VIBRATION AVOIDANCE AND CONTOUR ERROR COMPENSATION IN HIGH SPEED FIVE-AXIS MACHINE TOOLS USING COMMAND SHAPING TECHNIQUES

by

MOHAMMAD REZAYI KHOSH DARREGI
B.Sc. Sharif University of Technology, Iran, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies
(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

September 2012
© Mohammad Rezayi Khoshdarregi, 2012
Abstract

The present machine tools travel at high velocities with large acceleration and jerk, which creates large inertial loads with wide frequency content, resulting in excitation of the natural frequencies of the machine. This thesis presents a command shaping module which avoids structural vibrations and improves contouring accuracy in five-axis machine tools.

Inertial-induced vibrations of feed drives are avoided by applying a sequence of input shaping impulses on the axis position commands as a function of natural frequencies of the machine. Input shapers block those harmonics of the command which can excite the modes of the machine. While command shaping avoids the excitation of transient vibrations caused by inertial loads, it introduces a time delay which increases the contouring errors in multi-axis motion control systems.

Although input shaping can effectively avoid vibrations, it distorts the toolpath in multi-axis contouring due to added time delays. The distortion increases the contour errors caused by the limited bandwidth of axis servo drives, resulting in relatively large geometric errors and violation of the required tolerance. The transfer function of the axis drives and kinematic configuration of the machine tool are used to estimate the contouring error, which is decoupled into axis components, and compensated prior to sending the position commands to each drive.

The proposed integrated input shaping and contour error compensation module can be applied on any arbitrary trajectory, and the shaped-compensated command results in vibration-free and accurate contouring of the desired path. The input shaping and contouring error compensation techniques introduced in the thesis have been experimentally validated in single, two and five axis machine tools.
Table of Contents

Abstract .............................................................................................................................. ii
Table of Contents ............................................................................................................. iii
List of Tables .................................................................................................................... v
List of Figures ................................................................................................................... vi
Nomenclature ................................................................................................................... x
Acknowledgment ............................................................................................................ xv
1 Introduction ................................................................................................................ 1
2 Literature Review ....................................................................................................... 6
   2.1 Overview ............................................................................................................... 6
   2.2 Vibration Reduction in Machine Tool Feed Drives .............................................. 6
   2.3 Input Shaping of Reference Trajectory ................................................................. 8
   2.4 Improving Contouring Performance in Multi-axis Machines ............................. 12
   2.5 Conclusion ........................................................................................................... 14
3 Input Shaping of Reference Commands for Vibration Avoidance...................... 15
   3.1 Introduction ......................................................................................................... 15
   3.2 Vibration Cancellation Technique ...................................................................... 15
   3.3 Reference Shaping Principle ............................................................................... 19
   3.4 Input Shaper Design ............................................................................................ 20
      3.4.1 Zero-Vibration (ZV) Shaper ........................................................................ 21
      3.4.2 Robustness of Zero-Vibration (ZV) Shaper ................................................. 23
      3.4.3 Zero-Vibration and Derivative (ZVD) Shaper ............................................. 25
      3.4.4 Robustness of the Zero-Vibration and Derivative (ZVD) Shaper ............... 31
      3.4.5 Further Improvement of Shaper Robustness ............................................. 33
   3.5 Discretization Effect in Digital Input Shaping .................................................... 36
   3.6 Comparing Input Shapers with Classical Frequency Domain Filters ................. 41
   3.7 Summary ............................................................................................................. 47
4 Improving Accuracy by Pre-compensation of Contour Errors ........................... 48
   4.1 Introduction ......................................................................................................... 48
   4.2 Definition of Contour Error ................................................................................. 48
4.3 Contour Error Estimation using Frenet Frame ....................................................... 50
4.4 Trajectory Distortion caused by Input Shaping ..................................................... 53
4.5 Pre-compensation of Contour Errors .................................................................... 58
  4.5.1 Pre-compensation of Path Distortion and Machine-dependent Contour Errors 60
  4.5.2 Simplified Versions of the Proposed Command Shaping Module ................. 67
4.6 Statistical Evaluation of Contour Error Compensation ....................................... 70
4.7 Summary ............................................................................................................. 77

5 Implementation and Experimental Results ............................................................. 78
  5.1 Introduction ......................................................................................................... 78
  5.2 Implementation on the Single-axis Linear Drive with Oscillator ....................... 78
    5.2.1 Experimental Setup ...................................................................................... 79
    5.2.2 System Modeling ......................................................................................... 80
    5.2.3 Controller Design ......................................................................................... 84
    5.2.4 Single-mode and Multi-mode Input Shaping ............................................... 85
    5.2.5 Experimental Demonstration ....................................................................... 88
  5.3 Multi-axis Implementation on the Linear Motor Driven XY Table ................. 89
    5.3.1 Experimental Setup ...................................................................................... 89
    5.3.2 System Modeling ......................................................................................... 90
    5.3.3 Controller Design ......................................................................................... 93
    5.3.4 Input Shaper Design ..................................................................................... 96
    5.3.5 Vibration Avoidance Experiment ................................................................. 98
    5.3.6 Contour Error Compensation Experiment ................................................. 104
  5.4 Five-axis Command Shaping with Avoidance of Real Structural Modes .... 108
    5.4.1 Experimental Setup .................................................................................... 108
    5.4.2 System Modeling ......................................................................................... 109
    5.4.3 Controller and Observer Design .................................................................. 113
    5.4.4 Identification of Structural Resonance Modes and Designing Input Shapers 117
    5.4.5 Implementation of Contour Error Compensation Module ......................... 118
  5.5 Summary ........................................................................................................... 127

6 Conclusions ............................................................................................................. 129

Bibliography ............................................................................................................... 131
List of Tables

Table 3.1: Comparison of performance achieved by input shapers and classical filters ..... 47
Table 5.1: Dynamic specifications of Siemens 1FN1 Linear Motor .................................. 79
Table 5.2: Dimensions of the Aluminum oscillator ......................................................... 80
Table 5.3: Dynamic parameters of the rigid model of the linear motor. ......................... 81
Table 5.4: Parameters of the oscillator tip vibrations ....................................................... 83
Table 5.5: Dynamic parameters of the rigid model of the linear motor. ......................... 91
Table 5.6: Actual resonance parameters of the closed loop systems ............................. 96
Table 5.7: Dynamic parameters of the rigid model of the drives ................................... 111
List of Figures

Figure 1.1: Schematic frequency-domain analysis of feed drive vibrations ....................... 3

Figure 1.2: Illustration of active damping and command shaping for vibration reduction in machine tool feed drives. ............................................................ 4

Figure 2.1: Basic cross coupling contour error control proposed by Koren [54]. .............. 13

Figure 3.1: Vector diagram of total vibration caused by a sequence of two impulses [46]. ................................................................................................................. 18

Figure 3.2: Self-canceling characteristic in a two-impulse sequence [42]. ..................... 19

Figure 3.3: Reference shaping using a sequence of impulses ........................................ 20

Figure 3.4: Sensitivity of ZV shaper against mismatch in frequency and damping ratio. 
\( r_c = \frac{\zeta_{act}}{\zeta_{mdl}} \),  \( r_\omega = \frac{\omega_{act}}{\omega_{mdl}} \) ......................................................... 24

Figure 3.5: Sensitivity of ZVD shaper against mismatch in frequency and damping ratio. 
\( r_c = \frac{\zeta_{act}}{\zeta_{mdl}} \),  \( r_\omega = \frac{\omega_{act}}{\omega_{mdl}} \) ................................................................. 32

Figure 3.6: Comparing the sensitivity of ZV and ZVD shapers against mismatch in frequency and damping ratio ................................................................. 33

Figure 3.7: Comparing the sensitivity of ZV, ZVD, EI and ZVDD shapers against mismatch in frequency and damping ratio ......................................................... 36

Figure 3.8: Two possible extreme cases for the discretization of a ZVD shaper [41] ......... 38

Figure 3.9: Minimum sampling time to keep the percentage of residual vibration caused by discretization of shaper below 5% ......................................................... 40

Figure 3.10: Frequency response function of ZVD and ZVDD shapers ......................... 43

Figure 3.11: System response to ZVD- and ZVDD-shaped step for the exact and mismatched (15%) models \( (\omega_\omega = 1\text{Hz}, \zeta = 0) \) ................................................. 44

Figure 3.12: Frequency response of the designed lowpass and bandstop filters .......... 45

Figure 3.13: System response to a step command shaped by lowpass and bandstop filters. (The mismatched model has 15% deviation from the modeled frequency) ...... 46
Figure 4.1: Illustration of tracking error and contouring error. ............................................. 49
Figure 4.2: Using Frenet frame in estimation of contour errors. ........................................... 50
Figure 4.3: Using shifted Frenet frame in estimation of the contour errors. ..................... 51
Figure 4.4: Trajectory distortion caused by input shaping. .................................................. 55
Figure 4.5: Contour error caused by input shaping for a circular path. ............................... 55
Figure 4.6: Maximum contour error caused by input shaping for a circular path contoured at different feedrates while avoiding different structural frequencies........ 56
Figure 4.7: Maximum contour error caused by input shaping in different distortion indices.......................................................................................................................... 57
Figure 4.8: Illustration of pre-compensation of contour errors. ......................................... 59
Figure 4.9: Schematic configuration of a five axis machine with tilting table (XYZAC) [Sencer, 2009]. ............................................................................................................. 61
Figure 4.10: Schematic block diagram of the proposed integrated five-axis input shaping and contour error compensation. ........................................................................ 62
Figure 4.11: Simplified version of the proposed command shaping module; only compensating the contour errors caused by the machine dynamics, no vibration avoidance............................................................................................................. 69
Figure 4.12: Schematic block diagram of the proposed integrated input shaping and contour error compensation for three-axis Cartesian machines. ......................... 70
Figure 5.1: Siemens 1FN1 Linear Motor with flexible oscillators. .................................... 79
Figure 5.2: Rigid body model of linear drives from input voltage to motor position. .......... 81
Figure 5.3: Modeling the tip position of the oscillator as base-excitation............................ 82
Figure 5.4: Block diagram of the closed loop system including the oscillator dynamics. .... 83
Figure 5.5: Bode plot of the system from input voltage to position of the table and oscillators......................................................................................................................... 84
Figure 5.6: Simulated motor position and oscillator tip position subject to the shaped and unshaped step inputs............................................................................................................. 87
Figure 5.7: Simulated vibration of the oscillator tip subject to shaped and unshaped step commands......................................................................................................................... 88
Figure 5.8: Simulated control signal in the shaped and unshaped cases. ......................... 89
Figure 5.9: Linear motor driven XY table used for multi-axis implementation of input
shaping ........................................................................................................................................ 90
Figure 5.10: Modeling of linear drives for each axis of the XY table. ......................... 91
Figure 5.11: Frequency response function of the open loop system for each axis of the
XY table. .................................................................................................................................. 92
Figure 5.12: Block diagram of the closed loop system for the XY table. ................ 94
Figure 5.13: Simulated frequency response function of the closed loop systems
including the imposed resonance modes. .............................................................................. 95
Figure 5.14: Toolpath and the interpolated axis commands in contouring the circle from
a quadrant point ...................................................................................................................... 99
Figure 5.15: Measured axis tracking errors for the shaped and unshaped commands in
contouring the circle starting from a quadrant point .......................................................... 100
Figure 5.16: Toolpath and the interpolated axis commands in contouring the circle from
a non-quadrant point .............................................................................................................. 102
Figure 5.17: Measured axis tracking errors for the shaped and unshaped commands in
contouring the circle starting from a non-quadrant point .................................................. 102
Figure 5.18: Measured axis control signals for the shaped and unshaped commands in
contouring the circle starting from a non-quadrant point .................................................. 103
Figure 5.19: Magnitude and axial components of shaping distortion in contouring the
circle starting from a non-quadrant point .............................................................................. 104
Figure 5.20: Experimental contour error caused by input shaping and drive dynamics in
contouring the circle starting from a non-quadrant point .................................................... 105
Figure 5.21: Axial contour error compensation signals (low-pass filtered at 3Hz) in
contouring the circle starting from a non-quadrant point .................................................... 106
Figure 5.22: Contour error comparison in following a circular path for the three cases:
unshaped, shaped, and shaped-compensated commands ..................................................... 106
Figure 5.23: Control signal comparison in following a circular path for the three cases:
unshaped, shaped, and shaped-compensated commands ..................................................... 107
Figure 5.24: Five-axis Fadal Machining Center used for the realistic experiment............. 108
Figure 5.25: Ferraris sensor used for measuring the torsional vibrations of ballscrews. .... 109

Figure 5.26: General configuration of ballscrew driven axes [Erkorkmaz, 2003]. ............ 110

Figure 5.27: Modeling the dynamic of rotary motors-driven feed drives. ...................... 110

Figure 5.28: Closed loop control system of the feed drives with state feedback controller. ........................................................................................................................................................................... 113

Figure 5.29: Closed loop control system of the feed drives with state feedback controller and Kalman filter ........................................................................................................................................................................... 117

Figure 5.30: FFT of the transient acceleration of the drives subject to the shaped and unshaped linear trajectory (maximum acceleration $a = 2000 \text{mm/s}^2$). ....................... 119

Figure 5.31: Sample five-axis helical toolpath used for contour error experiment. ............ 120

Figure 5.32: Trajectory components described in the workpiece frame (solid line) and machine axis commands (dash-line) in contouring a five-axis helical toolpath ($f = 150 \text{mm/s}$) ........................................................................................................................................................................... 121

Figure 5.33: Simulated (dash-line) and measured (solid) axis tracking errors in contouring the five-axis helical toolpath ($f = 150 \text{mm/s}$). ................................................................. 122

Figure 5.34: Machine-dependent contour error (measured-unshaped) and toolpath distortion caused by input shaping in contouring the five-axis helical toolpath ($f = 150 \text{mm/s}$). ........................................................................................................................................................................... 123

Figure 5.35: Predicted (simulated) components of tooltip contour error (in workpiece frame) in contouring the shaped toolpath ($f = 150 \text{mm/s}$). ................................................................. 124

Figure 5.36: Axis compensation signals (simulation) low-pass filtered at 10 Hz. ............... 125

Figure 5.37: Measured contour errors for the unshaped and shaped-compensated five-axis helical toolpath ($f = 150 \text{mm/s}$). ................................................................. 126

Figure 5.38: Simulated contour error for the unshaped and shaped-compensated commands when the axis control loops include feedforward friction compensation (in both cases) ($f = 150 \text{mm/s}$). ........................................................................................................................................................................... 127
Nomenclature

Symbols

A, B, C, D  State space model

$A_j$  Amplitude of the $j$th impulse in the shaping sequence

$A_{tot}$  Amplitude of the resultant summation vector

$b$  Bi-normal axis of Frenet frame

$D_{ind}$  Distortion index

$E$  Error level

$e$  Tooltip tracking error

$e_F$  Tooltip tracking error represented in the Frenet frame

$\hat{e}_F$  Tooltip tracking error represented in the compensated Frenet frame

$F$  Frenet frame

$\hat{F}$  Shifted Frenet frame

$f$  feedrate

$G$  Transfer function (general)

$G_c$  Transfer function of controller

$G_p$  Transfer function of plant

$h$  Displacement delay of Frenet frame

$h_F$  Displacement delay in the shifted Frenet frame

$I_a$  Motor current

$k$  Sampling counter

$k_a$  Amplifier gain
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_e$</td>
<td>Encoder gain</td>
</tr>
<tr>
<td>$k_j$</td>
<td>Delay counter of the $j$th impulse in the shaping sequence</td>
</tr>
<tr>
<td>$k_p, k_i, k_d$</td>
<td>Coefficients of PID controller</td>
</tr>
<tr>
<td>$k_{pos}, k_{vel}$</td>
<td>State feedback gains</td>
</tr>
<tr>
<td>$k_i$</td>
<td>Motor constant</td>
</tr>
<tr>
<td>$k_x, k_y$</td>
<td>Cross-coupling gains</td>
</tr>
<tr>
<td>$L$</td>
<td>Laplace transform operator</td>
</tr>
<tr>
<td>$N$</td>
<td>Normal distribution</td>
</tr>
<tr>
<td>$n$</td>
<td>Normal axis of Frenet frame</td>
</tr>
<tr>
<td>$OS$</td>
<td>Overshoot value</td>
</tr>
<tr>
<td>$P_a$</td>
<td>Actual tool pose represented in the workpiece frame</td>
</tr>
<tr>
<td>$P_d$</td>
<td>Desired tool pose represented in the workpiece frame</td>
</tr>
<tr>
<td>$P_n$</td>
<td>Nearest point (on the reference path) to the actual tooltip position</td>
</tr>
<tr>
<td>$R$</td>
<td>Radius</td>
</tr>
<tr>
<td>Rot()</td>
<td>Rotational operator</td>
</tr>
<tr>
<td>$s$</td>
<td>Laplace variable</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Period of residual vibrations</td>
</tr>
<tr>
<td>$T_f$</td>
<td>Friction torque</td>
</tr>
<tr>
<td>$T_m$</td>
<td>Motor torque</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sampling time</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$t$</td>
<td>Tangential axis of Frenet frame</td>
</tr>
<tr>
<td>$t_d$</td>
<td>Time delay of Frenet frame</td>
</tr>
</tbody>
</table>
\( t_j \) \hspace{1cm} \text{Time delay of the } j\text{th impulse in the shaping sequence}

\( V_{in} \) \hspace{1cm} \text{Input voltage}

\( V_{prv} \) \hspace{1cm} \text{Percentage of residual vibration}

\( W \) \hspace{1cm} \text{Weight function in cross-coupling controller}

\( X, Y, Z, A, C \) \hspace{1cm} \text{Axis drives of machine tool}

\( \delta \) \hspace{1cm} \text{Dirac impulse function}

\( \varepsilon \) \hspace{1cm} \text{Tooltip contouring error}

\( \hat{\varepsilon} \) \hspace{1cm} \text{Estimation of tooltip contouring error}

\( \zeta \) \hspace{1cm} \text{Damping ratio}

\( \mu \) \hspace{1cm} \text{Mean value of distribution}

\( \varphi_j \) \hspace{1cm} \text{Phase delay of the } j\text{th impulse in the shaping sequence}

\( \sigma \) \hspace{1cm} \text{Standard deviation of distribution}

\( \omega_a \) \hspace{1cm} \text{Actual angular velocity}

\( \omega_c \) \hspace{1cm} \text{Contouring frequency}

\( \omega_d \) \hspace{1cm} \text{Damped natural frequency}

\( \omega_n \) \hspace{1cm} \text{Natural frequency}

**Subscripts**

\( a, \text{act} \) \hspace{1cm} \text{actual}

\( \text{CL} \) \hspace{1cm} \text{closed loop}

\( \text{cmp} \) \hspace{1cm} \text{compensated}

\( \text{cmpErr} \) \hspace{1cm} \text{compensation error}

\( \text{cmpSig} \) \hspace{1cm} \text{compensation signal}
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cnt</td>
<td>controller</td>
</tr>
<tr>
<td>d</td>
<td>desired</td>
</tr>
<tr>
<td>ds</td>
<td>desired-shaped</td>
</tr>
<tr>
<td>dsc</td>
<td>desired-shaped-compensated</td>
</tr>
<tr>
<td>filt</td>
<td>filtered</td>
</tr>
<tr>
<td>j</td>
<td>impulse number</td>
</tr>
<tr>
<td>KF</td>
<td>Kalman filter</td>
</tr>
<tr>
<td>mdl</td>
<td>modeled</td>
</tr>
<tr>
<td>obs</td>
<td>observer</td>
</tr>
<tr>
<td>osc</td>
<td>oscillator</td>
</tr>
<tr>
<td>pp</td>
<td>pole placement</td>
</tr>
<tr>
<td>r, ref</td>
<td>reference</td>
</tr>
<tr>
<td>rand</td>
<td>random</td>
</tr>
<tr>
<td>rep</td>
<td>repetitive</td>
</tr>
<tr>
<td>rep-Nmdl</td>
<td>repetitive-not modeled</td>
</tr>
<tr>
<td>rs</td>
<td>reference-shaped</td>
</tr>
<tr>
<td>rsc</td>
<td>reference-shaped-compensated</td>
</tr>
<tr>
<td>s, shp</td>
<td>shaped</td>
</tr>
<tr>
<td>tb</td>
<td>table</td>
</tr>
<tr>
<td>w</td>
<td>represented in the workpiece frame</td>
</tr>
</tbody>
</table>
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMM</td>
<td>Coordinate Measuring Machine</td>
</tr>
<tr>
<td>CNC</td>
<td>Computer Numerical Control</td>
</tr>
<tr>
<td>EI</td>
<td>Extra Insensitive</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response</td>
</tr>
<tr>
<td>FRF</td>
<td>Frequency Response Function</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite Impulse Response</td>
</tr>
<tr>
<td>LPF</td>
<td>Low-pass Filter</td>
</tr>
<tr>
<td>MAL</td>
<td>Manufacturing Automation Laboratory at UBC</td>
</tr>
<tr>
<td>NURBS</td>
<td>Non-uniform rational B-spline</td>
</tr>
<tr>
<td>SI</td>
<td>Specified Insensitivity</td>
</tr>
<tr>
<td>ZPETC</td>
<td>Zero Phase Error Tracking Control</td>
</tr>
<tr>
<td>ZV</td>
<td>Zero-Vibration shaper</td>
</tr>
<tr>
<td>ZVD</td>
<td>Zero-Vibration-Derivative shaper</td>
</tr>
<tr>
<td>ZVDD</td>
<td>Zero-Vibration-Derivative-Derivative shaper</td>
</tr>
</tbody>
</table>
Acknowledgment

I would like to express my sincere gratitude to my research supervisor, Dr. Yusuf Altintas for his valuable instructions and unconditional support throughout my research and coursework. His impressive energy, patience and motivating attitude have been greatly encouraging to me, not only in my research and studies but also in my personal life.

I would also like to thank my friends at the Manufacturing Automation Laboratory for their support and making my M.A.Sc studies an enjoyable experience. I should particularly thank my friend Alex Yuen for his great help in preparation of the experimental setups.

Last but not least, I am grateful to my beloved family for their non-stop support, especially their patience regarding my long absence from home to do my graduate studies abroad. I would like to dedicate this work to my parents, who never stopped loving me and supporting me.
Chapter 1. Introduction

1 Introduction

Machining is widely used in different metal-working industries owing to the achievable quality and mechanical properties of the finished parts. However, machining requires removing a great portion of the original material to achieve the final part. For example, in order to manufacture a turbine impeller, more than 90% of the original block has to be machined away. Since the final cost of the part is directly related to the machining cycle time, increasing the productivity of the operation is crucial in industry. This motivation has led to a great deal of research in the field of high speed machining.

Increasing the speed of the machining operation involves different aspects related to the process and the machine. Owing to the advancement in the field of spindle design, the achievable torque and spindle speeds has significantly increased over the past two decades, facilitating the use of higher feedrates and depth of cut in order to increase the material removal rate. In addition, there has been a great deal of work focusing on optimizing the process to achieve higher productivity while avoiding chatter instability.

Another aspect of high speed machining is that the motion control system of the machine tool must be able to provide fast but accurate relative motions between the cutting tool and the workpiece. The control system must be able to maintain the required accuracy even in the presence of uncertainties, e.g. varying dynamic, as well as disturbance such as cutting forces. In order to have a reliable motion system, first of all, it is necessary for the machine to have well-designed and well-manufactured hardware; for example, the ballscrew drives which move the machine table must be relatively stiff with minimum backlash and assembly errors. On the software side, the CNC system which controls the axis motions must be able to provide smooth commands to the servo drives and maintain the accuracy even at the presence of uncertainties and disturbance. A great deal of effort has been put towards designing robust high bandwidth controllers to serve this purpose.

One of the main challenging problems in motion control of high speed machine tools is that the required rapid motions involve high acceleration and deceleration of the drives, resulting in excitation of the structural modes of the machine tool. These inertial-
induced vibrations leave unwanted surface marks on the part and result in poor surface quality. In order to move the axis drives smoothly and without vibration, the desired toolpath must be interpolated and parameterized with smooth curves such as B-splines or NURBS such that it does not cause large feedrate fluctuations. The motion trajectories must be realized using high order continuous feedrate profiles such as trapezoidal or cubic acceleration profiles. However, even if the trajectories are generated to move the tool smoothly along the path, it does not necessarily result in the smooth motion of the individual axes.

Figure 1.1 demonstrates a schematic frequency-domain analysis of feed drive vibrations. Assume that the axis has a low damping structural mode at a certain frequency, which appears as a large amplitude resonance in the Frequency Response Function (FRF) of the drive. The frequency spectrum of the input can be obtained by taking Fast Fourier Transform (FFT) of the command (or disturbance). The output response is also obtained by multiplying the FFT of the command and the FRF of the system, followed by taking Inverse Fast Fourier Transform (IFFT). However, if the input (command or disturbance) has significant harmonics in the vicinity of the system resonance mode, when it is multiplied by the FRF of the drive, it results in a large spike in the frequency spectrum of the output, which appears as dramatic vibrations in time domain.

In order to avoid vibrations of the feed drive, either the system resonance mode must be damped out, or the harmonics of the input around the resonance frequency must be removed. Figure 1.2 illustrates these two main approaches; in the first approach, which is called active damping, vibrations are measured in real time, e.g. by means of accelerometers, and a suppressing signal is calculated by a damping network. This approach can be effective in suppressing both the inertial-induced vibrations as well as the vibrations caused by disturbance such as cutting forces. However, active damping method needs additional sensors (accelerometer), and also alters the stability and dynamic of the closed loop system. Therefore, it may not be very trivial to integrate the damping network to the controller. In addition, this technique normally results in very active actuation signals.
with relatively large high frequency contents, which can saturate the drives and impose additional noise on the measurement devices, and thus degrade the control performance.

\[ X(s) = G(s)U(s) \]
\[ X(j\omega) = G(j\omega)U(j\omega) \]

**Figure 1.1:** Schematic frequency-domain analysis of feed drive vibrations.

Another approach, which is the main focus of this thesis, is that knowing the vibration characteristic of the system, the commanded trajectory can be modified ahead of time to avoid inertial-induced vibrations of the drive. Unlike active damping method, trajectory modification does not require any additional hardware, and it also results in much smoother control signal with less high-frequency harmonics. The main limitation of command modification method is that since it is an offline feedforward technique, it cannot reduce the vibrations caused by disturbance. However, machine vibrations in finishing operations are mostly dominated by the command-induced inertial vibrations, which makes it reasonable to employ trajectory modification techniques ahead of time.
Figure 1.2: Illustration of active damping and command shaping for vibration reduction in machine tool feed drives.

There have been several techniques proposed for performing command modifications. This thesis presents the investigation of input shaping technique, which can optimally remove the unwanted harmonics of the command, and avoid structural vibrations. In this method, the original axis commands are “shaped” by convolving them with a sequence of impulses called “input shaper”. The amplitudes and time locations of the impulses are designed based on the frequency and damping of the structural modes of the drives; the key characteristic of an input shaper is that it cancels its own vibrations when applied to the corresponding mechanical system. Such a sequence of impulses can be used as a shaping function in the sense that any arbitrary command if convolved with this shaper, the resultant shaped command will not cause any vibrations, either.

