ACC03-INV2705

Exploiting Iterative Learning Control for Input Shaping, with application to a wafer stage.

B.G.Dijkstra[1], O.H.Bosgra
Mechanical Engineering
Systems and Control Group,
Delft University of Technology
Mekelweg 2, 2628 CD Delft,
The Netherlands
Email: B.G.Dijkstra@wbmt.tudelft.nl

Abstract

The objective of an input design technique is to design an input to the system that results in an optimal tracking performance given some knowledge of the system response. Many such techniques focus on eliminating excitation of certain dominant system poles from the trajectory thus reducing any vibrations caused by these system poles. A downside to these methods is that they result in an elongation of the original trajectory. For a point-to-point control setting this means that there will be a trade-off between the elongation of the trajectory and the reduction of the settling time.

In this paper will be shown that Iterative Learning Control (ILC) can be used to design the input signal (trajectory) for a point-to-point motion in a way that eliminates all vibrations in the system without any elongation of the trajectory. This result is exactly the objective of classic input shaping techniques. The technique is illustrated with an application to a high precision wafer-stage.

1 Introduction

A high precision, and thus flexible, system that needs to perform a motion from one position to another tends to exhibit vibrations caused by excitation of the system poles, also referred to as flexible modes. A common way to avoid these vibrations is by eliminating the excitation of the dominant system poles from the trajectory, for instance with impulse input shaping [1][2][3] [4][5][6] or filtering. In this paper a tailored version of Iterative Learning Control (ILC) will be used to create an input to the system during the motion from one position into the next (time range 1) that eliminates all vibrations in the system after arrival at the second position (time range 2).

[1]This research project is being sponsored by Philips CFT

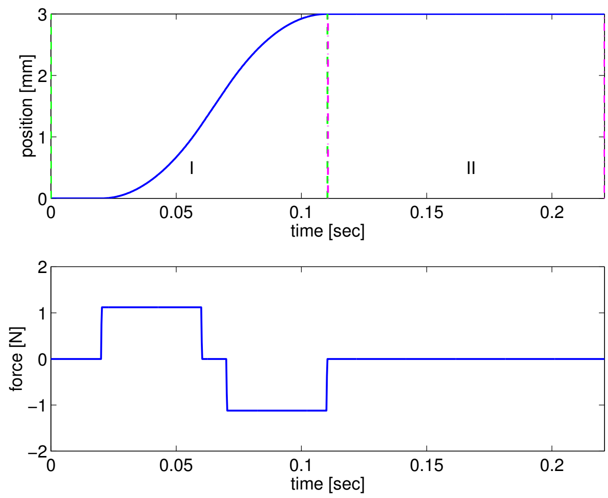

Figure 1: reference trajectory

2 Iterative Learning Control design

The system will be considered as a discrete time system over a finite time. This yields a mapping from an input vector (from $t = 0$ to $t = t_1$) to an output vector (from $t = t_2$ to $t = t_3$) denoted in the figures with I and II respectively. The top part of figure 1 shows the desired trajectory, a motion from one position into another position, and the feedforward (input to the system) that corresponds to that trajectory based on a rigid mass as a system model. Therefore this initial feedforward signal is the acceleration profile corresponding to the trajectory.

Consider the closed loop system shown in figure 2. The input to the system under consideration is denoted by x and the output by y. The mapping of the a finite input signal $\hat{x}$ to a finite output signal $\hat{y}$ is defined by the impulse response matrix H of the closed loop system (for an LTI system this is a lower triangular Toeplitz matrix). The ILC scheme used is shown in figure 3 where N is the number of samples in the input and output vectors, Z^{-1} denotes one trial delay for every sample in a vector, H and S denote the system mappings from the feedforward to the tracking error and from the reference trajectory to the tracking error respectively. See [7][8][9] for more details on this description.

