Title

Experimental performance analysis of an inverse dynamics CNC compensation scheme for high-speed execution of curved toolpaths

Journal

The International Journal of Advanced Manufacturing Technology, 73(1-4)

ISSN

0268-3768

Authors

Schraeder, Travis F
Farouki, Rida T

Publication Date

2014-07-01

DOI

10.1007/s00170-014-5720-z

Peer reviewed

Experimental performance analysis of an inverse dynamics CNC compensation scheme for high–speed execution of curved toolpaths

Travis F. Schraeder and Rida T. Farouki
Department of Mechanical and Aerospace Engineering,
University of California, Davis, CA 95616, USA.

Abstract

Experimental results from the implementation of an inverse dynamics compensation scheme on a 3–axis CNC mill with an open–architecture software controller are presented. The scheme amounts to imposing a continuously–variable displacement on the commanded toolpath, that compensates for the physical limitations (inertia and damping) of each machine axis, producing accurate execution of the prescribed toolpath. Numerical values for the dimensionless parameters that characterize the second–order inverse dynamics model are determined by feeding input and output data from machine runs into a system identification software tool. For a basic P–type controller, exact *a priori* solutions for the modified path geometry are possible, allowing implementation by simple alterations to the real–time interpolator. The experimental results based on a P–type controller indicate that the inverse dynamics scheme is highly effective in suppressing both feed (lag/lead) error and contour error (deviation from the desired path), even at high feedrates along strongly–curved toolpaths. The scheme thus provides a practical means of achieving smooth and accurate execution of free–form paths, without appealing to more complicated "active" control strategies.

Keywords: CNC machine, system identification, inverse dynamics, feedrate, P controller, contour error, feed error, path modification, real–time interpolator.

e–mail addresses: tfschraeder@ucdavis.edu, farouki@ucdavis.edu

1 Introduction

In a previous study [5], a scheme was presented that attempts to improve the accuracy of 3–axis CNC machines, when executing strongly–curved paths at high feedrates, by compensating for the inertia and damping of each axis. The scheme was based on a physical model of the axis drive system, governed by a PID controller. By transforming the variable in the dynamic model from time to the curve parameter, it was shown that the desired compensation can be achieved through a continuously–variable path modification, specified as the solution of a certain system of ordinary differential equations.

The focus in [5] was on formulation of the basic mathematical framework and algorithms for the inverse dynamics compensation (IDC) scheme, and illustration of its use by software simulation. The focus of the present study is on an experimental verification of the practical effectiveness of this scheme on a real 3–axis CNC machine, using model parameters obtained by analysis of input/output data from runs on the machine with a system identification software tool. The simulations in [5] cannot adequately predict the practical value of the IDC scheme, since a dynamic model cannot incorporate every physical limitation on the accuracy of a machine. The results described below demonstrate that, using a simple 2nd–order dynamic model with appropriate parameters, the IDC is extremely effective in suppressing motion errors due to axis inertia and damping when the axis accelerations implied by the feedrate and path curvature remain well within the capacity of the axis drive systems. This effectiveness is largely attributable to the use of analytic curved path descriptions, which admit exact closed–form expressions for the reference–point displacements required in each servosystem sampling interval.

Whereas the scheme in [5] was presented in the general context of a PID controller, the experiments described below employ only a P controller. The motivation for this is two–fold. In the first place, a simple and *exact* solution for the path modification is possible for P control. For PI or PID control, on the other hand, the path–modification differential equations no longer admit closed–form solutions. If approximations are employed, great care should be exercised in ensuring that the approximation error is much smaller than the typical machine positioning errors attributable to axis inertia and damping (which are themselves typically very small). In the second place, coupling the inverse dynamics scheme with a P controller is observed to yield performance equalling or surpassing that of PI or PID control. For example, an integrator is typically employed to suppress steady–state (lag) error along the toolpath.

But, as described below, the inverse dynamics compensation proves equally effective in this respect — when using the compensator, there is no discernible feed error improvement with a PI controller over a P controller.

As in [5], the present study emphasizes *planar* paths, since the extension to spatial paths is straightforward. Moreover, we focus only on the intrinsic machine dynamics, and do not consider cutting forces, external disturbances, etc. Dynamic forces are often dominant in *high–speed machining* (HSM) of complicated shapes, incurring large rates of acceleration/deceleration [2, 14, 16, 17]. It should also be emphasized that the scheme requires the toolpath geometry to be communicated to the real–time interpolator as a sequence of extended analytic curve segments — it is not compatible with part programs employing numerous short linear/circular G code segments.

We now outline the plan for the remainder of this paper. Following a brief review of the inverse dynamics model in Section 2, the system identification procedure (that determines the model parameters characterizing the specific CNC machine employed in the experiments) is described in Section 3. The experimental procedure, data analysis methods, and sample test curves used to assess the performance of the inverse dynamics scheme are then presented in Section 4. Section 5 describes the results of the experiments and discusses their significance. Finally, Section 6 summarizes the key findings of this study and identifies some problems that deserve further investigation.

2 Inverse dynamics model

Standard dynamic models [1, 4, 20] for CNC axis drive systems governed by PID controllers relate the commanded path $(X(t), Y(t))$ to the actual path $(x(t), y(t))$ executed by a machine through differential equations of the form

$$\begin{aligned} a_x \dddot{x} + b_x \ddot{x} + c_x \dot{x} + x &= d_x \ddot{X} + e_x \dot{X} + X \,, \\ a_y \dddot{y} + b_y \ddot{y} + c_y \dot{y} + y &= d_y \ddot{Y} + e_y \dot{Y} + Y \,, \end{aligned} \tag{1}$$

where dots denote derivatives with respect to time t. For brevity, we consider here only planar motion (the extension to spatial paths is elementary).

The block diagram of a typical system modelled by equations (1) is shown (for the x–axis) in Figure 1. The physical variables (and their dimensions) are as follows: u (V) is the control variable; k_a (A/V) and k_t (N m/A) are the current amplifier and motor torque gains; J (kg m^2) and B (kg m^2/s) are the

x–axis inertia and damping; T (N m), ω (rad/s), and θ (rad) are the motor torque, angular speed, and position; r_g (m/rad) is the transmission ratio (the axis translation for unit rotation of the motor shaft); and k_p, k_i, k_d are the proportional, integral, and derivative gains for the PID controller.

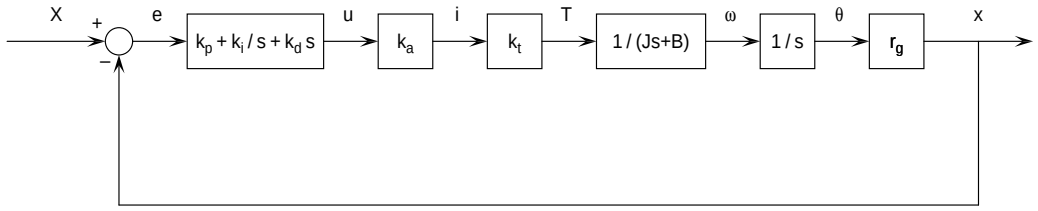

Figure 1: Block diagram for a dynamic model of the x–axis drive system.

In Figure 1, $e = X - x$ is the axis position error, i.e., the difference between the commanded and actual axis locations. The power amplifier converts the actuating signal u from the controller into a current i to the motor, which produces a torque T that determines the angular speed ω through the axis inertia J and damping B. The motor shaft angle θ, obtained by integrating ω, determines the axis linear position x through the transmission ratio r_g.

