An adaptive Iterative Learning Control algorithm with experiments on an industrial robot

Mikael Norrlöf

Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
E-mail: mino@isy.liu.se

17th June 2002

AUTOMATIC CONTROL
COMMUNICATION SYSTEMS
LINKÖPING

Report no.: LiTH-ISY-R-2434
Submitted to IEEE Transactions on Robotics and Automation, April, 2002

Abstract

An adaptive Iterative Learning Control (ILC) algorithm based on an estimation procedure using a Kalman filter and an optimization of a quadratic criterion is presented. It is shown that by taking the measurement disturbance into consideration the resulting ILC filters become iteration varying. Results from experiments on an industrial robot show that the algorithm is successful also in an application.

Keywords: Iterative learning control, disturbance rejection, synthesis, robot application

An adaptive Iterative Learning Control algorithm with experiments on an industrial robot

Mikael Norrlöf, *IEEE Member*

Abstract— **An adaptive Iterative Learning Control (ILC) algorithm based on an estimation procedure using a Kalman filter and an optimization of a quadratic criterion is presented. It is shown that by taking the measurement disturbance into consideration the resulting ILC filters become iteration varying. Results from experiments on an industrial robot show that the algorithm is successful also in an application.**

Keywords— **Iterative learning control, disturbance rejection, synthesis, robot application**

I. Introduction

Iterative Learning Control is a well established method for control of repetitive processes. It is in general considered to be an approach for trajectory tracking and this is how it is usually described in the literature, see for example the surveys [1], [2], [3]. In this paper we will use ILC in a different setting, applying ILC for disturbance rejection (see also [4]). In Section V we will show how we can apply the results in a standard tracking application for ILC. Disturbance rejection aspects of ILC have also been covered earlier in e.g., [5], [6], [7], where disturbances such as initial state disturbances and measurement disturbances are addressed.

In Figure 1 the structure used in the disturbance rejection formulation to ILC is shown as a block diagram.

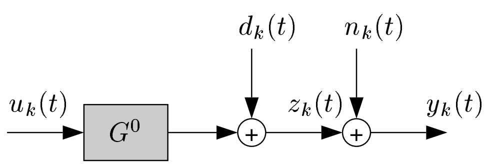


Fig. 1. A system with input $u_k(t)$ and two unknown disturbances, $d_k(t)$, and $n_k(t)$, acting on the output of the system G^0.

The goal in ILC is to, iteratively, find the input to a system such that some error is minimized. In the disturbance rejection formulation, the goal becomes to find an input $u_k(t)$ such that the output $z_k(t)$ is minimized. If the system is known and invertible, and the disturbance $d_k(t)$ is known, then the obvious approach would be to filter $d_k(t)$ through the inverse of the system and use the resulting $u_k(t)$ as a control input. This means that the optimal input looks like,

$$u_k(t) = -(G^0)^{-1} d_k(t)$$

Different aspects of this approach to ILC is considered in the paper. Results from using the methods on an industrial robot are also presented.

II. A state space based approach to ILC

A. Matrix description of the system

An ILC system is characterized by the fact that it is only defined over a finite interval of time. If the sampling time is equal to one, this means that $0 \le t \le n-1$. This is also the reason why it is possible to write the system description in matrix form

$$\begin{aligned} \boldsymbol{z}_k &= \boldsymbol{G}^0 \boldsymbol{u}_k + \boldsymbol{d}_k \\ \boldsymbol{y}_k &= \boldsymbol{z}_k + \boldsymbol{n}_k \end{aligned} \tag{1}$$

with

$$\boldsymbol{d}_{k+1} = \boldsymbol{d}_k + \boldsymbol{\Delta}_{d_k} \tag{2}$$

where $\boldsymbol{z}_k, \boldsymbol{u}_k, \boldsymbol{d}_k, \boldsymbol{y}_k, \boldsymbol{n}_k, \boldsymbol{\Delta}_{d_k} \in \mathbb{R}^n$ and $\boldsymbol{G}^0 \in \mathbb{R}^{n\times n}$. The vector $\boldsymbol{z}_k$ is defined as,

$$\boldsymbol{z}_k = \begin{bmatrix} z_k(0) & z_k(1) & \dots & z_k(n-1) \end{bmatrix}^T$$

with the other vectors defined accordingly. The matrix $\boldsymbol{G}^0$ is for a causal system a lower triangular matrix and if the system is linear time invariant it also becomes Toeplitz. This particular description of ILC systems has been exploited earlier in for example, the work by Moore [1], [8], and also in Phan et al. [9] and Lee et al. [10].

We assume that $\boldsymbol{d}_k$ and $\boldsymbol{n}_k$ are random with covariance matrices for $\boldsymbol{\Delta}_{d_k}$ and $\boldsymbol{n}_k$ given by $\boldsymbol{R}_{\Delta_d,k}$ and $\boldsymbol{R}_{n,k}$ respectively. In the following the components in $\boldsymbol{n}_k$ and $\boldsymbol{\Delta}_{d_k}$, $n_k(t)$ and $\Delta_{d_k}(t)$, are considered to be white stationary stochastic processes.

Using the updating formula for the disturbance $\boldsymbol{d}_k$ from (2) and a model $\boldsymbol{G}$ with a relative model error,

$$\boldsymbol{G}^0 = \boldsymbol{G}(I + \boldsymbol{\Delta}_G) \tag{3}$$

it is possible to rewrite (1) as

$$\begin{aligned} \boldsymbol{z}_{k+1} &= \boldsymbol{z}_k + \boldsymbol{G}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \boldsymbol{G}\boldsymbol{\Delta}_G(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \boldsymbol{\Delta}_{d_k} \\ \boldsymbol{y}_k &= \boldsymbol{z}_k + \boldsymbol{n}_k \end{aligned} \tag{4}$$

The last two terms in the first equation can be considered as disturbances since they are both unknown. It is however known that the first one depends on the difference between two consecutive control signals. If the model uncertainty is small and/or the updating speed of the control signal is slow, this disturbance will have a small effect on the resulting system.

