Model Based Control of Machine Tool Servo Drives

Michael Aaron Stephens

Submitted in total fulfilment of the requirements of the degree of Doctor of Philosophy

March 2012

Department of Mechanical Engineering
The University of Melbourne
Victoria Australia
This thesis investigates the control challenges associated with high precision machine tool servo drives. Machine tools are a fundamental component of modern manufacturing. Essentially every manufactured good is formed, in part, by a machining process. The demands on high precision machine tools are growing at a significant rate as the competing requirements of high accuracy and low cycle time become differentiators between manufacturers. At the heart of the modern machine tool are the motion control elements, which comprise a Computer Numerical Control (CNC) and a number of Digital Servo Drives (DSDs).

The high performance demands on commercial CNC machine tools have led to the widespread adoption of direct-drive servo axes. In cases where the workpiece is manipulated by the axis, the plant dynamics seen by the control system may then vary widely between different workpieces. In practice it is often found that an axis which has been optimally tuned for a given workpiece becomes unstable when a new workpiece with significantly different geometry is loaded. This thesis analyses such a situation. It is shown that conventional modelling approaches, in which the low-order vibrational modes of the axis are represented, but in which details of the current servo are neglected, are unable to predict the experimentally observed limit-cycles. Accurate prediction of these phenomena is obtained by a combination of modal analysis for the mechanical dynamics, and a single-
rate representation of the multirate control system, which takes account of computational delays and the current servo dynamics. As these automation systems become increasingly dependent on simulation and model-based control approaches, there is significant motivation for the development of low order models that capture the key characteristics of a machine tool servo axis.

The benefits of Model Predictive Control (MPC) as a control technique have been well established. However, its application to reference tracking on Digital Servo Drives (DSDs) which typically have very fast update rates is limited by the computational power of present day processors. This thesis presents a novel MPC formulation, which provides a mechanism to trade-off online computation effort with tracking performance, while maintaining stability. This is achieved by introducing a trajectory horizon, which is distinct from the prediction and control horizons typically encountered in MPC formulations. It is shown that increasing the trajectory horizon inherently leads to improved tracking; however, larger horizon lengths also have the unwanted effect of increasing online computation. The proposed MPC formulation is compatible with recently developed explicit MPC solutions, and hence the burden of online optimisation is avoided. The proposed approach (formulated as an explicit MPC solution) is successfully implemented on an industrial machine tool DSD, and is shown to outperform the incumbent approach of cascaded PID control.
DECLARATION

This is to certify that:

(i) the thesis comprises only my original work towards the PhD,
(ii) due acknowledgement has been made in the text to all other material used,
(iii) the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed,

________________________________________
Michael Aaron Stephens
6th March 2012
To my brother Jonathan.

In loving memory.
I would like to acknowledge and thank several people who were instrumental in helping me achieve my goals. First of all, I would like to express my deepest gratitude to my supervisors, Professor Malcolm Good and Associate Professor Chris Manzie. Your ongoing support and tireless devotion to the development of my skills is greatly appreciated. In addition to what you taught me on the subjects of dynamics and control, I take with me many other life and career changing lessons. Most importantly you taught me not to accept just because as the answer.

This research project was partially supported by ANCA Pty. Ltd. as part of their ongoing commitment to both advancement of technology in the area of precision motion control, and the development of their staff. A special thank you to Pat Boland, Pat McCluskey, David Fisher, David McPherson and John Koene for the generous financial support and for having faith in my ability. Thank you also to current and past ANCA colleagues: Troy Robinette, Lijing Qin, Lucas Hale, Roy Ko and Gerard Cullen. You all contributed to the success of this project.

To Joachim Pfleghaar, for his help with the system identification work. Your efforts significantly reduced the time to complete this phase of the project.

I would also like to thank those that I shared an office with at various times during my candidature: Craig Fletcher, Farzad Keynejad, Chih Feng Lee, Tae Soo "Terence" Kim, Rahul Sharmar, Will Moase, Alireza Mohammadi and Denise Lam. You all contributed to a fun and productive working environment, and also to the success of the Rabit Social Club.

To my parents Malcolm and Jan, and my brothers Jonathan and David, thank
you for your ongoing love and support, and for fostering my natural curiosity – a significant part of the motivation to undertake a PhD.

Finally I would like to thank my wife Samantha, for her continuous support and understanding during this, at times, challenging journey. Thank you for being such a wonderfully loving companion.
7.1 Contributions to the Modelling and Analysis of Machine Tool Servo Drives ............................................. 179
7.2 Contributions to the Control of Machine Tool Servo Drives ................................................................. 182
7.3 Further Research Opportunities .............................................................................................................. 185

References .................................................................................................................................................. 187

Nomenclature ........................................................................................................................................... 195

List of Acronyms ...................................................................................................................................... 201

A Background Theory and Additional Literature .................................................................................... 203
    A.1 Friction Modelling ............................................................................................................................. 203
    A.2 Clarke-Park Transformations .............................................................................................................. 210
    A.3 Cascaded State-Space Systems ........................................................................................................... 213
    A.4 System Identification Techniques ..................................................................................................... 215

B Linear Quadratic Control Formulation .................................................................................................... 229
    B.1 Linear Quadratic Control Design ....................................................................................................... 229
    B.2 Plant Model Linearisation .................................................................................................................. 231
    B.3 Linear Quadratic Regulator ............................................................................................................... 232
    B.4 Tracking ............................................................................................................................................ 233
    B.5 Kalman State Estimator ..................................................................................................................... 234
    B.6 Integrator ........................................................................................................................................... 236

C Model Predictive Control Formulation .................................................................................................. 239
    C.1 MPC Quadratic Optimisation Formulation ........................................................................................ 239
    C.2 MPC Constraint Formulation ............................................................................................................. 243
    C.3 Dimensions of MPC Variables ......................................................................................................... 244
    C.4 Solving the MPC Problem .................................................................................................................. 247
# LIST OF FIGURES

1.1 Relief carving of an Egyptian lathe ........................................... 2
1.2 Vaucanson’s lathe ................................................................. 3
1.3 Model of Wilkinson’s boring machine ....................................... 3
1.4 MIT NC milling machine ........................................................ 5
1.5 ANCA RX7 CNC grinding machine ........................................... 6
1.6 ANCA 5DX CNC ................................................................. 7
1.7 ANCA 5DX DSDs ............................................................... 8
1.8 Experimentally observed limit-cycle behaviour ............................ 9
1.9 CNC grinding wheel pack ....................................................... 11
1.10 CNC Grinding – Example Workpieces ..................................... 11
2.1 Classical cascaded PID control structure .................................. 14
2.2 Block diagram of compliantly coupled load .............................. 16
2.3 Bode diagram showing resonance and anti-resonance ............... 17
2.4 Bode diagram of PI controller and plant with mechanical resonance 18
2.5 Pole placement axis tracking control scheme ............................ 32
3.1 Production ANCA RX7 CNC grinding machine .......................... 43
3.2 RX7 CNC grinding machine – electrical cabinet ....................... 43
3.3 CAD model of RX7 CNC grinding machine layout .................... 44
3.4 Small and large workpieces ................................................... 44
3.5 The Fridge test rig ................................................................ 45
3.6 Permanent magnet synchronous motor ..................................... 45
3.7 High-Level servo drive system structure .................................. 46
3.8 Servo drive control algorithm structure ................................... 47
3.9 DSP task scheduling .............................................................. 51
3.10 Intelligent power module with insulated-gate bipolar transistors ... 51
3.11 Full nonlinear mechanical plant model .................................... 52
3.12 Two-pole permanent magnet synchronous motor .................... 53
3.13 RX7 A-axis CAD diagram ...................................................... 56
3.14 Large workpiece CAD diagram ............................................. 56
3.15 FEA mesh for the A-axis with large workpiece ......................... 57
3.16 Finite element modal analysis for the A-axis with large workpiece 58
3.17 Friction model for system identification ................................... 61
3.18 Experimental time response to chirp excitation ....................... 62
3.19 Simulated time response to chirp excitation ............................ 65
3.20 Example of friction model time series data .............................. 67
3.21 Friction model experimental data and model fit ....................... 68
3.22 Effect of Karnopp friction model ........................................... 69
3.23 Reduced-order linear mechanical plant model ......................... 70
3.24 Current loop frequency response data ................................... 71
3.25 Current loop frequency response ............................................. 72
3.26 Effective viscous friction .................................................... 74
3.27 Nonlinear friction model versus effective linear model ............................... 75

4.1 Servo drive control structure excluding current servo ......................... 81
4.2 Open-loop frequency response from current reference to motor velocity – excluding current servo dynamics ........................................ 83
4.3 Closed-loop poles of motor position loop – excluding current servo dynamics ................................................................. 84
4.4 Simulated limit-cycle behaviour .............................................. 85
4.5 Multirate System ................................................................. 86
4.6 Servo drive control algorithm structure including current servo .............. 89
4.7 Open-loop frequency response from current reference to motor velocity – including current servo dynamics .............................. 91
4.8 Closed-loop poles of the motor position loop – including current servo dynamics ............................................................... 92
4.9 Closed-loop poles of the motor position loop – including current servo dynamics and notch-filter .................................................. 93
4.10 Simulated step response .......................................................... 96
4.11 Root locus for closed position loop with default tuning ....................... 98
4.12 Effective viscous friction – close to zero .................................... 99
4.13 Root locus for closed position loop with notch-filter .......................... 100
4.14 Simulated response to a (large) disturbance torque .......................... 101
4.15 Simulated response to a (small) disturbance torque .......................... 102
4.16 Simulated response of existing control architecture – Step .................. 104
4.17 Response of existing control architecture – Sinusoidal ....................... 105
4.18 Settling time for existing control architecture ................................ 106
4.19 Stability boundary for default tuning ........................................ 107
4.20 Experimental results for existing control architecture – Standstill ......... 108
4.21 Experimental results for existing control architecture – Trapezoidal ve-
locity profile ........................................................................... 109

5.1 Time-optimal control result for 250 µs sample period ............................. 115
5.2 Time-optimal control result for 10 µs sample period ........................... 116
5.3 Restated time-optimal control problem ........................................... 117
5.4 Benchmarking test for time-optimal control ..................................... 118
5.5 Benchmarking test for time-optimal control – iterations ........................ 119
5.6 LQR with tracking, Kalman estimator, integrator and friction compensation ........................................................................ 121
5.7 LQGTI Simulated Step Response - Small and Large Workpieces .......... 125
5.8 Benchmarking test for LQGTI controller ......................................... 126
5.9 LQGTI Simulated Response to Sinusoidal Position Reference ................ 129
5.10 LQGTI Simulated Response Highlighting Hunting Due to Nonlinear Friction ......................................................................... 129
5.11 LQGTI Simulated Response – Friction Compensation .......................... 130
5.12 LQGTI Simulated Response to Sinusoidal Position Reference – Axis Re-
versal ..................................................................................... 130
5.13 Simulink Coder Model used for LQGTI ........................................... 132
5.14 LQ Controller implementation on the Fridge ................................... 134
5.15 LQGTI Experimental Response to Constant Position Reference with
Friction Compensation ................................................................. 137
5.16 LQGTI Experimental Response to Constant Position Reference without
Friction Compensation ............................................................... 138
5.17 LQGTI Experimental Response to a Trapezoidal Velocity Profile ...... 139

6.1 Comparison of MPC horizons ......................................................... 147
6.2 Simulated tracking performance for MPC using a FOH-ERT ............. 158
6.3 Simulated tracking performance for a hand tuned cascaded PID con-
troller ....................................................................................... 158
6.4 Comparison of simulated tracking performance between PID and MPC
(10 Hz) .................................................................................... 159
6.5 Comparison of simulated tracking performance between PID and MPC
(1 Hz) ....................................................................................... 159
6.6 Simulated waterfall chart for MPC with ZOH-ERT ......................... 160
6.7 Simulated waterfall chart for MPC with FOH-ERT ......................... 160
6.8 Benchmarking test for MPC controller ............................................ 162
6.9 Benchmarking test for MPC controller without active constraints ...... 162
6.10 Comparison of tracking error and energy usage versus the ratio of ob-
jective function weights .............................................................. 164
6.11 Simulated response for various ratios of the objective function weights. 165
6.12 Explicit MPC Simulink model ....................................................... 167
6.13 The Fridge test rig using EMPC with FOH-ERT ............................ 169
6.14 The Fridge test rig using PID ....................................................... 169
6.15 The Fridge test rig using EMPC with FOH-ERT – large disturbance 170
6.16 RX7 using EMPC with FOH-ERT standstill ............................... 174
6.17 RX7 using EMPC with FOH-ERT to track reference trajectory – constant
velocity ................................................................................... 174
6.18 RX7 using EMPC with FOH-ERT to track reference trajectory – reversal 175
6.19 RX7 using EMPC with FOH-ERT rejection ................................. 175

A.1 True contact between engineering surfaces .................................... 205
A.2 Stribeck curve ........................................................................... 206
A.3 Elastic bristles used to develop the LuGre friction model ................. 208
A.4 Clarke-Park Transformation ....................................................... 210
A.5 Cascaded State-Space Systems ................................................... 214
A.6 System Configurations Used in System Identification .................... 218
A.7 Magnitude Squared Coherence Estimate ($I_f$ to $I_q$) ................. 224
A.8 Bode Plot ($I_f$ to $I_q$) .............................................................. 224
A.9 Two-Inertia Model ................................................................... 225
A.10 Magnitude Squared Coherence Estimate ($I_r$ to $\Omega_m$) ............. 227
A.11 Bode Plot ($I_r$ to $\Omega_m$) .......................................................... 227

B.1 Linear Quadratic Regulator system architecture ........................... 233
B.2 LQ with tracking ...................................................................... 234
LIST OF TABLES

3.1 Default Controller Tuning Parameters ........................................... 49
3.2 Mechanical plant parameters ......................................................... 63
3.3 Damping ratios and natural frequencies of the poles of the mechanical plant with large workpiece ......................................................... 63
3.4 Poles, damping ratios and natural frequencies of mechanical plant with small workpiece ................................................................. 64
3.5 Friction model parameters ............................................................. 67
4.1 Damping ratios and natural frequencies of the poles of the discretised mechanical plant with large workpiece ........................................ 82
4.2 Damping ratios and natural frequencies of the poles of the discretised mechanical plant with small workpiece ........................................ 82
5.1 Time optimal control switching times ............................................... 118
5.2 Comparison of minimum-time optimal control with PID control ........ 120
5.3 Comparison of LQ control with cascaded PID and minimum-time optimal control ................................................................. 127
5.4 Small servo motor plant parameters ............................................... 133
5.5 RX7 A-axis mechanical plant parameters ...................................... 135
6.1 Comparison of MPC control with cascaded PID, minimum-time optimal control and LQ control .......................................................... 163
6.2 RX7 A-axis mechanical plant parameters ...................................... 172
A.1 Mechanical Plant Parameters ........................................................ 228
C.1 Dimensions of plant variables ....................................................... 245
C.2 Dimensions of MPC cost function variables .................................... 245
C.3 Dimensions of MPC constraint variables ........................................ 246
C.4 Dimensions of Hildreth’s QP procedure variables ............................. 249
C.5 Dimensions of Explicit MPC design variables .................................. 251
The automatic machine tool industry is currently worth in excess of US$70bn (Global Industry Analysts 2011). Globally, the demands on high precision machine tools are growing at a significant rate as accuracy and cycle time become differentiators between manufacturers. These two competing demands are largely driven by the performance of the motion control system at the heart of the machine. However, the control algorithms predominately employed have not fundamentally changed since the introduction of numerical control in the 1950s. The aim of this research is to investigate the application of advanced control techniques to machine tools. More specifically, it is motivated by the professional experience of the writer in attempting to deal with challenging operational problems in state-of-the-art machine tools utilising conventional control techniques.

This introduction begins with a brief description of the machine tool, including a timeline of events from its earliest known origins, through its rapid development during the Industrial Revolution, and then more recently, with the advent of Numerical Control. This is followed by a description of an operational problem encountered quite frequently with current machines, the resolution of which motivated the research reported here.

1.1 Development of the Machine Tool and its Controls

1.1.1 Origins of the Machine Tool

The machine tool is by no means a new invention. There is artefact evidence to support use of such devices dating back to 1000 BC (Burstall 1975). The Egyptian lathe shown in Figure 1.1 is the earliest known pictorial representation of such a
INTRODUCTION

Figure 1.1: Relief carving of an Egyptian lathe (Burstall 1975).

device. It was found on a wall in the tomb of Petosiris, a high priest of the 3rd century BC. It shows the man on the left holding a cutting tool, while the man on the right is making the workpiece rotate back and forth by pulling on a cord or leather strap.

In the modern sense of the word, an industrial machine tool is a device used to fabricate hard components (typically metal) by selective removal of material. The first industrial machine tools were developed around the mid 18th century at the beginning of the Industrial Revolution (Gilbert 1958). Some examples of early pioneers in industrial machine tool development are Jacques de Vaucanson and John Wilkinson. Vaucanson, a French engineer and inventor, is credited with developing the first all metal lathe with the cutting tool fixed to a moveable slide (see Figure 1.2). The slide was advanced by a hand-operated leadscrew. Developed in 1751, it was a single purpose machine built to manufacture large iron rollers.

In 1775, John Wilkinson, an English industrialist, developed a novel way of manufacturing cannon barrels. Traditionally cannon barrels had been cast with a core, but Wilkinson proposed casting them solid and boring the core out afterwards. To achieve this he developed a new kind of boring machine (see Figure 1.3), which was capable of producing a circular hole more precise than had previously been possible. The machine was water powered and included gearing,
but still relied on a human operator to move the cutting head forwards and backwards inside the cylinder.

Wilkinson’s machine was also used to produce steam engine cylinders. Due to the accuracy of the bores, it was possible to make a better seal between the cylinder wall and the piston, thus allowing for higher steam pressures. Wilkin-
son became the main supplier of steam engine cylinders to the firm Boulton & Watt (as in James Watt, inventor of the centrifugal governor), who before then had found it impossible to produce cylinders with the necessary degree of accuracy. Thus the machine tool played a pivotal part in the progress of the Industrial Revolution.

In an interesting synergy, while the machine tool had helped advance the development of the steam engine, so too had the steam engine helped advance the development of the machine tool. Provided with a significant power source, machine tools were now able to perform much more demanding tasks, and at far greater speeds. This allowed many manufacturing industries to further progress their production capabilities.

In 1750, the machine tools available to industry had barely advanced beyond those available in the Middle Ages; by 1850 the majority of modern machine tools had been invented.

1.1.2 The Introduction of Numerical Control

With the exception of electricity replacing steam, the next major breakthrough in the development of machine tools occurred soon after World War II, with the introduction of Numerical Control, or NC. In 1952, the Servomechanisms Laboratory at Massachusetts Institute of Technology (MIT) demonstrated the first NC machine tool; it was a three-axis milling machine (see Figure 1.4). This novel approach to machine tool control was patented in 1962 (Forrester et al.).

NC machine tools are automatically operated by numerical commands (a sequence of axes movements) that are received by the processing unit. In the beginning, the commands were typically stored on punch cards or punch tape; however this was soon upgraded to magnetic tape (Reintjes 1991). NC machines made it possible for large quantities of the desired component to be very precisely and efficiently machined in a reliable, repetitive manner. With the incorporation
of computers during the 1960s, Computer Numerical Control (CNC) provided even more flexible processing capabilities.

The CNC machine tool can have a large variety of configurations, but generally consists of two to six individual axes. Figure 1.5 shows an ANCA RX7 CNC Grinding Machine, specifically a Tool and Cutter Grinder (TCG) machine. It is a good example of a typical modern day machine tool. This particular machine has five axes (3 linear and 2 rotary) and is the basis for experimental work that will be conducted as part of this research.

1.1.3 The Modern Machine Tool Control System

There are typically two distinct components that handle motion control within a modern machine tool. The first is the CNC, whose main role is to generate
sequences of axis position commands, and the second is the Servo Drives, which power the axis motors to track the reference generated by the CNC.

The modern day CNC is typically based on an industrial computer. Figure 1.6 shows ANCA’s current generation 5DX CNC. The key component of a CNC is the software; the fundamental elements being the Part Program Interpreter (PPI), which converts a G-code program into physical axis positions, and the Velocity Profile Interpolator (VPI), which interpolates between positions to ensure a smooth trajectory. Another important element is the real-time communication layer, which transmits the axis position commands to the servo drives. Other tasks typically carried out by the CNC include operator interface, Programable Logic Controller (PLC), machine safety, networking and diagnostics.

A Servo Drive’s role is to accept a sequence of axis position commands from the CNC, and apply appropriate voltage levels to the motor terminals so as to locate the axis at the required positions. The Servo Drive is assisted by feedback,
usually position feedback (e.g., encoder) and sometimes also velocity feedback (e.g., tachometer). This feedback, together with the position command, is used by the control algorithm to determine an appropriate control output (voltage). The control algorithm usually includes a range of parameters, which must be tuned to achieve acceptable performance.

Servo Drives have been around since the days of NC. Back then they were analogue devices, which were tuned by adjusting numerous variable resistors. Over the last 20 years the move has been to Digital Servo Drives (DSDs), which use a Digital Signal Processor (DSP) to calculate control outputs. The servo drives shown in Figure 1.7 are ANCA 5DX Digital Servo Drives. There is one drive for each machine axis (i.e., five), and one for the grinding wheel spindle. This research investigation will be primarily concerned with the control actions carried out at the Servo Drive level.

1.2 MOTIVATION FOR THE RESEARCH

The work in this thesis is motivated by a challenge that is currently faced by machine tool builders in industry. To remain competitive, manufacturers must integrate the latest CNC technology into their machines and configure them to
INTRODUCTION

Figure 1.7: ANCA 5DX Digital Servo Drives.

operate with sufficient performance so as to meet the end customers’ expectations. The control architecture used in modern day DSDs has a fixed structure, which is configured and tuned by adjusting a myriad of different parameters. Furthermore, once a machine has left the factory very little, if any, ongoing support is desirable. So the real challenge is for the control engineer to develop the CNC system in such a way so as to be easily configured for a variety of different machine configurations and still provide adequate performance. In spite of this goal, the selection of suitable parameters continues to require the knowledge and skill of an experienced technician / engineer.

In cases where the workpiece is manipulated by the axis, the plant dynamics seen by the control system may vary widely between different workpieces. In practice it is often found that an axis which has been optimally tuned for a given workpiece suffers from a degradation in controller performance (and in some circumstances may even become unstable) when a new workpiece with significantly different geometry is loaded. This thesis analyses and proposes suitable solutions for such a situation. Figure 1.8 depicts the limit-cycle behaviour that is observed when a large workpiece is mounted onto the servo driven headstock of a CNC machine tool, when the axis is under closed-loop position control with
Figure 1.8: Experimentally observed limit-cycle behaviour during stationary operation of the axis loaded with a large workpiece. The position reference is the dashed line and the measured position is the solid line. The motor current saturation limits are also shown. The frequency of the observed oscillations is approximately 410 Hz.

a fixed reference. Note that the default servo tuning was calibrated for a significantly smaller workpiece. The consequence of the rapid oscillation is a severe vibration felt throughout the machine, which is detrimental to the accuracy of the machined workpiece. The motor position only remains bounded due to motor current saturation limits. While it may be possible to re-tune the axis to handle the changed dynamics introduced by the large workpiece, dispatching a suitably trained technician to the end customer’s facility (possibly located across the globe) every time an axis requires re-tuning is economically prohibitive. Furthermore, the new tuning is likely to be unsuitable for use with other workpieces with yet different dynamics.

In developing a solution for this problem, it has become apparent that the underlying mechanism of the observed phenomenon is not well understood in
industry, and that the currently implemented fixes are somewhat ad-hoc. Finally, it is noted that this performance limitation is not restricted to the examples presented in this thesis. The researcher has obtained further anecdotal evidence which indicates that this is a real and ongoing problem in industry.

1.3 THE GRINDING PROCESS

Machine tools produce finished product from blank stock, typically via selective removal of material. Machine tools come in a large variety of configurations. Examples of common machine tools are mills, lathes, hobbing machines, grinders, and profile cutters. The work presented in this thesis is applicable to essentially all types of machine tools. However due to the support provided by ANCA, the industry partner for the project, all simulation and experimental work is carried out on a grinding machine.

Grinding is an abrasive machining process that uses a grinding wheel as the cutting tool (Malkin and Guo 2008). It it typically suited to machining of very hard materials. Figure 1.9 shows a grinding wheel pack used in a CNC grinding machine. By coordinating the moments of the machine’s axes the goal is to cause interference between the rotating grinding wheels and the workpiece in such a way as to form the workpiece geometry into the predetermined shape. A common example of the type of workpieces manufactured on a CNC grinding machine are cutting tools that are used in other manufacturing processes. Figure 1.10 shows some examples of the types of cutting tools that can be manufactured using CNC grinding. A key feature of the grinding process is that due to the large cutting forces, very high disturbance loads are experienced by the servo axes. The cutting forces are functions of the machine’s complex kinematics and as such are difficult to predict efficiently (Demir et al. 2010). The modelling and control of the grinding process are outside the scope of this thesis.
Figure 1.9: CNC grinding wheel pack.

Figure 1.10: Examples of workpieces manufactured on a CNC grinding machine (from top to bottom, an end mill, a tap, a burr, and a fir tree cutter).
CHAPTER 2

LITERATURE REVIEW

As is often the situation in engineering, a number of areas contribute to the research topic. The review of these areas has been broken down into four main sections. Firstly, a review is made of existing control techniques used in machine tool servo drives including the limitations of such approaches. The next section presents an overview of work conducted on modelling machine tool servo drives. The motivation for this section of the review has two parts, firstly to investigate the phenomenon behind the instability described in Section 1.2, and secondly to develop models suitable for model-based controller design. In the third section, various model-based control approaches are reviewed for their potential to solve the identified problem, and for their suitability for implementation on an industrial servo drive. The final section outlines the current state-of-the-art in Model Predictive Control (MPC), which not only inherits all of the benefits of the other model-based approaches, but also includes the ability to explicitly handle system constraints.

2.1 CURRENT INDUSTRY PRACTICE

Historically, and still to this day, commercial machine tool servo drives have almost exclusively been composed of simple PID controllers, which date back to governor designs from the late 19th century (Bennett 1993). Early mathematical definitions feature in works by Minorsky (1922) and Callender et al. (1936). The most popular control structure used in modern systems is the cascaded PID. Figure 2.1 depicts a typical three-loop configuration.

The outermost loop contains the position controller, which accepts an axis po-
LITERATURE REVIEW

![Classical cascaded PID control structure.](image)

Figure 2.1: Classical cascaded PID control structure.

Position error – the difference between the CNC-generated position reference and the measured position. The error is used to calculate a control output, which in this case is a motor (or axis) velocity reference. The main role of the position controller is to ensure the axis tracks the reference trajectory. The middle loop contains the velocity controller. In a similar fashion to the position controller, it uses the motor (or axis) velocity error to generate a motor current reference. The velocity feedback can be measured directly using a tachometer, or inferred from the position feedback (encoder or resolver) by taking the derivative with respect to time. The main role of the velocity controller is to reject plant disturbances, such as the effects of friction and changing load (Leonhard 2001). The innermost loop contains the current controller (or current regulator). It accepts the motor current reference from the velocity loop. This is compared with the measured motor current, to calculate a control output (usually a voltage), which is applied to the axis motor via a Pulse Width Modulation (PWM) driven inverter (power electronics). It should be noted that this multi-loop control structure can only function effectively when the bandwidth of the control increases towards the inner loops with the current loop being the fastest and the position loop the slowest (Leonhard 2001).

Despite their simplicity, it is well accepted that PID controllers have several limitations associated with high performance control of machine tool servo drives. The key limitations of PID relating to this research are: control in the presence of mechanical compliance, maintaining robustness against changing plant dynamics, and tuning for optimal performance. A detailed review of these three
CURRENT INDUSTRY PRACTICE

areas is presented in the following subsections.

2.1.1 Limitations – Controlling Mechanical Compliance

There are several previous works which deal with the control challenges associated with non-rigid machine tool servo drive axes; that is, axes which are subject to mechanical resonance. In an excellent overview paper by Ellis and Lorenz (2000), the authors investigate the case when there is a compliant coupling between the motor and the load. A comparison of seven methods of resonant-load control are compared for their ability to improve performance in the presence of a low-frequency lightly damped resonance. The seven methods are claimed to be those used most often in industry. A key assumption of the work is that the dynamics of the axis remain constant. This conflicts with one of the goals of this thesis; that is, providing a solution for changing plant dynamics. Nevertheless, the work is still important to consider because it provides an overview of current industry practice. Given the block diagram of the plant model shown in Figure 2.2, the transfer function from electromagnetic torque, \( T_m \), to motor velocity, \( \omega_m \), is

\[ \frac{\omega_m}{T_m} = \frac{1}{(J_m + J_w)s} \left( \frac{J_w s^2 + c_{mw}s + k_{mw}}{J_m J_w s^2 + c_{mw}s + k_{mw}} \right) \]  

(2.1)

where \( J_m \) and \( J_w \) are the motor and load inertias respectively, and \( k_{mw} \) and \( c_{mw} \) are respectively the torsional stiffness and viscous damping coefficient of the coupling between the motor and the load. This is the transfer function that would represent the plant in the case where the sole position feedback sensor is on the motor, as is common in industry.

The bi-quad function on the right of (2.1) alters the phase and gain of the lumped inertia plant. It introduces an anti-resonance/resonance pair at frequencies given by

\[ \omega_{AR} = \sqrt{\frac{1}{J_w k_{mw}}} \]  

(2.2)
and

\[ \omega_R = \sqrt{\frac{J_m + J_w}{J_m J_w} k_{mw}} \quad (2.3) \]

respectively. Figure 2.3 depicts an example of the frequency response generated by (2.1), with and without the bi-quad term. In itself this is not an unstable system; however with the addition of a high performance (discrete-time) PI controller, it is possible for the closed-loop system to become unstable. Figure 2.4 shows the open-loop frequency response for the same plant with the inclusion of a typical industrial PI velocity-loop controller. Note that if the plant were rigid (with the same combined inertia) then the controller would provide adequate performance. Thus the methods used for controlling mechanical resonance typically rely upon modifying the effects of the bi-quad term.

The seven methods examined by Ellis and Lorenz include three filtering techniques (low pass, notch and bi-quad filters), and observer-based methods (acceleration feedback, observer filtering, active resonance damping and centre-of-mass control). Note that the observer-based methods require one or more observed signals in addition to the motor feedback. The paper discusses two such observers (rigid-body Luenberger observer and compliant-body observer). The conclusion is that the notch filter, the bi-quad filter, and the rigid-body observer
based acceleration feedback are the most effective at raising the system bandwidth and dynamic stiffness. The work was extended further by Ellis and Gao (2001) to investigate low-frequency mechanical resonance (200–400 Hz). The results presented indicate that acceleration feedback is most effective in dealing with low-frequency resonance.

Similar work was done by Moscrop et al. (2000). The authors analyse the factors which contribute to instability in the presence of a motor-load inertia mismatch. A key finding is that the coupling stiffness has a greater effect on stability than a large motor-load inertia mismatch. In a follow-up paper, Moscrop et al. (2001) investigate the control of such systems. Their approach is to use higher-order controllers and include sensors on the load (in addition to the motor). In many industrial applications this is both physically impractical, and economically unjustifiable. In Gannel and Welch (2003) the use of acceleration feedback to improve dynamic stiffness is proposed. However the authors recommend the
Figure 2.4: Bode diagram showing the open-loop response from velocity error to velocity feedback for the plant shown in Figure 2.3 with a (discrete) PI controller. For the compliant plant, notice the instability due to the negative gain margin.

approach of adding accelerometers to the load inertia. As just stated, this is not always possible. Another solution involving acceleration feedback is presented by Youkin (2004), where the feedback is derived by double-differentiating the position feedback. In practice, this would produce a very noisy signal, which may not be usable in the compensation algorithm.

In Wertz and Schütte (2000), the authors present a complete step-by-step controller design approach for speed control of servo drive systems which exhibit mechanical resonance. The control structure is the familiar cascaded PID, augmented with a low-pass filter. The paper includes strategies for identification of the mechanical system via experimentation, and subsequently, an analytic method for the controller design. Interestingly, only minimal prior knowledge of the mechanical plant dynamics is required, as the system is claimed to be identified in sufficient detail during the design process. Although claimed to be an auto-tuning process, a substantial number of “gains” must be tuned by a knowl-
edgable operator.

It is evident that a PID controller on its own is insufficient for controlling machine tool servo axes which are subject to lightly damped mechanical resonances. An ongoing issue with all the approaches reviewed to this point, is the need to effectively identify the mechanical resonance. Modelling and system identification is a key step, which is explored further in Section 2.2. However, even if a suitable model of the plant is available, appropriate selection of controller (and filter) parameters is still a task which requires significant expertise, and furthermore is difficult to automate. This limitation makes adaptively tuning the servo axis in the field an ongoing issue for these control approaches.

2.1.2 Limitations – Robustness to Changing Plant Dynamics

Over the last decade there has been a trend in high performance machine tools to incorporate direct-drive servo axes. This configuration eliminates backlash, reduces friction and improves acceleration. However, in contrast with geared systems, the direct-drive motor experiences the unattenuated effect of changes in load inertia and disturbances. One challenge observed anecdotally on a variety of commercial direct-drive systems is the degradation of the controller performance as a result of changes to the mechanical configuration of the axis, which is common for axes which hold the workpiece being machined (see Section 1.2). Axes that are especially susceptible are those where large cutting forces are routinely experienced (such as on milling and grinding machines), because they typically require highly tuned controllers to adequately reject the process disturbances. Furthermore, when the control loops are optimally tuned for a given workpiece, the system stability is often compromised when the workpiece is changed to one with significantly different geometry. To illustrate the point, the photograph on the left of Figure 3.4 depicts the normal axis configuration, while the photograph on the right depicts a situation when a large inertia is coupled to the motor. The
LITERATURE REVIEW

consequence on the axis performance is usually a change in the available bandwidth and a possible excitation of vibrational modes. A similar problem exists when the stiffness of the coupling element is changed. Very little literature is found with methodologies for dealing with this problem of variable load dynamics.

One of the few methods that has been proposed to deal with mechanical resonance in systems with uncertain dynamics is $H_\infty$ control (Iwasaki et al. 2002). $H_\infty$ control works by considering an upper bound on the perturbation of the nominal plant model, and subsequently designing a controller which ensures stability. The perturbation is essentially any parameter variation or modelling uncertainty. While $H_\infty$ can guarantee stability, it is generally accepted that it is a conservative approach, and thus does not result in a high performance controller.

Another approach which has been used to deal with large changes to load dynamics is Sliding Mode Control (SMC). In Van Brussel and Van den Brembussche (1998), the effect of large changes to load inertia was investigated. The application considered is a linear motor in a machine tool servo drive. The mechanical plant of this configuration is considered to consist of a single lumped mass, and hence devoid of any mechanical resonance. SMC was compared to a simple pole-placement controller and robust $H_\infty$ control in terms of performance and stability robustness to large changes in the mass of the moving carriage. The researchers found that SMC achieved the best results; however because of the inherent chattering introduced by the approach it is not well suited to high precision machining.

Gain-scheduling is another approach which can be used to compensate for varying mechanical resonance (Symens et al. 2004). It is applied to systems where the eigenfrequency is dependent on the pose (mechanical position/configuration) of the system. The idea is to design a controller with gain-scheduling that both stabilises the system and provides acceptable performance.
for the current pose. The set of gains is determined off-line via a manual process. In the application that is the focus of this thesis, the change in pose relates to the change in the mechanical dynamics of the workpiece. However, the key drawback of this approach is that it must be tuned for the full range of eigenfrequencies that is likely to be experienced, and furthermore, it must have knowledge of the system pose so as to schedule the appropriate gains. This makes the approach somewhat prohibitive for a system which is designed to be fully flexible in the range of workpieces which are able to be machined. Nonetheless, the writer is aware that this approach has been used successfully to some extent in industry; however a more robust solution is desirable.

Further compounding the problem is the continuing advance in permanent magnet materials, which has resulted in the proliferation of low inertia servo motors. The widespread introduction of low-inertia motors motivated Younkin et al. (1991). The researchers’ key observation in the application of low-inertia motors is the greater effect a load variation has on the controller tuning. Furthermore, a decrease in the motor inertia will reduce the dynamic stiffness of the system to reject disturbances. To compensate, the controller would require a possibly unattainable increase in bandwidth. Note that this work was conducted assuming a rigid coupling, and hence did not consider the effect of a compliantly coupled load. Nevertheless, this problem is still a core challenge for modern machine tool servo drive control. Schmidt and Lorenz (1992) attempts to address the issue of increasing dynamic stiffness via the use of acceleration feedback. Citing the difficulties associated with using accelerometers in practical applications, the researchers recommend the use of an acceleration observer. However the method is limited by the accuracy of the observer implementation, especially when the only available direct measurement is position feedback. As with most observer designs, a model of the system is required. The model proposed here is a two-mass model, however the authors do not describe how to determine the model
parameters.

2.1.3 Limitations – Tuning for Optimal Performance

The PID controller is unquestionably the most commonly used control algorithm in machine tool servo drives. An ongoing challenge in using this type of controller is the need to choose suitable tuning parameters. While this is true for all controllers, the tuning of a PID controller is certainly not the easiest task, requiring a certain level of expertise to accomplish effectively. Given the limitations presented in the previous subsection, the ability to simplify the tuning process if the plant dynamics change is important. One of the first approaches to selecting tuning parameters was described by Ziegler and Nichols (1942). The empirical Ziegler-Nichols (ZN) method has the advantage of requiring very little knowledge about the process, but a significant disadvantage is that it inherently produces a controller with poor damping. Due to the widespread use of PID controllers, there is motivation to develop better methods to design and tune them. This is especially true for those methods that can tune the controller automatically. For developers of off-the-shelf motion controllers, the benefit is to alleviate the need to provide significant amounts of technical support to OEM machine builders during the system integration phase. Developing ways to automatically tune PID controllers has been a topic that has attracted a significant amount of attention. The specification of optimisation criteria is typically used to arrive at a set of controller gains that generate the extremum of some objective function. The methods reviewed here can broadly be broken up into those that achieve local optimality and those which attempt global optimality.

