Proceedings Paper:

Ward, R. orcid.org/0000-0002-6201-0285, Ozkirimli, O. and Jones, B. (2021) Increasing part geometric accuracy in high speed machining using cascade iterative learning control. In: Ozturk, E., Curtis, D. and Ghadbeigi, H., (eds.) Procedia CIRP. 9th CIRP Conference on High Performance Cutting, 24-26 May 2021, Online conference. Elsevier BV , pp. 298-301.

https://doi.org/10.1016/j.procir.2020.10.006

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 101 (2021) 298–301

www.elsevier.com/locate/procedia

9th CIRP Conference on High Performance Cutting (HPC 2020)

Increasing Part Geometric Accuracy in High Speed Machining using Cascade Iterative Learning Control

Rob Ward[a,c,*], Omer Ozkirimli[b], Bryn Jones[c]

[a]Industrial Doctorate Centre in Machining Science, Advanced Manufacturing Research Centre with Boeing, University of Sheffield, Rotherham, S60 5TZ, UK
[b]Machining Dynamics Research Group, Advanced Manufacturing Research Centre with Boeing, University of Sheffield, Rotherham, S60 5TZ, UK
[c]Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield,S1 3JD

Abstract

High speed machining provides high productivity and low machining cycle times. Post machining, there can exist differences between desired and measured part geometry due to tool deflection induced from higher feedrates. Reducing the feedrate leads to an increase in machining time. Using predicted drive responses on a virtual CNC with an integrated surface location error model, this research is the first time Iterative Learning Control (ILC) has been applied to reduce part geometry errors from tool deflection. Validation machining trials demonstrated that the ILC scheme improved machining performance whilst maintaining machining times when compared to a baseline part program.

Keywords: Iterative Learning Control; Form Error Reduction; Milling;

1. Introduction

The primary objectives in manufacturing industries are to minimise overall production time, maintain quality and continuously seek efficiencies whilst remaining responsive to an ever increasing complex set of customer demands. Within all manufacturing methods there is a drive to create parts with tighter tolerance requirements whilst reducing the time to machine.

This research seeks to reduce part geometric errors and machining time through Iterative Learning Control (ILC) and tool deflection compensation by using a virtual model of the machine tool. The virtual model includes feed drives and a frequency domain surface location error (SLE) model. Using inputs of an unmodified part program, modal data from impact testing and pre-machining workpiece On-Machine Inspection (OMI) data, the system is able to improve machining performance, which is defined as a reduction in form error with no increase to machining time, by modifying a new part program designed using an ILC methodology.

* Corresponding author.
E-mail address: raward1@sheffield.ac.uk (Rob Ward).

1.1. Tool Deflection and Machine Tool Compensation

For part geometric errors caused by tool deflection during machining, many solutions have been explored which are generally split into offline and online methods. Online methods include the use of integrated force sensors and dynamometers [1] to predict tool deflection. Offline methods include calculating tool deflection analytically based upon process force estimation and tool stiffness [2] followed by compensation via feedrate and/or toolpath modification to adjust the radial and/or axial depth of cut as required. Increasing the accuracy of the offline and online models can be achieved by accurate workpiece holding and measurements - this can be achieved by accurate process monitoring and in-situ geometric measurements or OMI.

1.2. Iterative Learning Control

Iterative Learning Control is a control scheme which improves performance by reducing repeated errors in repetitive processes. Similar to human learning, by repeating the same action over again, a person is able to learn from the previous action and update the next action in order to minimise the error or expressed differently - the actions become increasingly more accurate. The practical applications of ILC have focused on industrial control processes and predominately have used the discrete domain and the lifted system representation [3]. Start-

10.1016/j.procir.2020.10.006

ing from the repetitive motions of robotic arms [4], ILC has been successfully applied to CNC position control, injection moulding, welding robots, chemical reactors and many more repetitive industrial processes. ILC has never been applied to reduction of form error in machining by tool deflection compensation.