B. Estimation procedure

A linear estimator for the system described in (4) is

$$\hat{\boldsymbol{z}}_{k+1} = \hat{\boldsymbol{z}}_k + \boldsymbol{G}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{z}}_k) \tag{5}$$

where $\boldsymbol{K}_k$ is the gain of the estimator. By applying standard Kalman filter techniques, see for example [11], the estimation procedure for $\hat{\boldsymbol{z}}_k$ becomes

$$\hat{\boldsymbol{z}}_{k+1} = \hat{\boldsymbol{z}}_k + \boldsymbol{G}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{z}}_k) \tag{6a}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k(\boldsymbol{P}_k + \widehat{\boldsymbol{R}}_{n,k})^{-1} \tag{6b}$$

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k + \widehat{\boldsymbol{R}}_{\Delta_d,k} - \boldsymbol{P}_k(\boldsymbol{P}_k + \widehat{\boldsymbol{R}}_{n,k})^{-1}\boldsymbol{P}_k \tag{6c}$$

where it is assumed that $\boldsymbol{\Delta}_{d_k}$ and $\boldsymbol{n}_k$ are uncorrelated. $\widehat{\boldsymbol{R}}_{n,k}$ and $\widehat{\boldsymbol{R}}_{\Delta_d,k}$ are estimates of the true covariance matrices. Compare also the discussion in Section II-A.

C. An optimization based approach to ILC

Consider the following criterion for control of (1),

$$J_k = \boldsymbol{z}_k^T \boldsymbol{W}_z \boldsymbol{z}_k + \boldsymbol{u}_k^T \boldsymbol{W}_u \boldsymbol{u}_k \tag{7}$$

By minimizing (7) it is possible to find an optimal input to the system, with respect to the criterion. This has been studied in

Mikael Norrlöf is with the Department of Electrical Engineering, Linköpings universitet, SE-581 83 Linköping, Sweden. Phone: +46 13 282704 Fax: +46 13 282622 Email: mino@isy.liu.se

for example [3], [12], [13], [14], [15] and [10] but in contrast to most of the approaches in the literature, the term containing $\boldsymbol{u}_k - \boldsymbol{u}_{k-1}$ is not included in the criterion here.

By using, in (7), the definition of $\boldsymbol{z}_k$ from (1) and taking the derivative with respect to $\boldsymbol{u}_k$ it follows that

$$\frac{\partial J_k}{\partial \boldsymbol{u}_k} = ((\boldsymbol{G}^0)^T \boldsymbol{W}_z \boldsymbol{G}^0 + \boldsymbol{W}_u)\boldsymbol{u}_k + (\boldsymbol{G}^0)^T \boldsymbol{W}_z \boldsymbol{d}_k$$

Now solve for $\boldsymbol{u}_k$ when $\frac{\partial J_k}{\partial \boldsymbol{u}_k} = 0$. This leads to

$$\boldsymbol{u}_{k+1}^* = -((\boldsymbol{G}^0)^T \boldsymbol{W}_z \boldsymbol{G}^0 + \boldsymbol{W}_u)^{-1} (\boldsymbol{G}^0)^T \boldsymbol{W}_z \boldsymbol{d}_{k+1} \tag{8}$$

where the * denotes the optimal input.

If $\boldsymbol{W}_u = 0$ and $\boldsymbol{d}_{k+1}$ is known, then the updating scheme for the control $\boldsymbol{u}_k$ becomes

$$\boldsymbol{u}_{k+1}^* = -(\boldsymbol{G}^0)^{-1} \boldsymbol{d}_{k+1} \tag{9}$$

which is also described in Section I. Note that this expression actually contains a feedforward from the disturbance $\boldsymbol{d}_{k+1}$. From a practical point of view (9) is not very useful since when $\boldsymbol{u}_{k+1}$ is calculated, $\boldsymbol{d}_{k+1}$ is in general not available. If $\boldsymbol{d}$ does not change as a function of iteration it will however work since old estimates of $\boldsymbol{d}$ can be used. In practice the control solution can not (of course) use the true system description. If instead a model of the system is available, the control signal $\boldsymbol{u}_{k+1}$ can be calculated as

$$\boldsymbol{u}_{k+1} = -(\boldsymbol{G}^T \boldsymbol{W}_z \boldsymbol{G} + \boldsymbol{W}_u)^{-1} \boldsymbol{G}^T \boldsymbol{W}_z \hat{\boldsymbol{d}}_{k+1} \tag{10}$$

In (10) it is also taken into account that the true $\boldsymbol{d}_{k+1}$ is not available directly as a measured signal. An estimate of $\boldsymbol{d}_{k+1}$ can be found as

$$\hat{\boldsymbol{d}}_{k+1} = \hat{\boldsymbol{z}}_{k+1} - \boldsymbol{G}\boldsymbol{u}_{k+1} \tag{11}$$

which means that the expression for $\boldsymbol{u}_{k+1}$ can be simplified

$$\boldsymbol{u}_{k+1} = -\boldsymbol{W}_u^{-1} \boldsymbol{G}^T \boldsymbol{W}_z \hat{\boldsymbol{z}}_{k+1} \tag{12}$$

by using (10) and (11). This can be plugged into the observer in (5) resulting in

$$\hat{\boldsymbol{z}}_{k+1} = \hat{\boldsymbol{z}}_k + (I + \boldsymbol{G}\boldsymbol{W}_u^{-1}\boldsymbol{G}^T\boldsymbol{W}_z)^{-1} \boldsymbol{K}_k (\boldsymbol{y}_k - \hat{\boldsymbol{z}}_k) \tag{13}$$

Together with (12) and the calculation of $\boldsymbol{K}_k$ from the previous section, this gives an ILC scheme with two iterative updating formulas, including the one for $\boldsymbol{P}_k$. Compared to the traditional ILC schemes,

$$u_{k+1}(t) = Q(q)(u_k(t) + L(q)e_k(t)) \tag{14}$$

the iterative behavior of the ILC algorithm has moved from the updating of the control signal to the estimator.

