Comparative study on first and second order ILC – frequency domain analysis and experiments[1]

M. Norrlöf
Department of Electrical Engineering, Linköping University,
SE-581 83 Linköping, Sweden
Email: `mino@isy.liu.se`

Abstract

Aspects on the behavior of a general second order iterative learning control (ILC) algorithm is presented from a frequency domain perspective. This includes stability as well as performance and robustness issues. The basis for the analysis is linear iterative systems and these are briefly described. A design algorithm for second order ILC schemes is proposed and analyzed both theoretically as well as in an experiment. In the experiment, done on a commercial industrial robot control system, the second order ILC design is compared with a first order ILC design. The result from both the analysis and the experiment is that the second order design is not better with respect to performance or robustness.

1 Introduction

It is a fact that many systems in applications like robotics repeat the same actions over and over again. Often it is also the case that there is a difference between what wants to be achieved and what is actually achieved. The idea of iterative learning control (ILC) is to use the information from previous *iterations* in such a way that the difference between what wants to be achieved and what actually is achieved eventually becomes smaller. The first papers on this topic are from 1984, [2, 5, 7], and since then a lot of publications have been published. Just to mention a few [9, 10, 4] that can serve as a deeper introduction to the topic.

The contributions of this paper are: An analysis of the behavior of second order ILC systems from a transient and asymptotic point of view, and a proposed design scheme, which is also tested on an industrial robot control system. The result from using the second order ILC algorithm is also compared with that of a first order ILC design. Second and higher order ILC methods have been analyzed in previous works, e.g., [3, 6, 4] and [14]. What has not been so much addressed, however, is what kind of transient behavior can be expected and how the filters in the second order ILC scheme shall be chosen.

2 Problem formulation

The discussion will be restricted to linear time invariant SISO systems that can be described in discrete time by

$$y_k(t) = T_r(q)r(t) + T_u(q)u_k(t) \tag{1}$$

where $y_k(t)$ is the output, $r(t)$ is the desired output, $u_k(t)$ is the control signal that can be modified by the ILC algorithm, and k is the iteration number. The signals are all assumed to be defined on a time interval $t \in [0, t_f]$. The transfer operators $T_r(q)$ and $T_u(q)$ are assumed to be stable discrete time filters.

The formulation in (1) is discussed in more detail in [13] and it covers easily both open loop as well as closed loop systems. It should be stressed however that here it will be assumed that a feedback control solution is applied before introducing ILC. This means that ILC will be considered as being a control strategy applied in *addition* to feedback control, not *instead* of feedback control.

In this paper a first and a second order ILC algorithm will be considered. By defining $e_k(t) \triangleq r(t) - y_k(t)$ the first order ILC updating equation becomes,

$$u_{k+1}(t) = Q(q)(u_k(t) + L(q)e_k(t)) \tag{2}$$

while the second order becomes,

$$\begin{aligned} u_{k+1}(t) = {} & Q_1(q)\big(u_k(t) + L_1(q)e_k(t)\big) \\ & + Q_2(q)\big(u_{k-1}(t) + L_2(q)e_{k-1}(t)\big) \end{aligned} \tag{3}$$

For the analysis of the second order ILC algorithm the notion of linear iterative systems will be used. These are presented briefly in the next section.

3 Linear Iterative Systems

A linear iterative system can be described as

$$z_{k+1}(t) = F(q)z_k(t) + F_r(q)r(t) \tag{4}$$

where $z_k(t) \in \mathbb{R}^N$, $r(t) \in \mathbb{R}$, and $F(q)$, $F_r(q)$ are matrices of discrete time transfer operators. In addition it is assumed that t is limited, $0 \le t \le t_f$. In the

---

[1] This work was supported by ABB Robotics within NUTEK's Center of Excellence ISIS at Linköping University.

analysis of high order ILC systems, $z_k(t)$ can be chosen as $[u_k(t),\ u_{k-1}(t)\ \dots\ u_{k-N+1}(t)]^T$. See [13] for more details. Now, a theorem that gives the condition for bounded input bounded output (BIBO) stability of a linear iterative systems is presented.

