Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 46 (2016) 391 – 395

www.elsevier.com/locate/procedia

7th HPC 2016 – CIRP Conference on High Performance Cutting

# Iterative learning for machine tools using a convex optimisation approach

Titus Haas[a,]*, Natanael Lanz[a], Roman Keller[a], Sascha Weikert[b] , Konrad Wegener[a]

[a]*Institute of Machine Tools and Manufacturing, ETH Zurich, Leonhardstrasse 21, 8092 Zurich, Switzerland*
[b]*Inspire AG, Technoparkstrasse 1, 8005 Zurich, Switzerland*

* Corresponding author. Tel.: +41-44-632-4845 ; fax: +41-44-632-1159. *E-mail address:* haas@iwf.mavt.ethz.ch

**Abstract**

Dynamic, quasi-static and motion control deviations lead to nonlinear but systematic tracking errors. It is shown that these errors can be reduced significantly by adjusting the set points using an optimization based iterative learning approach. This method uses either values obtained from internal encoders or alternatively tool center point measurements. The approach is presented, discussed and validated using simulation and measurement results.



Peer-review under responsibility of the International Scientific Committee of 7th HPC 2016 in the person of the Conference Chair Prof. Matthias Putz

*Keywords:* Control; Machine tool; Precision

## 1. Introduction

Dynamic, quasi-static and motion control deviations lead to nonlinear but systematic tracking errors of machine tools. The goal of iterative learning is to learn from systematic and repeatable errors of previous trials and to improve the following trial. Machine tools typically do not learn from previous experience. Especially in high-volume production, where a given part is produced multiple times, it is desirable to reduce tracking errors. A reduction of tracking errors would either allow higher feed rates, leading to the same dynamic deviations, or to reduce contour errors, using the same feed rate. According to Bristow et al. [1], iterative learning control (ILC) modifies the input of the controller of a system and not the controller itself, as for example adaptive control and neural network strategies do. Repetitive control is similar to ILC but for continuous operations, i.e. the next iteration follows immediately and the initial conditions are given by the final conditions of the previous trial.

A lot of work has been done in the field of ILC; a good survey is given by Bristow et al. [1] and Ahn et al. [2].

The basic idea of ILC was published by Garden [3] in 1971. The algorithm was presented for the first time in English by Arimoto et al. [4] in 1984, where an iterative learning scheme for a robot manipulator was proposed.

Feedforward control can compensate tracking errors caused by lag and has good performance if the system is known accurately. Stiction, not modelled nonlinear behavior and disturbances can limit the performance of feedforward control [1]. ILC can compensate any nonlinear, but repeatable disturbance. The performance of ILC is limited by unrepeatable disturbances and noise. The influence of the latter can be reduced by zero-phase filtering, e.g. Butterworth, which is possible without lag. A combination of feedback control and ILC is recommended by [1].

Togai and Yamano [5], e.g., formulated the iterative learning control problem as a quadratic optimization criterion and therefore, reduced the error and additionally the input. Penalizing also the input, the error cannot be reduced to the minimal achievable error. Amann et al. [6], Lee et al. [7] and Barton et al. [8] extended the cost function by the input change instead of the input. Therefore, the ILC algorithm has an integral action in the iteration domain and the minimal achievable error is attainable. It is possible to consider constraints, disturbances and model errors for example.

Kim and Kim [9] presented the proportional, integral and derivative (PID) type ILC algorithm of [4] for a machine tool,


Peer-review under responsibility of the International Scientific Committee of 7th HPC 2016 in the person of the Conference Chair Prof. Matthias Putz
doi:10.1016/j.procir.2016.04.033

where the actual machined path was measured using a roundness measuring instrument. A decrease of 58% of the error between the measued and the desired path is shown for cutting circles with radius of 29.7 mm and a feed rate of 200 mm/min. Tsai and Chen [10] applied the PID type ILC algorithm of [4] to reduce the deviation between the desired cutting and actual fracture trajectory for a $CO_2$ laser machine tool. Tsai et al. [11] proposed a P-type ILC algorithm with predicted tracking and contour error and compared the performance of the error reduction for different weightings of those errors. An application for improving contour error tracking in precision motion control was presented by Altin and Barton [12]. The norm optimal framework was used to minimize the tracking error, contour error and input change. Using a model of the contour error, Wu et al. [13] proposed an A-type iterative learning cross-coupled control that was based on a contour error model and showed the convergence of it. Khong et al. [14] proposed an extremum seeking approach to iterative learning for nonlinear time-varying systems.

