

An Experimental Study of a High Performance Motion Control System

John T. Wen and Ben Potsaid
 Center for Automation Technologies
 Rensselaer Polytechnic Institute, Troy, NY 12180
 Emails: {wen,potsaid}@cat.rpi.edu

Abstract—High speed precision motion is required in applications ranging from high performance machining to electronic packaging. To achieve tight performance requirement, careful design of the feedback and feedforward controllers is necessary. This paper presents the feedforward controller design for a high speed positioning system after a feedback controller has already been implemented. We adopt the control architecture that consists of a nominal feedforward based on the approximate inverse of the identified linear model supplemented by an additive corrective control obtained based on iterative learning. Simulation and experimental results are included to demonstrate the efficacy of the proposed scheme.

I. INTRODUCTION

High performance motion control systems require speed and precision. To achieve this goal, feedforward compensation as well as feedback control are needed. Motivated by the stringent motion requirement of positioning systems commonly used in electronic manufacturing, we aim to improve the performance of point-to-point motion of a positioning system by designing a feedforward controller after a feedback controller has already been implemented. Since only a single positioner is considered, the system is single-input/single-output (SISO). In the experimental setup, the angular position is measured through the joint encoder. The data acquisition system is capable of sample rates up to 200KHz. The overall setup is shown in Figure 1.

We pose the control design problem as follows:

Consider a dynamical system, G , initially at rest, with input u and output y . Given the desired output y_d , find the input u that minimizes T such that $|y(t) - y_d| < \epsilon$ for all $t \geq T$ subject to the constraints

$$\max_t |\dot{y}(t)| \leq v_{\max}, \quad \max_t |\ddot{y}(t)| \leq a_{\max}. \quad (1)$$

The velocity and acceleration constraints are imposed to avoid internal saturation within the feedback controller and servo amplifier. Based on our experimental testbed and requirement in electronic manufacturing, we use the following specification in this study, $\epsilon = 1\mu\text{m}$, and $v_{\max} = 2e6\mu\text{m}/\text{sec}$, $a_{\max} = 300g = 3.43e9\mu\text{m}/\text{sec}^2$. If only the constraint is considered (ignoring the plant

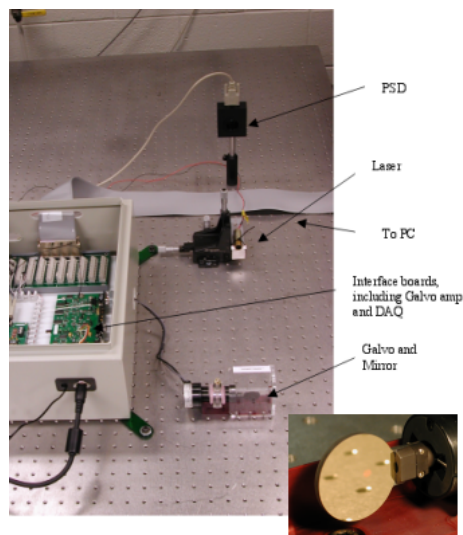


Fig. 1. Positioning System Experimental Testbed

dynamics), the solution of the above problem has a trapezoidal velocity profile. For performance illustration, we shall use $500\mu\text{m}$ move length for all simulation and experiment runs. With the trapezoidal profile, the settling time is 0.8ms. Since the dynamics is not considered, we call this the kinematic optimal solution.

For the feedforward control, we adopt the control architecture as shown in Fig. 2. The inverse dynamics filter F converts the desired input, y_{des} (generated from trapezoidal profile; other profiles can also be used), to the nominal command input u^* . To generate the tracking error for input correction, we choose the desired output trajectory, y^* , as the output predicted by the identified model, $G u^*$ (y_{des} by itself is typically too aggressive, leading to saturation). It is also possible to choose other reference outputs, e.g., shifted y_{des} . Due to the lack of a causal inverse (G is non-minimum-phase) and the plant/model mismatch in the actual experiment (including nonlinearities), a corrective input Δu is generated to reduce the tracking error $y - y^*$.