Theorem 1 (BIBO stability) *The iterative system given by* (4) *is BIBO stable if*

$$\bar{\rho} = \sup_{\omega \in [0, \pi/t_s]} \rho(F(e^{i\omega})) < 1$$

where $\rho(\cdot)$ *is the spectral radius of* $F(e^{i\omega})$.

For the proof see [13]. $F(e^{i\omega})$ is the frequency domain representation of $F(q)$. When the linear iterative system in (4) is BIBO stable, the asymptotic value of $z_k(t)$ can be calculated as

$$\lim_{k\to\infty} z_k(t) = z_\infty(t) = (I - F(q))^{-1} F_r(q) r(t) \tag{5}$$

The fact that higher order systems can be written on this form has also been explored in, e.g., [15] and [1].

4 Stability analysis

The results from the previous section can be used for the analysis of ILC systems. Here only first order and second order ILC algorithms will be discussed but it is straightforward to extend the results to higher order ILC systems.

4.1 First order ILC

Using the first order ILC updating formula given by (2) on the system described by (1) gives the updating equation for u_k as

$$\begin{aligned} u_{k+1}(t) = Q(q)\big(&(1 - L(q)T_u(q))u_k(t) \\ &+ L(q)(1 - T_r(q))r(t)\big) \end{aligned} \tag{6}$$

Now, let $F(q) = Q(q)(1 - L(q)T_u(q))$ and $F_r(q) = L(q)(1 - T_r(q))$. This means that it is possible to write (6) on the same form as (4) and from Theorem 1 the well known stability criterion follows,

$$|1 - L(e^{i\omega})T_u(e^{i\omega})| < |Q^{-1}(e^{i\omega})|, \quad \forall \omega \tag{7}$$

where the Q-filter can be used to increase the stability region. By choosing a Q different from 1 the asymptotic error will no longer be guaranteed to be zero.

4.2 Second order ILC

When using the second order ILC updating formula (3) on the system described by (1) the resulting update equation becomes,

$$\begin{aligned} u_{k+1}(t) = Q_1(q)&\big((1 - L_1(q)T_u(q))u_k(t)\big) \\ &+ Q_2(q)\big(1 - L_2(q)T_u(q)\big)u_{k-1}(t) \\ &+ \big(Q_1(q)L_1(q) + Q_2(q)L_2(q)\big)\big(1 - T_r(q)\big)r(t) \end{aligned}$$

Let

$$\begin{aligned} z_k(t) &= \begin{bmatrix} u_k(t) \\ u_{k-1}(t) \end{bmatrix}, F(q) = \begin{bmatrix} F_1(q) & F_2(q) \\ 1 & 0 \end{bmatrix}, \\ F_r(q) &= \begin{bmatrix} (Q_1(q)L_1(q) + Q_2(q)L_2(q))(1 - T_r(q)) \\ 0 \end{bmatrix} \end{aligned} \tag{8}$$

with $F_1(q) = Q_1(q)(1 - L_1(q)T_u(q))$ and $F_2(q) = Q_2(q)(1 - L_2(q)T_u(q))$ and it is possible to apply the general stability result of Theorem 1. If $Q_1 + Q_2$ is not chosen to be one the asymptotic error can not be guaranteed to be zero for this algorithm, see [13] for the proof. Compare this with the demand on Q in the first order case above.

5 Convergence behavior

It is not only the stability of the ILC system that is of importance. Also the performance is of great importance. The convergence speed is here analyzed from a frequency domain perspective. For a more detailed discussion the reader is referred to [12] and [13].

5.1 First order ILC

When considering convergence speed it is important to see how fast the system with the ILC algorithm converges to a pre-specified trajectory. This is the same as studying how

$$\tilde{U}^1_{k,\omega}(e^{i\omega}) = U_\infty(e^{i\omega}) - U_k(e^{i\omega}) \tag{9}$$

converges to zero. Using the first order ILC scheme in (2), the definition of the error $E_k(e^{i\omega}) = R(e^{i\omega}) - Y_k(e^{i\omega})$, and the system description in (1) it follows that

$$\tilde{U}^1_{k+1,\omega}(e^{i\omega}) = Q(e^{i\omega})\big(1 - L(e^{i\omega})T_u(e^{i\omega})\big)\tilde{U}^1_{k,\omega}(e^{i\omega})$$

With $F(e^{i\omega}) = Q(e^{i\omega})\big(1 - L(e^{i\omega})T_u(e^{i\omega})\big)$ this means that

$$|\tilde{U}_{k,\omega}| = |F(e^{i\omega})|^k |\tilde{U}_{0,\omega}| \tag{10}$$

which gives a clear message about the convergence. It is exponential with the rate decided by the frequency function $|F(e^{i\omega})|$.