Despite this, Commercial Computer Numerically Controlled (CNC) machines provide the ideal platform for application of ILC methodologies due to the nature of the manufacturing process. Batch CNC machining fulfils some of the key ILC system requirements as defined by Arimoto [5] such as each individual CNC process has the same initial conditions (as the program is reset each time), the same part is to be machined each time (therefore both workpiece and material to be removed are the same) and the CNC machine has measurable outputs via data acquisition. This provides the motivation for the application of ILC to CNC machining.

1.3. Cascade Iterative Learning Control

Cascade ILC [6] is formulated as an additional (cascaded) loop around an existing control architecture (similar to a standard feedback loop) but in the iteration domain rather than the time domain. Therefore the new structure uses the output from the existing system to shape the next input. The new system is not integrated or embedded into the existing controller and hence no rewriting of control software or replacement of controller hardware is required. This makes cascade ILC ideally suited to CNC machining centres where the interpolator and motion controller are closed architectures. The re-design of the input reference signal is the main control effort of the cascade ILC method and this corresponds to an offline modification of the NC code for CNC machines.

The formulation of the cascade ILC algorithms is as follows:

$$\begin{aligned} Y_i &= GU_i, \\ E_i &= Y_r - Y_i, \\ U_{i+1} &= U_i + \Gamma E_i, \end{aligned} \tag{1}$$

where, Y_i, G, and U_i represent the system output, plant model closed loop transfer function and system input respectively. The current iteration or trial is denoted by i with the n^{th} iteration $i{+}n$. The current error vector E_i is generated from the difference between the reference trajectory Y_r and current output Y_i. The next input signal U_{i+1} is generated from the previous input signal U_i updated with the current error signal E_i multiplied by a learning function Γ. In order to generate E_i both Y_i and Y_r must have the same dimensions - this is the fixed vector length. The design of the learning function is analogous to controller gain design in standard control systems. Cascade ILC is designed using the previous cycle learning (as opposed to current cycle) as once the input trajectory has been designed and executed it is generally closed to changes within real-time processes much like the operation of CNC machines.

Applying ILC to CNC machining requires a reformulation of the cascade ILC equations. Equations (1) assume a fixed interval or trial length (in the time domain) which is ILC's primary requirement [5]. However, due to the varying machining time in machining, this research introduces the discrete command index. The control input in CNC machining is set by the NC part program and the NC code is made up of a series of discrete 'G01' cutter location position and feedrate commands. These discrete commands form the core principle why ILC can be applied to CNC machining. The number of 'G01' commands in the NC code defines the fixed vector length. This is known as the discrete command index and it is used throughout this research to meet ILC's fixed length vector requirement. The CNC cascade ILC update equations become the following:

$$U_{i+1}(k) = U_i(k) + \Gamma(k)E_i(k), \tag{2}$$

where k is the discrete command index which relates to the command line number.

2. Virtual CNC Model

The virtual CNC model is made up of a number of submodels as shown in Figure 1. The reference trajectory was generated using a developed axes drive model which is based upon cubic acceleration profiles. Jerk time constants were calculated based upon experimental trials on the selected DMG DMU eVo 40 5-axis machining centre fitted with Heidenhain TNC640 controller. The reference trajectory in the time domain is the input to the feed drive model. The modified part program generated from the ILC algorithm is created using MATLAB which produces a Heidenhain (.h) part program file.

2.1. Feed Drive Modelling

The architecture of a commercial CNC feed drive is a cascaded series of PID loops with a variety of feedforward compensations and structural dynamics filters. For simulation purposes, the linear feed drives can be modelled as second order transfer functions with time delays as shown in Equation (3).

$$G_v(s) := \frac{V(s)}{U(s)} = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} e^{-sT_d} \tag{3}$$

Where $V(s)$ and $U(s)$ represent the drive velocity and control input respectively. Using the MATLAB System Identification Toolbox [7], the parameters in response to a 1000 mm/min step input in velocity results in the following parameters: gain $K = 1.09$ x 10^{-4}, cutoff frequency $\omega_0 = 95.6$ rad/s and damping ratio $\zeta = 1.0$ and time delay $T_d = 0.0066$s . The tool position is calculated based upon integrating the velocity profile of the toolpath [8]. Within this research the feed drives are modelled for the X and Y axes only.