We first apply sine sweep to the experiment to obtain the experimental frequency response. A time domain subspace algorithm (eigen-realization algorithm, ERA,

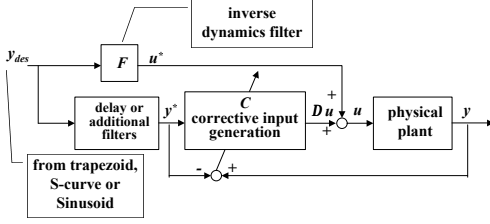


Fig. 2. Overall Inverse Dynamics Based Control Architecture

[1]) is used to obtain a transfer function model consisting of 9 zeros and 10 poles (listed in Table I) with the DC gain 0.9665. Note the presence of a weakly damped 7KHz mode, which, if uncompensated, can cause large amplitude oscillations (that would in turn lead to saturation in the feedback controller). The gain/phase comparison between the experiment and model is shown in Fig. 3. As is evident in these plots, the frequency responses match well except near the 7KHz weakly damped mode. The step response comparison (with a $27\mu\text{m}$ step input) is shown in Fig. 4. The oscillation is due predominantly to the 7KHz mode.

Poles	$\omega_n/2\pi$ (KHz)	ζ (%)
$-1.1840\text{E}4 \pm 0.8844\text{E}4 j$	2.352	80.12
$-0.3235\text{E}4 \pm 1.6269\text{E}4 j$	2.640	19.50
$-0.0794\text{E}4 \pm 4.3621\text{E}4 j$	6.944	1.82
$-1.4372\text{E}4 \pm 5.0167\text{E}4 j$	8.306	27.54
$-0.0254\text{E}4 \pm 7.1992\text{E}4 j$	11.458	0.35
Zeros	$\omega_n/2\pi$ (KHz)	ζ (%)
$0.1249\text{E}4 \pm 3.1757 j$	5.058	-3.93
$0.0065\text{E}4 \pm 4.2147 j$	6.708	-0.15
$3.2706\text{E}4 \pm 5.4823 j$	10.160	-51.23
$-0.0032\text{E}4 \pm 7.1750 j$	11.419	0.04
9.0389E4		

TABLE I

POLES AND ZEROS OF THE IDENTIFIED MODEL

The identified model has 7 unstable zeros. Direct inversion of this model (swapping numerator and denominator polynomials in the transfer function) would result in an unstable causal filter, preventing its implementation. We consider four alternate approaches: direct non-causal inversion, zero phase error tracking control (ZPETC) approximate causal inversion, and optimal model matching. Through simulation using the identified model, we determine that the latter two methods produce numerically sound results and can be used in the implementation of F in Fig. 2.

When $u^* = Fy_{des}$ is applied to the physical system, the settling time is considerably worse than the simulation result due to the model/plant mismatch. For a fixed desired trajectory, we apply an iterative refinement strategy to generate a corrective input Δu . The identified linear time invariant (LTI) model is used as the

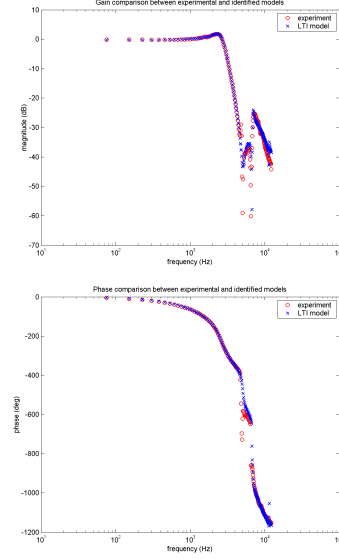


Fig. 3. Bode Plot Comparison between Experiment and Model

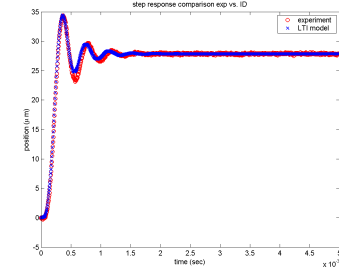


Fig. 4. Step Response Comparison between Experiment and Model

approximate gradient to generate a descent direction in each iteration. This method proves to be very effective, capable of achieving experimental performance similar to that in simulation in only a few iterations.