Setting $K = k_a k_t r_g$, the transfer function relating the Laplace transforms $x(s)$ and $X(s)$ of the output and input for the system in Figure 1 is

$$\frac{x(s)}{X(s)} = \frac{K(k_d s^2 + k_p s + k_i)}{Js^3 + (B + Kk_d)s^2 + Kk_p s + Kk_i},$$

and the coefficients in the differential equations (1) are given by

$$a_x = \frac{J}{Kk_i}, \quad b_x = \frac{B + Kk_d}{Kk_i}, \quad c_x = \frac{k_p}{k_i}, \quad d_x = \frac{k_d}{k_i}, \quad e_x = \frac{k_p}{k_i}.$$

Analogous results hold for the y–axis dynamics, but with physical parameters appropriate to that axis. Considerable simplification arises in the case of a P controller ($k_i = k_d = 0$), for which the transfer function reduces to

$$\frac{x(s)}{X(s)} = \frac{Kk_p}{Js^2 + Bs + Kk_p}, \tag{2}$$

and the coefficients in (1) become

$$a_x = 0, \quad b_x = \frac{J}{Kk_p}, \quad c_x = \frac{B}{Kk_p}, \quad d_x = 0, \quad e_x = 0, \tag{3}$$

with analogous results for the y–axis. Note that, in this case, the derivatives of $X(t)$, $Y(t)$ do not appear on the right–hand side in (1).

Suppose the toolpath is defined as a parametric curve $\mathbf{R}(\xi) = (X(\xi), Y(\xi))$. The *parametric speed* of $\mathbf{R}(\xi)$, defined as

$$\sigma(\xi) \;=\; |\mathbf{R}'(\xi)| \;=\; \sqrt{X'^2(\xi) + Y'^2(\xi)} \;=\; \frac{\mathrm{d}s}{\mathrm{d}\xi}\,,$$

specifies the rate of change of arc length s with respect to the parameter ξ. The IDC scheme proposed in [5] transforms the differential equations (1) so as to make the curve parameter ξ, rather than the time t, the independent variable. This is accomplished by invoking the chain rule to relate derivatives with respect t and ξ according to

$$\frac{\mathrm{d}}{\mathrm{d}t} \;=\; \frac{\mathrm{d}s}{\mathrm{d}t}\,\frac{\mathrm{d}\xi}{\mathrm{d}s}\,\frac{\mathrm{d}}{\mathrm{d}\xi} \;=\; \frac{V}{\sigma}\,\frac{\mathrm{d}}{\mathrm{d}\xi}\,, \tag{4}$$

where $V = \mathrm{d}s/\mathrm{d}t$ is the specified *feedrate* (which may be constant or variable) along the toolpath. This transformation results in differential equations with *non–constant* coefficients, and the goal is to solve them so as to determine a "modified" input path which, when modulated by the machine dynamics, yields the original commanded path as the actual machine motion [5].

For a general PID controller, exact solutions are not possible. However, specializing to the case of a P controller yields a simple and exact solution for the modified path. This may be interpreted as a "variable offset" $\Delta\mathbf{R}(\xi) = (\Delta X(\xi), \Delta Y(\xi))$ to the original commanded path $\mathbf{R}(\xi) = (X(\xi), Y(\xi))$, with displacement components specified [5] by

$$\begin{aligned} \Delta X \;&=\; b_x(V/\sigma)^2 X'' \;+\; (V/\sigma)\,[\,b_x(V/\sigma)' + c_x\,]\,X'\,, \\ \Delta Y \;&=\; b_y(V/\sigma)^2 Y'' \;+\; (V/\sigma)\,[\,b_y(V/\sigma)' + c_y\,]\,Y'\,, \end{aligned} \tag{5}$$

where primes denote derivatives with respect to ξ, $(V/\sigma)' = (\sigma V' - \sigma' V)/\sigma^2$, $\sigma' = \mathbf{R}' \cdot \mathbf{R}''/\sigma$, and b_x, c_x are defined by (3) with analogous expressions for b_y, c_y. Note that $\Delta\mathbf{R}(\xi)$ varies in *magnitude* and *direction* along $\mathbf{R}(\xi)$. If the feedrate is specified as a function of time, rather than the curve parameter, then $V' = \sigma\dot{V}/V$. Obviously, for a constant feedrate, $V' = 0$.

The displacement $\Delta\mathbf{R}(\xi)$ to the commanded path $\mathbf{R}(\xi)$, that compensates for the inertia and damping of the machine axes, is amenable to real–time computation if the parameters b_x, c_x, b_y, c_y are passed to the interpolator,

in addition to the commanded path $\mathbf{R}(\xi)$ and feedrate V. Because of the simplicity and exactitude of the solution, and also (see below) of the excellent results it yields, we focus henceforth on the case of a P controller.

3 System identification

As noted in [3], identification procedures for CNC axis drive systems typically entail performing a series of time or frequency domain response tests with the servo loop or interpolator disconnected. A simpler alternative approach — based on running standard G code test programs, and using an optimization scheme to determine the parameters of a postulated axis transfer function that best models the resulting measured axis position errors — was proposed in [3]. Our initial attempt at system identification was based on G code part programs generated by randomly–distributed points over a fixed interval on each axis, with feedrates set according to the method in [3].

During the tests, the real–time reference point and position encoder data were recorded for both the x and y axes, and subsequently fed into a custom `MATLAB` program, to determine a best–fit transfer function of the form (2). A typical result obtained (for the x–axis) by this method is

$$\frac{x(s)}{X(s)} \;=\; \frac{24900}{s^2 + 271.1\, s + 24900}\,,$$

from which one can identify the b_x and c_x values in (3) as

$$b_x \;=\; \frac{1}{24900} \;=\; 0.00004016\,, \qquad c_x \;=\; \frac{271.1}{24900} \;=\; 0.01089\,.$$

The focus of this study is inverse dynamics compensation for execution of smooth analytic curve segments, rather than discrete G code part programs. It was observed empirically that more accurate system parameters could be determined by using an extended C^2 spline curve that simultaneously excites the machine axes over their entire dynamic ranges. The "`clover`" test curve (see Section 4 below) proved to be an excellent choice for this purpose.

The b_x, c_x, b_y, c_y parameter values obtained from the system identification process, by executing the `clover` curve at feedrates increasing from 100 ipm to 800 ipm, are listed in Table 1. Clearly, c_x and c_y are essentially identical and vary by less than 1% over the full range of feedrate. However, b_x and b_y appear to decrease systematically with increasing feedrate, asymptoting to

feedrate	b_x	c_x	b_y	c_y
100 ipm	0.00006079	0.010888	0.00005112	0.010946
200 ipm	0.00004016	0.010888	0.00003624	0.010903
300 ipm	0.00003561	0.010897	0.00003214	0.010909
400 ipm	0.00003287	0.010897	0.00003000	0.010909
500 ipm	0.00002986	0.010896	0.00002926	0.010910
600 ipm	0.00002863	0.010898	0.00002870	0.010910
700 ipm	0.00002828	0.010892	0.00002839	0.010908
800 ipm	0.00002812	0.010889	0.00002815	0.010903

Table 1: Values of the dimensionless parameters b_x, c_x, b_y, c_y obtained from the system identification procedure at feedrates V from 100 ipm to 800 ipm.

nearly identical values for feedrates above ~ 500 ipm, as evident in Figure 2. This may be attributable to the fact that, at high feedrates, dynamic forces dominate all other factors. Since the asymptotic values $b_x = b_y = 0.0000281$ and $c_x = c_y = 0.0109$ were observed to yield excellent results at all feedrates, they have been adopted in all the experiments described below.

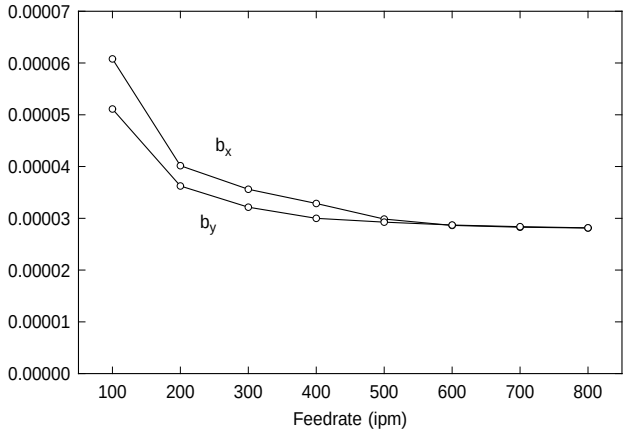

Figure 2: Variation of the b_x and b_y parameters (as determined by the system identification procedure) with the feedrate V between 100 ipm and 800 ipm.