5.2 Second order ILC

First the main result for second order ILC systems is presented.

Theorem 2 (Eigenvalue decomposition) *It is possible to write* $\tilde{U}_{k,\omega}$ *on the following decomposed form,*

$$\tilde{U}_k(e^{i\omega}) = \tilde{U}_0(e^{i\omega})(\kappa_{1,\omega}\lambda_{1,\omega}^{k+1} + \kappa_{2,\omega}\lambda_{2,\omega}^{k+1})$$

where $\tilde{U}_k(e^{i\omega})$ *is defined according to* (9) *and,*

$$\kappa_{1,\omega} = \frac{1 - \lambda_{2,\omega}}{\lambda_{1,\omega} - \lambda_{2,\omega}}, \quad \kappa_{2,\omega} = \frac{\lambda_{1,\omega} - 1}{\lambda_{1,\omega} - \lambda_{2,\omega}}$$

and $\lambda_{1,\omega},\ \lambda_{2,\omega}$ *are the eigenvalues of the matrix* $F(e^{i\omega})$.

The proof (see [13]) is based on the fact that $\tilde{Z}_k(e^{i\omega}) = Z_\infty(e^{i\omega}) - Z_k(e^{i\omega})$ can be written in the base constructed from the eigenvectors of $F(e^{i\omega})$. The result in Theorem 2 is the basis for the results on the behavior of $|\tilde{U}_{k,\omega}|$.

6 Design

Before going into the actual design example and the experiments, some general comments on the choices of eigenvalues that are actually reasonable.

The eigenvalues of the matrix in (8) can be calculated using

$$\lambda_{(1,2),\omega} = \frac{F_1(e^{i\omega})}{2} \pm \sqrt{\frac{(F_1(e^{i\omega}))^2}{4} + F_2(e^{i\omega})} \qquad (11)$$

with $F_1(e^{i\omega})$ and $F_2(e^{i\omega})$ as the frequency domain representations of the corresponding $F_1(q)$ and $F_2(q)$ in (8).

From now on it is assumed that the system where ILC is applied can be described as in (1).The proposed second order ILC updating formula is given by (3). A design methodology for this ILC algorithm has to be able to find the filters Q_1, Q_2, L_1, and L_2. How to choose the Q and the L filters in a first order ILC formulation is quite well known and there exist some algorithms that from a model can calculate the filters, see for example [4, 9, 13].

6.1 Design algorithm proposal

By considering (11) it is clear that given a value of $F_1(e^{i\omega})$ the best eigenvalue, with respect to the amplitude, is $\frac{F_1(e^{i\omega})}{2}$. This is the case when $F_2(e^{i\omega})$ is chosen such that

$$\frac{(F_1(e^{i\omega}))^2}{4} + F_2(e^{i\omega}) = 0$$

which is equivalent to

$$\begin{aligned} Q_2(e^{i\omega}) &= -\frac{(Q_1(e^{i\omega})^2}{4}, \\ L_2(e^{i\omega}) &= L_1(e^{i\omega})(2 - L_1(e^{i\omega})T_u(e^{i\omega})) \end{aligned}$$

This choice will, however, only fulfill the condition on $Q_1 + Q_2$, mentioned in Section 4.2, if $Q_1 = 2$. The approach here will instead be to choose an approximate solution based on a first order ILC design.