D. Relations to other ILC updating schemes

Consider the case when the estimated covariances are $\widehat{\boldsymbol{R}}_{n,k} = \hat{r}_{n,k} \cdot I$ and $\widehat{\boldsymbol{R}}_{\Delta_d,k} = \hat{r}_{\Delta_d,k} \cdot I$. Assume that the estimator is calculated according to a time varying Kalman filter as described in Section II-B. Note that in the calculation of $\boldsymbol{P}_k$ the measured values of $\boldsymbol{y}_k$ are not utilized. Instead the value of $\boldsymbol{P}_k$ is completely dependent on the initial value, $\boldsymbol{P}_0$. This initial choice indicates how well the initial estimate $\hat{\boldsymbol{z}}_0$ describes the real value.

Assume that $\boldsymbol{P}_0 = p_0 \cdot I$, this means that $\boldsymbol{K}_k$ and $\boldsymbol{P}_k$ will be equal to $\kappa_k \cdot I$ and $p_k \cdot I$ respectively. Since $\boldsymbol{K}_k$ is an identity matrix times a scalar the matrix $\boldsymbol{K}_k$ commutes with all other matrices. In particular, this means that it is possible to rewrite (13) according to

$$\boxed{\begin{aligned} \boldsymbol{u}_{k+1} &= (I - (I + \boldsymbol{W}_u^{-1}\boldsymbol{G}^T\boldsymbol{W}_z\boldsymbol{G})^{-1}\boldsymbol{K}_k)\boldsymbol{u}_k \\ &\quad - \boldsymbol{W}_u^{-1}\boldsymbol{G}^T\boldsymbol{W}_z(I + \boldsymbol{G}\boldsymbol{W}_u^{-1}\boldsymbol{G}^T\boldsymbol{W}_z)^{-1}\boldsymbol{K}_k\boldsymbol{y}_k \end{aligned}} \tag{15}$$

where (12) is used together with the fact that

$$\begin{aligned} &\boldsymbol{W}_u^{-1} G^T \boldsymbol{W}_z (I + G\boldsymbol{W}_u^{-1} G^T \boldsymbol{W}_z)^{-1} \boldsymbol{K}_k \\ &\quad = (I + \boldsymbol{W}_u^{-1} G^T \boldsymbol{W}_z G)^{-1} \boldsymbol{K}_k \boldsymbol{W}_u^{-1} G^T \boldsymbol{W}_z \end{aligned}$$

If the weights $\boldsymbol{W}_u$ and $\boldsymbol{W}_z$ are chosen such that $\boldsymbol{W}_u = I$ and $\boldsymbol{W}_z = \zeta \cdot I$ and ζ is chosen very large then the resulting updating equation becomes

$$\boldsymbol{u}_{k+1} \approx \boldsymbol{u}_k - \kappa_k \boldsymbol{G}^{-1} \boldsymbol{y}_k \tag{16}$$

which is recognized as a standard approach (although the gain κ_k is non standard), see e.g., [1], [4].

As a result of the fact that $\widehat{\boldsymbol{R}}_{n,k}$, $\widehat{\boldsymbol{R}}_{\Delta_d,k}$, and $\boldsymbol{P}_0$ are all equal to a scalar times an identity matrix it follows that (6b) and (6c) can be written as scalar equations,

$$\begin{aligned} \kappa_k &= \frac{p_k}{p_k + \hat{r}_{n,k}} \\ p_{k+1} &= p_k + \hat{r}_{\Delta_d,k} - \frac{p_k^2}{p_k + \hat{r}_{n,k}} = \frac{p_k \hat{r}_{n,k}}{p_k + \hat{r}_{n,k}} + \hat{r}_{\Delta_d,k} \end{aligned}$$

Assume that $\hat{r}_n$ and $\hat{r}_{\Delta_d}$ do not depend on k. Then it is possible to find the limit value, p_∞,

$$p_\infty = \frac{\hat{r}_{\Delta_d}}{2} \left(1 + \sqrt{1 + 2\frac{\hat{r}_n}{\hat{r}_{\Delta_d}}} \right) \tag{17}$$

Note that the value of p_∞ depends on the actual value of $\hat{r}_{\Delta_d}$ while for κ_∞ it is only the value of $\frac{\hat{r}_{\Delta_d}}{\hat{r}_n}$ that has an influence. Multiplying both $\hat{r}_{\Delta_d}$ and $\hat{r}_n$ with the same factor will not change the value of κ_∞.

If it is assumed that $\boldsymbol{d}_k = \boldsymbol{d}$, i.e., $\hat{r}_{\Delta_d} = 0$, it is clear that $p_\infty = 0$ which also implies that $\kappa_\infty = 0$. More important however is to study the transient behavior of p_k and κ_k for this case. If the initial guess of $\hat{\boldsymbol{z}}_0$ is not so reliable it is reasonable to assume that p_0 is chosen as a large number. If $p_0 \gg \hat{r}_n$ this means that $\kappa_0 \approx 1$ and since

$$p_{k+1} = \frac{p_k \hat{r}_n}{p_k + \hat{r}_n} \tag{18}$$

it follows that $p_1 \approx \hat{r}_n$ which in turn implies that $k_1 \approx \frac{1}{2}$. By considering (18) for general k it becomes clear that, in fact, $p_k \approx \frac{\hat{r}_n}{k}$ for all $k > 0$ and hence $\kappa_k \approx \frac{1}{k+1}$. For ILC applied to a linear time invariant system having white measurement noise the optimal ILC updating law will use the inverse system model as a learning filter and have a decreasing gain.