Local Optimality

The performance and robustness of a control system is frequently stated in terms of the sensitivity $M_s$ and the complementary sensitivity $M_p$ functions. For
a unity-feedback system they are

\[ M_s = \frac{1}{1 + G(j\omega)C(j\omega)} \]  
\[ M_p = \frac{G(j\omega)C(j\omega)}{1 + G(j\omega)C(j\omega)} \]

where \( G \) and \( C \) are the plant and controller transfer functions respectively. Several researchers use these as constraints when performing optimisation on the controller parameters (Persson and Åström 1992; Schei 1994). As is perhaps expected, the effectiveness of these approaches depend greatly on the accuracy of the system models used in the development. Nevertheless, assuming a suitable model exists, then given a PID controller of the form

\[ C(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \]

where \( K_p \) is the proportional gain, \( T_i \) is the integral time constant and \( T_d \) is the derivative time constant, Schei’s approach is to attenuate low-frequency disturbances. Since the integral term dominates at low frequencies, this is achieved by maximising

\[ f_i(K_p, T_i) = \frac{K_p}{T_i} \]

subject to the constraints

\[ M_s(j\omega) \leq m_s \]  
\[ M_p(j\omega) \leq m_p \]  
\[ K_p \leq K_{p,\text{max}} \]

where \( \omega \) is a set of frequencies in the working frequency range, \( m_s \) is a number in the range of 1.3–3.0, and \( m_p \) is a number in the range of 1.0–1.6. The parameters \( m_s \) and \( m_p \) determine the trade-off between robustness and performance in different frequency domains. Schei shows that the gain and phase margins will be given by \( m_s \) and \( m_p \) respectively, and hence it is relatively straightforward for the user to choose appropriate parameters for the optimisation.
Âström et al. (1998) extend this work by analysing the nature of the sensitivity constraints, and providing efficient numerical procedures for solving the optimisation. Their method, specifically for tuning PI controllers, is based on solving an optimisation problem that incorporates constraints on sensitivity to model uncertainties. However, they show that the optimisation problem is nontrivial because the constraints define a set which is not guaranteed to be convex; meaning that only local optimality is guaranteed. Solutions to this complication are provided, along with a discussion on the suitability of the approach to automatic tuning.

Pham et al. (2000) details a procedure to tune a machine tool servo drive based on cascaded PID control. The method includes consideration for the dynamics of elastic bodies, the dynamics and saturation of electric actuators, and the effect of delays and quantisation within the digital controller. Their approach is to initially select the controller gains based on a continuous controller and rigid plant model. They then discretise the control system, introduce the full plant dynamics and use this as the starting point for an optimisation procedure. The paper provides a good comparison with the Âström et al. (1998) method just presented, and while the approach is simpler to implement than that of Âström et al., the performance of the resulting controller is generally lower.

Another research area for automatically tuning PID controllers is Iterative Feedback Tuning (IFT) (Hjalmarsson 2002). Like many other auto-tuning algorithms, IFT aims to optimise the controller performance by minimising some cost function. A natural choice is a quadratic function involving the error between the achieved and the desired response. The optimisation is carried out on the controller parameters and avoids any system identification step. Instead, the controller parameters are updated via an iterative algorithm, which requires the machine to repeat the desired movement until the optimisation converges. A potential drawback of this approach is that it assumes the cost function is convex. Another more significant drawback is the requirement for the machine to
have repetitive movements. By its very nature the trajectory the machine must follow will be different for each machining job. Furthermore, it is possible that the movements used to tune the controller are not representative, and as such the performance may not be optimal for general machining operations. Finally, it is likely to be unacceptable for the first of a batch of workpieces to be destroyed while the machine’s servos are tuned.

A more recent approach for automatically tuning PID controllers is via the use of Extremum Seeking (ES) algorithms. One notable paper on this is by Killingsworth and Krstić (2005), where they compare the proposed ES algorithm with several other approaches, including Ziegler-Nichols (ZN) and Iterative Feedback Tuning (IFT). The effectiveness of the controller is determined by conducting a step-response experiment, which is used to evaluate a cost function similar to that used in IFT. The results from evaluating the cost function are used to update the discrete ES algorithm, the output of which is a set of PID gains that move the controller’s performance closer to the optimum. The iteration of the ES algorithm continues until the cost function converges on a local minimum. The optimised PID controller parameters obtained via this method are shown to produce a control system with performance similar to that of IFT, however it does not solve any of the issues associated with IFT.

Global Optimality

While the previous tuning methods will typically only guarantee local optimality, the following have the potential to achieve global optimality. Genetic Algorithms (GA) use the Darwinian strategy of evolution by survival of the fittest. A set of controller tuning parameters/functions is known as a chromosome, each parameter/function being a gene. Based on evolution of the fittest, chromosomes are chosen to mate and produce subsequent generations. The search for the optimum set of genes is an iterative process, which involves reproduction, in which
LITERATURE REVIEW

A set of the fittest chromosomes are selected; crossover in which two or more chromosomes swap a number of genes; and mutation in which a chromosome has its genes altered slightly via random noise. An early paper which uses this approach is by Pritschow and Bretschneider (1999), where they demonstrate the ability to automatically tune a servo drive’s position loop (P controller) and velocity loop (PI controller) using GA. Kuo and Yen (2001) present a similar method for automatically tuning a multi-axis machine tool. The quality (or fitness) of each chromosome is based on the machining accuracy. In a related approach, Jee and Koren (2004) used adaptive fuzzy logic to self-tune their proposed controller. On a positive note, none of these methods require any knowledge of the plant dynamics. However, they have the same major drawbacks as those highlighted for IFT and ES, and furthermore, there is no guarantee that they will actually converge to an optimum.

2.1.4 Conclusions

As identified there are a few good reasons for the popularity of cascaded PID. It is not only extremely simple to implement and computationally very efficient, but importantly, it can be quite effective at providing adequate levels of performance if designed and tuned appropriately. However, it is by no means perfect and does have several serious limitations. The most important ones relating to this work are the difficulty associated with controlling mechanical compliance, and maintaining robustness against changing plant dynamics. Both these points contribute to the difficulty associated with tuning the controller to achieve the required performance. The metric which determines performance is a combination of tracking accuracy and disturbance rejection.

From these observations it is clear that developing a better understanding of the critical dynamics affecting the system is a crucial step. While the type of instability demonstrated in Figure 1.8 is common in industry, specific research investi-
gating the combination of phenomena that cause it does not appear to have been conducted previously. This provides the motivation for further investigation and is the basis for the first research question posed in this thesis.

The outcome of this investigation should assist in the design of a controller with the required level of performance. Such a controller is likely to be model-based. Thus a key step is to develop suitable models of the system. The model should include enough detail to capture the key dynamics, but be of sufficiently low-order so as to be practical for control design. This is the topic of the next section.

### 2.2 MODELLING MACHINE TOOL SERVO DRIVES

Modelling of machine tool servo drives will form a large component of the work in this research project. Initially, it will be used to investigate and analyse the system dynamics which are responsible for the instability identified in Section 1.2, and then subsequently it will be used for model-based control design. While there are some papers which deal explicitly with modelling machine tool servo drives, which are presented below, none provide a holistic approach by exploring which dynamics are critical for inclusion in the model to fully investigate the accuracy and stability of these machines. Of significance to this project is the observation that structural compliance is often omitted and the electrical system, or current loop dynamics, are typically ignored when modelling such systems. Furthermore, modelling of multirate servo drive loops (as are often found in commercial systems) does not even warrant a mention in any of the literature reviewed.

#### 2.2.1 Determining the Suitable Model Order

In an early paper, del Re et al. (1996) suggest that developing a model of a machine tool axis based on component specifications can lead to very high-order
models, which are not necessarily more accurate than a reduced-order representation. Specifically, they state that the system is usually dominated by several key dynamics and accurate modelling of these is most important. This observation is encouraging as it provides evidence that low-order models, which are easier to identify, will provide the necessary information for controller design, and stability and performance analysis. Conversely, Poignet et al. (1999) present work which seeks to develop a generic model. The model they propose includes 8 masses and 7 compliance elements. While the motivation for this work is appealing, it results in a very complex mechanical model, and little detail is provide about how the model parameters are to be determined. The high-order model is likely to make model-based control design prohibitively difficult.

2.2.2 Identifying Critical Dynamics

Ebrahimi and Whalley (2000) explore the effect various mechanical phenomena (e.g. friction, stiffness and backlash) have on the axis dynamics. The paper investigates how variation in the system parameters affects performance. Importantly for this project, a decrease in stiffness is shown to increase oscillatory behaviour; however no stability analysis is provided.

Erkorkmaz and Altintas (2001a) outline the importance of the modelling step in developing high-performance servo drives. The model they develop assumes a rigid axis, devoid of vibrational modes. This work is extended by Zheng et al. (2006) to consider multi-inertia models; however no rationale for selecting the appropriate order model is given and again no stability analysis is provided.

Friction is inherent in nearly every mechanical device. In machine tools, friction is a problem when it comes to position regulation (hunting), tracking at low velocities (stick-slip), and velocity reversal (axis pause). This is primarily due to the dominance of highly nonlinear frictional effects around zero velocity. The outcome will usually manifest as a tracking error, resulting in an inaccuracy of
the workpiece geometry. The key problem here is that the standard PID controller is reactive, since it requires tracking error to exert a compensating reference signal. A survey paper by Armstrong-Hérouvry et al. (1994), which pulls together the research from some 280 publications, presents an excellent summary of the work done in the area of control of machines with friction. Further details on modelling friction can be found in Appendix A.1.

2.2.3 System Identification

Another area critical for modelling of machine tool servo drives is system identification. This is quite a mature area (Ljung 1999), and while the techniques developed from this research are important to this project, a complete review is not presented here. However, for completeness, and as a reference, a review of system identification techniques is provided in Appendix A.4.

2.2.4 Conclusions

Determining the correct level of detail (and the specific elements to include) in a dynamic system model is essential for successful identification and analysis of problematic dynamics in a high performance system. Further, the structure of the model is an important consideration if it is also to be used for control design. From the work undertaken to date it is unclear what level of detail is required to successfully analyse and predict the issues associated with mechanical resonance and changing load dynamics. For this reason there is strong motivation to investigate further and it is this gap in the literature which forms the basis for another of the research questions posed in this thesis.

2.3 Model-Based Control Approaches

Having a suitable dynamic model of the plant that is to be controlled can assist with the design of a high performance controller. Furthermore, in some model-
based design techniques the model is used directly to tune the controller. The ability to develop and identify plant models (see Section 2.2), and use computer-aided simulation for analysis has made model-based control a popular choice among researchers. It is hypothesised that model-based control may provide a suitable solution for the challenge identified in Section 1.2. It is held that a single controller with fixed gains will be unable to provide suitable performance and still be robust against changing load dynamics (see Section 2.1.2). Hence devising an adaptive-style model-based controller which can be easily tailored to different axis configurations via automated system identification is proposed.

2.3.1 Feedforward

One control technique which makes use of a model of the plant is feedforward control. While there are several feedforward algorithms that have been proposed over the years, probably the one which has gained the most attention is Zero Phase Error Tracking Control (ZPETC), which was first published by Tomizuka (1987). The general idea of the approach is to cancel all the discrete closed-loop poles and zeros, thereby resulting in an overall transfer function from reference input to actual output of unity. This is theoretically achieved by filtering the reference input by the inverse of the discrete closed-loop system (feedback controller and plant). The difficulty is that some of the closed-loop system zeros might be unsuitable as poles in the feedforward compensator; that is, there might be closed-loop zeros which are on or outside the unit circle. Tomizuka’s solution to this is to factorise the closed-loop zeros into two parts, those which are acceptable as poles, and those which are not. The poles which are not acceptable are modified before being included in the feedforward controller, in such a way that, in the frequency response of the proposed system, a zero phase error is in fact achieved; however the gain error is not fully eliminated. The main drawbacks of the ZPETC method are that it requires precise knowledge of the dynamic behaviour of the
drive system, and has a tendency to cause actuator saturation; that is, it does not
directly account for system constraints. Several researchers have investigated the
application of this approach to machine tool servo drives (Suzuki and Tomizuka
1991; Endo et al. 1996; Lee and Tomizuka 1996; Kempf and Kobayashi 1999; Ko
2003).

2.3.2 State-space Design

Perhaps the most widely used model-based control technique is state-space de-
sign. Typically, the resulting design includes state feedback gains which work to
drive the system states to the desired locations. The gains can be selected via a
variety of techniques; however common textbook approaches include pole place-
ment and the Linear Quadratic Regulator (LQR) (Franklin et al. 2006). Another
state-space design method which has been applied to machine tool servo drives
is the RST controller (Landau 1998). As full state feedback is often required, the
state-space controller will usually include state estimation to reproduce the states
of the system which cannot be obtained via direct measurement for practical or
economical reasons. The model of the system used in the controller design, and
also in the estimator, is critical to the success of the approach. In its classical
form, state-space design does not provide adequate disturbance rejection. How-
ever, techniques to include auxiliary mechanisms for controller integration have
been developed. While these can be viewed as ad-hoc additions to the standard
state-space controller form, they are effective in systems where dealing with dis-
turbances is the norm. Another limitation of state-space design is that there is
currently no means by which system constraints can be managed. Such a con-
troller would be inherently nonlinear. In terms of implementation, the computa-
tional overhead is slightly more than with a PID controller with comparable
performance (especially if a full state estimator is included in the design), but it
is still well within the capabilities of current day embedded digital processors.
Tuning of the state-space controller is achieved by selection of pole locations, or in terms of LQR design, selection of cost function weights, which is considerably easier than the requirements for tuning a PID controller with comparable performance.

In an investigation which pulls together a range of model-based machine tool servo drive control approaches, Erkorkmaz and Altintas (2001b) demonstrate the advantages of supplementing a pole-placement feedback controller ($K_C$) with ZPETC, a Kalman filter state and disturbance estimator, and feedforward friction compensation. Figure 2.5 displays the block diagram of the proposed axis tracking control scheme. Experimentation was carried out on a vertical high speed machining centre, which was set to track circular and diamond shaped trajectories. The results confirm that the addition of ZPETC, a Kalman filter state and disturbance estimator, and feedforward friction compensation significantly improve the achieved contouring accuracy.

This approach results in a very high performance control suitable for high speed and high accuracy machining. However to achieve these results a large amount of system identification (Erkorkmaz and Altintas 2001a) and hand tuning is required for the specific plant configuration. This contradicts part of the motivation of this thesis, which is to provide a method for **automatically** tuning a controller to handle changing mechanical plant dynamics. Finally, as with the
general case of model-based controllers, no consideration is given to system con-

straints.

2.3.3 Conclusions

Clearly model-based control is not new. There is plenty of evidence that it has
been successfully applied to motion control systems, including machine tool
servo drives. However, the controllers are almost always linear in nature and
do not explicitly account for system constraints. Constraints, where considered,
are typically handled in an ad-hoc fashion, for example post-saturation of the
control input. The topic of the next section is Model Predictive Control (MPC),
which does explicitly account for constraints. It is generally well accepted that
by exploiting the full capacity of a machine (i.e., operating on or near the actu-
ator constraints) it is possible to attain improved throughput (or productivity).
Furthermore, by controlling operational constraints explicitly a greater level of
safety is also achieved.

2.4 Model Predictive Control

Model Predictive Control (MPC), is the process whereby a model of the plant is
used to predict how the system will respond to a set of control inputs. The goal is
then to optimise these inputs so that the predicted output closely coincides with a
set-point or desired (and known) trajectory. The first element of the set of optimal
control inputs is then applied to the plant, and the process is repeated at the next
controller update (Maciejowski 2002).

An early contribution in the application of predictive control to machine tool
servo drives was made by Boucher et al. (1990) via the use of Generalized Pre-
dictive Control (GPC). The authors show how the tracking performance can be
improved when compared with classical PID-style control. This early approach
is similar to state-space design (introduced in Section 2.3.2), in that no online op-
timisation is required and system constraints are not considered. Because of this, implementation is possible on the comparatively slow micro-controllers available during the early 1990s. The process was extended to include control law and tracking error constraints by Dumur et al. (1996). A comparison of different controllers was conducted by del Re et al. (1996). These included PID, state-space (pole placement) and predictive control. The work relied on access to sufficiently accurate system models. However, while experimental results are presented in the paper, it is not clear how the authors implemented constrained predictive control with online optimisation given the limited power of embedded processors of the day. Another paper by Dumur et al. (2000) uses adaptive MPC to deal with motor drives with flexible modes (mechanical resonance). However, the method does not appear to consider any constraints in the servo system, and hence again, is similar to state-space design.

MPC is a proactive control approach, in that it predicts what control input (within the constraints of the system) is required to ensure accurate tracking of the reference trajectory. MPC retains a feedback element to account for model uncertainty. Furthermore, an integrator can be embedded into the formulation to account for unknown, low-frequency disturbances (Wang 2009). Moreover, given that a suitable dynamic model of the plant is available, tuning the controller is essentially a trade-off between tracking accuracy and control effort. Literature on modelling machine tool servo drives was covered in detail in Section 2.2.

2.4.1 Limitations

The main limitation of MPC is that significant online computation is required to perform the optimisation. Presently, predictive control with online optimisation is reserved for processes which have update rates in the order of tens of seconds to minutes, not hundreds of microseconds, which is typical for servo drive systems. Bemporad et al. (2002) present an algorithm to develop an equivalent
closed-form (explicit) solution to the original MPC problem offline. The result is a gain-scheduling algorithm which is a function of a parameter vector that contains the current state of the system and the desired future output. The solution is shown to be piecewise affine in the parameter vector, where different controllers are defined for discrete polyhedral regions $X_i$ within the parameter space. The online implementation then simplifies to a sequential search through the regions to locate the one which the current parameter vector belongs to. The controller associated with the identified region is then used to generate the input for the plant. The algorithm to construct the explicit solution is further refined by Tøndel et al. (2003a). While these approaches have the potential to reduce the online computation requirements, they do result in an increase in the memory used to store the program in the controller hardware. Moreover, as the MPC problem becomes more complex (higher order plant model, more input/output variables, more constraints, longer horizons, etc.) the number of regions in the explicit solution grows. The computational advantage over online optimisation is then somewhat diminished since it can take a long time to locate the appropriate region within the parameter space. In the worst case every region must be explored.

Tøndel et al. (2003b) present an algorithm which takes the explicit solution and organises it into a binary search tree. Not only does this drastically reduce the online search time, but it also results in a reduction in the storage requirements.

While the problem formulation by Bemporad et al. (2002) is adaptable to trajectory tracking problems, typically only set-point tracking is implemented. In fact the Hybrid Toolbox for MATLAB (Bemporad 2010) only allows for set-point tracking. Furthermore, Ferreau et al. (2008) claim that, even with Explicit MPC (EMPC) formulated as a binary search tree, trajectory tracking is impossible from a computation standpoint.
LITERATURE REVIEW

2.4.2 Conclusions

The benefits of Model Predictive Control (MPC) as a control technique have been well established. However its application to reference tracking on Digital Servo Drives (DSDs), which typically have very fast update rates, is limited by the computational power of present-day embedded processors. Hence assuming that this limitation can be overcome, and that suitable models which capture the changing dynamics can be developed, then MPC may provide a viable option to the challenge identified in Section 1.2. There does not appear to have been any substantial work done previously with MPC in this area; specifically in relation to machine tool servo drives. Investigation of implementation on such systems represents another gap in the literature and thus forms the basis for the final research question to be addressed in this thesis.

2.5 LITERATURE REVIEW CONCLUSIONS

From this review it is evident that machine tool servo drive control is a mature area of research. A large amount of work has been done in the fields that are relevant to this study; in particular, in the areas of tuning PID controllers, dealing with mechanical resonance, model-based control design, and modelling of machine tool servo drive axes. However, the key findings from the review are that in general little emphasis is given to modelling compliantly coupled loads in machine tool axes; no existing approach to deal with mechanical resonance is robust enough to handle the large changes in load dynamics experienced; and finally MPC has yet to be implemented on a servo drive system which has update rates in the order of hundreds of microseconds. Hence, there does appear to be scope for further research and the potential to make a contribution in the following areas:

- modelling and analysis of servo drive systems which are subject to large
(and unknown) changes in load dynamics, and

- the application and implementation of model-based control design to machine tool servo drives, which are characterised by the extremely fast rate at which the control input is updated.

### 2.6 Research Aims

Over the last decade there has been a trend in high performance machine tools to incorporate direct-drive servo axes. This configuration eliminates backlash, reduces friction and due to a reduced inertia, improves acceleration. However, the problem of degradation of the controller performance as a result of changes to the mechanical configuration of the axis, as was introduced in Section 1.2, becomes a real challenge. In response to this, the research questions which are to be answered in this thesis are:

*What are the critical dynamics which lead to the observed instability, and to account for it, what level of model fidelity is required?*

This thesis will present an investigation of the system dynamics, with the use of models, to analyse and identify the critical elements responsible for the identified instability.

*Are the models used for analysis also suitable for model-based controller design?*

Developing a suitable model is important for two reasons. Firstly, the model must be rich enough to reveal the problematic dynamics, and for it to be useful for controller design, it should also be of low-order. This thesis will assess the suitability of said models for the different stages of control system implementation: from plant analysis through to controller design.

*There has been significant advancement in the theory underpinning model-based control: can model-based control provide an improved level of performance and be easily adapted...*
LITERATURE REVIEW

to deal with both mechanical resonance and changing plant dynamics?
The benefits of model-based control have been well documented over the past two decades. In this thesis several advanced model-based control approaches will be investigated in regards to their suitability to provide viable solutions to the identified control challenges.

Can model-based control be applied to actual industrial hardware?
Proving that model-based control provides benefits over the incumbent approach is important, but less so if it cannot be implemented in practice. This thesis will investigate whether it is possible to implement advanced model-based approaches on standard industrial hardware.

2.7 THESIS LAYOUT

In the previous section, an introduction to the machining process of CNC grinding was provided, along with a description of the identified problem which is a focus of this thesis. Chapter 3 introduces the equipment, both production and testing, which are utilised throughout the rest of the thesis. It also presents the work completed during the project to model the system. Within this chapter models of the system are developed, along with justification for the chosen model order. Finally the model parameters are determined using experimental data and well established system identification techniques. The work presented in Chapter 4 utilises these models to analyse and highlight the limitations of the classical approach to machine tool servo drive control. While initially focusing on the linear elements of the system, attention is also given to analysing the effect of the dominant nonlinearity – mechanical friction. In summary, at the conclusion of Chapter 4 a detailed examination of the dynamics of the system has been completed, along with the creation of suitably detailed models of the plant. These models will serve as the basis for model-based controller design, which is the
Chapter 5 begins the work undertaken on model-based controller design. The aim is to provide servo control with adequate performance, while managing the large range of possible workpiece dynamics which may be encountered. The main topic of Chapter 5 is the implementation (in simulation) of both minimum-time optimal control and linear quadratic control. Minimum-time optimal control is explored in an effort to set an upper bound on what is possible in terms of trajectory tracking, given the constraints imposed on the system by the actuator. The linear quadratic controller incorporates reference scaling to enable tracking, a Kalman filter to estimate system states, an integrator to handle process disturbances and explicit nonlinear friction compensation. The linear quadratic controller is successfully implemented with sufficient performance on an actual CNC grinding machine. However, a limitation of the design is that it requires several ad-hoc modifications to its structure to ensure that system constraints are not violated. This weakness motivates the use of Model Predictive Control (MPC), which is presented in Chapter 6. The main focus of Chapter 6 is to adapt and extend existing MPC techniques so that the approach (including management of constraints) may be implemented on the system under investigation. The main restriction here is the rate at which the controller must update the control input. Traditional techniques are much too computationally intensive to be realised on current industrial hardware. At the conclusion of Chapter 6 the use of MPC with constraints is successfully demonstrated experimentally on the CNC grinding machine.

Finally in Chapter 7 a summary of the contributions of this thesis is presented, along with ideas and motivation for where future research may be focused.
CHAPTER 3

MODELLING A CNC MACHINE TOOL AXIS

The aim of this chapter is to develop models of the CNC machine tool servo drive axis. Initially a complete high fidelity model is generated. The structure and the parameters of this model are obtained from either manufacturer specifications or system identification using data collected from experimentation. As the model is built it will be continuously validated using additional experimental data. The fully parameterised and validated model will predominately be used for testing controller designs via simulation. It will also serve as the starting point for a linearised, reduced-order model which is developed in the second part of this chapter. The reduced-order model will be used in Section 4.1 for linear analysis of the key dynamics of the system, and in Chapters 5 and 6 for model-based controller design.

Section 3.1 begins by introducing the CNC machine which is the focus of this thesis – an ANCA RX7 Tool and Cutter Grinding (TCG) machine. Also presented is an example of two workpieces which the RX7 is designed to machine. Finally, a dedicated test rig which is available for preliminary testing of new algorithms is introduced.

The complete dynamic model of the servo drive can be broken-up into three main elements: the control algorithms, the electrical system and the mechanical plant. The role of the control algorithms is to ensure that the axis tracks the required trajectory as set forth by the CNC. The control algorithms rely on measured feedback from the plant to determine control signals for the electrical system. The electrical system applies the required voltage to the servo motor, which in turn drives the mechanical plant along the required trajectory.
Section 3.2 introduces the high-level structure of the servo drive axis, while Sections 3.3–3.5 provide more detail on the various elements, including system identification work where appropriate. Section 3.6 outlines the details of the linearised reduced-order model. Finally, Section 3.8 summarises the chapter, including some details of the key dynamics of the system which are captured by the model.

3.1 EXPERIMENTAL FACILITIES

As mentioned in the Introduction, the industrial partner provides access to commercial CNC machines and test rigs. The main two utilised for experimental work in this thesis are described here.

3.1.1 Production CNC Grinding Machine

Figure 3.1 shows a current state-of-the-art 5-axis CNC grinding machine. The electrical cabinet for the machine can be seen in Figure 3.2. Figure 3.3 shows the layout (as a 3D CAD model) of the CNC grinding machine from Figure 3.1. In this model the three linear axes (X, Y, Z) and two rotary axes (A, C) are labeled, as is the spindle motor which the grinding wheels mount to. The A-axis is also known as the headstock, and is the axis to which the workpiece is directly mounted.

3.1.2 Different Size Workpieces

The CNC grinding machine introduced in Figure 3.1 is designed to be flexible in terms of the relative size and shape of workpieces it can machine. Figure 3.4 shows two drastically different sized workpieces. As will be shown in Chapter 4, the large difference in workpiece geometry presents significant challenges for the DSD control algorithms.
3.1.3 Test Rig – “The Fridge”

The test rig shown in Figure 3.5 is used predominately for testing new CNC path planning techniques (including constraint management) and DSD motion control algorithms. It incorporates a full complement of the equipment used in a produc-
Figure 3.3: CAD model of RX7 CNC grinding machine layout.

Figure 3.4: Small and large workpieces mounted in a direct-drive headstock.

...tion machine, including a CNC (see Figure 1.6), numerous 600V DSDs (see Figure 1.7), and digital and analogue I/O. It can be connected to any of the servo or spindle motors used in the production machine. However, for general testing of new algorithms, small (relatively inexpensive) servo motors are typically used, an example of which is the Permanent Magnet Synchronous Motor (PMSM) shown in Figure 3.6. Once an algorithm has been validated on the test rig, it can then be implemented and tested on a real production machine.
3.2 High-Level Structure

The model developed in this chapter is based on the $A$-axis of the ANCA RX7, which is a direct-drive rotary axis that holds the workpiece being machined. A picture of the $A$-axis holding two different workpieces is shown in Figure 3.4. This axis was chosen because it is one which currently presents significant control...
challenges, mainly due to the wide working envelope in terms of both speed and workpiece size and geometry. It is also a good axis to test and validate new algorithms on due to its stroke being infinite, meaning that unlike a linear axis (and some rotary axes) it does not have dead stops that might be needed should the axis *take off* unexpectedly. Figure 3.7 shows the high-level system structure.

3.3 **CONTROL ALGORITHMS**

The standard ANCA servo drive control approach is the well established cascaded PID controller. Figure 3.8 depicts the overall structure of the scheme. The ANCA servo drive is digital; the discrete control algorithms are implemented on a DSP.

The general methodology of the control approach is for successive inner loops to have higher bandwidths than those which enclose them (Leonhard 2001). This requirement ensures that the reference commands from outer loop controllers are able to be satisfactorily delivered by the inner loops. The bandwidth of a particular loop is influenced by both the update rate of the controller and the response time of the section of plant which it controls (be it electrical or mechanical). Furthermore, the update rate for each loop is typically selected with consideration for the response time of the plant. For example, since the dynamics of the electrical system are typically quicker than those of the mechanical plant, the current loop update rate should be faster than that of the velocity loop. It would be acceptable to have all loops operating at the rate dictated by the fastest dynamics of the plant, but due to limited computation capacity on the DSP it is generally
Figure 3.8: Servo drive control algorithm structure.

the approach that different loops have different update rates. As it turns out, the
dynamics controlled by the inner loops are typically faster than those of the outer
loops, and hence the inner loops have the faster update rates.

Before each of the control loops from Figure 3.8 are explained in detail, it is
necessary to introduce a concept known as vector control. Vector control is a
method used to control three-phase AC electric motors. The A-axis of the RX7
uses a Permanent Magnet Synchronous Motor (PMSM), which falls into this cat-
egory. While the PMSM will be covered in detail in Section 3.5.1, it is enough
to state here that the key aspect of vector control is the notion of direct (d) and
quadrature (q) axes that align and rotate with the motor rotor. The location of
the d and q axes are shown graphically (for a two-pole PMSM) in Figure 3.12 by
\(\psi_d\) and \(\psi_q\) respectively. The d-axis aligns to the magnetic north pole, hence any
resultant magnetic field produced by the motor stator in this direction will pro-
duce zero torque. On the other hand, the q-axis aligns halfway between adjacent
north and south poles, so that any magnetic field from the stator in this direction
will interact with the magnetic field from the permanent magnets on the rotor
to produce a torque as the two fields attempt to align. Hence the objective is to
to control the magnetic field in the d-axis to be zero, and the field in the q-axis to
produce the required torque.

In the ANCA control approach, the outermost loop contains the position con-
troller (P-control). It compares the reference position from the CNC to the mea-
MODELLING A CNC MACHINE TOOL AXIS

sured position, and generates a suitable velocity reference to compensate for any deviation. The main task of the position controller is to ensure that the axis tracks the required trajectory.

The middle loop contains the velocity controller (PI-control), which compares the reference velocity to the actual velocity and produces a \( q \)-axis current reference, \( i_r \). While it will be explained further in Section 3.5.1, it is enough to state here that that motor torque is directly proportional to the \( q \)-axis current. Hence, when a current reference is discussed, it may be interpreted as a motor torque reference. Thus what the velocity controller is effectively doing is calculating a suitable motor torque to compensate for the difference between reference velocity and actual velocity. A key task of the velocity controller is to reject mechanical disturbances acting on the system. These include both friction and load torques.

The position controller is a proportional controller, with gain \( K_{\text{pos}} \). The velocity controller is proportional plus integral with gain \( K_{\text{vel}} \) and integral time constant \( T_i \). Both of these controllers have a sample period of 250 \( \mu \)s. For this axis, the default values for the controller parameters are those given in Table 3.1. These values are chosen so as to maximise the closed-loop bandwidth of each respective loop, while still maintaining a degree of gain margin to achieve reasonable robustness to unknown plant dynamics. Typically a bandwidth of \( 200-300 \) Hz for the velocity loop and \( 10-20 \) Hz for the position loop is desirable. Furthermore, the current loop should have a bandwidth of at least 1 kHz.

The inner most loop contains the current controller, which is actually composed of two individual PI-controllers; one for each of the \( d \) and \( q \) axes. These controllers are both proportional plus integral. As mentioned above, the \( q \)-axis current reference is derived from the velocity controller, and the \( d \)-axis current reference is held at zero. The output of the controllers are \( d \)- and \( q \)-axis voltage levels. These are converted into voltage levels for the three armature axes \( (v^*_a, v^*_b, v^*_c) \) via the Inverse Clarke-Park Transform, which is described shortly in
Table 3.1: Default Controller Tuning Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{\text{pos}}$</td>
<td>3.6111</td>
<td>RPM/deg</td>
</tr>
<tr>
<td>$K_{\text{vel}}$</td>
<td>0.4444</td>
<td>A/RPM</td>
</tr>
<tr>
<td>$T_i$</td>
<td>2000</td>
<td>$\mu$s</td>
</tr>
</tbody>
</table>

Section 3.5.

The three armature voltage levels are the inputs into a Pulse Width Modulation (PWM) algorithm which determines the switching times $(a_p, a_n, b_p, b_n, c_p, c_n)$ for the solid-state power electronics (the key component of the electrical system).

An incremental analogue sine/cosine optical encoder with 11 840 lines per revolution which, with interpolation results in an angular resolution of approximately $1.5 \times 10^{-5}$ degrees, provides position feedback of the motor shaft, $\theta_m$. This signal is also used to generate velocity feedback via differencing at consecutive controller updates. The system includes anti-aliasing filters. The analogue signals are low-pass filtered at 200 kHz. Given the maximum speed of the axis (600 RPM) and the resolution of the encoder, the maximum frequency of the sin/cos signals is approximately 100 kHz. These signals are then fed into a comparator circuit that detects and counts zero crossings (quadrature counts). This circuit includes a dead zone, which reduces noise in the feedback without any frequency-dependent magnitude or phase changes. The quadrature counts are sampled at 4 kHz, as are the filtered sin/cos signals. The latter are used to interpolate a fine position correction on top of the coarse quadrature counts. The filtering of the encoder signals is at a frequency much too high to affect the dynamics of the position and velocity loops.

Hall effect sensors are used to sense current in two of the three armature axes.\footnote{Since the motor represents a balanced three-phase load, to determine $d$- and $q$-axis current feedback, measurement of only two armature axis currents is required, (3.6).} As with the analogue encoder signal, the analogue current signals are fed into low-pass anti-aliasing filters. The filters have an effective cutoff frequency of
6 kHz. The filtered signal is then sampled at 20 kHz. All measured signals are fed through Analogue to Digital Converters (ADCs).

The control system also includes optional low-pass and notch filters between the velocity controller and current loop. The low-pass filter is used to filter out the influence of measurement noise, while the notch filters are used to filter out any current reference signals from the velocity controller that may excite structural vibration modes within the axis. All the integrators within the PI-controllers include anti-windup. Furthermore, there is an output limit on the current reference so as not to exceed the capabilities of the PMSM.

Finally, due to the finite computation capacity of the DSP, the reference signals for successive inner loops of the control structure are generally not immediately available. Clearly, each task must be executed in series on the DSP and thus each has a different priority. While this is a problem for practical implementation, the theoretical implications have been studied recently (Lozoya et al. 2008). At the beginning of a 250 µs update the current controller executes first (highest priority) using the old velocity controller output – 1st delay. When the current controller and the motor commutation calculations are finished, other parts of the system may execute. As it turns out, the position and velocity controllers do not complete executing before they are interrupted by the need to re-update the current controller, which again operates on the old velocity controller output – 2nd delay. On the next current controller execution, the velocity controller has completed its calculations and hence the new output is used. This process is illustrated in Figure 3.9. It will be shown later that such multi-sample delays must be accounted for in the model.

### 3.4 Electrical System

The key component of the electrical system is the solid-state power electronics. The electronics take the physical form of an Intelligent Power Module (IPM) con-
Table 3.1: DSP task scheduling showing multi-sample delays.

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Controller</td>
<td>0</td>
</tr>
<tr>
<td>Commutation</td>
<td>50</td>
</tr>
<tr>
<td>Other Tasks</td>
<td>100</td>
</tr>
<tr>
<td>Velocity and Position</td>
<td>150</td>
</tr>
<tr>
<td>Controllers</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>350</td>
</tr>
</tbody>
</table>

Figure 3.9: Schematic layout of a three-phase intelligent power module with insulated-gate bipolar transistors.

Figure 3.10: Schematic layout of an intelligent power module with insulated-gate bipolar transistors.

The RX7 uses three-phase mains power (AC), which is rectified to produce the DC supply for the IPM. The dynamics of this rectifying circuitry are excluded from the model, since during operation it is known that the DC bus voltage is reasonably stable.
3.5 MECHANICAL PLANT

As the name suggests, the mechanical plant is the section of the servo drive axis which includes all the mechanical components. The main components are the actuator, the transmission elements and the actual workpiece being machined. Modelling the overall dynamics of the mechanical plant will be achieved by dividing it up into logical sections. The first of these is the mechanical actuator, a Permanent Magnet Synchronous Motor (PMSM). This element converts the electrical energy supplied by the electrical system into mechanical work. The second section is the transmission elements, in this case the motor shaft, workpiece holding apparatus and the workpiece itself, which are all made from elastic materials. The dynamics of this section will be dominated by how the axis reacts (moves/deforms) when a load is applied. The other key characteristic which is to be modelled is the friction torque acting upon the motor shaft. Figure 3.11 depicts the structure of the model of the mechanical plant. In this model, the interface between the DSD and the actuator is a three phase voltage signal. Notice that most of the blocks represent nonlinear subsystems, but their existence makes analysis and control design difficult. This is the motivation for Section 3.6, where a reduced-order linear model is developed.