In this paper, the optimization based ILC approach, proposed e.g. by Barton et al. [8], is used and compared to the commonly used PD-ILC algorithm presented by [1]. The quadratic optimization formulation is called convex optimization (CO) ILC in the remainder of this paper.

## 2. Comparison of ILC algorithms

### 2.1. Overview of ILC methods

The general application scheme of the ILC algorithm is shown in Fig. 1. The plant **P** represents any dynamic system, e.g. a machine tool servo axis and **C** represents the controller. The tracking error of the iteration j, $\mathbf{e}_j$, is given by the deviation between the desired trajectory, $\mathbf{y}_d$, and the actual measurement, $\mathbf{y}_j$. The input of the iteration j is $\mathbf{u}_j$. Only asymptotically stable closed-loop systems are considered in this paper. Note that $\mathbf{e}_j$, $\mathbf{y}_d$, $\mathbf{y}_j$ and $\mathbf{u}_j$ are vectors of length *N*, where *N* is the number of time discrete trajectory samples. The computation of $\mathbf{u}_j$ is repeated for each iteration. Therefore, ILC can be used online or offline.

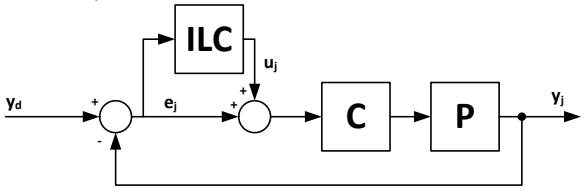


Fig. 1. General application scheme of the ILC algorithm.

In this paper, two types of ILC algorithms have been implemented and compared:

- PD-ILC with a Butterworth low pass filter as presented in [1]
- CO-ILC with a second order model representing the closed loop servo axis behavior of each axis

No model is needed for PD-ILC, whereas for CO-ILC a linear time invariant model of the closed loop system dynamics is required (1). For an initial state $\mathbf{x}_j[0]=0$, the servo loop dynamic matrix $\mathbf{P}_{cl}$, using the state space closed loop dynamics in (2), can be derived as shown in (3). The indices j and n denote the iteration and time sample, respectively.

$$\mathbf{y}_j = \mathbf{P}_{cl}(\mathbf{y}_d + \mathbf{u}_j) \tag{1}$$

$$\begin{cases} \boldsymbol{x}_j[n+1] = \boldsymbol{A}_{cl}\,\boldsymbol{x}_j[n] + \boldsymbol{B}_{cl}\left(\boldsymbol{u}_j[n] + \boldsymbol{y}_d[n]\right) \\ \boldsymbol{y}_j[n] = \boldsymbol{C}_{cl}\,\boldsymbol{x}_j[n] \end{cases} \tag{2}$$

$$\boldsymbol{P}_{cl} = \begin{pmatrix} \boldsymbol{C}_{cl}\boldsymbol{B}_{cl} & 0 & \cdots & 0 \\ \boldsymbol{C}_{cl}\boldsymbol{A}_{cl}\boldsymbol{B}_{cl} & \boldsymbol{C}_{cl}\boldsymbol{B}_{cl} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{cl}\boldsymbol{A}_{cl}^{N-1}\boldsymbol{B}_{cl} & \boldsymbol{C}_{cl}\boldsymbol{A}_{cl}^{N-2}\boldsymbol{B}_{cl} & \cdots & \boldsymbol{C}_{cl}\boldsymbol{B}_{cl} \end{pmatrix} \tag{3}$$