The rest of this paper is organized as follows. Section II discusses four inverse dynamics approaches for non-minimum-phase systems and the related simulation and experimental results. Section III presents the iterative refinement approach to producing a corrective input using a model-based gradient descent algorithm. Conclusion and future work are given in Section IV.

II. INVERSE DYNAMICS CONTROL

If the plant model is accurate and minimum-phase (all zeros are stable), then an obvious choice of the feedforward filter, F , is simply its inverse. If the plant is strictly proper (more poles than zeros), then higher derivatives of the desired output, y_{des} , will need to be used in the continuous time case, and future samples of y_{des} in the discrete time case. When the model is non-minimum-phase, its causal inverse is unstable and therefore cannot be implemented. There are several partial solutions to this problem:

- 1) Non-Causal inverse
- 2) Zero phase error tracking control (ZPETC)
- 3) Approximate causal inverse
- 4) Model matching

We will discuss each approach and compare their applicability to our test system. The plant, G , is assumed to be the identified model.

A. Non-Causal inverse

The non-causal inverse, F , implements the G^{-1} as a non-causal filter [2]–[5]. The output y would then be exactly y_{des} for $t \geq 0$. However, since F is non-causal, u will be nonzero for some period before $t = 0$. In effect, this will delay the output response. Through simulation study, we observe the following drawbacks of this scheme:

- 1) The output response is delayed due to the non-causal input. The amount of the delay is related to the time constant of the unstable zeros. For our experiment, the complex zeros at 6.7KHz is particularly poorly damped (0.15%), which could lead to long delays (which also depends on the desired input).
- 2) For systems with widely separated poles and zeros, the numerical errors in the non-causal input calculation could lead to significant output tracking error. For our test system, we were not able to achieve good tracking using this approach due to this issue.

Due to these problems, this approach was not further pursued.

B. Zero phase error tracking control

The zero phase error tracking control (ZPETC) was proposed in [6] and later refined in [7], [8]. The basic idea is simple: choose the input filter to achieve the zero phase of the combined system. Specifically, let $G_d(z^{-1})$ be the discretized plant (using zero-order-hold) and

$$G_d(z^{-1}) = \frac{G_{dnum}(z^{-1})}{G_{dden}(z^{-1})} = \frac{G_{dnum}^{(s)}(z^{-1})G_{dnum}^{(u)}(z^{-1})}{G_{dden}(z^{-1})} \quad (2)$$

where $G_{dnum}^{(s)}(z^{-1})$ contains the stable zeros and $G_{dnum}^{(u)}(z^{-1})$ contains the unstable zeros. If the input filter F is chosen to be

$$F(z^{-1}) = \left(\frac{G_{dden}(z^{-1})}{G_{dnum}^{(s)}(z^{-1})G_{dnum}^{(u)}(1)} \right) \left(\frac{G_{dnum}^{(u)}(z)}{G_{dnum}^{(u)}(1)} \right), \quad (3)$$

then the combined transfer function is

$$G_d(z^{-1})F(z^{-1}) = \frac{G_{dnum}^{(u)}(z^{-1})G_{dnum}^{(u)}(z)}{\left(G_{dnum}^{(u)}(1)\right)^2}, \quad (4)$$

which is of zero phase. The problem of this approach is that it is not possible to simultaneously achieve unity

gain as well (since that would require causal inversion of a non-minimum-phase transfer function). Many subsequent modifications have been proposed to minimize the deviation from the unity gain over a specified frequency range. However, the rigid zero phase requirement limits its broad applicability. We therefore chose not to pursue this scheme further either.

