

# Evaluation of (unstable) non-causal systems applied to iterative learning control

**Citation for published version (APA):**

Schneiders, M. G. E., Molengraft, van de, M. J. G., & Steinbuch, M. (2002). Evaluation of (unstable) non-causal systems applied to iterative learning control. In J. A. M. Amerongen, van (Ed.), *Proceedings of the 8th mechatronics forum international conference (Mechatronics 2002), 24-26 June 2002, University of Twente, The Netherlands* (pp. 1-10). Drebbel Institute for Mechatronics.

**Document status and date:**

Published: 01/01/2002

**Document Version:**

Accepted manuscript including changes made at the peer-review stage

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# EVALUATION OF (UNSTABLE) NON-CAUSAL SYSTEMS APPLIED TO ITERATIVE LEARNING CONTROL

M.G.E. Schneiders<sup>1</sup>, M.J.G. van de Molengraft<sup>2</sup> and M. Steinbuch<sup>3</sup>

Eindhoven University of Technology,  
Control Systems Technology,  
Department of Mechanical Engineering,  
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

email:

<sup>1</sup>m.g.e.schneiders@tue.nl

<sup>2</sup>m.j.g.v.d.molengraft@tue.nl

<sup>3</sup>m.steinbuch@tue.nl

## Abstract

*This paper presents a new approach towards the design of iterative learning control (Moore, 1993). In linear motion control systems the design is often complicated by the inverse plant sensitivity being non-causal and even unstable. To overcome these problems the inverse system is split up in a causal and a non-causal part. We apply high-performance differentiating filters together with a mixed boundary value ODE solver to compute the control signal. This allows for an accurate approximation of the theoretical solution and offers the advantage of control over the boundary conditions of the learnt signal. The advantages of this approach over the existing commonly-used ZPETC technique are demonstrated by two examples, respectively a non-minimum-phase system and an industrial H-drive.*

## 1 Introduction

In this paper an algorithm for *ILC* (Iterative Learning Control) design is proposed that can deal with both non-causal and unstable inverse plant sensitivities. Iterative learning control improves the tracking accuracy of a (closed-loop) control system by learning from previous experience with the same trajectory. This is done by updating the feed-forward signal  $u_{ff}$  in an iterative way according to a learning law. If we have a SISO loop with plant  $P$  and feedback-controller  $C$  (both LTI systems), the process (or plant) sensitivity is defined as:

$$R = PS = \frac{P}{1 + PC} \quad (1)$$

where  $P$  is the plant transfer function,  $C$  the controller transfer function and  $S$  is the sensitivity of the control loop. In a general ILC setup (Fig. 1) the learning filter  $L(s)$

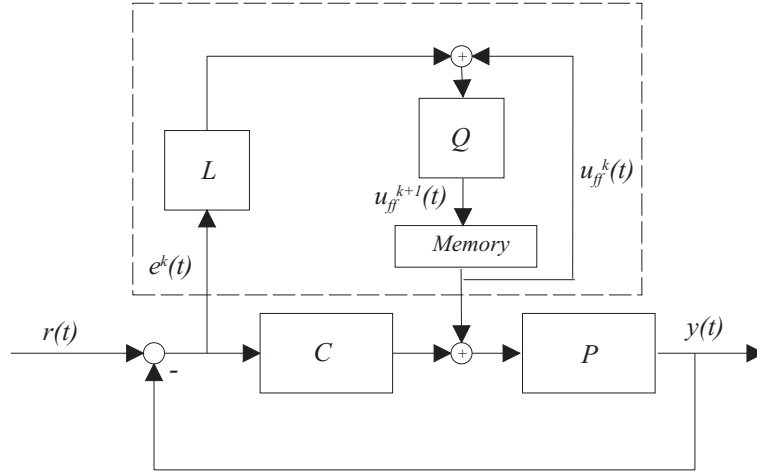


Figure 1: iterative learning setup

translates the tracking error  $e(t)$  into the necessary feedforward action, which is then added to the existing feedforward signal  $u_{ff}(t)$ .  $Q$  is a so-called robustness filter.

$$u_{ff}^{k+1}(t) = Q\{u_{ff}^k(t) + Le^k(t)\} \quad (2)$$

In theory, choosing the learning filter as the inverse of the plant sensitivity results in optimal convergence, because  $u_{ff} = R^{-1}e$ . To guarantee robust convergence of the learning process, a robustification filter  $Q(s)$  is added to cope with model uncertainties and measurement noise. Usually  $Q(s)$  is chosen as a lowpass filter.

