Title
Inverse dynamics toolpath compensation for CNC machines based on model predictive control

Permalink
https://escholarship.org/uc/item/08j1s3xh

Journal
The International Journal of Advanced Manufacturing Technology, 109(7-8)

ISSN
0268-3768

Authors
Margolis, Benjamin WL
Farouki, Rida T

Publication Date
2020-08-01

DOI
10.1007/s00170-020-05719-7

Peer reviewed

Inverse dynamics toolpath compensation for CNC machines based on model predictive control

Benjamin W. L. Margolis and Rida T. Farouki
Department of Mechanical and Aerospace Engineering,
University of California, Davis, CA 95616, USA.

July 14, 2020

Abstract

The use of model predictive control (MPC) as a form of inverse dynamics compensation for multi–axis CNC machines, to subdue the inaccuracies incurred by axis inertia and damping, is investigated by both simulation studies and experimental performance analysis using a 3–axis milling machine governed by an open–architecture software controller. The results indicate that MPC is a viable tool for inverse dynamics compensation with a controller sampling frequency $f = 1024$ Hz running on a 500 MHz processor, with only modest prediction horizons offering excellent performance in terms of feedrate accuracy and contour error suppression. Unlike inverse dynamics schemes based upon linear time–invariant dynamic models, the MPC scheme provides the flexibility to compensate for non–linear physical effects such as backlash in the machine axes and hard constraints on axis accelerations imposed by motor torque constraint.

keywords: model predictive control; inverse dynamics; CNC machine;
system identification; feedrate; contour error; feed error.

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

1 Introduction

Manufacturing processes such as computer numerical control (CNC) machining and 3D printing are critically dependent on precision multi–axis motion control to fabricate engineering parts that satisfy stringent tolerances and complex geometrical or functional requirements. Feedback controllers, in which high–frequency comparison of commanded and measured machine positions is employed to modulate actuation of the axis drives, are essential to ensure accurate traversal of complex paths at high speeds.

Classical PID controllers are commonly employed in motion control systems, with gains optimized to improve accuracy and responsivity while maintaining system stability. However, this optimization may not suffice for applications requiring high–speed execution of complex curvilinear paths, in which the smoothing influence of inertia and damping of the machine axes can significantly degrade the tracking accuracy.

The adverse effect of inherent machine dynamics can be addressed through *inverse dynamics compensation* (IDC) schemes. Such schemes employ *system identification tools*, which determine basic machine physical parameters (e.g., the inertia and damping of each axis) based upon the response to test inputs. A machine dynamic model can then be solved in an "inverse" manner — i.e., to determine the required input to achieve a desired output, subject to the influence of inertia and damping. The result is a compensatory term, to be added to the nominal commanded machine position, in each sampling interval of the controller. IDC schemes cannot compensate for "disturbance" effects that are unpredictable, or for which there is no sensory information. By compensating for known physical machine limitations,[1] however, IDC relieves the burden on the feedback controller, and allows it to focus on minimizing the influence of disturbances.

[1] In high–speed machining [5, 11, 18, 20] axis inertia and damping may be the dominant limitations on tracking accuracy.

An IDC scheme for curvilinear paths was proposed in [9], and it was shown that a closed–form expression for the compensation term is possible in the case of a P controller. Experimental tests [17] have indicated that significant improvement in tracking accuracy is possible, even when the P gain is successively reduced toward the open–loop limit. A further extension of this approach, to accommodate PI and PID controllers, was subsequently developed in [27]: in these cases, the correction term includes a single irreducible integral, which can nevertheless be evaluated in real time by means of an efficient adaptive quadrature scheme.

These previous IDC schemes were all based on second–order linear time invariant (LTI) models of the machine dynamics, which cannot satisfactorily model important effects that influence tracking accuracy (such as backlash of the machine axes, incurred at reversals of the sense of motion of each axis). In the present study, we investigate the use of model predictive control (MPC) as a form of inverse dynamics compensation. MPC is a powerful and versatile scheme for controlling complex dynamical systems by solving an optimal control problem over a finite future time horizon that recedes with each time step [16]. Since MPC solves an optimal control problem at every time step, there is flexibility in modeling the objective function, system dynamics, and trajectory constraints. However this requires the availability of appropriate numerical algorithms to solve the modeled optimal control problem in real time.

Previous studies on the use of MPC in real–time motion control focus primarily on developing a model to minimize predicted contour error [6, 12, 19, 23, 24, 25]. However, due to the coupling and nonlinearity of the objective function, this approach creates challenges for real–time application. Additionally, most of these studies are focused on using MPC for motor control, based on an optimization approach at each time step to determine the voltage or current for each motor. Yang et al. [24] do compute a trajectory compensation, but it is pre–computed offline. The study [25] also pre–computes the compensation using MPC, presumably because the algorithm used to solve the optimization problem cannot operate in real time. Uchiyama et al. [6] develop a linear system of equations to minimize the objective analytically, that can be solved in real time but cannot accommodate constraints. Some approaches to formulating the problem as a quadratic program, solved by the Hildreth method [21], are presented in [12, 19, 23] with promising results, but the experimental results depend on sufficiently long sampling intervals or relatively powerful computational resources.

By comparison, this study computes a trajectory compensation for each machine axis independently using high performance interior point methods. This approach relies upon the resulting precision in each axis to achieve the goal of a small contour error, rather than explicit real–time monitoring of the contour error [4]. Treating each axis independently trades the need for a single, computationally more burdensome optimization for multiple optimizations that can be performed more efficiently. Since computation times for optimization problems nominally scale with the cube of the problem size, this trade–off should reduce computation time while providing sufficiently accurate performance.

The plan for the remainder of this paper follows. In Section 2 a model is presented for the system dynamics of each machine axis, and the IDC scheme based upon MPC is described in Section 3. The performance of this IDC scheme is assessed in Section 4 using simulation studies, and Section 5 presents the results of an experimental performance analysis using a 3–axis CNC milling machine governed by an open–architecture software controller. Section 6 describes results from augmenting the basic MPC compensation scheme with axis acceleration limits. Finally, Section 7 summarizes the key findings of the present study, and identifies aspects of the scheme that deserve further investigation.

2 Machine Dynamics

The axes of multi–axis CNC machines are usually driven by independent servo systems. The individual axis drives can be modelled as linear dynamic systems governed by PID controllers [1, 8, 26]. We focus here on the case of a P controller, for which the actual position axis position p (measured by the position encoder) is determined from the commanded position c through a differential equation of the form

$$a_2\,\ddot{p} \,+\, a_1\,\dot{p} \,+\, p \,=\, c\,, \tag{1}$$

with $a_2 = J/Kk_p$, $a_1 = B/Kk_p$ where J and B denote the axis inertia and damping, k_p is the proportional gain, and K represents the overall current amplifier, motor torque, and transmission ratio gains [9]. The input to the controller is the axis position error $e = c - p$. Correspondingly, the Laplace forms $c(s)$ and $p(s)$

of the commanded and actual axis positions are related [9] by the transfer function

$$\frac{p(s)}{c(s)} \;=\; \frac{Kk_p}{Js^2+Bs+Kk_p}\,, \tag{2}$$

To implement the MPC scheme, the coefficients a_2, a_1 (which are assumed to be the same for both the x and y axes) must be determined. System identification procedures for CNC axis drive systems are typically based [7] on a series of time or frequency domain response tests. In the present context, a number of curved paths were executed at various feedrates, with the real–time reference point and position encoder data being recorded for each axis. The data were input to a `Matlab` program, to determine a best–fit transfer function of the form (2), from which the a_2 and a_1 values in (1) may be identified. Further details on the system identification scheme may be found in [17].

Several additional transformations must be performed on this standard CNC axis model to find a discrete–time state–space representation suitable for MPC. First, the transfer function (2) must be converted to a state–space model of the form

$$\dot{x} \;=\; Fx + Gu\,,$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ is the controlled input, and F and G are the appropriately sized state and and input matrices. Since this state–space model will be used internally in the MPC formulation, the system output is omitted from the discussion for brevity. To develop the state–space model, the axis linear velocity, $v = \dot{p}$, is identified to define the state $x = [p \quad v]^T$ and input $u = [c]$, which gives the state–space model in control canonical form,

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{a_2} & -\frac{a_1}{a_2} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{a_2} \end{bmatrix} u.$$

This continuous–time state space model can be transformed to a discrete–time state space model, representing the axis dynamics sampled on the digital computer, of the form

$$x_{k+1} \;=\; \Phi x_k + \Gamma u_k\,.$$

The discrete–time state and input matrices are determined using the zero–order hold transformation

$$\Phi \;=\; e^{\Delta t\, F}\,, \qquad \Gamma \;=\; \int_0^{\Delta t} e^{\tau\, F}\, d\tau\; G$$

where Δt is the sampling rate of the continuous–time system and $e^{(\cdot)}$ represents the matrix exponential of $(\cdot)$. Finally, to enable control of the input slew rate, the discrete–time state is augmented to $\tilde{x} = [p \quad v \quad c]^T$ and the controlled input becomes $\tilde{u} = [\Delta c]$ where $\Delta c = c_{k+1} - c_k$ is the control increment. Then the final state space model has the form

$$\tilde{x}_{k+1} \;=\; \begin{bmatrix} \Phi & \Gamma \end{bmatrix} \tilde{x}_k + \begin{bmatrix} \Gamma \\ 1 \end{bmatrix} \tilde{u}_k.$$

For convenience, we will use A and B to refer to the state and input matrices of this augmented model and omit the tilde on the augmented state and input variables in the sequel.

3 Model Predictive Control

The basic idea of Model Predictive Control is to apply the first portion of the solution to an optimal control problem over the prediction horizon at every time step. One approach is to formulate the optimal control problem as a quadratic program (QP) because of the availability of reliable, efficient, and accurate solvers. As discussed in [22], several strategies can be implemented to reduce computational burden so that the QP can be solved fast enough to be used in real–time applications. We can express the QP at the kth time step as

$$\begin{array}{ll} \underset{\{x_{k+i}\},\{u_{k+i}\}}{\text{minimize}} & \displaystyle\sum_{i=1}^{H} (x_{k+i} - r_{k+i})^T\, Q\, (x_{k+i} - r_{k+i}) + u_{k+i}^T\, R\, u_{k+i} \\ \text{subject to} & x_{k+i+1} = A\, x_{k+i} + B\, u_{k+i}, \qquad i = 0, \ldots, H-1 \\ & Cx_{k+i} \le d, \qquad i = 0, \ldots, H \end{array} \tag{3}$$

where r_i, x_i, and u_i are the reference state, predicted state, and predicted input at the ith time step, H is the prediction horizon, and Q and R are weighting matrices for ℓ_2 norms in the objective function. C and d are the inequality matrix and vector defining inequality constraints that can be used to enforce safety or performance limits. In this pilot study, we construct the inequality matrix and vector to ensure that $|c_{k+i} - x_{k+1}| \leq \tau$ for $i \in \{0, \ldots, H\}$, where $\tau = 0.05\,\text{in}$ is a safety limit on the machine test–bed. The compensation value at the kth time step is $c^*_{k+1} - r_{k+1}$, where the $*$ superscript indicates the value of the variable that solves the minimization problem.