C. Approximate causal inverse

Instead of zero phase, we can try to maintain unity gain by choosing F as

$$F(z^{-1}) = \frac{G_{dden}(z^{-1})}{G_{dnum}^{(s)}(z^{-1})G_{dnum}^{(u)}(z)}. \quad (5)$$

The gain of the combined system G_dF is now unity for all frequencies, but there is a phase lag, as seen in the Bode plot in Fig. 5 (the slight gain variation is due to errors in numerical computation). The commanded inputs and corresponding outputs for the move length $500\mu\text{m}$ is shown in Fig. 6–Fig. 7. The settling time for $500\mu\text{m}$ move is 0.86ms, only 7.5% degradation of the kinematic optimal of 0.80ms.

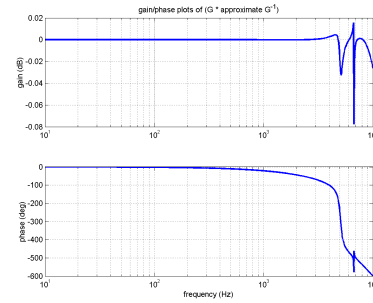


Fig. 5. Bode Plot of Overall Transfer Function for the Approximate Causal Inverse Case

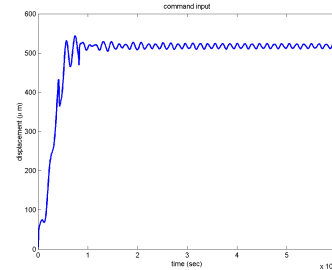


Fig. 6. Command Input: Approximate Causal Inverse Case

D. Model matching

The basic idea of the model matching approach [9], [10] is to find F to minimize the worst case frequency domain gain deviation from a desired combined transfer function W (chosen based on the frequency content in y_{des}):

$$F = \arg \min_{F \in RH_{\infty}} \|GF - W\|_{H_{\infty}}. \quad (6)$$

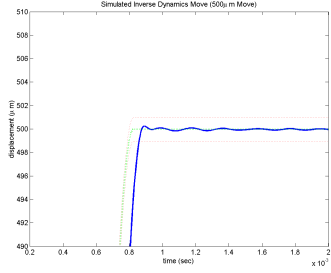


Fig. 7. Output Response: Approximate Causal Inverse Case

When G is non-minimum-phase, choosing $W = 1$ for all frequencies would cause numerical difficulty as the optimization will try to approach an infeasible solution (namely $F = G^{-1}$). However, with W chosen to be close to unity for the frequency range of interests, a feasible solution may be found.

Once F is found from (6), we can also implement FW^{-1} , which is in general non-causal, so that the combined transfer function is closer to unity. However, this tends to accentuate the mismatch in the high frequency, leading to poorer tracking performance.

To illustrate this approach, we

$$W(s) = \left(\frac{2\pi \cdot 7000}{s + 2\pi 7000} \right)^4. \quad (7)$$

Increasing the cut-off frequency and the order of the low pass filter increases the speed of the response and reduces the effect of high frequency modes. However, it also introduces greater phase lag, causing longer delay in the response. The resulting Bode plot comparison between GF and W is shown in Fig. 8. The corresponding commanded inputs and their shifted version (with the shift given by the difference between the settling times of the simulated output and the desired output) are shown in Fig. 9. The simulated output (assuming perfect model knowledge), shifted output, and desired output are shown in Fig. 10. The settling time is 0.86ms, the same as in the approximate causal inverse case.

The open loop poles of G are all contained in the zeros of F , and the stable zeros and stable images of the unstable zeros of G are contained in the poles F . Some additional faster poles and zeros are also present due to the H_∞ optimization to give additional high frequency roll-off. However, up to about 7KHz, F is very similar to the approximate stable causal inverse filter. Note that the outputs are close to the shifted trapezoidal profile. Therefore, if appropriate preview length (i.e., time shift) of the desired input is applied, the kinematic optimal profile can be obtained.