2.2. Frequency Response Surface Location Error Model

The SLE is the maximum distance between the milled surface and the desired surface [9]. The milled surface is generated from the path of the cutting edges rotating at tooth passing frequency and their forced vibrations. The SLE model in this research uses the frequency response method [10]. Using impact testing to determine the direct tool frequency response function (FRF), the tool displacement is calculated from the inverse Fourier transform of the tool displacement in the frequency do-

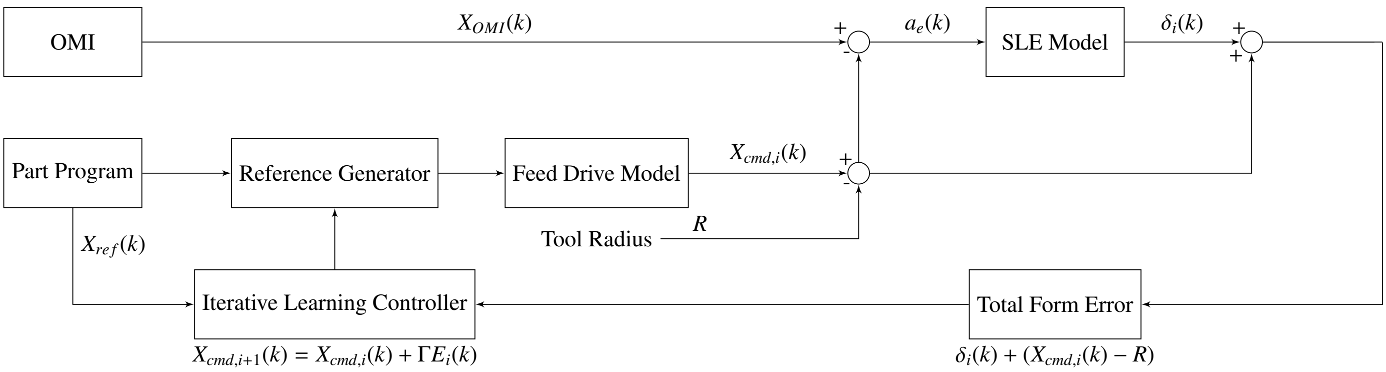

Fig. 1: Virtual CNC Model with Cascaded ILC

main. The subsequent milled surface and thus the SLE is calculated based upon the tool displacement. The model is a function of the radial depths of cut calculated from the pre-machined workpiece OMI and the axis positions based upon the feed drive model responses. The result is a surface location error map calculated along the axial depth of cut. Runout was not included in the SLE prediction model.

2.3. ILC Form Error Reduction Formulation

The ILC equations 1 can be reformulated for the virtual CNC model (Figure 1) as follows:

$$\begin{aligned} a_e(k) &= X_{OMI}(k) - (X_{cmd,i}(k) - R) \\ \delta_i(k) &= f(a_e) \\ E_i(k) &= X_{ref}(k) + (\delta_i(k) + (X_{cmd,i}(k) - R)) \\ X_{cmd,i+1}(k) &= X_{cmd,i}(k) + \Gamma E_i(k), \end{aligned} \tag{4}$$

where a_e, R, $\delta_i(k)$ represent the radial depth of cut, tool radius and surface location error respectively. The frequency response surface location error model is represented as $f(a_e)$. X_{OMI}, $X_{cmd,i}(k)$, $X_{ref}(k)$ and $E_i(k)$ denote the precut OMI position, tool position command, surface reference geometry and ILC error respectively.