This QP formulation uses both the state and input as optimization variables over the prediction horizon and enforces the system model using affine equality constraints. Although this requires more variables to solve, it introduces a sparsity structure in the linear systems equations to be solved during optimization that ultimately reduces the number of floating point computations involved. This QP formulation is also analogous to a collocation method of solving differential equations. Using weights matrices $Q = \text{diag}\left(\begin{bmatrix}1 & 0 & 0\end{bmatrix}\right)$ and $R = [0]$, placing the entire weight of the objective on the predicted positions, this QP can be interpreted as "find the trajectory that most closely matches the reference positions over the prediction horizon while satistfying the system dynamics and safety constraints." Since the reference positions were generated by sampling a continuous path with continuous velocities and accelerations, tracking the reference positions closely will also track the desired velocities at each time step.

We can determine the compensation in open–loop and at each time step the values for x_k and u_k come from the solution found at the previous time step, with initial values coincident with the reference path.

4 Simulation Study

In this section the results of a simulation study to determine the best horizon length are shown. Simulations of the system dynamics with an MPC controller were performed with the `SimuPy` simulation framework [14]. For the simulation study, at each time step the quadratic program (3) is solved using `cvxopt`, a Python library for convex optimization [2]. The controller is modeled as a discrete–time system with an update rate of 1024 Hz. For this study, the asymptotic values for the axis parameters from [17] were used, namely $a_2 = 2.828 \times 10^{-5}$ and $a_1 = 1.089 \times 10^{-2}$. The simulation and experimental results presented below are expressed in units commonly employed in the US manufacturing industry: inches (in) for path length; inches per minute (ipm) for feedrate, and in/s^2 for acceleration. The equivalent metric values may be obtained by noting that 1 in = 25.4 mm, 100 ipm = 42.33 mm/s, and 1 in/s^2 = 25.4 mm/s^2.

For simplicity, we employ only paths defined by planar parametric test curves $\mathbf{r}(\xi) = (x(\xi), y(\xi))$. Since the MPC scheme involves independent control of each machine axis, the extension to spatial paths is straightforward once the dynamic coefficients for the third axis are determined by the system identification tool. If $\mathbf{c} = (x(\xi_c), y(\xi_c))$ is the commanded position along $\mathbf{r}(\xi)$, generated at each time step for a specified feedrate by the real–time interpolator, and $\mathbf{p} = (x_p, y_p)$ is the actual machine position (as measured by the axis position encoders) at each instant, the position error $\mathbf{e} = \mathbf{c} - \mathbf{p}$ may be resolved into components parallel and perpendicular to the tangent of $\mathbf{r}(\xi)$ at the point $\mathbf{c}$. These components are called the instantaneous *feed error* (i.e., lag/lead along the path) and *contour error* (geometrical deviation from the commanded path).

In Figures 1 and 2, the simulated response for the `sharp turn` test curve is shown for various feedrates without compensation is compared to the computed compensation to highlight their inverse relationship. Note that the simulated response to the compensated test curve is omitted because it would be indistinguishable from the reference path at the resolution of the figures. This curve exhibits a pronounced curvature extremum that serves to emphasize the limitations of the machine intrinsic dynamics. To highlight the contour error and compensation term, a "zoomed in" view is also shown. In these figures, only a nominal MPC compensation is shown because the difference from MPC parameters is difficult to distinguish visually.

Figures 3–6 illustrate the feedrate error, position error, contour error, and feed error for the simulated response, both without compensation and with MPC–based compensation using various prediction horizons. Throughout this discussion, the current time step is included in the horizon, so a value $H = 2$ corresponds to the minimal MPC horizon, determining the optimal compensation based on a 1–step look ahead. In Figures 7–9, we show the root–mean–square (rms) values for the position, contour, and feed errors plotted against the horizon length. The uncompensated cases are excluded from the rms plots because the errors are many orders of magnitude worse than in the compensated case.

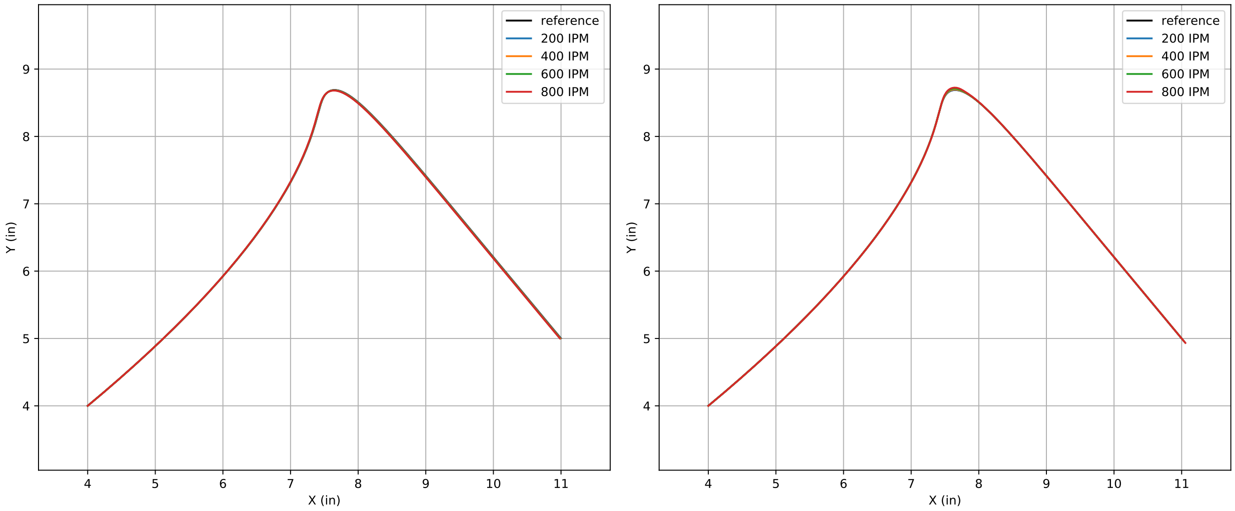

Figure 1: Simulated machine response without compensation (left) and simulated MPC compensation (right) at various feedrates.

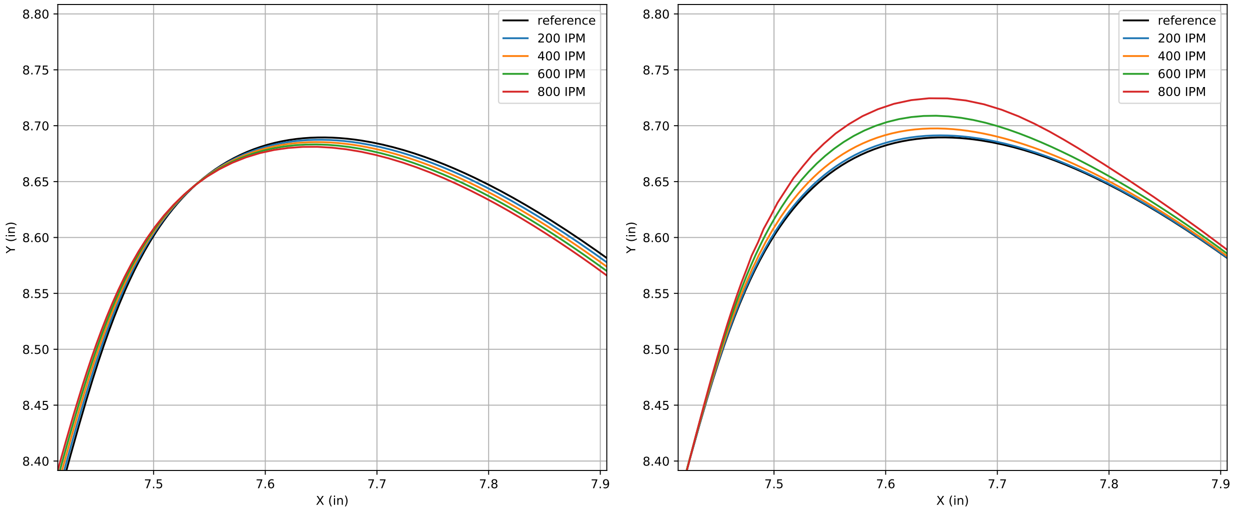

Figure 2: Close-up view of simulated machine response without compensation (left) and simulated MPC compensation (right) at various feedrates.

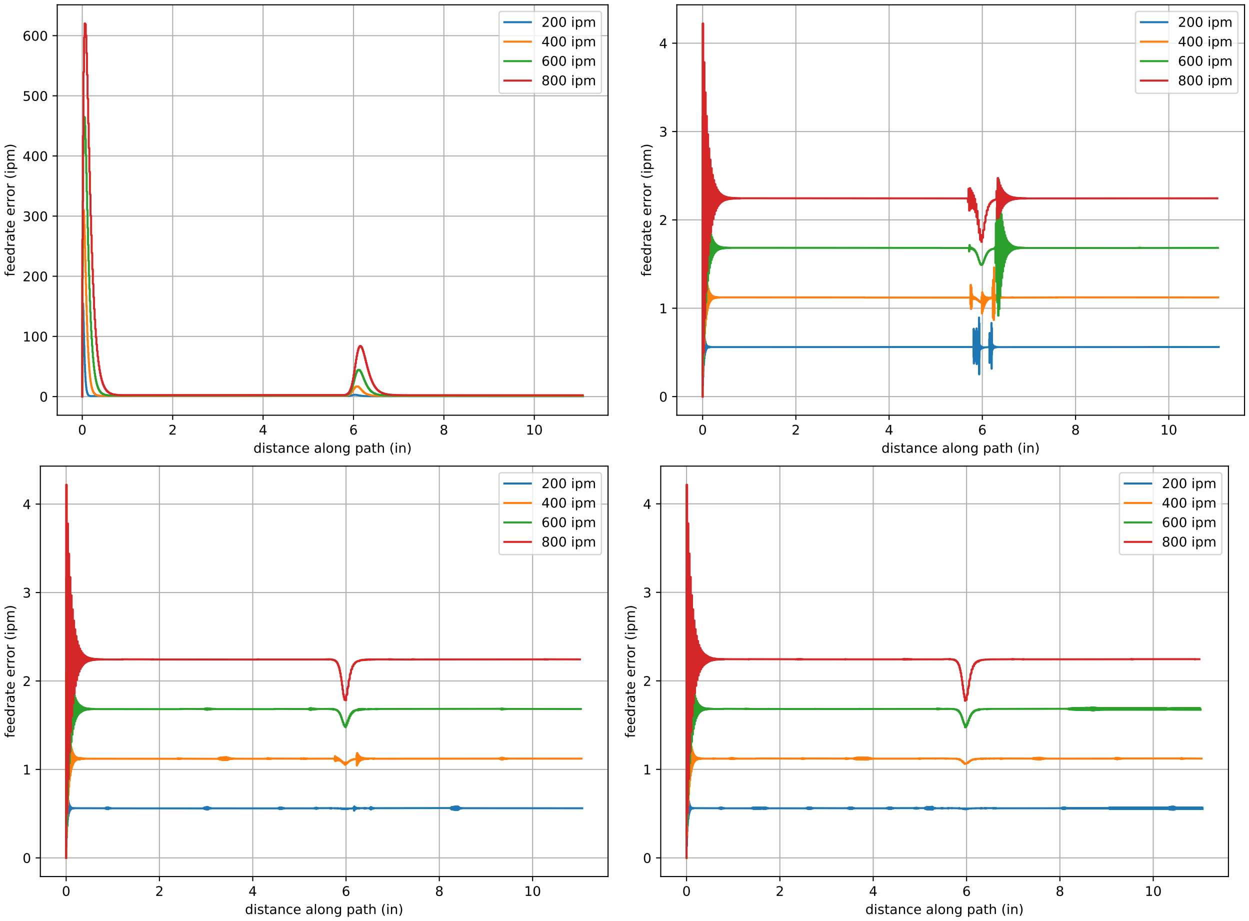

Figure 3: Simulated feedrate error for various prediction horizons and feedrates. Note the orders of magnitude difference in y–axis scale.