E. Experimental Performance

In Fig. 11, the simulation result is compared with the experimental response. The experimental responses

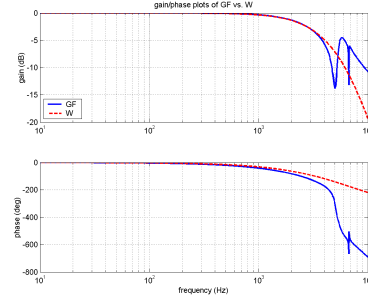


Fig. 8. Bode Plot Comparison

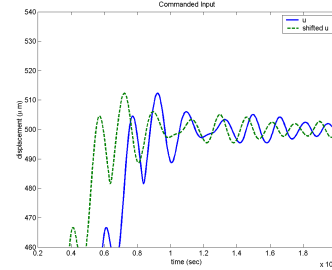


Fig. 9. Command Input Comparison

clearly lag behind the model predicted responses; this model/experiment discrepancy is especially pronounced near the completion of the movement (at around $490\mu\text{m}$) when the velocity is near zero. Since the largest mismatch occurs at low velocity, a possible cause is Coulomb friction, which is not modeled.

III. ITERATIVE REFINEMENT

The input filter F designed based on the nominal model performs less than satisfactory on the experiment due to the mismatch between the model and the physical system. We next apply the iterative refinement approach to update of the input based on the output tracking error. Using the complete trajectory tracking error to iteratively update the command input is known as iterative learning control (ILC). This concept was originally introduced in [11] for robot tracking control. Most ILC algorithms (proportional, Newton, Secant) updates the input at time t based on the previous input and tracking error also at

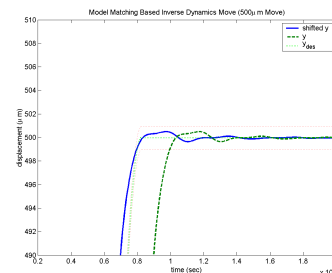


Fig. 10. Output Comparison

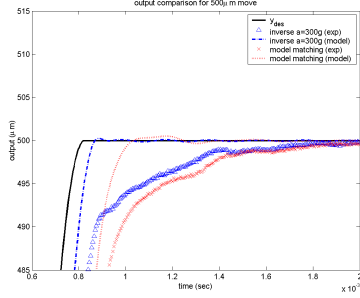


Fig. 11. Output Comparison

time t [12], [13]. In [14], [15], the input is updated based on the gradient of the complete trajectory tracking error with respect to the input. Similar idea is also proposed in a ILC general setting in [16]. For this study, we apply the gradient descent approach with the nominal LTI model, G , as the approximate gradient. We will show that provided that the modeling error is sufficiently small, the algorithm will converge.

The basic algorithm is summarized below:

Algorithm 1: Given $\underline{y}^* := \{y^*(t_i) : i \in 0, 1, \dots, N\}$ and $\underline{u}^* := \{u^*(t_i) : i \in 0, 1, \dots, N\}$. Set $\underline{u} = \underline{u}^*$.

- 1) Apply \underline{u} to the physical system and obtain the output sequence $\underline{y} := \{y(t_i) : i \in 0, 1, \dots, N\}$.
- 2) Update \underline{u} by adding a corrective term

$$\Delta \underline{u} = -\alpha G^*(\underline{y} - \underline{y}^*) \quad (8)$$

where G^* is the adjoint of G .

- 3) Iterate until $\|\underline{y} - \underline{y}^*\|$ or $\|\Delta \underline{u}\|$ becomes sufficiently small.

The key step in the above algorithm is the update equation (8). The adjoint of the LTI model, $G^*(s)$, is defined as $G^*(s) = G^T(-s) = G(-s)$ since the plant is SISO. The plant is stable, so $G(-s)$ is completely unstable (all poles in open right half plane), and may be considered as a non-causal filter. Let $g(t)$ be the impulse response of $G(s)$ and therefore $g_1(t) := g(-t)$ is the impulse response of $G(-s)$. Since $G(-s)$ is non-causal, $g_1(t) = 0$ for all $t > 0$.