Algorithm 1 (Second order ILC design)

1. *Design a first order ILC algorithm, i.e., choose the filters Q and L according to one design methodology for first order ILC algorithms.*

2. *Choose Q_1 and L_1 according to,*

$$Q_1(e^{i\omega t_s}) = \frac{5}{4}Q(e^{i\omega}), \quad L_1(e^{i\omega}) = L(e^{i\omega})$$

3. *Choose Q_2 and L_2 such that*

$$\begin{aligned} Q_2(e^{i\omega}) &= -\frac{(Q(e^{i\omega}))^2}{4}, \\ L_2(e^{i\omega}) &= L_1(e^{i\omega})(2 - L_1(e^{i\omega})T_u(e^{i\omega})) \end{aligned}$$

The approach suggested in Algorithm 1 is model based since T_u is used in the construction of the L_2 filter. One important difference compared to many suggested first order ILC design schemes is however that it is the model and not its inverse that is included. If the first order ILC algorithm is chosen as the optimal solution $L_1 = T_u^{-1}$, without considering robustness, then from the choice of L_2 in the algorithm it is obvious that also $L_2 = T_u^{-1}$. The choice of Q_1 and Q_2 stem from the condition mentioned in Section 4.2. In the frequency band where the first order ILC system has zero error convergence, i.e., $Q_\omega = 1$, the second order design using Algorithm 1 will also converge to zero since

$$Q_1(e^{i\omega}) + Q_2(e^{i\omega}) = \frac{5}{4}Q(e^{i\omega}) - \frac{(Q(e^{i\omega}))^2}{4} \approx 1$$

6.2 Analysis of resulting design

A natural way of evaluating the second order ILC design is to compare it with the corresponding first order design, both from a performance as well as a robustness point of view.

6.2.1 Performance and robustness for a first order ILC: From Section 5.1 it is clear that the first order ILC algorithm will give an exponential convergence in the frequency domain. The rate will depend on the frequency function $F(e^{i\omega}) = Q(e^{i\omega})\big(1 - L(e^{i\omega})T_u(e^{i\omega})\big)$.

Assume that there is a relative model uncertainty $\Delta_r(e^{i\omega})$,

$$F(e^{i\omega}) = \bar{F}(e^{i\omega})(1 + \Delta_r(e^{i\omega})), \quad |\Delta_r(e^{i\omega})| < \gamma(\omega)$$

where $\bar{F}(e^{i\omega})$ is the nominal value. A sufficient condition for stability becomes

$$\gamma(\omega) < \frac{1}{|\bar{F}(e^{i\omega})|} - 1$$

If there is an absolute uncertainty,

$$F(e^{i\omega}) = \bar{F}(e^{i\omega}) + \Delta_a(e^{i\omega}), \quad |\Delta_a(e^{i\omega})| < \gamma(\omega)$$

the corresponding sufficient criterion for robust stability becomes

$$\gamma(\omega) < 1 - |F(e^{i\omega})|$$

Now the proposed second order ILC design can be compared with the first order ILC scheme.

6.2.2 Eigenvalue based design: When using Algorithm 1 the eigenvalues are given by

$$\lambda_{(1,2),\omega} = \frac{Q(e^{i\omega})(1 - L(e^{i\omega})T_u(e^{i\omega}))}{8}(5 \pm 3)$$

Obviously this means that if $F(e^{i\omega}) = Q(e^{i\omega})(1 - L(e^{i\omega})T_u(e^{i\omega}))$ is the result of the original first order ILC design, the nominal eigenvalues are $\lambda_{1,\omega} = F(e^{i\omega})$ and $\lambda_{2,\omega} = \frac{1}{4}F(e^{i\omega})$. From a robustness perspective this is exactly the same result as for the first order ILC algorithm since the absolute amplitude margin for the second order case,

$$1 - \max(|\lambda_{1,\omega}|, |\lambda_{2,\omega}|) = 1 - |F(e^{i\omega})|$$

is exactly the same as the one of the first order.