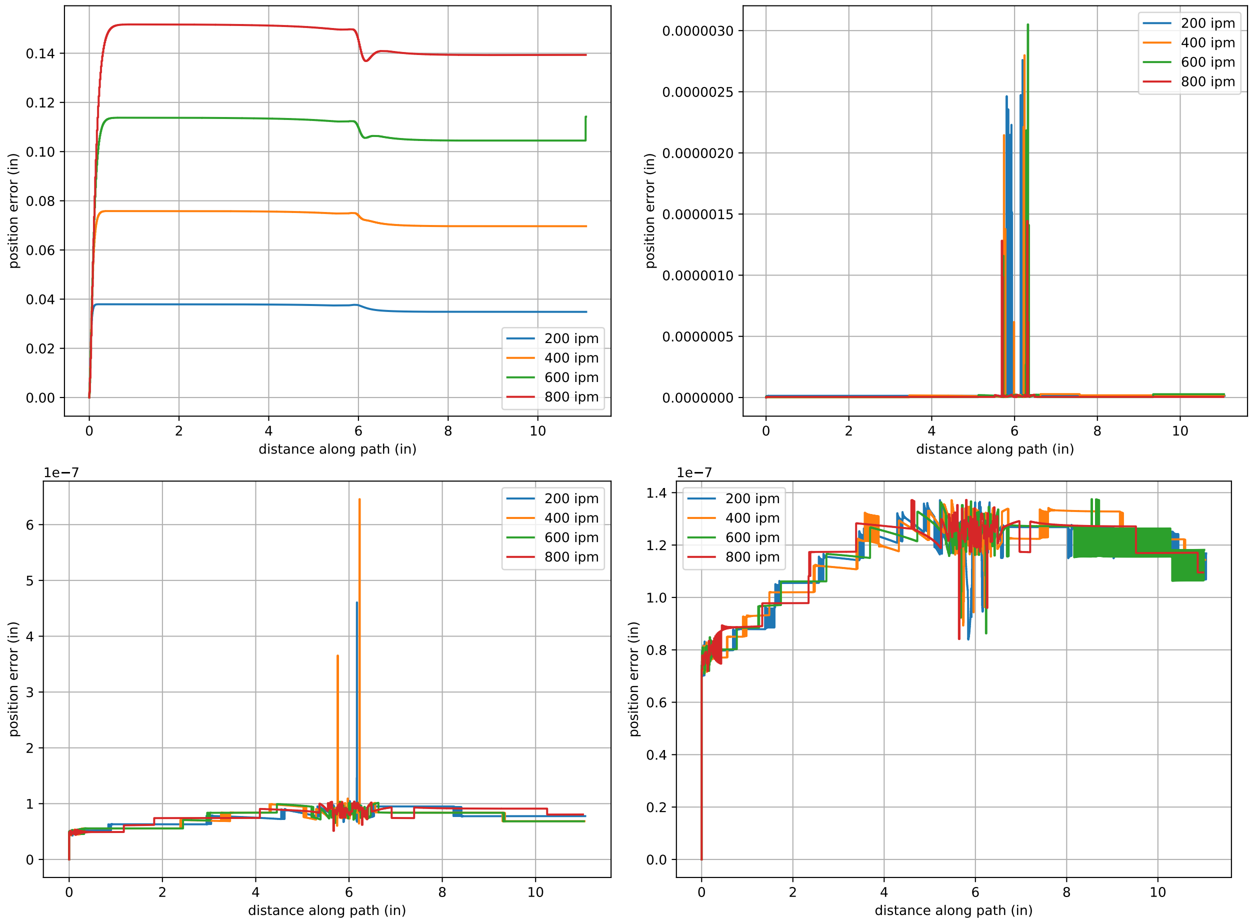

Figure 4: Simulated position error for various prediction horizons and feedrates.

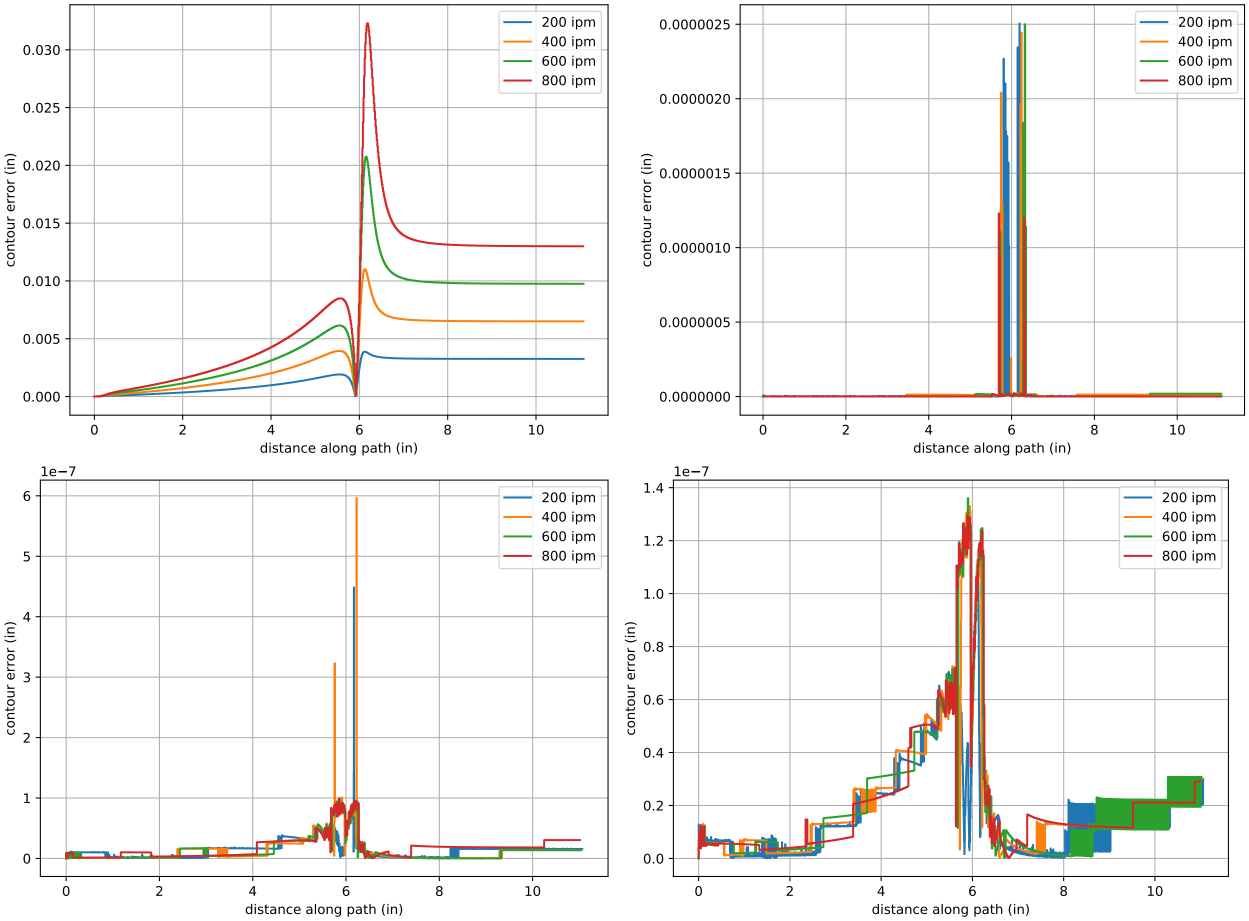

Figure 5: Simulated contour error for various prediction horizons and feedrates.

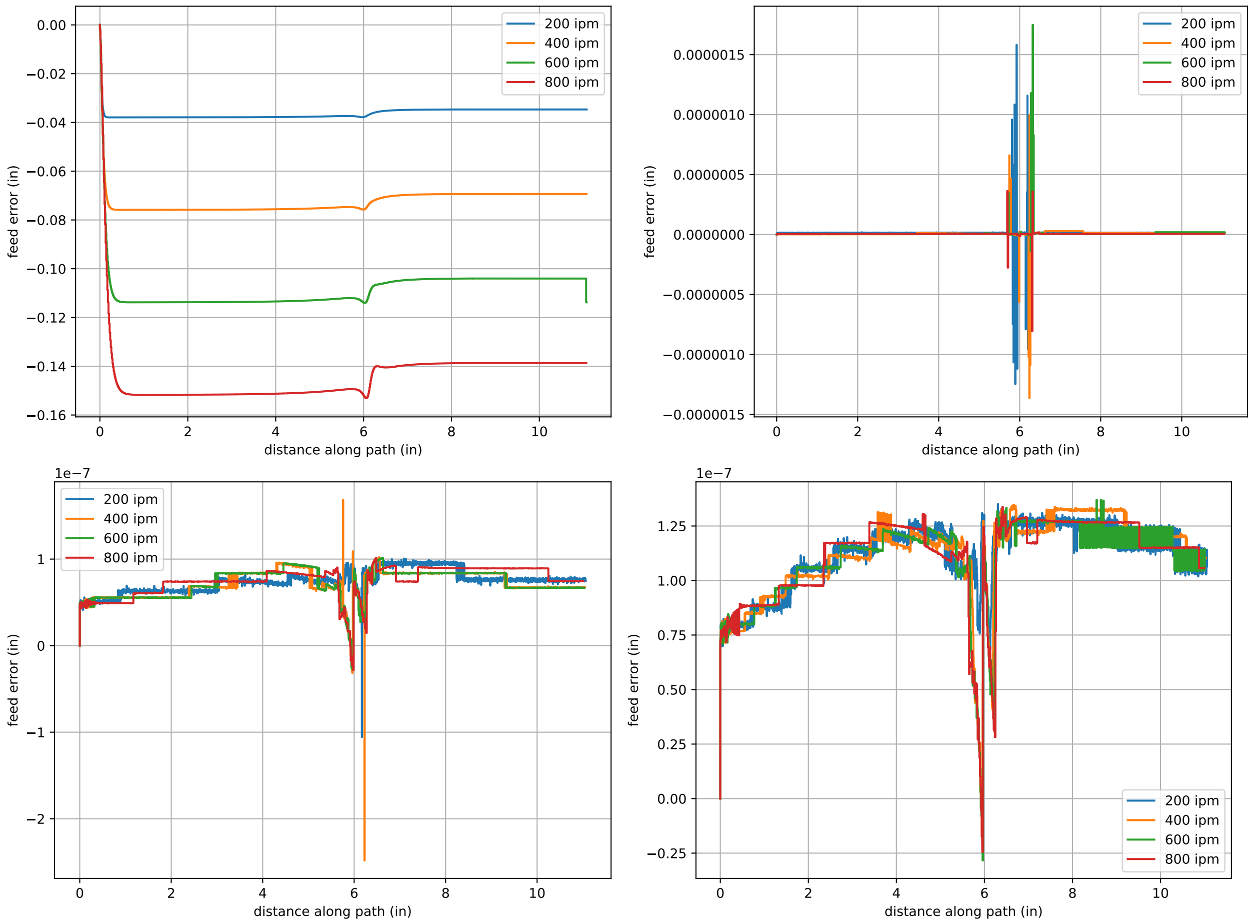

Figure 6: Simulated feed error for various prediction horizons and feedrates.