However, a challenging problem with input shaping method is that it distorts the trajectory, meaning that the shaped reference toolpath has deviations from the original desired one. The shaping distortion adds up to the geometric contour errors originated from the limited bandwidth of servo drives, resulting in violation of the required tolerance. This
thesis proposes a pre-compensation technique which estimates the contour errors originated from both input shaping distortion and machine dynamics.

The chapters of the thesis are organized as follows: Chapter 2 reviews the related research on command shaping techniques published in the literature. The theory of input shaping technique and the corresponding mathematical derivations are presented in Chapter 3, followed by the proposed method for pre-compensation of contour errors in Chapter 4. Experimental verifications are provided in Chapter 5, and the conclusions and possible future directions are presented in Chapter 6.


2 Literature Review

2.1 Overview

Due to the demand for higher productivity in industry, high speed machining has been studied extensively in the literature. Basically, demanding higher accelerations and velocities from the feed drives results in two main problems; first, the required axis motions contain high frequency harmonics, which can excite the structural modes of the machine and cause dramatic vibrations. The second issue is that due to the limited bandwidth of the servo drives, the actual response of the machine at high feedrates exhibits relatively large deviations from the desired toolpath.

This chapter briefly reviews the achieved advancement and the remaining challenges in motion control of high speed machine tools. Section 2.2 presents several techniques to avoid or suppress the vibrations in machine tool feed drives. Section 2.3 particularly focuses on the literature which employs input shaping to avoid structural vibrations. Basic approaches for improving contouring accuracy are presented in Section 2.4 followed by the conclusion of the chapter in Section 2.5.

2.2 Vibration Reduction in Machine Tool Feed Drives

Structural vibration in flexible systems and particularly in machine tool feed drives can be reduced by two main approaches, namely suppression and avoidance. In vibration suppression techniques, also known as active damping, a sensor is used to measure the vibrations in real time, and a suppressing signal is added to the control output to damp out the detected vibrations. In vibration avoidance approach, on the other hand, excitation of the structural modes is avoided through smooth trajectory generation and command modification techniques.

In order to implement active damping techniques, the flexible dynamic of the plant must be considered in the control law. There has been great deal of research focusing on
prediction and modeling of the flexibilities in machine tool feed drives. For example, Allotta et al. [1] as well as Okwudire and Altintas [31] used finite element methods to estimate the structural modes of the drives considering the stiffness of the lead screw and joint dynamics. One of the earliest work in implementation of active damping was carried out by Chen and Tlusty [9]; they used accelerometric feedback to suppress the drive vibrations, and also used feedforward controller to achieve better tracking performance. Altintas and Okwudire [5] enhanced the dynamic stiffness of a direct-driven feed drive using sliding mode controllers with disturbance recovery. Kamalzadeh et al. [22] achieved improved stiffness in ball screw drives by using a robust sliding mode controller and compensating the elastic deformations and lead errors. In order to consider the varying position-dependent stiffness in ballscrew drives, Hanafizadegan and Nagamune [17] designed a switching gain-scheduled controller which can maintain the performance despite the varying dynamic along the ballscrew.

Another approach in reducing vibrations in machine tool feed drives is to avoid excitation of the structural modes by using smooth trajectories. There has been great deal of research in this field resulting in jerk limited [13] and jerk-continuous [2] feedrate profiles. Optimal control techniques have been also used to generate commands for flexible systems [21]. In these techniques, a penalty index such as integral squared error is minimized such that the “optimal” trajectory causes minimum vibrations. However, there are several issues associated with optimal control techniques. First of all, the computation load is extensive as each motion of the system requires re-computation of the control system. Secondly, the penalty index, e.g. integral squared root of tracking error, cannot be exclusively associated with the vibrations as it includes the lumped error originated from various effects.

Alternatively, the inverse dynamics of the flexible plant can be used to reduce the command-induced vibrations [7]. In this case, the inverted model of the plant is used to find the required input trajectory which gives the desired output. However, this technique has also certain drawbacks. For example, it requires a detailed model which can reliably
describe the dynamic behavior of the system; the success of the method is very sensitive to mismatch between the modeled and actual parameters of the system.

In addition, there are several techniques proposed in the literature which generate the required trajectory by adding harmonics of a continuous template function. For example, Meckle and Seering [28, 29] proposed construction of the input command from ramped sinusoids functions; the harmonics which have high energy at resonance frequencies of the system are excluded to avoid structural vibrations. Aspinwall [8] also proposed a similar approach which generates the input trajectory by adding a series of sine waves, and the coefficients are chosen to minimize the frequency content of the input at the natural frequencies of the system.

Notch filters are also widely used in filtering the input commands to avoid excitation of flexible systems. However, the main issue with these real time (causal) filters is that they distort the phase of the signal, which becomes more problematic by increasing the order of analog filters or the length of digital filters.

Singer and Seering [43] proposed a technique called “input shaping” which can be applied on any arbitrary trajectory to optimally remove the unwanted harmonics of the command. They compared input shapers with classical Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) lowpass/bandstop filters, and showed that input shapers significantly outperform classical filters in terms of vibration reduction, incurred delay and simplicity [44]. Since input shaping has been employed in this thesis, it is reviewed in more details in the next section.

2.3 Input Shaping of Reference Trajectory

The earliest form of input shaping was the use of Posicast control proposed by O.J.M. Smith [53] in late 1950’s. In this method, a step command is broken into two smaller sub-commands, one of which is shifted in time such that the vibration caused by the first step is cancelled by that of the second one. However, this technique is extremely sensitive to model mismatch, and the vibration reduction degrades rapidly with the deviation of the modeled natural frequency from the actual one. Specially, in calculation of
the magnitudes and time delay of the second step, it is assumed that the system behaves linearly; any nonlinearity appears as model mismatch and dramatically degrades the performance of the technique. As a result, input shaping method did not get into real applications until late 1980’s when Singer and Seering [43] restudied the approach and made it possible to design input shapers with adjustable robustness. In their method, they forced the derivative of the residual vibrations to zero so that a wider range of tolerance against model mismatch can be achieved. This idea for improvement of robustness rapidly re-gained attention of many research groups, and over the past two decades, a great deal of work has been published on employing input shaping in various applications such as robotics, spacecrafts, cranes, and even in more accurate systems such as Coordinate Measuring Machines (CMM).

Soon after publishing the first idea of imposing derivative constraints [43], researchers developed other methods to achieve even higher robustness. For example, Singhose et al. [48, 49] and Park et al. [35] showed that further robustness can be achieved by relaxing the zero vibration constraint in designing the shaper. For instance, if 5% residual vibration is tolerated in the design, then the resulting shaper, which is called Extra-Insensitive (EI), can maintain its performance at a wider range of frequencies. Singhose et al. [50, 47] proposed Specified-Insensitivity (SI) shapers, in which the parameters can be designed to achieve desirable level of robustness over any specified range of frequencies. Pao et al. [32, 33] showed that the behavior of the shaper can be further customized by weighting the importance of the frequencies within the suppression range.

If the natural frequency of the system varies too much in different operating conditions, e.g. by more than 40%, then the required robust shaper might add significant delay to the command. If the change in the natural frequency is deterministic and predictable by the model, for instance, varying natural frequency along the ballscrew, then feedforward techniques can be used to gain-schedule the shaper based on the predicted conditions; this is possible as the reference axis positions are known ahead of time. However, if the change in the natural frequency of the system is too difficult to model, adaptive techniques which update the shaper parameters in real time using the vibration
data can be employed. This approach results in faster rise time since the original non-robust shaper is designed with only few impulses, and the robustness is achieved by real time tuning of the parameters.

Adaptation of the shaper parameters can be accomplished by either indirect methods, which first identify the vibration frequencies and then update the shaper, or by using direct adaptation law, which relates the shaper parameters to the sensor output. Tzes et al. [56] estimated the frequency of vibrations using frequency-domain identification technique, and then updated the delay time of the impulses in the shaper. However, the main challenge associated with this approach is that frequency-domain identification can be often computationally extensive and thus not practical for some real time applications. Another approach for indirect adaptation of the shaper is the use of a known relationship between the lower and higher resonance frequencies of the system; time-domain identification of high frequency modes can be accomplished in much shorter time compared to the low frequency vibrations. The identified higher modes are then used to theoretically estimate the lower modes, which are slower but often more dominant [23].

On the other hand, direct adaptation methods adjust the shaper parameters directly from the system output without identifying the frequencies. Rhim [40] and Park [34] showed that this approach can result in better convergence of the shaper provided that the adaptation law can effectively relate the system output to the shaper parameters. However, the problem with adaptation methods, regardless of direct or indirect approach, is that the measurement noise can be interpreted as vibrations, and thus, resulting in wrong tuning of the shaper parameters. Rhim and Book [39] studied the noise effect and investigated how to optimize the solution of time-domain adaptation in the presence of measurement noise.

Input shaping can be easily implemented for avoidance of multi-mode systems as well, although in most cases the vibrations caused by the first mode is much more dominant. The easiest way to design a shaper which avoids multiple frequencies is to design a separate shaper for each mode, and then convolve them all together. In this case, the length of the total shaper is the summation of the individual ones. Hyde and Seering [19, 18] showed that shorter multi-mode shapers can be achieved if the constraints on
system frequencies are all stacked together and solved simultaneously, although it requires solving more complicated set of equations compared to the convolution method. Singhose and Crain [46] thoroughly studied the tradeoff between the convolution method and the simultaneous solution to investigate how much improvement can be achieved relative to the complexity of finding the solution. In addition to multi-mode systems, input shaping can be implemented on multi-input systems as well [26].

Singhose has conducted a thorough review [45] on the advancement in input shaping research since it was first introduced in late 1950’s. Over the past two decades, input shaping has been implemented on many systems in different applications such as robotics, spacecrafts, and cranes. It has been even implemented on systems which have relatively high frequency and low amplitude vibrations. For example, Jones and Ulsoy [20], and Singhose et al. [52] employed input shaping to improve the accuracy and repeatability of Coordinate Measuring Machines (CMM). In addition, there has been some research on implementation of input shaping on XY stages [51] and Cartesian machine tools [36]; however, most of them try to avoid excitation of an auxiliary oscillator instead of the real structural modes of the machine (originated from the ballscrews or column). For example, Pelaez et al. [36] mounted a two degree of freedom (bending around X and Y axes) oscillator to show the effectiveness of input shaping in reducing the exaggerated unshaped vibrations. This thesis, on the other hand, starts with similar exaggerated cases but goes all the way to avoiding the real structural modes of a commercial machining center.

Although input shaping can effectively avoid structural vibrations of the machine, it might add considerable delay to the command. This delay results in toolpath distortion and significant increase in geometric contour errors [51, 36]. Pelaez [36] proposed that the distortion can be minimized if input shaping is applied on the velocity commands instead of the position commands. However, this method is not very practical as it needs to calculate the velocities after the interpolator, and re-integrate the shaped velocity values to obtain the shaped position commands. This thesis, on the other hand, proposes a pre-compensation technique which improves the contouring performance by estimating the
distortion ahead of time and correcting the shaped commands before sending them to the servo drives. The available methods for improving the contouring performance in machine tool feed drives are reviewed in the next section.

2.4 Improving Contouring Performance in Multi-axis Machines

The accuracy in following the desired path is a crucial criterion in evaluating the performance of a multi-axis machine tool. Pitschow [37] showed that it is very crucial for machine tools to have high bandwidth servo drives so that they can respond rapidly to sharp changes in the input; otherwise, large geometric contour errors will be incurred at these points.

One obvious approach in decreasing contour errors is to improve the tracking performance of individual axes of the machine. There has been great deal of work dedicated to increasing the bandwidth of the servo drives and compensating the friction. A pioneer work in this area is a feed forward technique called Zero Phase Error Tracking Control (ZPETC) proposed by Tomizuka [54]. This technique applies the inverse of the stable dynamic of the closed loop system on the input to compensate for the axis dynamic. However, the performance of ZPETC rapidly degrades as the modeled parameters of the system deviate from the actual ones. Erkorkmaz and Altintas [14] proposed an unbiased least square technique for the accurate identification of the system parameters. In addition, the robustness of the closed loop system has been improved significantly by implementing sliding mode controller [3] and disturbance observers [15].

For the cases when the plant dynamic changes considerably in different conditions, adaptive control schemes have been proposed. For example, Tsao and Tomizuka [55] extended their ZPETC to include adaptation capability so that it can maintain high performance even at the presence of model mismatch. Similarly, Pritschow and Kosiedowski [38] proposed adaptive feed forward controller for the cases when the plant exhibits slowly varying dynamics.

Another approach in improving the accuracy of the finished part is to directly control and compensate the resultant contour errors (instead of individual axis tracking
This technique was first proposed by Koren [24] in the form of Cross Coupling Controller illustrated in Figure 2.1; \( K_x \) and \( K_y \) are the cross coupling gains which relate the axis tracking errors to total contouring error. These gains were first defined for linear toolpaths [24], but later Koren and Lo [25] extended the algorithm to have variable cross coupling gains so that it can work for circular and parabolic toolpaths as well. Chen et al. [11] considered higher order terms in approximation of cross coupling gains so that the technique can be used for any free-form planar toolpath. Later, they extended their method and proposed cross coupling position command shaping [10], which approximates the contour error for free-form three dimensional toolpaths and corrects the command in real time. They estimated the contour error by calculating the component of tracking error normal to the average velocity vector. A similar work was proposed by Altintas and Sencer [6] which approximates the tooltip and orientation contour error in five-axis toolpaths using shifted Frenet frame, and employs sliding mode controller to move the axes with the corrected commands.

**Figure 2.1:** Basic cross coupling contour error control proposed by Koren [24].

Contour error control techniques cited above require approximation of the error in real time; however, since there is no analytical solution for calculation of contour errors, these techniques are not very practical for industrial machine tools. This thesis, on the other hand, proposes another approach which predicts the resulting contour errors ahead of time.
using the dynamic model of the machine, and pre-compensates the errors before they occur. An earlier form of this approach was proposed by Lo and Hsiao [27] for repeated machining processes; they machined one part and measured the resulting contour error using a Coordinate Measuring Machine (CMM); the measured error was then used to compensate the error by offsetting the original command. Any contour error estimation technique available in the literature can be used to calculate the contour errors from the predicted axis responses. In this thesis, the method proposed by Altintas and Sencer [6] which approximates the error using shifted Frenet frame is adopted; however, since the proposed pre-compensation technique is accomplished off-line ahead of time, even iterative methods such as the one proposed by Erkorkmaz et al. [16] can be used for further improved accuracy.

2.5 Conclusion

A brief review on the command shaping research available in the literature was presented in this chapter. Numerous techniques have been proposed to smoothly interpolate a desired toolpath using B-splines and NURBS, and generate smooth trajectories using high-order continuous feedrate profiles. These techniques result in smooth commands with low level of high-frequency content. However, in applications where the machine has some low damping resonance modes, even these smooth trajectories might excite the structural modes and cause vibrations.

Several command shaping techniques have been proposed to avoid vibrations by filtering the unwanted harmonics of the command. However, most of these methods add considerable delay to the input trajectory, which results in distortion of the toolpath. In order to reduce the resulting contour errors, researchers have implemented real time contour error control techniques which estimate the error and correct the command in real time. This thesis, on the other hand, proposes a pre-compensation technique which predicts and compensates the error ahead of time before it occurs.
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

3 Input Shaping of Reference Commands for Vibration Avoidance

3.1 Introduction

Due to the flexibilities originated from the mechanical parts such as couplings and lead screw, machine tool feed drives exhibit structural resonance modes at certain frequencies. These modes can get excited by both the input command and the disturbance, which then result in undesired vibrations. While disturbance-induced vibrations have to be dealt with in real time, e.g. by using active damping and disturbance rejection techniques, command-induced vibration may be avoided ahead of time. This chapter presents a filtering technique called input shaping, which effectively blocks unwanted harmonics of the command.

The chapter is organized as follows: Section 3.2 explains the technique of generating self-cancelling vibrations, which is the basis of input shaping method. Section 3.3 describes how a sequence of impulses with self-cancelling properties can be used as a command shaper to block unwanted harmonics of any arbitrary trajectory. Having the frequency and damping of the structural modes of the system, the procedure presented in Section 3.4 can be used to design different types of shapers with different characteristics. Since the shapers must be implemented on digital computers, Section 3.5 investigates the effect of discretization and the incurred degradation in vibration reduction. Section 3.6 provides a rough comparison between classical frequency-domain filters and input shapers in terms of robustness and vibration reduction. Investigations and contributions presented in this chapter are summarized in Section 3.7.

3.2 Vibration Cancellation Technique

Consider a flexible system with the natural frequency of $\omega_n$ and damping ratio of $\zeta$.

The dynamic of the system can be represented by a second order transfer function:
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

\[ G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \]  

(3.1)

If an impulse with the magnitude of \( A_1 \) is applied to this system at time \( t_1 \), the flexible system will vibrate at its damped natural frequency \( \omega_d = \omega_n \sqrt{1 - \zeta^2} \). The response to such an impulse can be found using inverse Laplace transform:

\[ \text{input: } R_1(t) = \text{Imp}(A_1, t_1) = A_1 \delta(t - t_1) \rightarrow R_1(s) = A_1 e^{-s t_1} \]

\[ \text{response: } X_1(t) = L^{-1}\{G(s)R_1(s)\} = L^{-1}\left\{\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} A_1 e^{-s t_1}\right\} \]

\[ = \frac{A_1 \omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_d(t-t_1)} \sin \omega_d(t-t_1) = B_1(t) \sin(\omega_d t + \varphi_1) \]  

(3.2)

where

\[ B_1(t) = \frac{A_1 \omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_d(t-t_1)} , \quad \varphi_1 = -\omega_d t_1 \]

(3.3)

If a second impulse with the amplitude of \( A_2 \) is applied to the system at time \( t_2 \), the system will similarly respond as:

\[ X_2(t) = B_2(t) \sin(\omega_d t + \varphi_2) \]  

(3.4)

where \( B_2(t) \) and \( \varphi_2 \) are similarly defined in Eq. (3.3). Assuming the dynamic of system to be linear, if both impulses are applied to the flexible system, the total response can be represented as:

\[ X_{\text{tot}}(t) = X_1(t) + X_2(t) = A_{\text{tot}}(t) \sin(\omega_d t + \psi) , \quad t \geq t_2 \]  

(3.5)

where \( A_{\text{tot}}(t) \) and \( \psi \) are calculated using trigonometric relations:
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

\[ X_{tot}(t) = B_1 \sin(\omega_d t + \varphi_1) + B_2 \sin(\omega_d t + \varphi_2) \]
\[ = B_1 \sin \omega_d t \cos \varphi_1 + B_1 \cos \omega_d t \sin \varphi_1 \]
\[ + B_2 \sin \omega_d t \cos \varphi_2 + B_2 \cos \omega_d t \sin \varphi_2 \]
\[ = (B_1 \cos \varphi_1 + B_2 \cos \varphi_2) \sin \omega_d t + (B_1 \sin \varphi_1 + B_2 \sin \varphi_2) \cos \omega_d t \]
\[ = \sqrt{C_1^2 + C_2^2} \left( \frac{C_1}{\sqrt{C_1^2 + C_2^2}} \sin \omega_d t + \frac{C_2}{\sqrt{C_1^2 + C_2^2}} \cos \omega_d t \right) \]
\[ = \sqrt{C_1^2 + C_2^2} \sin(\omega_d t + \psi) \]

Therefore,
\[ \psi = \tan^{-1} \left( \frac{C_2}{C_1} \right) = \tan^{-1} \left( \frac{B_1 \sin \varphi_1 + B_2 \sin \varphi_2}{B_1 \cos \varphi_1 + B_2 \cos \varphi_2} \right) \]
\[ A_{tot} = \sqrt{C_1^2 + C_2^2} = \sqrt{(B_1 \cos \varphi_1 + B_2 \cos \varphi_2)^2 + (B_1 \sin \varphi_1 + B_2 \sin \varphi_2)^2} \]

where \( B_j(t) \) and \( \varphi_j \) (\( j = 1, 2 \)) are defined in Eq. (3.3). Figure 3.1 demonstrates a graphical proof for Eq. (3.6) using vector diagram [36]. In a more general form, when there are more than two impulses, Eq. (3.7) can be written as:

\[ \psi(\omega_n, \zeta) = \tan^{-1} \left( \frac{S(\omega_n, \zeta, t)}{C(\omega_n, \zeta, t)} \right) \]
\[ A_{tot}(\omega_n, \zeta, t) = \frac{\omega_n}{\sqrt{1 - \zeta^2}} \sqrt{C(\omega_n, \zeta, t)^2 + S(\omega_n, \zeta, t)^2} \]

where

\[ C(\omega_n, \zeta, t) = \sum_{j=1}^{m} A_j e^{-\zeta \omega_n (t - t_j)} \cos(\omega_d t_j) \]
\[ S(\omega_n, \zeta, t) = \sum_{j=1}^{m} A_j e^{-\zeta \omega_n (t - t_j)} \sin(\omega_d t_j) \]

(3.9)
Figure 3.1: Vector diagram of total vibration caused by a sequence of two impulses [36].

As derived in Eq. (3.6), the response of the flexible system after applying the second impulse can be represented as a decaying sine wave oscillating at the damped natural frequency $\omega_d$. If the damping ratio of the resonance mode is small, which is usually the case in machine tool feed drives, the induced residual vibrations can last for a long time. However, as can be seen in Eq. (3.3), the amplitude of the residual vibrations depends on not only the damping ratio but also the magnitude and time location of the two impulses; ideally, if the second impulse is given at the right time with the right amplitude, the vibration induced by the first impulse can be completely cancelled by that of the second impulse; in that case, the system will exhibit absolutely zero residual vibrations.

Figure 3.2 demonstrates the described self-canceling technique for a sequence of two impulses. Without losing generality, it can be assumed that the first impulse is given at time zero, i.e. $t_1 = 0$. Such a sequence of impulses which leaves zero or very small residual vibration is called an “Input Shaper” or a “Shaping Function”.
3.3 Reference Shaping Principle

Once a sequence of self-canceling impulses is found for a flexible system, it can be used as a shaping function; that is, any arbitrary initial reference command if convolved with the sequence in the time domain, the so called “Shaped Command” will not cause any residual vibration either [43]. Basically input shaping acts as a filter and effectively blocks those harmonics of the command which coincide with the resonance mode of the system.

Figure 3.3 illustrates the principle of reference shaping using a sequence of impulses. If \( r(t) \) is the initial unshaped command and \( \text{InShp}(t) = A_1 \delta(t) + A_2 \delta(t-t_2) \) is the input shaper designed for the system mode, the shaped command \( r_{\text{shp}} \) can be generated as:

\[
r_{\text{shp}}(t) = r_{\text{shp,1}}(t) + r_{\text{shp,2}}(t) = A_1 r(t) + A_2 r(t-t_2) 1(t-t_2)
\]

(3.10)

where \( 1(t-t_2) = 1 \) at \( t \geq t_2 \) and \( 1(t-t_2) = 0 \) at \( t < t_2 \). Basically, input shaper breaks the initial command into smaller sub-commands, and shifts them in time such that the vibration caused by each sub-command is cancelled by that of the others. In order to keep the rigid body motion of the shaped command the same as the original one, the shaper must fully recover the original trajectory; in order to fulfil this requirement, the summation of impulse magnitudes in an input shaper has to be one:
\[ \sum_{i=1}^{m} A_i = 1 \]  

(3.11)

It should be noted that the time length of the shaped command is equal to the length of the original trajectory plus the length of the shaper. This time delay, if significant, can cause distortion of the trajectory in multi axis toolpaths, which is thoroughly discussed in Chapter 4.

**Figure 3.3:** Reference shaping using a sequence of impulses.

### 3.4 Input Shaper Design

Since an input shaper is a sequence of impulses, designing the shaper means finding the proper magnitudes and time locations of these impulses. The number of impulses is dictated by the constraints imposed on the performance and robustness of the shaper. As a general rule, the larger number of impulses a shaper has, the more robust it is to the mismatch in the modeled frequency and damping ratio. However, longer time delay and more significant toolpath distortions will be incurred as the number of impulses increases.

In order to investigate the effectiveness of input shaping in vibration reduction, and also for comparing different types of shapers, it is more convenient to have a dimensionless value as the criterion for vibrations; based on Eq. (3.2), the amplitude of system vibrations in response to a unit impulse at time zero can be obtained as:
This value is used to normalize the amplitude of residual vibrations. The “percentage of residual vibration”, $V_{prv}(\omega_n, \zeta)$, caused by a sequence of $m$ impulses is defined as the vibration amplitude immediately after applying the last impulse divided by the value in Eq.(3.12), that is:

\[ V_{prv}(\omega_n, \zeta, t_m) = \frac{A_{tot}(t = t_m)}{B_1(t = 0)} = \sqrt{C(\omega_n, \zeta, t_m)^2 + S(\omega_n, \zeta, t_m)^2} \]  

(3.13)

where $C(\omega_n, \zeta, t_m)$ and $S(\omega_n, \zeta, t_m)$ are defined in Eq. (3.9). This value will be used widely as a performance index for shaper design and comparison.

### 3.4.1 Zero-Vibration (ZV) Shaper

Zero-Vibration (ZV) shaper is the simplest type of input shapers and can be designed using only two impulses [43]. Without losing generality, the first impulse is assumed to be applied at time zero, that is, $t_1 = 0$. In order to design a ZV shaper, three unknowns must be determined: the magnitudes $A_1$, $A_2$, and also the time location of the second impulse, $t_2$; therefore, three equations are needed to determine these unknowns. The first constraint, which must hold for any type of shaper to fully recover the rigid body motion, is the unity summation:

\[ \sum_{i=1}^{m} A_i = A_1 + A_2 = 1 \]  

(3.14)

The next set of equations can be achieved from the zero-vibration constraint; in order to get theoretically zero residual vibrations after applying the last impulse, the percentage of residual vibration defined in Eq. (3.13) is forced to zero, i.e. $V_{prv} = 0$:

\[ V_{prv}(\omega_n, \zeta, t_m) = \sqrt{C(\omega_n, \zeta, t_m)^2 + S(\omega_n, \zeta, t_m)^2} = 0 \]  

(3.15)
where \( C(\omega_n, \zeta, t_m) \) and \( S(\omega_n, \zeta, t_m) \) are defined in Eq. (3.9). This constraint gives two equations:

\[
V_{pr^r}(\omega_n, \zeta, t_2) = 0 \quad \Rightarrow \quad \begin{cases} 
C(\omega_n, \zeta, t_2) = 0 \\
S(\omega_n, \zeta, t_2) = 0 
\end{cases}
\] (3.16)

Hence,

\[
C(\omega_n, \zeta, t_2) = 0 \quad \Rightarrow \quad A_1 e^{-\zeta \omega_d t_2} + A_2 e^{-\zeta \omega_d t_2} \cos(\omega_d t_2) = 0
\] (3.17)

\[
S(\omega_n, \zeta, t_2) = 0 \quad \Rightarrow \quad A_2 e^{-\zeta \omega_d t_2} \sin(\omega_d t_2) = 0
\] (3.18)

In order to have non-zero solution for \( A_2 \) in Eq. (3.18), the sine function must be zero:

\[
\sin(\omega_d t_2) = 0 \quad \Rightarrow \quad \omega_d t_2 = k\pi \quad k = 1, 2, ...
\] (3.19)

The shortest shaper is achieved when \( k = 1 \):

\[
t_2 = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} = 0.5T_d
\] (3.20)

where \( T_d \) is the period of vibrations to be avoided. Substituting \( t_2 \) from Eq. (3.20) into Eq. (3.17) along with Eq. (3.14) provides the required equations for determining the impulse magnitudes:

\[
A_1 + A_2 = 1
\]

\[
A_1 - A_2 e^{\zeta \omega_d / \sqrt{1 - \zeta^2}} = 0
\] (3.21)

Finally, the magnitude and time locations of the two impulses in a ZV shaper can be summarized as:
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

\[
\begin{bmatrix}
A_j(t)
\end{bmatrix} = \begin{bmatrix}
\frac{1}{1+K_j}, \frac{K_j}{1+K_j}, K_j \\
1+K_j, 0, 0.5T_d
\end{bmatrix}, \quad j = 1, 2
\]  

\[K = \exp\left(-\frac{\zeta_{\pi}}{\sqrt{1-\zeta^2}}\right)\]  

(3.22)

3.4.2 Robustness of Zero-Vibration (ZV) Shaper

As calculated in Eq. (3.20), the time length of ZV shaper is one half period of the vibrations, which makes it the shortest possible input shaper. However, this type of shaper lacks robustness against mismatch between the modeled and actual parameters of the system. The percentage of residual vibration \(V_{prv}\), which was defined in Eq. (3.13), is used as an index to investigate the robustness of a shaper.