III. An adaptive algorithm for ILC

The calculations of $\boldsymbol{P}_k$ and $\boldsymbol{K}_k$ in the time varying Kalman filter do not depend on the measurements made upon the system. In this section a possible extension to the algorithm presented in the previous sections is given. The algorithm takes advantage of the measurements from the system and use them to adapt a measure of the variability of the system disturbance, $\widehat{\boldsymbol{R}}_{\Delta,k}$. The algorithm is adaptive since the value of $\boldsymbol{K}_k$ will depend on the variability measure through $\boldsymbol{P}_k$.

To explain the idea behind the measure of variability used in the algorithm first note that the system model $\boldsymbol{G}$ does not capture the true system dynamics perfectly. Instead the relation given by (4) describes the true system in terms of the model and the uncertainty.

The idea is to use

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \boldsymbol{G}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \underbrace{\boldsymbol{G}\boldsymbol{\Delta}_G(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \boldsymbol{\Delta}_{d_k}}_{\Delta}$$

and find a measure of the size of the variation of Δ. The following equation gives this measure

$$\hat{r}_{\Delta,k} = \frac{1}{n-1}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k)^T \hat{\boldsymbol{\Delta}}_G^T \boldsymbol{G}^T \boldsymbol{G} \hat{\boldsymbol{\Delta}}_G (\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \hat{r}_{\Delta_d}$$

where $\hat{\boldsymbol{\Delta}}_G$ is an estimate of the true model uncertainty and $\hat{r}_{\Delta_d}$ is an estimate of the variance of $\boldsymbol{\Delta}_{d_k}$. The algorithm can now be formulated.

Algorithm 1 (Adaptive optimization based ILC)

1. Design an ILC updating equation using the LQ design in Section II-C.

2. Assume $\widehat{\boldsymbol{R}}_{\Delta_d}$ and $\widehat{\boldsymbol{R}}_n$ diagonal with the diagonal elements equal to $\hat{r}_{\Delta_d}$ and $\hat{r}_n$ respectively, i.e., $\widehat{\boldsymbol{R}}_{\Delta_d} = \hat{r}_{\Delta_d} \cdot I$ and $\widehat{\boldsymbol{R}}_n = \hat{r}_n \cdot I$. Choose $\hat{r}_{\Delta_d}$ and $\hat{r}_n$ from physical insight or such that p_∞ in (17) and the corresponding κ_∞ get the desired values.

3. Let $\hat{\boldsymbol{z}}_0 = 0$.

4. Choose an initial value for p_0. This can be a large number since p_k will converge to $\approx \hat{r}_{\Delta,k}$ already after one iteration.

5. Implementation of the ILC algorithm:

(a) Let $k = 0$, and $\boldsymbol{u}_0 = -\boldsymbol{W}_u^{-1}\boldsymbol{G}^T\boldsymbol{W}_z\hat{\boldsymbol{z}}_0$.

(b) Apply $\boldsymbol{u}_k$ and measure $\boldsymbol{y}_k$.

(c) Calculate,

$$\begin{aligned}
\kappa_k &= \frac{p_k}{p_k + \hat{r}_n}\\
\hat{\boldsymbol{z}}_{k+1} &= \hat{\boldsymbol{z}}_k + (I + \boldsymbol{G}\boldsymbol{W}_u^{-1}\boldsymbol{G}^T\boldsymbol{W}_z)^{-1}\kappa_k(\boldsymbol{y}_k - \hat{\boldsymbol{z}}_k)\\
\boldsymbol{u}_{k+1} &= -\boldsymbol{W}_u^{-1}\boldsymbol{G}^T\boldsymbol{W}_z\hat{\boldsymbol{z}}_{k+1}\\
\hat{r}_{\Delta,k} &= \frac{1}{n-1}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k)^T \hat{\boldsymbol{\Delta}}_G^T \boldsymbol{G}^T \boldsymbol{G} \hat{\boldsymbol{\Delta}}_G (\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) + \hat{r}_{\Delta_d}\\
p_{k+1} &= \frac{p_k \hat{r}_n}{p_k + \hat{r}_n} + \hat{r}_{\Delta,k}
\end{aligned}$$

(d) Let $k = k + 1$. Start again from (b).

The important properties of the proposed algorithm and this, especially, includes stability and performance are discussed in [4]. The main result is to show boundedness of the estimate $\hat{\boldsymbol{z}}_k$ which implies that the resulting ILC algorithm is stable. The analysis in [4] covers two important cases, the first is when the system, $\boldsymbol{G}^0$, is iteration invariant but uncertain and the second is when the system is iteration variant and uncertain.

The idea of using an optimization based ILC updating equation and an estimation procedure is also covered in Chapter 9 (written by Lee and Lee) of [3]. Their solution does not have the same criterion in the control design and their observer is not adaptive as is the case here. Adaptive ILC algorithms are also covered in, e.g., [16], [17], [18]. Notice that many proposed adaptive ILC algorithms are combination of adaptive feedback controllers and non-adaptive ILC algorithms. The adaptive ILC algorithm presented in this paper is instead truly adaptive and does not say anything about the feedback control solution of the system.