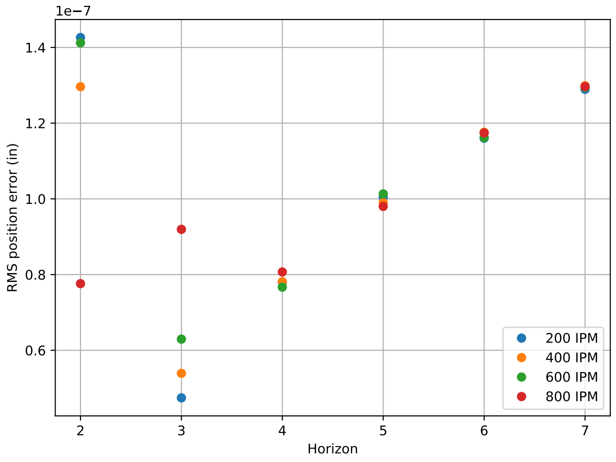

Figure 7: The rms position error vs horizon length for various feedrates.

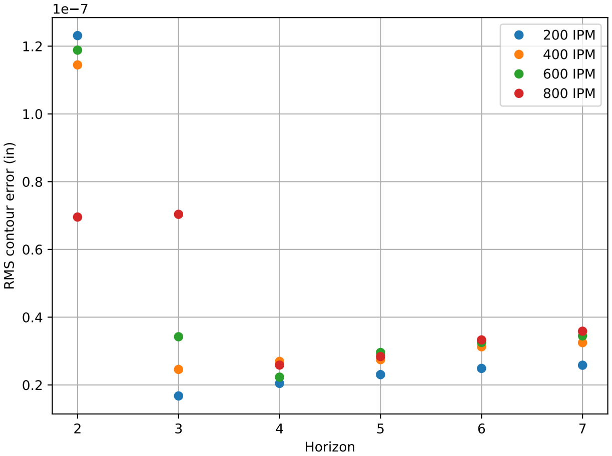

Figure 8: The rms contour error vs horizon length for various feedrates.

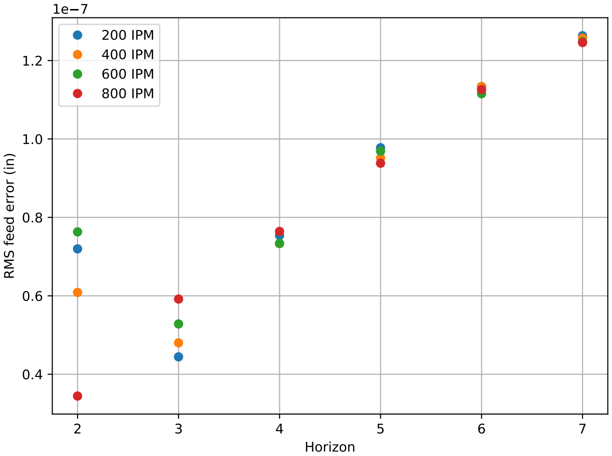

Figure 9: The rms feed error vs horizon length for various feedrates.

Figure 3 shows that the MPC compensation subdues the extremum feedrate error by several orders of magnitude relative to the uncompensated case (note the difference in the vertical axis scales), with increasing horizon smoothing the deviation near the point extremum curvature. However, the duration of the deviations is so small that the rms feedrate error for the entire traversal is not significantly influenced.

Figure 4 shows that the position error for the uncompensated case is proportional to the feedrate, and approximately five orders of magnitude worse than the MPC–compensated response. With the MPC compensation, increasing the horizon seems to smooth out spikes in error which may be numerical in nature. As with the feedrate error, local extrema in the position error occur near the curve point of maximal curvature, and the magnitude of these extrema decrease as the horizon length is increased. Again, because of the short duration of these errors, they do not have a significant impact on the overall rms position error.

Figure 7 shows that the rms position error is, on average, smallest for a horizon of three time steps. The outlier to this trend is the 800 ipm case, which has a local maximum. One possible explanation is that the longer look–ahead causes the error to be distributed over more sampling intervals, and since the error is so small the numerical noise is significant. In general, the literature does support the empirical observation that MPC performance on nonlinear systems is not monotonic with horizon length [3, 13] so it may also be that the high feedrate and large curvature drive a nonlinear behavior.

The contour error plots in Figure 5 exhibit patterns similar to the position error plots in Figure 4, likely because the position error is so small with MPC that the contour error must also be very small. This verifies the hypothesis that treating the machine axes independently can be used to ensure a small contour error, provided the position error on each axis is sufficiently small.

Figure 6 shows the signed feed error. Without compensation, the feed error is non–positive (lagging the reference) as expected for a causal feedback system. However with compensation, the feed errors tend to be positive. Figure 9 shows that the rms feed error is almost monotonic in performance, with $H = 2$ performing about as well as $H = 3$. The slope of the rms feed error curve in Figure 9 is steeper than the rms contour error curve in Figure 8, indicating that as the horizon increases, the feed error dominates the contour error.

Based on the results from the simulation study, a prediction horizon of $H = 3$ is selected for the experimental study. This horizon length showed the best position error performance. Since computational cost scales linearly with horizon length, the relatively short horizon will also help reduce computation time so the compensation can be calculated in real time.

5 Experimental Implementation

The MPC–based inverse dynamics scheme was implemented on the 3–axis CNC milling machine shown in Figure 10. This machine is controlled by the OpenCNC open–architecture software, which incorporates a low–level PID controller for each axis and provides a programming interface to send position commands to each axis and measure their actual positions using position encoders at a sampling frequency of 1024 Hz. The control computer is based on a 500 MHz processor running Windows NT, augmented by an extension compatible with execution of real–time processes. It has sufficient RAM to record approximately 40 seconds of machine run–time data.

A custom C language solver that takes advantage of the sparse structure of the QP (3) was generated using `cvxgen` [15] and integrated into the OpenCNC architecture. Initially, it was assumed that the 1024 Hz update rate was enforced through an interrupt. After testing and comparison with the simulation results, however, it was determined that the 1024 Hz update rate is an upper bound. Furthermore, it was found that executing the QP solver twice (once for each axis) caused the control loop to update slower than the expected 1024 Hz. In turn, this created the appearance of a machine dynamics faster than expected. To remedy this, the feedback gain k_p was reduced to 10,000 resulting in model parameters $a_2 = 4.681 \times 10^{-5}$ and $a_1 = 1.936 \times 10^{-2}$. The reduced feedback gains were chosen to dampen the effect of the control loop timing delay and to amplify the difference between the machine position errors with and without MPC.

Two test curves were employed in the experiments. Figure 11 shows, for various feedrates, the commanded path both without and with the compensation term (computed in real time by the control computer) for the `sharp turn` test curve, which is essentially the same curve used in the simulation study but augmented by lead–in and lead–out segments to minimize transient effects. Figure 11 shows plots of the actual executed paths for this test curve, and a "zoomed in" view of the system response is provided in Figure 13. Figure 14

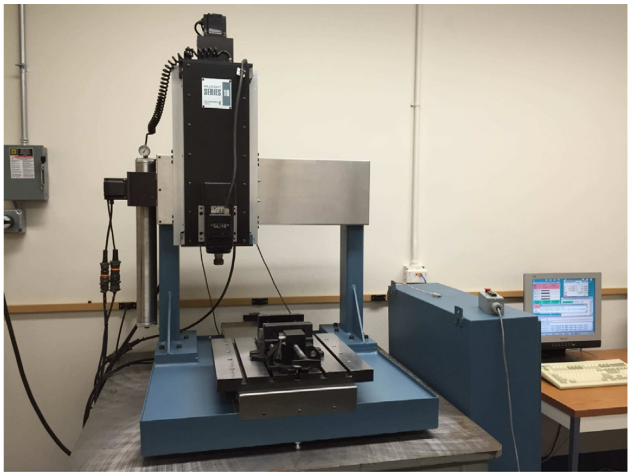

Figure 10: Experimental testbed — gantry mill with OpenCNC control system.

shows, for various feedrates, the commanded path both without and with the compensation term (computed in real time by the control computer) for the `clover` test curve for each feedrate, with the deviation from the uncompensated curve magnified.[2] Figure 15 shows the uncompensated and compensated system response with magnified contour errors.