Important is also to consider the performance that can be achieved compared to the first order ILC design. For the second order design it follows from Theorem 2 that

$$\frac{|\tilde{U}^2_{k,\omega}|}{|\tilde{U}^2_{0,\omega}|} = |F(e^{i\omega})|^k \frac{1}{3 \cdot 4^k}|4^{k+1} - 1 - F(e^{i\omega})(4^k - 1)|$$

To compare the first and the second order ILC algorithms consider,

$$\frac{|\tilde{U}^2_{k,\omega}|}{|\tilde{U}^2_{0,\omega}|} - \frac{|\tilde{U}^1_{k,\omega}|}{|\tilde{U}^1_{0,\omega}|}$$

Using the following bound

$$1 < \frac{1}{3 \cdot 4^k}|4^{k+1} - 1 - F(e^{i\omega})(4^k - 1)| < \frac{5}{3}$$

which is found using the triangular inequality and the fact that $|F(e^{i\omega})| < 1$. It is obvious that the second order ILC algorithm designed using the eigenvalue based design will never work better compared to the first order ILC design on which it is based.

7 Experiment

The theory developed in the previous sections can now be applied on a design example for a real industrial system. The system, an ABB IRB1400 industrial robot, is depicted in Figure 1. For a more thorough description of the technical part of the experimental setup see [13].

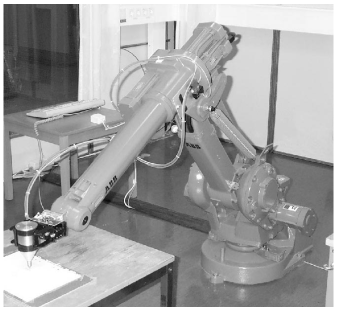


Figure 1: The ABB IRB1400 manipulator.

```
%% starting at p1
movec p2,p3,v60;
movec p4,p5,v60;
movec p6,p3,v60;
movec p4,p5,v60;
movec p7,p1,v60;
```

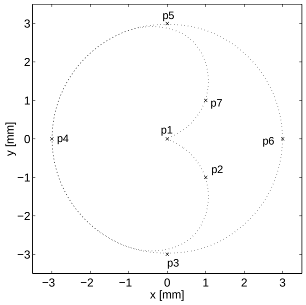


Figure 2: The program used to produce the trajectory used in the example (left) and the resulting trajectory translated such that the origin coincide with p1 (right).

In this example ILC is applied to three of the robot's six joints. Each of the three joints is modeled as a transfer operator description from the ILC control input to the measured motor position on the robot, i.e., T_u in (1). It should be stressed that T_u is in fact a model of a closed loop system. The conventional feedback controller in the S4C control system is working in parallel with the ILC scheme. The models, T_u are calculated using *System Identification Toolbox* [8] and are given by,

$$\begin{aligned} \hat{T}_{u_1}(q) = \hat{T}_{u_2}(q) &= \frac{0.1q^{-1}}{1 - 0.9q^{-1}} \\ \hat{T}_{u_3}(q) &= \frac{0.13q^{-1}}{1 - 0.87q^{-1}} \end{aligned} \tag{12}$$

7.1 Description of the experiment

The experiment is done on the ABB IRB1400 robot in the research lab of Division of Automatic Control at Linköping University. In Figure 2 the program used in the experiment is shown together with the resulting trajectory on the arm-side of the robot. The instruction `movec p2,p3,v60` refers to an instruction that produces an arc on the arm-side of the robot. The arc starts from the current position, not explicitly stated, and goes through the points `p2` and `p3`. The speed along the path is programmed to be 60 mm/s. Actual position of `p1` in the base coordinate system is x = 1300 mm, y = 100 mm, and z = 707 mm. The configuration of the robot is also shown in Figure 1.

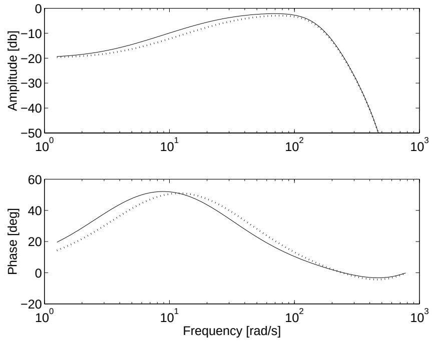


Figure 3: The result from the design of the first order ILC algorithm. $Q(e^{i\omega})(1 - L(e^{i\omega})T_{u_i}(e^{i\omega}))$, for $i = 1, 2$ (solid line) and $i = 3$ (dotted).

7.2 ILC design

The design of the first order ILC scheme is based on an algorithm discussed in, e.g., [13]. The procedure will only be briefly described here.