In addition to the axis inertia and damping parameters, the maximum physically feasible acceleration for each axis should be determined. The axis acceleration limits arise from the torque capacity of the axis drive motors and

the axis inertias. If the path geometry and feedrate incur axis accelerations exceeding these limits, the machine will fail to realize the commanded motion and the IDC can no longer be expected to satisfactorily compensate for the axis inertia and damping. To identify the acceleration limits, each axis was commanded to accelerate from rest to a speed of 900 ipm at rates increasing from 100 in/s^2 to 700 in/s^2 with increments of 100 in/s^2.

By a second–order differencing of the position encoder data, it is possible to determine the actual axis accelerations achieved by the machine. Figure 3 shows the actual mean x–axis acceleration, up to a final speed of 900 ipm, for commanded accelerations increasing from 100 to 700 in/sec^2. The machine faithfully reproduces low commanded accelerations, but at $\sim$ 250 in/s^2 and above the actual acceleration becomes increasingly deficient relative to the commanded acceleration. Similar results were observed for the y–axis. Based on these observations, a conservative bound of 250 in/s^2 on the physically acheivable acceleration of each axis has been adopted.

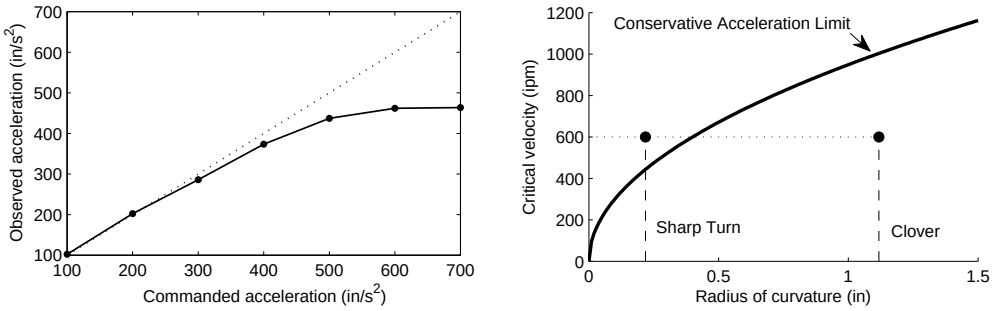

Figure 3: Left: the mean acceleration achieved by the x–axis, for commanded accelerations from rest to a speed of 900 ipm between 100 and 700 in/sec^2. Right: maximum constant feedrate versus minimum path radius of curvature consistent with a 250 in/sec^2 acceleration bound. Minimum radii of curvature for the `clover` and `sharp-turn` curves (see Section 4 below) are indicated.

Figure 3 also presents a different interpretation of the acceleration limit $a_{\max} = 250$ in/s^2. For a given curve executed at a fixed feedrate V, the purely centripetal acceleration $a = V^2/r$ is largest where the radius of curvature r has its minimum value $r_{\min}$. The condition $a \leq a_{\max}$ can thus be interpreted as defining an upper bound $V_{\max} = \sqrt{a_{\max} r_{\min}}$ on the constant feedrate that will avoid violation of the acceleration bound.

The "`clover`" and "`sharp-turn`" test curves described in Section 4 were

designed so that in the first case the acceleration limit is not exceeded, but in the second case it is. The experimental results for these curves (see Section 5) corroborate the expectation that the IDC gives excellent results if the axis acceleration bounds are observed, but poor results if they are violated.

4 Test curves and data analysis

Two curves, shown in Figures 4 and 6, were selected to test the performance of the inverse dynamics compensation scheme, at a sequence of increasing (constant) feedrates. These test curves were designed with strong curvature variations, so as to incur substantial centripetal accelerations. To minimize the effect of transients on the performance results, tangential linear "lead–in" and "lead–out" segments were included, to allow the machine to accelerate from rest before entering the test curve, and decelerate to rest after leaving it. The lead–in/out segments employ cubic ramp–up/down feedrate profiles.

The "`clover`" test curve (see Figure 4) was constructed as a periodic C^2 Pythagorean–hodograph (PH) quintic spline curve interpolating a sequence of 13 points [7]. Figure 5 illustrates the curvature profile for this curve. The maximum curvature is $\kappa_{\max} \approx 0.8940$ in^{-1} and the corresponding minimum radius of curvature is $r_{\min} \approx 1.1186$ in.

The "`sharp-turn`" curve (see Figure 6), on the other hand, is a single PH quintic segment, constructed as a Hermite interpolant [10], that exhibits a single pronounced curvature extremum: this is (a scaled version of) the test curve used in [5]. Figure 7 illustrates the curvature profile for this curve. The maximum curvature magnitude is $\kappa_{\max} \approx 4.5945$ in^{-1} and the corresponding minimum radius of curvature is $r_{\min} \approx 0.2177$ in.

PH curves were selected as test cases, because of the availability of simple and exact real–time CNC interpolators that can drive the machine from the analytic path description [8, 9, 11] — further details on the construction and applications of PH curves may be found in [6]. It should be noted, however, that the inverse dynamics compensation scheme is applicable to *any* analytic curves that admit accurate real–time interpolators [12, 13, 15, 18, 19] and for which precise real–time evaluation of expressions (5) is possible.

The experiments were performed on a 3–axis CNC milling machine with axes powered by independent Yaskawa servomotors driving precision–ground ball screws. The machine is controlled by the OpenCNC software, running on an off–the–shelf PC with a 500 MHz processor. The position loop sampling

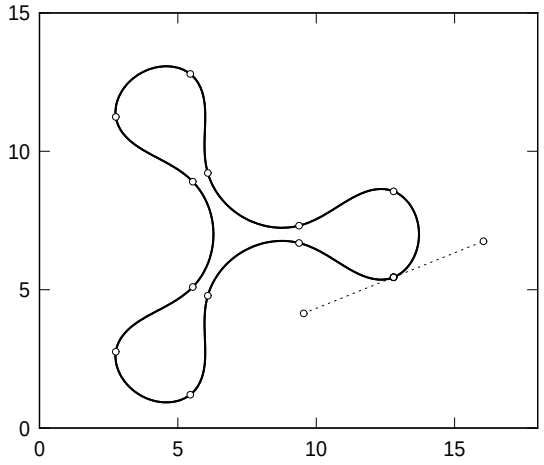

Figure 4: The `clover` test curve, defined by a planar C^2 PH quintic spline interpolating a set of 13 points with periodic end conditions (dimensions are in inches). The dashed lines indicate the "lead–in" and "lead–out" segments.

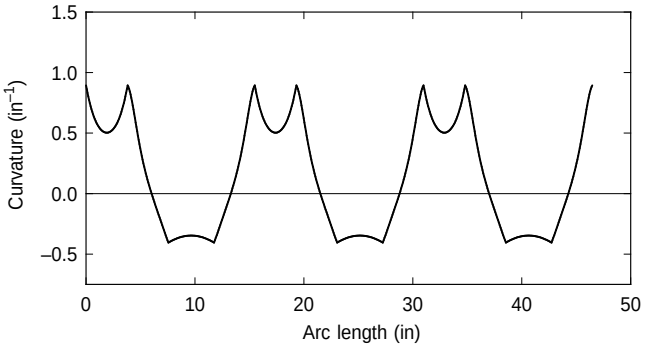

Figure 5: Variation of curvature with arc length for the `clover` test curve.