The ILC controller L in figure 3, is the feedback interconnection between the error signal $\hat{e}_k$ and the input of the system $\hat{x}_k$. The system we now consider is a multivariable system given by:

$$\begin{aligned} \hat{x}_{k+1} &= \hat{x}_k + \hat{u}_k \\ \hat{e}_k &= -H\hat{x}_k + \hat{y}_{ref} \\ \hat{u}_0 &= \hat{z}_0 \end{aligned} \tag{1}$$

this clearly shows that the states of the ILC system are $\hat{x}_k$, and the objective of the ILC, will be to stabilize the poles of this system (generated by the discrete time integrator). A state feedback design method using the full ILC states enables a balance between system noise and ILC tracking performance as shown in [8]. The size of H however is equal to the square of the number of samples in the vectors under consideration (N), so for a reasonable trajectory this will be an extremely large matrix (square of N elements). These high dimensional control problems can be solved relatively effectively through the use of singular value decompositions (see e.g. [10] and [11]), but for LTI systems a more simple solution is presented in [7].

In [7] an LQR design was discussed, with a linear quadratic objective to balance the inputs and outputs of the system, resulting in the optimal controller that minimizes this objective:

$$\begin{aligned} J &= \sum_{k=1}^{M} \hat{y}_k^T Q \hat{y}_k + \hat{u}_k^T R \hat{u}_k \\ &= \sum_{k=1}^{M} \hat{x}_k^T H^T Q H \hat{x}_k + \hat{u}_k^T R \hat{u}_k \end{aligned} \tag{2}$$

Consider weighting $Q = I$ and $R = \beta$:

$$J = \sum_{k=1}^{M} x_k^T H^T H x_k + \beta u_k^T u_k \tag{3}$$

The corresponding Riccati equation is:

$$-S(\beta I + S)^{-1} S + H^T H = 0 \tag{4}$$

with feedback interconnection:

$$L = S^{-1} H^T \tag{5}$$

The solution to the Riccati equation (4) can be approximated by:

$$S = H^T H + \beta I \tag{6}$$

For a stabilizing solution to this optimal control problem, Q must be positive definite, and R positive semi-definite. Substitution of the approximate solution in the Riccati equation

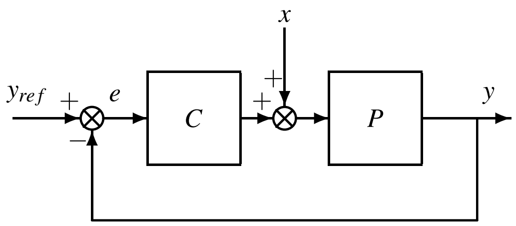

Figure 2: Closed loop servo control system.

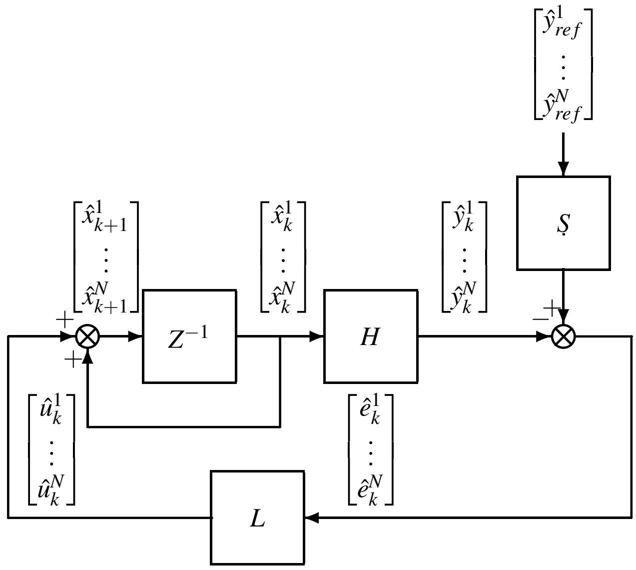

Figure 3: Iterative Learning Control scheme

shows that it is a solution to an optimal control problem with slightly different weighting Q_{eff}, which is still positive definite so it is still a solution for a stabilizing (convergent) ILC:

$$Q_{eff} = \left(H^T H + \beta^2 I \left(H^T H + 2\beta I\right)^{-1}\right) > 0 \tag{7}$$

This solution can be calculated with a simulation of a Hamiltonian system, where only a state space model of the process sensitivity (A, B, C) and the relative weighting (β) is needed, which is not very numerically demanding:

$$\begin{aligned} x(l+1) &= Ax(l) + Bu(l) \\ p(l) &= A^T p(l+1) + C^T C x(l) \\ u(l) &= \beta^{-1} y(l) - \beta^{-1} B^T p(l) \end{aligned} \tag{8}$$

For a detailed derivation and explanation of this equation please refer to [7]. This method can be used to exactly generate the solution in equation 5 without the need for solving a large dimensional Riccati equation.