To find the learning filter  $L$  the inverse model of the process sensitivity must be determined. Because closed-loop control systems (so also  $R$ ) are causal and (at least designed to be) stable, the inverse system cannot be evaluated directly. There are two reasons why this evaluation is not straightforward:

- inversion of a causal system leads to a non-causal system
- if the system is non-minimum phase, the inverse system will be unstable

A commonly used method is ZPETC, Zero Phase Error Tracking Control (Tomizuka, 1986). ZPETC needs a system description in discrete-time (transfer function) form. The non-causality can be canceled by a simple time shift. Since the ILC update is performed off-line this is no problem. Unstable zeros, which would become unstable poles of the inverse system, are mapped back as stable poles inside the unit circle. This assures that the frequency response of the calculated inverse system compared to the theoretic one exhibits no phase errors for the complete frequency range. The method is fast, but not always accurate since the amplitude behaviour of the inverse system is not exact. Especially for non-minimum phase systems, ZPETC can yield a bad approximation of the inverse model (see results in Section 5). The next sections will discuss another approach towards evaluating an inverse dynamic system, not yet encountered in ILC practice.

## 2 Inverse systems

We assume that we have a reliable (linear) model  $R(s)$  of the plant sensitivity at hand. It is also assumed that  $R(s)$  is a minimal realization of the system. The first step is to divide  $R^{-1}(s)$  into a causal part  $H_c(s)$  (3) and a strictly non-causal part  $H_{nc}(s)$  (4), as  $R^{-1}(s) = H_c(s)H_{nc}(s)$ , where it is assumed that  $L \leq K$ . The causal part  $H_c(s)$  can be unstable.

$$H_c(s) = \frac{\sum_{l=0}^L b_l s^l}{\sum_{k=0}^K a_k s^k} = \frac{b_L s^L + b_{L-1} s^{L-1} + \dots + b_1 s + b_0}{a_K s^K + a_{K-1} s^{K-1} + \dots + a_1 s + a_0} \quad (3)$$

$$H_{nc}(s) = \sum_{m=0}^M c_m s^m = c_M s^M + c_{M-1} s^{M-1} + \dots + c_1 s + c_0 \quad (4)$$

As we see (4), the non-causal system  $H_{nc}(s)$  can be written as a linear combination of differentiators of different order. Consider the problem where a discrete-time series  $e(k)$  has to be smoothed by  $R^{-1}$ . As  $e(k)$  is known beforehand, non-causal differentiating filters can be used as we will see in the next section. The output  $y_{nc}$  of this evaluation serves as an input for solving the causal part (Section 4). We only need an accurate solver to reduce numerical errors to the minimum. This is accomplished by using a decoupling technique to separate the stable and unstable parts of the system (van Loon, 1988). Together with an accurate ODE solver and partitioning of the time axis (using integration restarts) the desired accuracy is achieved. Furthermore, this method can handle mixed boundary value problems, which permits us to put extra constraints on begin and end values of the output signal. Section 5 will show the advantages of this new approach over ZPETC in two examples. Conclusions will be given in Section 6.

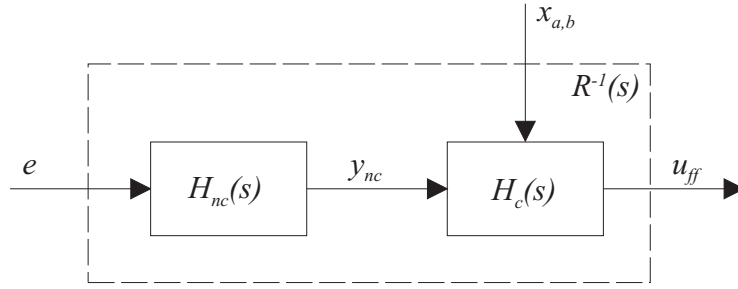


Figure 2: schematic representation of the new approach to calculate the learning feed-forward signal

