Proceedings of the 17th Canadian Congress of Applied Mechanics - CANCAM 99
Hamilton, Ontario, Canada May 30-June 3, 1999
pp. 341-342.

An Iterative Learning Control Algorithm for Improving Robot and Machine Tool Path Tracking

Gary M. Bone* and Paul Clayton
Department of Mechanical Engineering
McMaster University
Hamilton, Ontario, Canada L8S 4L7
*Email: gary@immrc.mcmaster.ca

Abstract

In this paper a novel iterative learning control algorithm is developed, and applied to the path tracking problem. The conditions for convergence are analysed, including the effect of modelling errors. In experiments performed using an unmodified industrial robot the path tracking errors were reduced by 70% after two learning passes, and by 95% after twenty learning passes.

1. Introduction

Path tracking errors are one of the current limitations of robots (and to a lesser extent machine tools) when performing precision manufacturing tasks. These errors consist of a repeatable component and a random component. In this paper a simple new form of Iterative Learning Control (ILC) will be investigated as a method for improving path tracking precision.

ILC is suitable for any operation which is performed repeatedly and which is subject to repeatable errors. It uses the information from the previous repetitions (or "trials") to modify the process input and drive the process close to the desired trajectory. The substantial memory requirements of ILCs, which made their full scale implementation difficult in the past, can easily be handled by today's high speed, high capacity, storage devices.

In 1995, Bone [1] developed a formulation of the popular long-range predictive control method generalized predictive (GPC) which incorporated ILC into GPC. The new algorithm demonstrated improved regulation performance over standard GPC, without affecting its stability properties.

The reader is referred to the surveys by Moore *et al.* [2] and Horowitz [3] for more extensive reviews of ILC research.

In this paper a new ILC algorithm is described. The algorithm is then implemented on an industrial robot to improve cartesian path control.

2. Iterative Learning Control Algorithm

It is assumed that the plant may be reasonably modelled by a linear discrete time model of the form:

$$\frac{y_k(t)}{u_k(t)} = \frac{B(q^{-1})}{A(q^{-1})} \tag{2}$$

where y is the plant output, u is the plant input, $B(q^{-1}) = b_0 q^{-d} + b_1 q^{-d-1} + \ldots + b_m q^{-d-m}$, $A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$, d is the dead time, and q^{-1} is the backwards shift operator.

The ILC algorithm is then:

$$u_{k+1}(t) = L u_k(t) + M_k \frac{A(q^{-1})}{z^{-d} B^{+}(q^{-1})} e_k^{*}(t) \tag{3}$$

Where k is the trial number, L and M are gains, $B^{+}(q^{-1})$ is the B polynomial with all factors outside of the unit circle replaced by their steady state gains, and $e*$ is the learnt error. Exponential forgetting is used to filter out the random portion of the error as follows:

$$e_k^{*}(t) = \begin{cases} e_k(t) & for\ k = 0 \\ (1-\lambda) e_k(t) + \lambda e_{k-1}^{*}(t) & for\ k > 0 \end{cases} \tag{4}$$

$$e_k(t) = r(t) - y_k(t) \tag{5}$$

Where r is the reference trajectory.

Although a fixed value of gain M_k may be used, this can lead to oscillatory behaviour as discussed further in section 4.2. Improved performance may be obtained by starting with a large gain, C, and then reducing it, as follows:

$$M_k = \frac{C}{(k+1)} \tag{6}$$

3. Convergence Analysis

Over a single trial the ILC acts as a finite gain feedforward controller so its closed-loop stability in the classical sense is not an issue. However, its convergence to the desired trajectory over the series of trials is not guaranteed, but a function of the gains, L, M and λ, and of the accuracy of the plant model. The ILC's convergence will be analysed in this section.

If we assume the true plant may be represented by eq. (2) with the perfect model given by A=A*, and B=B*, then eq. (3) may be written as:

$$\frac{A*}{B*} y_{k+1} = L \frac{A*}{B*} y_k + M \frac{A}{B^{+}} e_k^{*} \tag{7}$$

The sampling interval for eqs. (2) and (3) is defined as T. If each trial is N samples long, and we redefine the sampling interval here to equal the length of a single trial (i.e. $N{\cdot}T$), then we can rewrite eq. (4) as:

$$e^{*}(t) = \frac{(1-\lambda)(r(t) - y(t))}{1 - \lambda q^{-1}} \tag{8}$$

and eq. (7) as:

$$\frac{A*}{B*} y(t) = L \frac{A*}{B*} y(t) q^{-1} + M \frac{A}{B^{+}} e_k^{*} q^{-1} \tag{9}$$