### 2.2. PD-ILC

The PD-ILC algorithm is defined, using the nomenclature in Fig. 1, the gains $k_P$, $k_D$ and the low pass filter **Q**, as follows:

$$\begin{aligned} \mathbf{u}_{j+1}[n] = \quad & \mathbf{Q}[z](\mathbf{u}_j[n] + k_P\,\mathbf{e}_j[n+1] \\ & + k_D(\mathbf{e}_j[n+1] - \mathbf{e}_j[n])) \end{aligned} \tag{4}$$

A stability criterion and a condition for monotonic convergence is presented in [1]. It is shown that, in order to ensure convergence, a low pass filter, **Q**, is required.

### 2.3. CO-ILC

CO-ILC is a special case of the optimization based ILC framework presented in [8]. The difference of the input between subsequent iterations is given by

$$\Delta\mathbf{u}_{j+1} = \mathbf{u}_{j+1} - \mathbf{u}_j\,. \tag{5}$$

The predicted error $\mathbf{e}_{j+1}$ of the next iteration, using the linear plant model $\mathbf{P}_{cl}$, is given by

$$\mathbf{e}_{j+1} = \mathbf{e}_j - \mathbf{P}_{cl}\,\Delta\mathbf{u}_{j+1}\,. \tag{6}$$

The cost function of the optimization problem consists of two parts, the tracking error and the weighted velocity of the input difference:

$$\mathbf{J}_1 = \mathbf{e}_j - \mathbf{P}_{cl}\,\Delta\mathbf{u}_{j+1} \text{ and } \mathbf{J}_2 = d\,\mathbf{D}\Delta\mathbf{u}_{j+1}\,. \tag{7}$$

Minimizing $\mathbf{J}_1^{\mathrm{T}}\mathbf{J}_1 + \mathbf{J}_2^{\mathrm{T}}\mathbf{J}_2$, using (5) as state vector and the scalar weighting factor $d$, the following quadratic program can be stated:

$$\min_{\Delta\mathbf{u}_{j+1}}\left(\frac{1}{2}\Delta\mathbf{u}_{j+1}^T\mathbf{H}\,\Delta\mathbf{u}_{j+1} + \mathbf{f}^T\Delta\mathbf{u}_{j+1}\right). \tag{8}$$

**H** and **f** are defined as

$$\begin{aligned} \mathbf{H} &= \mathbf{P}_{cl}^T\mathbf{P}_{cl} + d^2\,\mathbf{D}^T\mathbf{D} \\ \mathbf{f} &= -\mathbf{P}_{cl}^T\mathbf{e}_j \end{aligned}\,. \tag{9}$$

The matrix **D** is given by

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix}, \tag{10}$$

which leads to smooth input signals.

Without boundary conditions on the states and the assumption that the matrix **H** is positive semidefinite, the solution of the convex optimization problem can be found as

$$\Delta \mathbf{u}_{j+1} = \mathbf{H}^{-1}\left(\mathbf{e}_j^T \mathbf{P}_{cl}\right), \tag{11}$$

which is used for the input of the next iteration

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \Delta \mathbf{u}_{j+1}. \tag{12}$$

## 3. Numerical Results

The presented ILC algorithms are tested and compared using a positioning movement. The simulation model consists of a proportional cascaded closed-loop controlled unit mass, representing a second order model with band width $\omega_0$ and damping factor δ. The continuous transfer function is given by

$$G(s) = \frac{{\omega_0}^2}{s^2 + 2\omega_0 \delta s + {\omega_0}^2}. \tag{13}$$

Later, the gantry test bench, used for the measurements, is introduced. Finally, the measurements without ILC and using the CO-ILC are presented.

### 3.1. Simulation results

The two ILC methods presented in section 2 are tested and evaluated for a 1-dimensional system with a trajectory that consists of two positioning movements from 0 to 100 and from 100 to 50 mm. The first positioning speed is limited to 0.2 m/s, the second to 0.1 m/s. The trajectory used for the simulations is shown in Fig. 2.