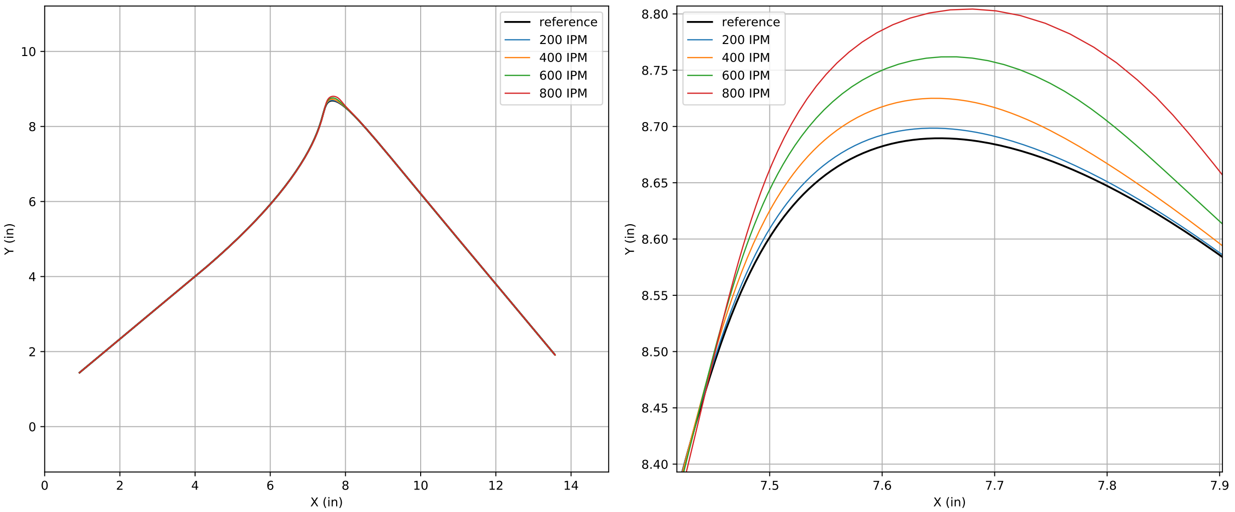

Figure 11: Experimental predictive curve compensation for the `sharp turn` curve at feedrates of 200, 400, 600, and 800 ipm

Figure 16 shows the measured feedrates for the `sharp turn` test curve without and with compensation. For the cases with nominal feedrates below 800 ipm, the measured feedrate with compensation is significantly better than without compensation. For the 800 ipm case there is still significant deviation in the feedrate with compensation, likely due to acceleration limits of the machine. The feedrate tends to exceed the specified feedrate with compensation, due to the optimization of the predicted position error over the moving horizon. In contrast, the feedrate tends to be reduced near the point of the curve with maximal curvature without compensation, as expected for a causal system based only on feedback control. Figure 17 shows the measured feedrates for the `clover` test curve without and with compensation. The MPC–based compensation reduces

[2] Magnified views of the compensated `sharp turn` test curve are omitted, since their extreme behavior in the vicinity of the curvature maximum makes them difficult to interpret.

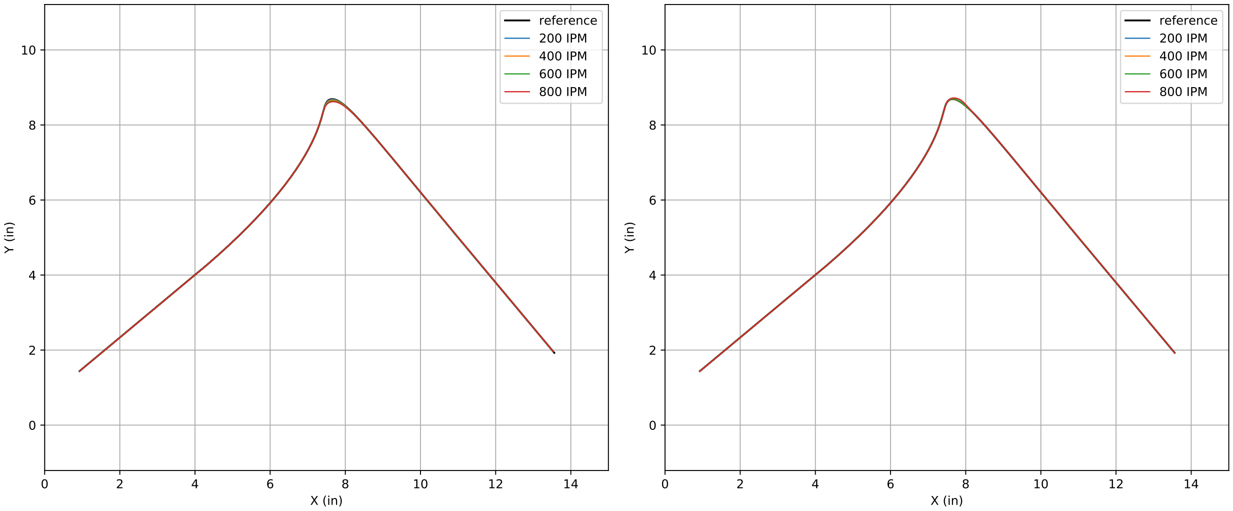

Figure 12: Contour of system response for the `sharp turn` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

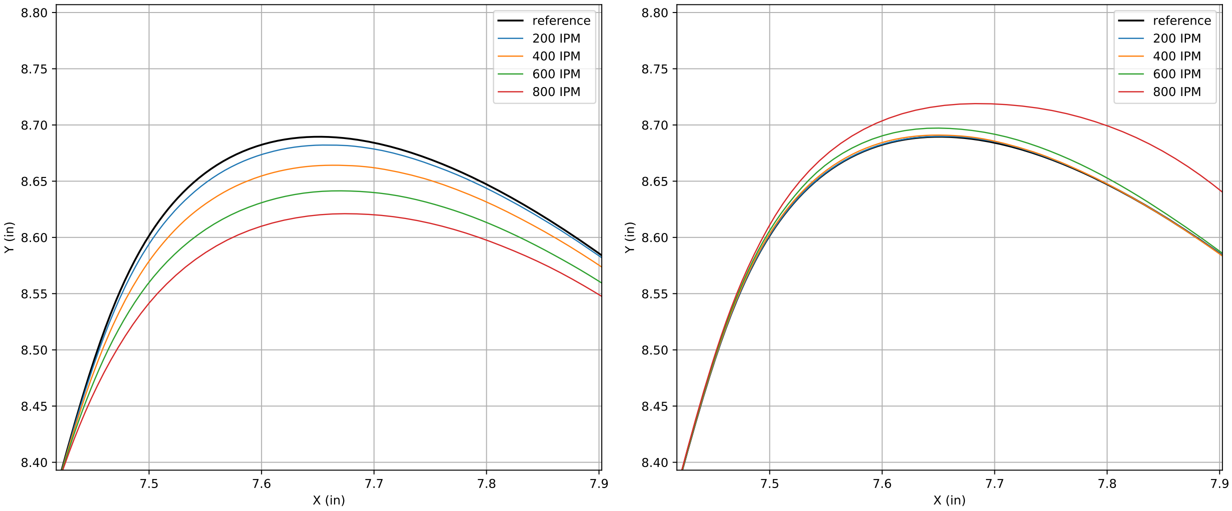

Figure 13: Close-up of contour for the `sharp turn` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

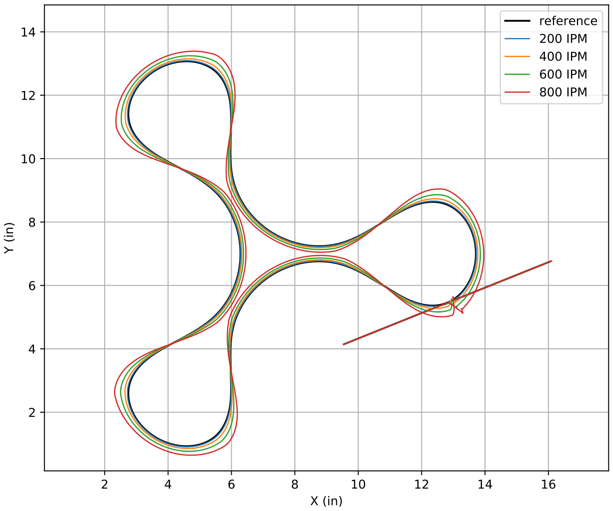

Figure 14: Predictive curve compensation magnified ×20 for the `clover` test curve at feedrates of 200, 400, 600, and 800 ipm.

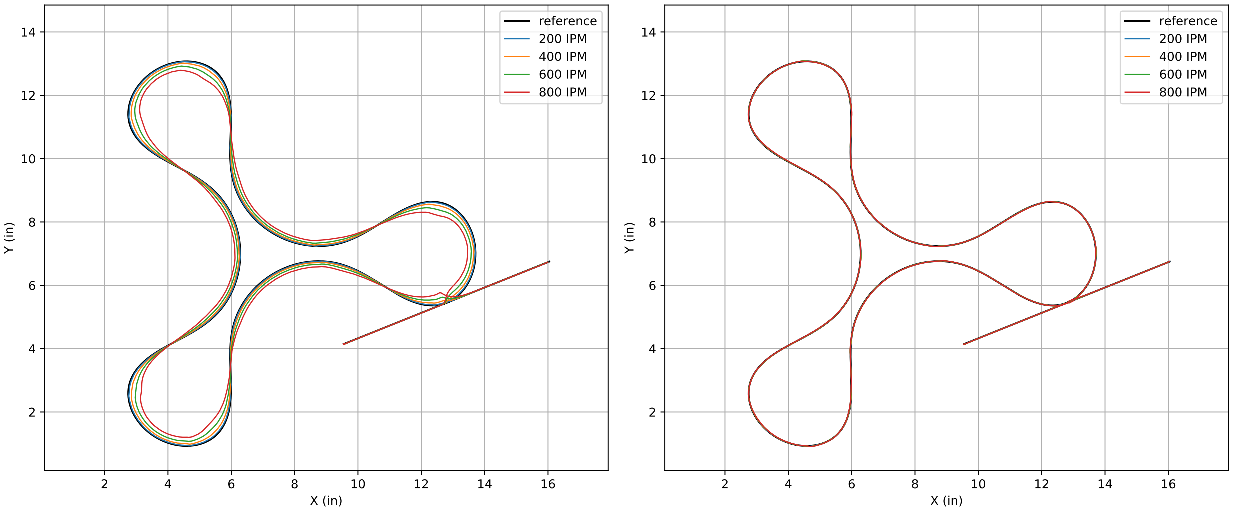

Figure 15: Contour error magnified ×20 for the `clover` curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

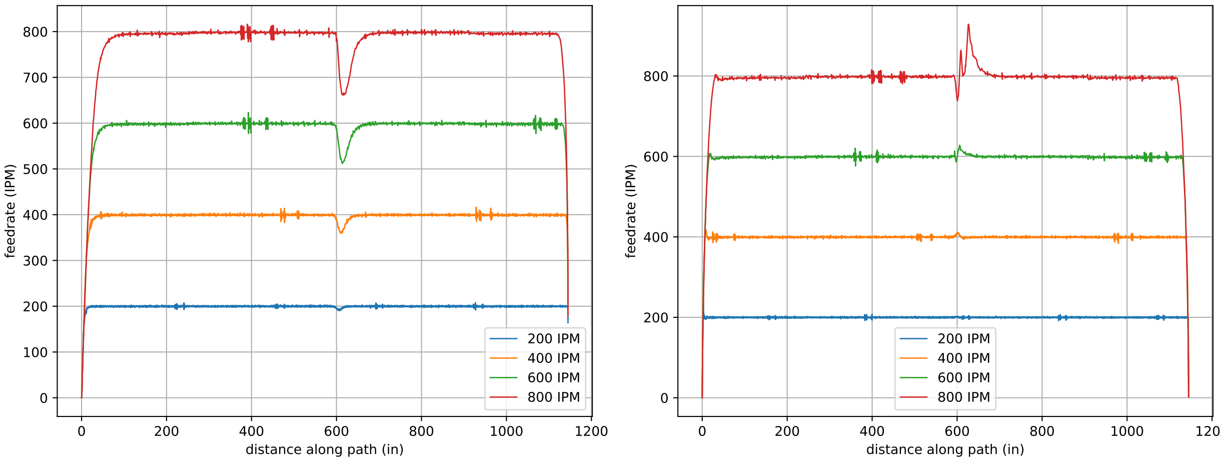

Figure 16: Measured feedrates for the `sharp turn` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