Assume \(\omega_{n-mdl}\) and \(\zeta_{mdl}\) are the modeled natural frequency and damping ratio used in designing the shaper. If the actual parameters \(\omega_{n-act}\) and \(\zeta_{act}\) deviate from the modeled ones, vibrations will not cancel out completely, and thus the system will exhibit residual vibrations after applying the last impulse. The \(V_{prv}\) calculated from the actual parameters shows the level of residual vibrations, and thus, can be used as an index of robustness:

\[V_{prv}(\omega_{n-act}, \zeta_{act}, A_1-mdl, A_2-mdl, t_{2-mdl}) = \sqrt{C(\omega_{n-act}, \zeta_{act}, t_{2-mdl})^2 + S(\omega_{n-act}, \zeta_{act}, t_{2-mdl})^2}\]  

(3.23)

where \(C(\omega_{n-act}, \zeta_{act}, t_{2-mdl})\) and \(S(\omega_{n-act}, \zeta_{act}, t_{2-mdl})\) are calculated from Eq. (3.9). It is common to define \(V_{prv} < 5\%\) as the criterion for the robustness, which means how much mismatch the system can tolerate before the residual vibration becomes larger than 5\% of the unshaped case.

Figure 3.4 demonstrates the robustness of ZV shaper against the mismatch in natural frequency and damping ratio. In order to have a clearer view, the cross section of the plot for the case when one of frequency or damping deviates and the other one is kept constant has been also plotted. It can be seen that ZV shaper is very sensitive especially to mismatch.
in natural frequency. For example, in order to meet the criterion $V_{prv} < 5\%$, only less than 4% mismatch in frequency is tolerated. However, in real ballscrew feed drives, as the table moves along the lead screw, the natural frequency can shift by 10% or even more. It can be also observed that input shaping is not too sensitive to mismatch in damping ratio; even with 30% deviation, the criterion $V_{prv} < 5\%$ is still satisfied. This can be explained from Eq. (3.22), in which damping ratio has exponential effect on the magnitude of the impulses while natural frequency is inversely proportional to the time location of the impulses. Since input shaping works based on vibration cancellation, it is more sensitive to timing, and thus, natural frequency.

**Figure 3.4**: Sensitivity of ZV shaper against mismatch in frequency and damping ratio.

\[
\left( r_\zeta = \frac{\zeta_{\text{act}}}{\zeta_{\text{mdl}}} \quad , \quad r_\omega = \frac{\omega_{n-\text{act}}}{\omega_{n-\text{mdl}}} \right)
\]
3.4.3 Zero-Vibration and Derivative (ZVD) Shaper

As discussed in the previous section, Zero-vibration (ZV) shaper, which is designed based on only constraining the percentage of residual vibration to zero at the modeled frequency and damping ratio, is not robust enough to be used in real feed drive applications. As Figure 3.4 shows, this sensitivity problem of ZV shaper is due to the sharp change in the sensitivity curve at the modeled natural frequency. Therefore, the robustness can be significantly improved by smoothing the sensitivity curve, for instance, by imposing derivative constraints at the mentioned point [43].

Zero-vibration and Derivative (ZVD) shaper is obtained by adding a new constraint which ensures zero derivative of residual vibration with respect to natural frequency at the modeled frequency, i.e.

$$\frac{\partial V_{prv}}{\partial \omega_n} \Big|_{\omega_n=\omega_{n-mdl}} = 0 \quad \text{(3.24)}$$

It should be noted that the zero-vibration constraint in Eq. (3.15) is prerequisite for the subsequent derivative constraints; therefore, $$C(\omega_{n-mdl}, \zeta) = 0$$ and $$S(\omega_{n-mdl}, \zeta) = 0$$ is already presumed for the following derivations. Since ZVD shaper has been widely used in experimental verifications conducted in this thesis, the mathematical derivations for this type of shaper are presented here in details.

Using the definition of percentage of residual vibrations $$V_{prv}$$ from Eq. (3.13), the derivative constraint in Eq. (3.24) can be expanded as:

$$\frac{\partial V_{prv}}{\partial \omega_n} = \frac{\partial}{\partial \omega_n} \left( \sqrt{C(\omega_n, \zeta)^2 + S(\omega_n, \zeta)^2} \right) = \frac{2C \frac{\partial C}{\partial \omega_n} + 2S \frac{\partial S}{\partial \omega_n}}{2\sqrt{C^2 + S^2}} \quad \text{(3.25)}$$

Direct substitution of the zero-vibration constraints from Eq. (3.15) into Eq. (3.25) results in indeterminate form of “zero-divided-by-zero”. However, since $$C(\omega_{n-mdl}, \zeta) = 0$$ and $$S(\omega_{n-mdl}, \zeta) = 0$$, the root $$\omega_n = \omega_{n-mdl}$$ can be cancelled out from the numerator and
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

denominator. Therefore, the derivative in Eq. (3.25) after evaluating the indeterminate form will be zero if:

\[
\frac{\partial C(\omega_n, \zeta)}{\partial \omega_n} \bigg|_{\omega_n = \omega_{n-\text{adj}}} = 0
\]  
(3.26)

\[
\frac{\partial S(\omega_n, \zeta)}{\partial \omega_n} \bigg|_{\omega_n = \omega_{n-\text{adj}}} = 0
\]  
(3.27)

The derivative constraint brings two additional equations, and thus, requires two additional unknowns. These two unknowns can be provided by adding another impulse to the sequence, which brings the magnitude and time location as the unknowns; therefore, ZVD shaper is formed by three impulses. In order to solve for the unknowns, Eqs. (3.26) and (3.27) are expanded:

\[
\frac{\partial C(\omega_n, \zeta)}{\partial \omega_n} = \frac{\partial}{\partial \omega_n} \left[ \sum_{j=1}^{3} A_j e^{-\zeta t_j (t_j - t_f)} \cos(\omega_n \sqrt{1 - \zeta^2 t_j}) \right]
\]

\[
= \sum_{j=1}^{3} \left[ -\zeta (t_j - t_f) A_j e^{-\zeta \omega_n (t_j - t_f)} \cos(\omega_n \sqrt{1 - \zeta^2 t_j}) \right.
\]

\[
- \sqrt{1 - \zeta^2 t_j} A_j e^{-\zeta \omega_n (t_j - t_f)} \sin(\omega_n \sqrt{1 - \zeta^2 t_j}) \]

\[
= \sum_{j=1}^{3} \left[ -\zeta t_j A_j e^{-\zeta \omega_n (t_j - t_f)} \cos(\omega_n \sqrt{1 - \zeta^2 t_j}) \right. \\
+ \zeta t_j A_j e^{-\zeta \omega_n (t_j - t_f)} \cos(\omega_n \sqrt{1 - \zeta^2 t_j}) \\
- \sqrt{1 - \zeta^2 t_j} A_j e^{-\zeta \omega_n (t_j - t_f)} \sin(\omega_n \sqrt{1 - \zeta^2 t_j}) \]  
(3.28)
Based on the zero-vibration constraint, the terms $C$ and $S$ in Eqs. (3.28) and (3.29) are zero. If the terms labeled as $C'$ and $S'$ which have appeared in both derivative equations are zero at the modeled natural frequency, the zero-derivative equations in Eq. (3.26) and Eq. (3.27) are satisfied simultaneously. Therefore, the final constraints for designing ZVD shaper can be summarized as:

\[
\sum_{j=1}^{3} A_j = 1
\]

(3.30)

\[
C(\omega_n, \zeta, t_j) = \sum_{j=1}^{3} A_j e^{-\zeta_0 (t_j - t_j)} \cos(\omega_n \sqrt{1 - \zeta^2 t_j}) = 0
\]

(3.31)

\[
S(\omega_n, \zeta, t_j) = \sum_{j=1}^{3} A_j e^{-\zeta_0 (t_j - t_j)} \sin(\omega_n \sqrt{1 - \zeta^2 t_j}) = 0
\]

(3.32)

\[
C'(\omega_n, \zeta, t_j) = \sum_{j=1}^{3} t_j A_j e^{-\zeta_0 (t_j - t_j)} \cos(\omega_n \sqrt{1 - \zeta^2 t_j}) = 0
\]

(3.33)
The derivative constraints in Eq. (3.33) and Eq. (3.34) have been obtained based on improving the robustness against mismatch in natural frequency. However, since damping ratio in mechanical systems is often inconsistent and hard to identify, the shaper must be significantly robust against the mismatch in damping as well. This requires the derivative of the sensitivity curve with respect to damping ratio be zero at the modeled damping. Interestingly, the zero-derivative conditions obtained for natural frequency guarantees the zero-derivative with respect to damping ratio as well. This fact can be easily seen as:

\[
\frac{\partial C(\omega_n, \zeta)}{\partial \zeta} = \frac{\partial}{\partial \zeta} \left[ \sum_{j=1}^{3} A_j e^{-\zeta \omega_n(t_j-t_1)} \cos(\omega_n \sqrt{1-\zeta^2} t_j) \right] = 0
\]

\[
(3.35)
\]
\[
\frac{\partial S(\omega_n, \zeta)}{\partial \zeta} = \frac{\partial}{\partial \zeta} \left[ \sum_{j=1}^{3} A_j e^{-\zeta \omega_n (t_1 - t_j)} \sin(\omega_n \sqrt{1 - \zeta^2} t_j) \right] \\
= \sum_{j=1}^{3} \left[ -\omega_n (t_3 - t_j) A_j e^{-\zeta \omega_n (t_1 - t_j)} \sin(\omega_n \sqrt{1 - \zeta^2} t_j) \\
- \frac{2\zeta}{\sqrt{1 - \zeta^2}} \omega_n t_j A_j e^{-\zeta \omega_n (t_1 - t_j)} \cos(\omega_n \sqrt{1 - \zeta^2} t_j) \right] \\
= \sum_{j=1}^{3} \left[ -\omega_n t_3 A_j e^{-\zeta \omega_n (t_1 - t_j)} \sin(\omega_n \sqrt{1 - \zeta^2} t_j) \\
+ \omega_n t_j A_j e^{-\zeta \omega_n (t_1 - t_j)} \sin(\omega_n \sqrt{1 - \zeta^2} t_j) \\
- \frac{2\zeta}{\sqrt{1 - \zeta^2}} \omega_n t_j A_j e^{-\zeta \omega_n (t_1 - t_j)} \cos(\omega_n \sqrt{1 - \zeta^2} t_j) \right] \\
\tag{3.36}
\]

Therefore, the zero-derivative constraints in Eq. (3.33) and Eq. (3.34) simultaneously improve robustness against both frequency and damping.

It should be noted that if negative-amplitude impulses are allowed, mathematically there are infinite number of solutions for the set of constraints in Eqs. (3.30)-(3.34). However, from the practical point of view, input shapers with negative-amplitude impulses may result in more demanding commands, which can saturate the actuators and also excite the high frequency modes of the system. On the other hand, imposing positive amplitude constraint, i.e.:

\[A_j > 0 \quad , \quad j = 1, 2, \ldots, m\tag{3.37}\]

along with the unity summation constraint in Eq. (3.30) guarantees that the magnitude of all impulses in the shaper is smaller than unity:

\[\sum_{j=1}^{3} A_j = 1 \quad \Rightarrow \quad A_j < 1 \quad , \quad j = 1, 2, 3\tag{3.38}\]
Basically, the commands shaped with positive-amplitude shapers are always less aggressive and more stable compared to the unshaped commands; this means that if the original reference has been generated not to violate the drive limits, the shaped input will not violate them either.

The next constraint which can be added to solve the ZVD equations is the length of the shaper. As calculated in Eq. (3.20), the second impulse in a ZV shaper is applied at the half period of vibrations; similarly we can assume that the impulses in the ZVD shaper are also applied at half vibration periods, that is:

\[
t_j = (j-1)\frac{T_d}{2} = (j-1)\frac{\pi}{\omega_d} = (j-1)\frac{\pi}{\omega_n\sqrt{1-\zeta^2}}, \quad j = 1, 2, 3
\]

Therefore, the length of ZVD shaper is a full period of vibrations, which is reasonable for practical applications. Substituting the time location of the impulses from Eq. (3.39) into the defined constraints, the three required equations for calculating the magnitude of the three impulses in ZVD can be obtained from Eqs. (3.30), (3.31) and (3.33):

\[
\begin{align*}
t_1 &= 0 \\
t_2 &= 0.5T_d = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \\
t_3 &= T_d = \frac{2\pi}{\omega_n\sqrt{1-\zeta^2}}
\end{align*}
\]

After multiplying the last equation by \(\omega_n\sqrt{1-\zeta^2}/\pi\), the system of equations in Eq. (3.40) can be written in the matrix form:
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

\[
\begin{bmatrix}
1 & 1 & 1 \\
\frac{1}{e^{-\pi \zeta / \sqrt{1 - \zeta^2}}} & \frac{1}{-e^{-\pi \zeta / \sqrt{1 - \zeta^2}}} & \frac{1}{1} \\
0 & 0 & 2
\end{bmatrix}
\begin{bmatrix}
A_1 \\
A_2 \\
A_3
\end{bmatrix}
= \begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix}
\tag{3.41}
\]

Thus,

\[
\begin{bmatrix}
A_1 \\
A_2 \\
A_3
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 1 \\
\frac{1}{e^{-\pi \zeta / \sqrt{1 - \zeta^2}}} & \frac{1}{-e^{-\pi \zeta / \sqrt{1 - \zeta^2}}} & \frac{1}{1} \\
0 & 0 & 2
\end{bmatrix}^{-1}
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix}
(3.42)
\]

Finally, the Zero-Vibration and Derivative (ZVD) input shaper can be summarized as:

\[
\begin{bmatrix}
\{A\} \\
\{t\}_j
\end{bmatrix}
= \begin{bmatrix}
\left\{\frac{1}{K^2 + 2K + 1}\right\}_1, \\
\left\{\frac{2K}{K^2 + 2K + 1}\right\}_2, \\
\left\{\frac{K^2}{T_d}\right\}_3
\end{bmatrix}
\tag{3.43}
\]

where \(K = e^{-\pi \zeta / \sqrt{1 - \zeta^2}}\) and \(T_d = \frac{2\pi}{\omega_n \sqrt{1 - \zeta^2}}\).

### 3.4.4 Robustness of the Zero-Vibration and Derivative (ZVD) Shaper

Similar to the procedure explained for ZV shaper, the robustness of ZVD shaper can be investigated using the dimensionless value \(V_{prv}\) defined in Eq. (3.23). Figure 3.5 demonstrates the percentage of residual vibrations with ZVD shaper when the natural frequency and damping ratio deviate from the modeled ones. Comparing this sensitivity curve with that of the ZV shaper shown previously in Figure 3.4, it can be clearly observed that ZVD shaper is significantly more robust than ZV shaper. For example, in order to keep
the percentage of residual vibration below 5%, ZV shaper can tolerate only less than 4% mismatch in natural frequency $\omega_n$; ZVD shaper, on the other hand, can satisfy the criterion $V_{prev} < 5\%$ even with 16% deviation from the modeled natural frequency. This level of robustness is fairly sufficient for most feed drive applications. It can also be seen that ZVD shaper is extremely robust against uncertainty in damping ratio, which is very crucial as damping cannot be measured accurately and also changes significantly in different operating conditions.

**Figure 3.5:** Sensitivity of ZVD shaper against mismatch in frequency and damping ratio.

\[
\begin{align*}
( r_\zeta &= \frac{\zeta_{act}}{\zeta_{mdl}} , \quad r_\omega = \frac{\omega_{act}}{\omega_{mdl}} )
\end{align*}
\]
Figure 3.6 compares the sensitivity curves for ZV and ZVD shaper. It can be seen that the zero-derivative constraints has significantly improved the robustness of the shaper.

![Sensitivity of ZV and ZVD shapers](image)

**Figure 3.6:** Comparing the sensitivity of ZV and ZVD shapers against mismatch in frequency and damping ratio.

### 3.4.5 Further Improvement of Shaper Robustness

While ZVD shaper provides 15% robustness against mismatch in natural frequency, still it might not be enough for some applications, especially in systems with varying dynamics. Robustness of the shapers can be further improved by enforcing additional constraints or modifying the existing ones. For example, one effective method to improve the robustness is relaxing the zero-vibration constraint; that is, instead of forcing the
residual vibrations to be exactly zero at the modeled natural frequency and damping ratio, a small amount of vibration \( V_{prv-mdl} \neq 0 \) will be allowed. This relaxation of the constraints let the shaper keep the vibrations below the threshold for a larger range of mismatch. The shaper obtained by this method is called Extra-Insensitive (EI) input shaper [48]. It can be shown that Extra-Insensitive shaper has the same length as ZVD shaper, and can be obtained as:

\[
\left[ \begin{array}{c}
A \\
t
\end{array} \right]_j = \left[ \begin{array}{c}
\frac{1+V_{prv-mdl}}{4} \\
\frac{1-V_{prv-mdl}}{2} \\
0 \\
\end{array} \right]_1 + \left[ \begin{array}{c}
\frac{1+V_{prv-mdl}}{4} \\
0.5T_d \\
T_d \\
\end{array} \right]_2 + \left[ \begin{array}{c}
\frac{1+V_{prv-mdl}}{4} \\
0 \end{array} \right]_3 
\]

\( j = 1, 2, 3 \) (3.44)

where \( V_{prv-mdl} \) is the allowed percentage of residual vibration at the modeled natural frequency and damping ratio, and \( T_d \) is the period of vibrations. Figure 3.7 demonstrates the sensitivity curve for the EI shaper with the allowed \( V_{prv-mdl} = 5\% \) at the modeled frequency. It can be seen that without increasing the length of the shaper, the robustness has improved compared to the ZVD shaper.

Another method to further improve the robustness of the shaper is imposing higher order derivative constraints on the sensitivity curve. The equations for zero-derivative constraints were derived in Eqs. (3.33) and (3.34). Similarly, it can be shown that the constraint for the \( k \)th order derivative of the sensitivity curve for a shaper with \( m \) impulses can be represented as [42]:

\[
\frac{\partial^k V_{prv}}{\partial \omega_n^k} = 0 \rightarrow \sum_{j=1}^{m} \left( t_j \right)^k A_j e^{-\xi \omega_n (t_m-t_j)} \cos(\omega_n \sqrt{1-\xi^2} t_j) = 0 
\]

(3.45)

For example, ZVDD (Zero-Vibration-Derivative-Derivative) shaper which has zero vibration as well as zero first and second derivative at the modeled frequency can be obtained by stacking Eq. (3.45) and the constraints obtained for ZVD shaper in Eqs. (3.30)-(3.34):
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

The magnitudes can be solved by rearranging Eq. (3.46) into matrix form:

\[
\begin{bmatrix}
A_1 \\
A_2 \\
A_3 \\
A_4
\end{bmatrix}
= 
\begin{bmatrix}
e^{-\pi\zeta/\sqrt{1-\zeta^2}} & -e^{-2\pi\zeta/\sqrt{1-\zeta^2}} & e^{-\pi\zeta/\sqrt{1-\zeta^2}} & -1 \\
0 & -e^{-2\pi\zeta/\sqrt{1-\zeta^2}} & 2e^{-\pi\zeta/\sqrt{1-\zeta^2}} & -3 \\
0 & -e^{-2\pi\zeta/\sqrt{1-\zeta^2}} & 4e^{-\pi\zeta/\sqrt{1-\zeta^2}} & -9
\end{bmatrix}
\begin{bmatrix}
1 \\
3K \\
3K^2 \\
K^3
\end{bmatrix}
\]

Therefore, the four-impulse ZVDD shaper can be summarized as:

\[
\begin{bmatrix}
A_1 \\
A_2 \\
A_3 \\
A_4
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
K^3+3K^2+3K+1 \\
0.5T_d \\
1.5T_d
\end{bmatrix}, \quad j = 1, 2, 3, 4
\]

where \( K = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \) and \( T_d = 2\pi/\omega_n\sqrt{1-\zeta^2} \).

Figure 3.7 compares the sensitivity curve against mismatch in natural frequency for the four mentioned types of input shapers. Combining the different types of constraints introduced in the previous sections, a shaper with any desired level of robustness can be
achieved but at the expense of larger time delays. Considering the compromise between robustness and delay, the three-impulse ZVD and four-impulse ZVDD shapers are the most practical input shapers for real applications. If the dynamic of the system changes too much, it might be a better idea to use adaptive or gain-scheduling techniques, which can result in shorter shapers while maintaining the performance over a larger range of parameters. However, adaptive shapers can also involve other complexities such as stability issues and non-smooth transition between different states.

Figure 3.7: Comparing the sensitivity of ZV, ZVD, EI and ZVDD shapers against mismatch in frequency and damping ratio.

3.5 Discretization Effect in Digital Input Shaping

As derived in the previous section, the time location of impulses in an input shaper is determined by the period of vibrations, which can be any positive real number. However, in computer-controlled machines, the servo loop is run in digital mode at constant sampling intervals, and the discrete reference command is also generated only at this sampling
periods. Therefore, it usually happens that the calculated time for an impulse in the sequence lies between two sampling times. In this case, the impulse has to be shifted to the nearest sampling time counter, which might prevent complete cancellation of residual vibrations [42]. The same dimensionless value for the percentage of residual vibration $V_{prv}$ defined in Eq. (3.13) is used here to investigate the effect of discretization error.

Assume a ZV shaper is designed for a system with natural frequency of $\omega_n$. In order to simplify the mathematical derivations, the damping is assumed to be zero, which is not too unrealistic for real structural modes. Using Eq. (3.22), ZV shaper ($\zeta = 0$) can be represented as:

$$
\begin{bmatrix}
\{A_j\} \\
\{t_j\}
\end{bmatrix}
= \begin{bmatrix}
\{0.5\} \\
\{0\}
\end{bmatrix} \cdot \begin{bmatrix}
\{0.5\} \\
\{0.5T_d\}
\end{bmatrix}
$$

$$j = 1, 2 \quad (3.49)
$$

where $T_d = 2\pi / \omega_d$ is the period of vibrations. If this system is run at a sampling period of $\delta T$, the maximum discretization error in allocating the impulse time is $\delta T / 2$ and occurs when the second impulse lies exactly at the midpoint between two sampling times. In this case, the second impulse is shifted to the previous counter, that is:

$$
t_z = 0.5T_d - \frac{\delta T}{2} \quad (3.50)
$$

The percentage of residual vibration $V_{prv}$ caused by this error can be calculated from Eq. (3.13):

$$
V_{prv}(\omega_n, \zeta, t_z) = \sqrt{C(\omega_n, \zeta, t_z)^2 + S(\omega_n, \zeta, t_z)^2}
$$

$$
= \sqrt{\left(\sum_{j=1}^{2} A_j e^{-\zeta \omega_n (t-t_j)} \cos(\omega_d t_j)\right)^2 + \left(\sum_{j=1}^{2} A_j e^{-\zeta \omega_n (t-t_j)} \sin(\omega_d t_j)\right)^2}
$$

$$\begin{array}{c}
\zeta = 0 \\
A_1 = A_2 = 0.5 \\
t_z = \frac{T_d}{2} - \frac{\delta T}{2}
\end{array}
\rightarrow V_{prv} = \sqrt{\left[0.5 + 0.5 \cos\left(\omega_d \left(\frac{T_d}{2} - \frac{\delta T}{2}\right)\right)\right]^2 + \left[0.5 \sin\left(\omega_d \left(\frac{T_d}{2} - \frac{\delta T}{2}\right)\right)\right]^2}
$$

$$\quad (3.51)
$$
The expression for $V_{prv}$ in Eq. (3.51) can be simplified using trigonometric relations:

$$V_{prv} = 0.5 \sqrt{1 - 2 \cos \left( \frac{\omega_d \Delta T}{2} \right) + \cos^2 \left( \frac{\omega_d \Delta T}{2} \right)} + \sin^2 \left( \frac{\omega_d \Delta T}{2} \right)$$

$$= 0.5 \sqrt{2 - 2 \cos \left( \omega_d \frac{\Delta T}{2} \right) - \frac{1 - \cos \left( \omega_d \frac{\Delta T}{2} \right)}{2}} = \sin \left( \omega_d \frac{\Delta T}{4} \right)$$

(3.52)

Therefore, if $\omega_d \frac{\Delta T}{4} \ll 1$, the percentage of residual vibrations caused by discretization of a ZV shaper can be approximated as:

$$V_{prv} \approx \omega_d \frac{\Delta T}{4} = \frac{\pi \Delta T}{2 \Delta t}$$

(3.53)

Similarly, discretization effect on three-impulse ZVD shapers can be investigated as well. As shown in Figure 3.8, two extreme conditions are considered: 1) when the second impulse lies exactly at the middle point between two sampling times, 2) when the third impulse lies at the middle point.