3.5.1 Permanent Magnet Synchronous Motor Model

The actuator for the RX7 A-axis is a Permanent Magnet Synchronous Motor (PMSM). Figure 3.12 illustrates both the physical layout and the electrical dia-
Figure 3.12: Two-pole permanent magnet synchronous motor – mechanical configuration (left) and electrical diagram (right).

The standard nonlinear differential equations for a PMSM are (Lyshevski 2000)

\[
\begin{align*}
\frac{di_q}{dt} &= \frac{1}{L_l + \frac{3}{2}L_m}v_q - \frac{R_s}{L_l + \frac{3}{2}L_m}i_q - \frac{1}{2}p\psi_m i_d - \frac{1}{2}p\psi_m \omega_m \\
\frac{di_d}{dt} &= \frac{1}{L_l + \frac{3}{2}L_m}v_d - \frac{R_s}{L_l + \frac{3}{2}L_m}i_d + \frac{1}{2}p\psi_m i_q \\
\frac{d\omega_m}{dt} &= \frac{1}{J_m} \left( \frac{3}{2}p\psi_m i_q - T_f - T_l \right) \\
\frac{d\theta_m}{dt} &= \omega_m
\end{align*}
\] 

(3.1)

where the states are the \(d\)- and \(q\)-axis currents, \(i_d\) and \(i_q\), and the angular velocity and position of the motor shaft, \(\omega_m\) and \(\theta_m\). The inputs are the \(d\)- and \(q\)-axis voltages, \(v_d\) and \(v_q\), and the friction torque and load torque acting on the motor shaft, \(T_f\) and \(T_l\). The output is the angular position of the motor shaft, \(\theta_m\). There are six parameters for the model: the moment of inertia of the motor shaft, \(J_m\); the number of permanent magnets (or motor poles), \(p\); the magnitude of the flux linkages established by the permanent magnets, \(\psi_m\); the resistance of the stator windings, \(R_s\) and the stator magnetising inductance and leakage inductance, \(L_m\).
MODELLING A CNC MACHINE TOOL AXIS

and $L_l$. The latter two parameters are related to the self-inductance of the stator windings, $L_s$ by

$$L_m = (1 - \xi_i)L_s$$

(3.2a)

$$L_l = \xi_iL_s$$

(3.2b)

where $\xi_i$ is the proportion of leakage inductance (typically 5–10%). The $d$- and $q$-axis currents and voltages are related to the $abc$ stator axes’ equivalents by the Clarke-Park Transformation (Lyshevski 2000)

$$\begin{bmatrix}
    x_d \\
    x_q \\
    x_0
\end{bmatrix} = \frac{2}{3}
\begin{bmatrix}
    \sin \theta_e & \sin \left(\theta_e - \frac{2\pi}{3}\right) & \sin \left(\theta_e + \frac{2\pi}{3}\right) \\
    \cos \theta_e & \cos \left(\theta_e - \frac{2\pi}{3}\right) & \cos \left(\theta_e + \frac{2\pi}{3}\right) \\
    \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}
\begin{bmatrix}
    x_a \\
    x_b \\
    x_c
\end{bmatrix}$$

(3.3)

where $x \in \mathbb{R}^3$ represents either current or voltage, and $\theta_e$ is the electrical angle. For the full derivation of the Clarke-Park Transformation see Appendix A.2. The electrical angle is related to the angular position of the motor shaft by

$$\theta_e = \frac{1}{2}p\theta_m$$

(3.4)

The inverse of (3.3) is given by

$$\begin{bmatrix}
    x_a \\
    x_b \\
    x_c
\end{bmatrix} = \begin{bmatrix}
    \sin \theta_e & \cos \theta_e & 1 \\
    \sin \left(\theta_e - \frac{2\pi}{3}\right) & \cos \left(\theta_e - \frac{2\pi}{3}\right) & 1 \\
    \sin \left(\theta_e + \frac{2\pi}{3}\right) & \cos \left(\theta_e + \frac{2\pi}{3}\right) & 1
\end{bmatrix}
\begin{bmatrix}
    x_d \\
    x_q \\
    x_0
\end{bmatrix}$$

(3.5)

For a balanced three-phase system, where $x_a + x_b + x_c = 0$, (3.3) and (3.5) can be expressed as

$$\begin{bmatrix}
    x_d \\
    x_q
\end{bmatrix} = \frac{2}{\sqrt{3}}
\begin{bmatrix}
    \sin \left(\theta_e + \frac{\pi}{6}\right) & -\sin \left(\theta_e - \frac{\pi}{2}\right) \\
    \cos \left(\theta_e + \frac{\pi}{6}\right) & -\cos \left(\theta_e - \frac{\pi}{2}\right)
\end{bmatrix}
\begin{bmatrix}
    x_a \\
    x_c
\end{bmatrix}$$

(3.6)

$x_0$ is the zero-axis which is perpendicular to both $d$ and $q$. For a balanced three-phase system $x_0$ is always equal to zero, and hence does not feature in (3.1).
and

\[
\begin{bmatrix}
  x_a \\
  x_c \\
\end{bmatrix} = \begin{bmatrix}
  \cos \left( \theta_e - \frac{\pi}{2} \right) & -\sin \left( \theta_e - \frac{\pi}{2} \right) \\
  \cos \left( \theta_e + \frac{\pi}{6} \right) & -\sin \left( \theta_e + \frac{\pi}{6} \right) \\
\end{bmatrix} \begin{bmatrix}
  x_d \\
  x_q \\
\end{bmatrix}
\]

\[x_b = -(x_a + x_c)\] (3.7a)

See Appendix A.2 for derivations of the above transformations.

The electromagnetic torque, \( T_m \), acting on the motor shaft can be identified from (3.1c) as

\[T_m = \frac{3}{4} p \psi_m i_q = K_t i_q\] (3.8)

where \( K_t \) is known as the motor torque constant.

The stator resistance, \( R_s \), and self-inductance, \( L_s \), can be calculated by measuring the line-to-line resistance, \( R_{l-l} \), and line-to-line self-inductance, \( L_{l-l} \), from the terminals of the armature cable connected to the stator windings. For a wye-connected stator, as shown in Figure 3.12, \( R_s \) and \( L_s \) are

\[R_s = \frac{1}{2} R_{l-l}\] (3.9a)

\[L_s = \frac{1}{2} L_{l-l}\] (3.9b)

For a delta-connected stator, they are

\[R_s = \frac{3}{2} R_{l-l}\] (3.10a)

\[L_s = \frac{3}{2} L_{l-l}\] (3.10b)

Equation (3.2) can then be used to estimate \( L_m \) and \( L_l \). The moment of inertia of the motor shaft, \( J_m \), and the motor torque constant, \( K_t \), can be estimated from system identification of the axis mechanical dynamics (see Section 3.5.2). Given that the number of motor poles, \( p \), is known, (3.8) can then be used to determine the permanent magnet flux, \( \psi_m \).

For the \( A \)-axis of the RX7 the number of motor poles is 12. The stator is delta-connected, with a line-to-line resistance of 14.8 Ω and a line-to-line inductance of 68 mH.
MODELLING A CNC MACHINE TOOL AXIS

Figure 3.13: RX7 A-axis CAD diagram – complete axis (left) and rotating components only (right).

Figure 3.14: Large workpiece CAD diagram.

3.5.2 Axis Mechanical Dynamics

The final component of the mechanical plant model captures the mechanical dynamics of the motor rotor, the workpiece collet, and the actual workpiece. Figure 3.13 shows a Computer Aided Design (CAD) diagram of the RX7 A-axis, while Figure 3.14 shows a CAD diagram of the large workpiece, which was depicted in Figure 3.4. In reality the mechanical plant is a continuous elastic body, which to represent would require a model order approaching infinity. This is both impractical and unnecessary.

To determine what order model is most appropriate to capture the key dynamics of the system, modal analysis was carried out using the Finite Element Analysis (FEA) capabilities of SolidWorks (Steffen 2007). Incorporating the model of the large workpiece shown in Figure 3.14 with the components shown on the righthand side of Figure 3.13 produces a complete representation of the workpiece.

The collet is the apparatus used to hold or mount the workpiece within the machine; it is sometimes known as a chuck.
ing mechanical plant. The next steps were to constrain the motion at the bearing locations to rotational along the long axis, and apply a mesh to the model geometry. Figure 3.15 shows the constraints (green arrows) and the mesh. The largest nominal mesh size which could be applied to the model geometry was found to be 6mm.\(^4\) With everything defined, the modal analysis was then carried out.

The FEA should identify the vibrational modes and the frequencies at which they occur. Generally, modes up to the Nyquist frequency of the position and velocity controllers should be included in the model; since these modes have the potential to be excited by the controllers. From the modal analysis carried out, two torsional modes were identified at frequencies less than the Nyquist frequency. Figure 3.16(a) shows the axial twist along the length of the axis/workpiece in the first torsional mode, 3.16(b) shows the second torsional mode, and 3.16(c) shows a cross-section view of the 3D CAD model that is colour coded to correspond to the different sections of the mode shape graphs. From the graphs shown in Figures 3.16(a) and 3.16(b) it is possible to identify three segments where the slopes of the lines are essentially zero. These segments correspond to sections of the mechanical plant, which in this frequency range, can be approximated as being rigid bodies. Likewise, the segments where the axial twist varies, are assumed to be the spring-damper elements; two in this case. The labels in Figure 3.16(c) indicate the

\(^4\)A finer mesh results in an FEA that takes too long to converge to a solution.
approximate location of the identified rigid bodies and spring-damper elements. The three rigid bodies are termed the motor \((m)\), the drawbar \((d)\) and the workpiece \((w)\). The conclusion from the FEA is that, in the considered frequency range, the mechanical plant can be adequately represented by a three-inertia model.

After defining the angular positions of the three rigid bodies as \(\theta_m\), \(\theta_d\) and \(\theta_w\),
the linear equations of motion are readily found to be

\[ J_m \ddot{\theta}_m = T_m - T_f - \left[ k_{md}(\theta_m - \theta_d) + c_{md}(\dot{\theta}_m - \dot{\theta}_d) \right] \]
\[ J_d \ddot{\theta}_d = \left[ k_{md}(\theta_m - \theta_d) + c_{md}(\dot{\theta}_m - \dot{\theta}_d) \right] \]  
\[ J_w \ddot{\theta}_w = \left[ k_{mw}(\theta_m - \theta_w) + c_{mw}(\dot{\theta}_m - \dot{\theta}_w) \right] - T_l \]  

(3.11a) (3.11b) (3.11c)

where \( J_m \), \( J_d \) and \( J_w \) are respectively the motor, drawbar and workpiece moments of inertia, \( T_m \) and \( T_f \) are respectively the electromagnetic torque and friction torque acting on the motor, \( T_l \) is the load torque acting on the workpiece, \( k_{md} \) and \( k_{mw} \) are respectively the torsional stiffnesses of the drawbar shaft and workpiece shaft, and \( c_{md} \) and \( c_{mw} \) are respectively the torsional damping coefficients of the drawbar shaft and workpiece shaft. The torque acting between the motor and drawbar rigid bodies is

\[ T_{md} = k_{md}(\theta_m - \theta_d) + c_{md}(\dot{\theta}_m - \dot{\theta}_d) \]  

(3.12)

and between the motor and workpiece rigid bodies is

\[ T_{mw} = k_{mw}(\theta_m - \theta_w) + c_{mw}(\dot{\theta}_m - \dot{\theta}_w) \]  

(3.13)

The next task was to parameterise the model given by (3.11) via system identification. As outlined in Appendix A.4, system identification is typically carried out using measured data obtained from experimentation. Furthermore, for data collected under closed-loop operation an appropriate identification technique is the well-established Predictive Error Method (PEM) (Ljung 1999). This is the approach adopted here.

The first step is to design an experiment to collect input-output data that is suitable for use with the PEM. The experiment involves applying a known (or at least measurable) signal to the system input, while measuring the response at the system output. The input for the mechanical plant is the torque producing current, \( i_q \), and the output is the motor velocity, \( \omega_m \) (see Figure 3.8). Due to safety
MODELLING A CNC MACHINE TOOL AXIS

constraints, it is not possible to open the velocity loop, hence the signal used to excite the system must be applied at the motor velocity reference, \( \omega_r \). It is important that the excitation signal is suitably rich so as to excite all critical dynamics within the system. A signal with this property is the linear swept-frequency sinusoid, commonly known as a chirp signal. It has the form

\[
\omega_r(k) = A_c \sin \left(2\pi \cdot \frac{1}{2} f_c(k) \cdot kT \right) + B_c
\]  

(3.14)

where \( A_c \) is the amplitude of the chirp, \( B_c \) is a constant offset, and \( f_c(k) \) is the instantaneous frequency of the signal at sample \( k \), and is defined as

\[
f_c(k) = \frac{k}{2KT}
\]  

(3.15)

where \( K \) is the number of samples in the data set. Using (3.14) ensures that the input signal will excite across the full frequency spectrum up to the Nyquist frequency \( \left( \frac{1}{2T} \right) \). The data logging system on the ANCA DSD allows four variables, each of 2048 data points to be logged simultaneously; hence \( K \) is 2048.

The constant offset parameter, \( B_c \), is used to place the system within a specific operating region. The model given by (3.11) includes friction torque, \( T_f \), which is known to be nonlinear around zero velocity (and is investigated in detail in Section 3.5.3). Here the goal is to identify the inertias, stiffnesses, damping coefficients and the torque constant, hence the constant offset is used to move the operating region away from nonlinear frictional effects around zero velocity. Hence \( B_c \) is chosen so that the motor velocity reference, \( \omega_r \), is strictly positive so that \( B_c > A_c \).

Using (3.8) and (3.11), with constant motor velocity at the operating point, \( \omega_m = B_c \), produces

\[
T_f(\omega_m) = T_m = K_i q(\omega_m)
\]  

(3.16)

After defining the zero mean \( q \)-axis current, \( \bar{i}_q \), and the zero mean motor velocity, \( \bar{\omega}_m \), (3.16) may be rewritten as

\[
T_f^*(\bar{\omega}_m) = K_i \bar{i}_q - (J_m \ddot{\omega}_m + T_{md} + T_{mw}) = B_m \bar{\omega}_m
\]  

(3.17)
where $B_m$ is the viscous damping coefficient. The last step assumes that in the operating region $\omega_m \in [B_c - A_c, B_c + A_c]$, $T_f^*$ varies linearly with $\bar{\omega}_m$. This is illustrated graphically in Figure 3.17.

![Friction model for system identification](image)

**Figure 3.17:** Friction model for system identification.

The value chosen for parameter $A_c$ is a trade-off between signal-to-noise ratio, saturation of the actuator (current limit) and safety. For a good signal-to-noise ratio, $A_c$ should be large, but to avoid actuator saturation and for safety, $A_c$ should be small. A good compromise was found with $A_c = 3\text{ RPM}$ and $B_c = 13\text{ RPM}$. With the set of parameters for (3.14) selected, the experiment was conducted 10 times so as to enable measurement noise to be reduced via averaging.

Figure 3.18 shows an example of the measurement data to be used for system identification, obtained from the A-axis of the RX7 loaded with the large work-piece (as in Figure 3.4). The first chart is the motor velocity reference, where the red line is the constant offset $B_c$ from (3.14)$^5$, and in blue, the superimposed chirp signal. The second chart is the $q$-axis current, which is the input signal for the system identification process. The third chart is the motor velocity, which is the output signal for the system identification process.

---

$^5$The offset is not actually constant because it originates from the closed position loop, where the input is a ramp signal.
Having collected the required data, and removed the mean signals, it is now possible to complete the system identification work for the mechanical plant using the PEM. The MATLAB `pem` function is used for the task. It implements the PEM method outlined in Section A.4.2. The function can operate on the assumption of a black-box model, but since the model structure is known, superior outcomes are likely to result from using a grey-box model. The `pem` function requires a set of initial values for each of the model parameters. From the FEA, values for the inertias and stiffnesses were calculated; initial values for the other parameters were inferred from manufacturer’s data. The initial values used are given in Table 3.2. After building the grey-box model based on (3.11) substituted with (3.8) and (3.17), the `pem` function is executed using the measured data. The results are shown in Table 3.2.

Given the estimate for the torque constant, $K_t$, shown in Table 3.2, the permanent magnet flux, $\psi_m$, can be calculated using (3.8) as 0.3978 Wb.

Table 3.3 shows the damping ratios and natural frequencies of the poles of...
the mechanical plant with large workpiece. As expected, there are two sets of complex poles. These represent the two flexible modes within the system. They are both lightly damped, with the dominant mode being between the motor shaft and the workpiece ($-31.56 \pm 2768i$).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initial Value</th>
<th>Optimal Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_t$</td>
<td>3.5100</td>
<td>3.5801</td>
<td>N·m/A</td>
</tr>
<tr>
<td>$J_m$</td>
<td>0.0172</td>
<td>0.0127</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_d$</td>
<td>0.0002</td>
<td>0.0002</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_w$</td>
<td>0.0528</td>
<td>0.0500</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$B_m$</td>
<td>0.0500</td>
<td>0.0264</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$c_{md}$</td>
<td>0.0700</td>
<td>0.0658</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$c_{mw}$</td>
<td>0.6000</td>
<td>0.6309</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$k_{md}$</td>
<td>56 364</td>
<td>73 570</td>
<td>N·m/rad</td>
</tr>
<tr>
<td>$k_{mw}$</td>
<td>78 528</td>
<td>78 603</td>
<td>N·m/rad</td>
</tr>
</tbody>
</table>

Table 3.2: Mechanical plant parameters.

$\begin{array}{ccc}
\text{Eigenvalue} & \text{Damping} & \text{Frequency [Hz]} \\
-0.000 & 1.000 & 0.000 \\
-0.419 & 1.000 & 0.067 \\
-31.56 \pm 2768i & 0.011 & 440 \\
-167.5 \pm 19331i & 0.009 & 3076 \\
\end{array}$

Table 3.3: Damping ratios and natural frequencies of the poles of the mechanical plant with large workpiece.

When considering the small workpiece (left hand side of Figure 3.4), the results of the FEA indicate that the model reduces to a two-inertia system. In the considered frequency range, the small workpiece is not compliantly coupled to the motor shaft. The equations of motion are thus

\begin{align*}
J_m \ddot{\theta}_m &= T_m - T_f - \left[ k_{md}(\theta_m - \theta_d) + c_{md}(\dot{\theta}_m - \dot{\theta}_d) \right] - T_l \tag{3.18a} \\
J_d \ddot{\theta}_d &= \left[ k_{md}(\theta_m - \theta_d) + c_{md}(\dot{\theta}_m - \dot{\theta}_d) \right] \tag{3.18b}
\end{align*}

where the parameters have the same meaning as before, except that the inertia of the workpiece, which is very small, is included in $J_m$. Using a separate collection of experimental data, the values for the parameters are found to be essentially identical to those shown in Table 3.2. Table 3.4 shows the damping ratios and
natural frequencies of this configuration. Apart from the missing pole pair that is attributed to the compliantly coupled large workpiece, the values are effectively equal to those in Table 3.3.

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Frequency [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>−0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>−2.044</td>
<td>1.000</td>
<td>0.325</td>
</tr>
<tr>
<td>−167.1 ± 19329i</td>
<td>0.009</td>
<td>3076</td>
</tr>
</tbody>
</table>

Table 3.4: Poles, damping ratios and natural frequencies of mechanical plant with small workpiece.

To validate that the model with the identified parameters adequately represents the actual response of the system it is possible to compare measured results to that of a simulation. For the case of the large workpiece, Figure 3.19 shows a comparison between the measured motor velocity, and that of a simulation using the same motor current input. While not identical, the simulated response does mimic the important characteristics of the measured data: specifically, the large resonance which occurs at around 0.12s, the sustained high frequency oscillations, and the general drift about zero velocity.

3.5.3 Friction Model

There are several well known models which can be used to characterise the mechanics of friction in machines. A comprehensive review of the literature in this area is included in Appendix A.1. Of those presented, the two models which are best suited are presented in Armstrong-Hélouvry et al. (1994) and Canudas de Wit et al. (1995). The so called LuGre friction model (Canudas de Wit et al. 1995) has the advantage that it can capture the frictional effect during both steady-state and changing velocity regimes. However it is very difficult to obtain experimental data that is suitable for parameterising such a model. The Armstrong-Hélouvry model on the other hand naturally captures the key steady-state characteristics of the experimentally observed friction, including: static, Coulomb and
vicious friction, as well as the Stribeck effect. As will be highlighted shortly, it is straightforward to obtain suitable experimental data during stead-state operation.

The model for the friction torque acting on the motor shaft, $T_f$, is hence based on Armstrong-Hérouvry et al. (1994), but modified to also include the Karnopp friction model (Karnopp 1985). The Karnopp friction model is a remedy to difficulties associated with simulating a system which includes a discontinuous transfer characteristic. Friction always opposes relative motion, hence is discontinuous at zero velocity. When integrating discontinuous differential equations, the appropriate value of the derivative must be used on each side of the discontinuity. Unfortunately, discontinuities generally occur inside integration subintervals, which presents computational issues when simulating. Hence by using a small neighbourhood of zero velocity $\varepsilon$, as defined by the Karnopp friction model, the
MODELLING A CNC MACHINE TOOL AXIS

simulation difficulty is reduced. The friction model is given by

\[
T_f(\omega_m, T_e) = \begin{cases} 
    
    (T_c + B_m |\omega_m| + \frac{T_{st}}{1 + (\omega_m/\omega_s)^2}) \text{sgn}(\omega_m) & |\omega_m| > \varepsilon \\
    T_e & |\omega_m| \leq \varepsilon, |T_e| \leq T_s \\
    T_s \text{sgn}(T_e) & |\omega_m| \leq \varepsilon, |T_e| > T_s
\end{cases}
\] (3.19)

which is a function of the motor angular velocity, \( \omega_m \), and the total external torque, \( T_e = T_m - (T_{md} + T_{mw}) \). There are four parameters for the model: the Coulomb friction torque, \( T_c \); the viscous damping coefficient, \( B_m \); the magnitude of the Stribeck friction torque, \( T_{st} \) and the characteristic velocity of the Stribeck friction torque, \( \omega_s \). The total friction torque at breakaway, known as the static friction torque, is \( T_s = T_c + T_{st} \). A plot which highlights the various features of the friction model appears shortly, in Figure 3.22.

The next step was to parameterise the model represented by (3.19). To achieve this, a series of data collecting experiments were conducted. The data was used to estimate the model parameters in a least-squares fit. The experiments involved running the axis motor with a constant motor velocity reference, \( \omega_r \), while logging \( q \)-axis current, \( i_q \), and motor velocity, \( \omega_m \). The constant motor velocity references used for the experiments are \( \omega_r \in V \) where

\[
V = \pm \{0.2, 0.3, \cdots, 1.0, 1.2, 1.4, \cdots, 5.0, 7.5, 10.0, 12.5, 13.9\} \text{ RPM}
\]

Due to a combination of cogging torque and torque ripple, as shown in Figure 3.20, the motor current required to maintain constant motor velocity varies sinusoidally. The data logging system on the ANCA DSD allows just 2048 data points to be logged in each experiment. By ensuring that the data record contains only full cycles of the cogging/ripple torque, the data can be averaged to arrive at a constant for the motor current.

Since during the experiments there is no external load torque acting on the axis, and it is not accelerating, the electromagnetic torque can be solely attributed to overcoming friction and (3.16) may be applied. Using the torque constant, \( K_t \), shown in Table 3.2, it is possible to construct a plot of friction torque versus motor
velocity. Figure 3.21 shows the plots for both the large and small workpieces, which are depicted in Figure 3.4.

Using a least-squares optimisation to fit the model given by (3.19) to the data, with $\varepsilon = 0$, the parameters in Table 3.5 are found. Using these parameters to plot the model on the data in Figure 3.21, again with $\varepsilon = 0$, reveals that the parameterised model fits the data well. Furthermore, this result shows that the friction acting on the motor shaft is largely dependant on the size (mass) of the workpiece.

To illustrate the effect of including the Karnopp friction model on (3.19), take
the case where $\varepsilon = 0.02 \text{ rad/s}$. Figure 3.22 shows the friction torque, $T_f$, as a function of motor velocity, $\omega_m$, and total external torque, $T_e$, for the large workpiece. Within the small boundary around zero velocity, the friction torque acts as a ramp, reflecting the total external torque up to the level of the static friction torque, $T_s$. Beyond this point, the total torque $(T_e - T_f)$ acting on the motor shaft is no longer zero and the axis will begin to accelerate away from zero velocity. At absolute motor velocities greater than $\varepsilon = 0.02 \text{ rad/s}$, the friction torque is independent of the total external torque. The Stribeck effect can clearly be seen in the operating region displayed in the figure.

Figure 3.21: Friction model experimental data and model fit for both large and small workpieces.
Figure 3.22: Effect of Karnopp friction model for the axis with large workpiece and $\varepsilon = 0.02 \text{ rad/s}$. 
MODELLING A CNC MACHINE TOOL AXIS

3.6 REDUCED-ORDER LINEAR MODEL

The goal of producing a lower-order, linear version of the model that was produced in the previous sections is to facilitate both linear analysis and model-based controller design. The major areas of the model which are to be simplified are the current loop (which includes the current controller and all the electrical dynamics) and the nonlinear friction model. Figure 3.23 shows the structure of the reduced-order linear model. Notice that in comparison to Figure 3.11, the interface between the DSD and the actuator is now the $q$-axis current signal, and most importantly, all the elements are represented by linear subsystems.

3.6.1 Reduced-Order Linear Current Loop Model

Since the response of the electrical system is highly influenced by the current controller, the controller’s dynamics are included in a simplified linear model of the electrical system. Moreover, it is typical that the electrical and mechanical dynamics are of such sufficiently different time scales that the fast dynamics of the electrical system may be ignored. That is, the (nonlinear) dynamics of the power electronics switching will be disregarded.

From inspection of the closed-loop frequency response data in Figure 3.24, it is apparent that a second-order model should be adequate to capture the key dynamics of the electrical system. Thus the model used to fit the data is a strictly

![Figure 3.23: Structure of the reduced-order linear mechanical plant model.](image-url)
proper, parametric, second-order transfer function of the form

\[ G_{i_f q} = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0} \] (3.20)

which has two poles and a single zero.

Since the data is collected under closed-loop operation, pem is again used to parameterise the model. The result is shown in Figure 3.25 and is given by

\[ G_{i_f q} = \frac{-0.2398 z + 0.3253}{z^2 - 1.7927 z + 0.8892} \] (3.21)

where the sampling time is 50 µs.\(^6\) The damping ratio of this second-order system is 0.183 and the natural frequency is 1.02 kHz.

In conjunction with using this reduced-order linear current loop model, the PMSM block from Figure 3.11 is replaced by a linear gain – the motor torque constant \( K_t \) from (3.8).

\(^6\)While the actual current loop executes at 50 µs, the measurement data was captured at 250 µs, and hence the initial transfer function fitted to the data was also at 250 µs. The MATLAB function \textit{d2d} is used to convert to the result given in (3.21).
Figure 3.25: Closed-loop frequency response from motor current reference to motor current for the discrete model of the current servo.

3.6.2 Linear Friction Model

For the friction model to be linear, the friction torque acting on the motor shaft must be assumed to be purely viscous, so that

\[ T_f^* = B^*_m \omega_m \]  

(3.22)

where \( B^*_m \) is the effective viscous damping coefficient. Clearly it is not possible to simply replace the nonlinear friction model given by (3.19) with (3.22), since there is no value for \( B^*_m \) which will be suitable for all operating regimes. The standard linearisation approach is to find a local linear approximation of the nonlinear function about specific operating points. In effect this would produce a piecewise linear model, however given the severe nonlinearity about zero velocity, the model would need to be re-linearised about many operating points to adequately capture the friction dynamics. Furthermore, scheduling a piecewise model about the area of interest violates the requirements for linear analysis. Instead an averaging approach is used to develop a linear friction model which will enable
linear analysis and design techniques to be utilised. Of course, this approach will ignore the effect of Coulomb friction. However final validation of controller designs before implementation will be done using the full nonlinear model and hence any unaccounted for side effects should be uncovered.

The averaging approach relies upon the existence of a known velocity profile. Given this profile it is possible to calculate a value for $B^*_m$. This is achieved by considering the energy dissipation that would result from the nonlinear friction model, $T_f$, and equating that to the energy dissipation from the linear friction model, $T_f^*$ (Thomson and Dahleh 1998). That is

$$E_f = \int_{t_1}^{t_2} T_f^* \omega m \, dt = \int_{t_1}^{t_2} T_f \omega m \, dt \quad (3.23)$$

For simplicity, assume a sinusoidal velocity profile of the form

$$\omega_m = A \sin(\omega t) \quad (3.24)$$

The first integral in (3.23) can be evaluated over a single cycle of the velocity profile as

$$E_f = \int_{t_1}^{t_2} T_f^* \omega_m \, dt$$

$$= \int_{-\frac{\pi}{\omega}}^{\frac{\pi}{\omega}} (B^*_m A \sin(\omega t)) (A \sin(\omega t)) \, dt$$

$$= \frac{B^*_m A^2 \pi}{\omega} \quad (3.25)$$

Likewise, evaluating the second integral using (3.19) with $\varepsilon = 0$ produces

$$E_f = \int_{t_1}^{t_2} T_f \omega_m \, dt$$

$$= \int_{-\frac{\pi}{\omega}}^{\frac{\pi}{\omega}} \left( T_c \text{ sgn} \left( A \sin(\omega t) \right) + B_m A \sin(\omega t) + \frac{T_{st} \text{ sgn} \left( A \sin(\omega t) \right)}{1 + (A \sin(\omega t)/\omega_s)^2} \right) (A \sin(\omega t)) \, dt$$

$$= \frac{4T_c A}{\omega} + \frac{B_m A^2 \pi}{\omega} + \frac{2\omega_s^2 T_{st}}{\omega} \log e \left( \frac{\sqrt{\omega_s^2 + A^2} + A}{\sqrt{\omega_s^2 + A^2} - A} \right) \quad (3.26)$$

Equating (3.25) and (3.26) and solving for $B^*_m$ produces

$$B^*_m = \frac{4T_c}{A \pi} + B_m + \frac{2\omega_s^2 T_{st}}{A^2 \pi \sqrt{\omega_s^2 + A^2}} \log e \left( \frac{\sqrt{\omega_s^2 + A^2} + A}{\sqrt{\omega_s^2 + A^2} - A} \right) \quad (3.27)$$
Using the parameters shown in Table 3.5, Figure 3.26 shows how the effective viscous friction coefficient given by (4.18) varies with $A$, the amplitude of the sinusoidal velocity profile. Note that this expression is not a function of the frequency of the oscillation, $\omega$. The first term in (4.18) is the well known standard result for purely Coulomb friction and is given, for example, by Thomson and Dahleh (1998). From Figure 3.26 it can be seen that, as expected, the effective friction coefficient is very large close to zero velocity, but quickly asymptotes towards the viscous friction coefficient value from the nonlinear model. Given a sinusoidal velocity profile with an amplitude of 1.5 rad/s, Figure 3.27 shows how the effective linear friction model compares to the true nonlinear model.

The conclusion of this section is that the effective viscous friction coefficient is dependent on the magnitude of the motor velocity, and as such, must be selected accordingly. The effect of using different values of $B_m^*$ for linear analysis is explored further in Section 4.1.4.
3.7 PERFORMANCE OBJECTIVES

The CNC grinding machines introduced in the previous sections are required to produce workpieces with a high degree of geometric accuracy, while at the same time achieving the highest possible material removal rate that the grinding process will allow. The geometric tolerances which the machines are required to achieve are measured in microns (thousands of a millimetre). Grinding to within 5 microns of the desired geometry is considered acceptable. While clearly it is the combined performance of all axes which dictates the overall geometric accuracy of the finished workpiece, improving the individual tracking errors of each axis is one way of achieving this. Thus the aim of this thesis is to reduce the tracking error for the headstock axis to below 10 deg for fast contouring moves. As will be seen in later chapters the tracking error for this axis using the existing control approach is as high as 30 deg.
3.8 CONCLUSION

This chapter began by introducing the experimental facilities that are made available by the industrial partner. These facilities are used extensively in most of the remaining sections of the research. The core piece of equipment is a commercial 5-axis CNC grinding machine, which utilises the latest technology in digital motion control. The modelling work undertaken in this chapter is based on a specific axis of this machine.

The modelling work completed in this chapter began with a comprehensive model of the direct-drive RX7 A-axis. The core of the complete model is represented by the discrete control algorithms implemented on the DSP, the electrical system employing PWM to generate variable frequency 3-phase power, the equations of motion for the PMSM (3.1), the mechanical plant dynamics (3.11) and (3.18), and the nonlinear friction model (3.19). This model is used initially to gain an understanding of the problematic dynamics and then, where possible, as a final test-bed for new controller designs.

A reduced-order linear representation was also presented. Justification for the structure and relative order of the various components is provided, along with details of experiments used to capture data used for system identification. In contrast, for the reduced-order linear model the electrical system and PMSM equations are replaced with a linear second-order transfer function (3.20), and the nonlinear friction model by a linear interpretation given by (3.22). Key to this representation of the overall servo drive dynamics is the fact that the model is constructed from linear components only. The reduced-order linear model is used extensively in the analysis sections of the research and later as the core model for model-based controller design.

While the complete model is the closest representation of the actual system, its complexity can make it prohibitively slow to use for simulation work. Hence quite often the electrical system is replaced by the linear second-order transfer
function mentioned previously; however the nonlinear friction model is retained.

In the next chapter, system analysis will be carried out using the linear model developed in this chapter. The analysis will aim at identifying the cause of the experimentally observed instability when the large workpiece is mounted in the machine. And furthermore, the limitations of the existing control architecture in dealing with a servo drive axis that exhibits this type of behaviour will be identified.
The major contributions of Chapter 4 have been published in Stephens et al. (2010) and accepted for publication in the Journal of Dynamic Systems, Measurement, and Control.

As described in Section 2.6, it has been observed anecdotally on a variety of commercial direct-drive systems that the controller performance degrades as a result of changes to the mechanical configuration of the axis. It is experimentally observed that when the large workpiece (depicted in Figure 3.4) is mounted into the machine, the A-axis begins to vibrate under closed-loop position control. The frequency of the observed oscillation was identified in the Introduction (Figure 1.8) as approximately 410 Hz. When first enabling closed-loop control, the state of the axis corresponds to a point of unstable equilibrium. However, with only a slight disturbance (perturbation of this state) the system quickly transitions into the observed limit-cycle. The hypothesis is that the lightly-damped mechanical resonance corresponding to the first torsional mode, shown in Figure 3.16(a), contributes to the observed instability.

This chapter will analyse the system to confirm this theory, and further, identify the limitations of the existing control architecture in dealing with it. In Section 4.1, the linear reduced-order model developed in Section 3.6 will be utilised in a linear analysis of the system to uncover the root cause of the instability. Section 4.3 summarises the results and identifies the limitations of the existing control architecture.
PERFORMANCE LIMITATIONS OF EXISTING CONTROL ARCHITECTURE

4.1 LINEAR ANALYSIS OF EXISTING CONTROL ARCHITECTURE

This section explores the cause of the instability using linear analysis techniques.

4.1.1 Axis Stability – Excluding Current Servo Dynamics

Linear analysis techniques require either the system be continuous or discrete with a single sample-rate. The servo drive system structure (Figure 3.7) includes continuous-time elements (e.g. the mechanical plant) and discrete-time elements (e.g. the control algorithms). If the discrete-time elements operate at different sample-rates, then the system is classified as being multirate. As described in Section 3.3, the cascaded controllers do indeed operate at different sample-rates.

There exists a straightforward approach for discretising the continuous-time elements. The plant in Figure 4.5 can be represented by a Linear Time-Invariant (LTI) state-space model of the form

\[
\dot{x}(t) = A_P x(t) + B_P u_2(t) \tag{4.1a}
\]

\[
y(t) = C_P x(t) + D_P u_2(t) \tag{4.1b}
\]

Since the input to this is the output of a Zero Order Hold (ZOH) with sample-period \(T\), the equivalent discrete-time model for (4.1) has the form

\[
x(k + 1) = A_P x(k) + B_P u_2(k) \tag{4.2a}
\]

\[
y(k) = C_P x(k) + D_P u_2(k) \tag{4.2b}
\]

where (Franklin et al. 2006)

\[
A_P = e^{A_P T} \tag{4.3a}
\]

\[
B_P = \left( \int_0^T e^{A_P \eta} d\eta \right) \overline{B_P} \tag{4.3b}
\]

\[
C_P = \overline{C_P} \tag{4.3c}
\]

\[
D_P = \overline{D_P} \tag{4.3d}
\]

In practice, this transformation can be achieved using the MATLAB function c2d.
Unfortunately, converting the multirate discrete-time elements into a single-rate representation is not as straightforward. However, since the bandwidth of the current servo is higher than that of the position/velocity loops and mechanical plant, as a first attempt to analyse the system stability, the current servo dynamics will be ignored. From the surveyed literature it appears that this is the conventional approach. Figure 4.1 shows the structure of the system to be analysed.

Tables 4.1 and 4.2 show the poles of the mechanical plant for the large and small workpieces respectively, after being discretised at sample period $250 \mu s$. The continuous-time linear equations of motion are given in (3.11) and (3.18). The friction model $T_f$ used is given by (3.22). The parameters for the models are found in Table 3.2. An important aspect of these results is that the continuous time poles at $3076 \text{ Hz}$ are aliased back within the Nyquist range ($< 2 \text{ kHz}$) to $923 \text{ Hz}$ when the model is discretised. This is a legitimate result, since dynamics outside the Nyquist range will appear at the aliased frequency and subsequently have an influence on the closed-loop behaviour.