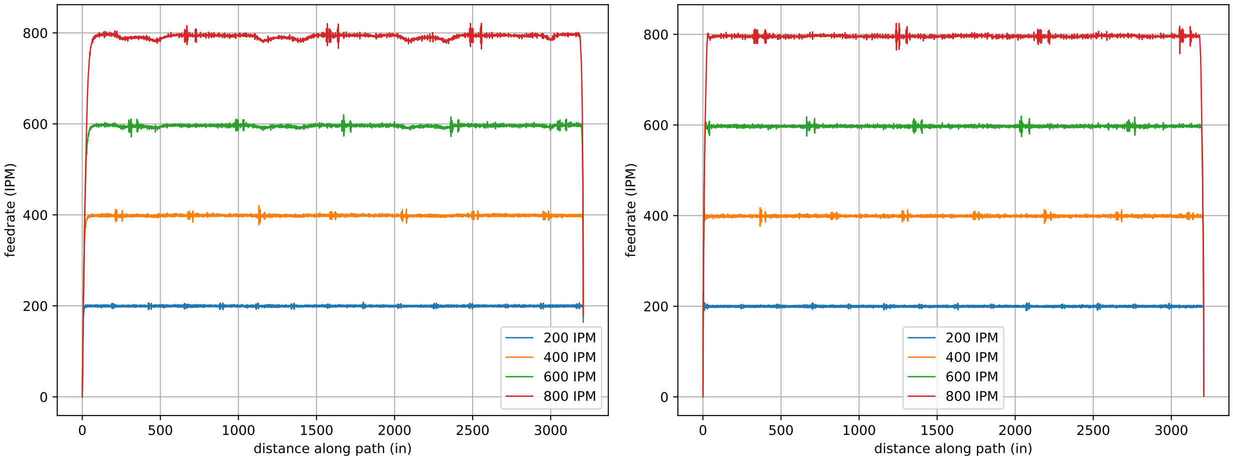

Figure 17: Measured feedrates for the `clover` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

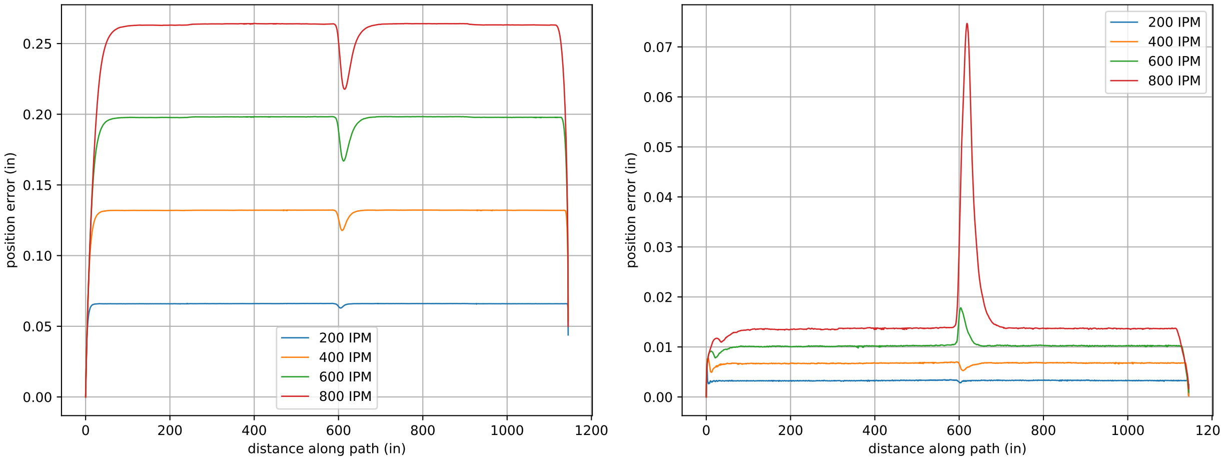

Figure 18: Measured total position error for the `sharp turn` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

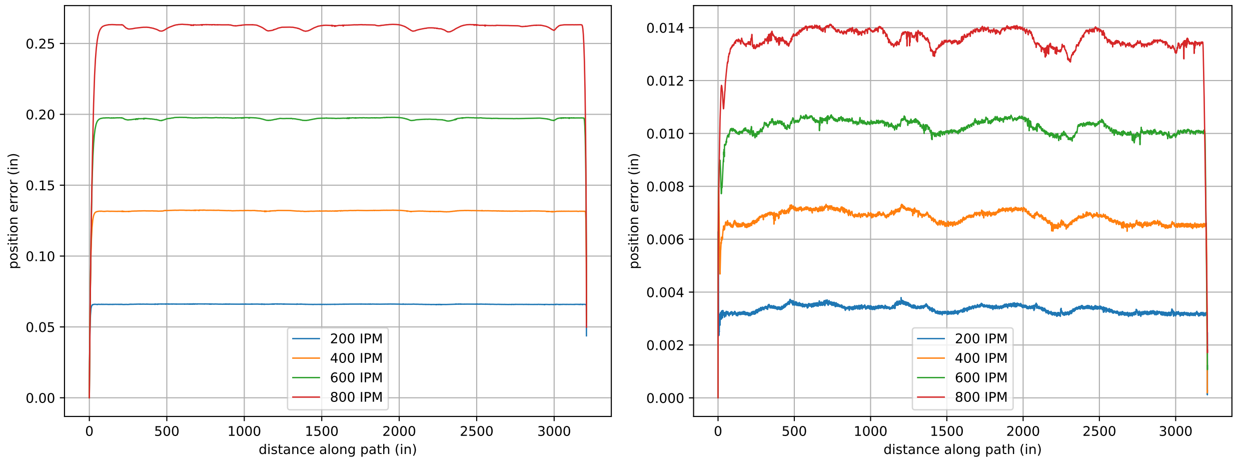

Figure 19: Measured total position error for the `clover` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

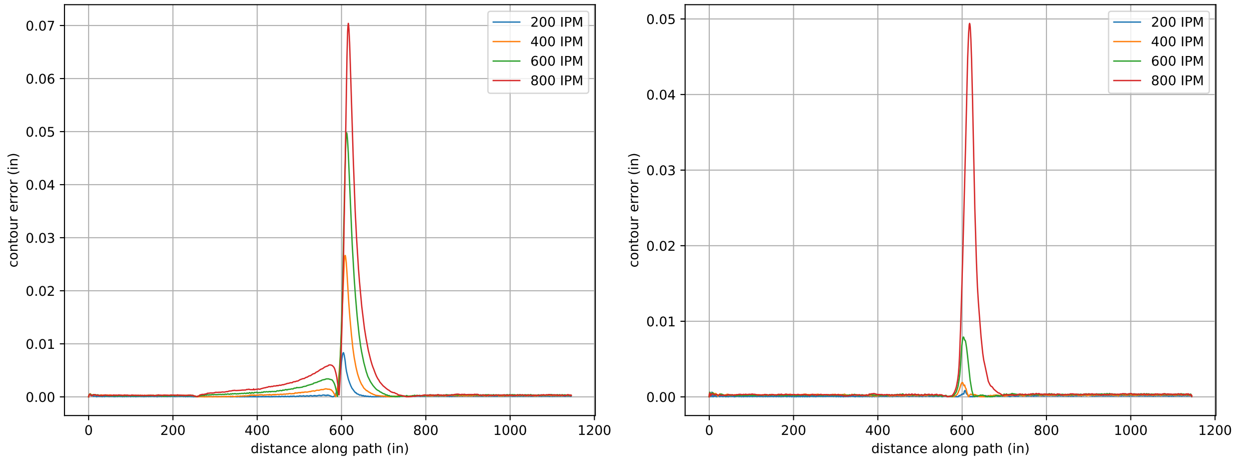

Figure 20: Measured contour error for the `sharp turn` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

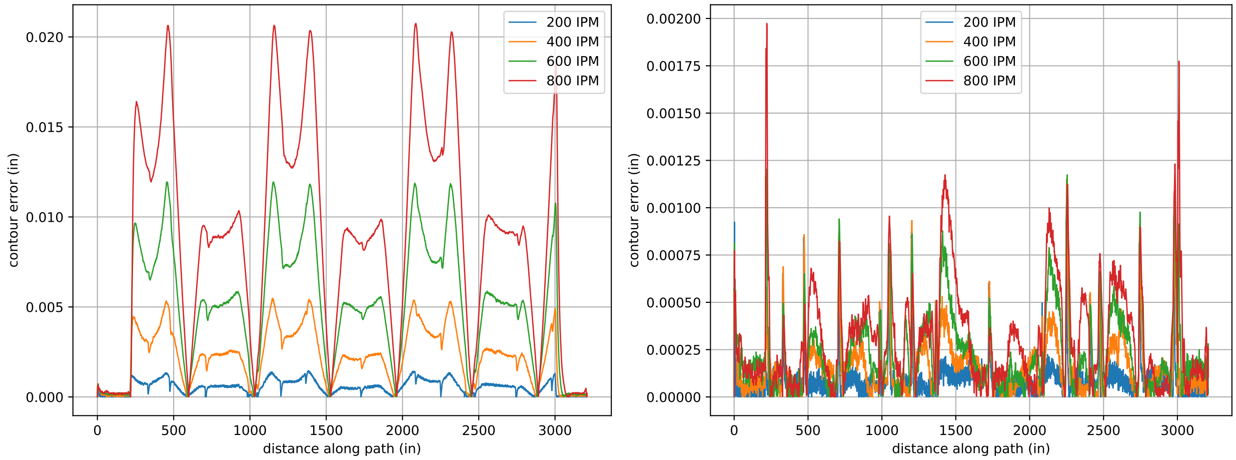

Figure 21: Measured contour error for the `clover` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