**Figure 3.8:** Two possible extreme cases for the discretization of a ZVD shaper [42].

In the first case, the third impulse lies exactly on a sampling counter and can be applied without error; therefore, the resulting residual vibration in this case is similar to the case of ZV shaper derived in Eq. (3.53). In the second case, it is assumed that the second
impulse lies at $\delta T/4$ from a sampling time, which means that the third impulse lies at a midpoint with $\delta T/2$ discretization error (see Figure 3.8). The residual vibration $V_{prv}$ for this case can be calculated as:

$$V_{prv}(\omega, \zeta, t_m) = \sqrt{\sum_{j=1}^{3} A_{j} e^{-\zeta \omega_d (t-t_j)} \cos(\omega_d t_j)} + \sqrt{\sum_{j=1}^{3} A_{j} e^{-\zeta \omega_d (t-t_j)} \sin(\omega_d t_j)}$$

$$= 0.25 \sqrt{1 + 2 \cos \left(\pi - \omega_d \frac{\delta T}{4} \right) + \cos \left(2\pi - \omega_d \frac{\delta T}{2} \right) + \ldots}$$

$$= 0.25 \sqrt{2 \sin \left(\pi - \omega_d \frac{\delta T}{4} \right) + \sin \left(2\pi - \omega_d \frac{\delta T}{2} \right)^2}$$

The following trigonometric relations can be used to simplify the above expression:

$$\cos a \cos b + \sin a \sin b = \cos(a - b)$$

$$\cos(2a) = 2 \cos^2 a - 1$$

Therefore,

$$V_{prv} = 0.25 \sqrt{6 + 4 \cos \left(\pi - \omega_d \frac{\delta T}{4} \right) + 2 \cos \left(2\pi - \omega_d \frac{\delta T}{2} \right) + \ldots}$$

$$= 0.25 \sqrt{6 + 8 \cos \left(\pi - \omega_d \frac{\delta T}{4} \right) + 2 \cos \left(2\pi - \omega_d \frac{\delta T}{2} \right) + \ldots}$$

$$= 0.25 \sqrt{6 + 8 \cos \left(\pi - \omega_d \frac{\delta T}{4} \right) + 2 \cos \left(2\pi - \omega_d \frac{\delta T}{2} \right)}$$

$$= \cos^2 \left(\frac{\pi}{2} - \omega_d \frac{\delta T}{8} \right) = \sin^2 \left(\omega_d \frac{\delta T}{8} \right)$$
Therefore, if $\omega_d \frac{\delta T}{8} \ll 1$, the residual vibrations caused by discretization of a ZVD shaper when there is $\delta T/4$ error in applying the second impulse (and consequently $\delta T/2$ error in the third impulse) can be approximated as:

$$V_{prv} \approx \left( \omega_d \frac{\delta T}{8} \right)^2 = \left( \pi \frac{\delta T}{4T_d} \right)^2$$  \hspace{1cm} (3.57)

However, since $\delta T/T_d \ll 1$, the residual vibration in this case is much smaller than the first case calculated in Eq. (3.53). Therefore, the worst possible condition in discretization of a ZVD shaper is the same as ZV shaper, which can be calculated from Eq. (3.53).

In order to keep the residual vibration caused by discretization error below a certain level, the sampling time of the system must be small enough. Figure 3.9 provides the minimum sampling time to keep the discretization-induced residual vibration below 5%, that is $V_{prv} < 0.05$.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure3.9.png}
\caption{Minimum sampling time to keep the percentage of residual vibration caused by discretization of shaper below 5%.
}
\end{figure}
3.6 Comparing Input Shapers with Classical Frequency Domain Filters

As discussed in the previous sections, input shapers are basically time-domain filters which can effectively block unwanted harmonics of the command. Filtering of the input can be also accomplished using classical Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters designed in the frequency domain. In many commercial CNC machine tools, a low pass filter is applied on the input trajectory to avoid excitation of the machine. Lowpass filters can add significant delay to the command, which results in toolpath distortions and sluggish response of the machine.

In this section, it is shown that input shapers can significantly outperform frequency domain filters in terms of incurred delay and achievable robustness. The criterion for measuring the length of the filter is the 5% settling time of the filter response to a step input. In order to quantify the robustness, a simple second order system with a resonance mode at 1 Hz is used as the plant, and the filter is applied on a step input command. In order to have a fair comparison between different filters, the plant is assumed to have zero damping so that the performance of the filter is affected only by the mismatch in the natural frequency. The maximum amplitude of vibrations in the plant response after the filter length is considered as the residual vibrations. The robustness of the filter can be investigated by comparing the amplitude of residual vibrations for the cases when the actual natural frequency of the system is equal to the modeled one, i.e. 1 Hz, and also when it is 15% lower, which means 0.85 Hz.

As derived in Section 3.4, input shapers with any desired level of robustness can be achieved but at the expense of larger time delays. Based on the sensitivity curve shown in Figure 3.7, the three-impulse ZVD and four-impulse ZVDD shapers are the most practical choices as they have very good robustness while causing only $1.0 T_d$ and $1.5 T_d$ delay, respectively ($T_d$ is the period of vibrations). In order to compare input shapers with frequency domain filters, first the representation of input shapers in frequency domain will be derived. The mathematical derivations are presented for only three-impulse ZVD shaper; the four-impulse ZVDD follows the same procedure.
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

Assume a ZVD shaper with three impulses is designed to avoid excitation of a plant which has a natural frequency at \( \omega_n = 1 \text{Hz} \) with zero damping. As derived in Eq. (3.43), this ZVD shaper can be represented in time domain as:

\[
G_{ZVD}(t) = \sum_{j=1}^{3} A_j \delta(t-t_j) = 0.25 + 0.5 \delta(t-0.5T_d) + 0.25 \delta(t-T_d)
\]  
(3.58)

where \( T_d = 1 \text{s} \) is the vibration period. The transfer function of the shaper can be found by taking Laplace transform:

\[
G_{ZVD}(s) = \sum_{j=1}^{3} A_j e^{-st_j} = 0.25 + 0.5 e^{-0.5s} + 0.25 e^{-s}
\]  
(3.59)

Substituting \( s \) with \( j \omega \) in Eq. (3.59), the Frequency Response Function (FRF) of the shaper is then obtained:

\[
G_{ZVD}(j\omega) = \sum_{j=1}^{3} A_j e^{-j\omega t_j} = 0.25 + 0.5 e^{-0.5j\omega} + 0.25 e^{-j\omega}
\]

\[
= (0.25 + 0.5 \cos(0.5\omega) + 0.25 \cos(\omega)) - j (0.5 \sin(0.5\omega) + 0.25 \sin(\omega))
\]  
(3.60)

Therefore, the magnitude and phase of the shaper can be calculated as:

\[
|G_{ZVD}(j\omega)| = \sqrt{(0.25 + 0.5 \cos(0.5\omega) + 0.25 \cos(\omega))^2 + (0.5 \sin(0.5\omega) + 0.25 \sin(\omega))^2}
\]

\[
\angle G_{ZVD}(j\omega) = \tan^{-1}\left(\frac{-[0.5 \sin(0.5\omega) + 0.25 \sin(\omega)]}{0.25 + 0.5 \cos(0.5\omega) + 0.25 \cos(\omega)}\right)
\]  
(3.61)

Figure 3.10 demonstrates the bode plot of ZVD and ZVDD shapers. As can be seen in the figure, input shapers have notches at the modeled natural frequency \( \omega_n \) as well as the odd multiples of it, i.e. \((2k+1)\omega_n\), \( k = 1, 2, \ldots \).
Chapter 3. Input Shaping of Reference Commands for Vibration Avoidance

Figure 3.10: Frequency response function of ZVD and ZVDD shapers.

Figure 3.11 also shows the response of the described plant to a step input shaped by ZVD and ZVDD shapers. The solid-line is the system response when the actual natural frequency is exactly the same as the modeled one, that is \( \omega_n = 1 \text{Hz} \). The dash-line response, on the other hand, shows the response when the actual frequency is 15% below the modeled one. The figure clearly shows that input shapers result in vibration free response with satisfactory robustness. The FFT plots of the residual vibrations show that ZVDD shaper is much more robust than ZVD shaper. Also note that the shaped step response has much smaller transient overshoot, which means not only the residual vibration but also the transient deflection of the flexible part during the maneuver is much smaller in the shaped case.
Figure 3.11: System response to ZVD- and ZVDD-shaped step for the exact and mismatched (15%) models ($\omega_p = 1$ Hz, $\zeta = 0$)

In order to evaluate the performance of classical filters, a Butterworth lowpass filter with the passband at 0.7 Hz and stopband at 0.95 Hz with 30 dB attenuation, as well as a Butterworth bandstop filter with the passbands at 0.5 Hz and 1.5 Hz and stopbands at 0.95 Hz and 1.05 Hz with 30 dB attenuation have been designed using MATLAB Filter Design Toolbox. Note that designing filters involves compromise on length, robustness, simplicity and attenuation level of filters. In order to have a fair comparison, the filters are designed such that they have practically reasonable length and robustness. Figure 3.12 shows the frequency response plots of the designed lowpass and bandstop filters. Group delay function, which is defined as $d\phi/d\omega$ ($\phi$ is the phase) has been plotted in the same
figure. This value has dimension of time and corresponds to the delay caused by the filter. For example, for the designed lowpass and bandstop filters, the value of the group delay function is $4\, \text{s}$ and $2.5\, \text{s}$, respectively. This delay is much smaller in input shapers, e.g. $1\, \text{s}$ and $1.5\, \text{s}$ for ZVD and ZVDD shapers, respectively, as can be seen in Figure 3.11.

Figure 3.12: Frequency response of the designed lowpass and bandstop filters
Figure 3.13 shows the system response to a step input shaped by the designed lowpass and bandstop filters. In order to investigate the robustness, the response has been plotted for the exact and mismatched (15%) systems. The residual vibrations have been identified as the vibrations left after the maximum group delay \((d\varphi/d\omega)\) in Figure 3.12, which are 4s and 2.5s for the designed lowpass and bandstop filters, respectively.

**Figure 3.13**: System response to a step command shaped by lowpass and bandstop filters. (The mismatched model has 15% deviation from the modeled frequency)

Table 3.1 summarizes the performance comparison for the investigated shapers and filters. Rise time and maximum overshoot of the system response have been also presented in the table to evaluate the transient performance and deflection of the flexible part during the maneuver. These results show that input shapers significantly outperform classical filters in both transient and residual states.
<table>
<thead>
<tr>
<th>Performance Index</th>
<th>Exact</th>
<th>15% Mismatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>delay (filter length) [s]</td>
<td>ZVD 1.0</td>
<td>ZVD 1.0</td>
</tr>
<tr>
<td>maximum percentage of residual vibration [%]</td>
<td>ZVD 0.0</td>
<td>ZVD 0.0</td>
</tr>
<tr>
<td>settling time (5%) [s]</td>
<td>ZVD 1.0</td>
<td>ZVD 1.0</td>
</tr>
<tr>
<td>rise time [s]</td>
<td>ZVD 1.0</td>
<td>ZVD 1.0</td>
</tr>
<tr>
<td>percentage overshoot [%]</td>
<td>ZVD 0.0</td>
<td>ZVD 0.0</td>
</tr>
</tbody>
</table>

**3.7 Summary**

In this chapter, the physical interpretation and mathematical derivations of input shaping technique was presented. It was shown that depending on the constraints and design criteria, different types of shapers with different characteristics and limitations can be designed. Since there is always mismatch between the modeled and actual system, robustness of input shapers was thoroughly studied in this chapter. It was shown that mathematically, any desired level of robustness is achievable, but at the expense of longer incurred delay.

Furthermore, the performance of input shaping technique was compared against classical Butterworth lowpass and band-stop filters; it was shown that input shapers significantly outperform these frequency-domain filters in terms of vibration reduction and robustness. However, the main problem with input shaping is that it adds delay to the command, which results in toolpath distortion and large geometric contour errors. This effect is investigated and compensated for in the next chapter.
4 Improving Accuracy by Pre-compensation of Contour Errors

4.1 Introduction

Due to the limited bandwidth of servo drives, each axis of a machine tool exhibits tracking errors in following the reference command. In multi-axis trajectories, these errors result in deviation of the actual toolpath from the desired one. In addition, the delay caused by input shaping causes path distortion and adds up to the total contour error. This chapter proposes a feed forward pre-compensation technique which predicts and avoids these contour errors through command shaping techniques.

The chapter is organized as follows; Section 4.2 provides the mathematical definition of contour error. Section 4.3 presents an approximation method which uses shifted Frenet frame to estimate the total contour errors. Trajectory distortion caused by input shaping is studied in Section 4.4, followed by the proposed pre-compensation technique in Section 4.5. The effectiveness of pre-compensation is investigated in Section 4.6 by conducting a statistical analysis on the observed contour errors in the actual response. Finally, Section 4.7 summarizes the investigations and contribution presented in the chapter.

4.2 Definition of Contour Error

As shown in Figure 4.1, tracking error vector is defined as the difference between the desired and actual position of the tooltip, that is:

\[
\mathbf{e} = [e_x, e_y, e_z] = \mathbf{p}_d - \mathbf{p}_a = [x_d - x_a, y_d - y_a, z_d - z_a]
\]  

(4.1)

where \( \mathbf{p}_d = [x_d, y_d, z_d] \) and \( \mathbf{p}_a = [x_a, y_a, z_a] \) are the desired and actual position of the tool respectively. However, tracking error is a function of time and does not directly show the geometric accuracy and tolerance of the machined part. Contour error, on the other hand, is defined as the deviation of the actual position from the reference toolpath, and as shown in
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

Figure 4.1, it can be roughly interpreted as the component of tracking error normal to the reference toolpath. Note that unlike tracking error, contour error is a spatial variable and can be measured by CMM (Coordinate Measuring Machine) without any knowledge about the motion control specifications of the machine.

![Figure 4.1: Illustration of tracking error and contouring error.](image)

As illustrated in Figure 4.1, contour error vector can be represented as:

\[ \mathbf{\varepsilon} = [\varepsilon_x, \varepsilon_y, \varepsilon_z] = \mathbf{p}_n - \mathbf{p}_a = [x_n - x_a, y_n - y_a, z_n - z_a] \]  \hspace{1cm} (4.2)

where \( \mathbf{p}_n = [x_n, y_n, z_n] \) is a point on the reference toolpath which is closest to the actual position \( \mathbf{p}_a \); however, there is no straightforward method to analytically find this nearest point \( \mathbf{p}_n \). Iterative method proposed by Erkorkmaz and Altintas [16] can be used to search for the nearest point \( \mathbf{p}_n \) and calculate the exact value of contour error. However, iterative techniques may require a lot of computation to converge, which makes them impractical to be implemented in real time control laws. This thesis adopts an approximation method proposed by Altintas and Sencer [6] which directly estimates the contour errors using shifted Frenet frame. The basic theory of this method is explained in the following section.
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

4.3 Contour Error Estimation using Frenet Frame

Figure 4.2 illustrates the reference and actual position of the tooltip in following a desired path. Frenet frame [12], which is shown by $\mathbf{F}$ in the figure, is a coordinate frame with its origin coincident on the reference position $p_d$ and its axes defined by the mutually orthogonal unit tangent $t$, normal $n$, and bi-normal $b$ vectors as:

\[
t = \left[ t_x, t_y, t_z \right]^T = \frac{\dot{p_d}}{\|\dot{p_d}\|} = \frac{1}{\sqrt{\dot{x}_d^2 + \dot{y}_d^2 + \dot{z}_d^2}} \left[ \dot{x}_d, \dot{y}_d, \dot{z}_d \right]^T , \quad \dot{p_d} = \frac{d p_d}{dt} \tag{4.3}
\]

\[
n = \left[ n_x, n_y, n_z \right]^T = \frac{\ddot{p_d}}{\|\dot{p_d}\|} = \frac{1}{\sqrt{\dot{x}_d^2 + \dot{y}_d^2 + \dot{z}_d^2}} \left[ \ddot{x}_d, \ddot{y}_d, \ddot{z}_d \right]^T , \quad \ddot{p_d} = \frac{d \dot{p_d}}{dt} \tag{4.4}
\]

\[
b = t \times n
\tag{4.5}
\]

and the Frenet frame is formed by:

\[
\mathbf{F} = [t, n, b] = \begin{bmatrix}
t_x & n_x & b_x \\
t_y & n_y & b_y \\
t_z & n_z & b_z
\end{bmatrix}
\tag{4.6}
\]

Figure 4.2: Using Frenet frame in estimation of contour errors [41].
As Figure 4.2 illustrates, Frenet frame can be used directly to estimate the contour error \( \hat{\epsilon} \). The tracking error vector \( e = [e_x, e_y, e_z] \) (in the workpiece frame) can be represented in the Frenet frame using the Frenet frame transformation:

\[
e_r = [e_x, e_y, e_z]^T = F^{-1} \cdot e = F^T \cdot e
\] (4.7)

The contour error can be roughly estimated as the component of the tracking error normal to the tangential axis \( t \) of the frame [6], which can be obtained by subtracting the tangential component of the tracking error:

\[
\hat{\epsilon} = e_r - e_r \cdot t = (e_t + e_n \cdot n + e_b \cdot b) - e_t = e_n \cdot n + e_b \cdot b
\] (4.8)

However, as the curvature of the toolpath and also the contouring speed increases, the estimation error becomes relatively large. As shown in Figure 4.3, the approximation can be significantly improved by shifting the Frenet frame to a point which is closer to the actual \( p_n \).

**Figure 4.3:** Using shifted Frenet frame in estimation of the contour errors [41].

The delay or lead time between the current reference position \( p_d \) and nearest point \( p_n \), can be estimated as:
where \( e_t \) is the tracking error component in the tangential direction \( t \). The shifted Frenet frame is obtained by applying the time shift as:

\[
\hat{F} = F(t - t_d)
\]  

(4.10)

However, it should be noted that this time shift only corrects the axis directions of the frame, but the origin of the shifted frame is still coinciding with the current reference position \( p_d \). In order to shift the position of the frame to the delayed reference position, the displacement delay vector \( h \) is defined as:

\[
h = p_d(t) - p_d(t - t_d)
\]  

(4.11)

which can be transformed to the shifted Frenet frame \( \hat{F} \) as:

\[
h_v = \hat{F}^T h
\]  

(4.12)

Only the normal and bi-normal components of the displacement delay \( h_v \) are used in the contour error estimation; the tangential component is excluded as:

\[
\hat{h}_v = D h_v
\]  

(4.13)

where \( D = \text{diag}(0,1,1) \). Therefore, the tracking error represented in the compensated Frenet frame can be written as:

\[
\hat{e}_v = \hat{F}^T e - \hat{h}_v
\]  

(4.14)

Finally, the corrected contour error estimation is obtained from the normal and bi-normal components of the tracking error in Eq. (4.14) represented in the compensated Frenet frame:
\[ \hat{e}_F = [\hat{e}_x, \hat{e}_y, \hat{e}_z]^T \approx \hat{e}_n + \hat{e}_b = e_n \hat{n}_F + e_b \hat{b}_F \quad (4.15) \]

### 4.4 Trajectory Distortion caused by Input Shaping

Input shapers can effectively block unwanted harmonics of the command but at the expense of adding delay to the trajectory, meaning that the shaped command has always a time shift relative to the original unshaped input. This delay might not be problematic in single axis systems, but in multi-axis machines, even if all axes are shaped with the same input shaper, the time delay results in path distortion. In other words, the machine can follow the shaped input without vibration, but the command itself has inaccuracy with respect to the original desired path. Note that this deviation is different from the error caused by the limited bandwidth of the servo drives.

The discrepancy between the shaped and unshaped toolpath depends on the feedrate, curvature of the path, and the structural frequency which must be avoided. For example, assume a circle with the radius of 40 mm is to be contoured at 80 mm/s by a machine which has a structural resonance mode at 3 Hz on its X-axis and a mode at 5 Hz on the Y-axis, both with damping of 0.1 (For the sake of demonstration, the structural modes have been chosen to be very low so that the distortion will be significant). A ZVD shaper is designed for each axis using Eq. (3.43):

\begin{align*}
\text{X-axis:} & \quad \omega_n = 3 \text{ Hz} \\
& \quad \zeta = 0.1 \\
& \quad T_d = 2\pi / \omega_n \sqrt{1 - \zeta^2} = 0.335 \text{ s} \\
& \quad K = e^{-\pi \zeta / \sqrt{1 - \zeta^2}} = 0.73 \\
& \quad [A_j] = \begin{bmatrix}
1 \\
K^2 + 2K + 1
\end{bmatrix}_1, \begin{bmatrix}
2K \\
K^2 + 2K + 1
\end{bmatrix}_2, \begin{bmatrix}
K^2 \\
K^2 + 2K + 1
\end{bmatrix}_3 \\
0 \\
0.5T_d \\
T_d
\end{bmatrix}_j, j=1,2,3 \\
& = \begin{bmatrix}
0.3344 \\
0.4877 \\
0.1778
\end{bmatrix}_1, \begin{bmatrix}
0.4877 \\
0.1675 \\
0.335
\end{bmatrix}_2, \begin{bmatrix}
0.1778 \\
0.335
\end{bmatrix}_3
\end{align*}
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

Y-axis: \[ \omega_n = 5 \text{Hz} \] \[ \zeta = 0.1 \] \[ T_d = \frac{2\pi}{\omega_n \sqrt{1 - \zeta^2}} = 0.201 \text{s} \]

\[ K = e^{-\zeta \sqrt{\frac{1}{2} - \zeta^2}} = 0.73 \]

\[
\begin{bmatrix}
A_j \\
T_j
\end{bmatrix}_j = \begin{bmatrix}
\begin{bmatrix}
1 \\
K^2 + 2K + 1 \\
0
\end{bmatrix}_1, & \begin{bmatrix}
2K \\
K^2 + 2K + 1 \\
0.5T_d
\end{bmatrix}_2, & \begin{bmatrix}
K^2 \\
K^2 + 2K + 1 \\
T_d
\end{bmatrix}_3
\end{bmatrix}
\]

\[ j = 1, 2, 3 \]

(4.17)

The two individual shapers designed in Eqs. (4.16) and (4.17) are then convolved together to obtain the total input shaper, which is then used for both axes:

Total Shaper: \[ A_j^{\text{total}} = \text{conv}(A_j^x, A_j^y) \]

\[
\begin{bmatrix}
A_j \\
T_j
\end{bmatrix}_j = \begin{bmatrix}
0.11 & 0.1631 & 0.1631 & 0.0595 & 0.2379 & 0.0595 \\
0 & 0.1005 & 0.1675 & 0.2011 & 0.2680 & 0.3350 \\
0.0867 & 0.0867 & 0.0316 & 0.3686 & 0.4355 & 0.5361
\end{bmatrix}
\]

(4.18)

Figure 4.4 shows the shaped and unshaped axis commands in contouring the described circular path. It can be seen that the shaped trajectories have a time delay with respect to the original commands, which have resulted in a distorted toolpath. Note that the assumptions here are very exaggerated for clear observations. In realistic cases, this distortion is normally in the order of few microns. For the case when the original path is a circle, the path distortion can be explicitly calculated as:

\[
\text{Contour Error: } R - \sqrt{x_{\text{shp}}^2 + y_{\text{shp}}^2}
\]

(4.19)

where \( x_{\text{shp}} \) and \( y_{\text{shp}} \) are the coordinates of each point on the shaped toolpath, and \( R \) is the radius of the original circular path. Positive error implies that the shaped point lies inside the circle whereas the negative error shows the points outside.
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

Figure 4.4: Trajectory distortion caused by input shaping ($R = 40\,\text{mm}$, $f = 80\,\text{mm/s}$).

Figure 4.5 shows the path distortion evaluated by calculating the contour error value from Eq. (4.19). It can be seen that the error is positive, which based on the definition in Eq. (4.19) implies that all points of the shaped toolpath lie inside the circle.

Figure 4.5: Contour error caused by input shaping for a circular path ($R = 40\,\text{mm}$, $f = 80\,\text{mm/s}$).
In order to investigate how feedrate and resonance frequency affect the incurred path distortion, the same circular path ($R=40\text{mm}$) has been contoured at different conditions. The maximum distortion error, which is considered as the comparison index for each case, is shown in Figure 4.6. The trend of the plot shows that higher feedrates and lower resonance modes result in larger distortions. This is because lower frequencies require longer shapers, and higher feedrates causes larger displacement delay between the original and shaped reference positions.

**Figure 4.6:** Maximum contour error caused by input shaping for a circular path contoured at different feedrates while avoiding different structural frequencies ($R=40\text{mm}$).

In addition to the contouring feedrate and resonance frequency, the curvature of the path also affects the shaping distortions. In order to include all of these three effects into investigations, it is convenient to define a dimensionless number Distortion Index $D_{ind}$ as:

$$D_{ind} = \frac{f}{(f_{X\text{-filt}} + f_{Y\text{-filt}}) \cdot R}$$

(4.20)
where \( f [\text{mm/s}] \) is the feedrate, \( f_{X_{\text{filt}}} \) and \( f_{Y_{\text{filt}}} \) are the structural resonance frequencies in [Hz], and \( R \) is the radius of the circle in [mm]. This dimensionless number can be interpreted as how large the distance delay between the original and shaped reference is relative to the curvature of the path. Eq. (4.20) shows that distortion index is directly related to the feedrate and inversely related to the resonance frequency and path radius. Figure 4.7 plots the maximum distortion in contouring a circular path with different distortion indices \( D_{\text{ind}} \). It can be observed that distortion error increases as the distortion index gets larger.

![Maximum Contour Error in Different Distortion Indices](image)

**Figure 4.7**: Maximum contour error caused by input shaping in different distortion indices.

As a more realistic example, assume that the mentioned circular path with radius \( R = 40 \text{mm} \) is contoured at 80 mm/s, and the trajectory is shaped with ZVD shapers to avoid the 30 Hz mode of the X-axis and 50 Hz mode of the Y-axis; in this case, the distortion index is:

\[
D_{\text{ind}} = \frac{f}{(f_{X_{\text{filt}}} + f_{Y_{\text{filt}}}) \cdot R} = \frac{80}{(30 + 50) \cdot 40} = 0.025 \quad (4.21)
\]
and from the plot in Figure 4.7, the maximum contour error is found to be 12 micron. If the same path is contoured with the same condition but at 200 m/s feedrate, the distortion index increases to 0.0625, and from Figure 4.7, the maximum contour error also increases to 75 micron.

It is worth reminding that this error is related to the command itself; the contour error caused by the limited bandwidth of servo drives will add up to this error after machining. Depending on the application, if the structural mode is relatively high and the feedrate is fairly low, shaping distortion may be ignored; however, in high speed CNC machine tools which require high accuracy and speed at the same time, the error has to be corrected. The proposed technique for contour error compensation is discussed in details in the next section.

4.5 Pre-compensation of Contour Errors

The actual response of the machine always deviates from the reference path in contour machining. The contouring error consists of some repetitive part, which can be caused by the machine dynamics and friction, as well as some random non-repetitive portion caused by disturbance or inconsistency in friction. In order to reduce the total geometric error in following a toolpath, one suggested method is to estimate the contour error in real time using the measured axis positions, and then add a compensating signal either to the reference or the control output [10, 6]. Although this technique is capable of reducing both the repetitive and non-repetitive parts of the contour error, it is computationally extensive, which might not be practical to be implemented on real time CNC machine tools.

On the other hand, owing to the available feed drive models which can closely simulate the actual response of the machine, the repetitive part of contour error can be predicted and corrected beforehand. This is especially practical, as accurate and vibration-free motion of the machine is needed mostly for the finishing operations, in which the chip load and thus cutting forces are very small. As a result, the machine response in finishing
operations is mostly dominated by deterministic behavior of the drives, and is not considerably affected by random disturbance-induced errors.

Figure 4.8 illustrates the basic idea of pre-compensation of contour errors. Assume that the perfect circle shown in dash-dot line is to be contoured by a machine, which has relatively low-bandwidth drives. The conventional way to generate the axis commands is to interpolate the perfect circle with a certain trajectory generation technique, e.g. cubic acceleration, and then map the path on the corresponding axes. However, due to the limited bandwidth of the drives, the chosen feedrate must be relatively low; otherwise, the response dramatically deviates from the desired path and violates the tolerance requirements. For example, if the commanded trajectories are generated only based on the interpolated circle, the actual response of the machine might be similar to the dash-line path in Figure 4.8, although it is very exaggerated for the sake of demonstration. The common approach to alleviate this problem is either using an aggressive controller, which might result in saturation of the drives, or using relatively low feedrates, which leads to long cycle time.