## 3 Differentiating filters

### 3.1 A reconstruction problem

In most cases we do not have the pure continuous signal available to calculate the derivative. Most measurements, so also the error signal  $e(k)$ , are noisy and in discrete time:

$$e(k) = s(k) + v(k) \quad (5)$$

where  $s(k)$  is the real value of the signal and  $v(k)$  the measurement noise. With only  $e(k)$  given, we try to reconstruct the  $R^{\text{th}}$  derivative  $d_R(k)$  of the signal  $s(k)$ :

$$\hat{d}_R(k) = H_R(q = e^{i\omega T})y(k) \quad (6)$$

The design of a differentiating filter is nontrivial for at least two reasons:

- the relation between the sampled-time series  $s(k)$  and the continuous-time signal  $s(t)$  is not known in general.
- the measurement  $e(k)$  is corrupted with noise  $v(k)$ .

Because in learning control a preceding filter  $Q$  will bother about the noise-pollution of the signal in a decent way, the differentiating filter does not have to cope with noise-reduction in the first place. The reconstruction problem mentioned is a much bigger problem. There are two main methods to reconstruct a continuous signal out of discrete-time data:

- Shannon reconstruction
- polynomial interpolation/fitting

Differentiating filters based on both methods will be briefly discussed in the next subsection. Important work in this field is done in (Carlsson, Söderström and Ahlén, 1987) and (Carlsson, 1989), where a number of first order differentiating filters are presented. A  $R^{\text{th}}$ -order differentiator in Laplace-domain,  $H_R(s) = s^R$ , has  $\frac{\pi}{2}R$  phase-lead and is non-causal. The ideal frequency response for a discrete-time differentiator is given in (7), where  $q$  is the shift operator:  $x_{k+1} = qx_k$ .

$$H_R^i(q = e^{i\omega T}) = (i\omega)^R \quad \omega \leq \frac{\pi}{T} \quad (7)$$

### 3.2 Filter selection

Accurate differentiating filters presented in (Carlsson, 1989) are all non-causal FIR filters. Digital filters with Finite-duration Impulse Response (all-zero, or FIR filters) have both advantages and disadvantages compared to Infinite-duration Impulse Response (IIR) filters. Primary advantages are:

- filters can have exactly linear phase
- always stable
- startup transients have finite duration

The primary disadvantage of FIR filters is that they often require a much higher filter order than IIR filters to achieve a given level of performance. Because linear phase-behaviour of IIR filters cannot be guaranteed (except using (forward-backward) non-causal zero(!)-phase filtering), they are no good way to design accurate differentiating filters.

A general layout for a  $N^{th}$  order differentiating FIR filter to calculate the  $R^{th}$  derivative can be given as:

$$H_R(q) = \frac{1}{2T^R} \left( \sum_{n=1}^N c_n (q^n + (-1)^R q^{-n}) + c_0 \right) \quad (8)$$

This filter is symmetric for odd derivatives and anti-symmetric for even derivatives. Using this layout we guarantee the desired phase behaviour (as in (7)), and we calculate filter coefficients which are independent of the sample time  $T$ . In (Carlsson, 1989), a design criterion is suggested which covers both reconstruction methods. The filter coefficients are calculated by minimizing the following cost function:

$$E = \int_0^{\frac{\pi}{T}} |(i\omega)^R - H_R(q = e^{i\omega T})|^2 d\omega \quad (9)$$

subject to a set of equality constraints

$$\mathbf{W}\tilde{c} = \tilde{e} \quad (10)$$

with  $\tilde{c} = [c_0 \ c_1 \ \dots \ c_N]^T$ . The constraints in (10) will not be further specified here, but mostly they represent constraints on the filter characteristic around  $\omega = 0$  (see (Carlsson et al., 1987) for more details).