IV. Design and implementation issues for the optimization based approach to ILC

The design process involves a lot of steps and there are many degrees of freedom in the design. The design parameters involved are:

- In the LQ design,
  - $\boldsymbol{G} \in \mathbb{R}^{n\times n}$, $\boldsymbol{W}_z \in \mathbb{R}^{n\times n}$, and $\boldsymbol{W}_u \in \mathbb{R}^{n\times n}$
- In the Kalman filter,
  - $\boldsymbol{G} \in \mathbb{R}^{n\times n}$, $p_0 \in \mathbb{R}$, $\hat{r}_{\Delta_d} \in \mathbb{R}$, $\hat{r}_n \in \mathbb{R}$, and $\hat{\boldsymbol{\Delta}}_G \in \mathbb{R}^{n\times n}$

The model $\boldsymbol{G}$ is used in both the LQ design and the Kalman filter. By just considering the number of possibilities that are offered by these parameters it might seem that the usefulness of the proposed scheme can be questioned. From a user's point of view it is important that the number of parameters is small and that the effects of the parameters are easy to understand. Note that the suggested parameters, given above, also imply a simplification compared to the originally proposed algorithm. Only scalar $\boldsymbol{P}_k = p_k \cdot I$ and $\boldsymbol{K}_k = \kappa_k \cdot I$ are considered here. The effect of the different design parameters on the design is discussed next.

A. Design scheme

Assume that the model of the system, $\boldsymbol{G} \in \mathbb{R}^{n\times n}$, is available from an identification experiment. This experiment can also give an idea on which kind of uncertainties are present in the model, i.e., the size of $\hat{\boldsymbol{\Delta}}_G$. Methods like the model error modeling technique by Ljung [19] give for example this information. In many traditional design schemes for ILC the updating equation is,

$$\boldsymbol{u}_{k+1} = \boldsymbol{Q}(\boldsymbol{u}_k + \boldsymbol{L}\boldsymbol{e}_k)$$

where $\boldsymbol{u}_k, \boldsymbol{e}_k \in \mathbb{R}^n$ and $\boldsymbol{Q}, \boldsymbol{L} \in \mathbb{R}^{n\times n}$. Often it is suggested that, for robustness of the ILC algorithm, $\boldsymbol{Q}$ should be chosen as a realization of a low pass filter. This makes the ILC method robust against model errors at high frequencies, where usually the model of the system does not capture the true dynamics very well. The LQ solution of the ILC problem can take this into consideration by introducing a kind of frequency domain weighting in the optimization criterion (7). This is done by using the fact that the matrices $\boldsymbol{W}_z$ and $\boldsymbol{W}_u$ do not have to be diagonal. With a frequency domain perspective to the optimization problem, high frequencies in the control signal $\boldsymbol{u}_k$ should have a higher weight in the criterion than low frequencies. This can be done by choosing the matrix $\boldsymbol{W}_u^{-1}$ as a realization of a zero phase low pass filter with cut-off frequency at the desired bandwidth of the ILC algorithm. To create such a matrix let $\boldsymbol{H}$ be a lower triangular Toeplitz matrix with the first column being the n first Markov parameters of a low pass filter, e.g., a Butterworth filter. Next define $\boldsymbol{W}_u = (\boldsymbol{H}\boldsymbol{H}^T)^{-1}$, i.e., as the inverse of the zero phase low pass filter $\boldsymbol{H}\boldsymbol{H}^T$. The matrix $\boldsymbol{W}_z$ is here simply chosen as a scalar times an identity matrix, $\boldsymbol{W}_z = \zeta \cdot I$, and the value of ζ will decide how much the ILC scheme should try to resemble the inverse system approach, as was also discussed in Section II-D.

For the Kalman filter, the system model $\boldsymbol{G}$ and an estimated model uncertainty $\hat{\boldsymbol{\Delta}}_G$ are supposed to be available from the identification experiments. The algorithm is not sensitive to the initial value of p_0 as was noted in Section II-D. If the value is initially set to be a large number the value of κ_0 will be close to one and the next value of p_1 will be, $p_1 \approx \hat{r}_n + \hat{r}_{\Delta,0}$. This shows that the initial value is not so important for the behavior of the algorithm as long as it is large enough.

The values of $\hat{r}_{\Delta_d}$ and $\hat{r}_n$ are still to be chosen. As was shown in Section II-D it is true that asymptotically, if $\|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\|$

becomes small, it is only the value of $\frac{\hat{r}_{\Delta_d}}{\hat{r}_n}$ that has an impact on the value of κ_k. To decide the value of the two parameters the following strategy will be used here: Let the value of $\hat{r}_n$ be based on physical knowledge of the process and adjust $\hat{r}_{\Delta_d}$ such that the limit value of p_k, p_∞ in (17), and the corresponding κ_∞ have the right value. Note that it is important that the value of $\hat{r}_{\Delta_d}$ is chosen not too large. A too large value would imply that $\hat{r}_{\Delta,k}$ is only determined by the value of $\hat{r}_{\Delta_d}$. The algorithm would in this case loose the adaptivity and the gain would decrease as $\frac{1}{k+1}$.

V. Experiments

A. The process

The algorithm presented in the previous sections will now be applied to a real industrial system. The system, an ABB IRB1400 industrial robot, is depicted in Fig. 2. For a more thorough description of the technical part of the experimental setup see [4]. The IRB1400 is a standard industrial robot having gear boxes with gear ratio of 118:1 for the main axes. Previous experimental studies on ILC applied to industrial robots can be found in, e.g., [20], [5], [21], and [22].