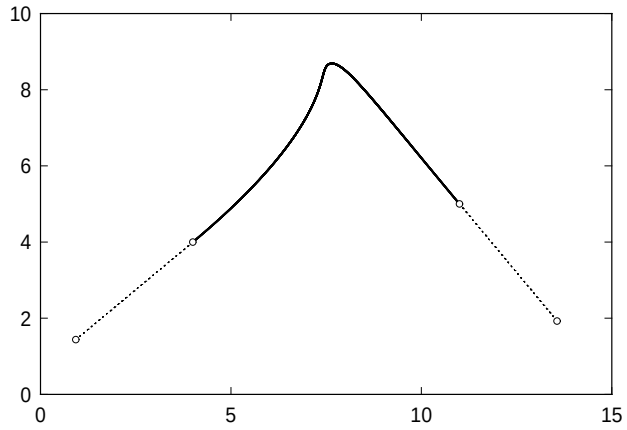

Figure 6: The `sharp-turn` test curve, defined by a single planar PH quintic segment interpolant $\mathbf{R}(\xi)$ for $\xi \in [\,0, 1\,]$ to the Hermite data $\mathbf{R}(0) = (4, 4)$, $\mathbf{R}'(0) = (30, 25)$ and $\mathbf{R}(1) = (11, 5)$, $\mathbf{R}'(1) = (25, -30)$. The dashed lines indicate the "lead–in" and "lead–out" segments (all dimensions are in inches).

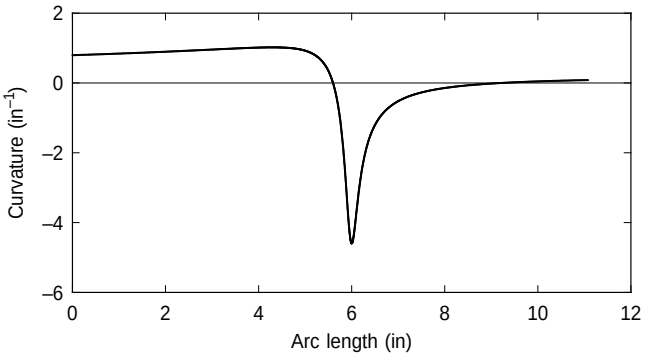

Figure 7: Variation of curvature with arc length for the `sharp-turn` curve.

frequency of the controller is $f = 1024$ Hz. OpenCNC provides the software interfaces needed to enable linking of custom real–time interpolators for the PH curves used as test cases, and other user–developed functions.

During each run, real–time position encoder data is stored in memory for subsequent analysis of the machine performance. From the position encoder data and the known exact path geometry, it is possible to compute the actual machine position error, feed error, and contour error in each sampling interval $\Delta t = 1/f \approx 0.001$ s. Axis velocities and accelerations may also be computed by first– and second–order differencing of the position data, and the feedrate along the path can be determined as the magnitude of the velocity vector $\mathbf{v}$ whose components are specified by the individual–axis velocities (v_x, v_y).

5 Experimental results

Experiments were performed on the `clover` and `sharp-turn` test curves at a sequence of feedrates increasing from 200 ipm to 800 ipm, with and without IDC. Data from these runs was used to assess the performance improvement obtained with IDC, in terms of total position error, feed error, contour error, and maintenance of the commanded feedrate.

5.1 Feedrate performance

Figure 8 compares the actual feedrate achieved for the `clover` test curve with IDC "on" and "off" for commanded feedrates of 200, 400, 600, and 800 ipm. Acceleration to and decleration from the commanded feedrate is performed on the linear lead–in/out segments. The IDC has no significant influence on maintenance of the commanded feedrate. This is as expected, since the axis accelerations incurred by this curve remain well below the 250 in/sec^2 limit at even the highest feedrates. Figure 9 shows the x and y axis acceleration and jerk (derivative of acceleration) for a feedrate of 600 ipm — clearly, the individual axis accelerations never exceed 100 in/sec^2.

Analogous comparisons of feedrate performance for the `sharp-turn` curve with IDC "on" and "off" are shown in Figure 10. As seen in Figure 11, on this curve the 250 in/sec^2 axis acceleration limit is strongly violated at the point of maximum curvature with a 600 ipm feedrate, so the IDC cannot be expected to perform satisfactorily at the higher feedrates. Figure 3 indicates that, for this curve, the highest feedrate consistent with the axis acceleration limit is

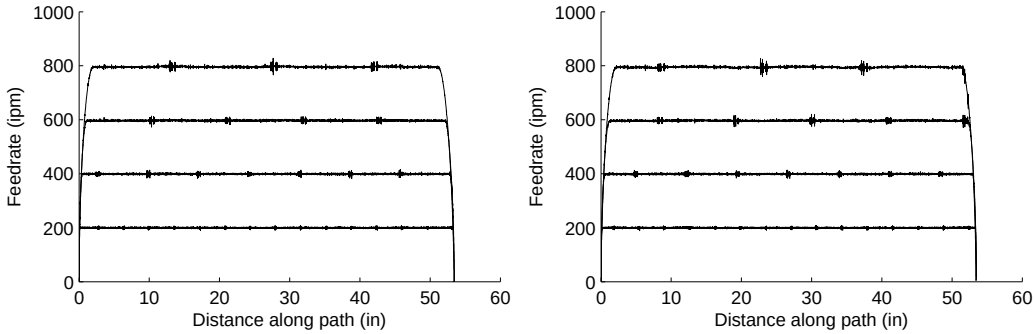

Figure 8: Measured feedrates for the `clover` curve at commanded feedrates of 200, 400, 600, and 800 ipm with the IDC switched on (left) and off (right).

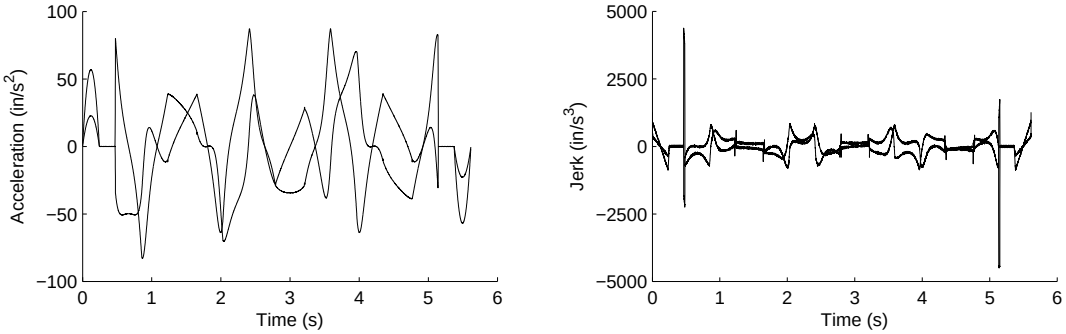

Figure 9: Variation of the x and y axis acceleration (left), and jerk (right) during execution of the `clover` test curve at a constant feedrate of 600 ipm.

a little over 400 ipm. Figure 10 shows that the 200 and 400 ipm feedrates are accurately maintained, regardless of whether or not IDC is active. At 600 and 800 ipm, however, significant feedrate fluctuations are incurred in negotiating the sharp turn in the curve. At 800 ipm, the fluctuations with IDC "on" are actually larger than with IDC "off" — this may be attributable to the fact that a given constant feedrate along the nominal path may induce a higher feedrate [5] along the modified path defined by (5).