Figure 4.8: Illustration of pre-compensation of contour errors.
However, if a fairly good model of the feed drive dynamics is available, which is usually the case in machine tools, this deviation can be predicted beforehand. In this case, we can simply use our knowledge of the machine to modify the command such that considering all the limitations, the actual response is closer to the circle. For example, considering the uncompensated response shown by the dash-line in Figure 4.8, we might be able to modify the command to get a compensated toolpath such that when the machine is following this compensated command, it deviates due to its dynamic; however, since the command is modified based on this behavior, the actual response of the machine will be much closer to the perfect circle even though the input trajectory itself is not a perfect circle. The implementation of pre-compensation technique is explained in the following section.

4.5.1 Pre-compensation of Path Distortion and Machine-dependent Contour Errors

The pre-compensation idea presented above can be used to simultaneously compensate for the toolpath distortion caused by input shaping as well as the machine-dependent contour errors. The procedure is explained here for a five axis tilting-table XYZAC, as shown in Figure 4.9, but the method can be used for other machine configurations as well using the proper kinematic transformations.

Figure 4.10 demonstrates the overall algorithm for the pre-compensation of contour errors. Assume that the original toolpath specified by the user has been interpolated using an arbitrary type of trajectory generation method, e.g. cubic acceleration profile. This gives the desired tool pose in the workpiece frame, denoted as $[Px_d, Py_d, Pz_d, \theta a_d, \theta c_d]_w$, where $Px_d, Py_d, Pz_d$ are the tooltip coordinates, and $\theta a_d, \theta c_d$ are the primary ($a$) and secondary ($c$) angles describing the tool orientation; the subscript $w$ indicates that all of these coordinates are described in the workpiece frame. Note that the interpolated trajectory is in the form of discrete points, i.e. $[Px_d(k), Py_d(k), Pz_d(k), \theta a_d(k), \theta c_d(k)]_w$, where $k$ is the sampling counter; however, for the sake of simplicity of equations and diagrams, the argument ($k$) is not written in the following derivations.
Figure 4.9: Schematic configuration of a five axis machine tool with tilting table (XYZAC) [41].

The corresponding reference axis commands \([x_r, y_r, z_r, a_r, c_r]\) which result in the desired tool pose trajectory can be obtained by applying the inverse kinematic of the machine on the desired tool pose trajectory \([P_{X_d}, P_{Y_d}, P_{Z_d}, \theta_{a_d}, \theta_{c_d}]\) [6]:

\[
\begin{bmatrix}
x_r \\
y_r \\
z_r
\end{bmatrix} =
\begin{bmatrix}
\cos(\theta_{c_d}) & -\sin(\theta_{c_d}) & 0 \\
\cos(\theta_{a_d})\sin(\theta_{c_d}) & -\cos(\theta_{a_d})\cos(\theta_{c_d}) & -\sin(\theta_{a_d}) \\
\sin(\theta_{a_d})\sin(\theta_{c_d}) & -\sin(\theta_{a_d})\cos(\theta_{c_d}) & \cos(\theta_{a_d})
\end{bmatrix}
\begin{bmatrix}
P_{X_d} \\
P_{Y_d} \\
P_{Z_d}
\end{bmatrix}
\]

\[
a_r = -\theta_{a_d} \\
c_r = -\theta_{c_d}
\]

where \(x_r, y_r, z_r\) correspond to the Cartesian axes of the machine, and \(a_r, c_r\) are the commands to the primary and secondary axes of the rotary table, respectively.
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

Figure 4.10: Schematic block diagram of the proposed integrated five-axis input shaping and contour error compensation.

If the axis commands obtained in Eq. (4.22) are sent directly to the machine, they might excite the structural vibrations of the system and also cause contour errors due to the dynamic limitation of the machine. Instead, as illustrated in Figure 4.10, these “unshaped” commands can be fed into the proposed command shaping module to remove the excitation harmonics of the input and compensate for the contour errors. The “shaped-compensated” trajectories calculated by this module result in vibration-free and accurate contouring of the path.
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

As the first step in the command shaping module, the input shaper designed in Chapter 3 is applied on the original axis reference commands \([x_r, y_r, z_r, a_r, c_r]\) to remove the excitation harmonics, resulting in the shaped reference commands \([x_{rs}, y_{rs}, z_{rs}, a_{rs}, c_{rs}]\):

\[
q_{rs}(k) = \text{conv}[q_r(k), \text{shaper}] = \sum_{j=1}^{m} A_j q_r(k - k_j) \mathbf{1}\{k - k_j\} \quad q = x, y, z, a, c
\]  

(4.23)

where \(A_j\) and \(t_j\) are the magnitude and time delay of the \(j\)th impulse, \(T_s\) is the sampling time, and \(k_j = \text{fix}(t_j / T_s)\) is the corresponding number of delayed sample times for the \(j\)th impulse; \(\mathbf{1}\{k - k_j\}\) is also the unit step function applied at the sampling counter \(k = k_j\):

\[
\begin{align*}
\mathbf{1}\{k - k_j\} &= 1 \quad k \geq k_j \\
\mathbf{1}\{k - k_j\} &= 0 \quad k < k_j
\end{align*}
\]  

(4.24)

Using the available models describing the closed loop dynamic of axis servo drives, the actual response of each axis to the corresponding shaped command can be approximately predicted as:

\[
\tilde{q}_{as}(k) = \text{conv}[q_{rs}(k), G_q(z^{-1})] \quad q = x, y, z, a, c
\]  

(4.25)

where \(G_q(z^{-1})\) is the closed loop model of the axis. This model can be as simple as a second order system, or a very sophisticated model including friction and nonlinearities. For example, for a linear model:

\[
\tilde{q}_{as}(z^{-1}) = G_q(z^{-1}) \cdot q_{rs}(z^{-1}) \quad q = x, y, z, a, c
\]  

(4.26)

Several experimental examples for modeling of ball-screw drives and linear motor-driven drives are provided in Chapter 5.

The predicted shaped axis responses \([\tilde{x}_{as}, \tilde{y}_{as}, \tilde{z}_{as}, \tilde{a}_{as}, \tilde{c}_{as}]\) obtained in Eq. (4.25) can be used to predict the resultant contour errors caused by both input shaping and machine dynamic. However, since contour error is defined in the workpiece frame, direct kinematic
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

of the machine must be used to obtain the predicted shaped tool pose 
\[ \tilde{P}_{as} = \left[ \tilde{P}_{X_{as}}, \tilde{P}_{Y_{as}}, \tilde{P}_{Z_{as}}, \tilde{\theta}_{a_{as}}, \tilde{\theta}_{c_{as}} \right] \] in the workpiece frame [6]:

\[
\begin{bmatrix}
\tilde{P}_{X_{as}} \\
\tilde{P}_{Y_{as}} \\
\tilde{P}_{Z_{as}} \\
\tilde{\theta}_{a_{as}} \\
\tilde{\theta}_{c_{as}}
\end{bmatrix}
= 
\text{ROT}_{z}^{-1}(\tilde{c}_{as}). \text{ROT}_{x}^{-1}(\tilde{a}_{as}). \text{Trans}(\tilde{x}_{as}, \tilde{y}_{as}, \tilde{z}_{as}). Tp
\]

\[
\begin{align*}
\tilde{\theta}_{a_{as}} &= -\tilde{\alpha}_{as} \\
\tilde{\theta}_{c_{as}} &= -\tilde{\phi}_{as}
\end{align*}
\]  (4.27)

where

\[
\text{ROT}_{z}(\tilde{c}_{as}) = 
\begin{bmatrix}
\cos(\tilde{c}_{as}) & -\sin(\tilde{c}_{as}) & 0 & 0 \\
\sin(\tilde{c}_{as}) & \cos(\tilde{c}_{as}) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  (4.28)

\[
\text{ROT}_{x}(\tilde{a}_{as}) = 
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(\tilde{a}_{as}) & -\sin(\tilde{a}_{as}) & 0 \\
0 & \sin(\tilde{a}_{as}) & \cos(\tilde{a}_{as}) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  (4.29)

\[
\text{Trans}(\tilde{x}_{as}, \tilde{y}_{as}, \tilde{z}_{as}) = 
\begin{bmatrix}
1 & 0 & 0 & -\tilde{x}_{as} \\
0 & 1 & 0 & -\tilde{y}_{as} \\
0 & 0 & 1 & \tilde{z}_{as} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  (4.30)

\[
Tp = [0, 0, 0, 1]^T
\]  (4.31)

Therefore,

\[
\begin{align*}
\tilde{P}_{X_{as}} &= \tilde{x}_{as} \cos(\tilde{c}_{as}) - \tilde{y}_{as} \sin(\tilde{c}_{as}) + \tilde{z}_{as} \sin(\tilde{a}_{as}) \sin(\tilde{c}_{as}) \\
\tilde{P}_{Y_{as}} &= \tilde{x}_{as} \sin(\tilde{c}_{as}) - \tilde{y}_{as} \cos(\tilde{a}_{as}) \cos(\tilde{c}_{as}) + \tilde{z}_{as} \cos(\tilde{a}_{as}) \cos(\tilde{c}_{as}) \\
\tilde{P}_{Z_{as}} &= \tilde{y}_{as} \sin(\tilde{a}_{as}) + \tilde{z}_{as} \cos(\tilde{a}_{as})
\end{align*}
\]  (4.32)

\[
\begin{align*}
\tilde{\theta}_{a_{as}} &= -\tilde{\alpha}_{as} \\
\tilde{\theta}_{c_{as}} &= -\tilde{\phi}_{as}
\end{align*}
\]
Next, the contour error estimation method presented in Section 4.3 is used to calculate the tooltip contour error \( \tilde{e} = \left[ \tilde{e}_{xs}, \tilde{e}_{ys}, \tilde{e}_{zs} \right]_w \) between the predicted shaped tooltip position \( \left[ \tilde{P}_{xs}, \tilde{P}_{ys}, \tilde{P}_{zs} \right]_w \) and the original desired one \( \left[ P_{x_d}, P_{y_d}, P_{z_d} \right]_w \). This predicted contour error is used to pre-compensate the geometric errors caused by both input shaping and machine dynamics.

However, due to the nonlinear kinematic transformation between the axis commands (in the base frame) and tool pose (in the workpiece frame), the components of the predicted tooltip contour error \( \tilde{e} = \left[ \tilde{e}_{xs}, \tilde{e}_{ys}, \tilde{e}_{zs} \right]_w \) cannot be directly added to the shaped axis commands for compensation. Instead, as illustrated in the block diagram in Figure 4.10, they must be added to the shaped tool pose described in the workpiece frame. Direct kinematic of the machine derived in Eq. (4.32) is used again to obtain the shaped tool pose \( \left[ P_{x_{ds}}, P_{y_{ds}}, P_{z_{ds}}, \theta a_{ds}, \theta c_{ds} \right]_w \) from the shaped axis commands \( \left[ x_{rs}, y_{rs}, z_{rs}, a_{rs}, c_{rs} \right] \):

\[
\begin{align*}
P_{x_{ds}} &= -x_{rs} \cos(c_{rs}) - y_{rs} \cos(a_{rs}) \sin(c_{rs}) + z_{rs} \sin(a_{rs}) \sin(c_{rs}) \\
P_{y_{ds}} &= x_{rs} \sin(c_{rs}) - y_{rs} \cos(a_{rs}) \cos(c_{rs}) + z_{rs} \sin(a_{rs}) \cos(c_{rs}) \\
P_{z_{ds}} &= y_{rs} \sin(a_{rs}) + z_{rs} \cos(a_{rs}) \\
\theta a_{ds} &= -a_{rs} \\
\theta c_{ds} &= -c_{rs}
\end{align*}
\] (4.33)

The components of the predicted tooltip contour error \( \left[ \tilde{e}_{xs}, \tilde{e}_{ys}, \tilde{e}_{zs} \right]_w \) are then added to the shaped tooltip position \( \left[ P_{x_{ds}}, P_{y_{ds}}, P_{z_{ds}} \right]_w \) (in the workpiece frame) to obtain the shaped-compensated tool pose \( \left[ P_{x_{ds}c}, P_{y_{ds}c}, P_{z_{ds}c}, \theta a_{ds}c, \theta c_{ds}c \right]_w \):

\[
\begin{align*}
P_{x_{ds}c} &= P_{x_{ds}} + \tilde{e}_{xs} \\
P_{y_{ds}c} &= P_{y_{ds}} + \tilde{e}_{ys} \\
P_{z_{ds}c} &= P_{z_{ds}} + \tilde{e}_{zs} \\
\theta a_{ds}c &= \theta a_{ds} \\
\theta c_{ds}c &= \theta c_{ds}
\end{align*}
\] (4.34)
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

The corresponding shaped-compensated axis commands \([\hat{x}_{rsc}, \hat{y}_{rsc}, \hat{z}_{rsc}, \hat{a}_{rsc}, \hat{c}_{rsc}]\) which result in the shaped-compensated tool pose \([P_{x\,dsc}, P_{y\,dsc}, P_{z\,dsc}, \theta_{a\,dsc}, \theta_{c\,dsc}]\) can be obtained by using the inverse kinematic of the machine:

\[
\begin{bmatrix}
\hat{x}_{rsc} \\
\hat{y}_{rsc} \\
\hat{z}_{rsc}
\end{bmatrix}
= 
\begin{bmatrix}
-\cos(\theta_{c\,dsc}) & -\sin(\theta_{c\,dsc}) & 0 \\
\cos(\theta_{a\,dsc})\sin(\theta_{c\,dsc}) & -\cos(\theta_{a\,dsc})\cos(\theta_{c\,dsc}) & -\sin(\theta_{a\,dsc}) \\
\sin(\theta_{a\,dsc})\sin(\theta_{c\,dsc}) & -\sin(\theta_{a\,dsc})\cos(\theta_{c\,dsc}) & \cos(\theta_{a\,dsc})
\end{bmatrix}
\begin{bmatrix}
P_{x\,dsc} \\
P_{y\,dsc} \\
P_{z\,dsc}
\end{bmatrix}
\]

\[
\hat{a}_{rsc} = -\theta_{a\,dsc} \\
\hat{c}_{rsc} = -\theta_{c\,dsc}
\]

Compared to the shaped axis commands \([x_{rs}, y_{rs}, z_{rs}, a_{rs}, c_{rs}]\), the shaped-compensated commands \([\hat{x}_{rsc}, \hat{y}_{rsc}, \hat{z}_{rsc}, \hat{a}_{rsc}, \hat{c}_{rsc}]\) obtained in Eq. (4.35) include additional correcting terms which can be calculated as:

\[
\begin{align*}
\delta\hat{x}_{cmp} &= \hat{x}_{rsc} - x_{rs} \\
\delta\hat{y}_{cmp} &= \hat{y}_{rsc} - y_{rs} \\
\delta\hat{z}_{cmp} &= \hat{z}_{rsc} - z_{rs}
\end{align*}
\] (4.36)

However, the compensation terms \([\delta\hat{x}_{cmp}, \delta\hat{y}_{cmp}, \delta\hat{z}_{cmp}]\) added to the shaped axis commands might again bring some excitation harmonics to the axis input and cause structural vibration. Therefore, the compensation signals calculated in Eq. (4.36) must be low-pass filtered at five times to ten times smaller than the lowest structural mode of the machine:

\[
\begin{align*}
\delta x_{cmp} &= \text{LPF}\left(\delta\hat{x}_{cmp}\right) \\
\delta y_{cmp} &= \text{LPF}\left(\delta\hat{y}_{cmp}\right) \\
\delta z_{cmp} &= \text{LPF}\left(\delta\hat{z}_{cmp}\right)
\end{align*}
\] (4.37)

It should be noted that due to the nonlinear kinematic transformation, filtering of the compensation terms \([\hat{e}_{xs}, \hat{e}_{ys}, \hat{e}_{zs}]\) in Eq. (4.34) does not guarantee that the resultant
correcting terms added to the axis commands will be filtered as well. Therefore, low-pass filtering could not be accomplished in Eq. (4.34).

Finally, the filtered shaped-compensated commands \([x_{rsc}, y_{rsc}, z_{rsc}, a_{rsc}, c_{rsc}]\), which result in vibration-free and accurate contouring of the desired path, are obtained by adding the filtered compensation signals \([\delta x_{cmp}, \delta y_{cmp}, \delta z_{cmp}]\) from Eq. (4.37) to the shaped axis commands \([x_{rs}, y_{rs}, z_{rs}, a_{rs}, c_{rs}]\) calculated in Eq. (4.23):

\[
\begin{align*}
    x_{rsc} &= x_{rs} + \delta x_{cmp} \\
    y_{rsc} &= y_{rs} + \delta y_{cmp} \\
    z_{rsc} &= z_{rs} + \delta z_{cmp} \\
    a_{rsc} &= a_{rs} \\
    c_{rsc} &= c_{rs}
\end{align*}
\]  

(4.38)

It should be noted that the compensation technique described above has been developed to correct only the tooltip contour error. In case of five-axis machines, in addition to the tooltip contour error, another form of error is defined which is related to the deviation of the tool orientation from the desired one. Similar to tooltip position error, the orientation contour error must be investigated and compensated as well, which is left for future work.

4.5.2 Simplified Versions of the Proposed Command Shaping Module

The command shaping module presented in the previous section and illustrated in Figure 4.10 is a very general form developed for 5-axis machines; it includes both input shaping for vibration avoidance as well as compensation of contour errors caused by both input shaping distortion and machine dynamics. However, depending on the application, simplified versions of this module which serves a certain purpose can be also extracted.

As the simplest case, in some applications only the start and end points of the path are important and the relatively small contour error along the toolpath caused by the shaping distortion and machine dynamics is not problematic. For instance, in positioning of the tool in point-to-point drilling, only vibration-free conditions at the beginning and end is
important but the travelled path in the air does not particularly matter. In this case, only the input shaping block which shapes the individual axis commands will suffice.

Another case is when the tracking performance of axis drives are good enough that the contour error caused by the machine dynamics is not significant, and thus does not need to be compensated; however, the machine has some flexible components with low frequency resonance modes which must be avoided. In this case, the toolpath distortion caused by input shaping might be considerable, and must be compensated for. This is usually the case for most CNC machine tools with high bandwidth axis control loops but with low-frequency column modes. A simplified version of the proposed command shaping module can be used very effectively in these applications. The overall procedure is almost the same as the block diagram in Figure 4.10, the only difference is that instead of using the prediction of the actual tool pose \([\tilde{P}_x, \tilde{P}_y, \tilde{P}_z, \tilde{\theta}_a, \tilde{\theta}_c]\), the contour error is calculated between the shaped tool pose \([P_x, P_y, P_z, \theta_a, \theta_c]\) and the original desired pose \([P_x, P_y, P_z, \theta_a, \theta_c]\). This way, only the path distortion caused by input shaping is compensated. Note that in this case, the dynamic models of the feed drives are not needed, and the path distortion is calculated semi-analytically only based on the shaped and desired tool pose.

The proposed command shaping module can be used for the opposite case as well, that is when the contour error caused by the machine dynamics is large, but there is no vibration problem. Figure 4.11 illustrates the simplified command shaping module for this case. Note that although input shaping of the original axis commands is not needed, the estimated components of the contour errors must be still low-pass filtered before being added to the original axis commands; otherwise, the high frequency harmonics of the compensation signals might saturate the drives and cause large contour errors.
Figure 4.11: Simplified version of the proposed command shaping module; only compensating the contour errors caused by the machine dynamics, no vibration avoidance.

For the case of three-axis Cartesian machines, the module is simplified significantly as the workpiece frame always stays parallel to the base frame. Since there is no nonlinear kinematic transformation involved, the components of the predicted contour error can be added directly to the axis commands. Figure 4.12 illustrates the general form of the proposed module for three-axis Cartesian machines. Note that it includes vibration avoidance and compensates for contour errors caused by both input shaping and machine dynamics.
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

![Schematic block diagram of the proposed integrated input shaping and contour error compensation for three-axis Cartesian machines.](image)

**Figure 4.12:** Schematic block diagram of the proposed integrated input shaping and contour error compensation for three-axis Cartesian machines.

### 4.6 Statistical Evaluation of Contour Error Compensation

Assume that an arbitrary toolpath (tooltip positions) denoted by $TP_{ref}$ is to be contoured by a five-axis machine tool. In order to avoid excitation of the machine, the reference has been shaped by an arbitrary shaper. As discussed earlier, the shaped toolpath $TP_{shp}$ exhibits some distortion relative to the original one, which can be represented as:

$$
\epsilon_{shp} = TP_{ref} - TP_{shp}
$$

where $\epsilon_{shp}$ can be calculated as the shortest distance from each point on the shaped toolpath $TP_{shp}$ to the original one $TP_{ref}$.

If the shaped trajectory is given to the machine without any modification, the machine response generates the actual toolpath denoted as $TP_{act}$, which has contouring...
error due to the shaping distortion and machine dynamic. Using the same contour error
definition presented in Section 4.2, the actual observed error $\varepsilon_{\text{act}}$ can be written as:

$$\varepsilon_{\text{act}} = TP_{\text{ref}} - TP_{\text{act}}$$  \hspace{1cm} (4.40)

From the pre-compensation point of view presented in the previous section, this contour
error can be decomposed into the reference- and machine-related terms:

$$\varepsilon_{\text{act}} = \varepsilon_{\text{shp}} + \varepsilon_{\text{machine}}$$  \hspace{1cm} (4.41)

where $\varepsilon_{\text{shp}}$ represents the path distortion caused by input shaping, and $\varepsilon_{\text{machine}}$ is the contour
eroerror caused due to the machine dynamics, both presented in Section 4.4. Since the original
and shaped reference commands are well-known vectors, the exact value of the distortion
error $\varepsilon_{\text{shp}}$ can be calculated, except for the estimation error of the employed contour error
calculation method. The machine-dependent contour error $\varepsilon_{\text{machine}}$, on the other hand, is not
known but still can be broken into two parts:

$$\varepsilon_{\text{machine}} = \varepsilon_{\text{rep}} + \varepsilon_{\text{rand}}$$  \hspace{1cm} (4.42)

where $\varepsilon_{\text{rep}}$ represents the repetitive part of contour error observed every time that the path is
contoured with the same machine, and $\varepsilon_{\text{rand}}$ is the stochastic part of the error which occurs
randomly, and thus, cannot be modeled or predicted. Depending on the available model of
the feed drives, some part of the repetitive part $\varepsilon_{\text{rep}}$ might be possible to predict; that is:

$$\varepsilon_{\text{rep}} = \varepsilon_{\text{mdl}} + \varepsilon_{\text{rep-Nmdl}}$$  \hspace{1cm} (4.43)

where $\varepsilon_{\text{mdl}}$ is the part of the repetitive error which can be modeled, and $\varepsilon_{\text{rep-Nmdl}}$ represents
the part which is repetitive but cannot be predicted by the available model. Combining
Eqs. (4.41), (4.42) and (4.43), the total actual contour error can be decomposed into four
terms:

$$\varepsilon_{\text{act}} = \underbrace{\varepsilon_{\text{shp}} + \varepsilon_{\text{mdl}}}_{\text{Predictable}} + \underbrace{\varepsilon_{\text{rep-Nmdl}} + \varepsilon_{\text{rand}}}_{\text{Non-predictable}}$$  \hspace{1cm} (4.44)
where $\varepsilon_{\text{shp}}$ and $\varepsilon_{\text{mdl}}$ can be predicted knowing the reference, shaper, and model of the drives. From the statistical point of view, we can assume that the components in Eq. (4.44) are mutually independent and Gaussian (normally) distributed, that is:

$$\varepsilon_{\text{shp}} \sim N\left(\mu_{\text{shp}}, \sigma_{\text{shp}}^2\right)$$ (4.45)

$$\varepsilon_{\text{mdl}} \sim N\left(\mu_{\text{mdl}}, \sigma_{\text{mdl}}^2\right)$$ (4.46)

$$\varepsilon_{\text{rep-Nmdl}} \sim N\left(\mu_{\text{rep-Nmdl}}, \sigma_{\text{rep-Nmdl}}^2\right)$$ (4.47)

$$\varepsilon_{\text{rand}} \sim N\left(0, \sigma_{\text{rand}}^2\right)$$ (4.48)

where $\mu$ is the mean (expected) value:

$$\mu = E[\varepsilon] = \text{mean}(\varepsilon)$$ (4.49)

and $\sigma^2$ is the variance ($\sigma$ is the standard deviation):

$$\sigma^2 = E[(\varepsilon - \mu)^2] = E[\varepsilon^2] - 2\mu E[\varepsilon] + \mu^2 = E[\varepsilon^2] - \mu^2 = E[\varepsilon^2] - \left(E[\varepsilon]\right)^2$$ (4.50)

which is verbally stated as “mean of squared values minus squared of mean value”. Note that the mean value of random part in Eq. (4.48) has been considered to be zero, which means that the random part is not biased towards negative or positive contour errors. Using the definitions and assumptions given in Eqs. (4.41)-(4.48), the total actual contour error can be evaluated by its mean value $\mu_{\text{act}}$ and error level $EL_{\text{act}}$ as:

$$\mu_{\text{act}} = E[\varepsilon_{\text{act}}] = \text{mean}(\varepsilon_{\text{act}}) = \mu_{\text{shp}} + \mu_{\text{mdl}} + \mu_{\text{rep-Nmdl}}$$ (4.51)
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

\[ EL_{\text{act}} = \sqrt{E\left(\varepsilon_{\text{act}}^2\right)} = \sqrt{\mu_{\text{shp}}^2 + \mu_{\text{mdl}}^2 + \mu_{\text{rep-Nmdl}}^2 + \sigma_{\text{shp}}^2 + \sigma_{\text{mdl}}^2 + \sigma_{\text{rep-Nmdl}}^2 + \sigma_{\text{rand}}^2} \]  

(4.52)

Basically, the proposed contour error compensation tries to decrease these error values by pre-compensating for the path distortion \( \varepsilon_{\text{shp}} \) and modeled machine-dependent contour error \( \varepsilon_{\text{mdl}} \):

\[ \varepsilon_{\text{compSig}} = \varepsilon_{\text{shp}} + \varepsilon_{\text{mdl}} \]  

(4.53)

However, as described in Section 4.4, the predicted contour error compensations signals are low-pass filtered before being added to the shaped command. This filtering might result in some deviation between the modeled contour error and the final compensating terms. In addition, since calculating the exact value of contour error requires time-consuming iterative methods, usually an estimation technique such as the one presented in Section 4.3 is employed. This brings additional error in approximating the actual values of \( \varepsilon_{\text{shp}} \) and \( \varepsilon_{\text{mdl}} \). Therefore, if \( \hat{\varepsilon}_{\text{shp}} \) and \( \hat{\varepsilon}_{\text{mdl}} \) are the approximated compensation terms after filtering, the total correcting term can be represented as:

\[ \hat{\varepsilon}_{\text{compSig}} = \hat{\varepsilon}_{\text{shp}} + \hat{\varepsilon}_{\text{mdl}} \]  

(4.54)

and the error in calculating the compensation signal is:

\[ \text{err}\hat{\varepsilon}_{\text{compSig}} = \varepsilon_{\text{compSig}} - \hat{\varepsilon}_{\text{compSig}} \]  

(4.55)

which can also be assumed to have normal distribution with mean value of \( \mu_{\text{err}} \) and variance of \( \sigma_{\text{err}}^2 \):

\[ \text{err}\hat{\varepsilon}_{\text{compSig}} \sim N\left(\mu_{\text{err}}, \sigma_{\text{err}}^2\right) \]  