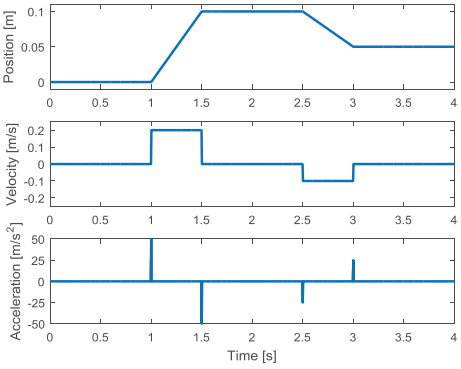


Fig. 2. 1-dimensional positioning.

The specifications of the simulation model (13) are set to a bandwidth of $\omega_0$=157 rad/s and a damping factor of δ=0.5.

The PD-ILC algorithm for tracking error reduction was used with a proportional gain of $k_P$=1.0, a derivative gain of $k_D$=3.25 and a low pass Q-filter (Butterworth) with cut-off frequency of $f_c$=50 Hz. These parameters were defined by tuning the algorithm to be asymptotically stable. The difference of input velocity is punished by a factor $d$=3 for the CO-ILC algorithm.

The infinity norm of the tracking error over 50 iterations for both methods is shown in Fig. 3. The tracking error is reduced by both methods, but the CO-ILC algorithm reaches a bigger reduction.

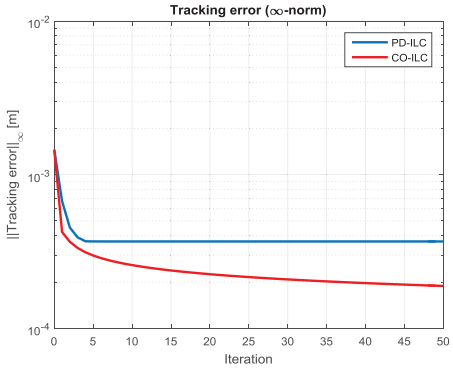


Fig. 3. Infinity norm of the tracking errors for the positioning for the PD- and CO-ILC.

### 3.2. System description

The test bench used for measurements in this paper is a two axes gantry stage, consisting of a gantry X-axis and a crossbeam Y-axis, shown in Fig. 4. All axes are driven by linear synchronous motors.

The control bandwidth of the X- and Y-axis has been set to 15 Hz using a cascaded loop velocity and position controller. The moving mass is 150 kg for the X-axes and 20 kg for the Y-axis. As internal measurement system, Heidenhain encoders were used. For the TCP measurement a Heidenhain cross grid was used.

Further, it needs to be mentioned that the slides of the X- and Y-axis show high stiction and a low first natural frequency of the crossbeam at around 20 Hz.

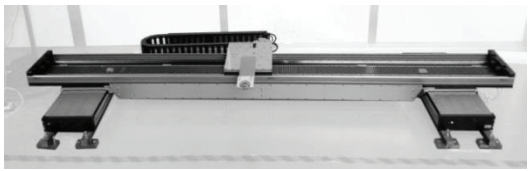

Fig.4. Two axes test bench used for measurements.

### 3.3. Measurements

Internal and external measurements, the encoder and the cross grid signals, respectively, are used and compared for the CO-ILC. The PD-ILC algorithm is not tested on the test bench due to better simulation results of CO-ILC. The test geometry is shown in Fig. 7. A feed rate of 0.1 m/s is used.

First, the internal measurements are used for the CO-ILC algorithm. The tracking error of each axis, compared to tracking without ILC, is significantly reduced as shown in Fig. 5. The contour error, shown in Fig. 6, is reduced too.