The filter L is chosen as

$$L(q) = 0.9q^2 \tag{13}$$

The corresponding Q filter is chosen as $Q(e^{i\omega}) = Q_{1/2}(e^{i\omega})Q_{1/2}(e^{-i\omega})$, i.e., a zero-phase filter, with $Q_{1/2}(q)$ as a second order Butterworth filter with cut-off frequency 0.2 of the Nyquist frequency. The resulting nominal $Q(e^{i\omega})(1 - L(e^{i\omega})T_{u_i}(e^{i\omega}))$ are depicted in Figure 3 and it is obvious that the convergence criterion in (7), is fulfilled for all the three joints.

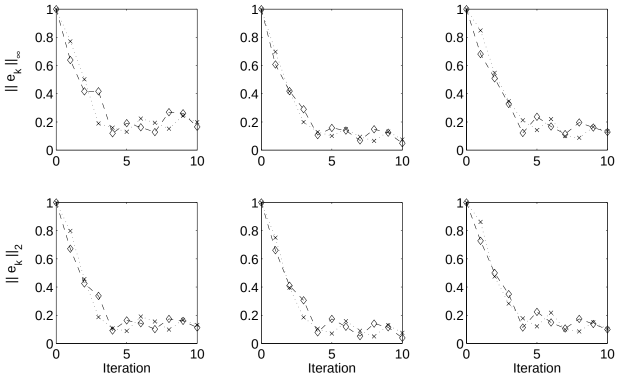
The second order ILC algorithm is designed based on the first order ILC design according to Algorithm 1.

7.3 Results from the experiments

The result from the experiments can be evaluated from two different points of view. First the result achieved on the motor side can be studied. This is the measure used by the ILC algorithms and it is the error in this measure that is supposed to be minimized.

7.3.1 Motor-side: The two ILC algorithms have run for 10 iterations. Since in the first iteration ILC is not applied, $u_0 \equiv 0$, the same circle has been done 11 times. In Figure 4 the resulting normalized ∞-norm and 2-norm of the error on the motor-side is shown. In the first iteration all the algorithms use the same updating equations and should,in theory, therefore also reach the same level of error. As can be seen in Figure 4 this is not the case. The ILC designed according to Algorithm 1 gives a lower value of the error in the first iteration. In the second iteration the size of the error is again about the same and after the fifth iteration the error stabilizes on a level where $\|e_k\|_\infty$ is about 15 % of the initial value.

7.3.2 Arm-side: When evaluating the result of the ILC iterations on the arm-side of the robot a pen is used, as shown in Figure 1. The resulting drawings are not so easy to evaluate since the pen in itself produces a quite thick line but in Figure 5 the resulting circles are shown for the designs in the first 5 iterations (iteration 0 to 5, from left to right). A conclusion that can be drawn from the result in Figure 5 is that they give a similar result also on the arm-side.



(a) Motor 1. *(b)* Motor 2. *(c)* Motor 3.

Figure 4: The error in ∞-norm and 2-norm for the two designs. First order ILC design ($\times$) and Algorithm 1 ($\diamond$).

It is also possible to evaluate the result from the ILC experiments on the arm-side by a transformation of the measured motor angles to the arm side using the forward kinematic model of the IRB1400, see e.g., [11]. In Figure 6 the result from this transformation is shown.

8 Conclusion

The analysis of second order ILC systems based on linear iterative systems is very promising and gives a lot of insight into the behavior of second order ILC schemes. From the results presented here it is not possible to say that a second order ILC algorithm does better than a first order algorithm. From the analysis and, in fact, also the experiments it is evident that it works as well as the first order design. Some facts are however important to stress when thinking of moving from a first order ILC design to a second order ILC design.

