Using iterative learning control to get better performance of robot control systems

Mikael Norrlöf and Svante Gunnarsson
Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden
www: http://www.control.isy.liu.se
email: mino@isy.liu.se, svante@isy.liu.se

1997-06-04

Technical reports from the Automatic Control group in Linköping are available by anonymous ftp at the address 130.236.20.24 (ftp.control.isy.liu.se/pub/Reports/). This report is contained in the compressed postscript file 1955.ps.Z.
Using iterative learning control to get better performance of robot control systems

M. Norrlöf and S. Gunnarsson
Department of Electrical Engineering, Linköping University,
Linköping, Sweden
Email: mino@isy.liu.se, svante@isy.liu.se
May, 1997

Abstract

Many manipulators at work in factories today repeat their motions over and over in cycles and if there are errors in following the trajectory these errors will also be repeated cycle after cycle. The basic idea behind iterative learning control (ILC) is that the controller should learn from previous cycles and perform better every cycle. Iterative learning control is used in combination with conventional feed-back and feed-forward control, and it is shown that learning control signal can handle the effects of unmodeled dynamics and friction. Convergence and disturbance effects as well as the choice of filters in the updating scheme are also addressed.

1 Introduction

In factories many manipulators repeat their motions over and over in cycles, e.g. in laser cutting applications. However, the problem today is that if there are errors in following the trajectory these errors will also be repeated cycle after cycle and evidently a system that learns would be convenient. The basic idea behind iterative learning control (ILC) is that the controller should learn from previous cycles and perform better every cycle. The first article presented in this topic was written by Arimoto et al [1] in 1984 and since than many papers has been addressing robot control in combination with iterative learning control, e.g. [2], [5], and [10]. The convergence properties when using iterative learning control is another very important aspect, addressed already in [1], and further covered in many articles e.g. [6], [7], [9].

In this paper iterative learning control is studied as a complement to conventional feed-forward and feed-back control. We will mainly consider linear systems, but also study the effects of non-linear friction. The aim is to illustrate the fundamental properties of the ILC algorithm applied in this framework, with focus on convergence, robustness and disturbance effects.
2 Problem Statement

If we introduce a load disturbance in figure 1 we get

\[ Y = G(U + D) \]  \hspace{1cm} (1)

where \( U, Y, \) and \( D \) represents input, output, and load disturbance respectively. \( G \) is the transfer function of the system, in this case the robot arm. When capital letters are used in the following it indicates that the signals are transformed, the discussion covers both continuous and discrete time signals unless otherwise stated. The system is controlled using a combination of feed-forward and feed-back using

\[ U = F_fY_D + F(Y_D - (Y + N)) \]  \hspace{1cm} (2)

where \( Y_D \) and \( N \) denotes the reference signal and measurement disturbance respectively. \( F_f \) and \( F \) denote the transfer function of the feed-forward and feed-back regulators.

![Figure 1: A system controlled with feed-forward and feed-back regulators](image)

The control signal generated by the feed-back regulator will be considered as error signal,

\[ E = F(Y_D - Y) \]  \hspace{1cm} (3)

If the feed-back controller is of PD type the error will be a linear combination of the position error and the velocity error which is reasonable, because the control objective is to minimize the position error and the velocity error. Using equations (1), (2), and (3), the error can be formulated like

\[ E = G_C((G^{-1} - F_f)Y_D - D + FN) \]  \hspace{1cm} (4)

where \( G_C \) is the transfer function of the closed loop system given by

\[ G_C = \frac{FG}{1+FG} \]  \hspace{1cm} (5)

3 Outline of the Method

As stated in the introduction the input signal, \( y_D \), in many applications is repetitive. This means that if there exists an error in the following of the trajectory in the first iteration this error will be repeated cycle after cycle. If the dynamics of the system is largely repeatable a control algorithm that improves performance from trial to trial can be constructed. A
new control signal $\Delta U_k$ is added to the control signal $U$ in figure 1 and the input signal to the system will thus be given by

$$U_k = F_f Y_D + F(Y_D - (Y_k + N_k)) + \Delta U_k \quad \text{(6)}$$

The index $k$ indicates the iteration number. Considering only linear operations the updating of the correction signal can, in the frequency domain, be expressed as

$$\Delta U_{k+1} = \sum_{j=0}^{k} \tilde{H}_j E_j \quad \text{(7)}$$

where $\tilde{H}_j, \quad j = 0, \ldots, k$ are linear filters. For convenience we shall here however consider recursive update equations on the form

$$\Delta U_{k+1} = H_1 \Delta U_k + H_2 E_k \quad \text{(8)}$$

where $H_1$ and $H_2$ are linear filters. The choice of the filters $H_1$ and $H_2$ is a main task when designing a learning control algorithm, since the filters determine the convergence and robustness properties. One method for choosing appropriate filters in the update equation is presented in [3] where methods from design of robust controllers are applied. The filters are designed to give a convergent ILC algorithm despite uncertainties in the process model. In [4] the problem is considered from a different viewpoint and the choice of the ILC input signal is formulated as an optimization problem, resulting in a time domain updating equation for the input signal.