(4.56)

The shaped-compensated toolpath is obtained by adding the approximated compensating term to the shaped reference:

\[ TP_{\text{shp-cmp}} = TP_{\text{shp}} + \hat{\varepsilon}_{\text{compSig}} \]  

(4.57)
If this modified trajectory is given to the machine, the actual contour error can be represented as:
\[ \varepsilon_{\text{act-cmp}} = TP_{\text{ref}} - TP_{\text{act-cmp}} \]  
(4.58)

where \( TP_{\text{ref}} \) is the original desired toolpath; \( TP_{\text{act-cmp}} \) is also the actual compensated response, which can be decomposed using the definition of the components in Eq. (4.44):
\[ TP_{\text{act-cmp}} = TP_{\text{shp-cmp}} - \varepsilon_{\text{mdl}} - \varepsilon_{\text{rep-Nmdl}} - \varepsilon_{\text{rand}} \]  
(4.59)

Substituting the expressions from Eqs. (4.53)-(4.59) into Eq. (4.58), the actual shaped-compensated contour error can be derived as follows:
\[
\begin{align*}
\varepsilon_{\text{act-cmp}} &= TP_{\text{ref}} - TP_{\text{act-cmp}} \\
&= TP_{\text{ref}} - TP_{\text{shp-cmp}} + \varepsilon_{\text{mdl}} + \varepsilon_{\text{rep-Nmdl}} + \varepsilon_{\text{rand}} \\
&= TP_{\text{ref}} - TP_{\text{shp}} - \hat{\varepsilon}_{\text{cmpSig}} + \varepsilon_{\text{mdl}} + \varepsilon_{\text{rep-Nmdl}} + \varepsilon_{\text{rand}} \\
&= err\hat{\varepsilon}_{\text{cmpSig}} + \varepsilon_{\text{rep-Nmdl}} + \varepsilon_{\text{rand}} \\
\end{align*}
\]  
(4.60)

where the underlined terms in each step are the ones substituted in the subsequent step. Similar to the mean value and error level calculated for the shaped response in Eqs. (4.51) and (4.52), for the shaped-compensated response it can be written as:
\[
\begin{align*}
\mu_{\text{act-cmp}} &= E\left[\varepsilon_{\text{act-cmp}}\right] = \text{mean}\left(\varepsilon_{\text{act-cmp}}\right) = \mu_{\text{cmpErr}} + \mu_{\text{rep-Nmdl}} \\
E L_{\text{act-cmp}} &= \sqrt{E\left(\varepsilon_{\text{act-cmp}}^2\right)} = \sqrt{\mu_{\text{cmpErr}}^2 + \mu_{\text{rep-Nmdl}}^2 + \sigma_{\text{cmpErr}}^2 + \sigma_{\text{rep-Nmdl}}^2 + \sigma_{\text{rand}}^2} \\
\end{align*}
\]  
(4.61)

In order to evaluate the effectiveness of pre-compensation technique, the values in Eqs. (4.61) and (4.62) can be compared against the uncompensated case in Eqs. (4.51) and (4.52). Also as a more explicit tool, we can compare the achieved improvement against the original contour error: considering the squared of error level \( EL \) as a rough index representing the overall level of error, the improvement achieved by compensation can be evaluated as:
Chapter 4. Improving Accuracy by Pre-compensation of Contour Errors

\[ I_{\text{impr}} = E L_{\text{act}}^2 - E L_{\text{act-comp}}^2 \]

\[ = \left( \mu_{\text{shp}}^2 + \mu_{\text{mdl}}^2 + \mu_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{shp}}^2 + \sigma_{\text{mdl}}^2 + \sigma_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{rand}}^2 \right) \]

\[ - \left( \mu_{\text{cmpErr}}^2 + \mu_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{cmpErr}}^2 + \sigma_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{rand}}^2 \right) \]

\[ = \mu_{\text{shp}}^2 + \mu_{\text{mdl}}^2 - \mu_{\text{cmpErr}}^2 + \sigma_{\text{shp}}^2 + \sigma_{\text{mdl}}^2 - \sigma_{\text{cmpErr}}^2 \]  

(4.63)

Using pre-compensation is beneficial only if this improvement is significant compared to the non-compensatable part:

\[ I_{\text{impr}} > \mu_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{rand}}^2 \]  

(4.64)

Finally, combining Eqs. (4.63) and (4.64), a rough criterion to evaluate the effectiveness of pre-compensation can be presented as:

\[ \mu_{\text{shp}}^2 + \mu_{\text{mdl}}^2 + \sigma_{\text{shp}}^2 + \sigma_{\text{mdl}}^2 > \mu_{\text{cmpErr}}^2 + \mu_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{cmpErr}}^2 + \sigma_{\text{cmpErr-rep-Nmdl}}^2 + \sigma_{\text{rand}}^2 \]  

(4.65)

where as defined earlier, subscript \( \text{shp} \) relates to shaper distortion, \( \text{mdl} \) represents the error predictable by the available model, \( \text{cmpErr} \) is the difference between the raw calculated compensation signal and the final correcting terms after applying the low-pass filter; In addition, subscript \( \text{rep-Nmdl} \) represents the part of the error which is repetitive but cannot be predicted by the model, and finally, \( \text{rand} \) relates to the stochastic part of contour error caused randomly and non-repetitively.

Remarks

Looking more closely at the criterion derived in Eq. (4.65) gives a good insight to decide when the pre-compensation is beneficial. Basically the greater the left hand side is relative to the right hand side, the more it makes sense to use the compensation; therefore:

1. Pre-compensation is definitely necessary if the path distortion \( \varepsilon_{\text{shp}} \) caused by input shaping is relatively significant; the higher feedrate we choose and the lower resonance frequencies the machine has, the larger path distortions will be observed, and thus, the more it becomes necessary to pre-compensate this error.
2. From the repetitive part of the machine/process-dependent contour error, the part which can be predicted by the available model is on left hand side of the inequality whereas the non-predictable part is on the right-hand side. Expectedly, this means that the more we can model the behavior of the machine, the more dominant the left hand side becomes relative to the right hand side, and thus, the more it becomes beneficial to use pre-compensation. For example, the contour error caused by tool deflection is normally considered as repetitive but not predictable; however, if we can model the tool engagement and predict the resulting cutting forces and tool deflection, this effect will be taken to the left hand side of the inequality. Also assembly errors and structural misalignments are normally classified as repetitive but not modeled, although metrology researchers have been working on modeling and compensating these errors as well.

3. Contour error compensation technique is meant to correct only the mean value or relatively low frequency harmonics of the predicted contour error. Therefore, the compensating terms have to be low-pass filtered before being added to the shaped commands; otherwise they might excite the structural modes of the machine. The error caused by filtering the compensation signal, i.e. $N\left(\mu_{cmpErr}, \sigma_{cmpErr}^2\right)$, depends on the path geometry, machine dynamics, process, etc, but is not controlled by the user; generally, sharp changes in the reference or machine response cannot be totally compensated even if they are predictable by the model; this is because compensating these sharp changes requires high frequency contents in the correcting signal, which are inevitably blocked by the filter applied on the compensating signals. For example, the contour error observed at sharp corners, which is due to the limited bandwidth of the machine, might be possible to reduce but cannot be completely compensated. As an another example, at the locations of the path where axis drives change directions, static friction behaves as a step disturbance to the system and causes small glitches; although employing pre-compensation may result in smoother response with smaller jumps at these points, the sharp change in contour error cannot be completely eliminated.
4. Finally, as the name suggests, the error $\varepsilon_{\text{rand}}$ caused by random effects cannot be predicted or compensated. However, it might be possible to reduce these errors by improving the overall machine condition such as hardware accuracy and reliability, which is not studied in this work.

4.7 Summary

Contouring errors exhibited in five-axis machining was thoroughly investigated in this chapter. Mathematical representation and a method for direct approximation of contour errors were presented. It was shown that due to the delay caused by input shapers, the shaped toolpath is always distorted compared to the original one. This distortion adds up to the errors originated from the limited bandwidth of the servo drives.

A novel command shaping module was proposed which first shapes the original reference commands to avoid excitation of the structural modes of the five-axis machine. The available model of the machine dynamics is then used to predict the contour errors caused by both input shaping and machine dynamics. Correcting signals are calculated based on the predicted contour errors, and the compensation terms are added to the shaped commands to avoid these errors before their occurrence. Finally, a statistical analysis was performed to investigate the cases when pre-compensation can be effective.
5 Implementation and Experimental Results

5.1 Introduction

In order to verify the effectiveness of the command shaping techniques proposed in chapter 3, a number of experiments have been conducted on several machines with different types of feed drives. The implementation procedure for each set of experiment is presented in this chapter. It is worth reminding that the main focus is on the command modification part and not the controller. Therefore, minor details of system identification and controller design is not presented here; cited references can be studied accordingly.

The chapter is organized as follows: implementation of single mode and multi mode input shaping to avoid excitation of two oscillators mounted on a single axis linear drive is presented in Section 5.2. Followed in Section 5.3, the proposed integrated input shaping and contour error compensation technique is implemented on a two-axis linear motor-driven XY table to achieve accurate and vibration free contouring of a circular path. However, since the vibration of the linear drive is not significant, artificial low damping resonance modes are imposed in the closed loop through controller design, and input shaping is used to avoid excitation of these modes. Section 5.4 presents a set of realistic experiments conducted on a five-axis machining center, and structural vibrations were measured by accelerometers and Ferraris sensor to verify the vibration reduction effect.

5.2 Implementation on the Single-axis Linear Drive with Oscillator

In order to visually demonstrate how input shaping can effectively avoid mode excitations, the single axis linear drive shown in Figure 5.1 was used as the test bed. Two flexible oscillator are mounted on the table to model the structural modes of a machine tool feed drive. It should be noted that unlike active damping techniques, there is no feedback from the vibrations of the oscillators in the control law.
5.2.1 Experimental Setup

Figure 5.1 shows the Siemens 1FN1 Linear Motor used for the preliminary single-axis experiments. Table 5.1 provides the relevant dynamic specifications of this high force motor.

![Siemens 1FN1 Linear Motor with flexible oscillators.](image)

**Figure 5.1**: Siemens 1FN1 Linear Motor with flexible oscillators.

<table>
<thead>
<tr>
<th>Max. Force</th>
<th>Max. Acceleration</th>
<th>Max. Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1720 N</td>
<td>2 g (19.6 m/s²)</td>
<td>200 m/min</td>
</tr>
</tbody>
</table>

**Table 5.1**: Dynamic specifications of Siemens 1FN1 Linear Motor.

The force delivered to the table is proportional to the current going through the motor coil. The required current is provided by a SIMODRIVE 611U unit, which is also shown in Figure 5.1. This unit can work in either velocity mode, in which the built-in controller in 611U is used, or it can be used in torque (current) mode, in which 611U work as a simple constant gain amplifier. In order to investigate the effect of command shaping techniques, it is preferable to use the 661U in the torque mode so that the controller can be
Chapter 5. Implementation and Experimental Results

designed and implemented by the user. Due to the large currents going through the motor, water at 20°C and 4L/min is flowing inside the table.

The position of the table is measured using Heidenhain LIDA18 linear encoder cascaded with a Heidenhain IBV606 interpolation box. Vibration of the oscillator is also measured using a Kistler 8702B500 accelerometer with the range of ±500g and sensitivity of 14.4mV/g, although it should be noted that the acceleration data is not used as feedback in the controller. Two flexible oscillators are made out of thin Aluminum bars with the dimensions provided in Table 5.2.

Table 5.2: Dimensions of the Aluminum oscillator.

<table>
<thead>
<tr>
<th>Oscillator</th>
<th>Height</th>
<th>Width</th>
<th>Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>328.5 mm</td>
<td>14.7 mm</td>
<td>1.6 mm</td>
</tr>
<tr>
<td>2</td>
<td>251.2 mm</td>
<td>14.7 mm</td>
<td>1.5 mm</td>
</tr>
</tbody>
</table>

Data acquisition and real time control of the system is implemented on a dSPACE DS1103, which is interfaced with an external PC through ControlDesk© software.

5.2.2 System Modeling

Since the mass of the oscillator is totally negligible compared to the mass of the table, it can be assumed that the vibration of the oscillator has no effect on the table. Therefore, the entire system can be decoupled into two sub-models; one from input command to table position, and the other from table position to oscillator position. The former is a simple rigid feed drive model, and the latter is a base-excitation vibration problem. Figure 5.2 shows the block diagram of the rigid model from input voltage to motor position. It is assumed that the electrical parts of the system such as the amplifier have relatively faster dynamics, and thus, they can be modeled as a single gain $K_a$. This assumption is reasonable, even for real industrial machines, unless for the cases when the controller has very high bandwidth, for example over 200Hz. Dynamic parameters of the
system were identified by a former student of MAL through unbiased least square technique [30]. Table 5.3 provides the relevant parameters of the linear motor.

**Figure 5.2:** Rigid body model of linear drives from input voltage to motor position.

**Table 5.3:** Dynamic parameters of the rigid model of the linear motor.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifier gain ($K_u$) [A/V]</td>
<td>1.1121</td>
</tr>
<tr>
<td>Motor force constant ($K_i$) [N/A]</td>
<td>142.0</td>
</tr>
<tr>
<td>Motor mass ($m_{tb}$) [Kg]</td>
<td>30.7</td>
</tr>
<tr>
<td>Table viscous friction coefficient ($b_{tb}$) [Kg/s]</td>
<td>156.1</td>
</tr>
<tr>
<td>Average Coulomb friction ($\mu_k$) [N]</td>
<td>43.4</td>
</tr>
</tbody>
</table>

Based on the block diagram shown in Figure 5.2, the transfer function of the rigid model for the linear drive can be represented as:

$$\frac{X_{tb}(s)}{V_{in}(s)} = \frac{K_uK_i}{s(m_{tb}s + b_{tb})} \tag{5.1}$$

The transfer function from the table position $x_{tb}(t)$ to the tip positions of the oscillator $x_{osc}(t)$ can be obtained by modeling it as a base excitation problem, as illustrated in Figure 5.3. If the oscillator has natural frequency of $\omega_n$ with damping of $\zeta$, the equation of motion for the tip oscillator can be written as:
\[
\ddot{x}_{\text{osc}}(t) + 2\zeta\omega_n (\dot{x}_{\text{osc}}(t) - \dot{x}_{tb}(t)) + \omega_n^2 (x_{\text{osc}}(t) - x_{tb}(t)) = 0
\]
\[
\rightarrow \quad \ddot{x}_{\text{osc}}(t) + 2\zeta\omega_n \dot{x}_{\text{osc}}(t) + \omega_n^2 x_{\text{osc}}(t) = 2\zeta\omega_n \dot{x}_{tb}(t) + \omega_n^2 x_{tb}(t)
\]  
(5.2)

**Figure 5.3:** Modeling the tip position of the oscillator as base-excitation.

The Laplace transform of Eq. (5.2) can be calculated as:

\[
X_{\text{osc}}(s) \left[ s^2 + 2\zeta\omega_n s + \omega_n^2 \right] = X_{tb}(s) \left[ 2\zeta\omega_n s + \omega_n^2 \right]
\]  
(5.3)

Therefore, the transfer function from the table position to the oscillator tip vibration is represented as:

\[
\frac{X_{\text{osc}}(s)}{X_{tb}(s)} = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}
\]  
(5.4)

Figure 5.4 shows the block diagram of the entire closed loop system including the rigid dynamics of the motor and oscillator vibrations. Note that unlike active damping techniques, there is no feedback from the vibrations of the oscillator; the control loop is closed with only motor position.
Figure 5.4: Block diagram of the closed loop system including the oscillator dynamics.

Damped natural frequency $\omega_d$ and damping ratio $\zeta$ of the oscillators can be identified by measuring the free vibrations after a small impulse disturbance to the oscillator tip. The vibrations can be measured using accelerometer or laser vibrometer, although some accelerometers do not give reliable results in low frequencies. Table 5.4 provides the oscillator parameters identified by laser vibrometer.

**Table 5.4: Parameters of the oscillator tip vibrations**

<table>
<thead>
<tr>
<th>Natural Frequency ($\omega_n$) [Hz]</th>
<th>Damping Ratio ($\zeta$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscillator 1</td>
<td>8.8</td>
</tr>
<tr>
<td>Oscillator 2</td>
<td>13.7</td>
</tr>
</tbody>
</table>

Substituting the parameters of the linear motor and the flexible oscillator into Eqs. (5.1) and (5.4), the transfer functions from the input voltage to motor position and from motor position to oscillator positions can be calculated as:

**Motor:** \( \frac{X_{th}(s)}{V_{in}(s)} = \frac{K_a K_i}{s(m_b s + b_b)} = \frac{157.9}{30.7s^2 + 156.1s} \) \[ (5.5) \]

**Oscillator 1:** \( \frac{X_{ox1}(s)}{X_{th}(s)} = \frac{2\zeta \omega_n s + \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} = \frac{1.659s + 3057.2}{s^2 + 1.659s + 3057.2} \) \[ (5.6) \]
Chapter 5. Implementation and Experimental Results

Oscillator 2: \[
\frac{X_{osc2}(s)}{X_{th}(s)} = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1.2051 s + 7409.7}{s^2 + 1.2051 s + 7409.7}
\] (5.7)

Figure 5.5 shows the bode plot of the system from input voltage to the position of the table and oscillator. \[
\left(\frac{X_{osc}(s)}{V_{in}(s)} = \frac{X_{osc}(s)}{X_{th}(s)} \times \frac{X_{th}(s)}{V_{in}(s)}\right)
\]

![Bode Diagram](image)

**Figure 5.5:** Bode plot of the system from input voltage to position of the table and oscillators.

5.2.3 Controller Design

In implementation of input shaping, only the rigid part of the drive is considered for controller design. In order to investigate the effect of input shaping on vibration of the oscillator, any simple controller which can move the table with small overshoot would work. However, in order to avoid plastic deformation of the oscillator, the amplitude of vibrations should be kept below 5cm. Therefore, the magnitude of the reference step
command must be chosen according to controller rise time. The more aggressive the
controller is, the smaller reference input the user should choose.

For the following set of experiment, a simple PID controller which gives 2%
overshoot and 0.4s rise time was designed using MATLAB Control Toolbox. The transfer
function of the tuned controller is:

\[
G_{\text{cont}}(s) = K_p + \frac{K_i}{s} + K_d\left(\frac{Ns}{s+N}\right)
\]

where \(N\) is the lowpass filtering parameter for realization of the derivative term.

\[
= 4.33 + \frac{0.156}{s} + 0.486\left(\frac{10s}{s+10}\right) \\
(5.8)
\]

5.2.4 Single-mode and Multi-mode Input shaping

Three cases are investigated in this experiment: 1) unshaped step, 2) single-mode
shaping based on only the first oscillator, 3) multi-mode shaping to avoid excitation of both
oscillators. Zero-Vibration and Derivative (ZVD) shaper, which gives descent robustness
and time delay as described in Section 3.4.3, is used for shaping the step input. Plugging
the values of the natural frequency and damping ratio of the oscillators into Eq. (3.43), the
magnitude and time locations of the three impulses in the ZVD shapers can be calculated
as:

Oscillator 1:

\[
\omega_n = 8.8 \text{ Hz} = 55.3 \text{ rad/s} \quad \zeta = 0.015
\]

\[
T_d = 2\pi/\omega_n\sqrt{1-\zeta^2} = 0.114 \text{ s} \\
K = e^{-\pi\zeta/\sqrt{1-\zeta^2}} = 0.9540
\]

\[
\begin{bmatrix} A \\ t_j \end{bmatrix} = \begin{bmatrix} \frac{1}{K^2+2K+1}, \{2K\}^{1}_{1}, \{K^2\}^{1}_{2}, \{K^2\}^{1}_{3} \\ 0, \{K^2\}^{2}_{1}, \{0.5T_d\}^{2}_{2}, \{T_d\}^{2}_{3} \\ 0.2619, \{0.4997\}^{3}_{1}, \{0.2384\}^{3}_{2}, \{0.114\}^{3}_{3} \end{bmatrix}
\]

\[
(5.9)
\]
Chapter 5. Implementation and Experimental Results

Oscillator 2: \( \omega_n = 13.7 \text{ Hz} = 86.1 \text{ rad/s} \) \[ \zeta = 0.007 \] \[ T_d = 2\pi / \omega_n \sqrt{1 - \zeta^2} = 0.073 \text{ s} \] \[ K = e^{-\pi \zeta / \sqrt{1 - \zeta^2}} = 0.9782 \]

\[
\begin{bmatrix}
A_j \\
t_j
\end{bmatrix} = \begin{bmatrix}
\frac{1}{K^2 + 2K + 1} & \frac{2K}{K^2 + 2K + 1} & \frac{K^2}{K^2 + 2K + 1} \\
0 & 0.5T_d & T_d
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0.2555 & 0.4999 & 0.2445 \\
0 & 0.037 & 0.073
\end{bmatrix}
\] (5.10)

In order to obtain the total shaper which avoids both modes, the easiest way is to convolve the individual shapers calculated in Eqs. (5.9) and (5.10):

\[
\text{Total Shaper : } \begin{bmatrix}
A_j \\
t_j
\end{bmatrix} = \text{conv}\left( \begin{bmatrix}
A_j \\
t_j
\end{bmatrix}_{\text{osc1}}, \begin{bmatrix}
A_j \\
t_j
\end{bmatrix}_{\text{osc2}} \right)
\]

\[
\begin{bmatrix}
A_j \\
t_j
\end{bmatrix} = \begin{bmatrix}
0.067 & 0.1309 & 0.1277 & 0.0640 & 0.2498 & 0.0609 \\
0.1222 & 0.1192 & 0.0583 & 0.093 & 0.114
\end{bmatrix}
\] (5.11)

As Eq. (5.11) shows, the number of impulses in the total shaper is the multiplication of the number of impulses in each individual shaper. The length of the total shaper is also summation of the individual shaper lengths.

A step input with the magnitude of 70cm was used as the unshaped reference command. Figure 5.6 shows the simulated motor position and oscillator vibrations for the three cases, i.e. unshaped, single-mode shaped, and multi-mode shaped command.
Figure 5.6: Simulated motor position and oscillator tip position subject to the shaped and unshaped step inputs.

In order to see the vibrations more clearly, a new variable $v_{osc}(t)$ is defined as the oscillator tip vibration which can be calculated from the difference between the oscillator tip position and motor position:

$$v_{osc}(t) = x_{osc}(t) - x_{sh}(t)$$  \hspace{1cm} (5.12)

Figure 5.7 shows the simulated tip vibrations in the three cases of the experiment. It can be seen that in the single-mode shaping, oscillator 1 shows zero vibration whereas oscillator 2 still has large residual oscillations. However, in the case of multi-mode shaping, both oscillators show vibration-free responses. Also note that the shaped command results in smoother transient response at the beginning of the motion, which is because the shaper breaks the input into smaller sub-inputs; as a result, the motion begins with just a portion (depending on the shaper type) of the original command. This smoother start is especially important for avoiding large deflections of the flexible part during the maneuver.
Chapter 5. Implementation and Experimental Results

Figure 5.7: Simulated vibration of the oscillator tip subject to shaped and unshaped step commands.

As explained in Section 3.3, input shapers with positive-amplitude impulses always result in smaller control signals. Figure 5.8 shows the control output for the shaped and unshaped cases described above. It can be seen that the magnitude of the control signal in the shaped case is smaller, especially at the beginning of the motion. Therefore, if the unshaped command does not saturate the drives, the shaped command will not, either. Also note that the shaped control signal has some sharp changes due to the impulses of the shaper affecting the step command; however, these jumps in the control signal result in smooth and vibration free motion of the flexible system. Besides, in real applications where the command is continuous, the shaped control signal will be continuous, too.

5.2.5 Experimental Demonstration

These preliminary sets of experiment were conducted mostly for demonstration purposes. The corresponding videos of the experiment are available online at:
Single-mode:  http://www.youtube.com/watch?v=jBNttYB9kG4
Multi-mode:  http://www.youtube.com/watch?v=XLILHMA4SQQ
Chapter 5. Implementation and Experimental Results

Figure 5.8: Simulated control signal in the shaped and unshaped cases.

5.3 Multi-axis Implementation on the Linear Motor Driven XY Table

A two-axis linear motor driven table shown in Figure 5.9 was used to evaluate the effect of command shaping in multi-axis machines. However, since the electromagnetic force in linear motors is directly applied on the table, there is no mechanical transmission such as ballscrew in between. As a result, linear drives do not exhibit any considerable structural resonance modes. In order to implement input shaping on this machine, artificial modes were imposed on the system by tuning the controllers such that the closed loop response of each axis exhibits a resonance mode at a low frequency. Details of controller design and input shaping implementation is explained in this section.

5.3.1 Experimental Setup

Figure 5.9 shows the high speed linear motor driven XY table used as the experiment platform. The machine has been assembled by former students at MAL. Each axis is driven by a NORMAG BLC08 brushless dc motor; the required current for the motors are provided by two Copley Controls Corp513 servo amplifiers, which are set to work in current mode. The X and Y positions of the table are measured by two
Chapter 5. Implementation and Experimental Results

PULSCALE FM30NA incremental linear encoders. Data acquisition and real time control of the system is implemented on a dSPACE DS1103, which is interfaced with an external PC through ControlDesk© software. The sampling time in all experiment has been chosen as $T_s = 0.1\text{ms}$.

![Figure 5.9: Linear motor driven XY table used for multi-axis implementation of input shaping.](image)

5.3.2 System Modeling

Similar to the modeling procedure presented in Section 5.2.2 for the single axis linear drive, each axis of the XY table can be modeled as illustrated in Figure 5.10. Table 5.5 provides the dynamic parameters of the machine identified by former students of MAL.
**Figure 5.10:** Modeling of linear drives for each axis of the XY table.

**Table 5.5:** Dynamic parameters of the rigid model of the linear motor.

<table>
<thead>
<tr>
<th></th>
<th>X-axis</th>
<th>Y-axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifier gain ($K_a$) [A/V]</td>
<td>0.2857</td>
<td>0.2802</td>
</tr>
<tr>
<td>Motor force constant ($K_i$) [N/A]</td>
<td>99.545</td>
<td>135.117</td>
</tr>
<tr>
<td>Reflected mass ($m$) [Kg]</td>
<td>5.447</td>
<td>31.482</td>
</tr>
<tr>
<td>Viscous friction ($b$) [Kg/s]</td>
<td>18.093</td>
<td>124.938</td>
</tr>
<tr>
<td>Average Coulomb friction ($\mu_c$) [N]</td>
<td>2.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Using the block diagram of the system shown in Figure 5.10, and after plugging in the values of the parameters from Table 5.5, the open loop transfer functions from the input voltage to the table position for each axis can be obtained as:

X Axis: $\frac{X_o(s)}{V_{in}(s)} = \frac{K_aK_{se}}{s(m_x s + b_x)} = \frac{28.44}{5.447s^2 + 18.093s} \text{ [m/V]}$ \hspace{1cm} (5.13)

Y Axis: $\frac{Y_o(s)}{V_{in}(s)} = \frac{K_yK_{sy}}{s(m_y s + b_y)} = \frac{37.86}{31.482s^2 + 124.938s} \text{ [m/V]}$ \hspace{1cm} (5.14)

Since the servo drives are controlled by a dSPACE computer, the command is generated in the form of discrete values at sampling periods $T_s = 0.1 \text{ms}$. The D/A (digital to analogue) output of dSPACE holds these discrete values between the sampling counters. Therefore, in
order to analyze the realistic behavior of the system, the zero-order hold equivalent of the plant (linear drives) must be calculated. These discrete transfer functions can be obtained as:

**X Axis:**

\[
\frac{X_a(z^{-1})}{V_{in}(z^{-1})} = (1-z^{-1})Z\left\{ \frac{K_pK_rK_e}{s(s(m_s+b_s))} \right\} \quad (T_s = 0.1\text{ms})
\]

\[
= \frac{2.61 \times 10^{-5}z^{-1} + 2.61 \times 10^{-5}z^{-2}}{1 - 2z^{-1} + 0.9997z^{-2}} \quad [\text{m/V}]
\]

**Y Axis:**

\[
\frac{X_a(z^{-1})}{V_{in}(z^{-1})} = (1-z^{-1})Z\left\{ \frac{K_pK_rK_e}{s(s(m_s+b_s))} \right\} \quad (T_s = 0.1\text{ms})
\]

\[
= \frac{6.01 \times 10^{-6}z^{-1} + 6.01 \times 10^{-6}z^{-2}}{1 - 2z^{-1} + 0.9996z^{-2}} \quad [\text{m/V}]
\]

Figure 5.11 shows the frequency response function of the open loop system for X and Y axes. As mentioned before, there is no low frequency structural resonance mode in linear motors.