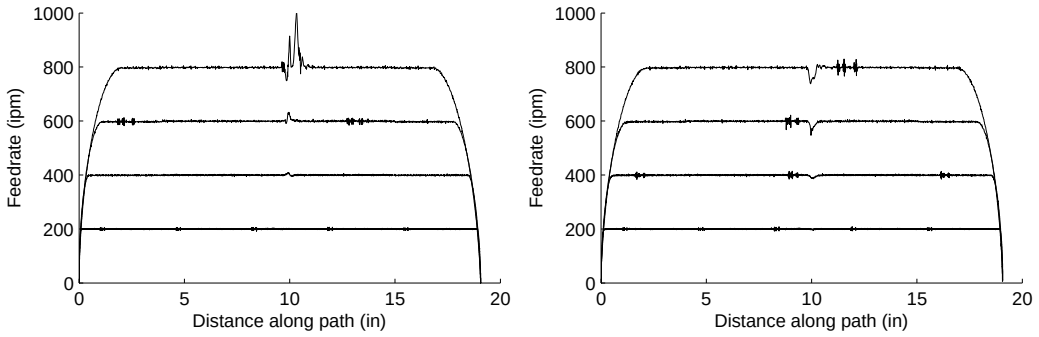

Figure 10: Measured feedrates for the `sharp-turn` curve with 200, 400, 600, 800 ipm commanded feedrates with the IDC switched on (left) and off (right).

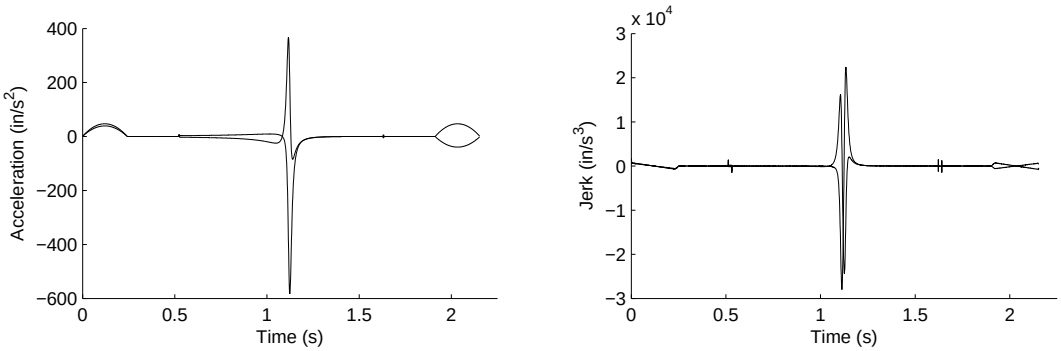

Figure 11: Variation of the x and y axis acceleration (left), and jerk (right) during execution of the `sharp-turn` curve at a constant feedrate of 600 ipm.

5.2 Contour and feed error

At each instant, the *total position error* is defined as the difference between the commanded machine position, computed by the real–time interpolator, and the actual machine position, measured by the axis encoders. The total position error may be resolved into components parallel and perpendicular to the local path tangent, known as the *feed error* and *contour error*. Whereas feed error corresponds to a timing discrepancy (lag or lead) along the path, contour error represents a geometrical deviation from the desired path, and is therefore much more serious in most motion control applications.

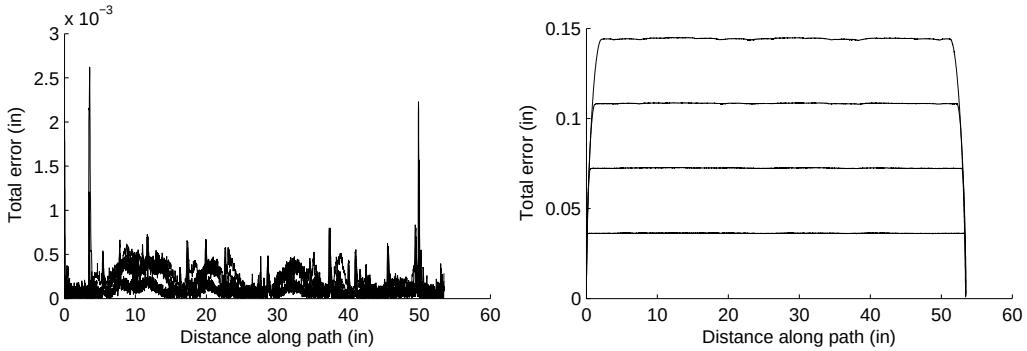

Figure 12: Measured total position error for the `clover` curve at feedrates of 200, 400, 600, and 800 ipm with the IDC switched on (left) and off (right).

Figure 12 compares the total position error with the IDC "on" and "off" during execution of the `clover` curve at feedrates of 200, 400, 600, 800 ipm (note the vastly different scales on these plots). With the IDC "on" spikes in the total position error are apparent where the lead–in/out segments connect to the `clover` curve. Since these junctures are only tangent–continuous, they entail a curvature discontinuity and corresponding acceleration discontinuity, i.e., an infinite jerk. Apart from these anomalies, with IDC "on" the position error remains well below 0.001 in at all feedrates up to 800 ipm. However, with IDC "off" there is a pronounced steady–state position error, increasing from ~ 0.035 in at 200 ipm to ~ 0.14 in at 800 ipm.

A comparison of the contour error, with and without IDC, is shown for the `clover` curve at feedrates of 200, 400, 600, 800 ipm in Figure 13. With IDC "on" the contour error remains well below 0.001 in at all feedrates (apart from the spikes cause by the acceleration discontinuities where the lead–in/lead–out segments connect to the `clover` curve). By comparison, with IDC "off"

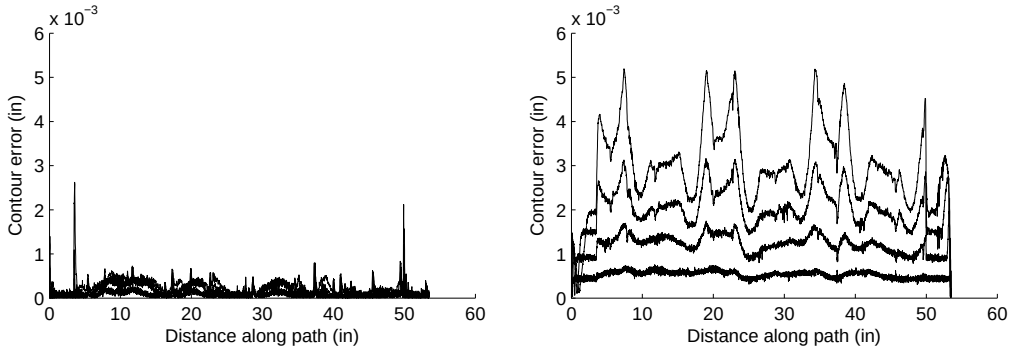

Figure 13: Measured contour error for the `clover` curve at feedrates of 200, 400, 600, and 800 ipm with the IDC switched on (left) and off (right).

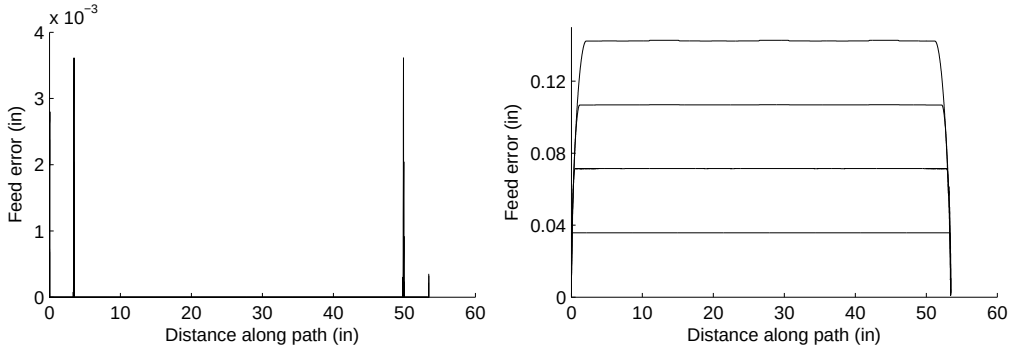

Figure 14: Measured feed error for the `clover` test curve at feedrates of 200, 400, 600, and 800 ipm with the IDC switched on (left) and off (right).

the contour errors are substantially greater at all feedrates, increasing to peak values of 0.005 in at 800 ipm. A similar comparison of the feed error is shown in Figure 14. With IDC "on" the feed error is (apart from the acceleration–discontinuity spikes and some initial/final transients) essentially below the resolution of the feed measurement at all feedrates. However, with IDC "off" substantial feed error is observed, that increases with feedrate from ~ 0.035 in at 200 ipm to ~ 0.14 in at 800 ipm.