- The second order design should not use more information about the system than the first order design.
- The amount of memory required for the second order ILC scheme, as implemented in this paper, is double the amount used by the corresponding first order ILC scheme.
- One aspect that has not been considered in this

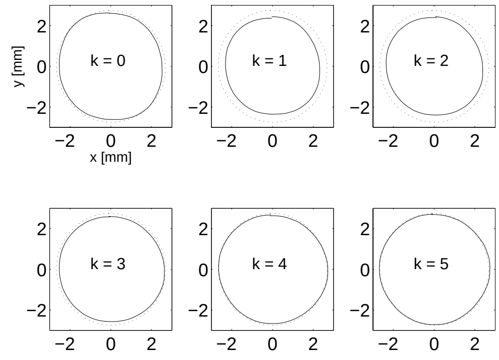
report but that can make the second order ILC scheme very competitive is when there is an uncertainty in the plant that makes the plant different between the iterations. The second order algorithm can smooth also the behavior of the system by using the control and the error signal from more than one iteration.

Further work in the area could be to consider the effects of nonlinearities on the resulting control signals and the resulting error. For example, Coloumb friction is a nonlinear effect that is always present in real servo systems and a comparison between a first and second order ILC algorithms for dealing with this could be worth to consider.

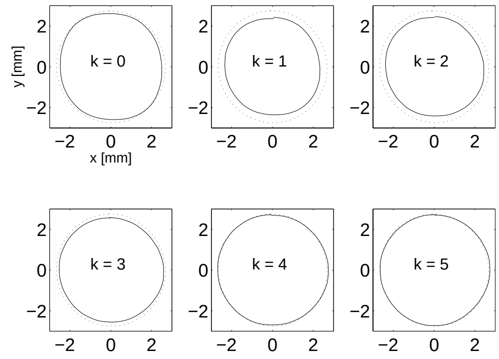
(a) First order ILC.

(b) Second order ILC.

Figure 5: Result on arm-side of the ILC iterations 0 to 5 (from left to right).



(a) First order ILC.



(b) Second order ILC.

Figure 6: Result from kinematic transformation from motor to arm.

References

[1] N. Amann, D. H. Owens, and E. Rogers. 2d systems theory applied to learning control systems. In *Proc. of the 33rd IEEE Conf. on Decision and Control*, Lake Buena Vista, FL, USA, Dec 1994.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[3] Z. Bien and K.M. Huh. Higher-order iterative learning control algorithm. In *IEE Proceedings*, volume 136, pages 105 – 112, May 1989.

[4] Z. Bien and J.-X. Xu. *Iterative Learning Control: Analysis, Design, Integration and Application*. Kluwer Academic Publishers, 1998.

[5] G. Casalino and G. Bartolini. A learning procedure for the control of movements of robotic manipulators. In *IASTED Symposium on Robotics and Automation*, pages 108–111, 1984.

[6] Y. Chen, Z. Gong, and C. Wen. Analysis of a high order iterative learning control algorithm for uncertain nonlinear systems. *Automatica*, 34(3):345–353, March 1998.

[7] J.J. Craig. Adaptive control of manipulators through repeated trials. In *Proc. of ACC*, San Diego, CA, June 1984.

[8] L. Ljung. *System Identification Toolbox - For Use with Matlab*. The MathWorks Inc., 1995.

[9] K. L. Moore. *Iterative Learning Control for Deterministic Systems*. Advances in Industrial Control. Springer-Verlag, 1993.

[10] K. L. Moore. Iterative learning control - an expository overview. *Applied and Computational Controls, Signal Processing and Circuits*, 1998.

[11] M. Norrlöf. Modeling of industrial robots. Technical Report LiTH-ISY-R-2208, Department of Electrical Engineering, Linköping University, Dec 1999.

[12] M. Norrlöf. Analysis of a second order iterative learning controller. Technical Report LiTH-ISY-R-2181, Department of Electrical Engineering, Linköping University, Feb 2000.

[13] M. Norrlöf. *Iterative Learning Control: Analysis, Design, and Experiments*. PhD thesis, Linköping University, Linköping, Sweden, 2000. Linköping Studies in Science and Technology. Dissertations No. 653.

[14] M. Norrlöf and S. Gunnarsson. A frequency domain analysis of a second order iterative learning control algorithm. In *Proc. of the 38th IEEE Conference on Decision and Control*, Pheonix, Arizona, USA, Dec 1999.

[15] E. Rogers and D. H Owens. *Stability Analysis for Linear Repetitive Processes*, volume 175 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1992.