### 3.3 Shannon-based differentiators

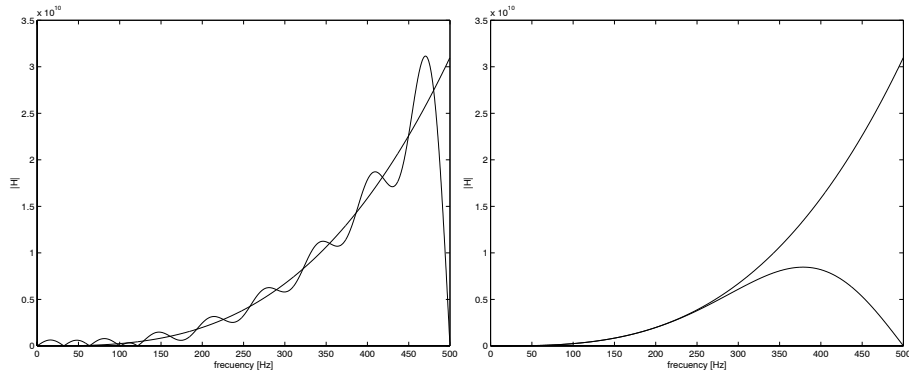
Differentiating filters based on Shannon reconstruction let  $N \rightarrow \infty$ , so they are unrealizable. However the filter order can be truncated to make a realizable filter. It can be proven that using this method is the same as constructing a filter by only optimizing the cost function (9). The result will be a wide-band (up to  $\omega \rightarrow \frac{\pi}{T}$ ) differentiator. The disadvantage of truncating the filter to order  $N$  are oscillations in the amplitude response of the filter, known as the Gibb's phenomenon. Other approaches minimize (9), sometimes with decreased integral bounds ( $\int_0^{\alpha \frac{\pi}{T}}$ , with  $\alpha \in [0, 1]$ ), subject to a set of constraints. For first order derivative filters all these methods have been discussed earlier in (Carlsson et al., 1987). Some of the above mentioned methods were expanded for higher order derivatives, resulting in wide band differentiators all with Gibb's oscillations. These oscillations give big relative errors, especially for low frequencies in higher order derivative filters. Therefore these methods are not very suited for our goal; we need very accurate higher order differentiators, especially in the low frequency range.

### 3.4 Polynomial interpolation/fitting

Design methods based on the polynomial reconstruction method only use the set of equality constraints (10). Constructing a specific set of  $N$  constraints (for details see (Carlsson, 1989)) to compute  $N$  coefficients  $c_1, \dots, c_N$  (solving a determined set), gives the same filter as one based on polynomial interpolation. A filter based on polynomial interpolation:

- determines the (unique) interpolating polynomial of degree  $2N$  from  $2N + 1$  data points
- estimates the  $R^{th}$  derivative of the midpoint  $N + 1$  by differentiating the polynomial

This method has extremely low error in the low-frequency range but some low-pass behaviour for higher frequencies. Note that  $2N \geq R$  to obtain a usable filter. Besides polynomial interpolation, we can use polynomial fitting to design a differentiating filter. This technique determines a polynomial fit of degree  $2N$  from  $2P + 1$  data points ( $P > N$ ). We obtain an overdetermined set of equations, which can be solved by least squares minimization. Such filter smooths out some high frequencies so this method has more lowpass behaviour than the interpolation technique. The low frequency accuracy is almost the same for both methods. Since the filter is not responsible for noise cancellation in the first place (we can use the robustification filter  $Q$  for this aim), polynomial interpolation is chosen as the best method to design differentiating filters for this particular purpose.



(a) Amplitude response for a truncated ideal Shannon-based differentiator ( $R=3$   $N=15$   $T=10^{-3}$  s)

(b) Amplitude response for a differentiator based on polynomial interpolation ( $R=3$   $N=6$   $T=10^{-3}$  s)

Figure 3: Comparison between two 3<sup>rd</sup>-order differentiating filters, based on the two different reconstruction methods. Note that the phase response is exact for both methods.

## 4 An accurate ODE solver

Now we are able to evaluate the non-causal system  $H_{nc}(s)$ , and denote the output signal as  $y_{nc}$ . We use this signal as an input for the causal system  $H_c(s)$  of order  $K$  (according to (3)). This system can easily be converted to state space, using the notation of Fig. 2:

$$\begin{cases} \dot{\tilde{x}}(t) &= \mathbf{A}\tilde{x}(t) + \mathbf{B}y_{nc}(t) \\ u_{ff}(t) &= \mathbf{C}\tilde{x}(t) + D y_{nc}(t) \end{cases} \quad (11)$$

The real problem is solving the time-invariant inhomogeneous linear set of  $K$  coupled first order differential equations on a finite time-span, say  $a \leq t \leq b$ :

$$\dot{\tilde{x}}(t) = \mathbf{A}\tilde{x}(t) + \mathbf{B}y_{nc}(t) \quad (12)$$

Shooting methods partition the time-interval into parts and solve several Initial Value Problems (IVP's). By doing this, unstable increasing modes will not reach unacceptable large grow factors that lead to numerical inaccuracy. After finding all sub-solutions of the IVP's, the total problem can be computed by solving a linear algebraic equation. The main drawback of this technique is that for very stiff problems the number of subintervals can become unacceptably large.