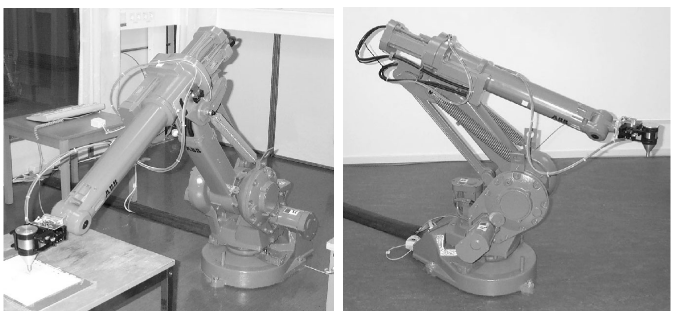


Fig. 2. The ABB IRB1400 manipulator.

In this example ILC is applied to three joints. The robot has a total of 6-DOF but for the three wrist joints ILC is not applied. Each of the joints are modeled as a transfer operator description from the ILC control input to the measured motor position of the robot, i.e., $\boldsymbol{G}^0$ in (1). It should be stressed that this $\boldsymbol{G}^0$ is in fact a closed loop system. The feedback controller, implemented by ABB, is working in parallel with the ILC and since the controller is doing a very good job, the closed loop from reference angular position to measured angular position can be described using a low order linear discrete time model. The models are calculated using *System Identification Toolbox* [23] and are given by,

$$\hat{G}_1(q) = \hat{G}_2(q) = \frac{0.1q^{-1}}{1-0.9q^{-1}}, \quad \hat{G}_3(q) = \frac{0.13q^{-1}}{1-0.87q^{-1}} \qquad (19)$$

The accuracy in repeating the same task for the IRB 1400 is very high and therefore the initial error at each iteration can be assumed to be the same in every iteration.

B. Description of the experiment

The experiment with the adaptive ILC method presented in Algorithm 1 is performed on the ABB IRB1400 shown in Fig. 2. First we note that the problem we get when controlling the robot is a classical ILC tracking problem. In Fig. 3 the configuration of the system considered in the experiments is shown. Clearly this is different from the structure of the standard system description in the disturbance rejection approach in Fig. 1. The reference signal is one of the inputs to the system and the

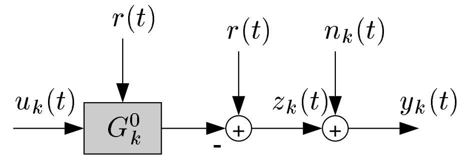


Fig. 3. The system configuration used in the experiments.

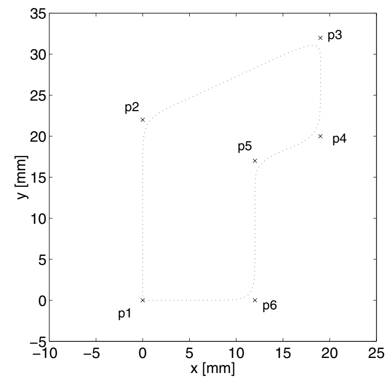


Fig. 4. The trajectory used in the experiments, shown on the arm-side and translated such that p1 is in the origin. Programmed speed is 100 mm/s in the first experiment and 250 mm/s in the second experiment.

goal of the control now becomes to track the reference trajectory $r(t)$. By using the reference signal $r(t)$ as the disturbance we get the control variable $z_k(t)$ as the control error that we want to minimize. It is now straightforward to apply the algorithm presented in Section III to the problem. Note that in the application the parameter $\hat{r}_{\Delta_d}$ does not have a direct interpretation as a physical parameter. The parameter $\hat{r}_n$ however, still have a physical meaning and can be chosen accordingly. From Algorithm 1 we also know that $\hat{r}_{\Delta_d}$ can be chosen such that p_∞ and κ_∞ get the desired values and this is the approach taken here.

In Fig. 4 the desired trajectory on the arm-side of the robot is shown. The actual position of `p1` in the base coordinate system is x = 1300 mm, y = 100 mm, and z = 660 mm for the first and x = 600 mm, y = 250 mm, and z = 800 mm for the second experiment. The actual configurations of the robot in the two experiments are also shown in Fig. 2 (experiment 1 left and 2 right). The programmed velocities in the two experiments are 100 mm/s and 250 mm/s, respectively.

To make it possible to evaluate the adaptive ILC algorithm, two different algorithms have been chosen for comparison. The first is a traditional ILC algorithm with the updating scheme given by (14). The second algorithm is the same as the adaptive ILC algorithm, except that the Kalman gain κ_k is fixed to a value slightly less than one. The second algorithm is to show the advantage of having an adaptive gain in the updating formula.

C. Design

From the design procedure presented in Section II it is obvious that it is necessary to have a model of the system in order to find the ILC scheme. In the description of the process in Section V-A it is shown that there exist models for each of the three joints of the robot and that these models are represented by linear discrete time transfer functions. The design that will be used here is based on the ideas presented in Section IV.

The matrix $\boldsymbol{H}$ is simply chosen as a realization of a second order Butterworth filter with cut-off frequency 0.2 of the Nyquist frequency. $\boldsymbol{W}_u^{-1}$ is found as $\boldsymbol{W}_u^{-1} = \boldsymbol{H}\boldsymbol{H}^T$. It is of course necessary to decide values for the other design variables. The following values were used in the experiment,

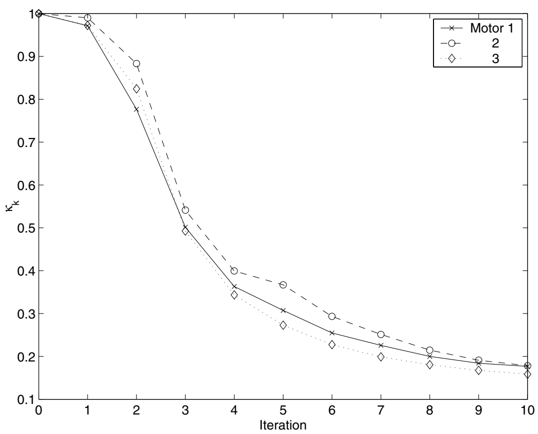
$$p_0 = 10^4, \quad \zeta = 10^3, \quad \hat{r}_{\Delta_d} = 10^{-6}$$
$$\hat{r}_n = 5 \cdot 10^{-5}, \quad \boldsymbol{\Delta}_G = 0.5 \cdot I$$

This means that p_∞ defined according to (17) becomes equal to $5.5 \cdot 10^{-6}$ and the corresponding κ_∞ becomes $\kappa_\infty = 0.10$ which is a reasonable lower limit for the gain κ_k.