**Figure 5.11:** Frequency response function of the open loop system for each axis of the XY table.
5.3.3 Controller Design

Since linear drives do not exhibit any low frequency modes, an artificial low damping resonance has been imposed on the closed loop of each axis through controller design. These sets of experiment are also designed to demonstrate the general idea of command shaping and contour error compensation; therefore, any simple controller which can give a low damping but stable response of the system would work. More practical cases such as modern State Feedback controller will be presented in the next section for realistic experiments on a commercial CNC machining Center.

In order to clearly demonstrate how input shaping avoids triggering the resonance modes, simple PD controllers are designed such that X and Y axes exhibit low damping modes around 15Hz and 10Hz, respectively. Using the model of the system, MATLAB Control Toolbox was used to tune the parameters of the PD controllers:

\[
X \text{ Axis: } G_{\text{cont}}(s) = K_p + K_d \left( \frac{N_s}{s + N} \right) = 2 + 0.001 \left( \frac{300s}{s + 300} \right) \quad (5.17)
\]

\[
Y \text{ Axis: } G_{\text{cont}}(s) = K_p + K_d \left( \frac{N_s}{s + N} \right) = 4 + 0.01 \left( \frac{300s}{s + 300} \right) \quad (5.18)
\]

where \( K_p \) and \( K_d \) are the proportional and derivative coefficients, respectively, and \( N \) is the filter coefficient for realizing the derivative part. In order to implement the controllers on dSPACE, they have to be discretized at the control intervals \( T_s = 0.1\text{ms} \). Using Tustin method, the discrete transfer functions of the designed controllers can be obtained as:

\[
X \text{ Axis: } G_{\text{cont}}(z^{-1}) = \frac{2.322 - 2.257 z^{-1}}{1 - 0.9678 z^{-1}} \quad (T_s = 0.1\text{ms}) \quad (5.19)
\]

\[
Y \text{ Axis: } G_{\text{cont}}(z^{-1}) = \frac{7.217 - 7.089 z^{-1}}{1 - 0.9678 z^{-1}} \quad (T_s = 0.1\text{ms}) \quad (5.20)
\]

Figure 5.12 demonstrates the block diagram of the closed loop system including both X and Y axes. Note that input shaping module modifies the command ahead of the
closed loop, and thus, it is not necessarily run at real time control intervals. This means that the original command can be shaped and stored in a buffer so that it can be read later at control sampling times.

**Figure 5.12:** Block diagram of the closed loop system for the XY table.

Using the transfer functions of the feed drives derived in Eqs. (5.15) and (5.16), and also plugging the PD controllers from Eqs. (5.19) and (5.20) into the block diagram shown in Figure 5.12, the closed loop transfer functions of the X and Y axes can be obtained as:

**X Axis:**  
\[
G_{CLX}(z^{-1}) = \frac{G_{px}(z^{-1})G_{Cx}(z^{-1})}{1 + G_{px}(z^{-1})G_{Cx}(z^{-1})} \quad (T_s = 0.1\text{ms})
\]

\[
= \left(3.03 + 3.114 z^{-1} - 2.862 z^{-2} - 2.946 z^{-3}\right) \times 10^{-5}
\]
\[
\left/ \left(1 - 2.967 z^{-1} + 2.935 z^{-2} - 0.9675 z^{-3}\right) \right. \quad [\text{mm/mm}]
\]

**Y Axis:**  
\[
G_{CLY}(z^{-1}) = \frac{G_{py}(z^{-1})G_{Cy}(z^{-1})}{1 + G_{py}(z^{-1})G_{Cy}(z^{-1})} \quad (T_s = 0.1\text{ms})
\]

\[
= \left(2.169 + 2.208 z^{-1} - 2.092 z^{-2} - 2.131 z^{-3}\right) \times 10^{-5}
\]
\[
\left/ \left(1 - 2.967 z^{-1} + 2.935 z^{-2} - 0.9675 z^{-3}\right) \right. \quad [\text{mm/mm}]
\]

where \(G_c\) and \(G_p\) are the transfer functions of the controller and plant (linear motor), respectively.
Chapter 5. Implementation and Experimental Results

Figure 5.13 shows the frequency response function of the closed loop systems derived in Eqs. (5.21) and (5.22). It can be seen that X and Y axes exhibit resonance modes at 16.4 Hz and 11.1 Hz, respectively, with damping of 0.04 and 0.11. However, due to model mismatch, the resonance frequencies of the experimental setup are slightly different from these simulated ones. Besides, the abnormally high friction in X axis, which is due to worn out bearings, has resulted in higher observed damping in the X direction compared to the modeled one. In order to evaluate the performance of input shaping in vibration avoidance, the actual resonance frequency and damping ratio exhibited by the machine will be used for shaper design.

![Bode Diagram](image)

**Figure 5.13:** Simulated frequency response function of the closed loop systems including the imposed resonance modes.

Table 5.6 provides the actual resonance parameters of the closed loop systems. Using these parameters, the closed loop transfer functions of the drives can be approximated by flexible second order systems as:

---

95
Chapter 5. Implementation and Experimental Results

X Axis: \[ G_{CLx}(z^{-1}) = \frac{\omega_{nx}^2}{s^2 + 2\zeta_x \omega_{nx} s + \omega_{nx}^2} \quad (T_s = 0.1 \text{ ms}) \]
\[ = \frac{(3.157 + 6.314 z^{-1} + 3.157 z^{-2}) \times 10^{-5}}{1 - 1.997 z^{-1} + 0.9966 z^{-2}} \quad \text{[mm/mm]} \] (5.23)

Y Axis: \[ G_{CLy}(z^{-1}) = \frac{\omega_{ny}^2}{s^2 + 2\zeta_y \omega_{ny} s + \omega_{ny}^2} \quad (T_s = 0.1 \text{ ms}) \]
\[ = \frac{(1.093 + 2.186 z^{-1} + 1.093 z^{-2}) \times 10^{-5}}{1 - 1.999 z^{-1} + 0.9991 z^{-2}} \quad \text{[mm/mm]} \] (5.24)

Table 5.6: Actual resonance parameters of the closed loop systems.

<table>
<thead>
<tr>
<th>Natural Frequency ($\omega_0$)</th>
<th>Damping Ratio ($\zeta$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Axis</td>
<td>17.9 Hz</td>
</tr>
<tr>
<td>Y Axis</td>
<td>10.5 Hz</td>
</tr>
</tbody>
</table>

5.3.4 Input Shaper Design

In order to avoid excitation of the imposed resonance modes, a ZVD shaper has been designed for each axis by plugging the vibration parameters (Table 5.6) into the ZVD shaper Eq. (3.43):

X Axis: \[ \omega_0 = 17.9 \text{ Hz} = 112.5 \text{ rad/s} \] \[ \zeta = 0.15 \] \[ T_d = 2\pi / \omega_0 \sqrt{1 - \zeta^2} = 0.0565 \text{ s} \] \[ K = e^{-\pi \zeta / \sqrt{1 - \zeta^2}} = 0.6209 \]

\[
\begin{bmatrix}
A \\
t
\end{bmatrix}_j = \begin{bmatrix}
1 \\
K^2 + 2K + 1 \\
0
\end{bmatrix}_1, \begin{bmatrix}
2K \\
K^2 + 2K + 1 \\
0.5T_d
\end{bmatrix}_2, \begin{bmatrix}
K^2 \\
K^2 + 2K + 1 \\
T_d
\end{bmatrix}_3, \quad j = 1, 2, 3 \quad (5.25)
\]
\[
= \begin{bmatrix}
0.3806 \\
0.4726 \\
0.1467 \\
0.0283 \\
0.0565
\end{bmatrix}_1, \begin{bmatrix}
0.4726 \\
0.1467 \\
0.0565
\end{bmatrix}_2, \begin{bmatrix}
0.1467 \\
0.0565
\end{bmatrix}_3
\]
Chapter 5. Implementation and Experimental Results

Y Axis: \[ \omega_n = 10.5 \text{ Hz} = 66.0 \text{ rad/s} \quad \zeta = 0.07 \quad \Rightarrow \quad T_d = \frac{2\pi}{\omega_n \sqrt{1 - \zeta^2}} = 0.0952 \text{ s} \]
\[ K = e^{-\pi\zeta/\sqrt{1 - \zeta^2}} = 0.8022 \]

\[
\begin{bmatrix}
\{ A_j \} \\
\{ t_j \}
\end{bmatrix}
= \begin{bmatrix}
\frac{1}{K^2 + 2K + 1} \\
0
\end{bmatrix}, \begin{bmatrix}
\frac{2K}{K^2 + 2K + 1} \\
0.5T_d
\end{bmatrix}, \begin{bmatrix}
\frac{K^2}{K^2 + 2K + 1} \\
T_d
\end{bmatrix}
\]
\[ j = 1, 2, 3 \quad (5.26) \]

and the total shaper which is used for both axes can be obtained by convolving the individual shapers in Eqs. (5.25) and (5.26):

Total Shaper : \[
\begin{bmatrix}
A_j \\
t_j
\end{bmatrix}
= \text{conv} \left( \begin{bmatrix}
A_j \\
t_j
\end{bmatrix}_{\text{one}}, \begin{bmatrix}
A_j \\
t_j
\end{bmatrix}_{\text{two}} \right)
\]

\[
\begin{bmatrix}
A_j \\
t_j
\end{bmatrix}
= \begin{bmatrix}
0.117 \\
0
\end{bmatrix}, \begin{bmatrix}
0.1455 \\
0.0283
\end{bmatrix}, \begin{bmatrix}
0.1880 \\
0.0477
\end{bmatrix}, \begin{bmatrix}
0.0452 \\
0.0565
\end{bmatrix}, \begin{bmatrix}
0.2335 \\
0.0758
\end{bmatrix}, \begin{bmatrix}
0.0754 \\
0.0953
\end{bmatrix}, \begin{bmatrix}
0.0725 \\
0.0936
\end{bmatrix}, \begin{bmatrix}
0.0291 \\
0.1235
\end{bmatrix}, \begin{bmatrix}
0.1518
\end{bmatrix}
\]

\[ j = 1, 2, \ldots, 9 \quad (5.27) \]

The total shaper consists of nine impulses and can be represented as:

\[ F_s(t) = \sum_{j=1}^{9} A_j \delta(t - t_j) \quad (5.28) \]

where \( \delta(t - t_j) \) is the Dirac unit impulse function applied at time delay \( t_j \). Since the reference command is generated in discrete time intervals, the input shaper represented in Eq. (5.28) must be realized in discrete time domain with the same sampling time, i.e. \( T_s = 0.1 \text{ ms} \):

\[ F_s(k) = \sum_{j=1}^{9} A_j \delta(k - k_j) \quad (5.29) \]
where \( k_j = \text{fix}(t_j / T_i) \) is the corresponding number of delayed sample times for the \( j \)th impulse. Therefore, the total shaper designed in Eq. (5.27) can be represented in discrete time domain as:

\[
\text{Total Shaper} : (A, t) = \text{conv}(A_j, k_j)_{\text{osc1}} \times (A_j, k_j)_{\text{osc2}}
\]

\[
\begin{bmatrix}
A, t
\end{bmatrix} = \begin{bmatrix}
0.117, 0.1455, 0.1880, 0.0452, 0.2335, 0.0754, 0.0725, 0.1042, 0.0936, 0.0291, 0.1235, 0.1518
\end{bmatrix}
\]

The percentage of residual vibration due to discretization of ZV/ZVD shaper was derived in Eq. (3.53). Plugging the values of \( \delta T = T_s = 0.1\,\text{ms} \) and \( T_d = 0.1518 \):

\[
V_{prv} \approx \omega_d \frac{\delta T}{4} = \pi \frac{\delta T}{2T_d} = 0.001 = 0.1\%
\]

which is very small, and therefore, the effect of discretization of the shaper can be neglected.

\subsection*{5.3.5 Vibration Avoidance Experiment}

As shown in Figure 5.14, a circular path with the radius of \( R=40\,\text{mm} \) has been used as the reference toolpath for evaluation of input shaping performance. In order to clearly excite the imposed modes of the system while avoiding saturation of the drives, the toolpath was interpolated using constant feedrate profile with amplitude of \( f=80\,\text{mm/s} \).

In the first experiment, contouring of the path was started from a quadrant point on the circle. Axis commands are obtained by interpolating the circle at the control intervals \( T_s = 0.1\,\text{ms} \) with a constant feedrate profile \( f=80\,\text{mm/s} \):

\[
x_r(k) = R\sin(\omega_k t) = R\sin(\omega_k kT_s)
\]
where $\omega_c$ is the contouring frequency of the circle and can be calculated as:

$$T_c = \frac{2\pi R}{f} \text{[s]} \quad \rightarrow \quad \omega_c = \frac{2\pi}{T_c} = \frac{f}{R} = 2 \text{ rad/s}$$

Figure 5.14 shows the toolpath and the corresponding axis commands. Note that the structural modes of the machine can be also excited at contouring frequency $\omega_c$ in repetitive steady state contouring of the circle; however, in this research we are investigating the transient vibrations caused by acceleration and decelerations of the drives, and not the contouring frequency.

The shaped reference command can be obtained by convolving the original axis commands calculated in Eqs. (5.32) and (5.33) with the total shaper represented in Eq.(5.29), that is:

$$x_{r_\omega}(k) = \text{conv}[x_r(k), \text{shaper}] = \sum_{j=1}^{q} A_j x_r(k - k_j) \mathbf{1}(k - k_j)$$

(5.35)
\[ y_{sa}(k) = \text{conv}[y_r(k), \text{shaper}] = \sum_{j=1}^{9} A_j y_r(k-k_j) 1\{k-k_j\} \]

where \( k_j = \text{fix}(t_j / T_i) \) is the corresponding number of delayed sample times for the \( j \)th impulse. The function \( 1\{k-k_j\} \) is the unit step function applied at sampling counter \( k = k_j \) and:

\[
1\{k-k_j\} = \begin{cases} 
1 & k \geq k_j \\
0 & k < k_j 
\end{cases}
\]  

(5.37)

The commands are then sent to the axis servo amplifiers through D/A outputs of dSPACE, and the actual position of the axes are measured by linear encoders. Figure 5.15 shows the experimental axis tracking errors \( (x_{ref} - x_{act}) \) corresponding to the shaped and unshaped commands.

**Figure 5.15:** Measured axis tracking errors for the shaped and unshaped commands in contouring the circle starting from a quadrant point \( (R = 40\text{mm}, f = 80\text{mm/s}) \).
It can be seen in Figure 5.15 that the X axis exhibits significant vibration responding to the unshaped command whereas the shaped input has led to very smooth and vibration-free motion of the drive. The FFT (Fast Fourier Transform) of the X tracking error also shows that the response to the unshaped input has a strong peak at around 18Hz, which is the resonance mode of the closed loop system in the X direction. However, the time domain and FFT response in the Y direction has not exhibited any significant vibration, which can be explained considering the toolpath and axis commands in Figure 5.14; when the circle is contoured starting from the \( y_{\text{min}} \) quadrant, the corresponding axis command for the Y axis will be in the form of Cosine function, which has zero derivative at time zero and thus inherently leads to smooth and vibration free start.

However, in order to evaluate the performance of multi-axis input shaping, it is more practical to generate a trajectory which can excite both axes at the same time. Therefore, the experiment was repeated for the same toolpath but this time contouring of the circle is started from a point which lies on the line at 45 degree relative to the positive X direction. Figure 5.16 illustrates the toolpath and the corresponding axis commands, which can be obtained as:

\[
x_r(k) = R \cos \left( \omega_c kT + \frac{\pi}{4} \right) - \sin \left( \frac{\pi}{4} \right)
\]

\[
y_r(k) = R \sin \left( \omega_c kT + \frac{\pi}{4} \right) - \cos \left( \frac{\pi}{4} \right)
\]

where \( \omega_c \) is the contouring frequency defined and calculated in Eq. (5.34). The axis commands are then shaped using Eqs. (5.35) and (5.36), and sent to the axis servo drives.

Figure 5.17 shows the axis tracking errors for the shaped and unshaped commands. The time domain plots of the tracking errors clearly show the unshaped command has excited both axes and caused significant vibration. The response to the shaped case, on the other hand, results in very smooth and vibration free motion of the drives, which means that the imposed modes of the drives have not been excited.
Figure 5.16: Toolpath and the interpolated axis commands in contouring the circle from a non-quadrant point ($R = 40\,mm$, $f = 80\,mm/s$).

Figure 5.17: Measured axis tracking errors for the shaped and unshaped commands in contouring the circle starting from a non-quadrant point ($R = 40\,mm$, $f = 80\,mm/s$).
The FFT plots in Figure 5.17 also show that the frequency spectrum of the tracking errors in the X and Y directions exhibit significant peaks at around 18Hz and 10.5Hz, which have been eliminated in the shaped response. It should be noted that the amplitude of vibrations are normally much smaller in real applications; however, in order to clearly demonstrate the feasibility of input shaping, the experiments have been conducted for exaggerated cases. Realistic experiment will be presented in the next section.

The controller output, which is shown as $V_{in}$ in Figure 5.12, was also saved during the experiment. Figure 5.18 shows the controller voltage signal for the shaped and unshaped commands. While active damping techniques normally result in very active actuation signals with high frequency content, the experimental result proves that the implemented input shaping has led to much smoother actuation of the drives with less resonance frequency content.

**Figure 5.18:** Measured axis control signals for the shaped and unshaped commands in contouring the circle starting from a non-quadrant point ($R = 40\text{mm}$, $f = 80\text{mm/s}$).
5.3.6 Contour Error Compensation Experiment

Figure 5.19 shows the magnitude and axis components of path distortion caused by input shaping. It can be seen that the shaped toolpath starts and ends on the unshaped original circle but deviates towards inside, resulting in a smaller distorted circle.

![Toolpath Distortion by Input Shaping](image1)

![Toolpath Distortion Mapped on the Axes](image2)

Figure 5.19: Magnitude and axial components of shaping distortion in contouring the circle starting from a non-quadrant point ($R = 40\, \text{mm}$, $f = 80\, \text{mm/s}$).

As shown in Figure 5.20, the total experimental contour error caused by both shaping distortion and limited bandwidth of servo drives can be calculated from the measured positions of the drives. It can be observed that the time domain contour error shows a lot of vibrations in the unshaped case whereas the response to the shaped command exhibits significantly smoother and vibration-free motion. The small jumps corresponding to the circle quadrants are due to the friction effect at direction reversal of the drives. The FFT of the contour error also exhibits two peaks at around 18Hz and...
10.5\text{Hz}, which have been successfully avoided in the shaped case. However, it can be observed in Figure 5.20 that the mean value of contour error has increased to almost 180\,\mu m in the shaped case while the mean value of contour error is significantly smaller in the unshaped case. This increase is due to the trajectory distortion effect of input shaping and has to be compensated.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{contour_error.png}
\caption{Experimental contour error caused by input shaping and drive dynamics in contouring the circle starting from a non-quadrant point ($R=40\,\text{mm}, f=80\,\text{mm/s}$).}
\end{figure}

Figure 5.21 shows the axial correction signals calculated by the contour error compensation module described in Section 4.5. In order to make sure that these added signals do not bring any exciting harmonics to the shaped command, they have been low pass filtered at 3\text{Hz}. The axial compensation signals are then added to the shaped commands to obtain the shaped-compensated commands:

\begin{align}
\text{rsc}_x(k) &= \text{rs}_x(k) + \varepsilon_{cx}(k) \\
\text{rsc}_y(k) &= \text{rs}_y(k) + \varepsilon_{cy}(k)
\end{align}

where $\text{rs}_x$ and $\text{rs}_y$ represent the shaped and shaped-compensated commands, respectively, and $\varepsilon_{cx}$ is also the correction signal. The modified commands are then sent to the machine to achieve accurate and vibration-free contouring of the circle.
Figure 5.21: Axial contour error compensation signals (low-pass filtered at 3 Hz) in contouring the circle starting from a non-quadrant point ($R = 40\, \text{mm}$, $f = 80\, \text{mm/s}$).

Figure 5.22 compares the experimental contour error in the three investigated cases, namely unshaped, shaped, and shaped-compensated command. It can be clearly seen that the shaped-compensated command has led to vibration-free contouring of the circle while still keeping the accuracy of the motions.

Figure 5.22: Contour error comparison in following a circular path for the three cases: unshaped, shaped, and shaped-compensated commands ($R = 40\, \text{mm}$, $f = 80\, \text{mm/s}$).
In order to make sure that the compensated command does not saturate the drives, the control signals for the three mentioned cases are plotted in Figure 5.23. It can be seen that the required actuation for the shaped and shaped-compensated commands are almost the same owing to the fact that the additional compensating terms only correct the small distortion of the toolpath, but does not significantly increase the demand from the servo drives.

**Figure 5.23:** Control signal comparison in following a circular path for the three cases: unshaped, shaped, and shaped-compensated commands (\( R = 40 \text{mm}, f = 80 \text{mm/s} \)).
5.4 Five-axis Command Shaping with Avoidance of Real Structural Modes

In the sets of experiment presented earlier, input shaping performance for vibration avoidance was tested by either adding a flexible beam on the drive or imposing artificial resonance modes on the closed loop of the system. However, in order to verify the effectiveness of the presented command shaping module in real applications, the five-axis Fadal Machining Center shown in Figure 5.24 was used as the experiment platform; input shaping is used to avoid the measurable structural modes of the system, and contour error compensation technique is employed to correct the distortion of the five-axis toolpath.

5.4.1 Experimental Setup

Figure 5.24 shows the CNC machine tool used as the platform for the realistic experiment. The machine is a Fadal VMC2216, which has originally three Cartesian ballscrew driven axes with XY on the table and Z on the spindle. As shown in the figure, a NIKKEN 5AX-130YA-M rotary table has been mounted on the machine table so that the entire system can be used as a five-axis XYZAC machine.

Figure 5.24: Five-axis Fadal Machining Center used for the realistic experiment.
In order to implement the command shaping method, the built-in CNC of the machine was disabled by setting all the servo amplifiers to work in current mode; this way, the generated torque of the motors can be controlled through the user-defined controllers. Similar to the previous sets of experiment, the real time data acquisition is implemented on a dSPACE DS1103, and the measured data is monitored and saved on an external computer.

In addition, a PCB PIEZOTRONICS accelerometer with sensitivity of $51.2\,\text{mV/g}$ has been used to measure the axial vibrations of the table and spindle. Also a Ferraris sensor, as shown in Figure 5.25, has been mounted at the end of X ballscrew to measure the torsional vibrations of the lead screw. Due to the configuration of the machine, there is no room on Y or Z ballscrews to mount the Ferraris sensor. Therefore, the torsional resonance mode of only X leadscrew will be considered for input shaping.

**Figure 5.25**: Ferraris sensor used for measuring the torsional vibrations of ballscrews.

### 5.4.2 System Modeling

The three translational axes of the machine, i.e. XYZ, are powered by rotary brushless dc motors. As shown in Figure 5.26, the rotation of the motor is converted to the linear motion of the table using ballscrew transmission. The pitch of the X, Y and Z leadscrews are $10\,\text{mm}$, $10\,\text{mm}$, and $8\,\text{mm}$, respectively.
Figure 5.26: General configuration of ballscrew driven axes [Erkorkmaz, 2003].

Similarly, the rotary table is driven by two YASKAWA dc motors cascaded by gearbox to reduce the rotational speed. Therefore, the rigid body dynamics of all axes can be simulated using the model shown in Figure 5.27. The corresponding parameters are provided in Table 5.7.

Figure 5.27: Modeling the dynamic of rotary motors-driven feed drives.