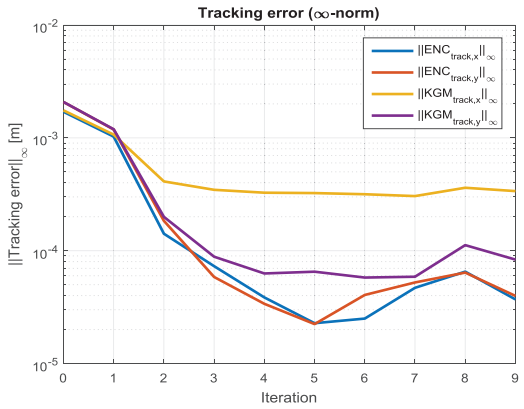


Fig. 5. Infinity norm of the tracking errors of the test trajectory for both, the internal encoder (ENC) and the TCP (KGM) measurements, respectively. Internal encoder measurements are used for the learning procedure.

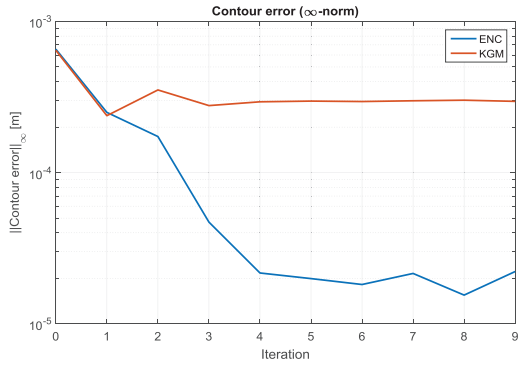


Fig. 6. Infinity norm of the contour errors for the internal encoders (ENC) and the TCP (KGM) measurements of the test trajectory. Internal encoder measurements are used for the learning procedure.

As expected, not the same amount of improvement can be reached for the TCP accuracy by using the internal encoder measurements for learning. The tracking and the contour errors of the TCP position are reduced, but errors due to not modelled dynamic effects cannot be compensated. On average, the contour error is reduced after 9 iterations, which is shown in Fig. 7 (green). An overshoot of the X-axis at the upper left corner of the test trajectory occurs, which is not seen by the internal measurement system and therefore, results in a bigger contour error. Due to reference deviations of the two X-axes, an orthogonality error occurs between the X- and Y-direction which leads to a contour error that can be clearly seen on the left part in Fig. 7. This deviation cannot be compensated using internal encoder measurements for learning.

Using cross grid measurements at the TCP for the CO-ILC algorithm, the contour error at the TCP is reduced significantly, which can be seen in Fig. 7 (blue).

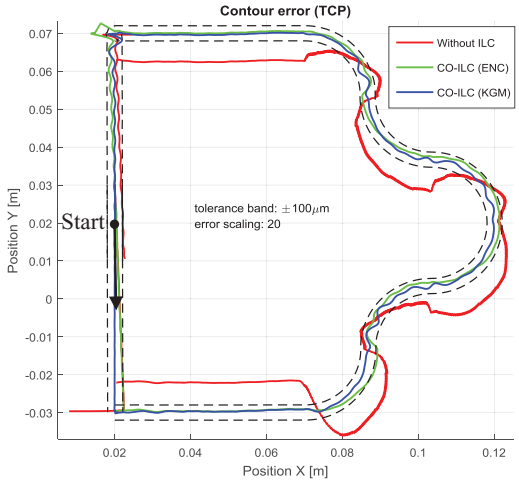


Fig. 7. Contour errors without ILC and after 9 iterations using the CO-ILC algorithm with internal encoder (ENC) and external cross grid (KGM) measurements. The feed rate is 0.1 m/s.

A comparison of the infinity norm of the tracking and contour error is given in Table 1. The best contour error reduction is possible by using TCP measurements. A reduction of 52% for the infinity norm of the contour error is possible by using internal encoder measurements and a reduction of 85%, by using TCP measurements.