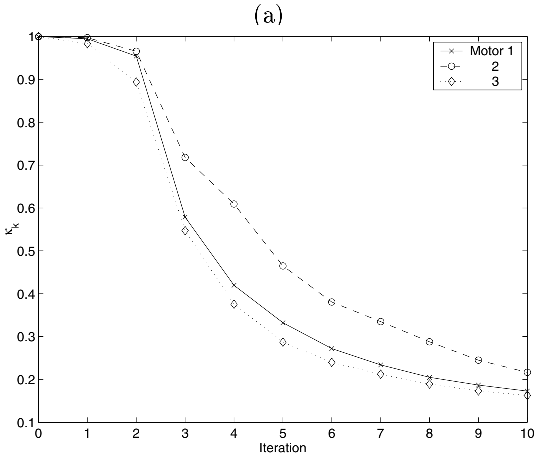
The filters in the traditional ILC algorithm, given by (14), are chosen such that $Q(q)$ is a second order Butterworth filter with cut-off frequency 0.2 of the Nyquist frequency and $L(q) = 0.9q^4$. This choice of L-filter is based on the model that we used for the design of the adaptive ILC algorithm and it gives good robustness properties [4].

D. Results

The experiments described in Section V-B are run three times. Once using the proposed adaptive ILC scheme, once with the design according to the proposed adaptive ILC design but with fixed gain ($\kappa_k = 0.99$), and finally using a "traditional" ILC updating scheme. The result from the experiments are evaluated on the motor side of the robot. This is also where the measurements and the control are performed.



(a)



(b)

Fig. 5. The value of κ_k for the ILC associated with the three different motors. (a) is from experiment 1 and (b) from experiment 2.

The results on the motor-side from the two experiments with the three ILC algorithms are shown in Fig. 6. Obviously, the transient response of the learning is best with the adaptive ILC scheme. Notice that the ILC algorithm designed according to the adaptive ILC scheme but with the Kalman gain kept constant is not so robust. This can be seen from the fact that $\|y_k\|$ for motor 1 in experiment 1 actually starts growing after 6-7 iterations. In Fig. 5 the values of the gains, κ_k, in the adaptive ILC algorithms are shown as a function of iteration. Obviously they are large in the first iterations where $d_k(t)$ has not been compensated for completely. When the errors decrease the gains also decrease. For experiment 1 and motor 1 (see Fig. 6) the error does not decrease as fast as for the other motors and this is also reflected in the gain, which keeps a higher value than for the other motors in Fig. 5.

It is important to choose the correct size of $\hat{r}_{\Delta_d}$ in order to get this effect, cf. Algorithm 1. If $\hat{r}_{\Delta_d}$ is chosen too large this value will dominate $\hat{r}_{\Delta,k}$ and the κ_k will not be like in Fig. 5, instead the value of κ_k will decrease like $\frac{1}{k+1}$.

It is also important to evaluate the result on the arm-side of the robot. In the experiments described here it is not possible to show any improvement on the arm-side (for more details see [4]). One important reason is that there are no measurements from the arm-side included in the ILC algorithm. This result indicates that it is necessary in this application to include more sensors in order to minimize the true path error on the arm-side.

VI. Conclusions

When taking the measurement disturbance into account it becomes clear that it is possible to get a better result by introducing an iteration varying gain in the ILC algorithm. Results from state space modeling and design are used to create ILC method. The resulting ILC algorithm works also when the system is not perfectly known, i.e., it is robust. The algorithm is based on an LQ-solution and a time variable Kalman filter where one of the design variables in the Kalman filter is calculated from data. The algorithm is therefore, in fact, adaptive.

The proposed adaptive algorithm is also applied to an industrial process, an ABB IRB 1400 industrial robot. The results show an improvement in the path following on the motor-side of the robot and the proposed adaptive and model based ILC algorithm is shown to give better result than a traditional ILC algorithm with constant gain.

Acknowledgments

The author would like to thank VINNOVA's Center of Excellence ISIS at Linköpings universitet, Linköping, Sweden, for the financial support.

References

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

[2] K. L. Moore, "Iterative learning control - an expository overview," *Applied and Computational Controls, Signal Processing and Circuits*, vol. 1, 1998.

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

[4] M. Norrlöf, *Iterative Learning Control: Analysis, Design, and Experiments*, Ph.D. thesis, Linköping University, Linköping, Sweden, 2000, Linköping Studies in Science and Technology. Dissertations No. 653. Download from http://www.control.isy.liu.se/publications/.

[5] S. Panzieri and G. Ulivi, "Disturbance rejection of iterative learning control applied to trajectory tracking for a flexible manipulator," in *Proceedings of the 3rd European Control Conference*, Sep 1995, pp. 2374–2379.

[6] C.-J. Chien, "A discrete iterative learning control of nonlinear time-varying systems," in *Proc. of the 35th IEEE Conf. on Decision and Control*, Kobe, Japan, Dec 1996, pp. 3056–3061.

[7] Y. Chen, C. Wen, J.-X. Xu, and M. Sun, "An initial state learning method for iterative learning control of uncertain time-varying systems," in *Proc. of the 35th Conf. on Decision and Control*, Kobe, Japan, Dec 1996, pp. 3996–4001.