Using the least square identification method proposed by Erkorkmaz and Altintas [14], the parameters of the drives were identified by a former student of MAL. Table 5.7 summarizes these parameters. In order to design a state feedback controller, which is explained in the next section, the state space model of the system is required. Based on the block diagram in Figure 5.27, the axis position \( x_a \) and motor velocity \( \omega_e \) can be written as:

\[
\omega_e(s) = \frac{1}{J e s + B_e} \left( K_i K_d V_{in}(s) - T_d(s) \right)
\]  

(5.42)
Chapter 5. Implementation and Experimental Results

\[ X_a(s) = \frac{r_g}{s} \omega_a(s) = \frac{r_g}{s(J_s + B_e)} \left( K_i K_a V_{in}(s) - T_d(s) \right) \]  

\[ (5.43) \]

which can be re-arranged in time domain as:

\[ \dot{\omega}_a(t) = -\frac{B_e}{J_e} \omega_a(t) + \frac{K_i K_a}{J_e} V_{in}(t) - \frac{1}{J_e} T_d(t) \]  

\[ (5.44) \]

\[ \dot{x}_a(t) = r_g \omega_a(t) \]  

\[ (5.45) \]

Therefore, the state space model of the system can be written as:

\[
\begin{bmatrix}
\dot{x}_a(t) \\
\dot{\omega}_a(t)
\end{bmatrix} =
\begin{bmatrix}
0 & r_g \\
0 & -\frac{B_e}{J_e}
\end{bmatrix}
\begin{bmatrix}
x_a(t) \\
\omega_a(t)
\end{bmatrix} +
\begin{bmatrix}
0 & 0 \\
\frac{K_i K_a}{J_e} & -\frac{1}{J_e}
\end{bmatrix}
\begin{bmatrix}
V_{in}(t) \\
T_d(t)
\end{bmatrix}
\]

\[ (5.46) \]

\[ y(t) =
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
x_a(t) \\
\omega_a(t)
\end{bmatrix} +
\begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
V_{in}(t) \\
T_d(t)
\end{bmatrix}
\]

Table 5.7: Dynamic parameters of the rigid model of the drives

<table>
<thead>
<tr>
<th>X-axis</th>
<th>Y-axis</th>
<th>Z-axis</th>
<th>A-axis</th>
<th>C-axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifier gain ($K_a$) [A/V]</td>
<td>6.5723</td>
<td>6.2274</td>
<td>6.4841</td>
<td>1.27</td>
</tr>
<tr>
<td>Motor force constant ($K_v$) [N/A]</td>
<td>0.4769</td>
<td>0.4769</td>
<td>0.4769</td>
<td>1.0</td>
</tr>
<tr>
<td>Reflected inertia ($J_e$) [Kgm^2]</td>
<td>0.0070</td>
<td>0.0082</td>
<td>0.0077</td>
<td>0.00022</td>
</tr>
<tr>
<td>Viscous friction ($B_e$) [Kg/s]</td>
<td>0.0236</td>
<td>0.0430</td>
<td>0.0323</td>
<td>0.00354</td>
</tr>
<tr>
<td>Positive Coulomb friction ($\mu^+$) [N]</td>
<td>2.58</td>
<td>1.30</td>
<td>1.60</td>
<td>0.1969</td>
</tr>
<tr>
<td>Negative Coulomb friction ($\mu^-$) [N]</td>
<td>-1.77</td>
<td>-1.20</td>
<td>-1.95</td>
<td>-0.1712</td>
</tr>
<tr>
<td>Gear ratio ($r_g$) [mm/rad] (or [rad/rad])</td>
<td>1.5915</td>
<td>1.5915</td>
<td>1.2732</td>
<td>1/180</td>
</tr>
</tbody>
</table>
However, since the control system is implemented on digital computer (dSPACE), the state space model derived in Eq. (5.46) must be transformed to discrete time domain. If the sampling time of control interval is $T_s$, the discrete state space model can be obtained as:

\[
\begin{bmatrix}
    x_a(k+1) \\
    \omega_a(k+1)
\end{bmatrix} = A_d \begin{bmatrix}
    x_a(k) \\
    \omega_a(k)
\end{bmatrix} + B_d \begin{bmatrix}
    V_{in}(k) \\
    T_d(k)
\end{bmatrix}
\]

\[
y(k) = C_d \begin{bmatrix}
    x_a(k) \\
    \omega_a(k)
\end{bmatrix} + D_d \begin{bmatrix}
    V_{in}(k) \\
    T_d(k)
\end{bmatrix}
\]

(5.47)

where

\[
A_d = e^{A_T} = L^{-1}\{sI - A\}
\]

(5.48)

\[
B_d = \left(\int_{\tau=0}^{T} e^{A_T} d\tau\right) B
\]

(5.49)

\[
C_d = C, \quad D_d = D
\]

(5.50)

The friction torque $T_d$ is a nonlinear function, and in the simplest way it can be modeled as $T_d = \mu \text{sgn}(\omega)$, where $\mu$ is the Coulomb friction coefficient. However, state feedback controller, which is designed in the next section, can only include the linear model. Therefore, only the first column of $B$ and $D$ matrices are considered in the model.

Substituting the parameters from Table 5.7 into the discrete state space model in Eq. (5.47), the corresponding state and input matrices for the five axes can be obtained as:

\[
\text{X Axis:} \quad A_d = \begin{bmatrix}
    1.0 & 1.5912 \times 10^{-4} \\
    0 & 0.9996
\end{bmatrix} \quad B_d = \begin{bmatrix}
    3.5618 \times 10^{-6} \\
    4.7510 \times 10^{-2}
\end{bmatrix}
\]

(5.51)

\[
\text{Y Axis:} \quad A_d = \begin{bmatrix}
    1.0 & 1.5911 \times 10^{-4} \\
    0 & 0.9948
\end{bmatrix} \quad B_d = \begin{bmatrix}
    2.8856 \times 10^{-6} \\
    3.6251 \times 10^{-2}
\end{bmatrix}
\]

(5.52)
Chapter 5. Implementation and Experimental Results

Z Axis: \[ A_d = \begin{bmatrix} 1.0 & 1.2729 \times 10^{-4} \\ 0 & 0.9958 \end{bmatrix} \quad B_d = \begin{bmatrix} 2.5661 \times 10^{-6} \\ 4.0300 \times 10^{-2} \end{bmatrix} \] \hspace{1cm} (5.53)

A Axis: \[ A_d = \begin{bmatrix} 1.0 & 5.5111 \times 10^{-7} \\ 0 & 0.9984 \end{bmatrix} \quad B_d = \begin{bmatrix} 5.3784 \times 10^{-8} \\ 0.1935 \end{bmatrix} \] \hspace{1cm} (5.54)

C Axis: \[ A_d = \begin{bmatrix} 1.0 & 1.1100 \times 10^{-6} \\ 0 & 0.9981 \end{bmatrix} \quad B_d = \begin{bmatrix} 2.8391 \times 10^{-7} \\ 0.5106 \end{bmatrix} \] \hspace{1cm} (5.55)

and also the output matrices for all axes are:

\[ C_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad D_d = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \] \hspace{1cm} (5.56)

5.4.3 Controller and Observer Design

In control systems of machine tool feed drives, a better performance can be achieved by using both position and velocity in the feedback. For example, in commercial CNC machine tools, it is common to use cascaded controller, which has an inner PI-controlled velocity loop and outer P-controlled position loop. Another method to employ the velocity feedback is by using modern control techniques which are based on the state space model of the system. In order to control the five axes of the machine, a simple state feedback controller has been designed for each axis. Figure 5.28 demonstrates the general block diagram of the closed loop system. Note that unlike active damping techniques, only the rigid body dynamic of the drives are considered for the controller design.

![Figure 5.28: Closed loop control system of the feed drives with state feedback controller.](image)
Based on the linear state space models derived in Eqs. (5.51)-(5.55), the control loop of each axis has two states, i.e. the position $x_a(t)$ and velocity $\omega_a(t)$. Designing the state feedback controller means finding the gain matrix $K_{cnt} = \begin{bmatrix} K_{pos} & K_{vel} \end{bmatrix}$, which is then used to calculate the voltage command sent to the amplifiers:

$$V_{in}(k) = K_{cont} \left( X_{ref}(k) - X_a(k) \right) = \begin{bmatrix} K_{pos} & K_{vel} \end{bmatrix} \begin{bmatrix} x_{ref}(k) - x_a(k) \\ \omega_{ref}(k) - \omega_a(k) \end{bmatrix}$$

(5.57)

In order to achieve a high bandwidth controller, the required rise time for the closed loop system of each axis is chosen as $t_r = 0.01s$; the exhibited overshoot of the system is also limited to $OS = 5\%$. Based on the transfer function derived in Eq. (5.43), the closed loop system of the drive has two poles, which can be placed at the locations corresponding to the desired design specifications. The damping and natural frequency of the closed loop poles can be calculated as:

$$\zeta_{pp} = \frac{\ln(\text{OS})^2}{\pi^2 + (\ln(\text{OS}))^2} = 0.69$$

(5.58)

$$\omega_{pp} \approx \frac{2.16 \zeta_{pp} + 0.6}{t_r} = 209.1 \text{rad/s} = 33.3 \text{Hz}$$

(5.59)

and thus the poles in the continuous domain are obtained as:

$$s_{1,2} = -\zeta_{pp} \omega_{pp} \pm j \omega_{pp} \sqrt{1 - \zeta_{pp}^2} = -144.28 \pm j151.30$$

(5.60)

Since the control system is run in $T_s = 0.1ms$ sampling time, the corresponding desired poles in the discrete time can be derived as:

$$z_{1,2} = e^{s_{1,2}T_s} = 0.9856 \pm j0.0149$$

(5.61)

In order to calculate the controller gain $K_{cnt} = \begin{bmatrix} K_{pos} & K_{vel} \end{bmatrix}$ such that the closed loop poles are placed at the desired locations, there are several techniques available in the literature. Since this research focuses on the command shaping part and not the controller, the
theoretical details are not presented here; instead, the MATLAB command “\texttt{place}” is used to obtain the controller gain:

$$K_{\text{cnt}} = \begin{bmatrix} K_{\text{pos}} & K_{\text{vel}} \end{bmatrix} = \texttt{place} \left( A_d, B_d, \begin{bmatrix} z_1, z_2 \end{bmatrix} \right);$$  \hfill (5.62)

Using the discrete state space models of the drives in Eqs. (5.51)-(5.55), and the desired location of the closed loop poles in Eq. (5.61), the controller gains for the five axes can be obtained using Eq. (5.62):

- **X Axis:** $K_{\text{cnt}} = \begin{bmatrix} 53.1934 & 0.5957 \end{bmatrix}$ \hfill (5.63)
- **Y Axis:** $K_{\text{cnt}} = \begin{bmatrix} 65.7219 & 0.7307 \end{bmatrix}$ \hfill (5.64)
- **Z Axis:** $K_{\text{cnt}} = \begin{bmatrix} 73.8632 & 0.6596 \end{bmatrix}$ \hfill (5.65)
- **A Axis:** $K_{\text{cnt}} = \begin{bmatrix} 89.9194 & 0.3656 \end{bmatrix}$ \hfill (5.66)
- **C Axis:** $K_{\text{cnt}} = \begin{bmatrix} 34.1177 & 0.1381 \end{bmatrix}$ \hfill (5.67)

Finally, the command control voltage sent to the amplifiers can be calculated as:

$$V_{\text{in}}(k) = K_{\text{cont}} \left( X_{\text{ref}}(k) - X_a(k) \right) = \begin{bmatrix} K_{\text{pos}} & K_{\text{vel}} \end{bmatrix} \begin{bmatrix} x_{\text{ref}}(k) - x_a(k) \\ \omega_{\text{ref}}(k) - \omega_a(k) \end{bmatrix}$$ \hfill (5.68)

where position $x_a$ is measured by the linear encoder, and velocity $\omega_a$ is obtained from the tachometers of the motors. However, since the measured data is normally noisy, a fixed-gain Kalman filter is used to smoothen the signal and improve the control performance. The observer equation which estimates the states can be written as:

$$\begin{bmatrix} \hat{x}_a(k+1) \\ \hat{\omega}_a(k+1) \end{bmatrix} = A_d \begin{bmatrix} \hat{x}_a(k) \\ \hat{\omega}_a(k) \end{bmatrix} + B_d V_{\text{in}}(k) + K_{KF} C_d \begin{bmatrix} x_a(k) - \hat{x}_a(k) \\ \omega_a(k) - \hat{\omega}_a(k) \end{bmatrix}$$ \hfill (5.69)
where \( x_a(k) \) and \( \omega_a(k) \) are the actual measured position and velocity; \( \hat{x}_a(k) \) and \( \hat{\omega}_a(k) \) are also the estimated states of the system calculated by the observer. In order to find the Kalman gain \( K_{KF} \), the variance of the noise of encoders and tachometers are required. The procedure of designing the Kalman filter can be found in details in [30], but for the sake of conciseness, only the final results are presented here:

\[
\begin{align*}
\text{X Axis:} & \quad K_{obs} = \begin{bmatrix} 0.0913 & 1.1 \times 10^{-4} \\ 44.2929 & 0.1625 \end{bmatrix} \\
\text{Y Axis:} & \quad K_{obs} = \begin{bmatrix} 0.0909 & 1.072 \times 10^{-4} \\ 43.1568 & 0.1447 \end{bmatrix} \\
\text{Z Axis:} & \quad K_{obs} = \begin{bmatrix} 0.0911 & 8.693 \times 10^{-5} \\ 54.6914 & 0.1535 \end{bmatrix} \\
\text{A Axis:} & \quad K_{obs} = \begin{bmatrix} 0.0924 & 7.2925 \times 10^{-5} \\ 74.3783 & 0.3293 \end{bmatrix} \\
\text{C Axis:} & \quad K_{obs} = \begin{bmatrix} 0.0927 & 6.8602 \times 10^{-5} \\ 69.9690 & 0.4864 \end{bmatrix}
\end{align*}
\] (5.70-5.74)

Note that in order to calculate the control voltage in Eq. (5.68), the estimated state values obtained from Kalman filter are used instead of the raw measurement data. Therefore, the estimation equation in Eq. (5.69) can be re-written as:

\[
\begin{bmatrix} \hat{x}_a(k+1) \\ \hat{\omega}_a(k+1) \end{bmatrix} = A_d \begin{bmatrix} \hat{x}_a(k) \\ \hat{\omega}_a(k) \end{bmatrix} + B_d K_{cm} \begin{bmatrix} x_{ref}(k) - \hat{x}_a(k) \\ \omega_{ref}(k) - \hat{\omega}_a(k) \end{bmatrix} + K_{KF} C_d \begin{bmatrix} x_a(k) - \hat{x}_a(k) \\ \omega_a(k) - \hat{\omega}_a(k) \end{bmatrix}
\] (5.75)

where the reference velocity \( \omega_{ref}(k) \) can be obtained by taking the derivative of position \( x_{ref}(k) \). Figure 5.29 demonstrates the entire closed control loop of the axis feed drives.
Figure 5.29: Closed loop control system of the feed drives with state feedback controller and Kalman filter.

5.4.4 Identification of Structural Resonance Modes and Designing Input Shapers

Structural modes of the machine are identified experimentally using the feed drive input voltage and measured structural vibrations. Since the Fadal machining center used for the experiment has very stiff structure, high acceleration trajectories are required to excite the structural modes. However, the feed drive motors are not powerful enough to provide very high torque. Based on simulations and experiments, maximum acceleration of 2000 mm/s² can be achieved without saturation of the servo amplifiers. In order to identify structural modes of the drives, a simple linear trajectory with maximum acceleration of 2000 mm/s² is given to each axis individually, and the vibrations are measured using accelerometer. However, based on the experimental results, only X and Z axes exhibit measurable axial vibrations; Y axis has a very short ball screw and thus stiff dynamics. Besides, for measuring the torsional mode of the ball screws, only X axis has enough space for mounting the Ferraris sensor.

Figure 5.30 shows the FFT of the measured transient acceleration signals at the beginning of the motion. Note that the peak at around 40 Hz, which has appeared in all plots, is due to the controller dynamics and is not a structural mode. Based on the peak responses in the FFT plots, X axis has an axial mode at \( \omega_d = 95 \) Hz and a lead screw torsional mode at 370 Hz. Z axis also exhibits two clear structural modes around 140 Hz
and 215Hz. Note that the machine might have other low frequency structural modes as well but they cannot be excited and measured reliably with the available hardware.

In order to avoid excitation of the detected structural modes, a ZVD shaper defined in Eq. (3.43) is designed for each resonance frequency. As discussed in Section 3.4.4, ZVD shaper is highly robust to model mismatch in damping. Therefore, the damping ratio of all detected modes are estimated as $\zeta = 0.2$, which is reasonable for structural modes of feed drives. Thus, the shapers can be obtained as:

$$X - \text{axial mode (95Hz):} \quad \begin{bmatrix} A \\ t_j \end{bmatrix} = \begin{bmatrix} 0.4291 \\ 0 \\ \end{bmatrix}_1, \begin{bmatrix} 0.4519 \\ 0.0052 \\ \end{bmatrix}_2, \begin{bmatrix} 0.1190 \\ 0.0105 \\ \end{bmatrix}_3$$

(5.76)

$$X - \text{torsional mode (370 Hz):} \quad \begin{bmatrix} A \\ t_j \end{bmatrix} = \begin{bmatrix} 0.4291 \\ 0 \\ \end{bmatrix}_1, \begin{bmatrix} 0.4519 \\ 0.0013 \\ \end{bmatrix}_2, \begin{bmatrix} 0.1190 \\ 0.0027 \\ \end{bmatrix}_3$$

(5.77)

$$Z - \text{axial mode (140Hz):} \quad \begin{bmatrix} A \\ t_j \end{bmatrix} = \begin{bmatrix} 0.4291 \\ 0 \\ \end{bmatrix}_1, \begin{bmatrix} 0.4519 \\ 0.0035 \\ \end{bmatrix}_2, \begin{bmatrix} 0.1190 \\ 0.0071 \\ \end{bmatrix}_3$$

(5.78)

$$Z - \text{axial mode (215Hz):} \quad \begin{bmatrix} A \\ t_j \end{bmatrix} = \begin{bmatrix} 0.4291 \\ 0 \\ \end{bmatrix}_1, \begin{bmatrix} 0.4519 \\ 0.0023 \\ \end{bmatrix}_2, \begin{bmatrix} 0.1190 \\ 0.0046 \\ \end{bmatrix}_3$$

(5.79)

Figure 5.30 also shows the FFT of the acceleration signal (the same time length) when the axis command is shaped with the corresponding shaper designed for that axis. All FFT plots clearly show that the peak responses corresponding to the structural modes have been greatly reduced in the shaped case.

### 5.4.5 Implementation of Contour Error Compensation Module

In order to implement the proposed contour error compensation technique, a sample five-axis helical toolpath is contoured by the machine. The discrete geometrical points describing the path and tool orientation have been generated by the following formulations:
Figure 5.30: FFT of the transient acceleration of the drives subject to the shaped and unshaped linear trajectory (maximum acceleration \( a = 2000 \text{mm/s}^2 \)).

\[
\begin{align*}
  u &= 0 : 10^{-5} : 2\pi \\
  U_{x_d} &= 40(1 - \cos u) \\
  U_{y_d} &= 40(\sin u) \\
  U_{z_d} &= 3u \\
  U_{a_d} &= -\arcsin \left( \frac{\sin u}{\sqrt{5}} \right) \\
  U_{c_d} &= -0.2 \left( \tan \left( \frac{\cos u}{\sqrt{2}} \right) - \tan \left( \frac{1}{\sqrt{2}} \right) \right)
\end{align*}
\]
where \( u \) is the toolpath parameter, \( U_x, U_y, U_z \) are the tooltip positions in the workpiece frame, and \( \alpha_d, \phi_d \) are the primary and secondary angles describing the tool orientation in the workpiece frame. Figure 5.31 demonstrates this sample toolpath.

![Figure 5.31](image)

**Figure 5.31:** Sample five-axis helical toolpath used for contour error experiment.

In order for the machine to be able to follow the path with reasonable contour error, the mentioned toolpath has been interpolated by a cubic acceleration profile, which is commonly used in commercial CNC machine tools. The feedrate along the path is set to \( f = 150 \text{ mm/s} \), and the maximum acceleration and jerk are limited to \( a = 2,000 \text{ mm/s}^2 \) and \( j = 10,000 \text{ mm/s}^3 \), respectively, with the sampling time of \( T_s = 1 \text{ ms} \). Considering the size and inertia of the machine, this is relatively an aggressive high speed trajectory.

The interpolated points \([P_{x_d}, P_{y_d}, P_{z_d}, \theta_{x_d}, \theta_{c_d}]_w\) obtained above are currently described in the workpiece frame. Inverse kinematic of the machine presented in Eq. (4.22) is used to obtain the corresponding unshaped axis commands \([x_r, y_r, z_r, a_r, c_r]\). Figure 5.32 demonstrates the interpolated trajectories described in the workpiece frame (solid line) as well as the axis commands described in the machine frame (dash line).
Figure 5.32: Trajectory components described in the workpiece frame (solid line) and machine axis commands (dash-line) in contouring a five-axis helical toolpath ($f = 150 \text{ mm/s}$).

In order to verify that the model derived in Section 5.4.2 can closely predict the response of the machine, the generated unshaped axis commands were sent to the servo drives, and the actual axis positions were measured using encoders. Figure 5.33 compares the simulated and measured tracking errors. It can be observed that the derived model can predict the rigid body dynamic of the machine fairly accurately; only in regions where one
or more axes change direction or move very slowly, slight deviation between the simulation and experiment is observed. This is mainly due to the nonlinear effect of coulomb friction (Stribecck effect) which is not considered in the model.

Figure 5.33: Simulated (dash-line) and measured (solid) axis tracking errors in contouring the five-axis helical toolpath (\( f = 150 \text{ mm/s} \)).

In order to calculate the resultant tooltip contour errors \( \epsilon = [\epsilon_{xa}, \epsilon_{ya}, \epsilon_{za}]_w^T \), direct kinematic of the machine derived in Eq. (4.27) is used to obtain the actual tool pose \([Px, Py, Pz, \theta a, \theta c]_w\) (described in the workpiece frame) from the actual unshaped axis
positions \([x_a, y_a, z_a, \alpha_a, \gamma_a]\). This error is caused only by the dynamic of the machine; on the other hand, if the input shaper designed in the previous section is also applied on the axis commands, additional contour error will be incurred due to distortion of the shaped toolpath. This distortion can be calculated by comparing the shaped and unshaped reference toolpaths described in the workpiece frame. Figure 5.34 shows these two types of contour errors in following the designed five-axis helical toolpath.

![Figure 5.34: Machine-dependent contour error (measured-unshaped) and toolpath distortion caused by input shaping in contouring the five-axis helical toolpath (\(f = 150\) mm/s).](image)

The proposed contour error compensation module can be employed not only to correct the reference distortion caused by input shaping but also to reduce the machine-dependent contour errors. The procedure has been explained thoroughly in Section 4.5, and the schematic block diagram of the algorithm is illustrated in Figure 4.10. The models of the feed drives are used to predict the response of each axis to the shaped axis commands. Next, using inverse kinematic of the machine derived in Eq. (4.32), the actual shaped tool pose (defined in the workpiece frame) is obtained. The predicted tooltip contour error
\( \left[ \tilde{e}_{xs}, \tilde{e}_{ys}, \tilde{e}_{zs} \right] \) can be also calculated by comparing the predicted actual shaped tool pose against the original desired one. Figure 5.35 shows the components of the predicted contour error described in the workpiece frame.

![Graphs showing predicted contour errors](image)

**Figure 5.35:** Predicted (simulated) components of tooltip contour error (in workpiece frame) in contouring the shaped toolpath (\( f = 150 \text{ mm/s} \)).

However, as explained in Section 4.5.1, due to the nonlinear kinematic transformation between the axis commands (in the base frame) and tool pose (in the workpiece frame), the contour error components obtained in Figure 5.35 cannot be directly added to the axis commands as compensation terms. In order to compensate for these tooltip contour errors, the corresponding axis compensation signals are calculated using
Eqs. (4.34)-(4.36). Since the lowest identified resonance mode of the feed drives is 95 Hz, the compensation signals are low-pass filtered at 10 Hz to avoid adding excitation harmonics to the shaped commands. Figure 5.36 shows the final compensation signals calculated by the proposed command shaping module. These signals are added to the shaped axis commands to compensate for the contour errors caused by the machine dynamic and reference distortion.

**Figure 5.36:** Axis compensation signals (simulation) low-pass filtered at 10 Hz.

Figure 5.37 compares the measured contour errors for the unshaped and shaped-compensated toolpaths. It can be clearly seen that the mean value of contouring error has decreased significantly in the case of shaped-compensated command, which means that the proposed module has successfully reduce the low-harmonics of contour error. It should be noted that the ripples in the contour error are due to the controller dynamics, and are not
structural vibrations. The spikes in the shaped-compensated contour error are also due to the coulomb friction acting as a step disturbance at the direction reversal of feed drives. These spikes can be greatly reduced by implementation of disturbance observer techniques, which is part of the controller design and is left for future work. It is worth reminding that the integrated command shaping module is a feed forward technique for reference modification, and does not change the closed loop tracking performance of the drives.

![Graph](image)

**Figure 5.37:** Measured contour errors for the unshaped and shaped-compensated five-axis helical toolpath ($f = 150$ mm/s).

Theoretically, a well-designed control loop including disturbance observer along with the proposed command shaping module can result in contour errors as small as few multiples of encoder resolution, provided that the original unshaped trajectory also respects the axis limits and does not saturate the drives. For example, Figure 5.38 shows the simulated contour errors for the same system but with simple feedforward friction compensation. It can be seen that the theoretical contour error has been significantly reduced in the shaped-compensated case. The relatively larger error at the initial and
ending points is due to the acceleration and deceleration of the axes, and can be reduced by either employing more aggressive controllers such as Sliding mode, or using trajectory profiles with higher continuity such as cubic jerk.

![Simulated Contour Error with Friction Compensation](image)

**Figure 5.38:** Simulated contour error for the unshaped and shaped-compensated commands when the axis control loops include feedforward friction compensation (in both cases) ($f = 150 \text{ mm/s}$).

### 5.5 Summary

In this chapter, the implementation and experimental verification of the proposed integrated input shaping and contour error compensation module was presented. First, a single-axis linear drive equipped with a flexible beam as an oscillator was used to show how input shaping can avoid vibration of the oscillator. In the next step, a linear motor-driven XY table was used to investigate the performance of the proposed command shaping method; however, instead of using a flexible beam, artificial resonance modes were injected into the closed loop of the system by tuning low-damping controllers.
Finally, more realistic experiment was conducted on a Fadal machining center to avoid the true structural modes of the machine while improving the contouring accuracy using the pre-compensation technique.
6 Conclusions

A novel command shaping module has been proposed to avoid structural excitation of machine tool feed drives without compromising contouring accuracy. Vibration avoidance is accomplished by applying a sequence of impulses, called an input shaper, on the original axis commands; however, shaping the commands results in distortion of the reference toolpath and causes large contour errors. The shaping distortion is estimated and corrected prior to sending the shaped commands to the servo drives. Contour errors caused by the limited bandwidth of servo drives are also predicted and compensated by using the dynamic model of the machine tool feed drives.

The theory of input shaping method has been reviewed in this thesis, and the mathematical derivations and design procedures have been presented. It has been shown that by imposing different constraints on the shaping sequence, desirable robustness can be achieved. The performance of input shaping technique has been compared against classical lowpass and bandstop filters; and it is shown by simulation that input shapers can significantly outperform frequency-domain filters in terms of vibration reduction, shorter delay, robustness, and simplicity.

However, due to the delay caused by the input shaper, the shaped toolpath exhibits distortions compared to the original reference path. Contour errors caused by shaping distortion are estimated beforehand using shifted Frenet frame, followed by adding correcting signals to the shaped trajectories in order to compensate the path distortion. Due to the limited bandwidth of servo drives, the machine response always exhibits additional contour errors in following a path. If a model which can reliably predict the dominant rigid body dynamics of the machine is available, the machine-dependent contour errors can be also compensated beforehand. A statistical evaluation on effectiveness of pre-compensation has been presented to investigate the conditions which makes it reasonable to employ pre-compensation technique; based on the derived equations, when the predictable contour error (due to input shaping or machine dynamics) is significant compared to the stochastic or non-modeled effects, the pre-compensation technique can be used effectively.
Chapter 6. Conclusions

The proposed integrated input shaping and contour error compensation module has been implemented on several machines to experimentally verify the effectiveness of the method. A basic zero-vibration and derivative input shaper has been used to avoid the vibrations of a flexible oscillator mounted on a linear feed drive subject to a step command. Two-axis input shaping has been implemented on a linear motor driven XY table to avoid excitation of the imposed auxiliary resonance modes without compromising the accuracy in following a circular path. Realistic experiments have been conducted on a commercial five-axis machine tool to avoid real structural vibrations with improved contouring performance. The contributions of the thesis has been published in [4]

There are several directions which can be taken to continue this research. It is essential to investigate how input shaping can be used in conjunction with other trajectory modification techniques such as look-ahead functions or corner smoothening; it must be assured that when used together, they do not interfere or degrade each other’s effect. The implementation of the proposed command shaping module requires access to the CNC of the machine tool. Since commercial machine tools have closed CNCs, it is also beneficial to look for other implementation solutions, e.g. through G-code. Finally, only the pre-compensation of tooltip contour error is studied in this thesis; developing techniques for modeling, prediction and pre-compensation of orientation contour errors is an important direction for future follow ups.
Bibliography


Bibliography