To overcome this problem a decoupling technique is used, which separates the stable (decreasing) and unstable (increasing) modes of the system. Unstable parts can be integrated backwards (finding a stable solution). Unfortunately, in this technique some variables (resulting from algebraic transformations) can grow unbounded, but much less catastrophic than the IVP's in the shooting technique. So, some partitioning of the time-interval (applying integration restarts) is still needed. In this setup it is easy to add constraints as a set of mixed boundary conditions. The  $K$  boundary conditions can be specified on initial and final values of the states:

$$V_a \tilde{x}(t=a) + V_b \tilde{x}(t=b) = \tilde{v}_{bc} \quad (13)$$

$V_a$  and  $V_b$  are constant  $K \times K$  matrices with  $rank(V_a) + rank(V_b) = K$ . A stiff ODE solver in MATLAB is used to solve the differential equation in the sub-intervals. To prescribe initial and end conditions in terms of the output of the system  $u_{ff}(t)$ , we have to include the output (static) equation:

$$u_{ff}(t) = \mathbf{C}\tilde{x}(t) + D y_{nc}(t) \quad (14)$$

By taking derivatives of this expression, it is possible to express the state-conditions at a certain moment  $\tilde{x}(t=t^*)$  in terms of the input and output values and their derivatives at moment  $t^*$  as:

$$\mathcal{O}\tilde{x}(t^*) = \begin{bmatrix} u_{ff}(t^*) \\ \dot{u}_{ff}(t^*) \\ \vdots \\ u_{ff}^{K-1}(t^*) \end{bmatrix} - \mathcal{R} \begin{bmatrix} y_{nc}(t^*) \\ \dot{y}_{nc}(t^*) \\ \vdots \\ y_{nc}^{K-1}(t^*) \end{bmatrix} \quad (15)$$

with,

$$\mathcal{R} = \begin{bmatrix} D & 0 & 0 & 0 & \cdots & 0 \\ \mathbf{CB} & D & 0 & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & D & 0 & \cdots & 0 \\ & & \vdots & & & \\ \mathbf{CA}^{K-2}\mathbf{B} & \mathbf{CA}^{K-3}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} & D \end{bmatrix}, \quad \mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{K-1} \end{bmatrix} \quad (16)$$

in which  $u_{ff}^r(t^*)$  is defined as the  $r^{th}$  time-derivative of  $u_{ff}$  at moment  $t^*$  ( $\left. \frac{\partial^r u_{ff}}{\partial t^r} \right|_{t=t^*}$ ). The same holds for the input  $y_{nc}$ . By transforming  $H_c(s)$  into the observable canonical state-space form, we ensure that  $\mathcal{O}$  (the observability matrix) is lower triangular. If we want to prescribe the output and its derivatives up to a certain order, we do not need



to fix all states. Since both  $\mathcal{R}$  and  $\mathcal{O}$  are lower triangular matrices, the first  $p$  states ( $1 \leq p \leq K$ ) will prescribe  $u_{ff}(t^*)$ ,  $\dot{u}_{ff}(t^*)$  up to  $u_{ff}^{p-1}(t^*)$  if  $y_{nc}(t^*)$ ,  $\dot{y}_{nc}(t^*)$  up to  $y_{nc}^{p-1}(t^*)$  are given. Since input signal  $y_{nc}(t)$  is given, derivatives at every moment can be calculated using a dedicated filter as presented in Section 3. Now it is possible to convert initial and end conditions of the feedforward signal (and its derivatives) to a set of boundary conditions as in (13).