A close inspection of Figures 12–14 reveals that the total position error is primarily contour error rather than feed error. The results indicate that with the `clover` curve (for which the axis acceleration bounds are not exceeded) the IDC scheme is extremely effective at (i) significantly subduing the contour error; and (ii) eliminating (to within measurement resolution) the feed error. The IDC scheme thus makes inclusion of an integrator (whose main function is to suppress steady–state error) in the controller redundant.

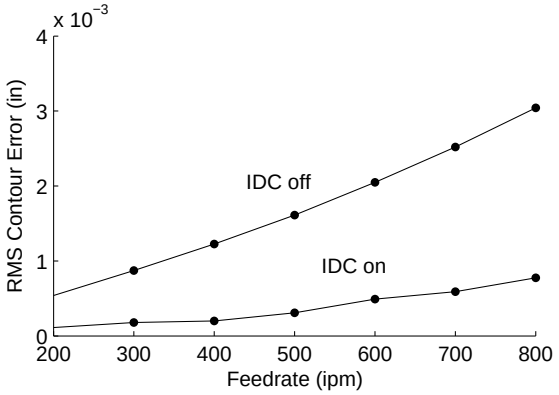

Figure 15: RMS contour error versus feedrate for the `clover` test curve.

With IDC "on" and "off" Figure 15 illustrates how the root–mean–square contour error varies with commanded feedrate along the `clover` test curve (omitting the lead–in/out segments). Clearly, use of the IDC scheme yields an RMS contour error that is smaller in both absolute value and growth rate. As an alternative means of visualizing the improvement in tracking accuracy achieved by the IDC scheme, Figure 16 shows the contour errors observed along the `clover` curve, using a $200\times$ magnification factor.

The `sharp-turn` test curve was designed so that, at the highest feedrates, the axis acceleration limit will be violated. Figure 17 compares total position error measurements for this curve with the IDC "on" and "off" at feedrates of

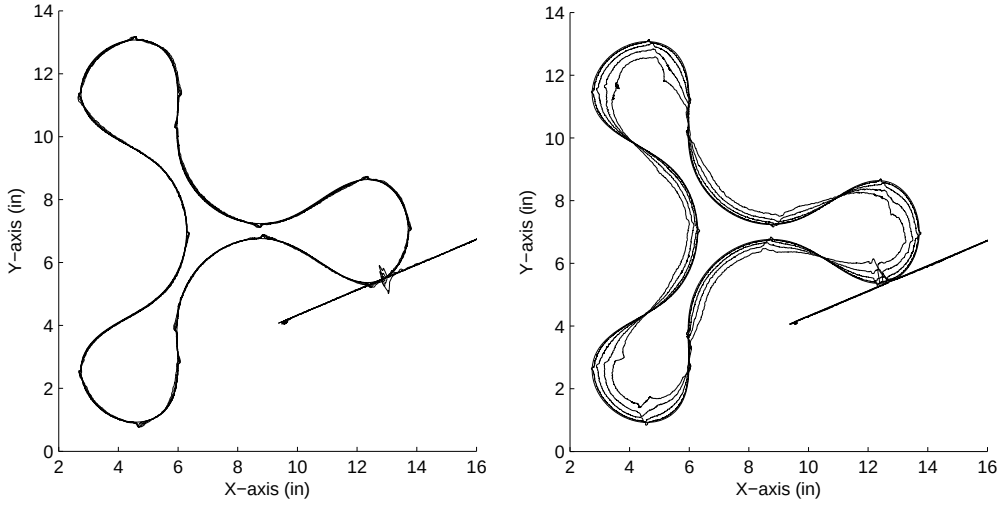

Figure 16: Contour error magnified 200× for the `clover` curve at feedrates of 200, 400, 600, and 800 ipm, with the IDC switched on (left) and off (right).

200, 400, 600, and 800 ipm. With IDC "on" the position error remains below 0.001 in at feedrates up to 400 ipm, but at 800 ipm it spikes to ~ 0.065 in at the point of extremum curvature. For the radius of curvature $r_{\min} = 0.2177$ in and feedrate 800 ipm, the y–axis acceleration at this point is ~ 817 in/sec^2 — greatly exceeding the 250 in/sec^2 limit (see Figure 3). By comparison, with IDC "off" substantial steady–state position errors arise at all feedrates, increasing from 0.035 in at 200 ipm to ~ 0.14 in at 800 ipm.

Figures 18 and 19 show the decomposition of the total position error into contour error and feed error components. At 200 and 400 ipm, the maximum contour error with IDC "on" is comparable to or less than that with IDC "off" but at 600 and 800 ipm it is worse, spiking to a peak of ~ 0.06 in at 800 ipm compared to ~ 0.015 in. This illustrates the importance of specifying feedrates, in coordination with the path curvature, that are consistent with the axis acceleration bounds in order to benefit from IDC. It is noteworthy in Figure 18 that the contour error along the weakly–curved portions of the path (incurring low acceleration) are much lower with IDC than without.

The feed error in Figure 19 with IDC "on" is seen to be negligible in all cases except at the point of extremum curvature with feedrate 800 ipm, for which it spikes to ~ 0.04 in. On the other hand, with IDC "off" substantial

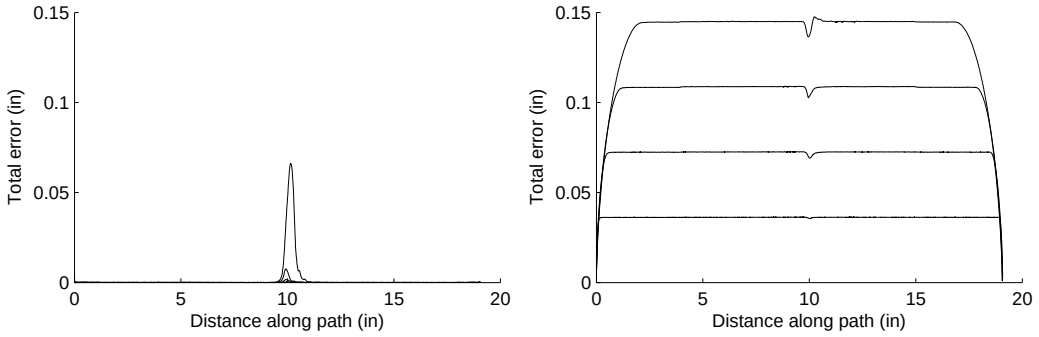

Figure 17: Measured total position errors for the `sharp-turn` curve at 200, 400, 600, 800 ipm feedrates with the IDC switched on (left) and off (right).

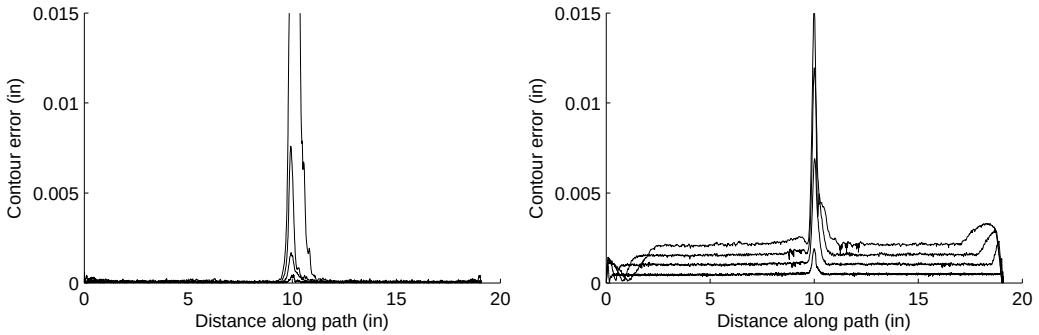

Figure 18: Measured contour error for the `sharp-turn` test curve at 200, 400, 600, 800 ipm feedrates with the IDC switched on (left) and off (right).