Another aspect of the results displayed in Tables 4.1 and 4.2 that should be highlighted, is that one of the discretised poles (located at $1.00 + 0i$) is an approximation. A non-zero frequency indicates the existence of an (albeit small) imaginary component, which would also necessitate the existence of a complementary pole. There are two possibilities for the true structure of the model: one,
the true frequency of this pole is 0.000 Hz, or two, the other pole located at 1.00+0i is the complimentary pole and has a frequency of 0.067 Hz (or 0.325 Hz in the case of the large workpiece).

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Frequency [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1.00</td>
<td>1.000</td>
<td>0.067</td>
</tr>
<tr>
<td>0.764 ± 0.633i</td>
<td>0.011</td>
<td>440</td>
</tr>
<tr>
<td>0.115 ± 0.952i</td>
<td>0.029</td>
<td>923</td>
</tr>
</tbody>
</table>

Table 4.1: Damping ratios and natural frequencies of the poles of the discretised \(T = 250\mu s\) mechanical plant with large workpiece.

Plotting the frequency response of the open-loop transfer function from motor current reference to motor velocity (Figure 4.2) reveals a large resonance for the large workpiece configuration. Figures 4.3(a) and 4.3(b) show the closed-loop poles of the position loop, for the small and large workpiece configurations, respectively. The controller parameters used in this analysis are the default values for the axis. They were tuned for optimal performance for small workpieces. Note that the extra two poles in Figure 4.3(b) are due to the extra second-order equation of motion in (3.11), when compared to (3.18). Since all the poles in both graphs are within the unit circle, the analysis suggests that both axis configurations are stable. This result contradicts the experimentally observed instability (leading to a limit-cycle) that occurs when the large workpiece is loaded into the machine.

Since the model of the mechanical plant is believed to be of sufficient order, the conclusion is made that there must be other critical aspects of the system not
Figure 4.2: Open-loop frequency response from motor current reference to motor velocity for the discrete model of the servo drive axis loaded with both small and large workpieces. Excludes the current servo dynamics.

being considered. The next two sections explore the effect the fast-rate elements (together termed the current servo) have on the overall dynamics of the system.
Figure 4.3: Closed-loop poles of the motor position loop for the discrete model of the servo drive axis loaded with both small and large workpieces. Excludes the current servo dynamics.
Figure 4.4: Simulated response of the full-order nonlinear model to highlight the limit-cycle behaviour during stationary operation of the axis loaded with the large workpiece. Position reference is the dashed line (red) and the simulated position feedback is the solid line (blue). Also shown is the motor current reference. The frequency of the observed oscillations is approximately 425 Hz.

4.1.2 Single-Rate Representation of a Multirate System

This section investigates the effect of including a more detailed model of the fast inner loop dynamics on the system stability. To test the theory that these dynamics are critical to predicting the observed instability, the nonlinear multirate simulation model of the complete closed-loop system that was developed in Chapter 3 is used.\footnote{Included is measurement noise to perturb the system from its unstable equilibrium point.} The simulated response to a constant position reference is shown in Figure 4.4. This closely resembles the experimentally observed limit-cycle from Figure 1.8.

To include the current servo dynamics in the analysis, it will be necessary to convert the multirate system into an equivalent single-rate representation. A typical multi-rate cascaded servo drive control system is depicted in Figure 4.5. In a
paper by Pillay and Krishnan (1988), the authors develop an analysis technique for multirate servo drive systems. The technique is based on the sampler decomposition technique (Kuo 1992), which is quite complex to apply to systems where the ratio between multirate sample periods is larger than three. Besides this work, an approach for capturing the effect of the multirate system architecture does not appear to have been explicitly accounted for previously in the literature on CNC machine tool modelling and control.

For the case where the lowest common ratio between all discrete sample periods in the system is a positive integer, denoted $H$, then an approach for generating a single sample-rate representation of the multirate system has been devised. The method presented here is based on an approach initially proposed by Berg et al. (1988), but tailored for this specific application. It is best illustrated by way of an example. Take the case where $H = 3$. The sample index correspondence between the fast-rate system ($h$) and the slow-rate system ($k$) is

$$
\begin{align*}
  h & : 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad \cdots \\
  k & : 0 \quad 1 \quad 2 \quad \cdots 
\end{align*}
$$

The fast-rate LTI subsystem contained within the dashed box of Figure 4.5 can be represented in discrete state-space form as

$$
\begin{align*}
x(h + 1) &= Ax(h) + Bu_1(h) \quad \text{(4.5a)} \\
y(h) &= Cx(h) + Du_1(h) \quad \text{(4.5b)}
\end{align*}
$$
LINEAR ANALYSIS OF EXISTING CONTROL ARCHITECTURE

Iterating (4.5a) using \( h \), the fast-rate sample index sequence from (4.4), produces

\[
x(1) = Ax(0) + Bu_1(0) \tag{4.6a}
\]

\[
x(2) = Ax(1) + Bu_1(0) = A(Ax(0) + Bu_1(0)) + Bu_1(0)
= A^2x(0) + (A + I)Bu_1(0) \tag{4.6b}
\]

\[
x(3) = Ax(2) + Bu_1(0) = A(A^2x(0) + (A + I)Bu_1(0)) + Bu_1(0)
= A^3x(0) + (A^2 + A + I)Bu_1(0) \tag{4.6c}
\]

\[
x(4) = Ax(3) + Bu_1(3) \tag{4.6d}
\]

\[
x(5) = Ax(4) + Bu_1(3) = A(Ax(3) + Bu_1(3)) + Bu_1(3)
= A^2x(3) + (A + I)Bu_1(3) \tag{4.6e}
\]

\[\vdots\]

In the preceding series of equations, the input \( u_1 \) is held constant for three samples since it originates from the slow-rate controller (see Figure 4.5). It is the effect of holding the input which allows the subsequent three iterations to be represented exclusively in terms of the first iteration’s state and input variables.

Next, iterating (4.5b) using the results from (4.6) yields

\[
y(0) = Cx(0) + Du_1(0) \tag{4.7a}
\]

\[
y(1) = Cx(1) + Du_1(0) = C(Ax(0) + Bu_1(0)) + Du_1(0)
= CAx(0) + (CB + D)u_1(0) \tag{4.7b}
\]

\[
y(2) = Cx(2) + Du_1(0) = C(A^2x(0) + (A + I)Bu_1(0)) + Du_1(0)
= CA^2x(0) + (CA + I)B + D)u_1(0) \tag{4.7c}
\]

\[
y(3) = Cx(3) + Du_1(3) \tag{4.7d}
\]

\[
y(4) = Cx(4) + Du_1(3) = C(Ax(3) + Bu_1(3)) + Du_1(3)
= CAx(3) + (CB + D)u_1(3) \tag{4.7e}
\]

\[\vdots\]
PERFORMANCE LIMITATIONS OF EXISTING CONTROL ARCHITECTURE

By restating (4.6c) and (4.7a) as

\[ x(3) = A^3 x(0) + (A^2 + A + I)B u_1(0) \]  
\[ y(0) = Cx(0) + Du_1(0) \]

and comparing the index numbers with those in (4.4), it can be seen that the slow-rate representation of the fast-rate subsystem is equivalent to

\[ x(k + 1) = A^3 x(k) + (A^2 + A + I)B u_1(k) \]  
\[ y(k) = Cx(k) + Du_1(k) \]

Defining

\[ A_3 = A^3 \]  
\[ B_3 = (A^2 + A + I)B \]  
\[ C_3 = C \]  
\[ D_3 = D \]

and substituting into (4.9) produces

\[ x(k + 1) = A_3 x(k) + B_3 u_1(k) \]  
\[ y(k) = C_3 x(k) + D_3 u_1(k) \]

which models the dynamics of the subsystem shown within the dashed box in Figure 4.5, as seen by the other elements of the system. For arbitrary values of \( H \), it can be shown that (4.10) becomes

\[ A_H = A^H \]  
\[ B_H = (A^{H-1} + A^{H-2} + \cdots + A + I)B \]  
\[ C_H = C \]  
\[ D_H = D \]

As an additional check it is also possible to use the MATLAB function dlinmod to determine an approximation for (4.11).
4.1.3 Axis Stability – Including Current Servo Dynamics

In this section, the technique developed in the previous section will be used to include the current servo dynamics in a linear analysis of the system stability. Figure 4.6 shows the relationship between the actual multirate system and the equivalent single-rate representation.

As shown in Figure 4.6, the ratio between the different update rates for the discrete-time elements is given by

$$H = \frac{250 \mu s}{50 \mu s} = 5 \quad (4.13)$$

There are three distinct elements within the current servo which are explained below.

Computation delays are an inevitable feature of real servo drive control algorithms calculated on a digital processor. In the system currently being studied, the control input calculated by the velocity controller executing at the slow sample-rate, is not immediately available to the current controller executing at
PERFORMANCE LIMITATIONS OF EXISTING CONTROL ARCHITECTURE

the fast sample-rate. This delay is typically an integer number, denoted \( d \), of fast-rate samples and can be represented in state-space form as

\[
x_d(h + 1) = A_d x_d(h) + B_d i_r(h) \tag{4.14a}
\]

\[
i_r^*(h) = C_d x_d(h) \tag{4.14b}
\]

where \( i_r \) is the current reference from the velocity controller, \( i_r^* \) is the reference delayed by \( d \) fast-rate samples, \( x_d \) is the state vector of length \( d \), and

\[
A_d = \begin{bmatrix} 0_r & 0 \\ I & 0_c \end{bmatrix}, \quad B_d = \begin{bmatrix} 1 \\ 0_c \end{bmatrix} \quad \text{and} \quad C_d = \begin{bmatrix} 0_r & 1 \end{bmatrix},
\]

where \( I \) is the identity matrix of size \((d - 1)\), and \( 0_c \) and \( 0_r \) are respectively column and row zero vectors of length \((d - 1)\). For the ANCA DSD there is a two-sample computation delay (i.e. \( d = 2 \)).

To reduce the influence of sensor measurement noise on the system, the current reference is low-pass filtered. The discrete filter is first-order and is implemented on the DSP at the fast sample-rate. It has the form

\[
x_{lp}(h + 1) = \frac{K - \omega_{lp} x_{lp}(h)}{K + \omega_{lp}} + \frac{1}{K + \omega_{lp}} i_r^*(h) \tag{4.15a}
\]

\[
i_f(h) = \frac{2\omega_{lp} K}{K + \omega_{lp}} x_{lp}(h) + \frac{\omega_{lp} K}{K + \omega_{lp}} i_r^*(h) \tag{4.15b}
\]

where \( i_r^* \) is the delayed current reference, \( i_f \) is the reference filtered by the low-pass filter, \( x_{lp} \) is the state, \( \omega_{lp} \) is the corner frequency and \( K = \frac{2}{T} \). The corner frequency used for the low-pass filter on the A-axis is 700 Hz or \( \omega_{lp} = 4398 \text{ rad/s} \).

The current controller/electrical system dynamics were found in Section 3.6.1 to be adequately represented by a discrete second-order model at the fast sample-rate. This second-order model represents the current loop in Figure 4.6.

Combining these three elements using the method in Appendix A.3, and applying the technique developed in Section 4.1.2, produces an equivalent single-rate representation of the multirate model of the system, as shown in the bottom of Figure 4.6. The open-loop frequency response from motor current reference to motor velocity for the discrete model of the servo drive axis is shown in Figure
Figure 4.7: Open-loop frequency response from motor current reference to motor velocity for the discrete model of the servo drive axis loaded with both small and large workpieces. Includes the current servo dynamics. The previous result, where the current servo dynamics were excluded, is shown by the dashed lines.

4.7. The figure also includes the results from the case where the current servo dynamics were ignored (i.e. Figure 4.2). This comparison shows that the inclusion of the fast sample-rate elements has the main effect of increasing the phase lag at high frequencies. Using the default controller parameters to plot the closed-loop poles of the position loop (Figure 4.8) reveals the expected outcome – the system is stable for the small workpiece configuration, but unstable when replaced with the large workpiece.

The experimentally observed limit-cycle occurs at a frequency corresponding closely to the natural frequency of the unstable poles in Figure 4.8(b), which represents a destabilisation of the first vibrational mode. A classical solution for dealing with it, is to notch-filter the current reference at the resonant frequency (Ellis and Lorenz 2000). This has the effect of attenuating control inputs that lead to excitation of the vibrational mode. The resonance frequency can easily be iden-
Figure 4.8: Closed-loop poles of the motor position loop for the discrete model of the servo drive axis loaded with both small and large workpieces. Includes the current servo dynamics.

\[ x_n(h + 1) = A_n x_n(h) + B_n \hat{i}_f(h) \]  
\[ i_f(h) = C_n x_n(h) + D_n \hat{i}_f(h) \]

where \( \hat{i}_f \) is the low-pass filtered current reference, \( i_f \) is the reference additionally filtered by the notch-filter, \( x_n \) is the state vector, and

\[ A_n = \begin{bmatrix} \frac{2Q_nK^2 - 2Q_n\omega_n^2}{Q_nK^2 + \omega_nK + Q_n\omega_n} & -\frac{Q_nK^2 + \omega_nK - Q_n\omega_n^2}{Q_nK^2 + \omega_nK + Q_n\omega_n} \\ 1 & 0 \end{bmatrix}, \quad B_n = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \]

\[ C_n = \begin{bmatrix} \frac{2Q_n\omega_n(-K^3 + (Q_n\omega_n - Q_n)K^2 + \omega_n^2K + (Q_n\omega_n^2 - Q_n\omega_n^3))}{Q_n\omega_n(2K^3 - (Q_n\omega_n - Q_n)K^2 + 2\omega_n^2K + (Q_n\omega_n^2 - Q_n\omega_n^3))^2} \\ \frac{Q_nK^2 + \omega_nK + Q_n\omega_n}{(Q_nK^2 + \omega_nK + Q_n\omega_n)^2} \end{bmatrix}^T, \quad D_n = \frac{Q_nK^2 + Q_n\omega_n^2}{Q_nK^2 + \omega_nK + Q_n\omega_n} \]

where \( Q_n \) is the quality factor, \( \omega_n \) is the center frequency and \( K = \frac{2}{T} \).
Augmenting the low-pass filter model with the notch-filter at the appropriate frequency (450 Hz or \( \omega_n = 2827 \text{ rad/s} \)) and quality factor (\( Q_n = 1 \)) results in the closed-loop poles shown in Figure 4.9. Note that for the large workpiece the system is now stable, but for the small workpiece the addition of the notch-filter actually destabilises the system. Again, both of these outcomes agree with the experimentally observed results.

While not shown here, the stability results presented in this section still apply if the order of the mechanical plant models are reduced as a result of combining the inertia \( J_d \) in (3.11) or (3.18) with the inertia \( J_m \) (\( c_{md} = k_{md} = \infty \)). As a final check, the complex nonlinear model that includes all dynamics down to the level of power electronics switching is implemented in a Simulink model. The simulation produces results that also agree with these outcomes.

Figures 4.10(a) and 4.10(b) show the simulated response to a step reference input for the stable configurations of the axis with both small and large workpieces. Since the small workpiece is not compliantly coupled to the motor, the motor and workpiece inertia positions are identical. For the large workpiece this
PERFORMANCE LIMITATIONS OF EXISTING CONTROL ARCHITECTURE

is not the case; however the maximum difference between the two inertias \( J_1 \) and \( J_3 \) is found to be at most 0.0072 deg, which is indistinguishable given the scale of Figure 4.10(b). Comparing these two figures reveals that, except for the slight oscillation during the initial rise for the large workpiece, the step responses are essentially identical.

Since a key performance criterion for the axis is rapid rejection of plant disturbances, Figures 4.10(c) and 4.10(d) are included to show the simulated response of the system to a 5 N·m step disturbance input applied to the workpiece inertia. The dashed line in Figure 4.10(d) is the position of the large workpiece inertia. Comparing Figures 4.10(c) and 4.10(d) reveals that there is a significant oscillation for the large workpiece configuration. Furthermore, comparison with Figure 4.10(b) indicates that, unlike the case for disturbance inputs, reference inputs excite the vibrational mode only weakly.\(^2\) This is expected, since the notch-filter zeros which cancel poles in the reference-to-output transfer function do not appear in the disturbance-to-output transfer function. In spite of this, the maximum deviations from the set-points and the settling times shown in Figures 4.10(c) and 4.10(d) are approximately equal. This result suggests that the control architecture (with optional notch-filter) can produce adequate performance over a large range of workpiece geometries. However, the controller must be hand tuned by an experienced operator whenever the performance is found to be unsatisfactory.

For the case investigated, the conclusion is drawn that the fast sample-rate elements must be considered for the analytic study to agree with the experimental observations. However, there is a caveat on this conclusion. Since linear analysis techniques have been used, the friction model also had to be linear. Recall that the data collection experiments conducted to identify the plant model were specifically designed to ensure the plant operated in the linear region (see Section 3.5.2). The actual friction is known to be nonlinear about zero velocity and hence

\(^2\)Note that a more realistic disturbance input would be far less abrupt than a step, and in practice the vibrational mode would not be excited to the extent shown in Figure 4.10(d).
it is prudent to investigate what effect, if any, this has on the results presented.
Figure 4.10: Simulated responses of the linear model of the servo drive axis loaded with both small and large workpieces, for both step reference inputs (top) and step disturbance inputs (bottom). The motor position is shown by the solid line, as is the workpiece position for (a)–(c). For (d) the workpiece position is shown by the dashed line. The system model for the responses of the large workpiece (right-hand side) includes a notch-filter.
4.1.4 Effective Viscous Friction Coefficient

As stated in Section 3.6.2, the effective viscous damping coefficient is dependant on the magnitude of the motor velocity. In the analysis of the preceding section, $B_m^* = 0.0264 \text{N.m/(rad/s)}$ was used. To confirm that the analysis is valid with this value, it is necessary to check the effect the viscous damping coefficient has on the linear system stability.

Figure 4.11 shows the root locus for the axis with both small and large workpieces for the default controller tuning, where the parameter varying is $B_m^*$. For the large workpiece, the black squares show the point at which the unstable closed loop poles transition into the stable region. This occurs at $B_m^* = 9.55 \text{N.m/(rad/s)}$. Figure 4.12 is a repeat of Figure 3.26, except that the domain of the sinusoidal amplitude is restricted to a region close to zero. From this figure it can be seen that a value of $B_m^* = 9.55 \text{N.m/(rad/s)}$ corresponds to a sinusoidal velocity profile with an amplitude of 0.097 rad/s. Figure 4.13 shows the root locus for the axis with both small and large workpieces for the default controller tuning, plus a notch-filter located to attenuate control inputs that lead to excitation of the first vibrational mode of the large workpiece configuration. For the small workpiece, the black squares show the point at which the unstable closed loop poles transition into the stable region. This occurs at $B_m^* = 8.25 \text{N.m/(rad/s)}$, and corresponds to a sinusoidal velocity profile with an amplitude of 0.068 rad/s (Figure 4.12). In both cases, as the axis transitions into the unstable operating region, the magnitude of the oscillations will increase, thus decreasing the effective viscous friction and further destabilising the system.

The simulated response of the Simulink representation of the nonlinear model (with large workpiece) can be used to confirm that the system enters a limit-cycle operating state after being perturbed slightly. Figure 4.14(a) shows the disturbance torque, which is applied to the workpiece inertia, $J_w$. The position ref-
Figure 4.11: Root locus for closed position loop with default velocity loop tuning, as the effective viscous friction coefficient is varied for both small and large workpieces.

The reference is set to a constant value, hence the controller’s challenge is to reject the deviation from this set-point caused by the disturbance torque. As the system begins to move in response to the disturbance, a growing (unstable) oscillation is observed in the velocity of the motor (Figure 4.14(b)). The controller’s attempt to compensate eventually causes the motor current to saturate (Figure 4.14(c)). Figure 4.14(a) also shows the friction torque acting on the motor shaft. The instantaneous power dissipation due to friction, $P_f$, can be calculated using

$$P_f = T_f \omega_m$$

The result is shown in Figure 4.14(d). The approach presented in Section 3.6.2, that is (3.23), can be used to calculate the effective viscous friction coefficient for each oscillation cycle of the motor as

$$\int_{t_1}^{t_2} T_f \omega_m \, dt = \int_{t_1}^{t_2} T_f \omega_m \, dt$$

$$\int_{t_1}^{t_2} B_m^* \omega_m^2 \, dt = \int_{t_1}^{t_2} P_f \, dt$$

$$B_m^* = \frac{\int_{t_1}^{t_2} P_f \, dt}{\int_{t_1}^{t_2} \omega_m^2 \, dt}$$

(4.18)
Figure 4.12: Effective viscous friction as a function of the amplitude of the sinusoidal velocity profile close to zero for both small and large workpieces. Indicates minimum sinusoidal amplitude required for axis instability.

where \([t_1, t_2]\) is the domain over one oscillation cycle. The result is shown in Figure 4.14(e). As expected, the coefficient is initially large, and greater than the critical value of 9.55 N.m/(rad/s), but quickly drops away as the system becomes unstable. The value of the coefficient in the fully developed limit-cycle is 0.06 N.m/(rad/s).

Figure 4.15 shows the results from a simulation similar to that of Figure 4.14, except that the duration of the applied disturbance torque is halved. Notice that due to the drastically different responses shown in the two graphs, the scale on the vertical axes of parts (b), (c) and (d) differ significantly. While the axis initially moves in response to the disturbance, it quickly settles back to steady state (zero velocity) without entering a limit-cycle. Figure 4.15(e) shows that the effective viscous friction coefficient never falls below the critical value of 9.55 N.m/(rad/s). As expected, the effective viscous friction coefficient during the initial onset of the disturbance for both scenarios is found to be identical –
PERFORMANCE LIMITATIONS OF EXISTING CONTROL ARCHITECTURE

Figure 4.13: Root locus for closed position loop with default velocity loop tuning and a notch-filter, as the effective viscous friction coefficient is varied for both small and large workpieces.

first two data points in Figures 4.14(e) and 4.15(e). As an aside, it can be seen in Figure 4.15(c) that the motor current does not return to zero. This is due to wind-up of the velocity controller’s integrator. The axis does not accelerate however, because the torque produced by this non-zero current is balanced by the non-zero (static) friction torque shown in Figure 4.15(a). The friction torque also balances the torque produced by the (decaying) oscillations of the large workpiece inertia relative to the stationary motor shaft. Hence, the contribution of the nonlinear friction acts to stabilise the otherwise unstable closed-loop system.

On rare occasions it is possible to put the axis with the large workpiece and no notch-filter under closed-loop position control and for it to remain in a stable state. However, if the workpiece is subsequently tapped, or the motor position reference changed, it will rapidly enter the limit-cycle operating behaviour. Hence during a normal machining operation, it is inevitable that the axis will begin vibrating.

From this analysis it can be concluded that as long as the axis is perturbed enough to reduce the effective viscous friction below the critical values shown
Figure 4.14: Simulated response of the complex nonlinear model to a large disturbance torque applied to the large workpiece mounted to the A-axis.

In Figure 4.12, then it will enter a limit-cycle as shown in Figure 4.14. Given the minimal amount of disturbance required, and the level of external excitation (vibration) that exists in the working environment, it is certainly feasible that the effective viscous friction will fall below the critical value. Thus the value for $B_m^*$ used in Section 4.1.3 is valid, since $B_m^* < 8.25 \text{ N.m/}(\text{rad/s})$ will ensure that all the
Figure 4.15: Simulated response of the complex nonlinear model to a small disturbance torque applied to the large workpiece mounted to the A-axis.

analytic results agree with the experimental observations.

4.1.5 Concluding Remarks

As discussed in Section 1.3, there can be considerably large disturbance torques acting on the axis. It is conceivable that these machining forces may add damping to the system and reduce/eliminate the mechanical vibrations. In practice it is
found that in general this is not the case: the axis still vibrates. This is especially so during the finishing pass of the grinding operation as it is typically done with a very low cut depth. Hence the machining forces are very low, and any damping effect is minimal. If the machine is allowed to complete the machining operation in this state, then in the best case the result is a workpiece with poor surface finish, and in the worst case, geometric accuracy outside acceptable tolerances or even a broken tool or chipped grinding wheel may occur.

Furthermore, the peak current limit for the motor is $13\, \text{A}$, but the long-term current limit is lower, that is, the motor can deal with high currents for only a short time. Sustained high currents, as experienced if the limit cycle shown in Figure 1.8 is allowed to persist, will overheat and damage or melt the motor windings and possibly demagnetise the permanent magnets. Finally, from a customer perspective, a machine tool with an axis that vibrates as soon as it is enabled is unlikely to be acceptable.

### 4.2 Existing Controller Performance

In this section the existing controller’s performance is benchmarked. This will provide a means for measuring the performance improvements relative to other control approaches developed during this work.

#### 4.2.1 Simulated Controller Performance

As described in Chapter 3, the existing control architecture is the cascaded PID. This control architecture is used as a baseline to which other controllers will be compared on the grounds of performance of minimising tracking error; disturbance rejection; robustness to unmodeled and/or changing system dynamics; and simplicity of implementation in terms of computation overhead. Both simulation and experimentation will be used to test the various schemes.

The input signals used in the simulation tests consist of a $4\, \text{deg}$ step reference
Figure 4.16: Simulated response of existing control architecture to a step input in position reference and also load disturbance.

trajectory, and a 19 N·m load disturbance applied at 0.5 s. The configuration of the axis includes models of both small and large workpieces (see Figure 3.4). For the small workpiece, the default PID tuning is used. For the large workpiece, both detuned PID\(^3\) and default PID tuning augmented with an appropriately placed notch-filter are used. Figure 4.16 shows the simulated response for these three scenarios.

From the graphs it can be seen that the settling time for the three scenarios is virtually identical. A slight oscillation is present in all three position step responses; however it is most pronounced in the detuned large workpiece configuration. This is further evident in the velocity graphs, which show the velocity reference and feedback signals. In terms of disturbance rejection, the system with small workpiece and the system with large workpiece with notch-filter perform similarly. However, the performance of the detuned large workpiece configura-

\(^3\)Recall that for the large workpiece, the system is not stable with the default tuning. Hence, the \textit{detuned PID} is the case where the velocity controller gains have been decreased until the point where the system response is stable.
Figure 4.17: Response of existing control architecture to a 1 Hz sinusoidal input in position reference.

...tion was notably worse. Figure 4.17 shows the simulated results for a 1 Hz sinusoidal input in position reference. The responses are essentially identical. Even in the position error graphs it is difficult to see any difference in performance. Thus from a reference tracking point of view the different state of tuning for the large workpiece has a minimal effect, however for disturbance rejection the detuned controller performs poorly.

These results confirm that, while the existing architecture is capable of adequately controlling the axis given large changes to the load dynamics, there is no single state of tuning which delivers adequate performance for all configurations. To quantify this result, so that it may be compared to other approaches, both the settling time and the energy consumption for the large workpiece (with notch-filter) configuration are calculated and recorded. The settling time is defined as the point when the axis comes to rest after completing the step maneuver. Theoretically a linear system will never come to rest in response to a step command, only asymptote towards a final value. However due to either quantisation of the feedback signal or static friction (both nonlinear effects) a stable real system will come to rest at an equilibrium point in finite time.

105
PERFORMANCE LIMITATIONS OF EXISTING CONTROL ARCHITECTURE

Figure 4.18: Settling time for existing control architecture. Large workpiece with notch-filter enabled.

能源消耗定义为在步进操作期间使用的电能，直到达到稳态时间。图4.18显示了此配置的4度步响应。红线表示稳态时间，等于0.3925 s。得到的能源消耗被计算为129.4 J。

使用第3章中开发的模型，可以计算出开环系统在默认调谐下的稳定性边界，作为工作部件的惯性$J_3$、耦合刚度$k_β$和阻尼$c_β$的函数。该边界定义为在给定的刚度和阻尼下，任何工作部件惯性的增加都会导致一个或多个闭环极点退出单位圆（见图4.8的右半部分）。计算得到的边界如图4.19所示。正如预期的那样，刚度和阻尼系数的增加，不允许的工作部件惯性也会随之增加。

4.2.2 Experimental Controller Performance

除了仿真结果外，还收集了实验数据来说明现有控制架构的性能限制。本节中的数据是从生产5轴CNC机床实验中获得的。
EXISTING CONTROLLER PERFORMANCE

![Diagram]

Figure 4.19: Stability boundary for default tuning.

grinding machine shown in Figure 3.1, using both the large and small workpieces (see Figure 3.4). As described in Section 3.5.2, the data acquisition software for this machine allows for 2048 data points, for four variables to be logged simultaneously in a single experiment. The sample period can be selected as any multiply of the base sample period, $50 \mu s$.

Figure 4.20 shows how the system behaves for both large and small workpieces for different states of tuning, where the axis is being commanded to hold a constant position. Notice that the range of the vertical axes are all the same for the different experiments, except for that of the large workpiece with default controller tuning. It is this case which exhibits the instability being investigated in this thesis. Shortly after the axis is enabled, it quickly enters the limit cycle behaviour described in Section 4.1.

Figure 4.21 shows the response of the system to a trapezoidal velocity profile, about the point where the axis is reversing, for cases where the tuning of the closed-loop system is stable. As can be seen, the magnitude of the tracking error is similar in all cases. The main difference in the responses is that of the current
PERFORMANCE LIMITATIONS OF EXISTING CONTROL ARCHITECTURE

Figure 4.20: Experimental results for existing control architecture with a constant position reference.

command, which is considerably smoother in the detuned PID experiment. What the graphs to not show however is how the stiffness of the axis varies between the default tuning and the detuned cases. This is a direct indication of the axis’ ability to effectively reject disturbances.
Figure 4.21: Experimental results for existing control architecture with a trapezoidal velocity profile.
4.3 CONCLUSION

This chapter has demonstrated that, in order to successfully analyse controllers for direct-drive machine tool axes with lightly-damped structural vibrations, it may not be sufficient to simply model the mechanical dynamics and neglect those of the electrical system. In fact, in the case investigated, accurate modelling of the current servo (taking account of multirate sampling and computation delays) was more significant in predicting experimentally observed limit-cycles than was increasing the number of vibrational modes in the mechanical plant model. This addresses the first two research aims of this thesis (as outlined in Section 2.6).

It was also demonstrated that a cascaded PID controller with fixed gains and filters is not ideally suited for controlling a direct-drive axis which is subject to large changes in load dynamics. The limiting factor here is the requirement to re-tune the controller whenever the performance is found to be unacceptable in terms of tracking performance, disturbance rejection, or when the combination of plant dynamics results in an unstable closed-loop system. Further, the tuning of the axis for adequate performance is a challenging manual exercise which requires a degree of expertise and experience, and as such has proven to be difficult to automate.

The content of the following chapters will concentrate on using the low-order linear models developed in Chapter 3 to develop an suitable control approach that results in optimal performance for the given axis configuration. The developed approach should provide an effective solution over a large range of possible axis configurations and the ensuing dynamics. The ultimate goal is to at least maintain, but hopefully improve, the tracking and disturbance rejection performance of the servo drive; while at the same time greatly simplifying the tuning process to the point where an automated approach can be developed.
In this chapter, two existing optimal control techniques are investigated, minimum-time optimal control (Section 5.1) and linear quadratic (LQ) control (Section 5.2). Minimum-time optimal control should provide the best possible performance given the hard constraints on the system (saturation of the actuator). However it is a nonlinear open-loop approach, and as will be discussed shortly, is not suitable for implementation on the actual hardware. The reason for its inclusion here is to provide a benchmark for the system performance. The speed and manner in which the system responds to a step change in the reference signal are standard criteria used to measure and evaluate the system’s performance. Given the input constraint, minimum-time optimal control should provide the quickest possible response.

Minimum-time optimal control is then compared to both the incumbent approach of cascaded PID control and to Linear Quadratic (LQ) control. LQ control is a closed-loop approach which can be implemented on the actual hardware. An assessment will be made as to whether or not this technique gets sufficiently close to the performance of minimum-time optimal control. In addition to the standard step test, LQ control will be assessed in terms of robustness to changing plant dynamics and suitability for implementation on the application under investigation. The evaluation of these criteria will, to some extent, highlight if any of the limitations of cascaded PID control, as identified in Chapter 4, have been mitigated. The formulation of both minimum-time optimal control and LQ control requires the model of the plant to be linearised so that the optimisation can be solved.
5.1 Minimum-Time Optimal Control

In this section minimum-time optimal control is investigated. While it is a useful tool to benchmark the system, minimum-time optimal control is not a suitable approach for implementation on a real system: due to its open-loop nature it performs badly in the presence of both unmodelled dynamics and unknown disturbances. The motivation for analysing this approach is to establish the best possible performance for the system to follow a change in the setpoint command without overshoot. Minimum-time optimal control is an open-loop approach which relies on an exact model of the system dynamics. Given the model, and specific constraints, an optimisation routine is performed to establish which sequence of control inputs will result in the minimum time response. The control input for this approach is the motor current reference, and the constraint is the magnitude of the maximum value. For the RX7 A-axis this is 13 A.

5.1.1 Pontryagin’s Minimum Principle

The fundamental theory behind minimum-time optimal control is Pontryagin’s Minimum Principle. In this work necessary conditions to transfer a system from an initial state to a specific target state in minimum-time are developed. Given the system modelled by

\[ \dot{x}(t) = h(x(t), u(t), t) \]  

(5.1)

and a cost function of the form

\[ J(u) = \int_{t_0}^{t_f} g(x(t), u(t), t) \, dt, \]  

(5.2)

where \( x \) is the state of the system and \( u \) is the control input. The Hamiltonian can then be written as

\[ H(x(t), u(t), p(t), t) \triangleq g(x(t), u(t), t) + p^T(t) \dot{x}(t), \]  

(5.3)

where the costate equations are

\[ \dot{p}(t) = -\frac{\partial H}{\partial x}(x(t), u(t), p(t), t). \]  

(5.4)
Pontryagin’s Minimum Principle is

\[ H(x^*(t), u^*(t), p^*(t), t) \leq H(x^*(t), u(t), p^*(t), t), \]

(5.5)

where the * superscript indicates the optimal state, control input and costate. In other words, the optimal control input, \( u^* \), is the one which minimises the Hamiltonian. Given a constraint on the absolute magnitude of the control input, the optimal input sequence is found to be bang-bang (Kirk 1998). That is, to obtain the minimum-time response, the maximum effort must be maintained throughout the interval of operation. For the minimum-time problem the cost function is

\[ J(u) = \int_{t_0}^{t_f} dt = t_f - t_0. \]

(5.6)

5.1.2 Numerical Optimal Control Solution

If the poles of the \( n \)-order system are all real, and a time-optimal control exists, the control signal can switch at most \((n - 1)\) times (Kirk 1998). However, as was revealed in Section 3.5.2, the nature of the RX7 A-axis dynamics means that some of the poles are complex (see Table 3.3). Hence it is unknown \textit{a priori} how many times the control input must switch. A numerical method for solving the time-optimal control problem for systems with flexible modes was developed by Pao and Franklin (1994) and Pao (1994). Consider the discrete-time model given by

\[ x(k + 1) = \Phi x(k) + \Gamma u(k), \]

(5.7)

with \(|u(k)| \leq u_{\text{max}}\). The state at index \( k \) is

\[ x(k) = \Phi^k x(0) + C U, \]

(5.8)

where

\[ C \triangleq \begin{bmatrix} \Phi^{k-1} \Gamma & \Phi^{k-2} \Gamma & \ldots & \Phi \Gamma & \Gamma \end{bmatrix} \]

(5.9)

and

\[ U \triangleq \begin{bmatrix} u(0) & u(1) & \cdots & u(k - 2) & u(k - 1) \end{bmatrix}^T. \]

(5.10)
Without loss of generality, it is assumed that $x(0)$ is a nonzero initial condition. Formulating the problem as a set of constrained least square problems, the challenge is to find the smallest $k$ such that there exists a $U$ such that the Euclidean norm of $L(U) = \Phi^k x(0) + CU$ is zero. That is, return all the system states to the origin as quickly as possible. In Pao and Franklin (1994), the authors solve the optimisation problem using a linear programming technique. In this implementation, the same approach is used.

The parameters $\Phi$ and $\Gamma$ in (5.8) are derived from the model developed in Section 3.5.2, specifically from (3.11) and the parameter values presented in Table 3.2. Using a sample period equal to that of the existing velocity controller, 250 $\mu$s, it is found that the time-optimal settling time\(^1\) to a 4 deg step (with zero initial velocity) for the large workpiece configuration is 19.500 ms. The response of the system to this control input is shown in Figure 5.1.

The energy consumed in this manoeuver is calculated to be 149.6 J. While the state of the system at the end of the manoeuver is correct (motor/workpiece position equal to 4 deg and zero velocity), the control input (the motor current) is not strictly bang-bang. It is hypothesized that this outcome is caused by the discretisation of the input signal. The true optimal switching times may occur during an update interval.

An obvious solution to this shortcoming is to reduce the sample period. Due to a computation limitation of the PC used for the optimisation, it is found that the smallest sample period which could be solved for is 10 $\mu$s.\(^2\) Using this sample period, the settling time is found to be 19.430 ms, and the energy consumption is 151.5 J. Figure 5.2 shows the result. Again the input signal is not precisely bang-bang, but it does appear that the true time-optimal control input will switch three times. Note that this is a direct result of the chosen initial conditions. In the

---

\(^1\)Settling time is defined in Section 4.2.1.

\(^2\)A sample period of 10 $\mu$s requires approximately 2000 inputs to be optimised, which requires a large quantity of system memory.
Figure 5.1: Time-optimal control result for 250 µs sample period.

middle of the move shown in Figure 5.2 it can be seen that the switching of the control inputs is required to correct the relative phasing of the two inertias, so as to ensure that they will both come to rest at the required end position simultaneously. While not demonstrated, it is conceivable that for other specific initial conditions, the control input may need to switch only once.

Given that the dominant flexible mode of the system is lightly damped (see Table 3.3), the location of the extra switches agrees with the observations presented by Pao (1994). That is, for lightly damped systems, the extra switches occur in the middle of the maneuver, as distinct from at the end for heavily damped systems.