3 ILC for input design

In order to design an input trajectory with ILC, the input signal and output signal have to be separated in time. For input shaping techniques usually the errors *after* the end of the input trajectory are desired to be small, so in the example shown below that is what is chosen. The method shown can be used for any choice of time ranges I and II.

Define N_1 as the sample corresponding to time $t = t_1$, and the divide the system matrix H like:

$$\begin{bmatrix} \hat{y}^1 \\ \vdots \\ \hat{y}^{N_1} \\ \hat{y}^{N_1+1} \\ \vdots \\ \hat{y}^N \end{bmatrix} = \begin{bmatrix} H_{11} & 0 \\ H_{12} & H_{11} \end{bmatrix} \begin{bmatrix} \hat{x}^1 \\ \vdots \\ \hat{x}^{N_1} \\ \hat{x}^{N_1+1} \\ \vdots \\ \hat{x}^N \end{bmatrix} \tag{9}$$

This makes time range I from sample 1 to N_1 and time range II from sample $N_1 + 1$ to N The mapping relevant for input design of the point-to-point trajectory presented above is then described by:

$$\begin{bmatrix} \hat{y}^{N_1+1} \\ \vdots \\ \hat{y}^N \end{bmatrix} = \begin{bmatrix} H_{12} \end{bmatrix} \begin{bmatrix} \hat{x}^1 \\ \vdots \\ \hat{x}^{N_1} \end{bmatrix} \tag{10}$$

which is a full Toeplitz matrix with a rank equal to the order of the underlying LTI system.

Using this system matrix, an ILC is designed according to the methods described in [7] and [10] which has been applied on a high precision wafer stage. The iterative learning control feedback then looks like:

$$L = \left(H_{12}^T H_{12} + \beta I\right)^{-1} H_{12}^T \tag{11}$$

Note that when using the method based on the simulation of the Hamiltonian in equation 8 as described in [7], special care must be taken during the simulation to extract exactly the time intervals relevant and

4 Experimental results

For a wafer stage making a stepping motion, the point-to-point control problem is defined as a motion from one position into the next (the *step*, hence *step time*), and the time required for vibrations to get within the desired boundaries (the *settling time*). In the case of the wafer stage discussed here these boundaries are $\pm 100nm$.

Figure 4 shows the error resulting from the application of the trajectory with only the simple feedforward shown in figure 1. A closer look in figure 5 shows some fast system

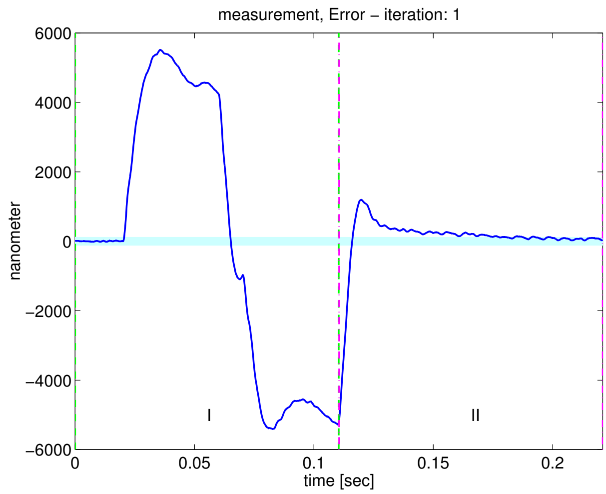

Figure 4: vibration error, trial 1 ($t = 0$ to $t = t_3$)