3. Methodology

The machining trials were conducted on the DMU eVo 40 machine tool. TNCremo software [11] was used to transfer files between a laptop and the TNC640 controller. A 2-fluted 12mm Sandvik 2P170-1200-NA H10F solid carbide end mill with a HSK-63A tool holder was used. The tool runout at the tip was measured as 30µm. The workpieces were 236mm x 30mm x 66mm Aluminium 7075-T6 blocks machined with 3mm and 6mm stepped machined surfaces (see Figure 2). A DMG Mori 60 Optical PowerProbe calibrated with a measured repeatable accuracy of less than ±0.08µm was used for OMI.

3.1. ILC Simulation

The toolpath was selected as a linear up-milling cut along the y axis, which due to the workpiece geometry, demonstrates varying radial immersion (75%, 50% and 25% tool diameter) along the cut. The simulation is updated with the workpiece pre-

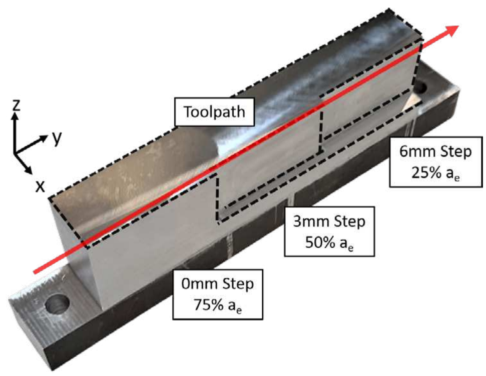

Fig. 2: Workpiece showing stepped surfaces and the radial depths of cut

machining OMI data to reflect the stock geometry. Along with the pre-cut OMI data the inputs to the virtual CNC (as shown in Figure 1) are the original part program (for initial conditions and reference toolpath) and the SLE model requires measured tool length, tool radius and FRF data from impact testing. The virtual ILC based model considers the maximum SLE as the reference and updates the X-axis position (changing the effective radial depth of cut) in the subsequent part program.

ILC learning gains are traditionally designed in either the frequency domain or by state space methods, however, the design in the position domain has yet to be explored. In this particular case study, to simply prove the application of ILC to machining, the selection of the ILC learning function was chosen to be a scalar. From a comparison study, a learning gain Γ of 0.5 was chosen which demonstrates a relatively fast convergence (within 7 iterations) whilst demonstrating both monotonic convergence and asymptotic stability as shown in Figure 3. Further research into the design of ILC learning gains in the position domain is part of the wider remit of this research.

3.2. Experimental Validation

The modified part program was validated by machining trials. The workpiece geometry (x-axis position) was measured using OMI and used as an input to the virtual CNC system. The modified part program generated from the ILC algorithm was conducted on the workpiece. Post machining OMI (in the

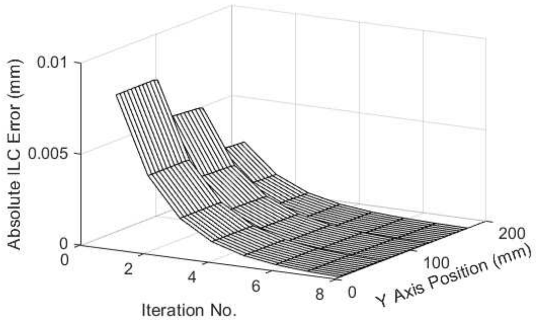

Fig. 3: Propagation of ILC Error through iteration

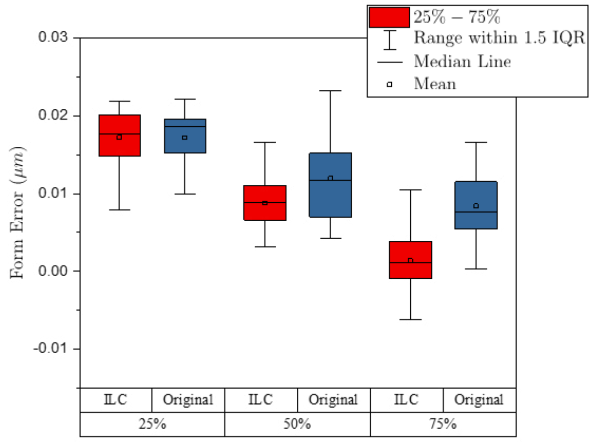

Fig. 4: Total form error for ILC and default toolpaths for 25%, 50% and 75% radial immersion

x-axis direction) was conducted to verify the results. These results were compared to machining cuts with an unmodified default part program of a straight line cut (Figure 4). The machining conditions were as follows: Feedrate 403.4 mm/min, cutting speed (V_c) 76.01 m/min and axial depth of cut (a_p) 8mm.