5.1.3 Numerical Optimal Control Solution Using An Improved Efficiency Algorithm

As a solution to the limitation that the sample period cannot be made arbitrarily small without substantially increasing the computation requirements, an algo-
Figure 5.2: Time-optimal control result for 10 µs sample period.

...rithm with improved efficiency over the method presented by Pao and Franklin (1994) is proposed here. Assuming that the true time-optimal control input switches three times (as can be inferred from Figure 5.2), then it is possible to restate the problem as shown in Figure 5.3. Of course this assumes that given the initial conditions, the first control input required for minimum-time optimality is positive. If not the case, then the control inputs must be mirrored through the horizontal axis. Consider the system model from (5.7) with

\[ u(k) = \begin{cases} 
  u_{\text{max}} & 0 \leq k < k_1 \\
  -u_{\text{max}} & k_1 \leq k < k_2 \\
  u_{\text{max}} & k_2 \leq k < k_3 \\
  -u_{\text{max}} & k_3 \leq k < k_4 \\
  0 & k_4 \leq k 
\end{cases} \] (5.11)

where \( k_x \in \mathbb{N}^+ \). Then, given nonzero initial conditions, the optimisation is

\[ \min_k \| x(k_4) \|_2 \] (5.12)
subject to

\[ k_1 - k_2 \leq 0, \quad (5.13a) \]
\[ k_2 - k_3 \leq 0, \quad (5.13b) \]
\[ k_3 - k_4 \leq 0, \quad \text{and} \]
\[ k \geq 0, \quad (5.13d) \]

where

\[ k \triangleq \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}. \quad (5.14) \]

In this problem formulation, there are only four variables to be optimised. Hence the sample time of the discrete-time model used in the search algorithm can be very small. The step size used in this analysis is 10 ns. While this very small step size does result in an extended time to complete each iteration of the search, there are minimal requirements on system memory, which was the constraint with reducing the step size in the algorithm proposed by Pao and Franklin. The approximate switching times identified in Figure 5.2 (step size 10 µs) can be used as initial conditions for k in this approach.
Figure 5.4 shows the result at the end of the optimisation. The switching times are shown in Table 5.1, and the settling time is found to be 19.42477 ms. The energy consumption for this sequence of control inputs is 151.4 J.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>9.18677</td>
</tr>
<tr>
<td>$k_2$</td>
<td>9.56647</td>
</tr>
<tr>
<td>$k_3$</td>
<td>10.10682</td>
</tr>
<tr>
<td>$k_4$</td>
<td>19.42477</td>
</tr>
</tbody>
</table>

Table 5.1: Time optimal control switching times.
Figure 5.5: Simulated benchmarking test for time-optimal control – iterations.

Figure 5.5 shows the evolution of the search algorithm. In Figure 5.2 the resolution of the switching was restricted to 10 µs, here with a resolution of 10 ns, it is seen that the true time-optimal solution lies within 10 µs of the initial value.

Finally, in Table 5.2 a comparison is made between the settling time of the existing cascaded PID controller (from Section 4.2.1) and that of the minimum-time optimal controller. While the time optimal approach does use 17% more energy, it completes the manoeuvre twenty times faster.

The results from this section motivate the use of model-based optimal control for machine tool servo drives. Clearly the possibility of decreasing the cycle time has economic benefits. Of course the ultimate limit to the cycle time of the system
<table>
<thead>
<tr>
<th>Controller</th>
<th>Time [ms]</th>
<th>Energy [J]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascaded PID</td>
<td>392.500</td>
<td>129.4</td>
</tr>
<tr>
<td>Minimum-Time</td>
<td>19.425</td>
<td>151.4</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of minimum-time optimal control with PID control.

is determined by the machining process, but at least in this way the limiting factor does not need to be the performance of the servo control system after taking into account the system constraints. The other benefit of model-based design is the ease by which the controller is tuned. Assuming a suitable system model is available, which is reasonable given modern system identification techniques, tuning is automatic.

As highlighted in the Introduction of this chapter, minimum-time optimal control is not suitable for implementation on a real system. Hence, the following sections of this thesis will concentrate on developing model-based optimal controllers which are suitable for implementation on real CNC machine tool servo drives. This starts with the design and implementation of a linear quadratic controller, and finally, in Chapter 6 with Model Predictive Control (MPC).

5.2 Linear Quadratic Controller

In the previous section, it was shown that significant additional servo performance is possible using minimum-time optimal control as the benchmark. Hence as a potential alternative to cascaded PID control, a Linear Quadratic (LQ) design is proposed. LQ control is a closed-loop model-based approach that uses known information about the plant dynamics (in the form of differential/difference equations) to compensate for output deviations from the reference. The goal of this work is to replace the $250 \mu s$ elements of the cascaded PID controller (position and velocity controllers) with a single LQ controller. This controller design will make use of the plant model developed in Chapter 3.
The structure of Linear Quadratic (LQ) controller considered here is shown in Figure 5.6. The controller comprises four main components: the LQ Tracking Controller, the Kalman Filter, the Integrator, and the Nonlinear Friction Compensator. These components represent standard control approaches that can be found in a number of control textbooks, for example in Dutton et al. (1997) and in Franklin et al. (2006). Each component will briefly be described here; for the full details of the LQ controller design see Appendix B.

The LQ control design requires that the plant model be linear. As highlighted in the Literature Review (see Section 2.2.2), nonlinear friction can be a problem for machine tool axes. Specifically this is the case when it comes to position regulation (hunting), tracking at low velocities (stick-slip) and velocity reversal (axis pause). Since nonlinear friction is known to influence the motion of the A-axis, and its characteristics have been identified and modelled (see Section 3.5.3), it is possible to directly account for its effects in the controller design. This compensation will in essence act to linearise the plant dynamics seen by the LQ controller.
The friction compensation is given by

\[
i_f(k) = \begin{cases} 
K_f \text{sgn } (\dot{\theta}_r(k)) & \dot{\theta}_r(k) \neq 0 \\
K_f \text{sgn } (\theta_r(k) - \theta_m(k)) & \text{otherwise.}
\end{cases}
\] (5.15)

Here \(K_f\) is chosen so that \(i_f\) is equivalent to the motor current required to compensate for the Coulomb friction acting on the axis, that is

\[
K_f = \frac{T_c}{K_t},
\] (5.16)

where \(T_c\) is the Coulomb friction identified in Section 3.5.3 and \(K_t\) is the motor torque constant. It is reasonable to only consider Coulomb friction here, since viscous friction is included in the linear plant model and hence will be dealt with by the LQ controller. Likewise, the small additional torque due to stiction is left for the integrator to manage.

The LQ Tracking Controller is given by

\[
i_{LQT}(k) = N_i \theta_r(k) - K (x(k) - N_x \theta_r(k)).
\] (5.17)

where \(K\) is the feedback gain determined by solving a discrete-time algebraic Riccati equation (DARE), and \(N_i\) and \(N_x\) are feedforward gains required to ensure the controller tracks the reference.

Since the full state is typically not available via direct measurement, it must be estimated. Here a Kalman filter is used to estimate the full state vector from the known control input \(i_{LQGTI}\) and the measured plant output \(\theta_m\). Note that the control input to the Kalman filter does not include the Nonlinear Friction Compensation \(i_f\). This is because the LQ model does not include the nonlinear friction dynamics, and hence the estimator does not need to know about the additional control effort used to compensate for the nonlinear component of the friction torque. When augmented by a Kalman state estimator, the LQ controller is often referred to as a Linear Quadratic Gaussian (LQG) controller. In the LQG Tracking (LQGT) system the current reference is

\[
i_{LQGT}(k) = N_i \theta_r(k) - K (\hat{x}(k|k) - N_x \theta_r(k)),
\] (5.18)
where

\[
\dot{x}(k + 1|k) = (A - LC)\dot{x}(k|k - 1) + (B - LD)i_{LQGT}(k) + L\theta_m(k) \tag{5.19a}
\]

\[
\dot{x}(k|k) = (I - MC)\dot{x}(k|k - 1) - MDi_{LQGT}(k - 1) + M\theta_m(k) \tag{5.19b}
\]

and where \(L\) is the Kalman gain and \(M\) is the innovation gain. Again they are derived by solving a DARE.

The final component of the LQ Design is the Integrator. The control law for the LQGT with Integrator (LQGTI) is

\[
i_{LQGTI}(k) = N_i\theta_r(k) - K(\dot{x}(k|k) - N_x\theta_r(k)) - K_Ix_I(k), \tag{5.20}
\]

where \(x_I\) is the integrator state and \(K_I\) is the integrator gain. See Section B.6 for the method to choose the integrator gain.

In its pure form, LQ control will not explicitly handle constraints. However the system under consideration does have a constraint on the magnitude of the control input. Hence post processing (saturation) of the control input is required to ensure that it does not exceed the capabilities of the plant. Furthermore, during times of output saturation, the integrator must include the provision for anti-windup. See Appendix B.2 for the details. When this happens, the resulting solution is no longer optimal in regards to the optimisation that was originally solved.

5.2.2 Simulation of LQ Control

As the first step to testing the LQ controller designed in Section 5.2.1, the controller is incorporated into the simulation models of the ANCA RX7 A-axis developed in Chapter 3. The 250 \(\mu\)s components in Figure 3.8 (the PID Position and Velocity Controllers) are replaced by the LQGTI with Friction Compensation controller shown in the dashed box of Figure 5.6. The update period of the LQ controller is 250 \(\mu\)s. Note that the Filters block shown in Figure 3.8 is no longer required.
Simulink is used as the simulation environment. The plant includes the non-linear friction model identified in Section 3.5.3. Random white noise signals are applied to encoder feedback and the motor current to simulate the actual noise experienced by these two signals. It is known that there is limited computational capacity on the DSD for which this control approach is targeted, hence for fair comparison with the eventual experimental implementation, only a 4th-order plant model is used for the controller design (plus one integrator state, so a 5th-order model overall). Q and R are chosen so as to minimise the likelihood of saturating the actuator (i.e. the motor current) on the benchmark test case, given the likely required acceleration and disturbance torques which will be encountered. This is a conservative design so as to avoid activating constraints. This is contrary to the case for minimum-time optimal control, where the constraints are always active. The values used in the simulation for both large and small workpieces are

\[
Q = \text{diag}
\begin{bmatrix}
0 & 0 & 5 \times 10^{12} & 0 & 1.5 \times 10^{-3}
\end{bmatrix},
\]

\[
R = 2, \quad Q_e = 1 \quad \text{and} \quad R_e = 10^{-2}.
\]

As in Section 4.2.1, the test signals applied as inputs for the first simulation tests are a constant 4 deg position reference applied at \( t = 0 \) s and a 19 N·m disturbance torque applied to the workpiece at \( t = 0.5 \) s. Figure 5.7 show the results from the simulation. For both small and large workpieces the initial response to the step change in position reference is quick and with no measurable overshoot. Furthermore, the controller is able to effectively reject the load disturbance and return the system to the required state.
Figure 5.7: Simulated step response for LQGTI with nonlinear friction compensation controller for both small (left) and large (right) workpieces.
OPTIMAL CONTROL TECHNIQUES

Figure 5.8 shows the benchmark test that can be directly compared to Figures 4.18 (existing control approach) and 5.4 (minimum-time optimal control approach). Again the test involved a step change in position of 4 deg. The benchmark result for rise-time is \( t = 46.500 \text{ ms} \) and \( E = 69.1 \text{ J} \). In the presented result a perfect plant model is used, with no encoder/current noise, no nonlinear friction, and cost function weights optimised to initially produce the maximum permitted motor current of 13 A. The weights are

\[
Q = \text{diag} \begin{bmatrix} 0 & 0 & 2.8 \times 10^{12} & 0 & 3.0 \times 10^{-3} \end{bmatrix},
\]

\[R = 46.42, \ Q_e = 1 \text{ and } R_e = 10^{-2}.\]

![Figure 5.8: Simulated benchmarking test for LQGTI controller on the RX7 A-axis. Rise-time is \( t = 46.500 \text{ ms} \) and energy usage is \( E = 69.1 \text{ J} \).](image)

The rise-time and energy used in the LQ controller, are tabulated in Table 5.3 alongside those for cascaded PID and minimum-time optimal control. The energy used by the LQ controller is much less than both cascaded PID and Optimal, and rise-time is of the same order of magnitude as time-optimal, and thus considerably faster than cascaded PID.
<table>
<thead>
<tr>
<th>Controller</th>
<th>Time [ms]</th>
<th>Energy [J]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascaded PID</td>
<td>392.500</td>
<td>129.4</td>
</tr>
<tr>
<td>Minimum-time</td>
<td>19.425</td>
<td>151.4</td>
</tr>
<tr>
<td>LQ</td>
<td>46.500</td>
<td>69.1</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of LQ control with cascaded PID and minimum-time optimal control.

Additional simulation results are presented next to further illustrate the performance of the LQ controller. Figure 5.9 shows the response of the controller to a 1 Hz sinusoidal position reference with an amplitude of 90 deg for both small and large workpieces. The tuning of the controller amounts to replacing the plant model with the appropriate one for each different workpiece. Sufficient performance can be achieved by using a single set of cost function weights (assuming the order of the plant model is maintained). Comparing these results with those for the cascaded PID controller presented in Figure 4.17 illustrates the significant improvement achieved by switching to a model-based control approach.

A feature common to most linear controllers is their inherently poor performance in handling the effects of nonlinear friction. The LQGTI controller with no nonlinear friction compensation is no different. However, with the inclusion of the nonlinear friction compensation algorithm given in (5.15), the controller’s performance shows marked improvements. Figures 5.10 and 5.11 highlight the controller’s ability to avoid the hunting phenomenon typically observed in machine tool axes. On the left-hand side of Figure 5.10, it can be seen that due to the nonzero position error, the motor current continuously increases (due to the integrator accumulating) until there is sufficient motor torque to overcome static friction. However, due to the Stribeck friction effect (see Section 3.5.3), as soon as the axis begins to move the friction quickly decreases resulting in too much motor torque and the axis overshooting the set-point. The integrator then begins to accumulate in the opposite direction and the process repeats itself. The inclusion of the friction compensation algorithm on the right-hand side of Figure
OPTIMAL CONTROL TECHNIQUES

5.10, where a type of bang-bang control is used, essentially eliminates the hunting phenomenon. Figure 5.11 shows how the friction compensation component of the overall control effort behaves when there is a nonzero position error.

Finally, Figure 5.12 shows the friction compensation algorithm’s ability to reduce the quadrant glitches, which are typically encountered as an axis reverses direction during a contouring operation. For this test a very small amplitude (and hence low velocity) 1 Hz sinusoidal maneuver is used. On the left-hand side of the figure the quadrant glitches are seen to be very large, and in fact dominate the tracking position error. In practice, this large tracking error would result in a geometrical inaccuracy in the finished workpiece. Comparing this to the graph on the right-hand side of the figure shows how the inclusion of the friction compensation algorithm can drastically reduce the effect of nonlinear friction. Looking closely at the motor current in both examples it can be seen that the current changes direction much quicker when friction compensation is used. This results in both a superior tracking performance and a lower peak motor current.
Figure 5.9: The response of the LQGTI with nonlinear friction compensation controller to a sinusoidal position reference for both small (left) and large (right) workpieces.

Figure 5.10: The hunting phenomenon which occurs as a result of nonlinear friction (left), compared to the response with the inclusion of nonlinear friction compensation (right) for the large workpiece.
Figure 5.11: The LQGTI with nonlinear friction compensation controller works to eliminate position error in the presence of nonlinear friction by using bang-bang style control.

Figure 5.12: The response of the LQGTI controller to a sinusoidal position reference without friction compensation (left) and with friction compensation (right) for the large workpiece. Given the small amplitude of the sinusoid the quadrant glitches are quite pronounced for the case without nonlinear friction compensation.
5.2.3 Implementation of LQ Control on Test Rig

Having successfully developed and tested the LQ controller in simulation the next step is to implement it on the industrial control system hardware used in the RX7 and test it experimentally. Specifically the LQ controller must be implemented on the Digital Servo Drive (Figure 1.7) in such a way as to seamlessly replace the $250 \mu s$ elements of the existing servo drive control algorithm (see Figure 3.8), those elements being the PID position and velocity loops. The existing current loop as well as other low level software functionality is left intact. For safety and to avoid damaging the RX7, initial testing is carried out on The Fridge test rig and small servo motor (see Figures 3.5 and 3.6).

The existing cascaded PID control system is implemented in a combination of C and assembler source code. Whilst it is certainly possible to code the LQ controller in a similar fashion, advanced tools exist which allow Simulink models, such as those developed in the previous section, to be converted directly into C–code which is suitable for compilation directly into an executable that will run on the DSD’s DSP. The product used to convert a Simulink model into C–code is Simulink Coder (formally Real-Time Workshop).

The Simulink model shown in Figure 5.13 is used to generate the C–code which is combined with the existing source code to produce executable drive code which includes the LQ controller that was developed in the previous section. One of the important requirements which must be considered when implementing the LQ controller is to include all of the safety mechanisms that are present in the existing controller. Using the Simulink modelling environment this is a straightforward task.

As was discovered in the simulation testing it is not possible to develop a single controller that is robust enough to handle large changes in plant dynamics; that is, an LQ controller with a fixed plant model and controller gains. The solution is then to implement the controller in such a way as to allow an automatic
calibration routine to tune the controller for different workpieces. To facilitate this both the system identification and controller design routines were implemented into a calibration software application, which allows both the plant model for the Kalman estimator and the controller gains to be updated on the servo drive automatically. This calibration application can be run whenever it is found that the controller’s performance is unacceptable as a result of changes to the plant dynamics. If the the machine operator is unable to determine if the performance, or stability, have been compromised by changing the workpiece geometry, it may be appropriate to execute the calibration routine at the beginning of each machining cycle.

Since it is known that the available computation time is limited, the plant model used in the implementation of the LQGTH controller is restricted to a 4th-order model. The model will principally capture the mechanical plant dynamics. The model used is the same as that given in (3.11), except that since on the real
RX7 $J_d$ was found to be insignificant (see Table 3.2) it is ignored, along with $c_{md}$ and $k_{md}$. This model will be used for all workpiece configurations. For the small workpiece (and the small servo motor test rig), the identified inertia of the workpiece, $J_w$, is very small and the coupling ($c_{mw}$ and $k_{mw}$) is very stiff, which results in the characteristic dynamics of a single inertia plant.

It is interesting to note that the model used in the LQ controller design does not include the fast time scales associated with the electrical dynamics or current controller. Although these were found in Chapter 4 to be critical in reproducing observed limit-cycle behaviour in simulation, there is clearly sufficient robustness in the controller designed around the dominant mechanical dynamics. Consequently, it appears that system identification for model-based controller design should focus on these aspects in practice.

Using the system identification techniques that are detailed in Appendix A.4 and utilised in Section 3.5.2, the parameters for the plant model for the small servo motor are given in Table 5.4. The weights used in the controller design are

$$Q = \text{diag} \begin{bmatrix} 0 & 0 & 5.0 \times 10^{12} & 0 & 1.5 \times 10^{-3} \end{bmatrix},$$

$$R = 5.0, \quad Q_e = 1 \quad \text{and} \quad R_e = 10^{-2}.$$  

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Small Servo Motor</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_t$</td>
<td>0.51</td>
<td>N·m/A</td>
</tr>
<tr>
<td>$J_m$</td>
<td>$4.7 \times 10^{-5}$</td>
<td>kg·m$^2$</td>
</tr>
<tr>
<td>$J_w$</td>
<td>$1.0 \times 10^{-6}$</td>
<td>kg·m$^2$</td>
</tr>
<tr>
<td>$B_m$</td>
<td>0.0264</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$c_{mw}$</td>
<td>0.6309</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$k_{mw}$</td>
<td>78 603</td>
<td>N·m/rad</td>
</tr>
</tbody>
</table>

Table 5.4: Small servo motor plant parameters.

Figure 5.14 shows the results from implementing the LQ controller on The Fridge test rig with the small servo motor. As can be seen in the graph, the controller is quite effective at tracking the trapezoidal velocity profile. The position error is maintained within $\pm1\,\text{deg}$ throughout the profile.
Figure 5.14: LQ Controller implementation on the Fridge test rig.
5.2.4 Implementation of LQ Control on RX7

After successful implementation of the LQ algorithm on The Fridge test rig, the final step is to test the implementation on an actual RX7. The ultimate test for the LQ controller is to ascertain its performance at controlling the axis with different workpieces that have drastically different geometries. The workpieces used for the test are those shown in Figure 3.4. For each workpiece, the calibration application described in the previous section is used to determine suitable plant models and controller gains. The plant model parameters for both the small and large workpieces are displayed in Table 5.5. The weights used in the controller design are the same as those used for the experiments on The Fridge test rig.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Small Workpiece</th>
<th>Large Workpiece</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_t$</td>
<td>3.3567</td>
<td>3.3947</td>
<td>N·m/A</td>
</tr>
<tr>
<td>$J_m$</td>
<td>0.0147</td>
<td>0.0142</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_w$</td>
<td>0.0519</td>
<td>0.0002</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$B_m$</td>
<td>0.0224</td>
<td>0.0257</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$c_{mw}$</td>
<td>0.6071</td>
<td>0.5750</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$k_{mw}$</td>
<td>87.791</td>
<td>80.391</td>
<td>N·m/rad</td>
</tr>
</tbody>
</table>

Table 5.5: RX7 A-axis mechanical plant parameters.

Figure 5.15 shows the system performance with the inclusion of nonlinear friction compensation for both small and large workpieces. The position reference is a constant in both experiments. As can be seen in the figure, in both axis configurations there is a persistent oscillation in the position feedback. Further investigation showed that the oscillation is due to the friction compensation algorithm. Since there is enough noise in the system (both electrical and mechanical) to constantly perturb the axis away from the set-point the friction compensation controller is constantly injecting control effort in alternating directions in an attempt to remove the position error. Decreasing the gain on the friction compensation controller does improve the performance, but to remove the oscillation completely the gain must be reduced to almost zero. So while in simulation, and on “The Fridge” test rig, the inclusion of nonlinear friction compensation was
OPTIMAL CONTROL TECHNIQUES

shown to provide a performance benefit, when implemented on the actual RX7 machine tool this proved not to be the case. It appeared that in spite of the friction identification work, there was sufficient dither in the operating environment to render the friction compensation algorithm unnecessary to the point where its affect was actually detrimental to the controller’s performance. As a result of this, the decision was made to disable the friction compensation algorithm for all future tests.

Figure 5.16 shows the experimental results from the same constant position reference test with the friction compensation algorithm disabled. Note that for each workpiece the correct plant model and controller gains are used. While there is still some oscillation (and possibly noise) on the position feedback, the controller operation is considered to be stable. Comparing this to the results from the existing control approach that are shown in Figure 4.20, it can be seen that the response is very similar. The axis is also found to be very stiff: that is, the controller effectively rejects external disturbances. Finally, as expected, if the small workpiece plant model and controller gains are used to control the axis with the large workpiece (and vise versa) then similar instabilities as noted in Chapter 4 are observed.

Figure 5.17 shows how the controller performs while tracking a contouring trajectory for both small and large workpieces. Note that while the reference trajectories for both axis configurations are the same, the data snapshots are taken at different points. Again comparing this to the results from the existing controller (Figure 4.21), it is seen that the LQ’s tracking performance is an order of magnitude superior. With the cascaded PID controller the maximum tracking error is approximately 30 deg, compared to only 0.5 deg for the LQ controller. Further note that the responses for both workpieces are essentially identical, thus validating the decision to use the same controller weights. The main difference between the two responses can be seen in the motor current. The large workpiece requires
more current to execute the same maneuver. This is expected since a larger total inertia requires more torque for a given acceleration.

While not a new approach, LQ design does provide an effective control methodology for machine tool servo drives. From the experimental results, it is evident that the performance of the LQ controller developed in this chapter is superior to that of the existing cascaded PID controller. The addition of trajectory tracking, state vector estimation, integral action and nonlinear friction compensation was reasonably straightforward to implement, and provided the necessary control characteristics to achieve the required performance. However, the main shortcoming of the LQ design method is the requirement to include ad-hoc elements to deal with system constraints: specifically, management of the integrator wind-up when the control output is saturated. Every real actuator has a limit to the achievable output; hence this problem affects any LQ control system which includes integral action. The result is that the output applied to the plant is no longer equal to the that determined from the optimisation of the objective func-
Figure 5.16: The experimental response of the LQGTI controller to a constant position reference without friction compensation for the small workpiece (left) and the large workpiece (right).

One approach, to overcome this limitation is to choose the objective function weights so that system constraints are never activated during normal operation, but clearly this would have a negative affect on the achievable performance.
Figure 5.17: The experimental response of the LQGTI controller to a trapezoidal velocity profile without friction compensation for the small workpiece (left) and the large workpiece (right).
5.3 Conclusion

It is clear that model-based controller designs such as the ones presented in this chapter have an advantage for control of machine tool DSDs in comparison to the incumbent approach of cascaded PID. The cost of this advantage is the requirement to initially identify and model the dynamics of the mechanical plant. The accuracy with which this is achieved will typically influence the servo tracking performance of the resulting controller. However, as was shown in Section 5.2.4, it is possible to create the required software infrastructure necessary to automate the system identification step. This routine takes the form of system identification using data collected from experimentation, followed by standard LQ approaches to select controller gains.

Contributions of this chapter include the observation that models used for system analysis can be successfully adapted for utilisation in model-based controller design. While in the system analysis sections it was found that accurate modelling of the fast system dynamics was critical; for model-based controller design this requirement was able to be relaxed. However, stability margins are likely to be less than predicted given that the model is not complete.

Another result from the research in this chapter was that a single set of cost function weights can be selected so that adequate performance is achieved regardless of the actual plant model parameters. Hence identifying an appropriate model is the key step to tune the controller. Furthermore, for weights which are not overly aggressive in terms of tracking precision, then a stable and high performance servo system is guaranteed. The final step in the research contributing to this thesis is to remove the requirement to include ad-hoc intervention to avoid exceeding system constraints.
CHAPTER 6

MODEL PREDICTIVE CONTROL

The major contributions of Chapter 6 have been published in Stephens et al. (2011) and accepted for publication in the IEEE Transactions on Industrial Electronics in a Special Section on Digital Control Systems in Power Electronics and Electrical Drives.

The benefits of Model Predictive Control (MPC) as a control technique have been well established. However, its application to reference tracking on Digital Servo Drives (DSDs) which typically have very fast update rates is limited by the computational power of present-day processors. The novel MPC formulation presented in this chapter provides the Control Engineer a mechanism to trade-off online computation effort with tracking performance, while still maintaining stability. It is shown that increasing what will be called a trajectory horizon inherently leads to improved tracking, but that larger horizon lengths have the unwanted effect of increasing online computation. The proposed MPC formulation is compatible with recently developed explicit MPC solutions, and hence the burden of online optimisation is avoided. The proposed approach (formulated as an explicit MPC solution) is successfully implemented on an industrial machine tool DSD, and is shown to outperform the incumbent approach of cascaded PID control.

6.1 INTRODUCTION

While generally effective and simple to implement, the PID controller does have several limitations, especially as the performance requirements become more exacting. By its nature, it is reactive and thus requires a degree of positioning error
MODEL PREDICTIVE CONTROL

to enact control effort. Secondly, in practice the process of tuning them often requires a high level of skill and experience. Model Predictive Control (MPC) on the other hand is proactive (Maciejowski 2002), in that it predicts what control effort, within the constraints of the system, is required to ensure accurate tracking of the reference trajectory. MPC retains a feedback element to account for model uncertainty. Furthermore, an integrator can be embedded into the formulation to account for unknown disturbances (Wang 2009). Moreover, given that a suitable dynamic model of the plant is available, tuning the controller is essentially a trade-off between tracking accuracy and control effort. Given the rich literature on system identification (Ljung 1999), developing a suitable plant model is generally a straightforward task (see Chapter 3 for details). MPC is strongly related to the LQ controller utilised in Section 5.2, in so much as that when there are no active constraints, the MPC controller is essentially the LQ controller, albeit with a cost function horizon that is finite instead of infinite.

The main limitation of MPC is that significant computation is required to perform the optimisation for all but the most trivial of applications. The ability to use MPC in applications with fast dynamics and limited computational resources is a topic which has received significant attention recently (Lozoya et al. 2008; Bautista-Quintero and Pont 2008). The focus has taken several paths, some of which include: tailoring the optimisation technique to exploit the specific structure of the MPC problem (Boyd 2004; Yildirim and Wright 2002); terminating the optimisation early to ensure a result is obtained within the available calculation window (Pannocchia et al. 2007; Wang and Boyd 2010); or using multiplex optimisation to distribute the computational load (Ling et al. 2005, 2001). While these methods have the potential to reduce the online computation requirements, none are particularly suited to trajectory tracking MPC for servo drive applications. However, an alternative approach is to modify the structure of the original MPC problem to reduce the degrees of freedom (Halldorsson et al. 2005; Cagien-
The method proposed in this thesis falls into this last category.

As it will be utilised in the implementation stages of this work, before proceeding it is necessary to introduce Explicit MPC (EMPC) developed by Bemporad et al. (2002). In this approach, an algorithm is developed to produce an equivalent closed-form (explicit) solution to the original MPC problem offline. The result is a gain-scheduling algorithm which is a function of a parameter vector that contains the current state of the system and the desired future output. The solution is shown to be piecewise affine in the parameter vector, where different controllers are defined for discrete polyhedral regions $X_i$ within the parameter space. The online implementation then simplifies to a sequential search through the regions to locate the one to which the current parameter vector belongs. The controller associated with the identified region is then used to generate the input for the plant. The algorithm to construct the explicit solution is further refined by Tøndel et al. (2003a).

While this approach has the potential to reduce the online computation requirements, it does result in an increase in the memory used to store the program in the controller hardware. Moreover, as the MPC problem becomes more complex (higher order plant model, more input/output variables, more constraints, longer horizons, etc.) the number of regions in the explicit solution grows. The computational advantage over online optimisation is then somewhat diminished since it can take a long time to locate the appropriate region within the parameter space. In the worst case every region must be explored. In another paper by Tøndel et al. (2003b) an algorithm which takes the explicit solution and organises it into a binary search tree is presented. Not only does this drastically reduce the online search time, but it also results in a reduction in the storage requirements.

While the problem formulation by Bemporad et al. (2002) is adaptable to trajectory tracking problems, typically only set-point tracking is implemented. In a paper by Cychowski et al. (2009), the authors successfully apply EMPC to a
drive system with an update period of 500 µs. However only set-point tracking is considered and the prediction and control horizons are very limited. While commercial tools such as the Hybrid Toolbox for MATLAB (Bemporad 2010) exist, they only allow for set-point tracking. Finally, in a paper by Ferreau et al. (2008) it is claimed that even with EMPC formulated as a binary search tree, trajectory tracking is impossible from a computation standpoint.

This chapter explores the application of MPC to trajectory tracking on the industrial machine tool DSD presented early in this thesis, where update periods are in the order of 100 µs. The chapter is organised as follows. Section 6.2 sets up the problem for trajectory tracking in a form suitable for EMPC. The general approach taken here is to modify the MPC problem to reduce the degrees of freedom. Appendix C.4 presents different methods for solving the problem. Section 6.3 presents simulation and experimental results from implementation on an industrial machine tool. Finally, in Section 6.4 some concluding remarks are made.

6.2 MPC DESIGN FOR TRAJECTORY TRACKING

Consider a plant with $n$ states, $m$ controlled outputs and $l$ inputs, which is modelled as the following LTI discrete-time system

$$x_p(k + 1) = A_p x_p(k) + B_p u(k)$$  \hspace{1cm} (6.1a)

$$y(k) = C_p x_p(k)$$  \hspace{1cm} (6.1b)

where $x_p \in \mathbb{R}^{n \times 1}$ is the state vector, $u \in \mathbb{R}^{l \times 1}$ is the input vector, $y \in \mathbb{R}^{m \times 1}$ is the output vector, and $A_p$, $B_p$, and $C_p$ are constant matrices of appropriate dimensions.

To ensure offset-free tracking in the face of model uncertainties and unknown constant disturbances, integral action in the controller is desirable. A method for incorporating an integrator within the MPC framework is to modify the plant model so that the input is the control update $\Delta u(k)$, rather than control $u(k)$ itself.
MPC DESIGN FOR TRAJECTORY TRACKING

(Prett and Garcia 1988; Wang 2009). This is achieved by taking the difference of both sides of (6.1a) to form

\[ \Delta x_p(k + 1) = A_p \Delta x_p(k) + B_p \Delta u(k), \]  

(6.2)

where \( \Delta x_p(k) = x_p(k) - x_p(k - 1) \) and \( \Delta u(k) = u(k) - u(k - 1) \). Next \( \Delta x_p(k) \) must be connected to the output \( y(k) \). Take the difference of both sides of (6.1b) to form

\[ y(k + 1) - y(k) = C_p \left( x_p(k + 1) - x_p(k) \right) \]

\[ = C_p \Delta x_p(k + 1) \]

\[ = C_p A_p \Delta x_p(k) + C_p B_p \Delta u(k) \]

(6.3)

Finally, defining a new state vector as

\[ x(k) = \begin{bmatrix} \Delta x_p(k) \\ y(k) \end{bmatrix} \]

(6.4)

allows (6.2) and (6.3) to be combined to form

\[ x(k + 1) = Ax(k) + B \Delta u(k) \]  

(6.5a)

\[ y(k) = Cx(k) \]  

(6.5b)

where

\[ A = \begin{bmatrix} A_p & 0^T_m \\ C_p A_p & I_p \end{bmatrix}, \quad B = \begin{bmatrix} B_p \\ C_p B_p \end{bmatrix}, \quad C = [0_m \ I_p], \]

and \( 0_m \) is an \( m \times n \) zero matrix and \( I_p \) is the identity matrix of size \( m \).

As was the case in Section 5.2, the full state vector, \( x_p \), is not available via direct measurement. Hence it is necessary to incorporate a state estimator in the solution. The estimator used in this solution is identical to the Kalman state estimator that was used as part of the LQ Controller design. Thus the plant state vector is estimated using

\[ \hat{x}_p(k + 1|k) = (A_p - L C_p) \hat{x}_p(k|k - 1) + B_p u(k) + Ly(k) \]  

(6.6a)

\[ \hat{x}_p(k) = (I - M C_p) \hat{x}_p(k|k - 1) + My(k) \]  

(6.6b)
The augmented state vector from (6.4) is reconstructed as

\[
\hat{x}(k) = \begin{bmatrix} \Delta \hat{x}_p(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} \hat{x}_p(k) - \hat{x}_p(k - 1) \\ y(k) \end{bmatrix}
\]

(6.7)

where \( \hat{x}_p(k) \) and \( \hat{x}_p(k - 1) \) are respectively, the estimated current and estimated previous plant states.

For a trajectory tracking system, the optimisation objective used in the MPC formulation typically includes a penalty on both predicted tracking error and predicted changes to the control input (Maciejowski 2002; Bemporad et al. 2002). It usually takes the form

\[
\Delta U^*(k) = \arg \min_{\Delta U(k)} \left\{ \sum_{i=1}^{H_p} [\hat{y}(i|k) - r(i|k)]^T Q(i) [\hat{y}(i|k) - r(i|k)] + \sum_{i=0}^{H_u-1} \Delta \hat{u}^T(i|k) R(i) \Delta \hat{u}(i|k) \right\}
\]

(6.8)

where \( \hat{y} \in \mathbb{R}^{m \times 1} \) is the predicted output, \( r \in \mathbb{R}^{m \times 1} \) is the reference trajectory, \( \Delta \hat{u} \in \mathbb{R}^{l \times 1} \) is the future change in the control input (\( \Delta \hat{u}(i|k) = 0 \) for \( i \geq H_u \)), and \( \Delta U(k) \triangleq [\Delta \hat{u}(0|k), \ldots, \Delta \hat{u}(H_u - 1|k)]^T \). Furthermore, \( H_p \) is the length of the prediction horizon, and \( H_u \leq H_p \) is the length of the control horizon. In the interest of a simplified notation, the use of \((i|k)\) in the equation should be interpreted as \((k + i|k)\).

The proposed optimisation objective considered here is

\[
\Delta U^*(k) = \arg \min_{\Delta U(k)} \left\{ \sum_{i=1}^{H_t-2} [\hat{y}(i|k) - r(i|k)]^T Q(i) [\hat{y}(i|k) - r(i|k)] + \sum_{i=H_t-1}^{H_p} [\hat{y}(i|k) - \hat{r}(i|k)]^T Q(i) [\hat{y}(i|k) - \hat{r}(i|k)] + \sum_{i=0}^{H_u-1} \Delta \hat{u}^T(i|k) R(i) \Delta \hat{u}(i|k) \right\}
\]

(6.9)

where \( \hat{r} \in \mathbb{R}^{m \times 1} \) is an extended reference trajectory (ERT) – a full definition of which is given shortly – and \( H_t \leq H_p \) is the length of a trajectory horizon. The introduction of \( H_t \) in the MPC formulation is believed to be a novel approach. Figure 6.1 shows the relations between the different horizons.
Figure 6.1: Comparison of MPC horizons, where $H_t = H_u = 6$ and $H_p = 10$. The extended reference trajectory is modelled as a first-order hold (FOH).

When evaluating (6.9) the contribution of the predicted tracking error is weighted by the nonnegative diagonal matrix $Q \in \mathbb{R}^{m \times m}$, while the magnitude of changes to the control inputs are weighted by a strictly positive diagonal matrix $R \in \mathbb{R}^{l \times l}$. These weights need not be constant at each step in the horizon, hence the inclusion of the index $i$.