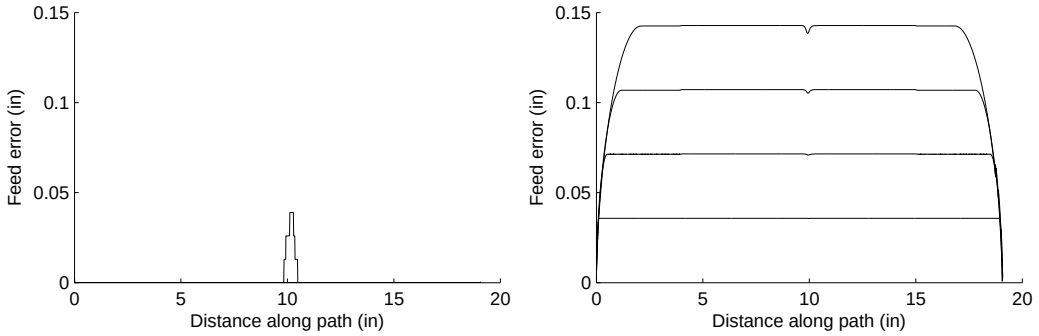

Figure 19: Measured feed error for the `sharp-turn` test curve at 200, 400, 600, 800 ipm feedrates with the IDC switched on (left) and off (right).

steady–state position errors arise at all feedrates, and at all points along the path — not just near the point of maximum curvature.

The results from the `clover` and `sharp-turn` test curves highlight the importance of determining the axis acceleration limits as part of the system identification procedure, and observing these limits when the IDC scheme is implemented on the machine. Subject to these constraints, the IDC scheme can dramatically improve the machine tracking accuracy.

5.3 P gain reduction

A primary function of the controller in CNC machines is to compensate for "disturbances" — primarily variations in the cutting force. Depending upon the specified feedrate, path geometry, tool type, material, depth of cut, etc., cutting forces may be less than, comparable to, or larger than the dynamic forces associated with acceleration/deceleration of the machine axes.

In contexts where dynamic forces are dominant, the IDC scheme should — given a sufficiently accurate machine characterization — provide accurate execution of commanded motions without the need for feedback control using axis position encoders. Although not advisable in practice, this can be used as a paradigm for testing the accuracy of the adopted machine model.

The nominal gain value $k_p = 20{,}000$ is chosen to ensure accurate tracking with a reasonable stability safety margin. Runs using the `clover` test curve were performed at 600 ipm with $k_p = 20{,}000$ and with k_p successively reduced by a factor of two, to $k_p = 10{,}000$, 5,000, and 2,500.

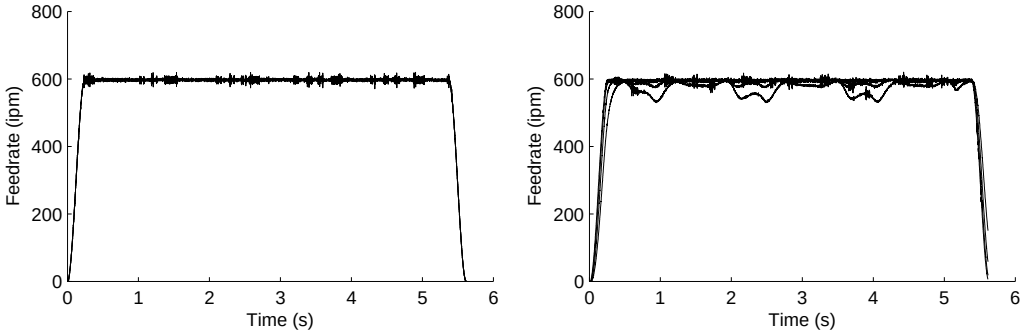

Figure 20: Measured feedrate for the `clover` curve using gains $k_p = 20{,}000$, 10,000, 5,000, 2,500 with the IDC "on" (left) and "off" (right) at 600 ipm.

Figure 20 shows the measured feedrate obtained with the reduced gain values. With IDC "on" accurate maintenance of the 600 ipm commanded feedrate is achieved even at $k_p = 2{,}500$ (an 8–fold reduction over the nominal value). Reasonably accurate feedrate maintenance is also observed with IDC "off" although substantial fluctuations are apparent at $k_p = 2{,}500$.

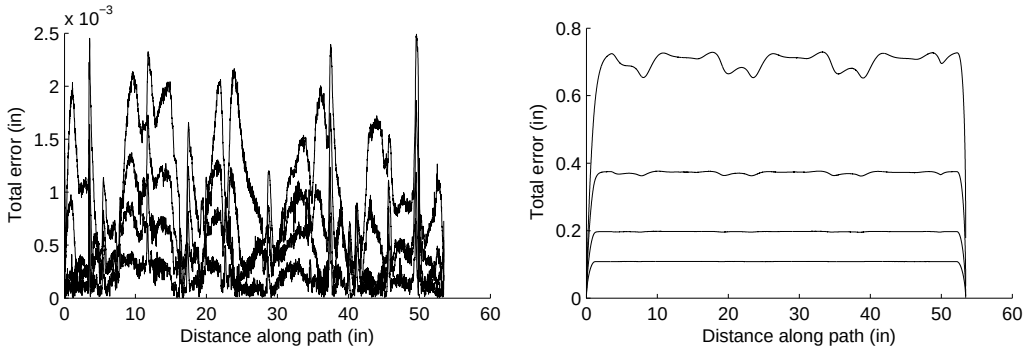

Figure 21: Measured position error for the `clover` curve using $k_p = 20{,}000$, 10,000, 5,000, 2,500 with the IDC "on" (left) and "off" (right) at 600 ipm.

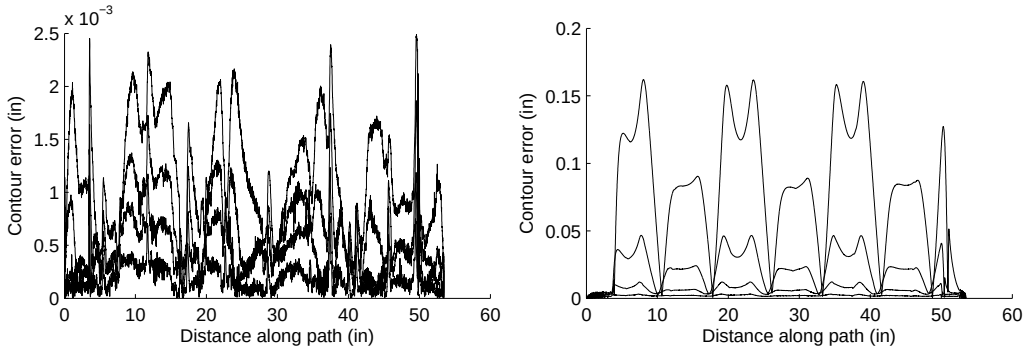

Figure 22: Measured contour error for the `clover` curve using $k_p = 20{,}000$, 10,000, 5,000, 2,500 with the IDC "on" (left) and "off" (right) at 600 ipm.