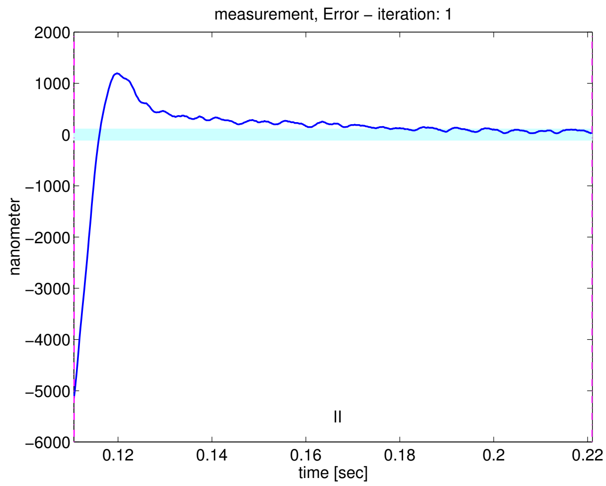

Figure 5: vibration error, trial 1 ($t = t_2$ to $t = t_3$)

vibrations along with a very dominant slow oscillation. The *settling time* in this case is nearly as long as the *step-time*, primarily dominated by the slow effect. The error during trial 1 is nowhere near the required accuracy of $\pm 100nm$ (denoted by the grey area).

After learning the measured residual vibrations, shown in figures 6 and 7, have been reduced significantly, to the level where they are within the required limits, hence completely eliminating settling time.

The feedforward signal in figure 8 that has been found to obtain this 'zero settling' behavior shows that extra vibrations have been added to the input signal rather than removed from the input signal. This has to do with the fact that ILC obtains information about the system zeros as well as the system poles. Classic input design techniques often only take into account the poles of a system. Figure 8 also shows that the input signal during the settling period is zero, un-

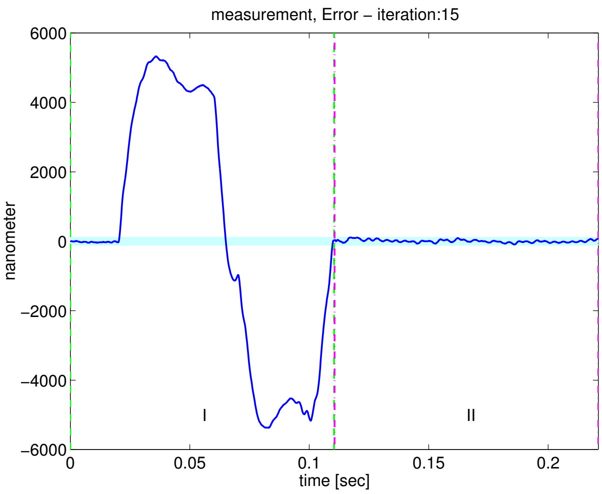

Figure 6: vibration error, trial 15 ($t = 0$ to $t = t_3$)

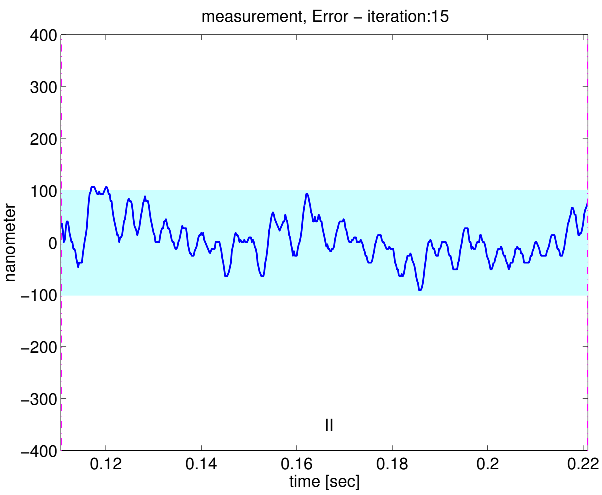

Figure 7: vibration error, trial 15 ($t = t_2$ to $t = t_3$)

like results obtained with conventional ILC methods. This means that the input to the system is, as required, only active during the step of the trajectory, and no signal is needed during the settling phase.

Another important thing to note is that the control signals needed for the elimination of the vibrations, are much larger than the initial input signals which is mainly caused by the large degree of excitation of the desired trajectory. If this causes a problem for a specific system, such a result may be motivation to design a different trajectory because apparently the system is not able to cope with the original trajectory. The input signals obtained with the discussed ILC methods could even be used to identify problematic dynamics if so required. Hence this form of ILC can be used to assess a trajectory as well as design the control signals to generate it.