The predicted output for the augmented plant model of (6.5) is given by

$$
\begin{bmatrix}
\hat{y}(1|k) \\
\vdots \\
\hat{y}(H_p|k)
\end{bmatrix} = \Psi \hat{x}(k) + \Theta \Delta U(k) 
$$

(6.10)
where

\[
\Psi = \begin{bmatrix}
CA \\
\vdots \\
CA^{H_p}
\end{bmatrix}, \quad \text{and} \quad (6.11a)
\]

\[
\Theta = \begin{bmatrix}
CB & 0 & \cdots & 0 \\
CAB & CB & 0 & \cdots \\
\vdots & \ddots & \vdots \\
CA^{H_u-1}B & CA^{H_u-2}B & \cdots & CB & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}. \quad (6.11b)
\]

To ensure stability, the prediction horizon \( H_p \) typically needs to be large (Mayne et al. 2000; Maciejowski 2002). The requirement is normally for the plant to be stable and \( H_p \) to be large enough to include the transients of the dominant plant dynamics – the plant settling time. Fortunately, the class of system considered in this work, machine tool servo drives, are nearly always open-loop stable, and extending \( H_p \) has only a minimal effect on the complexity of solving the MPC problem. On the other hand, extending \( H_t \) or \( H_u \) will increase the complexity significantly. The reason for this will become apparent shortly. For now it is enough to state that by adjusting the length of \( H_t \), improved tracking precision (large \( H_t \)) can be traded off against reduced online computation effort (small \( H_t \)). Furthermore, the lengths of both \( H_t \) and \( H_u \) have no influence on the closed-loop stability of the system. The inclusion of the trajectory horizon in the MPC formulation is especially suited to trajectory tracking where available computation time is limited; such is the case for controllers with fast update rates.

For modern motion control systems, the reference trajectory \( r \) is typically a set of discrete position commands and the input \( u \) is the motor torque, or equivalently, for a permanent magnet synchronous motor, the \( q \)-axis (torque producing) motor current (Krause et al. 2002). The condition \( \Delta \hat{u}(i|k) = 0 \) for \( i \geq H_u \) implies constant torque over the same interval. Given the existence of mechanical friction, a non-zero constant torque will result in the prediction of constant velocity.
at steady state. Under this reasoning, for the class of ERTs defined below, it makes sense for $H_u = H_t$. In the general case this may not be true, hence the decision in the MPC formulation proposed here to introduce the trajectory horizon and distinguish it from the control horizon.

Clearly the ERT can be defined in several ways. The simplest is $\hat{r}(i|k) = r(H_t|k)$, $\forall i \in [H_t + 1, H_p]$, effectively a zero-order hold (ZOH). With this choice, the predicted input $\hat{u}$ at the end of the control horizon will be required to 

reverse 

in order to drive the axis velocity to zero. Since machine tools typically follow smooth contours, a better choice, which is proposed in this thesis, is a first-order hold (FOH) as shown in Figure 6.1. In this way, at the end of the control horizon the predicted input $\hat{u}$ will approach a level that produces a velocity close to that of the reference trajectory at $k + H_t$. To implement a FOH-ERT, there is the additional requirement that $H_t \geq 2$. The FOH-ERT is given by

$$
\hat{r}(i|k) = r(H_t - 1|k) + (i - H_t + 1) \left( r(H_t|k) - r(H_t - 1|k) \right),
$$

$\forall i \in [H_t - 1, H_p]$, or in matrix form

$$
\begin{bmatrix}
\hat{r}(H_t - 1|k) \\
\vdots \\
\hat{r}(H_p|k)
\end{bmatrix}^T = 
\begin{bmatrix}
r(H_t - 1|k) \\
r(H_t|k)
\end{bmatrix}^T \Lambda_f,
$$

where

$$
\Lambda_f = \begin{bmatrix}
I_p & 0_p & -I_p & \cdots & (H_t - H_p)I_p \\
0_p & I_p & 2I_p & \cdots & (H_p - H_t + 1)I_p
\end{bmatrix},
$$

and where $I_p$ and $0_p$ are $m \times m$ identity and zero matrices respectively.

With respect to

$$
\mathcal{R} \triangleq \text{diag}(R(0), \ldots, R(H_u - 1)),
$$

$$
\mathcal{Q} \triangleq \text{diag}(Q(1), \ldots, Q(H_p)),
$$

$$
\mathcal{Q}_t \triangleq \text{diag}(Q(1), \ldots, Q(H_t - 2)),
$$

and

$$
\mathcal{Q}_p \triangleq \text{diag}(Q(H_t - 1), \ldots, Q(H_p)),
$$
MODEL PREDICTIVE CONTROL

equation (6.9) may be rewritten as

$$\Delta U^*(k) = \arg \min_{\Delta U(k)} \left\{ \frac{1}{2} \Delta U^T(k) \mathcal{H} \Delta U(k) + \theta(k) \mathcal{F} \Delta U(k) \right\}. \quad (6.12)$$

Here

$$\mathcal{H} = 2\mathcal{R} + 2\Theta^T Q \Theta, \quad \mathcal{F} = \begin{bmatrix} 2\Psi^T Q \Theta \\ 0_u \\ -2Q_t \Theta_t \\ -2\Lambda_f Q_p \Theta_p \end{bmatrix}, \quad \theta(k) = \begin{bmatrix} \hat{x}(k) \\ u(k-1) \\ r(1|k) \\ \vdots \\ r(H_t|k) \end{bmatrix}^T,$$

where $\Theta_t$ and $\Theta_p$ are respectively the first $m(H_t-2)$ rows and the last $m(H_p-H_t+2)$ rows of $\Theta$, and $0_u$ is a zero matrix of size $l \times l u$. See Appendix C.1 for the complete derivation of (6.12).

In addition, it is possible to define constraints on the input $\hat{u}$, the input change $\Delta \hat{u}$, and the output $\hat{y}$ as

$$\hat{u}_{\min}(i) \leq \hat{u}(i|k) \leq \hat{u}_{\max}(i) \quad \forall i \in [0, H_u - 1], \quad (6.13)$$

$$\Delta \hat{u}_{\min}(i) \leq \Delta \hat{u}(i|k) \leq \Delta \hat{u}_{\max}(i) \quad \forall i \in [0, H_u - 1], \quad \text{and} \quad (6.14)$$

$$\hat{y}_{\min}(i) \leq \hat{y}(i|k) \leq \hat{y}_{\max}(i) \quad \forall i \in [1, H_p]. \quad (6.15)$$

These constraints can be expressed in terms of $\Delta U$ and combined (see Maciejowski 2002) to form

$$\mathcal{G} \Delta U(k) \leq X \theta^T(k) + \mathcal{W}. \quad (6.16)$$

Equation (6.12) in conjunction with (6.16) is in the standard form of a quadratic optimisation problem, where $\theta(k)$ is a parameter vector and is known a priori. At each sample instant $k$, the optimisation is solved and the input $u(k) = u(k-1) + \Delta \hat{u}(0|k)$ is applied to the plant. At the next sample instant the optimisation is repeated using the updated parameter vector and a shifted horizon.

The complexity of the optimisation problem is in part a function of the length of $\theta(k)$. As the number of elements in $\theta(k)$ increases, so too does the effort required to complete the optimisation. This is why the inclusion of the trajectory

150
horizon $H_t$ into the formulation benefits practical implementations of trajectory tracking MPC. If classical trajectory tracking MPC is used, then the parameter vector would be

$$\theta(k) = \begin{bmatrix} x(k) \\ u(k-1) \\ r(1|k) \\ \vdots \\ r(H_p|k) \end{bmatrix}^T,$$  

which is of length $n + m + l + mH_p$. Typically $mH_p \gg n + m + l$, hence replacing $H_p$ with the shorter $H_t$ has a significant impact on reducing computational complexity. As will be shown later in the chapter, $H_t$ need only be a fraction of the length of $H_p$ in order to achieve a stable system with a high level of tracking precision. Finally, $H_t = 1$ with an ERT-ZOH is in effect set-point tracking, as implemented by Bemporad (2010).

### 6.2.1 Constraint on the Control Input

The architecture of the motion control system for the machine tool considered in this thesis is as follows. The CNC interprets the part program and generates a sequence of position commands for each of the axes at a rate of 250 Hz. These position commands are determined so that constraints on the axes in terms of position, velocity, acceleration and jerk (the time derivative of acceleration) are met. The only constraint which must be maintained at the DSD level is the magnitude of the motor current; that is, the input to the mechanical plant $u$. Hence only the constraint given by (6.13) must be included in the MPC formulation.

In terms of $\Delta \hat{u}$, (6.13) becomes

$$\hat{u}_{\min}(i) \leq u(k-1) + \sum_{j=0}^{i} \Delta \hat{u}(j|k) \leq \hat{u}_{\max}(i) \quad \forall i \in [0, H_u - 1].$$  

\[ (6.18) \]
The lower bound from (6.18) may be written in matrix form as
\[
\begin{bmatrix}
-I_l & 0_l & \cdots & 0_l \\
-I_l & -I_l & 0_l & \cdots & 0_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
-I_l & -I_l & \cdots & -I_l \\
I_l & 0_l & \cdots & 0_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
I_l & I_l & \cdots & I_l
\end{bmatrix}
\Delta U(k) \leq \begin{bmatrix} I_l \end{bmatrix} \begin{bmatrix} u(k-1) \end{bmatrix} + \begin{bmatrix} -u_{\text{min}}(0) \\
\vdots \end{bmatrix} u_{\text{min}}(H_u - 1),
\tag{6.19}
\]
where \( I_l \) and \( 0_l \) are an identity matrix of size \( l \times l \) and a zero matrix of size \( l \times l \) respectively. For the upper bound
\[
\begin{bmatrix}
I_l & 0_l & \cdots & 0_l \\
I_l & I_l & 0_l & \cdots & 0_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
I_l & I_l & \cdots & I_l \\
-I_l & 0_l & \cdots & 0_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
-I_l & -I_l & \cdots & -I_l \\
0_l & I_l & \cdots & 0_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_l & I_l & \cdots & I_l
\end{bmatrix}
\Delta U(k) \leq \begin{bmatrix} -I_l \end{bmatrix} \begin{bmatrix} u(k-1) \end{bmatrix} + \begin{bmatrix} u_{\text{max}}(0) \\
\vdots \end{bmatrix} u_{\text{max}}(H_u - 1),
\tag{6.20}
\]
The terms on the right-hand side of (6.19) and (6.20) are known at time \( k \), hence (6.13) has been converted into a set of linear inequality constraints on \( \Delta U(k) \). The matrices in (6.19) and (6.20) can be stacked to produce the coefficients of (6.16) as
\[
G_{\hat{u}} = \begin{bmatrix}
-I_l & 0_l & \cdots & 0_l \\
-I_l & -I_l & 0_l & \cdots & 0_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
-I_l & -I_l & \cdots & -I_l \\
0_l & I_l & \cdots & 0_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_l & I_l & \cdots & I_l \\
I_l & I_l & \cdots & I_l \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
I_l & I_l & \cdots & I_l
\end{bmatrix}, \quad X_{\hat{u}} = \begin{bmatrix} 0_x & I_l & 0_r \\
0_x & I_l & 0_r & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_x & -I_l & 0_r \\
0_x & -I_l & 0_r & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_x & -I_l & 0_r \\
0_x & -I_l & 0_r & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_x & -I_l & 0_r \\
0_x & -I_l & 0_r & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_x & -I_l & 0_r
\end{bmatrix}, \quad \mathcal{W}_{\hat{u}} = \begin{bmatrix} -\hat{u}_{\text{min}}(0) \\
-\hat{u}_{\text{min}}(1) \\
\vdots \\
-\hat{u}_{\text{min}}(H_u - 1) \\
\hat{u}_{\text{max}}(0) \\
\hat{u}_{\text{max}}(1) \\
\vdots \\
\hat{u}_{\text{max}}(H_u - 1)
\end{bmatrix}
\]
where \( 0_x \) is an \( l \times (n + m) \) zero matrix and \( 0_r \) is an \( l \times mH_t \) zero matrix. For details on constraints on the input change and the output see Appendix C.2.

### 6.2.2 Integration Effect

For insight into how the MPC formulation using (6.5) provides disturbance rejection via integration, the following analysis is provided. Given \( U(k) \triangleq [\hat{u}(0|k), \ldots, \hat{u}(H_p|k)]^T \), the predicted output \( \hat{y}^* \) for the plant model of (6.1)
MPC DESIGN FOR TRAJECTORY TRACKING

is given by

\[
\begin{bmatrix}
\hat{x}_p(k) \\
\hat{x}_p(1|k) \\
\hat{x}_p(2|k) \\
\vdots \\
\hat{x}_p(H_p|k)
\end{bmatrix}
= \begin{bmatrix}
I \\
A_p \\
A_p^2 \\
\vdots \\
A_p^{H_p}
\end{bmatrix}
\hat{x}_p(k)
\]

\[+
\begin{bmatrix}
0 & 0 & \cdots & 0 \\
B_p & 0 & \cdots & 0 \\
A_p B_p & B_p & 0 & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
A_p^{H_p} B_p & A_p^{H_p-1} B_p & \cdots & B_p
\end{bmatrix}
U(k) \quad (6.21a)
\]

Note that \(\hat{y}^*\) in (6.21) is not necessarily identical to \(\hat{y}\) in (6.10), because the former is derived from \(\hat{x}_p(k)\) and the latter from \(\Delta \hat{x}_p(k) = \hat{x}_p(k) - \hat{x}_p(k-1)\). The subtle distinction here is that the accuracy of the estimate \(\hat{x}_p(k-1)\) dictates whether \(\hat{y} = \hat{y}^*\).

To minimise the overall objective function in (6.9) the magnitude of the tracking error term, which is \(|\hat{y}(i|k) - r(i|k)|\), should be small for each point in the prediction horizon. Using (6.10) and (6.21) the tracking error term can be expanded as

\[
\hat{y}(i|k) - r(i|k) = CA^i \hat{x}(k) + CA^{i-1} B \Delta \hat{u}(0|k) + \cdots + CB \Delta \hat{u}(i-1|k) - r(i|k)
\]

\[= [0_p I_p] \begin{bmatrix}
A_p & 0^T_p \\
C_p A_p & I_p
\end{bmatrix}^i \begin{bmatrix}
\Delta \hat{x}_p(k) \\
y(k)
\end{bmatrix}
\]

\[+
[0_p I_p] \begin{bmatrix}
A_p & 0^T_p \\
C_p A_p & I_p
\end{bmatrix}^{i-1} \begin{bmatrix}
B_p \\
C_p B_p
\end{bmatrix} \Delta \hat{u}(0|k) + \cdots
\]

\[+
[0_p I_p] \begin{bmatrix}
B_p \\
C_p B_p
\end{bmatrix} \Delta \hat{u}(i-1|k) - r(i|k) \quad (6.22)
\]
That is,

\[
\hat{y}(i|k) - r(i|k) = \left( C_p A^i_p + \cdots + C_p A_p \right) \Delta \hat{x}_p(k) + y(k) \\
+ \left( C_p A_p^{-1} B_p + \cdots + C_p B_p \right) \Delta \hat{u}(0|k) + \cdots \\
+ C_p B_p \Delta \hat{u}(i-1|k) - r(i|k) \\
= C_p \left( \hat{x}_p(i|k) - \hat{x}_p(i - 1|k) \right) + C_p \left( \hat{x}_p(i - 1|k) - \hat{x}_p(i - 2|k) \right) + \cdots \\
+ C_p \left( \hat{x}_p(1|k) - \hat{x}_p(0|k) \right) + y(k) - r(i|k) \\
= \hat{y}^*(i|k) - y(k) - \left[ r(i|k) - \left( y(k) - \hat{y}^*(k) \right) \right]
\]

(6.23)

Note that the term \( y(k) - \hat{y}^*(k) \) is constant over the entire prediction horizon. It represents a measure of the error between the current predicted output and the measured output. The difference can be due to either modelling errors, or an unknown plant disturbance. The inclusion of this term has the affect of adjusting the reference trajectory to compensate for the error. The difference between the adjusted reference trajectory and the predicted future outputs is minimised over the prediction horizon by appropriate selection of control inputs. Furthermore, since the variable adjusted in the optimisation is \( \Delta u \), the control signal applied to overcome the error is cumulative and hence acts like an integrator. Obviously if \( y(k) = \hat{y}^*(k) \) then the predicted output is free from error, and hence no (additional) integral action is required.

The inclusion of the integrator is of course only one method which can be used to handle disturbance rejection in a feedback control system. Another established method is that of the Internal Model Principle (Francis and Wonham 1976). The purpose of the internal model is to account for the disturbance signals, be they an unknown external disturbance or a model uncertainty. The method thus requires an additional model that captures the dynamic structure of the disturbance. Due to the unpredictable nature and large magnitude of the disturbances encountered in the machining process under investigation, developing a suitable
6.3 RESULTS AND DISCUSSION

This section details the results from both simulation and experimentation for the proposed MPC approach. The models required for the implementation of MPC are determined by using the techniques outlined in Chapter 3. Depending on the available computation power and the required update rate, either online and offline optimisation is used.

6.3.1 Plant Models

Two plant models are considered here. The model for the servo motor shown in Figure 3.6, is given by (3.11), where the parameters come from Table 5.4. In state-space form the model may be written as

\[
A_p = \begin{bmatrix}
7.7 \times 10^{-1} & 1.6 \times 10^{-4} & 8.3 \times 10^{-2} & -8.3 \times 10^{-2} \\
7.7 \times 10^{-1} & 1.6 \times 10^{-4} & 8.3 \times 10^{-2} & -8.3 \times 10^{-2} \\
4.4 \times 10^{-4} & 9.1 \times 10^{-8} & 1.0 & 1.6 \times 10^{-4} \\
4.4 \times 10^{-4} & 9.1 \times 10^{-8} & 1.0 & 1.6 \times 10^{-4}
\end{bmatrix},
\]

\[
B_p = \begin{bmatrix}
2.3 \times 10^{-3} & 2.3 \times 10^{-3} & 6.1 \times 10^{-7} & 6.1 \times 10^{-7}
\end{bmatrix}^T,
\]

\[
C_p = \begin{bmatrix}
0 & 0 & 5.7 \times 10^1 & 0
\end{bmatrix}.
\]

In this fourth order model \((n = 4)\), the single input \((l = 1)\) is the motor current in milliamps, and the single output \((m = 1)\) is the motor position in degrees. The update period is 500 \(\mu\)s. The model was determined using a combination of system identification techniques and the manufacturer’s data sheet.

The plant model for the headstock axis of the RX7 that is shown in Figure 3.1 is presented in Chapter 3. For both of these plants, the only measured state is the motor position; hence the other states must be estimated using a Kalman filter. In addition, it should be noted that the axes which both models are based on are configured as rotary axes and operate in modulo 360 deg. This must be accounted for when calculating the plant states and also the ERT.

model (across all of the machine’s axes) is not feasible.
6.3.2 Simulation Results

Initially, results obtained via simulation are used to compare the different control approaches. The model used in the simulation to represent the plant, is the reduced-order linear model that was developed in Chapter 3. The investigation begins with a comparison between cascaded PID control, MPC set-point tracking (where $H_t = 1$), and MPC trajectory tracking (where $H_t > 1$), using both ZOH-ERT and FOH-ERT schemes. The objective function weights for the MPC are $Q(i) = 10^6$, $\forall i \in [0, H_p]$, and $R(i) = 1$, $\forall i \in [1, H_u - 1]$, and the input constraints are $\pm 13 \text{ A}$. Since in simulation there is no requirement for real-time calculation, online optimisation is used (see Appendix C.4.1). The reference trajectory is a sinusoid with an amplitude of $2 \text{ deg}$ and a frequency of $10 \text{ Hz}$. In terms of tracking performance, this is a challenging manoeuvre for the RX7 A-axis. Figure 6.2 shows the results for a FOH-ERT with $H_t = H_u = 8$ and $H_p = 50$. The maximum tracking error is found to be $0.0643 \text{ deg}$ or $3.2\%$ of the reference signal amplitude. This can be compared to the performance of a hand tuned cascaded PID controller (Figure 6.3). The maximum tracking error is found to be $0.5021 \text{ deg}$ or $25.1\%$. Figures 6.4 and 6.5 show the normalised cumulative tracking error for each of the control approaches. It is calculated by summing up the magnitude of the tracking error at each control update over a complete cycle of the sinusoidal reference trajectory, and then normalising to the result for MPC set-point tracking. From this graph it can be seen that as the MPC trajectory horizon is increased, the tracking error decreases. Also for $H_t > 2$, both MPC approaches perform better than the PID controller. And most importantly, for values of $H_t$ which are able to be implemented on the available hardware ($H_t = H_u \leq 6$), FOH-ERT is superior to ZOH-ERT.

Another observation from this result is that even if computation was not a constraint, that is, it was possible to select the Trajectory Horizon ($H_t$), the Control Horizon ($H_u$), and the Prediction horizon ($H_p$) to be long enough to in-
clude the reference commands of the entire machining operation, there is little to be gained. Figures 6.4 and 6.5 show that close to optimal performance can be achieved with horizon lengths considerably shorter than that of the entire reference, hence the limit on achievable performance is within the scope of what is possible given the real computational constraints.
Figure 6.2: Simulated tracking performance of the RX7 A-axis for MPC using a FOH-ERT, $H_t = H_u = 8$ and $H_p = 50$.

Figure 6.3: Simulated tracking performance of the RX7 A-axis for a hand tuned cascaded PID controller.
RESULTS AND DISCUSSION

Figure 6.4: Comparison of simulated tracking performance of the RX7 A-axis between cascaded PID control, MPC set-point tracking, and MPC trajectory tracking for both ZOH-ERT and FOH-ERT schemes, where $H_t = H_u$ and $H_p = 50$, for a 10 Hz sinusoidal reference.

Figure 6.5: Comparison of simulated tracking performance of the RX7 A-axis between cascaded PID control, MPC set-point tracking, and MPC trajectory tracking for both ZOH-ERT and FOH-ERT schemes, where $H_t = H_u$ and $H_p = 50$, for a 1 Hz sinusoidal reference.
MODEL PREDICTIVE CONTROL

Figure 6.6: Simulated waterfall chart of the RX7 A-axis for MPC with ZOH-ERT, $H_t = H_u = 4$ and $H_p = 50$.

Figure 6.7: Simulated waterfall chart of the RX7 A-axis for MPC with FOH-ERT, $H_t = H_u = 4$ and $H_p = 50$. 
RESULTS AND DISCUSSION

For insight into why FOH-ERT performs better than ZOH-ERT, Figures 6.6 and 6.7 include waterfall graphs with the motor current. The thin lines show the predicted control input along the entire control horizon. In both simulations $H_t = H_u = 4$ and $H_p = 50$. Knowing that for a ZOH-ERT, the controller is expecting the trajectory reference to come to rest after $H_t$ updates, it is not surprising that $\hat{u}(H_u|k) \approx 0$, $\forall k$. Note that the plant model includes viscous friction, hence the motor current does not actually have to reverse to ensure that the motor velocity approaches zero along the remainder of the prediction horizon. Comparing this to FOH-ERT, it is seen that the predicted control inputs are a closer match to the actual control inputs, that is, the thin lines follow the thick line more closely. As a result of this, the tracking error is greatly reduced.

Figure 6.8 shows the benchmark test that can be directly compared to Figures 4.18 (existing control approach), 5.4 (minimum-time optimal control approach) and 5.8 (LQ control approach). Again the test involved a step change in position of 4 deg. The benchmark result for rise-time is $t = 28.800$ ms and $E = 127.5$ J. In the presented result a perfect plant model is used, with no encoder/current noise and no nonlinear friction. The objective function weights are as stated above. However, the input constraint is set to that of the actual machine, that is $\pm 13$ A. One unique feature of this controller (in comparison to those presented previously in this thesis) is that it anticipates the step change in the reference signal. In the figure, the change in reference actually occurs at 0.0035 s, however the axis actually begins to move $H_t = 8$ control updates before this event.

Figure 6.9 shows how the same MPC controller performs if the objective function weights are chosen so that for the benchmark test, the system constraints never become active. This result can be more closely compared to Figure 5.8, for which the same tuning criterion was used. The weights for the MPC controller are $Q(i) = 5 \times 10^5$, $\forall i \in [0, H_p]$, and $R(i) = 1$, $\forall i \in [1, H_u - 1]$.\footnote{Remember that only the first control input is actually applied to the plant.}
**MODEL PREDICTIVE CONTROL**

Figure 6.8: Simulated benchmarking test for the MPC controller on the RX7 A-axis. Rise-time is \( t = 28.800 \text{ ms} \) and energy usage is \( E = 127.5 \text{ J} \).

Figure 6.9: Benchmarking test for MPC controller without active constraints. Rise-time is \( t = 56.500 \text{ ms} \) and energy usage is \( E = 93.9 \text{ J} \).
The rise-time and energy used in the MPC controller, are tabulated in Table 6.1 alongside those for cascaded PID, minimum-time optimal control and LQ control. The energy used by the MPC controller is more than for LQ control, especially for the case where input constraints are active. This is the expected outcome, since in this case the weights are chosen to be more aggressive, given that the controller can explicitly account for the constraints. As a further result of this tuning decision, the rise-time is also reduced.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Time [ms]</th>
<th>Energy [J]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascaded PID</td>
<td>392.500</td>
<td>129.4</td>
</tr>
<tr>
<td>Minimum-time</td>
<td>19.425</td>
<td>151.4</td>
</tr>
<tr>
<td>LQ</td>
<td>46.500</td>
<td>69.1</td>
</tr>
<tr>
<td>MPC</td>
<td>28.800</td>
<td>127.5</td>
</tr>
<tr>
<td>MPC</td>
<td>56.500</td>
<td>93.9</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison of MPC control with cascaded PID, minimum-time optimal control and LQ control. *Constraints never become active.

Figure 6.10 shows how the tracking error and energy usage vary with the ratio of objective function weights \((Q/R)\). The reference trajectory used in this analysis is a 10 cycle, 20 Hz sinusoid with an amplitude of 3 deg. The MPC horizon lengths are \(H_t = H_u = 8\) and \(H_p = 50\). As expected, the tracking error generally improves as the ratio \(Q/R\) is increased. However, what is interesting is that the energy used increases significantly in the mid-range, then tapers off along with the tracking error at higher ratios. From the figure it appears that the optimal operating point is when the ratio of Objective Function Weights equals approximately \(10^8\). Of course this result is partially a function of the reference signal (and also any disturbance which may exist). As the ratio is increased, it is found that the robustness of the system to unmodelled dynamics and sensor noise is reduced. This provides motivation for a somewhat conservative selection of the \(Q/R\) ratio in both the simulated and experimental sections of the research. Figure 6.11 shows the simulated system response to the sinusoidal reference trajectory for \(Q/R\) equal to \(10^5\), \(10^6\), \(10^7\) and \(10^8\). Note that the simulations do not include
sensor noise or system disturbances. Including these dynamics into the analysis is left for future work. In all the responses, the control input (motor current reference) is limited at various points by the input constraint given in (6.13), where \( \hat{u}_{\text{max}} = 13 \text{ A} \). As expected from Figure 6.10, Figure 6.11 shows how the tracking error is reduced as \( Q/R \) is increased. The reason for this behaviour is essentially that with a larger ratio of objective function weights the controller is more aggressive and the onset of increasing control input occurs earlier in each sinusoidal cycle.

![Graph showing tracking error and energy usage vs ratio of objective function weights](image)

Figure 6.10: Comparison of tracking error and energy usage versus the ratio of objective function weights. The MPC horizon lengths are \( H_t = H_u = 8 \) and \( H_p = 50 \).
Figure 6.11: Simulated response to a 20 Hz sinusoid reference trajectory with an amplitude of 3 deg for various ratios of the objective function weights (Q/R). The MPC horizon lengths are $H_t = H_u = 8$ and $H_p = 50$. 
6.3.3 **Experimental Results – The Fridge Test Rig**

Finally, implementation of MPC on a test rig constructed using an industrial CNC and DSDs (Figure 3.5) is presented. The performance of MPC is compared to that of both the classical PID controller and the LQ controller developed in Chapter 5. Again the input constraints are set to $\pm 1000 \text{ mA}$. Due to the limited computation power of the processor on the DSD, online optimisation is not possible; even EMPC with sequential search is found to be unachievable for $H_t = H_u > 2$ (essentially when more than 15 regions, or approximately 50 hyperplanes, must be checked to locate which region the parameter vector belongs to). However, using the binary search tree approach (Tøndel et al. 2003b), it is possible to implement MPC with longer horizons; as stated previously, up to $H_t = H_u = 6$. See Appendix C.4.2 for the method to generate the EMPC solution, and Appendix C.4.3 to convert it into a binary search tree.

Figure 6.12 shows the Simulink model used with Simulink Coder (MathWorks 2010) to create the drive code that is subsequently loaded onto the DSD. There are four main blocks which together make up the controller design. The first is the *Trajectory Generator*. This block buffers the position commands received from the CNC, so as to create a reference trajectory vector of length $H_t$. The block also adjusts the reference trajectory vector to take account of the modulo behaviour of the rotary axis. The next block is the *Kalman Estimator*. As with the LQ controller design presented in Chapter 5, this block estimates the system states as described by (5.19). The block also accounts for the modulo operation. The estimated states are then fed into the *State Augmentation* block which implements (6.2) to generate the augmented state vector. Finally, there is the *Optimiser* block, which uses the reference trajectory vector and the augmented state vector to search the pre-generated binary search tree representation of the EMPC for the optimal control input, which is then applied to the plant.

---

2The motor is actually rated to 13 A. The setting of such a low value for the experiments allows a sizeable disturbance to be introduced manually.
Figure 6.12: Explicit MPC Simulink model.

Figure 6.13 shows how an EMPC controller with a FOH-ERT, $H_t = H_u = 3$, and $H_p = 50$ performs on the test rig. The controller weights used are $Q(i) = 10^6$, $\forall i \in [0, H_p]$ and $R(i) = 1$, $\forall i \in [1, H_u - 1]$. The EMPC contains 11 controllers across 27 polyhedral regions, and is organised into a binary search tree with 139 nodes, where the maximum depth is 7. The trajectory being tracked by the axis is essentially a trapezoidal velocity profile, with a stroke of 540 deg and a maximum velocity of 100 RPM. Since the current remains well within the input limits, the EMPC never leaves Region 1, which corresponds to controller #1 and linear operation (no constraints are active over the entire control horizon). Comparing to the PID controller (Figure 6.14) it is seen that the magnitude of the tracking error is much smaller. Note that the reference trajectory is the same as that used in Figure 6.13, with the difference in the displayed graphs due to the data log being triggered at a different point in the trajectory. The observed oscillation in the tracking error in Figure 6.14 is caused by the motor cogging torque.

The results can also be compared to those obtained from the implementation of the LQ controller presented in Section 5.2.3. Figure 5.14 shows how the LQ controller performed when implemented on the same hardware. As can be seen,
the magnitude of the tracking error and the current commands are very similar to that of the EMPC controller implementation. This is expected since without active constraints, the two control approaches share a similar formulation and a comparable quadratic cost function.

Figure 6.15 shows how the MPC responds to a large position disturbance, introduced by twisting the motor shaft away from the desired set-point by hand. This is possible because of the low motor current limit that was set during the controller design. Shortly after the disturbance is removed at 0.35 s, the position feedback initially overshoots but eventually settles at the set-point. The graph also shows how the controller transitions between different polyhedral regions (with different controllers) as the input constraints along the control horizon become active. Note that while the EMPC solution actually contains 11 controllers, only three (#1, #2 and #3) are required during the transient stage after the disturbance is removed. As a final note, with the current limits set to 13 A the axis is found to be very stiff and the integrator within the MPC effectively rejects realistic disturbances.

Both the cascaded PID controller and the LQ controller have mechanisms to saturate the motor current command if required. The main difference to MPC is that with those approaches the saturation is done by saturating the calculated control input, which is somewhat ad hoc. In the MPC approach, the decision to saturate the control input is done as part of the optimisation, since it is explicitly included in the problem formulation.
Figure 6.13: The Fridge test rig tracking a trapezoidal velocity profile using EMPC with FOH-ERT, $H_t = H_u = 3$, and $H_p = 50$.

Figure 6.14: The Fridge test rig tracking a trapezoidal velocity profile using a hand tuned cascaded PID controller.
Figure 6.15: The Fridge test rig rejecting a large position disturbance using EMPC with FOH-ERT, $H_t = H_u = 3$, and $H_p = 50$. 
6.3.4 Experimental Results – RX7 CNC Grinding Machine

After successful implementation of MPC on The Fridge test rig, focus now switches to implementation on an actual CNC machine tool. As before, this is carried out on the RX7 shown in Figure 3.1. The control system used in the RX7 is identical to that of The Fridge test rig, and so implementation is straightforward.

The process starts by using a similar calibration application to the one which was described in Section 5.2.3. The routine has several steps which are carried out in order. Firstly, the system model is identified using the techniques outlined in Chapter 3. Next the solution to the Explicit MPC (EMPC) is calculated, which is then formed into a binary search tree (Tøndel et al. 2003b). The search tree is them embedded into the Optimiser block shown in Figure 6.12. Note that these steps may be readily integrated into a single software application for deployable versions of the control system to ensure good performance across a family of plants with significantly different dynamics.

The plant model is given by (3.11), where the parameters for both the small and large workpieces are displayed in Table 6.2. Note that the implementation of MPC was done on a different RX7 machine to the one used for the LQ Controller implementation. Its parameters, while not identical to those previously identified (Table 5.5), are nevertheless similar. The fact that the model parameters are observed to change further motivates the need to conduct the calibration routine to ensure optimum performance.

While the analysis in Section 6.3.2 (specifically Figure 6.10) showed that increasing the ratio $Q/R$ to $10^8$ would produce the best performance. The reality is that at such a high level, significant emphasis is placed on both the accuracy of the plant model and the accuracy of the sensor data (low noise). When this is not the case, the system can become unstable. Due to this, the weights used in the controller design are $Q(i) = 10^7, \forall i \in [0, H_p]$ and $R(i) = 1, \forall i \in [1, H_u - 1]$.  

171
The horizon lengths are $H_t = H_u = 3$ and $H_p = 50$. For the small workpiece the resulting EMPC contains 9 controllers across 23 regions. The resulting binary search tree has a maximum depth of 6. Conversely, for the large workpiece the resulting EMPC contains 11 controllers across 29 regions, with a maximum depth of 7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Small Workpiece</th>
<th>Large Workpiece</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_t$</td>
<td>3.4625</td>
<td>3.4481</td>
<td>N·m/A</td>
</tr>
<tr>
<td>$J_m$</td>
<td>0.0139</td>
<td>0.0143</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_w$</td>
<td>0.0507</td>
<td>0.0002</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$B_m$</td>
<td>0.0287</td>
<td>0.0241</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$c_{mw}$</td>
<td>0.6115</td>
<td>0.5646</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$k_{mw}$</td>
<td>89124</td>
<td>79953</td>
<td>N·m/rad</td>
</tr>
</tbody>
</table>

Table 6.2: RX7 A-axis mechanical plant parameters.

Figures 6.16 to 6.19 show the main results from the experiments carried out on the RX7. Figure 6.16 shows the closed-loop system’s performance at regulating a constant setpoint. The magnitude of the position error is slightly larger than that of the existing cascaded PID controller (Figure 4.20), but similar to that of the LQ controller (Figure 5.16). While a slight step backwards in performance, it is still within the specifications for tracking accuracy (see Section 3.7). This shortcoming is more than made up for by the performance gains achieved in the tracking experiments. As a side note, and as observed previously, the nominal motor current is observed to be non-zero. This is a direct result of accumulation by the integrator embedded into the MPC formation in response to the existence of nonlinear (static) friction.

The graphs shown in Figures 6.17 and 6.18 show how the controller performs when required to track a trapezoidal velocity profile. As can be seen, the position errors for both small and large workpieces are much smaller than those of the existing controller (Figure 4.21). While of the same order of magnitude, the position errors are slightly worse than those of the LQ controller (Figure 5.17). This may be due to the fact that the experiments were conducted on different RX7
machines. nevertheless, it is believed that this deficiency in performance may be remedied by a slight increase in the ratio $Q/R$. This would further penalise the tracking error in the objective function. However, due to a limited time allocation on the machine, this hypothesis could not be tested.

The real benefit of MPC over the LQ controller from the previous chapter is evident in Figure 6.19, where the controller is seen to actively account for the system constraints. Shown here is the controller’s response after a large persistent torque disturbance is removed from the workpiece. As was the case with the experiments on The Fridge test rig, the controller’s output saturates in response to the disturbance. As a result of this, the controller number is seen to be equal to five in both cases. This corresponds to the situation where a constraint (the input constraint) is active somewhere along the control horizon. Shortly after the disturbance is removed, the controller transitions back into the linear region (Controller Number 1), and quickly returns to the setpoint. When compared to the same experiment carried out on The Fridge test rig (Figure 6.15), the response appears to be more damped, exhibiting less oscillation and does not bounce between current limits. Again this is influenced by the selection of cost function weights. Finally, note that in the case of the LQ Controller, the cost function weights were selected so that the constraints would not typically be active during normal operation.

---

3^Note that the motor current limit has been intentionally reduced to 1A to enable the disturbance to be introduced manually.
Figure 6.16: RX7 using EMPC with FOH-ERT standstill, $H_t = H_u = 3$, and $H_p = 50$, for both small (left) and large (right) workpieces.

Figure 6.17: RX7 using EMPC with FOH-ERT to track reference trajectory (constant velocity), $H_t = H_u = 3$, and $H_p = 50$, for both small (left) and large (right) workpieces.
Figure 6.18: RX7 using EMPC with FOH-ERT to track reference trajectory (reversal), $H_t = H_u = 3$, and $H_p = 50$, for both small (left) and large (right) workpieces.