### 4 Convergence Properties

The convergence properties of the ILC algorithm is very important and we will now investigate how the error signal behaves when the update equation (8) is used. If we define $E_0$ as the disturbance free error signal obtained in the first iteration when $\Delta U_0 \equiv 0$ we get

$$E_0 = G_C(G^{-1} - F_f)Y_D \quad \text{(9)}$$

From equations (4), (6), and (8) the following can be derived

$$E_{k+1} = E_0 - G_C H_1 \Delta U_k - G_C H_2 E_k - G_C D_{k+1} + F G_C N_{k+1} \quad \text{(10)}$$

By adding and subtracting relevant terms on the right hand side we arrive at

$$E_{k+1} = (1 - H_1)E_0 + H_1(E_0 - G_C \Delta U_k - G_C D_k + F G_C N_k) - G_C H_2 E_k + H_1 G_C D_k - G_C D_{k+1} + H_1 F G_C N_k + F G_C N_{k+1} \quad \text{(11)}$$

which implies the following error update equation

$$E_{k+1} = (1 - H_1)E_0 + (H_1 - G_C H_2)E_k + G_C (H_1 D_k - D_{k+1}) + F G_C (N_{k+1} - H_1 N_k) \quad \text{(12)}$$
A corresponding equation is presented in [10] for the open loop case and for load disturbances only.

The convergence properties are determined by the homogeneous part of the difference equation (13) and referring to [2] the convergence condition, in the continuous-time case, is that

\[ |H_1(i\omega) - H_2(i\omega)G_C(i\omega)| < 1 \quad \forall \ \omega \]  \quad (13)

Provided that the learning procedure converges the error signal becomes

\[ \bar{E} = \frac{1 - H_1}{1 - H_1 + G_CH_2}E_0 \]  \quad (14)

We see that by using $H_1 \neq 1$ we are not able to eliminate the error completely, but as will be seen later other advantages are obtained by this choice. In e.g. [8] the case $H_1 = \mu$, where $\mu < 1$ is a scalar, is studied. An alternative parameterization of the filters in the learning law was presented in [3], where

\[ H_1 = Q, \quad H_2 = QL \]  \quad (15)

$Q$ and $L$ are filters. In [11] this formulation is used with $Q$ defined as

\[ Q = \frac{1}{1 + V} \]  \quad (16)

and $H$ scalar. The condition for convergence, based on [3], becomes

\[ \|1 - LG_C\|_\infty < \|Q^{-1}\|_\infty = \|1 + V\|_\infty \]  \quad (17)

and it is obvious that the stability region can be extended by a suitable choice of the filter $Q$, resulting in a so called stabilizing circle (see figure 2). By letting $Q$ be frequency dependant the stability region can be extended in a frequency dependant way and equation (17) shows that $Q$ should be a low-pass filter to make $Q^{-1}$ extend the stability region for high frequencies. In [3] it is also shown that the filter $L$ can be found through a 'model matching problem' where $L$ is found by solving the $H_\infty$ problem

\[ L = \arg \min_{L \in H_\infty} \|Q(1 - LG_C)\|_\infty \]  \quad (18)

This minimization will result in

\[ \|Q(1 - LsG_C)\|_\infty = \gamma_s < 1 \]  \quad (19)

It should be noted that the smaller $\gamma_s$ is the faster the convergence of $\Delta u$ and $e$. 


5 Disturbance Effects

A number of observations can be made using equation (13). Let us first consider the case 

\[ H_1 = 1, \]

which implies the update equation

\[ E_{k+1} = (1 - G_C H_2)E_k + G_C (D_k - D_{k+1}) + FG_C (N_{k+1} - N_k) \]  

(20)

The disturbances contribute to the error equation by their differences between the iterations. If a disturbance is of repetitive nature in the sense that the disturbance signals \( d_k(t) = d_{k+1}(t) \) and \( n_k(t) = n_{k+1}(t) \) for all \( k \) the contribution to the error difference equation is zero. This assumption is more likely for the load disturbance where for example load disturbances due to gravitational forces can be expected to be rather similar during different iterations. Measurement disturbances, on the other hand, are more likely to be of random character which means that \( n_{k+1}(t) \neq n_k(t) \) in general, and there will hence always be a driving term on the right hand side of equation (20) that prevents \( E_k(s) \) from tending to zero.

Let us also consider the situation with \( H_1 \neq 1 \), neglect measurement disturbances and assume that \( d_k(t) = d(t) \forall k \). This corresponds to the error difference equation

\[ E_{k+1} = (1 - H_1)E_0 + (H_1 - G_C H_2)E_k - G_C D(1 - H_1) \]  

(21)

The load disturbance will act as a driving term similar to the initial error \( E_0 \).