Table 1. Infinity norm of the tracking and the contour error without ILC, after 9 iterations using internal (ENC) and TCP (KGM) measurements, respectively, for CO-ILC.

| Method | $\|\text{Tracking error}\|_\infty$ [µm] | $\|\text{Contour error}\|_\infty$ [µm] | $\|\text{Contour error}\|_\infty$ reduction |
|---|---|---|---|
| Without ILC | 1848 | 611 | - |
| CO-ILC (ENC) | 337 | 296 | 52 % |
| CO-ILC (KGM) | 138 | 89 | 85 % |

## 4. Conclusion, Outlook

Both ILC algorithms, PD-ILC and CO-ILC, are introduced first. It is shown by simulation that using PD-ILC results in less tracking error reduction, compared to cascaded loop trajectory control, than using CO-ILC. The tracking error and therefore, the contour error reduction using CO-ILC is shown on a gantry test bench using internal encoder and TCP measurements. The reduction of the contour error is 52% by using internal encoder and 85% by using TCP measurements for the CO-ILC algorithm. Dynamic effects and static errors, which are not detected by internal encoder measurements, are only compensated when using TCP measurements.

The implementation of CO-ILC on commercial NC kernels needs further effort. Especially if the optimization is performed on the NC itself.

Further studies with more accurate models of the machine tool and dynamic constraints on set points have to be made. An accurate measurement method for TCP measurements without using a cross grid has to be sought.

**Acknowledgements**

The authors would like to thank various Swiss industrial partners for their support.

**References**

[1] D. Bristow, M. Tharayil and A. Alleyne, «A survey of iterative learning control,» *Control Systems, IEEE,* Bd. 26, Nr. 3, pp. 96-114, June 2006.

[2] H.-S. Ahn, Y. Chen and K. L. Moore, «Iterative learning control: brief survey and categorization,» *IEEE Transactions on Systems Man and Cybernetics part C Applications and Reviews,* Bd. 37, Nr. 6, p. 1099, 2007.

[3] M. Garden, *Learning control of actuators in control systems,* Google Patents, 1971.

[4] S. Arimoto, S. Kawamura and F. Miyazaki, «Bettering Operation of Robots by Learning,» *Journal of Robotic Systems,* Bd. 1, Nr. 2, pp. 123-140, 1984.

[5] M. Togai and O. Yamano, «Analysis and design of an optimal learning control scheme for industrial robots: A discrete system approach,» in *Decision and Control, 1985 24th IEEE Conference on*, 1985.

[6] N. Amann, D. H. Owens and E. Rogers, *Iterative Learning Control for Discrete Time Systems with Exponential Rate of Convergence,* 1995.

[7] J. H. Lee, K. S. Lee and W. C. Kim, «Model-based iterative learning control with a quadratic criterion for time-varying linear systems,» *Automatica,* Bd. 36, Nr. 5, pp. 641-657, 2000.

[8] K. Barton, J. van de Wijdeven, A. Alleyne, O. Bosgra and M. Steinbuch, «Norm optimal Cross-Coupled Iterative Learning Control,» in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, 2008.

[9] D. Kim and S. Kim, «An iterative learning control method with application for CNC machine tools,» in *Industry Applications Society Annual Meeting, 1993., Conference Record of the 1993 IEEE*, 1993.

[10] C.-H. Tsai and C.-J. Chen, «Application of iterative path revision technique for laser cutting with controlled fracture,» *Optics and Lasers in Engineering,* Bd. 41, Nr. 1, pp. 189-204, 2004.

[11] M.-S. Tsai, C. L. Yen and H. T. Yau, «Development of a hybrid iterative learning control for contouring NURBS curves,» *Asian Journal of Control,* Bd. 13, Nr. 1, pp. 107-125, 2011.

[12] B. Altin and K. Barton, «Robust iterative learning for high precision motion control through adaptive feedback,» *Mechatronics ,* Bd. 24, Nr. 6, pp. 549-561, 2014.

[13] J. Wu, C. Liu, Z. Xiong and H. Ding, «Precise contour following for biaxial systems via an A-type iterative learning cross-coupled control algorithm,» *International Journal of Machine Tools and Manufacture ,* Bd. 93, pp. 10-18, 2015.

[14] S. Z. Khong, D. Nešić and M. Krstić, «Iterative learning control based on extremum seeking,» *Automatica,* Bd. 66, pp. 238-245, 2016.