Figure 6.19: RX7 using EMPC with FOH-ERT rejection, $H_t = H_u = 3$, and $H_p = 50$, for both small (left) and large (right) workpieces.
6.4 CONCLUSION

The objective of this chapter was to develop a methodology to enable the benefits of MPC to be utilised for control of a machine tool servo drive – a trajectory tracking system which has both fast update rates and limited computational power. The key barrier to implementation on such a system is the large computational requirements to complete the optimisation. Initially, in an effort to reduce the online computation required, the use of recently developed explicit MPC solutions was investigated. However, this proved unsuccessful since the large complexity of the trajectory tracking MPC problem means that the amount of time required to search the resulting parameter space is prohibitively long. It became evident that what was required was a way to reduce the complexity of the optimisation problem. A caveat on this requirement was that where possible the performance benefits provided by the use of MPC had to be maintained.

In this chapter a novel MPC formulation is proposed. The inclusion of the trajectory horizon \(H_t\) and the first-order hold extended reference trajectory (FOH-ERT) demonstrated that it is possible to implement trajectory tracking MPC on an industrial servo axis with update periods of order \(100 \mu s\). Further investigations showed that up until a certain point, extending the trajectory horizon improves the tracking performance, but beyond this point little is gained. However, increasing the trajectory horizon comes at the expense of increasing the optimisation complexity. Fortunately, the critical length for the trajectory horizon was found to be a small fraction of the length of the prediction horizon that is required for stability. With this discovery it was shown that both the full benefits of MPC and also implementation on actual industrial hardware is possible. Superior performance is achieved in comparison to set-point tracking MPC, LQ control and cascaded PID control. Furthermore, to the best of the author’s knowledge this is the first implementation of trajectory tracking MPC on an industrial machine tool.
In combination with the calibration routine developed in Chapter 5, which in turn used the results from Chapter 3, this chapter has provided an implementable solution to the problem of maintaining a high level of servo tracking performance in the presence of changing load dynamics. This allows an inexperienced end-user to independently deal with the situation of reduced servo tracking performance or instability caused by a change in the plant dynamics.
This research project has made significant contributions in the areas of modelling machine tool servo drive axes, and analysis of the associated critical system dynamics. Furthermore, investigations into the performance limitations of traditional servo drive control techniques have been made, along with an analysis of the benefits provided by the use of advanced model-based control techniques. Finally, considerations for the practical implementation of MPC have been investigated. This lead to the development of a novel MPC formulation.

7.1 Contributions to the Modelling and Analysis of Machine Tool Servo Drives

The early sections of the thesis concentrated on developing models of the machine tool servo drive axis identified as the focus for this research project. The motivation for this work was to better understand the critical system dynamics which lead to an experimentally observed system instability.

What are the critical dynamics which lead to the observed instability, and to account for it, what level of model fidelity is required?

To answer the first research question an extensive investigation into the dynamics exhibited by the servo axis was undertaken. A systematic approach for the correct selection of model order, and which elements to include in the grey box model was provided, along with detailed descriptions of data collection experiments and system identification techniques. To simplify the subsequent analysis, where possible, linear models were identified. The main exception to this was
CONCLUSIONS

the model for the nonlinear friction. Since the observed instability occurs at relatively low velocities it is prudent to include an analysis of the frictional effects. The final stage of the modelling work was to validate the response of the models against a variety of experimentally collected data.

Given these models, an analysis into the root cause of the system instability was undertaken. From the modelling work a dominant, lightly-damped mechanical resonance was identified. Initially this dynamic mode alone was thought to be the key to explain the experimentally observed instability. Using the identified plant model in conjunction with a model of the cascaded PID control system, linear analysis techniques were used to test this hypothesis. However, using the standard approach of simply ignoring the inner loop dynamics, on the grounds that the time scale difference between mechanical and electrical system is sufficiently large, meant that predicting the instability was not possible. As part of this discovery it was revealed that it is also necessary to include the effects of computation delays, an unavoidable characteristic of practical implementations of digital control. The inclusion of the inner loop dynamics necessitated the development and use of a multirate analysis technique. Finally, to fully validate the results, an analysis of the effects of nonlinear friction was also conducted.

The critical dynamics which lead to the observed instability were found to be the combination of mechanical resonance, phase delay introduced by the inner loop dynamics, and the high state of controller tuning deemed necessary to achieve the performance requirements. Together these elements constitute the minimum requirement to analytically reproduce the observed results.

In analysing the cause of the closed-loop instability, several limiting factors of the existing cascaded PID control approach were discovered. The main aspects, which relate to this thesis, are that fixed controller gains are not suitable for changing load dynamics, and that lightly-damped mechanical resonances make controller tuning a challenging exercise for even experienced engineers and tech-
nicians. These deficiencies motivate the investigation into advanced model-based control approaches.

While the contributions in this section clearly have implications for control system engineers, designers of machine tools may also benefit. Gaining knowledge into how structural design decisions affect the system dynamics upfront, should allow designers to produce a superior machine. Hence reducing the requirement on the control system to deal with problematic dynamics during the system integration phase, or worse, require the machine mechanism to be re-designed.

Are the models used for analysis also suitable for model-based controller design?

To answer the second research question, the results from both the modelling/analysis and the controller design sections of the thesis must be considered. As stated above, the fast inner loop dynamics were required for successful prediction of the experimentally observed instability; however acceptable performance was achieved by designing model-based controllers without including these elements.

The main motivation for the decision to use a further reduced-order model was to moderate the complexity of the practical control implementation. Since constraints on computational complexity were known to be the main roadblock for successful implement of advanced model-based control; exploring the possibility of using reduced-order models was investigated at an early stage. This highlights the need to develop a range of models, with different levels of complexity, and select among them for the specific task at hand: specifically, it was found that in depth analysis required high fidelity models, while those with reduced-order were better suited to practical implementations of model-based control.
Conclusions

7.2 Contributions to the Control of Machine Tool Servo Drives

The later sections of the thesis were focused on investigating the performance benefits of model-based control, and developing techniques for successful implementation on actual industrial hardware. The criteria against which a model-based control approach was assessed included: its effectiveness at providing high performance servo control of the system under investigation, the ease with which it is tuned/calibrated, and the ability to be implemented in practice.

There has been significant advancement in the theory underpinning model-based control: can model-based control provide an improved level of performance and be easily adapted to deal with both mechanical resonance and changing plant dynamics?

The work which goes to answering the third research question was covered in Chapters 5 and 6 of the thesis: using the models developed in the early chapters to design model-based controllers that provided practical solutions to the identified control challenges. The goal was to, firstly, improve the servo performance of an axis containing mechanical resonance with an increase in tracking precision and a reduction in energy usage, and secondly, provide a straightforward way to tune the controller and thus provide a better method to deal with changing load dynamics.

To set the benchmark, minimum-time optimal control was investigated. This provided the standard by which other control methods could be compared in terms of a measure for the highest possible servo performance (within the constraints of the system). While useful as a benchmark, minimum-time optimal control is not suitable for implementation on the actual hardware. Owing to its open-loop structure, it cannot account for model uncertainty and unknown system disturbances.

The next approach investigated was a tailored version of Linear Quadratic
(LQ) control. The motivation here was to begin with a controller that could be easily implemented, to ascertain if model-based control could, and to what level, address the goals stated above. Since this control approach is intended for a linear system, the controller includes feedback linearisation to account for the effects of nonlinear friction. In practice it was discovered that even though the nonlinear friction was quite a dominant characteristic, there was enough dither within the working environment for the feedback linearisation to be unnecessary and even a hindrance. Also included is a Kalman filter to estimate the full state vector and an integrator to handle unknown system disturbances (including the nonlinear friction). A good feature of LQ control is the ease with which it is tuned. Once a suitable model of the system is developed, the controller tuning is achieved by solving the optimisation involving the objective function. While this controller was quite effective at improving tracking performance and reducing energy usage (as shown in both simulation and on the actual machine tool), several ad-hoc additions to the standard controller structure, specifically with regard to the anti-windup treatment of the integrator, were required to achieve this result. In other words, this control approach does not explicitly account for system constraints. This is the motivation for investigating the suitability of Model Predicative Control (MPC). Fortunately the advantages in terms of tuning the LQ controller also carry over to MPC.

It was shown that model-based control, specifically MPC, does provide a suitable replacement for the incumbent approach of cascaded PID. Not only was it possible to achieve superior tracking performance, but the controller effectively dealt with mechanical resonance. However, as was the case for the PID controller, robustness to changing plant dynamics was a problem for high performance model-based controllers. On the plus side though, it was shown that when necessary, the model-based controller can be easily retuned using automated system identification techniques.
CONCLUSIONS

Can model-based control be applied to actual industrial hardware?
Answering the final research question was pivotal to the success of the entire research project. Of the three model-based control approaches investigated in the research project, two were successfully implemented on the existing industrial hardware. The third, minimum-time optimal control, was not suitable for implementation for the reason given previously.

The benefits of MPC are well known. However the limitation, which up until now, has prevented its implementation on fast update trajectory tracker servo systems is the amount of computation time required. In this thesis, a novel MPC formulation was developed, which reduced the complexity of the MPC problem to something that can be implemented on the available hardware. Analysis of the proposed formulation shows that little is lost in terms of servo tracking performance, when compared to previous implementations of trajectory tracking MPC. The formulation is compatible with recently developed Explicit MPC (EMPC) solutions, and advantage is taken of this detail during the implementation stages of the approach. Again superior performance is achieved on the machine tool axis under investigation over the incumbent approach of cascaded PID. To the author’s best knowledge, this is the first time that MPC has been successfully implemented on an industrial trajectory tracking servo system with update periods in the order of 100 µs.

An important aspect of the work in this thesis is that the resulting controller design should be easy to tune, to the point where an automated tuning routine could be developed. While a complete software application to achieve this was not completed, each individual step in the process was successfully automated. Firstly a script to automatically conduct the system identification was developed. The resulting model was then fed into the controller tuning process, which included both the EMPC design and the conversion into a binary search tree. Finally the resulting controller parameters were automatically loaded into mem-
FURTHER RESEARCH OPPORTUNITIES

ory on the DSP. Fine tuning then amounts to selection of the objective function weights, which in effect trade-off tracking performance with control effort, and with that, system robustness. It is a straightforward task to integrate these steps into an automated software application.

Finally, the analysis techniques and controller designs are suitable to machine tool servo drives in the general sense. The machine tool axis used as the focus of the work in this thesis exhibits most of the main dynamics common to every machine tool axis. One exception that was not considered is backlash, a common characteristic found in systems with gearboxes or lead screws. This thesis focused on the modern approach of using direct-drive axes, where backlash is typically not an issue.

7.3 FURTHER RESEARCH OPPORTUNITIES

In terms of modelling and analysis, possible future research opportunities include:

- Applying the processes developed in this thesis to analyse cases of instability in other machine tool applications. Where possible, effort was given to generalise the work, but due to time constraints actual investigations on other systems was not carried out. Other machines may include different problematic dynamics, and as such, may require a slightly different approach to the one presented in this work.

- Investigating the effect vibration from the grinding process plays on axis stability. In this thesis, analysis and experiments were carried out in free air. However, it is conceivable that interaction between the grinding wheels and the workpiece may excite other dynamics in the machine’s mechanical structure. While any discovery in this regard may be difficult to compensate for in the controller design, it would provide valuable guidance for machine
CONCLUSIONS

tool designers.

In terms of model-based control, research topics include:

- Further improvements to practical implementations of MPC on fast update systems. While the work in this thesis did enable MPC to be implemented for the first time on an actual industrial machine tool servo drive, limits on the model order, the number of system constraints, and length of trajectory/control horizons are still issues to be addressed. Ways of approaching this may include: further improvements to online optimisation techniques which take additional advantage of the structure of the MPC problem, or methods to generate explicit solutions that have reduced complexity and online storage requirements. Of course, as the power of embedded processors continues to increase, especially with the recent emergence of multicore embedded processors, these research opportunities may cease to be required, but nevertheless would always provide a benefit.

- Investigating the possibility of applying MPC in a holistic approach to the control of machine tools. In this thesis, controller design was focused on a single machine tool axis. Clearly this work could be repeated on each individual axis to improve overall system performance. However further benefit may be gained by considering the overall machine tool structure and dynamics in a single MPC formulation. While likely to be computationally demanding, such an approach would, in essence, provide system wide optimal control.
REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_c$</td>
<td>Chirp signal amplitude</td>
</tr>
<tr>
<td>$a_n$</td>
<td>Stator $a$-axis negative IGBT switching time</td>
</tr>
<tr>
<td>$a_p$</td>
<td>Stator $a$-axis positive IGBT switching time</td>
</tr>
<tr>
<td>$B_c$</td>
<td>Chirp signal constant offset</td>
</tr>
<tr>
<td>$B_m$</td>
<td>Viscous damping coefficient</td>
</tr>
<tr>
<td>$B^*_m$</td>
<td>Effective viscous damping coefficient</td>
</tr>
<tr>
<td>$b_n$</td>
<td>Stator $b$-axis negative IGBT switching time</td>
</tr>
<tr>
<td>$b_p$</td>
<td>Stator $b$-axis positive IGBT switching time</td>
</tr>
<tr>
<td>$c_{md}$</td>
<td>Drawbar shaft torsional damping coefficient</td>
</tr>
<tr>
<td>$c_{mw}$</td>
<td>Workpiece shaft torsional damping coefficient</td>
</tr>
<tr>
<td>$c_n$</td>
<td>Stator $c$-axis negative IGBT switching time</td>
</tr>
<tr>
<td>$c_p$</td>
<td>Stator $c$-axis positive IGBT switching time</td>
</tr>
<tr>
<td>$C_{xz}$</td>
<td>Magnitude squared coherence estimate of $x$ and $z$</td>
</tr>
<tr>
<td>$d$</td>
<td>Number of sample delays due to computation</td>
</tr>
<tr>
<td>$d_c$</td>
<td>Coupling diameter</td>
</tr>
<tr>
<td>$E_f$</td>
<td>Energy dissipation due to friction</td>
</tr>
<tr>
<td>$f_c$</td>
<td>Chirp signal instantaneous frequency</td>
</tr>
<tr>
<td>$G$</td>
<td>Shear modulus of elasticity</td>
</tr>
<tr>
<td>$\hat{G}_{uy}$</td>
<td>Transfer function estimate from input $u$ to output $y$</td>
</tr>
<tr>
<td>$H$</td>
<td>Hamiltonian</td>
</tr>
<tr>
<td>$H_p$</td>
<td>Prediction horizon</td>
</tr>
<tr>
<td>$H_t$</td>
<td>Trajectory horizon</td>
</tr>
<tr>
<td>$H_u$</td>
<td>Control horizon</td>
</tr>
<tr>
<td>$i_a$</td>
<td>Stator $a$-axis current</td>
</tr>
<tr>
<td>$i_b$</td>
<td>Stator $b$-axis current</td>
</tr>
</tbody>
</table>
### NOMENCLATURE

- **$i_c$**: Stator $c$-axis current
- **$i_d$**: Direct axis current
- **$i_f, i_f^*$**: Motor current reference (filtered)
- **$i_{LQGTI}$**: Output of LQ tracking controller with Kalman filter and integrator
- **$i_{LQGT}$**: Output of LQ tracking controller with Kalman filter
- **$i_{LQT}$**: Output of LQ tracking controller with integrator
- **$i_q$**: Quadrature axis current
- **$\overline{i}_q$**: Quadrature axis current (zero mean)
- **$i_r$**: Motor current reference
- **$\overline{i}_r$**: Motor current reference (zero mean)
- **$i_r^*$**: Motor current reference (delayed)
- **$I_z$**: Polar moment of inertia
- **$J$**: Cost or objective function
- **$J_d$**: Drawbar moment of inertia
- **$J_m$**: Motor shaft moment of inertia
- **$J_w$**: Workpiece moment of inertia
- **$K$**: Number of samples in the data set
- **$K_f$**: Friction compensation gain
- **$K_I$**: Integrator gain
- **$k_{md}$**: Drawbar shaft torsional stiffness
- **$k_{mw}$**: Workpiece shaft torsional stiffness
- **$K_t$**: Motor torque constant
- **$L$**: Kalman gain
- **$l_c$**: Coupling length
- **$L_l$**: Stator leakage inductance
- **$L_{l-l}$**: Line-to-line self-inductance of the stator windings
- **$L_m$**: Stator magnetising inductance
- **$L_s$**: Self-inductance of the stator windings
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Innovation gain</td>
</tr>
<tr>
<td>$M_p$</td>
<td>Complementary sensitivity function</td>
</tr>
<tr>
<td>$M_p$</td>
<td>Complementary sensitivity function</td>
</tr>
<tr>
<td>$M_s$</td>
<td>Sensitivity function</td>
</tr>
<tr>
<td>$M_s$</td>
<td>Sensitivity function</td>
</tr>
<tr>
<td>$N_i$</td>
<td>LQ reference gain</td>
</tr>
<tr>
<td>$N_x$</td>
<td>LQ state gain</td>
</tr>
<tr>
<td>$p$</td>
<td>Number of motor poles</td>
</tr>
<tr>
<td>$p$</td>
<td>Costate equations</td>
</tr>
<tr>
<td>$P_f$</td>
<td>Instantaneous power dissipation due to friction</td>
</tr>
<tr>
<td>$Q$</td>
<td>Tracking error weight</td>
</tr>
<tr>
<td>$Q_e$</td>
<td>Covariance of the process noise</td>
</tr>
<tr>
<td>$Q_n$</td>
<td>Notch-filter quality factor</td>
</tr>
<tr>
<td>$R$</td>
<td>Control effort weight</td>
</tr>
<tr>
<td>$R_e$</td>
<td>Covariance of the measurement noise</td>
</tr>
<tr>
<td>$r$</td>
<td>Reference trajectory</td>
</tr>
<tr>
<td>$\hat{r}$</td>
<td>Extended reference trajectory</td>
</tr>
<tr>
<td>$R_{l-l}$</td>
<td>Line-to-line resistance of the stator windings</td>
</tr>
<tr>
<td>$R_s$</td>
<td>Resistance of the stator windings</td>
</tr>
<tr>
<td>$S_{xx}$</td>
<td>Auto power spectrum</td>
</tr>
<tr>
<td>$S_{xz}$</td>
<td>Cross power spectrum</td>
</tr>
<tr>
<td>$T$</td>
<td>Sample-period</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Coulomb friction torque</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Derivative time constant</td>
</tr>
<tr>
<td>$T_e$</td>
<td>Total external torque</td>
</tr>
<tr>
<td>$T_f$</td>
<td>Friction torque</td>
</tr>
<tr>
<td>$T_f^*$</td>
<td>Friction torque (purely viscous)</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Integral time constant</td>
</tr>
</tbody>
</table>
NOMENCLATURE

\[ T_l \] Load torque
\[ T_{md} \] Torque acting between the motor and drawbar rigid bodies
\[ T_{mw} \] Torque acting between the motor and workpiece rigid bodies
\[ T_m \] Electromagnetic torque
\[ T_s \] Static friction torque
\[ T_{st} \] Magnitude of the Stribeck friction torque
\[ u \] Input vector
\[ \Delta u \] Change in input vector
\[ \Delta \hat{u} \] Future change in the input vector
\[ \mathcal{U} \] Sequence of control inputs
\[ \Delta \mathcal{U} \] Sequence of change in the control input
\[ v_a \] Stator \( a \)-axis voltage
\[ v_a^* \] Stator \( a \)-axis voltage reference
\[ v_b \] Stator \( b \)-axis voltage
\[ v_b^* \] Stator \( b \)-axis voltage reference
\[ v_c \] Stator \( c \)-axis voltage
\[ v_c^* \] Stator \( c \)-axis voltage reference
\[ V_{dc} \] Servo drive bus voltage
\[ v_d \] Direct axis voltage
\[ v_q \] Quadrature axis voltage
\[ \hat{x} \] Augmented estimated plant state vector
\[ x \] Augmented plant state vector
\[ x_I \] Integrator state
\[ x_p \] Plant state vector
\[ \Delta x_p \] Change in plant state vector
\[ \hat{x}_p \] Estimated plant state vector
\[ \Delta \hat{x}_p \] Change in estimated plant state vector
\[ y \] Output vector
NOMENCLATURE

\( \hat{y} \)  Predicted output vector
\( \varepsilon \)  Small boundary around zero velocity
\( \theta_d \)  Drawbar position
\( \theta_e \)  Electrical angle
\( \theta_m \)  Motor position
\( \theta_r \)  Motor position reference
\( \theta_w \)  Workpiece position
\( \Lambda_f \)  FOH extended reference trajectory
\( \Lambda_z \)  ZOH extended reference trajectory
\( \xi_l \)  Proportion of leakage inductance
\( \psi_a \)  Stator \( a \)-axis magnetic flux
\( \psi_b \)  Stator \( b \)-axis magnetic flux
\( \psi_c \)  Stator \( c \)-axis magnetic flux
\( \psi_d \)  Direct axis magnetic flux
\( \psi_m \)  Permanent magnet magnetic flux
\( \psi_q \)  Quadrature axis magnetic flux
\( \omega_{lp} \)  Low-pass filter corner frequency
\( \omega_m \)  Motor velocity
\( \bar{\omega}_m \)  Motor velocity (zero mean)
\( \omega_n \)  Notch-filter center frequency
\( \omega_r \)  Motor velocity reference
\( \omega_s \)  Characteristic velocity of the Striebeck friction torque
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analogue to Digital Converter</td>
</tr>
<tr>
<td>APS</td>
<td>Auto Power Spectrum</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CNC</td>
<td>Computer Numerical Control</td>
</tr>
<tr>
<td>CPS</td>
<td>Cross Power Spectrum</td>
</tr>
<tr>
<td>DARE</td>
<td>Discrete-time Algebraic Riccati Equation</td>
</tr>
<tr>
<td>DSD</td>
<td>Digital Servo Drive</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>ES</td>
<td>Extremum Seeking</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
</tr>
<tr>
<td>FOH</td>
<td>First Order Hold</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithms</td>
</tr>
<tr>
<td>IFT</td>
<td>Iterative Feedback Tuning</td>
</tr>
<tr>
<td>IGBT</td>
<td>Insulated-Gate Bipolar Transistor</td>
</tr>
<tr>
<td>IPM</td>
<td>Intelligent Power Module</td>
</tr>
<tr>
<td>LQ</td>
<td>Linear Quadratic</td>
</tr>
<tr>
<td>LQG</td>
<td>Linear Quadratic Gaussian</td>
</tr>
<tr>
<td>LQGT</td>
<td>Linear Quadratic Gaussian with Tracking</td>
</tr>
<tr>
<td>LQT</td>
<td>Linear Quadratic with Tracking</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear Time-Invariant</td>
</tr>
<tr>
<td>MPC</td>
<td>Model Predictive Control</td>
</tr>
<tr>
<td>NC</td>
<td>Numerical Control</td>
</tr>
<tr>
<td>PEM</td>
<td>Prediction-Error identification Method</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional plus Integral plus Derivative</td>
</tr>
<tr>
<td>ACRONYM</td>
<td>FULL FORM</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>PLC</td>
<td>Programable Logic Controller</td>
</tr>
<tr>
<td>PMSM</td>
<td>Permanent Magnet Synchronous Motor</td>
</tr>
<tr>
<td>PPI</td>
<td>Part Program Interpreter</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>SMC</td>
<td>Sliding Mode Control</td>
</tr>
<tr>
<td>TCG</td>
<td>Tool and Cutter Grinding</td>
</tr>
<tr>
<td>VPI</td>
<td>Velocity Profile Interpolator</td>
</tr>
<tr>
<td>ZN</td>
<td>Ziegler-Nichols</td>
</tr>
<tr>
<td>ZOH</td>
<td>Zero Order Hold</td>
</tr>
<tr>
<td>ZPETC</td>
<td>Zero Phase Error Tracking Control</td>
</tr>
</tbody>
</table>
A.1 Friction Modelling

The move in research effort to model-based methods, which has occurred over the past 15 years, is partly due to the availability of increasingly powerful processors. The ability to execute the model simulation and calculate a control output in real-time is the Achilles’ heel of model-based methods. It is only recently that sufficient processing power has become available to facilitate the realisation of these techniques. Of course, a model-based approach is limited by the accuracy of the models it employs. One of the hardest effects to model is friction, so recent work in the area of model-based friction compensation has concentrated on developing new friction models. Examples include, the LuGre friction model (Canudas de Wit et al. 1995), and developing adaptive compensation algorithms like those proposed by Canudas de Wit and Lischinsky (1997) and Altpeter et al. (2000). The approaches presented can be grouped into both model-based and non-model-based friction compensation methods. Non-model-based methods include:

- **dither**, where a high frequency signal is inserted into the system with the effect of smoothing the discontinuity of friction at low velocity (Mossaheb 1983);

- **impulse control**, where the system is controlled around zero velocity via a series of short impulse signals (Yang and Tomizuka 1988); and

- **joint torque control**, which relies on torque or force sensors to feedback the effects of friction, thus creating a friction compensation loop (Wu and Paul
1980).

Model-based methods include:

- **fixed compensation**, where given there is a suitable friction model available, it is possible to compensate by applying a force/torque reference (feedforward) equal and opposite to the friction force (Johnson and Lorenz 1992; Tung and Tomizuka 1993); and

- **friction identification and adaptive control**, where a model of the friction can be developed on-line during normal machine operation and the controller adapts to compensate (Canudas de Wit et al. 1991; Johnson and Lorenz 1992).

To construct a good model of the friction dynamics in a machine tool servo drive, it is useful to have a thorough understanding of the mechanics behind the phenomenon. The survey paper by Armstrong-Hélouvry et al. (1994) contains a detailed look into the mechanics of friction in machines. The authors’ approach is to begin by examining the topography of contact between two sliding surfaces and then to explore how this contact typically behaves under load. From the macro viewpoint two surfaces that share a common radius will have a conformal contact, meaning the apparent area of contact is proportional to the surface dimensions. Conversely, surfaces without a common radius have a nonconformal contact, where the parts deform to create an apparent area of contact, an area that increases with increasing load. However, even apparently smooth surfaces are microscopically rough, thus true contact occurs where asperities come together (see Figure A.1).

As a shear load is applied the asperities will deform, causing microscope displacement, but no sliding velocity. In a lubricated system, this dynamic region (Region I) is known as static friction (or stiction) and presliding displacement, and sometimes referred to as the Dahl effect. Hence, while in the region of static
friction there is no sliding velocity per se, there is a displacement that is nearly linear with applied load, up until a point where breakaway occurs. Region II occurs after breakaway, but at very low sliding velocities. It is characterised by sliding of the solid boundary layer of lubricant that is stuck to the surfaces. Region III, partial fluid lubrication, is the process by which lubricant is drawn into the contact area; however the surfaces are still partially supported by solid-to-solid contact. Moving into this region can be linked to a drop in the frictional force. The fourth and final region (Region IV) is that of full fluid lubrication, where the surfaces are fully supported by the fluid lubricant. In this region the frictional force will typically increase proportionally with increasing sliding velocity. Together the four regions combine to form what is known as the Stribeck curve (see Figure A.2).

The Stribeck curve of Figure A.2 shows friction has a dependence on velocity, so that if there is a change in velocity one would presume an instantaneous change in friction. In fact, there is an observed delay in the change of friction, brought about by the time required for the fluid lubrication drag to reach a new steady state.

One of the first recognised modern attempts to model friction in machines was described by Dahl (1968). It is based on the stress-strain curve from classical mechanics. When two mating surfaces are subjected to rising shear stress, the frictional force increases as the asperities in contact deform.
plied load reaches a critical point, the bonds between asperities will rupture, and the surfaces will begin sliding. The differential equation used to model this behaviour is

$$\frac{dT_f}{d\theta} = \sigma_0 \left( 1 - \frac{T_f}{T_c} \text{sgn}(\omega) \right)^\alpha$$

(A.1)

where $T_f$ is the friction, $\theta$ and $\omega$ are respectively displacement and velocity, $T_c$ is the Coulomb friction, $\sigma_0$ is the stiffness coefficient and $\alpha$ is a parameter that determines the shape of the stress-strain curve. In this model, the friction is only a function of the displacement and the sign of the velocity, hence the friction is only position dependent. To obtain a time domain model, Dahl observed that

$$\frac{dT_f}{dt} = \frac{dT_f}{d\theta} \cdot \frac{d\theta}{dt} = \sigma_0 \omega \left( 1 - \frac{T_f}{T_c} \text{sgn}(\omega) \right)^\alpha$$

(A.2)

This model does not capture the Stribeck effect or stiction. An attempt to extend the Dahl model to include the Stribeck effect and stiction was made by Bliman and Sorine (1991). They replaced the time variable $t$ by a space variable $\lambda$ through

Figure A.2: Stribeck curve.
the transformation

\[ \lambda = \int_0^t |v(\tau)| \, d\tau \]  

(A.3)
hence (A.2) becomes

\[ \frac{dT_f}{d\lambda} = \sigma_0 \text{sgn}(\omega) - \sigma_0 \frac{T_f}{T_c} \]  

(A.4)
which is a linear first-order system if \( \text{sgn}(\omega) \) is regarded as an input. This first-order model does not give stiction. However by extending it to a second-order system, Bliman and Sorine were then able to model stiction. It should be noted that the Bliman-Sorine model does not give the Stribeck effect at steady state velocities; it will only give a transient Stribeck effect after a velocity reversal.

In an effort to combine all the frictional effects identified in their survey paper into a single model Armstrong-Hélouvry et al. (1994) proposed the seven-parameter model, where instantaneous friction is given by

\[
T_f(\theta, \omega, t, t^*) = \begin{cases} 
  k_t \theta & \omega = 0 \\
  T_c + B_m |\omega| + \frac{T_{st}(t^*)}{1 + \left(\frac{\omega}{\omega_s(t - \tau_m)}\right)^2} \text{sgn}(\omega) & \text{otherwise}
\end{cases}
\]  

(A.5)
and the rising static friction (friction level at breakaway) is

\[
T_{st}(t^*) = T_{st,a} + (T_{st,\infty} - T_{st,a}) \left( \frac{t^*}{t^* + \gamma} \right)
\]  

(A.6)
where \( T_f \) is the instantaneous friction, \( \theta \) is the pre-sliding displacement, \( \omega \) is the relative sliding velocity, \( t \) is the time, \( t^* \) is the dwell time (time at zero velocity), \( T_{st} \) is the magnitude of the Stribeck friction (friction at breakaway is \( T_c + T_{st} \)), and \( T_{st,a} \) is the magnitude of the Stribeck friction at the end of the previous sliding period. The seven parameters of the model are: \( k_t \), the tangential stiffness of the static contact; \( T_c \), the Coulomb friction; \( B_m \), the viscous friction coefficient; \( \omega_s \), the characteristic velocity of the Stribeck friction; \( \tau_m \), the time constant of frictional memory; \( T_{st,\infty} \), the magnitude of the Stribeck friction after a long time at rest; and \( \gamma \), the temporal parameter of the rising static friction. To establish values for the
parameters, a series of data collecting experiments are typically required. Each experiment should be tailored to identify a single parameter or group of related parameters.

The LuGre friction model, developed by Canudas de Wit et al. (1995) introduced a different empirical friction model to the one developed the previous year by Armstrong-Hélouvry et al. (1994). The key difference between them is that the LuGre model is continuous, whereas the seven-parameter model has a discontinuity at zero velocity. In the LuGre model the friction interface between two surfaces is thought of as a contact between elastic bristles (see Figure A.3).

\[ \frac{dz}{dt} = \omega - \left| \frac{\omega}{g(\omega)} \right| \]

Figure A.3: Elastic bristles used to develop the LuGre friction model.

The average deflection of the bristles is denoted by \( z \) and is modelled by

\[ \frac{dz}{dt} = \omega - \left| \frac{\omega}{g(\omega)} \right| z \]

where \( \omega \) is the relative velocity between the two surfaces, and the function \( g \) is positive and depends on material properties, surface finish, lubrication and temperature. At steady state velocity, the average rate at which the bristles deflect is zero and hence

\[ z_{ss} = g(\omega) \frac{\omega}{|\omega|} = g(\omega) \text{sgn}(\omega) \]

(A.8)
The friction generated from the bending of the bristles is described as

\[ T_f = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 \omega \]  

(A.9)

where \( \sigma_0, \sigma_1 \) and \( \sigma_2 \) are parameters. The function \( \sigma_0g(\omega) + \sigma_2\omega \) can be determined by measuring the steady-state friction force at constant velocity. The proposed parameterisation of \( g \) to describe the Stribeck effect is

\[ \sigma_0g(\omega) = T_c + (T_s - T_c)e^{-\left(\frac{\omega}{\omega_s}\right)^2} \]  

(A.10)

where \( T_c \) is the Coulomb friction, \( T_s \) is the level of the static friction force, and \( \omega_s \) is the Stribeck velocity. Given (A.8), (A.9) and (A.10), the relationship between steady state velocity and friction is given by

\[ T_{f,ss} = \sigma_0 z_{ss} + \sigma_1 \frac{dz_{ss}}{dt} + \sigma_2 \omega = \left( \sigma_0g(\omega) + \sigma_2|\omega| \right) \text{sgn}(v) \]  

(A.11)

Note that (A.11) is only applicable for constant velocity. When the velocity is not constant (A.9) must be used. Thus the complete model has six parameters, \( \sigma_0, \sigma_1, \sigma_2, T_c, T_s \) and \( \omega_s \). Note also that if \( g(\omega) = T_c/\sigma_0 \) and \( \sigma_1 = \sigma_2 = 0 \), then the model given by (A.9) becomes the Dahl model given by (A.2), assuming \( \alpha = 1 \).

Olsson et al. (1998) provide a good summary of the models described thus far and include a comprehensive comparison between the Bliman-Sorine model (Bliman and Sorine 1991, 1993, 1995) and the LuGre model (Canudas de Wit et al. 1995), which are both extensions of the Dahl model (Dahl 1968). A key issue identified by Olsson et al. is the problem of using equations such as (A.5) and (A.11), which have a discontinuity at zero velocity. A remedy for this is found in the model presented by Karnopp (1985). It was developed to overcome the problems with zero velocity detection and to avoid the simulation chattering between sticking and slipping. The model defines a small zero velocity interval, \( |\omega| < \varepsilon \). For velocities within this interval, the internal state of the system (the velocity) may change and be non-zero, but the output is maintained at zero by a dead-zone.
In conclusion, it has been shown that friction modelling is a well established field. Some of the most comprehensive models were developed in the 1990s. The majority of these are suitable for use in modelling friction which exists in typical machine tool servo drive axes. The choice of which one to use, will depend in part on which is easiest to parameterise via suitable system identification techniques.

## A.2 Clarke-Park Transformations

The majority of machine tool servo drives use Permanent Magnet Synchronous Motors (PMSM). The typical method for controlling such motors is vector control, where the $dq$ coordinate frame rotates with the rotor position. The transform from the stationary armature coordinate frame $abc$ to $dq$ is known as the Clarke-Park Transformation (Lyshevski 2000). Figure A.4 depicts how the two reference frames are related.