To justify the use of the gradient algorithm, Algorithm 1, consider the evolution of the output tracking error function in each iteration. Suppose the actual physical system is

$$y = G_0(u), \quad (9)$$

where the system is initially at rest and G_0 is possibly nonlinear. Let the output tracking error be

$$J = \frac{1}{2} \|\underline{G}_0(\underline{u}) - \underline{y}^*\|^2, \quad (10)$$

where \underline{y} and \underline{u} are the output and input sequences stacked up as vectors. For Δu sufficiently small, the

change of J over one iteration is

$$\Delta J = (G_0(\underline{u}) - \underline{y}^*)^T \frac{\partial G_0}{\partial \underline{u}} \Delta \underline{u}. \quad (11)$$

Note the $\frac{\partial G_0}{\partial \underline{u}}$ is the physical system linearized about \underline{u} (the complete input trajectory). Let $\Delta G := \frac{\partial G_0}{\partial \underline{u}} - G$ with G the LTI model. It follows from (8) that if

$$\|\Delta G(\Delta \underline{u})\| \leq \gamma_{\Delta G} \|\Delta \underline{u}\| + \beta_{\Delta G}, \quad (12)$$

and G is full rank, then J would converge to a residue set with a bound proportional to $\beta_{\Delta G}$. The update constant α in (8) determines the rate of convergence; it cannot be chosen too large, otherwise the first order approximation in (11) would not be valid.

When Algorithm 1 is applied to the experiment with \underline{u}^* from the inverse dynamics approach (stable causal inverse; model matching result is similar), the tracking error is within $1\mu\text{m}$ after only a few iterations, as shown in Fig. 12 for tracking $y^* = Gu^*$. The model based settling time 0.86ms is now achieved experimentally. In the input comparisons shown in Fig. 13, the modified input signal is similar to the model based input, but the transient and steady state behaviors are changed to correct for the output tracking error.

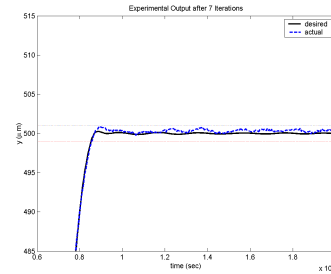


Fig. 12. Experimental Response after 7 Iterations

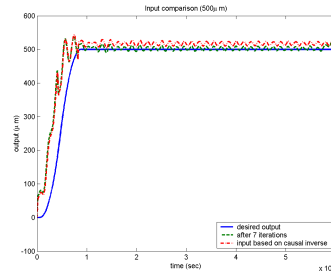


Fig. 13. Input Comparison: Iterative Refinement vs. Model Based Causal Inverse

It is also possible to directly apply the iterative refinement to the experiment without using model based inverse dynamics (i.e., start with zero initial control). The best case that we have obtained involves desired trajectories generated by the half sinusoidal acceleration profile (a slower motion profile than the trapezoid). In

Fig. 14 the desired outputs are tracked almost exactly after 7 iterations (but the settling time is now slower, to about 0.95ms). As seen in Fig. 15, the oscillation in the input now disappears since the input is no longer conditioned by the initial control that contains the oscillation from the inverse dynamics approach. The smoother sinusoidal profile has also contributed to the reduction of oscillation.

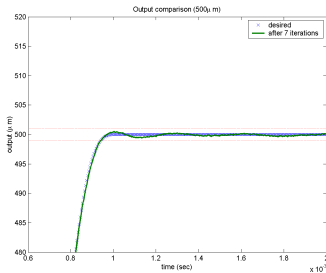


Fig. 14. Experimental Output: Iterative Refinement with Zero Initial Input and Sinusoidal Motion Profile

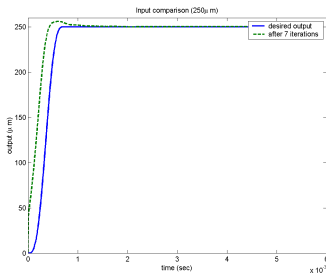


Fig. 15. Command Input: Iterative Refinement with Zero Initial Input and Sinusoidal Motion Profile

To summarize, as seen from the experimental results, the identified LTI system G is not good enough for a direct inverse dynamics approach, but performs very well in generating a gradient direction for iterative refinement. The best results were obtained using sinusoidal desired motion profile and zero initial control for the iteration.