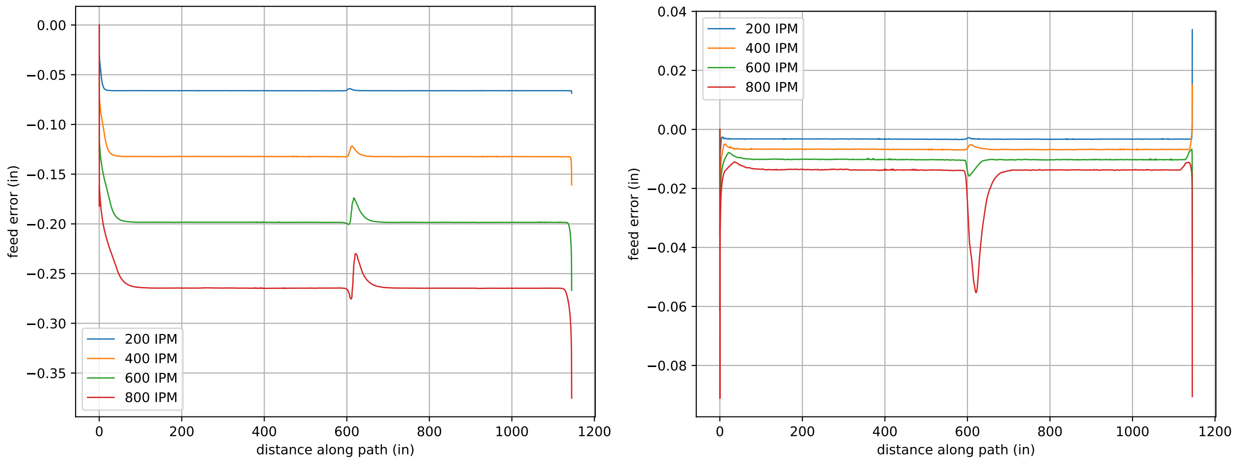

Figure 22: Measured feed error for the `sharp turn` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

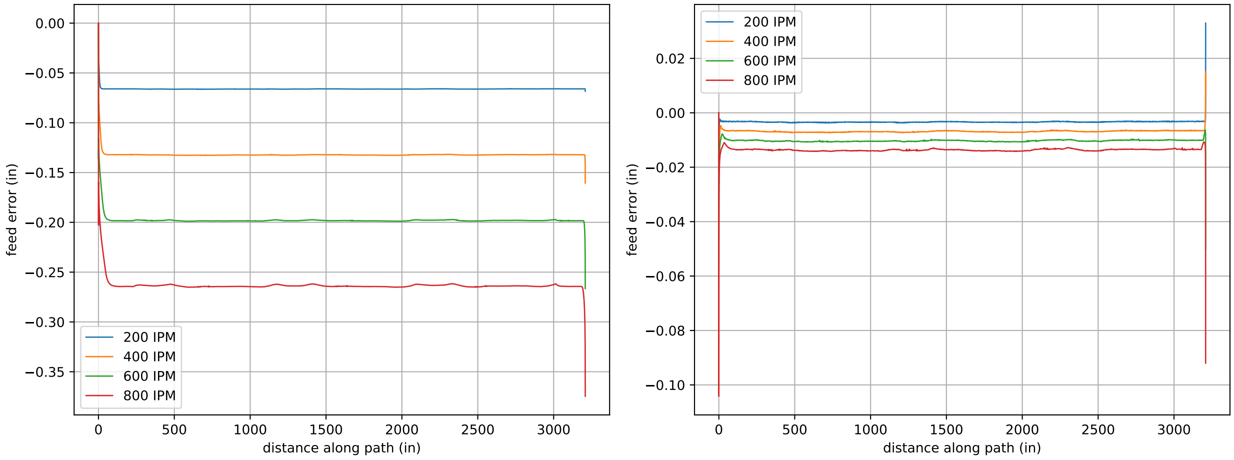

Figure 23: Measured feed error for the `clover` test curve at feedrates of 200, 400, 600, and 800 ipm without (left) and with (right) predictive compensation.

the variation of the feedrate for the 600 and 800 ipm cases. At lower nominal feedrates, the measured feedrate without compensation is about as consistent as with compensation.

Figures 18 and 19 show the measured position error for each test curve. For both test curves, the MPC–based compensation produces position errors that are generally an order of magnitude better than without compensation. Even for the 800 ipm case on the `sharp turn`, the peak position error is still about 4× smaller than for the uncompensated case, and an order of magnitude smaller away from the curvature extremum. The uncompensated cases show a local minimum of position error near the point of maximal curvature.

Figure 20 shows the measured contour error for the `sharp turn` without and with MPC compensation. The compensation significantly reduces the contour error across most of the curve. For the higher feedrates, the peak contour error with compensation approaches the peak contour error without compensation. This behavior may be due to the feedrate dependence of model parameters as discussed in [17] or acceleration limits of the machine axes. The contour errors for the `clover` curve shown in Figure 21 show that the compensation is an order of magnitude better throughout the curve.

Figure 22 illustrates the measured feed error for the `sharp turn` curve without and with compensation. Without compensation, the feed error is always non–positive as in the simulation study, but only temporarily improves slightly near the point of maximal curvature at high feedrates. With compensation, the feed error is generally an order of magnitude smaller than without. At the higher feedrates, the feed error with compensation does exhibit a local extremum at the point of highest curvature, although the magnitude is still smaller than without compensation by a factor of about 5. For the `clover` test curve, the feed error with MPC on is an order of magnitude better than with MPC off throughout the curve, as shown in Figure 23.

6 Acceleration Limits

To verify that the source of the tracking errors of the compensated path are indeed due to acceleration limits, a model of the acceleration limits was developed, incorporated as a constraint in the MPC optimization problem (3), and simulated. The simple axis model described in Section 2 treats the motor as a resistive element. To model the acceleration limits, we add a "back EMF" term so that the motor current i is related to the control voltage u through

$$i = k_a \left(u - k_e \omega\right)$$

where k_a is the current amplifier gain, k_e is the motor back EMF constant, and ω is the motor angular speed. This results in the modified transfer function

$$\frac{p(s)}{c(s)} = \frac{K k_p}{J s^2 + \left(B + k_a k_e k_t\right) s + K k_p}\,, \tag{4}$$

where k_t is the motor torque gain. Note that the difference between between the transfer functions (2) and (4) is that the damping coefficient of the former depends only on the mechanical damping B while the latter is augmented by the back EMF term. Therefore, the value a_2 determined from system identification already captures the damping from both mechanical and electromechanical sources. Following [10], we consider acceleration limits due to current limits and voltage limits modeled as constraints of the form

$$|a| \le a_i \qquad \text{and} \qquad |a| \le a_v - \frac{B}{J}\,|v|\,, \tag{5}$$

where a_i is the maximum acceleration due to the current limit, a_v is the maximum acceleration due to the voltage limit, and $\frac{B}{J}$ captures the acceleration effect of the mechanical damping. To estimate these values, the measured axis acceleration and velocity from the data collected in Section 5 were plotted on the acceleration-velocity phase plane and the hexagonal polytope defined by the constraints (5) was manually adjusted to obtain a good fit of the data. The identified values are $a_i = 600\,\text{in/s}^2$, $a_v = 750\,\text{in/s}^2$, and $\frac{B}{J} = 55$. Note that the identified value of $\frac{B}{J}$ implies that only about $^1/_7$th of the measured damping is due to mechanical effects with the majority of the effective damping due to the back EMF of the motor. A plot of the acceleration-velocity phase plane of a portion of the data and the fit polytope is shown in Figure 24.

Initial attempts to incorporate the acceleration constraints led to convergence problems which were ultimately remedied by dropping the acceleration constraint due to the voltage limit and adding a constraint

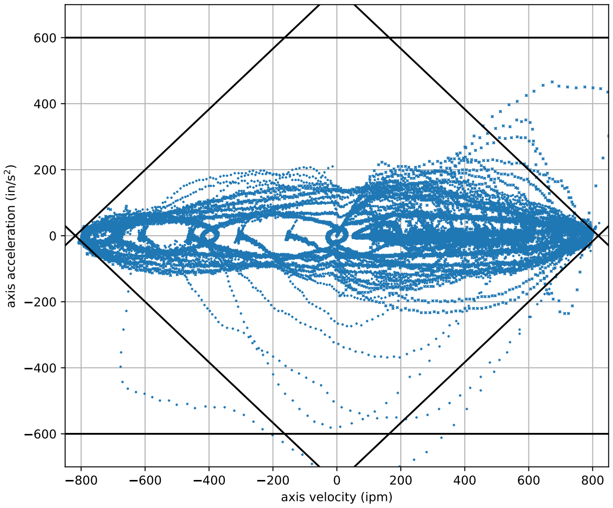

Figure 24: Acceleration-velocity phase plane of experimental data with estimates for acceleration limits.

on the predicted position to ensure the solution remained in the workspace of the machine. A close-up view of the compensated `sharp turn` test curve and the corresponding simulated response at a nominal feedrate of 800 ipm using MPC with several prediction horizons is shown in Figure 25. Comparing the responses to the compensated curves with short horizons shown in this figure to the experimental results of the compensated response shown in Figure 13 suggests that using the simplified acceleration limit is representative of the actual machine dynamics. It is also worth noting that after the introduction of the acceleration constraint, the compensated reference curves are no longer smooth. In contrast, the compensation computed without constraints on the acceleration produced smooth compensation curves, such as shown in Figure 2.

Figure 26 shows the contour error of the simulated response for several horizon lengths and the RMS errors for each simulated horizon length. The RMS contour error is not monotonic in the prediction horizon length although the compensation with the two longest horizons produce significantly better RMS contour errors than the compensation with shorter horizons. This suggests that in order for MPC to improve the performance under these types of actuator limits, the prediction horizon must be long enough that the restriction imposed by the actuator limits affects the optimization problem of MPC early enough to respond appropriately. To illustrate what “early enough” means, the plot of the compensated curves includes gray dots on the original reference curve to illustrate a prediction length of a 64 time steps. This horizon length is long enough to capture the entire corner of the curve. Inspection of the simulated response suggests that MPC compensation with this horizon length can account for the significant acceleration requirements while respecting the limits of the actuators. Compensation with longer prediction horizons was also simulated, with essentially indistinguishable responses, suggesting that longer prediction horizons have diminishing returns after some critical length. This behavior also suggests that the compensation and response from MPC may asymptotically approach the optimal compensation given the demands of the test curve and actuator limits.

The contour error along the path length in Figure 26 and the close-up view of the contour response in Figure 25 show oscillations present for the acceleration–limited compensated system. It is well known that MPC compensation without constraints approaches optimal feedback control as the horizon extends to infinity. However, it is surprising to see an MPC–controlled system with constraints and a short horizon exhibit classical second–order dynamic characteristics including overshoot and decaying oscillations. It is possible that the without the workspace constraints this dynamic behavior may have been unstable and the constraints, which should have been redundant, created a virtual damping effect that helped convergence.

Figure 27 shows the simulated feedrates for the compensated test curves for several horizon lengths and the RMS errors for each horizon length. The RMS feedrate error appears to decrease linearly with increasing horizon length. This is likely a consequence of the square–error objective function of the MPC optimization problem (3), since square–error minimization is related to mean minimization.

