 & 0.37 \\ -0.65 & -0.22 & 0.03 & 3.10 & -2.81 & -0.35 \end{array}\right],$$

and also $e(t) \to 0$ for both. Thus, the resulting learning feedforward controller for each case using (20) is given as follows:

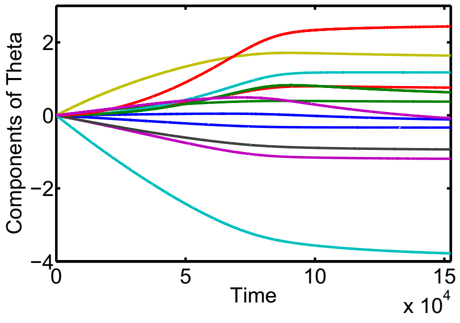

Fig. 8. Time Evolution of Θ_0: Nonlinear Case

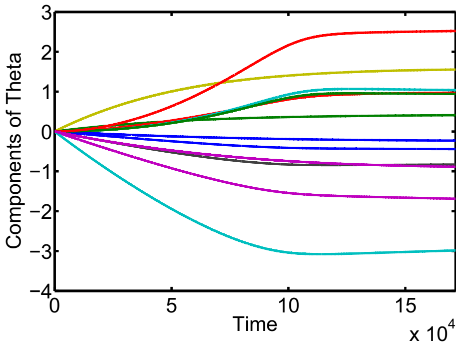

Fig. 9. Time Evolution of Θ_8: Nonlinear Case

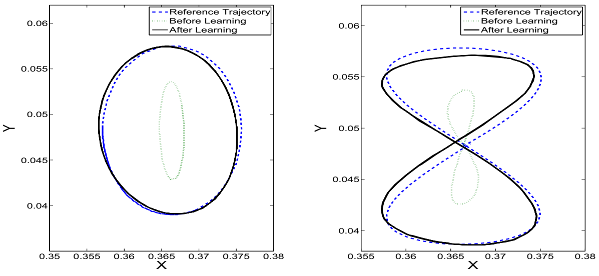

Fig. 10. Nonlinear Simulation Results for 0 and 8

$$Q_{\Theta_0}(s) =$$
$$\begin{bmatrix} \dfrac{-0.7721s^2 - 7.184s - 16.47}{s^2 + 4.257s + 2.456} & \dfrac{1.384s^2 + 6.162s + 1.355}{s^2 + 4.257s + 2.456} \\ \dfrac{-2.375s^2 - 18.14s - 34.25}{s^2 + 4.257s + 2.456} & \dfrac{-0.9749s^2 - 6.014s - 7.973}{s^2 + 4.257s + 2.456} \end{bmatrix},$$

$$Q_{\Theta_8}(s) =$$
$$\begin{bmatrix} \dfrac{-1.63s^2 - 8.405s - 6.354}{s^2 + 1.936s + 0.9322} & \dfrac{0.3725s^2 + 2.25s + 1.712}{s^2 + 1.936s + 0.9322} \\ \dfrac{-2.805s^2 - 14.82s - 12.47}{s^2 + 1.936s + 0.9322} & \dfrac{-0.3475s^2 - 1.954s - 1.606}{s^2 + 1.936s + 0.9322} \end{bmatrix}.$$

To confirm that the above learning controllers can really let the manipulator write its corresponding characters, we perform an experiment with the real manipulator.

4.3 Experimental Results

The experimental setup of the two-link manipulator is shown in Fig. 1. The target is to let the manipulator write the numerical numbers 0, 2, 3,···, 9. The first step is to obtain the learning controller for each character, as we