4. Experimental Results

Using an ILC learning gain of 0.5 there is a clear ILC error convergence within 7 iterations as shown in Figure 3. The modified part programs with updated x-axis positions based upon the converged ILC input commands were conducted. The test was repeated 3 times for both an unmodified (default) part program and modified (ILC) part program. The results showing total form error (measured position minus reference position) are shown in Figure 4. The results show a reduction in surface location error from a reference surface when using the virtual ILC system. The accuracy in surface generation position with respect to the reference surface is increased with increasing tool radial immersion. For 75% radial immersion the mean deviation from reference is 1.37μm when using ILC as compared to 8.41μm for the original toolpath which shows a clear reduction in maximum SLE. For 50% immersion the deviation from reference value is 8.76μm and 11.96μm for ILC and default toolpaths respectively. Finally for the 25% immersion cut the values are 17.15μm and 17.21μm for ILC and default toolpaths respectively.

5. Conclusions

The main findings of the work are:

- The accuracy of desired surface generation through form error reduction is improved by using a modified part program generated from a virtual CNC model using Iterative Learning Control.
- The ILC based virtual model shows greater performance with increasing radial depth of cut.
- Using a scalar ILC learning gain of 0.5 a monotonic convergence and asymptotic stability is demonstrated within 7 iterations.

The utility of ILC has been demonstrated but further research into the impact and design of learning gains at the discrete command indices will be investigated. In order for the research to be applicable to industrial applications further research will apply ILC to 3 and 5 axis geometries.

Acknowledgements

This research was supported by the Advanced Manufacturing Research Centre Members and EPSRC (grant EP/L016257/1).

References

[1] B. Denkena, H. Boujnah, Feeling Machines for Online Detection and Compensation of Tool Deflection in Milling, CIRP Ann. 67 (1) (2018) 423–426.
[2] P. Dépincé, J. Y. Hascoët, Active Integration of Tool Deflection Rffects in End Milling. Part 1. Prediction of Milled Surfaces, Int. J. Mach. Tools Manuf. 46 (9) (2006) 937–944.
[3] B. A. Francis, C. Ms, A. Tannenbaum, A Lifting Technique for Linear Periodic Systems with Applications to Sampled-Data Control *, Syst. Control Lett. 17 (1991) 79–88.
[4] M. Norrl, S. Link, An Adaptive Iterative Learning Control Algorithm with Experiments on an Industrial Robot, Department of Electrical Engineering, IEEE Trans. Robot. Autom. (2002).
[5] S. Arimoto, S. Kawamura, F. Miyazaki, Bettering Operation of Robots by Learning, J. Robot. Syst. 1 (2) (1984) 123–140.
[6] J.-X. Xu, S. K. Panda, T. H. Lee, Real-time Iterative Learning Control, Springer, 2009.
[7] MATLAB System Identification Toolbox.
[8] S. Stemmler, D. Abel, M. Schwenzer, O. Adams, F. Klocke, Model Predictive Control for Force Control in Milling, IFAC-PapersOnLine 50 (1) (2017) 15871–15876.
[9] A. K. Kiss, D. Bachrathy, G. Stepan, Cumulative Surface Location Error for Milling Processes Based on Tool-tip Frequency Response Function, Procedia CIRP 46 (2016) 323–326.
[10] K. S. Smith, T. L. Schmitz, Machining Dynamics: Frequency Response to Improved Productivity, Springer 2008.
[11] TNCremo Software, HEIDENHAIN NC Tools, DR JOHANNES HEIDENHAIN GmbH.