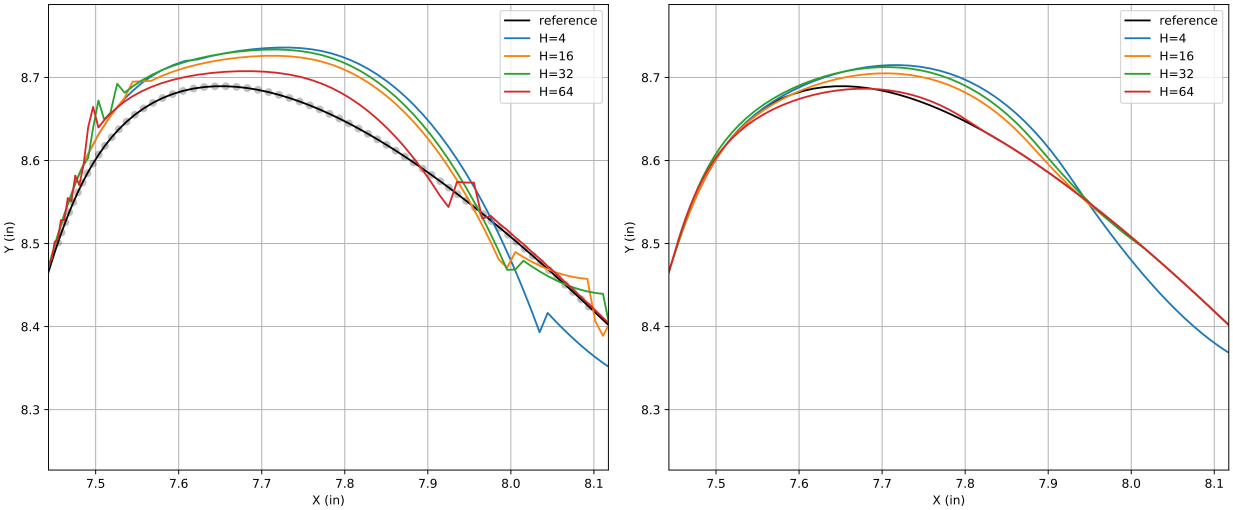

Figure 25: Close-up contour of the compensated `sharp turn` test curve (left) with corresponding simulated response (right) at a nominal 800 ipm for using MPC with several prediction horizons. The gray dots on the original reference curve illustrate a prediction length of a 64 time steps.

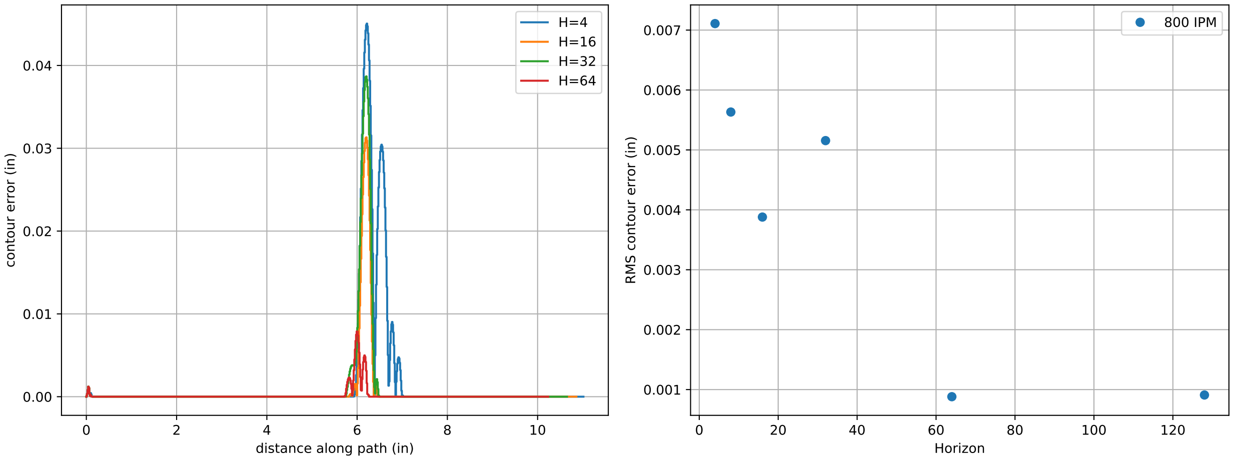

Figure 26: Simulated contour errror for the `sharp turn` test curve at feedrates of 800 ipm with predictive compensation using acceleration constraints.

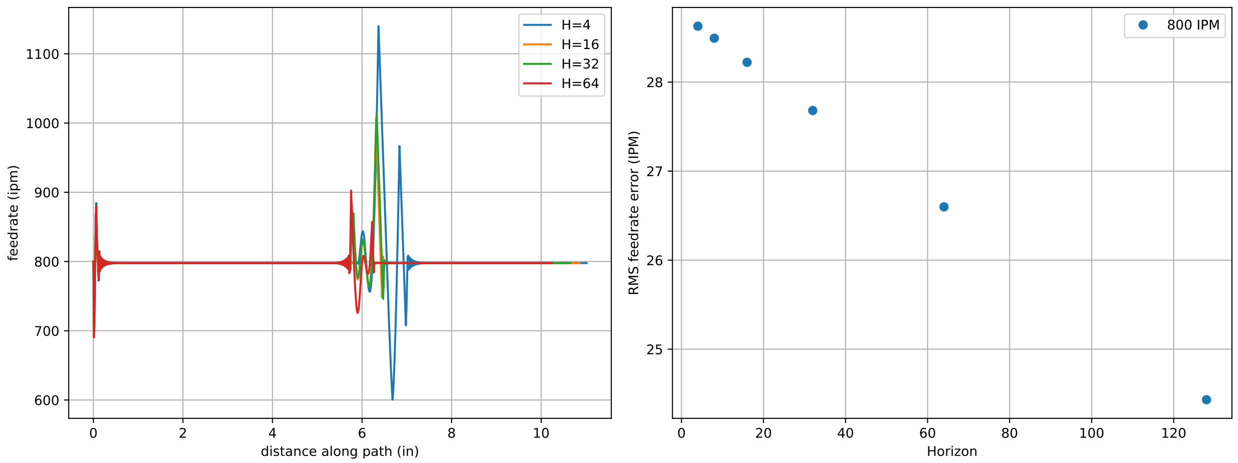

Figure 27: Simulated feedrates for the `sharp turn` test curve at feedrates of 800 ipm with predictive compensation using acceleration constraints.

7 Conclusion

A model predictive control (MPC) approach to inverse dynamics compensation (IDC) has been developed, and the performance was analyzed through both simulations and experiments on a CNC milling machine. The MPC approach to IDC was formulated as a quadratic programming optimization problem, and solved using off–the–shelf tools. In the simulation study, significant improvements in performance were found with even relatively short prediction horizons. In the experimental study, real–time performance enhancements obtained with MPC were demonstrated, in terms of the overall position error, contour error, and feedrate accuracy. The experimental study revealed potential areas of improvement, including the need to reduce solver times to increase the compensation update rate, and the inclusion of axis acceleration limits to improve tracking performance for high–curvature paths executed at high feedrates. Inverse dynamics compensation based on MPC is a versatile approach to improving tracking accuracy in motion control, offering the ability to also incorporate equality or inequality constraints and non-linear dynamic effects, which we hope to address in future studies.

In addition to compensating for machine dynamics, another potential application of MPC is in mitigating the effects of variable cutting forces, based on *a priori* knowledge of the tool and part geometry, depth and width of cut, workpiece specific cutting energy, the desire to suppress tool chatter, etc. This is a challenging problem, that will require a comprehensive independent investigation.

Acknowledgements

The authors are grateful to Bohan Zhu for running the system identification tool.

References

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

[2] M. S. Andersen, J. Dahl, and L. Vandenberghe (2014), CVXOPT: A python package for convex optimization, version 1.1.7.

[3] V. Bachtiar, E. C. Kerrigan, W. H. Moase, and C. Manzie (2016), Continuity and monotonicity of the mpc value function with respect to sampling time and prediction horizon, *Automatica* **63**, 330–337.

[4] J. R. Conway, C. A. Ernesto, R. T. Farouki, and M. Zhang (2012), Performance analysis of cross–coupled controllers for CNC machines based on precise real–time contour error measurement, *International Journal of Machine Tools and Manufacture* **52**, 30–39.

[5] 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.

[6] A. El Khalick M. and N. Uchiyama (2011), Discrete-time model predictive contouring control for biaxial feed drive systems and experimental verification, *Mechatronics* **21**, 918–926.

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

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

[9] 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.

[10] R. T. Farouki, Y. Tsai, and C. S. Wilson (2000), Physical constraints on feedrates and feed accelerations along curved tool paths, *Computer Aided Geometric Design* **17**, 337–359.

[11] R. Komanduri, K. Subramanian, and B. F. von Turkovich (eds.) (1984), *High-speed machining*, vol. 12, ASME, New York.

[12] D. Lam, C. Manzie, and M. C. Good (2012), Model predictive contouring control for biaxial systems, *IEEE Transactions on Control Systems Technology* **21**, 552–559.

[13] B. W. Margolis, M. A. Ayoubi, and S. S. Joshi (2020), Nonlinear model predictive control of reentry vehicles based on takagi-sugeno fuzzy models, *Journal of the Astronautical Sciences* **67**, 113–136.

[14] B. W. L. Margolis (2017), Simupy: A python framework for modeling and simulating dynamical systems, *Journal of Open Source Software* **2**, 396.

[15] J. Mattingley and S. Boyd (2012), Cvxgen: A code generator for embedded convex optimization, *Optimization and Engineering* **13**, 1–27.

[16] S. J. Qin and T. A. Badgwell (2003), A survey of industrial model predictive control technology, *Control engineering practice* **11**, 733–764.

[17] T. F. Schraeder and R. T. Farouki (2014), Experimental performance analysis of an inverse dynamics CNC compensation scheme for high-speed execution of curved toolpaths, *International Journal of Advanced Manufacturing Technology* **73**, 195–208.

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

[19] L. Tang and R. G. Landers (2012), Predictive contour control with adaptive feed rate, *IEEE/ASME Transactions On Mechatronics* **17**, 669–679.

[20] J. Tlusty (1993), High-speed machining, *CIRP annals* **42**, 733–738.

[21] L. Wang (2009), *Model Predictive Control System Design and Implementation Using MATLAB*, Springer Science & Business Media.

[22] Y. Wang and S. Boyd (2010), Fast model predictive control using online optimization, *IEEE Transactions on Control Systems Technology* **18**, 267–278.

[23] J. Yang, H.-T. Zhang, and H. Ding (2017), Contouring error control of the tool center point function for five-axis machine tools based on model predictive control, *International Journal of Advanced Manufacturing Technology* **88**, 2909–2919.

[24] S. Yang, A. H. Ghasemi, X. Lu, and C. E. Okwudire (2015), Pre-compensation of servo contour errors using a model predictive control framework, *International Journal of Machine Tools and Manufacture* **98**, 50–60.

[25] X. Yang, R. Seethaler, C. Zhan, D. Lu, and W. Zhao (2019), A model predictive contouring error pre-compensation method, *IEEE Transactions on Industrial Electronics* **67**, 4036–4045.

[26] 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.

[27] B. Zhu and R. T. Farouki (2019), A general framework for solving inverse dynamics problems in multi-axis motion control, *ISA Transactions* **95**, 130–143.