## 5 Applications

First the new approach is applied to a non-minimum phase system. Consider the double pendulum (Fig. 4.a), without friction or gravity but with a flexible joint with stiffness  $k_s$  and damping  $b_s$ . The linearized model (around  $\theta_1 = \theta_2$ ) is a non-minimum phase system:

$$\begin{aligned} \begin{bmatrix} J_1 + m_2 l_1^2 & l_1 l_2 m_2 \\ l_1 l_2 m_2 & J_2 + m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} b_s + b & -b_s \\ -b_s & b_s \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \\ + \begin{bmatrix} k_s & -k_s \\ -k_s & k_s \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} T \\ 0 \end{bmatrix} \end{aligned} \quad (17)$$

Now we want the first link to make a half revolution according to a reference (Fig. 4.b), applying a torque  $T$  to the base-joint. The second link adds parasitic dynamics to the system, and will disturb the tracking of  $\theta_1$  to the reference signal. As long as the relative motion between the two joints is small ( $\theta_2 - \theta_1 \ll 1$ ), the linearization is governed. With this second order system it is possible to prescribe  $K = 2$  boundary conditions. We choose to fix the feedforward signal at zero at both begin and end time of the trajectory. Now ILC is used to improve tracking applying 10 iterations.

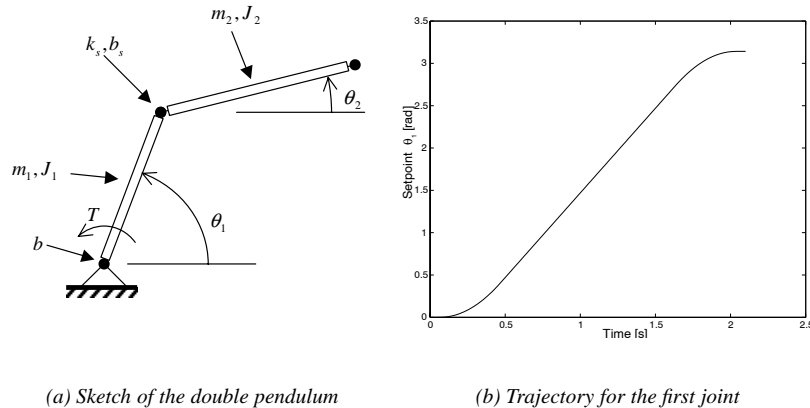


Figure 4: A non-minimum phase system: the linearized double pendulum

The new approach shows faster convergence and smaller residual errors, compared to ZPETC (see Fig. 5). Due to the action taken by ZPETC for unstable parts, this method shows oscillating behaviour in the feedforward signal. With both initial and final value of  $u_{ff}$  fixed to zero, it is possible to track cyclic setpoints using ILC without

startup or end effects. However forcing the feedforward signal in this form is probably the reason for the relative large residual error, which is expected to be almost zero in this theoretical example.

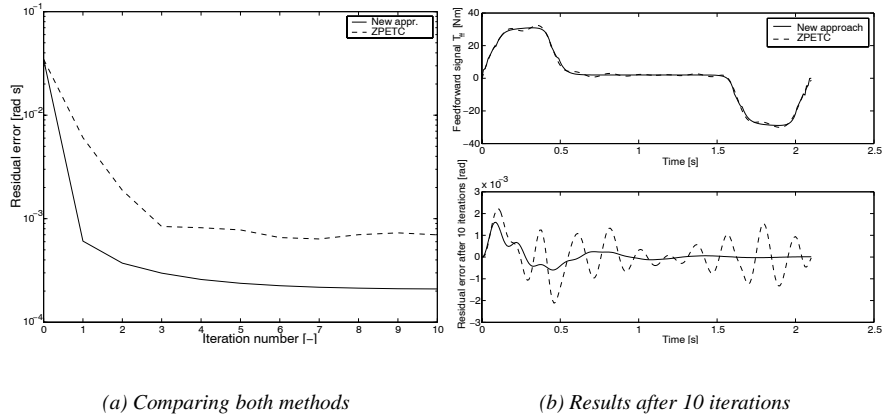


Figure 5: ILC results

Also, both the new approach and the standard ZPETC are applied to an industrial motion system, i.e. an H-drive. This robot uses 3 linear motion motor systems (LIMMs) to drive the 2 main sliders. Only the X-slide is considered in the learning process (the single-support slider in Fig. 6.a). The control loop is implemented with a dSPACE system. We used an 10<sup>th</sup> order model of the process sensitivity, obtained by frequency-domain identification. A lead-lag controller is used, resulting in a bandwidth of 30 Hz. A lowpass robustification filter is applied with a cutoff frequency of 250 Hz. After 7 iterations both methods have converged and the residual tracking errors are of the same order. As a first case on a higher-order minimum phase system, no specific initial conditions are imposed.