6 A Simulation Example

We shall consider a simplified description of a single robot joint modeled as a double integrator, i.e.

\[ G(s) = \frac{1}{J s^2} \]  

(22)

Since the system is computer controlled we shall use the discrete time representation given by the transfer function

\[ G(z) = \frac{T^2(z + 1)}{2 J (z - 1)^2} \]  

(23)

where \( J = 0.0094 \) is the moment of inertia. The system is controlled by a discrete time PD-regulator given by

\[ F(z) = K_P + K_D \frac{(z - 1)}{T z} \]  

(24)

where \( K_P = 12.7 \) and \( K_D = 0.4 \). The feed-forward filter is chosen as a double differentiation represented by

\[ F_f = \frac{J^*(z - 1)^2}{T^2 z^2} \]  

(25)

where \( J^* \) is the estimated moment of inertia. The correction signal will be updated according to equation (8) where \( H_1(z) \) and \( H_2(z) \) are filters that both may be non-causal. The model is simulated using 1 kHz sampling frequency. For evaluation of the algorithm we shall apply the reference trajectory shown in Figure 2.
Figure 2: The reference signal (left). Nyquist curves for $G_C H_2$ for the choices $H_2 = 1$ and $H_2 = G_C^{-1}(1 - H_B)$, Learning circle and stabilizing circle (right).

### 6.1 Unmodeled dynamics

The first goal is to investigate how the learning control approach can deal with unmodeled dynamics. We shall consider the case when there is a 30% error in $J^*$, i.e., the control system is based on an incorrect value of the moment of inertia. For $H_1(z) = 1$ the ideal choice of $H_2$ would be to choose it as the inverse of $G_C(z)$, which, theoretically, would result in convergence to zero in one step. This is however an unrealistic choice since it requires exact knowledge of the system and results in a filter with very high gain for high frequencies. Instead we consider

$$H_2(z) = G_C^{-1}(z)(1 - H_B(z)) \tag{26}$$

where $G_C(z)$ denotes the closed loop transfer function we obtain by using the model of the open loop system and $H_B(z)$ is a Butterworth high pass filter (here of second order) for which the gain tends to one for high frequencies. Choosing $H_2(z)$ according to this design rule, with cut-off frequency of the high pass filter equal to 0.4 times the Nyquist frequency, gives the Nyquist curve depicted in Figure 2. Figure 2 also shows $G_C(z)$ for comparison. The whole Nyquist curve is now inside the learning circle while it for large frequencies tends to the origin. The learning control algorithm is then tested in simulations. Figure 3 (upper left) shows the FFT of the error signal $e_k(t)$ for different iterations.

### 6.2 Friction

Since all robots contain some amount of friction it is of interest to evaluate the performance of the learning control algorithm under such conditions. The dynamics of the robot is then described by

$$J \ddot{y}(t) = u(t) - f_c \text{sign}(\dot{y}(t)) - f_v \dot{y}(t), \quad \dot{y}(t) \neq 0 \tag{27}$$

and

$$J \ddot{y}(t) = 0 \quad |u(t)| \leq f_c, \quad \dot{y}(t) = 0 \tag{28}$$
where the coefficient $f_c$ is chosen such that the Coulomb friction force corresponds to 30% of the maximum torque, in this case 0.12 and the viscous friction coefficient $f_v$ is set to $10^{-2}$. The linear analysis carried out above is not applicable when we have introduced nonlinear elements into the problem but we can still evaluate the learning control algorithm using simulations. If we carry out the same simulations as in the previous case we get the result shown in Figure 3 (upper right).

Even though the filter $H_2(z)$ designed above was robust enough to handle that it was designed based on an incorrect value of the moment of inertia it is of interest to further improve the stability margins of the learning control algorithm. This can be done by using the filter $V(z)$ discussed above. In the simulations we have chosen $V(e^{j\omega})$ as a first order high pass filter with cut-off frequency 0.7 times the Nyquist frequency. The high frequency gain of the filter is 0.1, which means that the stability region is extended in the high frequency regions. The result of this choice is shown in Figure 2, where the obtained stabilizing circle is shown. In Figure 3 (lower left) the simulations results are shown. The convergence properties are comparable with what was obtained without the use of $V(e^{j\omega})$ but we see that there is a convergence to a little higher energy level than without the $V$ filter. In figure 3 (lower right) the signal energy is shown as a function of iteration.

Figure 3: Error signal spectrum for Linear system (upper left), system with friction (upper right), system with friction and with V filter (lower left), and error signal energy (lower right)
7 Conclusions

A iterative learning control strategy for robot control systems has been studied. Aspects such as convergence, robustness, and choice of filters in the ILC algorithm have been discussed. It is shown that the proposed control law works well also in the presence of nonlinear friction.

8 Acknowledgments

This work was supported by CENIIT at Linköping University and ABB Robotics within ISIS at Linköping University.

References