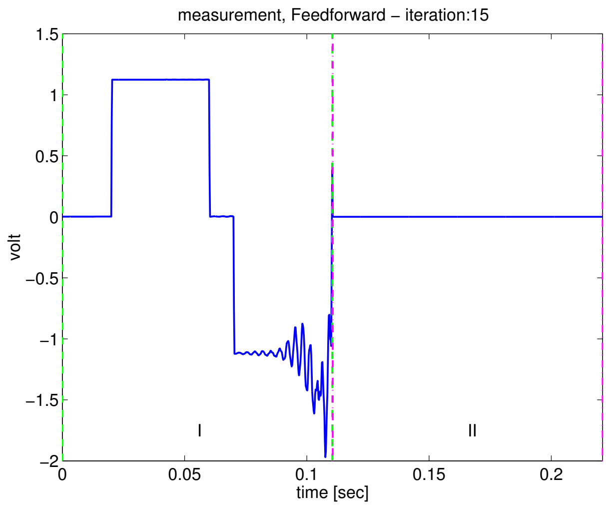

Figure 8: input signal, trial 15 ($t = 0$ to $t = t_3$)

5 Discussion

Using Iterative Learning Control for input shaping has been shown to work quite well, illustrated with experimental results on an industrial grade experimental set-up. This method of designing an input sets it apart from conventional input shaping techniques in two important ways:

Firstly the designed inputs are based on experimental designs. For the ILC some model information is needed but even if this information is not entirely accurate, with a stable ILC the result will still be a very accurately shaped signal. Of course this means that if system dynamics change, the signal will also change and the signals as such are not robust. The ILC method however is robust, but that falls outside the scope of this paper.

Secondly, this method will find an applicable control signal even when the desired trajectory is very exciting to the system. This can result in very large control signals (as is also visible in figure 8), showing it is difficult for the system to track such a trajectory with no residual vibrations. This information can be useful to use in conjunction with other input shaping techniques as experimental assessment of the quality of an input shaper. This input signal can even be used to identify certain system poles that may present problematic excitation for a certain trajectory.

The results shown in this paper illustrate that it is possible to obtain vibration free tracking for a trajectory without the need of an elongation of the trajectory. It also shows however that for a given trajectory obtaining this tracking may take a lot of control effort (large control signals).

References

[1] N.C. Singer and W.P. Seering. Preshaping command inputs to reduce systems vibration. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 112:76–82, 1990.

[2] P.H. Meckl and W.P. Seering. Experimental evaluation of shaped inputs to reduce vibration for a cartesian robot. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 112:159–165, June 1990.

[3] W.E. Singhose and N.C. Singer. Effects of input shaping on two-dimensional trajectory following. *IEEE Transactions on Robotics and Automation*, 12(6):881–887, 1996.

[4] Roberto Kinceler and Peter H. Meckl. Corrective input shaping for a flexible-joint manipulator. *American Control Conference*, pages 1335–1339, June 1997.

[5] Edwin Hou Lucy Y. Pao, Timothy N. Chang. Input shaper designs for minimizing the expected level of residual vibration in flexible structures. *Proceedings of the American Control Conference*, pages 3542–3546, 1997.

[6] Marc Bodson. An adaptive algorithm for the tuning of two input shaping methods. *Automatica*, 34(6):771–776, 1998.

[7] Branko G. Dijkstra and Okko H. Bosgra. Extrapolation of optimal lifted system ILC solution, with application to a waferstage. *American Control Conference*, May 2002.

[8] Branko G. Dijkstra and Okko H. Bosgra. Noise suppression in buffer-state iterative learning control, applied to a high precision wafer stage. *Conference on Control Automation*, September 2002.

[9] Branko G. Dijkstra and Okko H. Bosgra. Convergence design considerations of low order q-ILC for closed loop systems, implemented on a high precision wafer stage. *Conference on Decision and Control*, December 2002.

[10] Rob Tousain and Eduard Van der Meché. Design strategies for iterative learning control based on optimal control. *Conference on Decision and Control*, 40:4463–4468, December 2001.

[11] Won Cheol Kim, In Sik Chin, Kwang Soon Lee, and Jinhoon Choi. Analysis and reduced-order design of quadratic criterion-based iterative learning control using singular value decomposition. *Computers and Chemical Engineering*, 24:1815–1819, 2000.