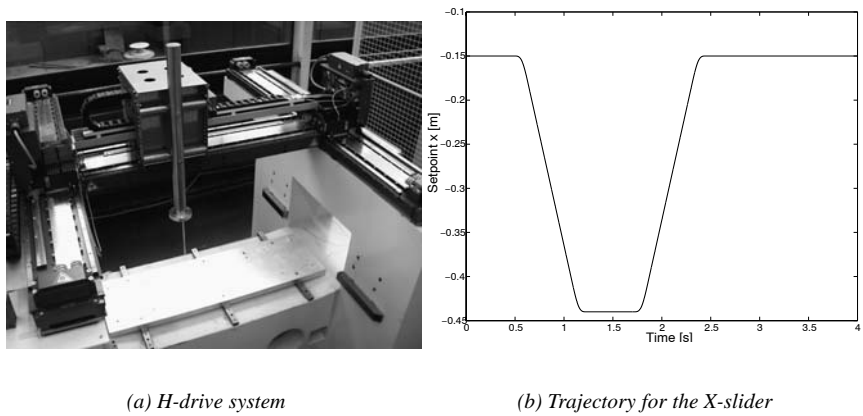
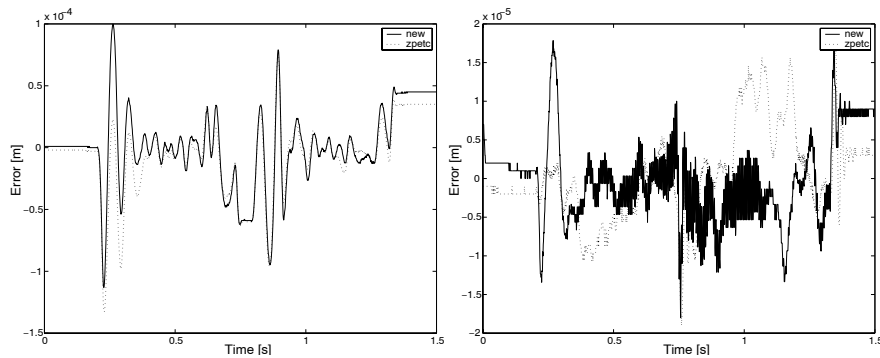


Figure 6: Mechanical system which can benefit significantly from ILC: an industrial H-drive



(a) Error after one iteration

(b) Error after seven iterations

Figure 7: ILC results

## 6 Conclusions

At this point, contrary to ZPETC, the new approach succeeds in calculating accurate inverse responses of non-minimum phase systems. For minimum phase systems of higher order the implementation of this new method is not competitive to ZPETC-method (Tomizuka, 1986) in sense of computation time. However, one major advantage is that we can put restrictions on begin and end values of the feedforward signal, which is very admirable in motion control. The possibilities of this freedom in prescribing boundary conditions and the effect on the tracking error is not fully explored at this moment. Further optimizing the numerical implementation can make this method more efficient and probably competing with ZPETC for higher-order real-world systems.

## References

- Carlsson, B. (1989). *Digital differentiating filters and Model based fault detection*, Uppsala dissertations from the faculty of science 28, 215pp., Acta Univ. Ups.
- Carlsson, B., Söderström, T. and Ahlén, A. (1987). *Digital differentiating filters*, *Technical report*, Teknikum, Institute of technology, Uppsala University.
- Moore, K. (1993). *Iterative Learning control for deterministic systems*, Springer Verlag, London.
- Roset, B. (2001). *Two-point boundary-value solver for stiff unstable linear systems (suited for application to ilc theory)*, Internal report 2001.50 (traineeship).
- Tomizuka, M. (1986). *Zero phase error tracking algorithm for digital control*, *ASME Journal of Dynamic Systems, Measurement, and Control* **109**(3): 65–68.
- van Loon, P. M. (1988). *Continuous decoupling transformations for linear boundary value problems*, Vol. CWI tract; 52 of VI, 198 p., Centrum voor Wiskunde en Informatica, Amsterdam.