[8] Kevin L. Moore, "Multi-loop control approach to designing iterative learning controllers," in *Proc. of the 37th IEEE Conference on Decision and Control*, Tampa, Florida, USA, Dec 1998.

[9] M. Phan and J.A Frueh, "Learning control for trajectory tracking using basis functions," in *Proc. of the 35th IEEE Conf. on Decision and Control*, Kobe, Japan, Dec 1996, pp. 2490–2492.

[10] Jay. H. Lee, Kwang S. Lee, and Won C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, May 2000.

[11] Brian D.O. Anderson and John B. Moore, *Optimal Filtering*, Prentice-Hall, 1979.

[12] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control using optimal feedback and feedforward actions," Tech. Rep. 95/13, Centre for Systems and Control Engineering, University of Exeter, 1995.

[13] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete time systems with exponential rate of convergence," Tech. Rep. 95/14, Centre for Systems and Control Engineering, University of Exeter, 1995.

[14] J.A Frueh M.Q. Phan, "Linear quadratic optimal learning control (LQL)," in *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, Florida, USA, 1998, pp. 678–683.

[15] S. Gunnarsson and M. Norrlöf, "On the design of ILC algorithms using optimization," *Automatica*, vol. 37, pp. 2011–2016, 2001.

[16] T. Kuc and J. S. Lee, "An adaptive learning control of uncertain robotic systems," in *Proc. of the 30th Conf. on Decision and Control*, Brighton, England, Dec 1991, pp. 1206–1211.

[17] J-X. Xu and B. Viswanathan, "Adaptive robust iterative learning control with dead zone scheme," *Automatica*, vol. 36, no. 1, pp. 91–99, 2000.

[18] D.H. Owens and G. Munde, "Error convergence in an adaptive iterative learning controller," *International Journal of Control*, vol. 73, no. 10, pp. 851–857, 2000.

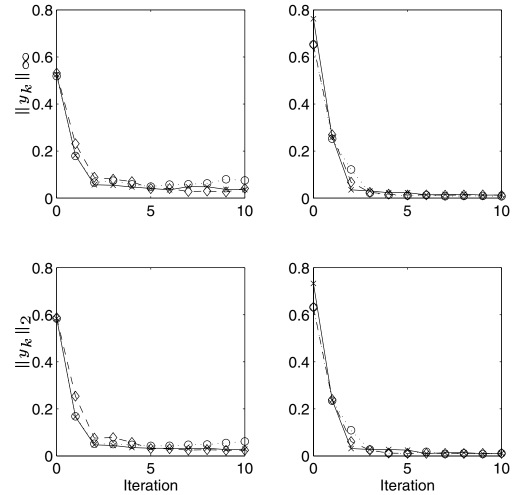
[19] L. Ljung, "Model error modeling and control design," in *The IFAC Symposium on System Identification, SYSID2000*, 2000.

[20] A. D. Luca, G. Paesano, and G. Ulivi, "A frequency-domain approach to learning control: Implementation for a robot manipulator," *IEEE Transactions on industrial electronics*, vol. 39, no. 1, Feb 1992.

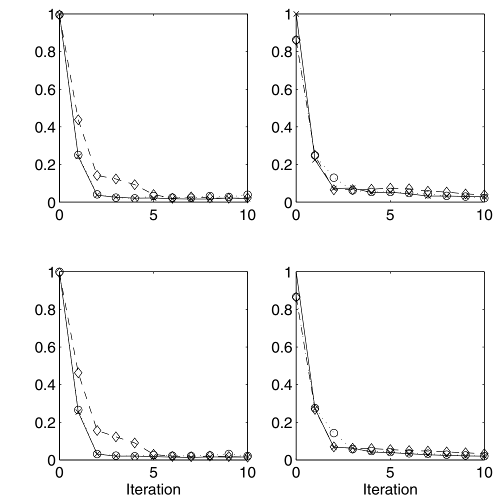
[21] K. Guglielmo and N. Sadegh, "Theory and implementation of a repetetive robot controller with cartesian trajectory description," *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, pp. 15–21, March 1996.

[22] S. Kawamura, F. Miyazaki, and S. Arimoto, "Realization of robot motion based on a learning method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 126–134, Jan/Feb 1988.

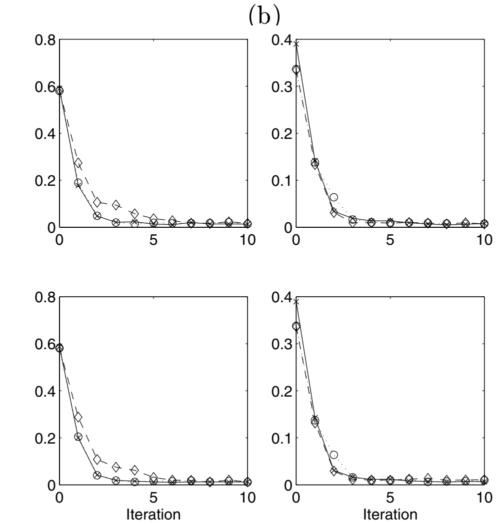
[23] L. Ljung, *System Identification Toolbox - For Use with Matlab*, The MathWorks Inc., 1995.



(a)



(b)



(c)

Fig. 6. The error in ∞-norm and 2-norm for the different ILC algorithms in the two experiments. The adaptive ILC scheme ($\times$), the adaptive scheme with κ_k constant ($\circ$), and the traditional ILC scheme given by (14) ($\diamond$). Experiment 1 is shown in the left diagrams while results from experiment 2 are shown in the right diagrams. (a) represent the results from motor 1, (b) from motor 2, and (c) from motor 3.