IV. CONCLUSION

Feedforward control design for a high performance positioning system is considered in this paper. When the model based inverse dynamics based input is applied to the experiment, large discrepancy between experimental response and predicted output is observed. We then apply the gradient based iterative refinement using the identified model as the approximate gradient. Excellent tracking performance is obtained using this approach. The best case obtained involves starting with the zero initial control and using sinusoidal motion profile as the reference output. To avoid the batch iteration nature of

the iterative refinement approach, we will next try to use an FIR filter with adaptive coefficients to generate the corrective input (similar to the adaptive least mean square filter).

ACKNOWLEDGMENT

This material is based upon work supported in part by the National Science Foundation under Grant CMS-0301827, and in part by the Center for Advanced Technology in Automation, Robotics & Manufacturing under a block grant from the New York State Science and Technology Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] J.-N. Juang, *Applied System Identification*. Englewood Cliffs, N.J.: Prentice Hall, 1994.
- [2] L. Silverman, "Inversion of multivariable linear systems," *IEEE Transactions on Automatic Control*, vol. AC-14, no. 3, pp. 270–276, June 1969.
- [3] P. Moylan, "Stable inversion of linear systems," *IEEE Transactions on Automatic Control*, vol. 22, pp. 74–78, Feb. 1977.
- [4] E. Bayo and B. Paden, "On trajectory generation for flexible robots," *Journal of Robotic Systems*, vol. 4, no. 2, pp. 229–235, 1987.
- [5] E. Bayo and H. Moulin, "An efficient computation of the inverse dynamics of flexible manipulators in the time domain," in *Proc. 1989 IEEE Robotics and Automation Conference*, Scottsdale, AZ, 1989.
- [6] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *ASME Journal of Dynamic Systems, Measurement, and Control*, pp. 65–68, Mar. 1987.
- [7] Y. F. M. Yamada and S. Fujiwara, "Zero phase error tracking system with arbitrarily specified gain characteristics," *ASME Journal of Dynamic Systems, Measurement, and Control*, pp. 260–264, June 1997.
- [8] Y. F. M. Yamada and Z. Riadh, "Generalized optimal zero phase error tracking controller design," *ASME Journal of Dynamic Systems, Measurement, and Control*, pp. 165–169, June 1999.
- [9] B. Francis, *A Course in H_∞ Optimization Theory*. Lecture Notes in Control and Information Sciences, Vol.88, Springer-Verlag, 1987.
- [10] J. Doyle, B. Francis, and A. Tannenbaum, *Feedback Control Theory*. Macmillan Publishing Co., 1990.
- [11] S. Arimoto, "Mathematical theory of learning with applications to robot control," in *Adaptive and Learning Systems: Theory and Applications*, K. Narendra, Ed. New York: Plenum Press, 1985.
- [12] J. Xu and Q. Ji, "New ILC algorithms with improved convergence for a class of non-affine functions," in *Conference on Decision and Control*, Tampa, FL, Dec. 1998, pp. 660–665.
- [13] J. Xu and Y. Tan, "On the P-type and Newton-type ILC schemes for dynamic systems with non-affine-in-input factors," *Automatica*, vol. 38, no. 7, pp. 1237–1242, July 2002.
- [14] W. Cheng and J. Wen, "A class of learning controllers with application to tracking control of a flexible beam," in *Proc. 1993 IEEE Conference on Robotics and Automation*, Atlanta, GA, May 1993, pp. 411–416.
- [15] D. Gorinevsky, "Direct learning of feedforward control for manipulator path tracking," in *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*, Glasgow, UK, Aug. 1992, pp. 42–47.
- [16] K. Avrachenkov, "Iterative learning control based on quasi-newton methods," in *Conference on Decision and Control*, Tampa, FL, Dec. 1998, pp. 170–174.