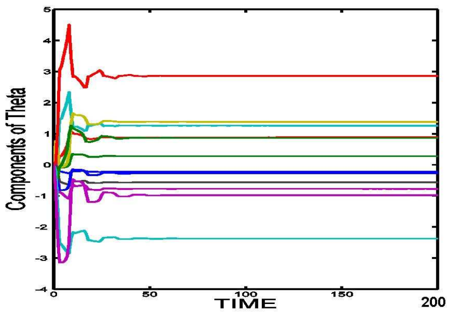

Fig. 11. Time Evolution of Θ_0: Linear Case

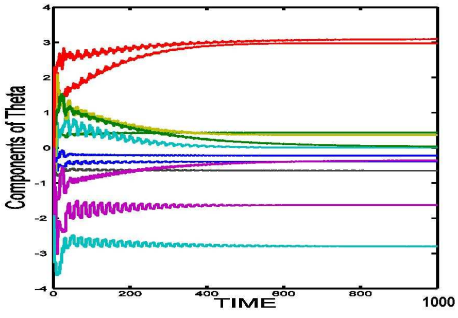

Fig. 12. Time Evolution of Θ_8: Linear Case

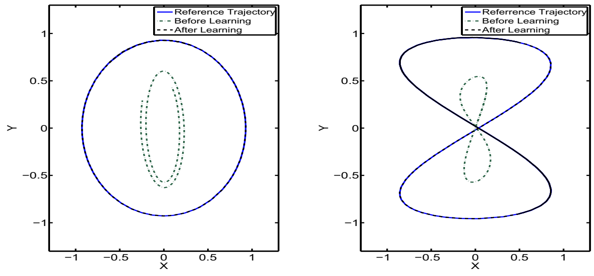

Fig. 13. Linear Simulation Results for 0 and 8

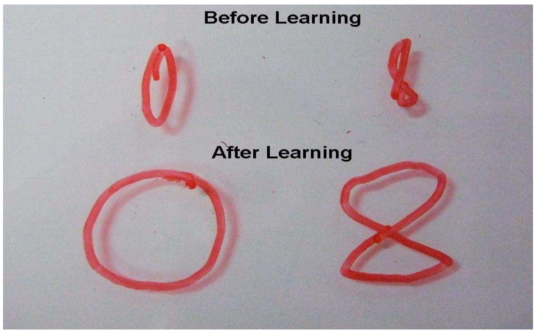

Fig. 14. Written Characters 0 and 8 by the Manipulator

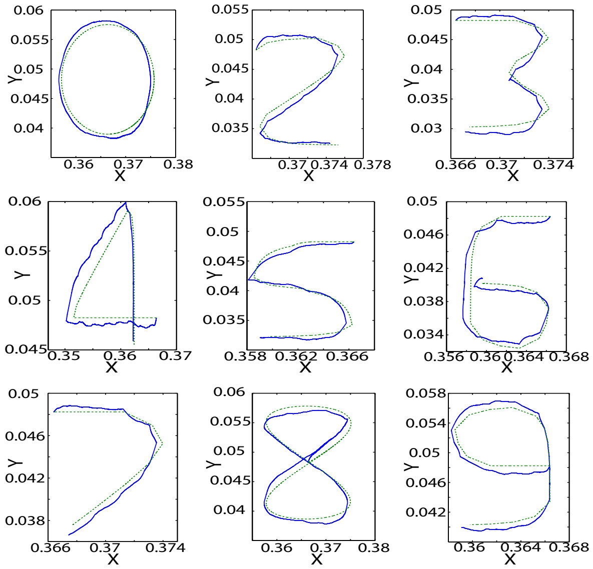

Fig. 15. Experimental Results of writing 0, 2, $\cdots$, 9

did before for "0" and "8" with (21). We then switch the feedforward controller depending on the objective. For example, we use $Q_{\Theta_0}(s)$ if we want to write "0" and $Q_{\Theta_2}(s)$ to write "2" and so on. Note that we use the same $K_{\rm fb}$ and $W(s)$ in the simulation part to perform the experiment. Fig. 14 shows the picture of written "0" and "8" before and after learning. The resulting picture is close to the numerical simulation result in Fig. 13. The experimental result in Fig. 15 shows the reference trajectory versus the actual one for each number written by the manipulator. The results indicate that the manipulator succeeded in writing different characters based on the switching strategy.

5. CONCLUSION

The main objective of this work is to demonstrate the practical effectiveness of the MIMO-FEL scheme proposed in Alali *et al.* (2006a) by an experiment. We verified that controllers generated by the numerical simulation show good tracking performance in the experiment with a real two-link manipulator. The next goal is to use on-line data during the learning process. Under this configuration, we can be free from the identification of the motor dynamics and the linearization of nonlinear dynamics of the manipulator.

REFERENCES

Kawato, M., K. Furukawa and R. Suzuki (1987). A hierarchical neural-network model for control and learning of voluntary movement, *Biol. Cybern.*, **57**, 169-185.

Miyamura, A and H. Kimura (2002). Stability of feedback error learning scheme, *Systems & Control Letters*, **45**, 303-316.

Muramatsu, E. and K. Watanabe (2004). Feedback error learning control without recourse to positive realness, *IEEE Trans. on Automatic Control*, **49 (10)**, 1762-1767.

Tao,G.(2003). Adaptive Control Design and Analysis, *John Wiley & Sons, Hoboken, NJ.*

de Vries,T., W. Velthuis and J. Amerongen (2000). Learning feedforward control: a survey and historical note, *Proceedings of IFAC Conference on Mechatronic Systems*, 197-202.

Colbaugh, R., K. Glass, and H. Seraji (1994). Direct adaptive control of robotic systems, *Journal of Intelligent and Robotic Systems*, **9**, 149-178.

Dawson, D.M., Z. Qu, J.F. Dorsey, and F.L. Lewis (1991). On the learning of a robot manipulator, *Journal of Intelligent and Robotic Systems*, **4**, 43-53.

Sun, Z. and T. Tsao (1999). Adaptive tracking control by system inversion, *Proceedings of the American Control Conference*, 29-33, San Diego, California USA, Jun.

Alali, B., K. Hirata and K. Sugimoto (2006a). MIMO feedback error learning control subject to strictly proper system, *Proceedings of the American Control Conference*, 5450-5455, Minneapolis, Minnesota USA, Jun. (also appeared in Trans. of the Society of Instrument and Control Engineers, **43 (4)** 2007)

Alali, B., K. Hirata and K. Sugimoto (2006b). MIMO Feedback Error Learning Control with Stability Proof and Generalizations, *17th International Symposium on Mathematical Theory of Networks and Systems*, 1155-1160, Kyoto, Japan, Jul.

Li, W. and J. Slotine (1989). An indirect adaptive robot controller, *Systems & Control Letters*, **12**, 259-266.

Bein, Z. and J. X. Xu (1998). Iterative learning control - Analysis, design, integration and applications, *Kluwer Academic Press, Boston.*

Sugie, T. and T. Ono (1991). An iterative learning control law for dynamical systems, *Automatica* **27 (4)**, 729-732.

Sugie, T (2004). On iterative learning control, *Proc. of the 12th International Conference on Informatics Research for Development of Knowledge Society Infrastructure (ICKS'04)*, 214-220.

Kawato, M., K. Furukawa andR. Suzuki (1988). Hierarchical neural networkmodel for voluntary movement with application to robotics, *IEEE Control System Magazine*, pp. 8-16, 1988.

X. Ruan, M. Ding, D. Gong, and J. Qiao (2007). Online adaptive control for inverted pendulum balancing based on feedback error learning, *Neurocomputing*, vol. 70, pp. 770-776, 2007.