The measured total position error is shown in Figure 21 (note the vastly different scales for the "on" and "off" plots). As expected, with IDC "off" the steady–state position error grows steadily with decreasing gain, from ~ 0.1 in at $k_p = 20{,}000$ to ~ 0.7 in at $k_p = 2{,}500$. By comparision, with IDC "on" the maximum total position error is no more than ~ 0.0005 in at $k_p = 20{,}000$

and ~ 0.0025 in at $k_p = 2{,}500$. The decomposition of the total position error into contour error and feed error is shown in Figures 22 and 23.

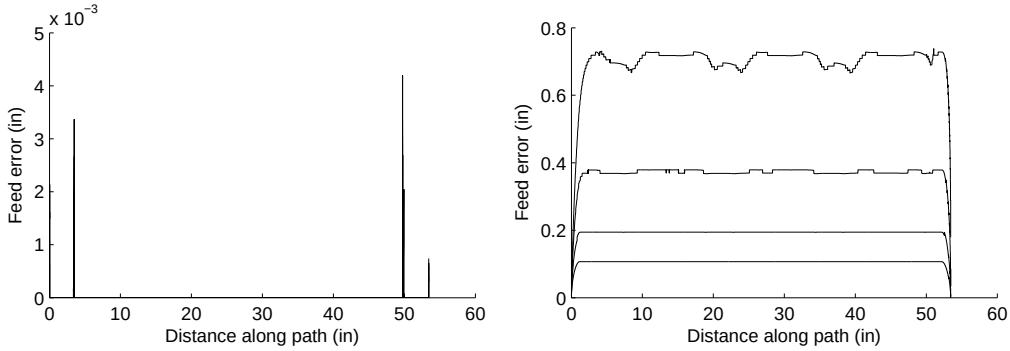

Figure 23: Measured feed error for the `clover` curve with gains $k_p = 20{,}000$, 10,000, 5,000, 2,500 with the IDC "on" (left) and "off" (right) at 600 ipm.

At each gain value, the contour error with IDC "on" is smaller than that with IDC "off" by a factor of ~ 100. Furthermore, the IDC scheme essentially eliminates the feed error, within the measurement resolution (except where the lead–in/out segments meet the `clover` curve) — even with $k_p = 2{,}500$. However, with IDC "off" substantial feed errors are observed, increasing systematically as k_p is successively reduced. Careful inspection of Figures 21–23 shows that with IDC "on" the total position error consists of a small contour error and hardly any feed error at all. With IDC "off" it consists of both substantial contour error and substantial feed error.

These results demonstrate that, in motion control applications dominated by the machine dynamics with no significant "disturbance" forces, the IDC scheme allows accurate execution of commanded curvilinear motions using only a very weak and simple position feedback control strategy.

6 Conclusion

The performance of a simple inverse dynamics compensation (IDC) scheme has been tested on a 3–axis CNC milling machine with an open–architecture software controller. The IDC scheme can be implemented on any machine with a real–time interpolator that accommodates analytic path descriptions, with well–defined first and second derivatives — including controllers that

perform an on–the–fly smoothing of G code part programs. In addition, the values of certain dimensionless machine parameters must be determined by analyzing input/output data from test runs, using a system identification software tool (one must also ensure that the part program will not incur accelerations that exceed the physical limits of the axis drive systems).

The simulations described in [5], being dependent on a relatively simple model of the machine axis dynamics, do not provide convincing evidence of the error suppression performance of the IDC scheme on a real CNC machine. The experiments described herein indicate that, with appropriately measured model parameters, the IDC scheme is indeed remarkably effective in ensuring the faithful execution of commanded motions. This is apparent in several key performance measures: maintenance of feedrate under strong path curvature, suppression of feed error (lag/lead along the path), contour error (geometric deviation from the path), total position error, and motion smoothness. For motion control applications that do not incur significant "disturbances" (e.g., cutting forces), reductions in the controller P gain show that the IDC scheme even continues to yield satisfactory performance near the open–loop limit.

Of course, the IDC scheme cannot compensate for cutting forces of large magnitude or strong variation in CNC machining. However, if these forces can be described by simple analytic models, it may be possible to compensate for them by methods analogous to those proposed herein for dynamic forces. Such an approach could prove simpler, and much cheaper to implement, than compensation methods based on real–time sensing of cutting forces.

Acknowledgement

The authors are grateful to Professor R. A. Hess for use of his `MATLAB` system identification software tool.

References

[1] Y. Altintas (2000), *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*, Cambridge University Press.

[2] A. F. de Souza and R. T. Coelho (2007), Experimental investigation of feed rate limitations on high speed milling aimed at industrial

applications, *International Journal of Advanced Manufacturing Technology* **32**, 1104–1114.

[3] K. Erkorkmaz and W. Wong (2007), Rapid identification technique for virtual CNC drives, *International Journal of Machine Tools and Manufacture* **47**, 1381–1392.

[4] K. Erkorkmaz, C–H. Yeung, and Y. Altintas (2006), Virtual CNC system. Part II. High speed contouring application, *International Journal of Machine Tools and Manufacture* **46**, 1124–1138.

[5] C. A. Ernesto and R. T. Farouki (2010), Solution of inverse dynamics problems for contour error minimization in CNC machines, *International Journal of Advanced Manufacturing Technology* **49**, 589–604.

[6] R. T. Farouki (2008), *Pythagorean–Hodograph Curves: Algebra and Geometry Inseparable*, Springer, Berlin.

[7] R. T. Farouki, B. K. Kuspa, C. Manni, and A. Sestini (2001), Efficient solution of the complex quadratic tridiagonal system for C^2 PH quintic splines, *Numerical Algorithms* **27**, 35–60.

[8] R. T. Farouki, J. Manjunathaiah, D. Nicholas, G.–F. Yuan, and S. Jee (1998), Variable feedrate CNC interpolators for constant material removal rates along Pythagorean–hodograph curves, *Computer Aided Design* **30**, 631–640.

[9] R. T. Farouki, J. Manjunathaiah, and G.–F. Yuan (1999), G codes for the specification of Pythagorean–hodograph tool paths and associated feedrate functions on open–architecture CNC machines, *International Journal of Machine Tools and Manufacture* **39**, 123–142.

[10] R. T. Farouki and C. A. Neff (1995), Hermite interpolation by Pythagorean–hodograph quintics, *Mathematics of Computation* **64**, 1589–1609.

[11] R. T. Farouki and S. Shah (1996), Real–time CNC interpolators for Pythagorean–hodograph curves, *Computer Aided Geometric Design* **13**, 583–600.

[12] R. T. Farouki and Y–F. Tsai (2001), Exact Taylor series coefficients for variable–feedrate CNC curve interpolators, *Computer Aided Design* **33**, 155–165.

[13] J.–T. Huang and D. C. H. Yang (1992), A generalized interpolator for command generation of parametric curves in computer–controlled machines, Proceedings of the *Japan/USA Symposium on Flexible Automation*, Vol. 1, ASME, 393–399.

[14] R. Komanduri, K. Subramanian, and B. F. von Turkovich (eds.) (1984), *High Speed Machining*, PED–Vol. 12, ASME, New York.

[15] R–S. Lin and Y. Koren (1996), Real–time interpolators for multi–axis CNC machine tools, *Manufacturing Systems* **25**, 145–149.

[16] S. Smith and J. Tlusty (1997), Current trends in high–speed machining, *ASME Journal of Manufacturing Science and Engineering* **119**, 664–666.

[17] J. Tlusty (1993), High–speed machining, *CIRP Annals* **42**, 733–738.

[18] D. C. H. Yang and T. Kong (1994), Parametric interpolator versus linear interpolator for precision CNC machining, *Computer Aided Design* **26**, 225–234.

[19] S–S. Yeh and P–L. Hsu (1999), The speed–controlled interpolator for machining parametric curves, *Computer Aided Design* **31**, 349–357.

[20] C–H. Yeung, Y. Altintas, and K. Erkorkmaz (2006), Virtual CNC system. Part I. System architecture, *International Journal of Machine Tools and Manufacture* **46**, 1107–1123.