![Figure A.4: Clarke-Park Transformation](image)

The transformation from $x_{abc}$ to $x_{\alpha\beta\theta}$ is known as the Clarke transformation, and
CLARKE-PARK TRANSFORMATIONS

is given by

\[
\begin{bmatrix}
x_\alpha \\
x_\beta \\
x_0
\end{bmatrix} = \frac{2}{3} \begin{bmatrix}
1 & -\cos\left(\frac{\pi}{3}\right) & -\cos\left(\frac{\pi}{3}\right) \\
0 & \sin\left(\frac{\pi}{3}\right) & -\sin\left(\frac{\pi}{3}\right) \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix} \begin{bmatrix}
x_a \\
x_b \\
x_c
\end{bmatrix}
\]

(A.12)

which for a balanced three-phase system, where \(x_a + x_b + x_c = 0\), is

\[
\begin{bmatrix}
x_\alpha \\
x_\beta \\
x_0
\end{bmatrix} = \begin{bmatrix}
\frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\
0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{bmatrix} \begin{bmatrix}
x_a \\
x_b \\
x_c
\end{bmatrix}
\]

(A.13)

The inverse Clarke transformation is

\[
\begin{bmatrix}
x_a \\
x_b \\
x_c
\end{bmatrix} = \frac{3}{2} \begin{bmatrix}
1 & -\cos\left(\frac{\pi}{3}\right) & -\cos\left(\frac{\pi}{3}\right) \\
0 & \sin\left(\frac{\pi}{3}\right) & -\sin\left(\frac{\pi}{3}\right) \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}^{-1} \begin{bmatrix}
x_\alpha \\
x_\beta \\
x_0
\end{bmatrix}
\]

(A.15)

\[
\begin{bmatrix}
x_a \\
x_b \\
x_c
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 1 \\
-\cos\left(\frac{\pi}{3}\right) & \sin\left(\frac{\pi}{3}\right) & 1 \\
-\cos\left(\frac{\pi}{3}\right) & -\sin\left(\frac{\pi}{3}\right) & 1
\end{bmatrix} \begin{bmatrix}
x_\alpha \\
x_\beta \\
x_0
\end{bmatrix}
\]

(A.16)

\[
\begin{bmatrix}
x_a \\
x_b \\
x_c
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 1 \\
-\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\
-\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1
\end{bmatrix} \begin{bmatrix}
x_\alpha \\
x_\beta \\
x_0
\end{bmatrix}
\]

(A.17)
which for a balanced three-phase system is

\[
\begin{bmatrix}
  x_a \\
  x_c \\
\end{bmatrix} = \begin{bmatrix}
  1 & 0 \\
  -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\
\end{bmatrix} \begin{bmatrix}
  x_\alpha \\
  x_\beta \\
\end{bmatrix}
\]  
(A.18a)

\[x_b = -(x_a + x_c)\]  
(A.18b)

The transformation from \(x_{\alpha\beta\theta}\) to \(x_{dq\theta}\) is known as the Park transformation, and is given by

\[
\begin{bmatrix}
  x_d \\
  x_q \\
  x_0 \\
\end{bmatrix} = \begin{bmatrix}
  \sin \theta_e & -\cos \theta_e & 0 \\
  \cos \theta_e & \sin \theta_e & 0 \\
  0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
  x_\alpha \\
  x_\beta \\
  x_\theta \\
\end{bmatrix}
\]  
(A.19)

The inverse Park transformation is

\[
\begin{bmatrix}
  x_\alpha \\
  x_\beta \\
  x_\theta \\
\end{bmatrix} = \begin{bmatrix}
  \sin \theta_e & \cos \theta_e & 0 \\
  -\cos \theta_e & \sin \theta_e & 0 \\
  0 & 0 & 1 \\
\end{bmatrix}^{-1} \begin{bmatrix}
  x_d \\
  x_q \\
  x_0 \\
\end{bmatrix}
\]  
(A.20)

The transformation from \(x_{abc}\) to \(x_{dq\theta}\) is known as the Clarke-Park transformation, and is derived by combining (A.12) and (A.19) to produce

\[
\begin{bmatrix}
  x_d \\
  x_q \\
  x_0 \\
\end{bmatrix} = \frac{2}{3} \begin{bmatrix}
  \sin \theta_e & -\cos \theta_e & 0 \\
  \cos \theta_e & \sin \theta_e & 0 \\
  0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
  1 - \cos \left(\frac{\pi}{3}\right) - \cos \left(\frac{\pi}{3}\right) \\
  \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\end{bmatrix} \begin{bmatrix}
  x_a \\
  x_b \\
  x_c \\
\end{bmatrix}
\]  
(A.22)
which for a balanced three-phase system is

\[
\begin{bmatrix}
  x_d \\
  x_q
\end{bmatrix} = \frac{2}{\sqrt{3}} \begin{bmatrix}
  \sin \left( \theta_e + \frac{\pi}{6} \right) & - \sin \left( \theta_e - \frac{\pi}{2} \right) \\
  \cos \left( \theta_e + \frac{\pi}{6} \right) & - \cos \left( \theta_e - \frac{\pi}{2} \right)
\end{bmatrix}
\begin{bmatrix}
  x_a \\
  x_c
\end{bmatrix}
\] (A.23)

The inverse Clarke-Park transformation is

\[
\begin{bmatrix}
  x_a \\
  x_b \\
  x_c
\end{bmatrix} = \frac{3}{2} \begin{bmatrix}
  \sin \theta_e & \sin \left( \theta_e - \frac{2\pi}{3} \right) & \sin \left( \theta_e + \frac{2\pi}{3} \right) \\
  \cos \theta_e & \cos \left( \theta_e - \frac{2\pi}{3} \right) & \cos \left( \theta_e + \frac{2\pi}{3} \right) \\
  \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}
\begin{bmatrix}
  x_d \\
  x_q \\
  x_0
\end{bmatrix}
\] (A.24)

\[
\begin{bmatrix}
  x_a \\
  x_b \\
  x_c
\end{bmatrix} = \begin{bmatrix}
  \sin \theta_e & \cos \theta_e & 1 \\
  \sin \left( \theta_e - \frac{2\pi}{3} \right) & \cos \left( \theta_e - \frac{2\pi}{3} \right) & 1 \\
  \sin \left( \theta_e + \frac{2\pi}{3} \right) & \cos \left( \theta_e + \frac{2\pi}{3} \right) & 1
\end{bmatrix}
\begin{bmatrix}
  x_d \\
  x_q \\
  x_0
\end{bmatrix}
\] (A.25)

which for a balanced three-phase system is

\[
\begin{bmatrix}
  x_a \\
  x_c
\end{bmatrix} = \begin{bmatrix}
  \cos \left( \theta_e - \frac{\pi}{2} \right) & - \sin \left( \theta_e - \frac{\pi}{2} \right) \\
  \cos \left( \theta_e + \frac{\pi}{6} \right) & - \sin \left( \theta_e + \frac{\pi}{6} \right)
\end{bmatrix}
\begin{bmatrix}
  x_d \\
  x_q
\end{bmatrix}
\] (A.26a)

\[
x_b = -(x_a + x_c)
\] (A.26b)

A.3 Cascaded State-Space Systems

Figure A.5 shows the arrangement of two cascaded state-space systems with identical sample-rates. Given that the output of the first system, \( G_\alpha \), becomes the input to the second system, \( G_\beta \), the goal is to combine the two systems into a single state-space representation, while preserving the original states. This can be achieved via matrix augmentation.

Start by defining two LTI systems in state-space form as

\[
x_\alpha(k + 1) = A_\alpha x_\alpha(k) + B_\alpha u_\alpha(k)
\] (A.27a)

\[
y_\alpha(k) = C_\alpha x_\alpha(k) + D_\alpha u_\alpha(k)
\] (A.27b)
and

\[
\begin{align*}
    x_\beta(k + 1) &= A_\beta x_\beta(k) + B_\beta y_\beta(k) \\
    y_\beta(k) &= C_\beta x_\beta(k) + D_\beta u_\beta(k)
\end{align*}
\]

where the relationship between them is

\[
u_\beta(k) = y_\alpha(k)
\]

Restating (A.27a) and substituting (A.27b) into (A.28) produces

\[
\begin{align*}
    x_\alpha(k + 1) &= A_\alpha x_\alpha(k) + B_\alpha u_\alpha(k) \\
    x_\beta(k + 1) &= A_\beta x_\beta(k) + B_\beta \left( C_\alpha x_\alpha(k) + D_\alpha u_\alpha(k) \right) \\
    y_\beta(k) &= C_\beta x_\beta(k) + D_\beta \left( C_\alpha x_\alpha(k) + D_\alpha u_\alpha(k) \right)
\end{align*}
\]

Now defining the overall cascaded system as

\[
\begin{align*}
    x_{\alpha\beta}(k + 1) &= A_{\alpha\beta} x_{\alpha\beta}(k) + B_{\alpha\beta} u_{\alpha\beta}(k) \\
    y_{\alpha\beta}(k) &= C_{\alpha\beta} x_{\alpha\beta}(k) + D_{\alpha\beta} u_{\alpha\beta}(k)
\end{align*}
\]

where the input and output are

\[
\begin{align*}
    u_{\alpha\beta}(k) &= u_\alpha(k) \\
    y_{\alpha\beta}(k) &= y_\beta(k)
\end{align*}
\]

the state vector is

\[
x_{\alpha\beta}(k) = \begin{bmatrix} x_\alpha(k) \\ x_\beta(k) \end{bmatrix}
\]
and from (A.30), the system matrices are

\[
A_{\alpha\beta} = \begin{bmatrix} A_\alpha & 0 \\ B_\beta C_\alpha & A_\beta \end{bmatrix} \tag{A.34a}
\]

\[
B_{\alpha\beta} = \begin{bmatrix} B_\alpha \\ B_\beta D_\alpha \end{bmatrix} \tag{A.34b}
\]

\[
C_{\alpha\beta} = \begin{bmatrix} D_\beta C_\alpha & C_\beta \end{bmatrix} \tag{A.34c}
\]

\[
D_{\alpha\beta} = D_\beta D_\alpha \tag{A.34d}
\]

### A.4 System Identification Techniques

The section looks at methods for populating the models presented in Section 2.2. One approach is to use first principles, that is, analyse the physical features of the components that make up the servo drive axis. In a complex machine tool this is a near impossible task, hence researchers have spent considerable effort to develop methods for measuring and approximating the parameters. This area is usually referred to as system identification.

Erkorkmaz and Altintas (2001a) propose a method to identify plant inertia as well as frictional characteristics. There are two stages to the identification process. The first involves collecting a set of output and commanded velocity samples and using a least squares objective function to estimate values for inertia and viscous damping. The excitation signal is a series of step inputs. The second stage seeks to identify the friction parameters, as well as fine-tune the viscous damping estimate. The frictional force is broken into the different regions of the Stribeck curve (see Figure A.2). The test involves jogging the axis back and forth at various speeds using a pole placement controller. The frictional forces acting on the axis are modelled as a disturbance, and estimated using a Kalman Filter designed for the state space model representing the system. The results presented show how friction torque compensation in feedforward, using the identified sys-
BACKGROUND THEORY AND ADDITIONAL LITERATURE

tem parameters, is able to significantly reduce the tracking error for a circular reference trajectory.

Erkorkmaz and Wong (2007) further developed this work with a process to identify the drive system as a whole, including the feed mechanism, motor, amplifier, and the control law. The method begins by developing a generic model for typical machine tool drives. The model contains eight parameters that must be determined. Constraints on the parameters are defined so that stability margins are maintained. The system is then subjected to a series of motions as defined by a sequence of G-codes\(^1\). From the set of commanded and measured dynamics, a set of simultaneous equations incorporating the eight unknown parameters can be solved. The set of simultaneous equations is developed around a least-squares objective function. Both simulation and experimental results show that due to limited excitation delivered by the CNC interpolator, convergence of the parameters is not guaranteed. The method does however capture the key dynamic features of the actual drives.

In the paper by del Re et al. (1996), the authors provide a comparison between design based system identification and black-box style identification. They conclude that while the black-box technique can generate models of very large order (when used on fast machine tools where structural deformation becomes increasingly important), the models typically have superior accuracy to that of design based system identification. However, for machine tools where models with large numbers of structural modes are less critical, the results from the design based approach are found to be equivalent to that of black-box. Given the design based approach, having a model which can be interpreted in terms of the physical system (i.e. inertias, stiffnesses and damping) can be of benefit when analysing the dynamics to be controlled.

Two other pieces of work looking at system identification by Lew et al. (2006)

\(^1\)G-code is an entry in a part program used to control NC and CNC machine tools.
and Chang (2006), make use of Genetic Algorithms (GA). This type of system identification uses the Darwinian strategy of evolution by *survival of the fittest*, which was described in Section 2.1.3 under the section on *Global Optimality*. The results presented show that for system requiring very large and complex models, good results can be achieved via the use of GA. However, the computation time required to obtain a model with acceptable accuracy can be significantly longer than a simple least-squares fit of a design based model. Furthermore, the approach proposed by Lew et al., results in a black-box style model which has the potential drawbacks outlined in the previous paragraph.

There are several techniques available to populate the derived parametric models. As stated previously, one approach is to use first principles, that is, analyse the physical features of the components represented by the models. In a complex machine tool servo drive axis this is a near impossible task. Another technique is to use computer aided analysis tools, such as FEA. This generally requires sufficiently detailed 3D CAD models, and also accurate knowledge of the material properties, both of which may not be available. A third, and well utilised method, is to estimate the parameters using measured data obtained from experimentation.

Figure A.6 shows the general structure of both an open-loop and a closed-loop system. The elements $G$ and $H$ are unknown. There are several well developed algorithms available to estimate the model parameters from measured data – two will be considered here. The first, presented in Section A.4.1, is a frequency domain method, which attempts to generate frequency response data representing the transfer function of the unknown system (Ljung 1999). This data is then used to estimate values for the unknown parameters using a least-squares fit. The second approach, presented in Section A.4.2, is known as the Prediction-Error identification Method (PEM) (Ljung 1999).
A.4.1 Power Spectrum Technique

With reference to Figure A.6, an estimate of the transfer function from input $u$ to output $y^*$, where the output is corrupted by measurement noise $w$, is given by (Ljung 1999)

$$\hat{G}_{uy} = G_{uy^*} = \frac{S_{uy^*}}{S_{uu}} + \frac{S_{uw}}{S_{uu}}$$  \hfill (A.35)

where $S_{uy^*}$ is the Cross Power Spectrum (CPS) of the input and the output, $S_{uw}$ is the CPS of the input and the measurement noise, and $S_{uu}$ is the Auto Power Spectrum (APS) of the input. The CPS is defined as

$$S_{xz} = \sum_{\tau=-\infty}^{\infty} \left( \frac{1}{K} \sum_{k=1}^{K} x(k) z(k - \tau) \right) e^{-j\tau\omega}$$  \hfill (A.36)

and the APS is defined as

$$S_{xx} = \sum_{\tau=-\infty}^{\infty} \left( \frac{1}{K} \sum_{k=1}^{K} x(k) x(k - \tau) \right) e^{-j\tau\omega}$$  \hfill (A.37)

where $K$ is the number of samples in the data set. If the input $u$ and the measurement noise $w$ are uncorrelated, then $S_{uw} = 0$ and (A.35) becomes

$$\hat{G}_{uy} = \frac{S_{uy^*}}{S_{uu}}$$  \hfill (A.38)

This represents a nonparametric model of the system to be identified. Unfortunately, experiments conducted to generate measurement data for system identification are typically carried out in closed-loop, hence the assumption of no correlation between the input and the measurement noise is invalid.
A solution to the difficulty resulting from correlation between the input and the measurement noise in closed-loop operation is found in work by Akaike (1967). Given that
\[ G_{uy} = \mathcal{Z}\{g(k)\} = \sum_{k=0}^{\infty} g(k) z^{-k} \] (A.39)
then the actual plant output is found via a discrete convolution with the control input \( u \), as
\[ y(k) = \sum_{\tau=-\infty}^{\infty} u(k - \tau) g(\tau) \] (A.40)
The measured plant output is then
\[ y^*(k) = y(k) + w(k) \] (A.41)
Taking the cross-correlation of the reference signal \( r \), and the measured plant output \( y^* \), gives
\[ R_{ry^*}(\sigma) = \Gamma\{r(k)y^*(k + \sigma)\} \]
\[ = \Gamma\{r(k)y(k + \sigma) + r(k)w(k + \sigma)\} \]
\[ = \Gamma\{r(k)y(k + \sigma)\} \] (A.42)
since \( r \) and \( w \) are uncorrelated. Substituting (A.40) into (A.42) gives
\[ R_{ry^*}(\sigma) = \Gamma\left\{r(k) \sum_{\tau=-\infty}^{\infty} u(k + \sigma - \tau) g(\tau)\right\} \]
\[ = \sum_{\tau=-\infty}^{\infty} \Gamma\{r(k)u(k + \sigma - \tau)\} g(\tau) \]
\[ = \sum_{\tau=-\infty}^{\infty} R_{ru}(\sigma - \tau) g(\tau) \] (A.43)
The cross-spectrum is then
\[ S_{ry^*} = \mathcal{Z}\{R_{ry^*}(\sigma)\} \]
\[ = \sum_{\sigma=-\infty}^{\infty} R_{ry^*}(\sigma) z^{-\sigma} \]
\[ = \sum_{\sigma=-\infty}^{\infty} \sum_{\tau=-\infty}^{\infty} R_{ru}(\sigma - \tau) g(\tau) z^{-\sigma} \]
\[ = \sum_{\tau=-\infty}^{\infty} g(\tau) z^{-\tau} \sum_{\sigma=-\infty}^{\infty} R_{ru}(\sigma - \tau) z^{-(\sigma-\tau)} \] (A.44)
BACKGROUND THEORY AND ADDITIONAL LITERATURE

Using $\gamma = \sigma - \tau$ gives

$$
S_{ry*} = \sum_{\tau=-\infty}^{\infty} g(\tau) z^{-\tau} \sum_{\gamma=-\infty}^{\infty} R_{ru}(\gamma) z^{-\gamma}
$$

(A.45)

$$
= G_{uy} S_{ru}
$$

Therefore, $G_{uy}$ in the block diagram on the right of Figure A.6 can be estimated via

$$
\hat{G}_{uy} = \frac{S_{ry*}}{S_{ru}}
$$

(A.46)

where $S_{ry*}$ is the CPS from the reference signal to the system output, and $S_{ru}$ is the CPS from the reference signal to the output of the controller block ($R_{eu}$). In the typical servo drive system all three measurements required for this calculation ($r$, $u$ and $y^*$) are able to be sampled and logged.

To improve the results from the least-squares fit, the frequency response data is filtered (weighted) so that data with a high coherence between input and output has a larger effect on the identified model. The coherence, or more specifically the magnitude squared coherence estimate, is calculated using

$$
C_{xz} = \frac{|S_{xz}|^2}{S_{xx} S_{zz}}
$$

(A.47)

A.4.2 Prediction-Error Identification Method

Again referring to Figure A.6 for the PEM method, the basic description for the LTI discrete-time model is

$$
y^*(n) = w(n) + \sum_{k=1}^{\infty} g(k, \rho) u(n - k)
$$

(A.48)

where $g$ is the impulse response of the system and $\rho$ is a parameter vector, which is to be determined. We can define the noise which is added to the output as

$$
w(n) = \sum_{k=1}^{\infty} h(k, \rho) v(n - k)
$$

(A.49)

where $h$ is the impulse response of the noise model and $v$ is white noise. Furthermore we impose the condition that (A.49) should be invertible, such that we can
compute $v$ as

$$v(n) = \sum_{k=1}^{\infty} \hat{h}(k, \rho) w(n - k)$$  \hspace{1cm} (A.50)

The goal of PEM is to find the set of parameters which produce a model that best represents the actual system. This is achieved by predicting future outputs of the system using data from the past

$$Z^K = [u(1), y^*(1), u(2), y^*(2), \cdots, u(K), y^*(K)]$$  \hspace{1cm} (A.51)

and then calculating the error between the measured, $y^*$, and the predicted, $\hat{y}$, output via

$$\varepsilon(n, \rho) = y^*(n) - \hat{y}(n | \rho)$$  \hspace{1cm} (A.52)

The set of parameters which produce the smallest error is considered the closest match to those of the actual system. The size of the error vector could be measured using any norm, but a popular choice is

$$V(\rho, Z^K) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{2} \varepsilon(k, \rho)^2$$  \hspace{1cm} (A.53)

where $N$ is the number of data samples.

The predictor used to predict future outputs can be a linear filter. One possibility, as given by Ljung (1999), is

$$\hat{y}(n | \rho) = \sum_{k=1}^{\infty} \ell(k, \rho) u(n - k) + \sum_{k=1}^{\infty} -\tilde{h}(k, \rho) y^*(n - k)$$  \hspace{1cm} (A.54)

where $\ell$ and $-\tilde{h}$ are defined by

$$\frac{G_{uy}(z, \rho)}{H_{vw}(z, \rho)} = \sum_{k=1}^{\infty} \ell(k, \rho) z^{-k}$$  \hspace{1cm} (A.55)

and

$$\frac{H_{vw}(z, \rho) - 1}{H_{vw}(z, \rho)} = \sum_{k=1}^{\infty} -\tilde{h}(k, \rho) z^{-k}$$  \hspace{1cm} (A.56)

where $G_{uy}$ and $H_{vw}$ are the discrete Fourier-transforms of $g$ and $h$. Since in practice we will only have access to data over the interval $[1, n-1]$, and not $[-\infty, n-1]$, we can approximate the predictor function, (A.54) by

$$\hat{y}(n | \rho) \approx \sum_{k=1}^{n-1} \ell(k, \rho) u(n - k) + \sum_{k=1}^{n-1} -\tilde{h}(k, \rho) y^*(n - k)$$  \hspace{1cm} (A.57)
and modify (A.53) so as not to predict the first output

\[ V(\rho, Z^K) = \frac{1}{K} \sum_{k=2}^{K} \frac{1}{2} \varepsilon(k, \rho)^2 \]  \hspace{1cm} (A.58)

For large enough values of \( K \), a satisfactory solution will result.

### A.4.3 Current Loop Identification via Power Spectrum Technique

The initial phase for developing a model of the Current Loop was to collect data from experiments conducted on an ANCA RX7 TCG machine. As outlined in Chapter 3.1, ANCA has established facilities for experiments of this nature. The input and output data sets required for the system identification of the Current Loop are respectively, the filtered motor current reference, \( I_f \) and the \( q \)-axis current feedback, \( I_q \). The input signal to the closed-loop system, applied as the motor velocity reference, \( \Omega_r \) is a chirp signal of the form

\[ \Omega_r(k) = A \sin \left( 2\pi f(k)kT \right) + B \]  \hspace{1cm} (A.59)

where \( A \) is the amplitude of the signal, \( B \) is a constant offset so as to avoid non-linear frictional effects around zero velocity, and \( f(k) \) is the instantaneous frequency of the signal and is defined as

\[ f(k) = \frac{k}{4KT} \]  \hspace{1cm} (A.60)

where \( K \) is the number of samples in the data set. Using (3.14) ensures that the input signal will excite across the full frequency spectrum up to the Nyquist frequency \( \left( \frac{1}{2T} \right) \). The data logging system on the ANCA Servo Drives allows four variables, each of 2048 data points to be logged simultaneously; hence \( K \) is 2048.

With the aim of pushing the system as hard as possible without reaching any saturation limits (for example, the motor current limit), increasing values of \( A \) in (3.14) were tried. The parameter \( B \) was chosen so that the motor velocity reference signal was always positive, hence \( B > A \). Once a set of parameters...
for (3.14) had been selected, the experiment was conducted 10 times so that the signals could be averaged to reduce measurement noise.

The collected data, with $A = 30$RPM and $B = 34$RPM, was then analysed using the MATLAB software suite, which provides most of the required analysis tools needed. The `tfestimate` function, which implements (A.38), was used to generate the frequency response data. The input to this function was the 10 individual data sets cascaded together to produce a single data record for each of $I_f$ and $I_q$. The two data records were further preprocessed by removing the bias currents, to produce zero mean signals. Also, due to the fact that the chirp signal is not a statistically stationary signal, the two data records were windowed with a rectangular window of length $K$ and with no overlap. The resulting frequency response data, $G_{i_f i_q}$ can be seen in Figure A.11.

The next step is to find the parameters that will fit (3.20) to the frequency response data. For this task the MATLAB function `invfreqz` was used. This function implements a least-squares fit of a transfer function to frequency response data. The user must specify the number of poles and zeros of the chosen model, in this case two and one respectively. As briefly outlined before, to improve the results from the least-squares fit, the frequency response data was filtered via a measure of coherence. Figure A.7 shows the magnitude squared coherence estimate of the input, $I_f$, to the output, $I_q$.

It can be seen that there is high coherence for input-output data comprising frequencies up to about 1kHz. Hence only frequency response data up to this frequency was included in the least-squares fit. Formally, the scheme is to only include data which has a corresponding coherence great than or equal to 0.5. These points are marked blue in Figure A.8.
Figure A.7: Magnitude Squared Coherence Estimate ($I_f$ to $I_q$)

Figure A.8: Bode Plot ($I_f$ to $I_q$)
A.4.4 Mechanical Plant Parameter Identification via Power Spectrum Technique

This section uses the power spectrum approach introduced in Section A.4.1 to identify the mechanical plant using data from the motor current reference, $I_r$, to the motor velocity, $\Omega_m$.

The process to be followed to identify the mechanical plant is similar to that of the current loop. That is, the same excitation signal was used, and the frequency response data was generated using `tfestimate`, which was subsequently filtered based on a measurement of input-output coherence. But instead of a non-parametric model, the aim is to fit a parametric model, where the parameters all have physical meaning. Using `invfreqz` consistently produced a non-parametric model, which when parameterised, produced non-physical parameters, for example, a negative moment of inertia. So instead of `invfreqz`, a customised least-squares fit was developed. The new method uses the MATLAB function `lsqcurvefit`, which implements an algorithm to fit the parameters of a user-defined function to data, in this case, fit a parameterised transfer function model of $I_r$ to $\Omega_m$, using the frequency response data just generated. This approach is chosen since it makes use of information already known about the system. It is thus a grey box model technique.

The continuous-time state-space representation of the mechanical plant is
given by

\[
\begin{bmatrix}
\ddot{\theta}_m \\
\ddot{\theta}_w \\
\dot{\theta}_m \\
\dot{\theta}_w
\end{bmatrix}
= 
\begin{bmatrix}
\frac{-B_m}{J_m} & \frac{c_w}{J_m} & \frac{-k_w}{J_m} & \frac{k_w}{J_m} \\
\frac{c_w}{J_w} & \frac{-c_w}{J_w} & \frac{k_w}{J_w} & \frac{-k_w}{J_w} \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\ddot{\theta}_m \\
\ddot{\theta}_w \\
\dot{\theta}_m \\
\dot{\theta}_w
\end{bmatrix}
+ 
\begin{bmatrix}
K_t \\
0 \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
i_q
\end{bmatrix}
\quad (A.61a)
\]

\[
\omega_m = \dot{\theta}_m 
\quad (A.61b)
\]

As before, we wish to filter the frequency response data via a measurement of input-output coherence. The magnitude-squared coherence estimate plot for input, \( I_r \) to output, \( \Omega_m \) is shown in Figure A.10. The coherence between input and output is high up to about 800Hz, except for the region between 150 to 300Hz, which as will be seen shortly, corresponds to the location of the fundamental anti-resonance. This drop in coherence is believed to be a result of the very low signal power associated with the anti-resonance.

The frequency response data is shown in Figure A.11, where the data with high coherence (greater than or equal to 0.7) is shown by the blue dots, and the low coherence data by green dots. Again, this was generated using zero-mean data cascaded together from 10 individual tests.

The \textit{lsqcurvefit} function requires the user to select an initial value for each of the unknown parameters. The initial value chosen for the torque constant, \( K_t \) is 3.51 N.m/A. This comes from the ANCA specification for the motor. The moments of inertia initial values were determined using SolidWorks and a 3D CAD model of the axis and workpiece. The values were calculated to be 0.0172 kg·m\(^2\) for the motor and 0.0528 kg·m\(^2\) for the workpiece. The initial value for the viscous friction, \( B_m \) was derived from the system identification work done on the friction model in Section 3.5.3. The value is 0.0500 N·m/(rad/s). The torsional damping parameter, \( c_w \) is the hardest to predict since it is associated with energy
dissipation, which in turn is dependent on material properties and the operating environment. For this parameter an arbitrary value of 0.6 N·m/(rad/s) was se-
lected. The final parameter is the torsional stiffness, $k_w$, which was determined using

$$k_w = \frac{I_z G}{l_c}$$  \hspace{1cm} (A.62)

where $G$ is the coupling material’s shear modulus of elasticity, $l_c$ is the length of the coupling, and $I_z$ is the polar moment of inertia, which for a cylindrical coupling is

$$I_z = \frac{\pi d^4}{32}$$  \hspace{1cm} (A.63)

where $d_c$ is the diameter. Using this formula a value of 78 528 N·m/rad was calculated.

Having defined a set of initial values, it is now possible to use \texttt{lsqcurvefit} to arrive at the optimal set which results in a transfer function that best matches the frequency response data. The resultant mechanical plant parameters are shown in Table 3.2, and the red line in Figure A.11 shows a comparison between the frequency response data and the transfer function using these parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initial Value</th>
<th>Optimal Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_t$</td>
<td>3.5100</td>
<td>3.5801</td>
<td>N·m/A</td>
</tr>
<tr>
<td>$J_m$</td>
<td>0.0172</td>
<td>0.0127</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_w$</td>
<td>0.0528</td>
<td>0.0500</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$B_m$</td>
<td>0.0500</td>
<td>0.0264</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$c_w$</td>
<td>0.6000</td>
<td>0.6309</td>
<td>N·m/(rad/s)</td>
</tr>
<tr>
<td>$k_w$</td>
<td>78 528</td>
<td>78 603</td>
<td>N·m/rad</td>
</tr>
</tbody>
</table>

Table A.1: Mechanical Plant Parameters

Using the parameters in Table A.1, it is possible to use the MATLAB function \texttt{lsim} to check that the simulated response approximates the measured output. Figure 3.19 shows the measured zero mean current reference input and zero mean velocity output in blue, and the simulated velocity output in red. While not an exact match, the simulated response does include the main characteristics of the measured output. Specifically, the constant high frequency oscillation, the large excitation at around 0.12 s, and the general trajectory.
LINEAR QUADRATIC CONTROL FORMULATION

B.1 LINEAR QUADRATIC CONTROL DESIGN

The system describing the combined dynamics of the fast-rate elements (50 µs) shown in Figure 3.8 and of the mechanical plant (Section 3.5), can be represented by the discrete-time (T = 250 µs) linear time-invariant state equations

\[ x(k+1) = Ax(k) + Bi_{LQ}(k) \]  
\[ \theta_m(k) = Cx(k) + Di_{LQ}(k) \]

(B.1a)

(B.1b)

Here \( x \in \mathbb{R}^{m \times 1} \) is the state vector (where \( m \) is the order of the system), \( \theta_m \) is the output, and \( i_{LQ} \) is the control input. The system matrices are \( A \in \mathbb{R}^{m \times m}, B \in \mathbb{R}^{m \times 1}, C \in \mathbb{R}^{1 \times m} \) and \( D \in \mathbb{R} \). As was shown in Section 3.5.3, nonlinear friction is known to exist within the mechanical elements of the plant, however as will be shown shortly, this control technique requires a linear plant model to obtain the optimal solution. Hence at this point nonlinear friction is ignored, but a suitable technique to compensate for its effects is introduced in Section B.2.

In LQ design, the control input is a linear combination of the state vector, that is

\[ i_r(k) = i_{LQ}(k) = -Kx(k). \]  

(B.2)

Hence, the first step in LQ design is to find \( K \in \mathbb{R}^{1 \times m} \). Note (B.1) can be derived using the methods presented in Section 4.1.2.

Substituting (B.2) into (B.1a) yields

\[ x(k+1) = Ax(k) - BKx(k). \]  

(B.3)
The characteristic equation of this closed-loop system is given by

\[ \det [zI - (A - BK)] = 0, \quad (B.4) \]

which will yield an \( m \)th-order polynomial in \( z \) containing the to be determined elements of \( K \). The roots of this polynomial are the closed-loop poles of the system. Hence if a set of desired pole locations are known then it is possible to choose the elements of \( K \) such that the roots of (B.4) match. A method for achieving this is presented in Section B.3.

The previous derivations assume that the plant model under control is linear. As highlighted in the Literature Review (see Section 2.2.2), nonlinear friction can be a problem for machine tool axes. Specifically when it comes to position regulation (hunting), tracking at low velocities (stick-slip) and velocity reversal (axis pause). Since nonlinear friction is known to influence the motion of the \( A \)-axis, and its characteristics have been identified and modelled (see Section 3.5.3), it is possible to directly account for its effects in the controller design. This will in essence act to linearise the plant model, and is presented in Section B.2.

Up to this point the system as designed is that of a regulator. That is, the controller will attempt to maintain the states of the system at identically zero. Since (B.1) represents an axis of a machine tool, some of its states must follow defined trajectories. Hence in Section B.4 a reference signal (the path profile) is introduced into the control architecture. Furthermore, (B.2) assumes that all the state variables are available. This is not typically true, as sensing all states is likely to be both impractical and costly. To overcome this limitation a state estimator is required. This is the subject of Section B.5. To improve the controller’s robustness to modelling uncertainties and the influence of unknown disturbances, an integrator is added to the architecture in Section B.6.
B.2 PLANT MODEL LINEARISATION

To linearise the plant model, feedback linearisation is used. Control input to specifically deal with the effects of nonlinear friction is added to the input for the LQ controller. Under this scheme, the current reference is

\[ i_r(k) = i_{LQ}(k) + i_f(k), \]  

where \( i_{LQ} \) is the control input from the LQ controller (B.2) and

\[ i_f(k) = \begin{cases} K_f \text{sgn} (\dot{\theta}_r(k)) & \dot{\theta}_r(k) \neq 0 \\ K_f \text{sgn} (\theta_r(k) - \theta_m(k)) & \text{otherwise.} \end{cases} \]

Here \( K_f \) is chosen so that \( i_f \) is equivalent to the motor current required to compensate for the Coulomb friction acting on the axis, that is

\[ K_f = \frac{T_c}{K_t}, \]

where \( T_c \) is the Coulomb friction identified in Section 3.5.3 and \( K_t \) is the motor torque constant. It is reasonable to only consider Coulomb friction here, since viscous friction is included in the linear plant model and hence will be dealt with by the LQ controller. Likewise, the small additional torque due to stiction is left for the integrator to manage (see Section B.6).

Since the actuators have a limited operating range, it is necessary to deal with actuator saturation. For added flexibility, the magnitude of the current limit (in effect the actuator limits) can be set dynamically to different levels for each direction of motion. This is useful when it is desirable to restrict the torque produced by the motor when acting in a specific direction. The parameters

\[ 0 \leq i_{\max}^+(k) \leq i_{\max} \quad \text{and} \quad 0 \leq i_{\max}^-(k) \leq i_{\max} \]

define the magnitude of the maximum allowable current in the positive and negative directions, respectively. Hence a restriction on \( i_f \) is made as

\[ -i_{\max}^-(k) - i_{LQ}(k) \leq i_f(k) \leq i_{\max}^+(k) - i_{LQ}(k). \]
LINEAR QUADRATIC CONTROL FORMULATION

B.3 LINEAR QUADRATIC REGULATOR

As highlighted in Section B.1, it is possible to design a feedback controller based on a LQ model of the plant. An effective and widely used technique is the Linear Quadratic (LQ) controller. In LQ control the challenge is to find the control law (B.2), such that the performance index

$$J(u) = \sum_{k=1}^{\infty} \rho \theta_m^2(k) + i_{LQ}^2(k)$$

is minimised. Here, $\theta_m$ is the sensed motor position (system output), $i_{LQ}$ is the motor current reference (system input), and $\rho$ is a weighting factor. Differing values of $\rho$ can provide a trade-off between a fast response and low control effort.

A more general form of (B.10) is given by

$$J(u) = \sum_{k=1}^{\infty} x^T(k)Qx(k) + R i_{LQ}^2(k),$$

where $x$ is the state vector. In this formulation of the cost function, each of the states can be weighted individually via $Q \in \mathbb{R}^{m \times m}$; in addition to the control effort, $R \in \mathbb{R}$ (again $m$ represents the order of the system). This is the form typically used by computer-aided software packages, such as MATLAB. In the LQ system, the current reference is formed as

$$i_r(k) = i_{LQ}(k) + i_f(k) = -Kx(k) + i_f(k),$$

where $K$ is given by

$$K = \left( R + B^T S B \right)^{-1} B^T S A$$

and $S \in \mathbb{R}^{m \times m}$ is found by solving the discrete-time algebraic Riccati equation (DARE)

$$S = Q + A^T S A - A^T S B \left( R + B^T S B \right)^{-1} B^T S A,$$

where $A$ and $B$ come from (B.1a).

Figure B.1 shows the structure of the LQ system. At this point the LQ controller is a regulator approach, hence it will attempt to drive the states of the
system to zero. While this is not the desired behaviour of the servo drive controller, the architecture of the method can be adapted for set-point and tracking behaviour (see Section B.4). Furthermore, the approach requires the feedback of the full state vector. The direct measurement of the full state vector is typically not possible. This limitation is dealt with in Section B.5.

**B.4 TRACKING**

To convert the regulator system shown in Figure B.1 into a tracking system, the position reference $\theta_r$ must be introduced. Given that the desired steady-state values of the input and state are $i_{LQT}^{ss}$ and $x^{ss}$ respectively, then the new control input is defined as

$$i_r(k) = i_{LQT}(k) + i_f(k) = i_{LQT}^{ss}(k) - K(x(k) - x^{ss}(k)) + i_f(k).$$ \hspace{1cm} (B.15)

Furthermore, in steady-state (B.1) becomes

$$x^{ss}(k) = Ax^{ss}(k) + Bi_{LQT}^{ss}(k)$$ \hspace{1cm} (B.16a)

$$\theta_m^{ss}(k) = Cx^{ss}(k) + Di_{LQT}^{ss}(k)$$ \hspace{1cm} (B.16b)
where the $ss$ superscript refers to the steady-state values of the input, output and state. For any value of $\theta_{rs}^{ss}$ it is required that $\theta_{ms}^{ss} = \theta_{rs}^{ss}$. Defining $x^{ss} = N_x \theta_{rs}^{ss}$ and $i_{LQT}^{ss} = N_i \theta_{rs}^{ss}$ and substituting into (B.16) produces

$$\begin{bmatrix} N_x \\ N_i \end{bmatrix} = \begin{bmatrix} A - I & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$  (B.17)

Hence in the LQ Tracking (LQT) system, the current reference becomes

$$i_r(k) = i_{LQT}(k) + i_f(k) = N_i \theta_r(k) - K(x(k) - N_x \theta_r(k)) + i_f(k).$$  (B.18)

Figure B.2 shows the system structure for the LQT controller, where the constants $N_i \in \mathbb{R}$ and $N_x \in \mathbb{R}^{m \times 1}$ are calculated using (B.17). The order of the system is given by $m$.

**B.5 Kalman State Estimator**

As mentioned in previous sections, the full state vector is typically not available via direct measurement. A solution to this limitation is to estimate the states from the known/measured signals. Namely the control input $i_{LQT}$ and the measured plant output $\theta_m$ from Figure B.2. Note that the control reference input to the Kalman state estimator remains $i_{LQT}$. This is because the LQ model does not
include the nonlinear friction dynamics, and hence the estimator does not need to know about the additional control effort used to compensate for the nonlinear component of the friction torque. Perhaps the best known and widely accepted approach to achieving this is the Kalman state estimator or as it is sometimes known as, the Kalman filter.

The model of the system from (B.1) is corrupted by an additive vector of random noise signals, \( w \), which represents all system disturbances and modelling errors. There can be any number of disturbances contributing to \( w \), hence the matrices \( G \) and \( H \) are needed to couple these into the system. Furthermore, the output is corrupted by an additional random signal, \( v \), which represents measurement noise. The augmented system model is thus

\[
\begin{align*}
x(k + 1) &= Ax(k) + B_iLQGT(k) + Gw(k) \\
\theta_m(k) &= Cx(k) + D_iLQGT(k) + Hw(k) + v(k)
\end{align*}
\]  

(B.19a)  

(B.19b)

where \( w \in \mathbb{R}^{p \times 1} \), \( G \in \mathbb{R}^{m \times p} \) and \( H \in \mathbb{R}^{1 \times p} \). Here \( m \) is the order of the system and \( p \in \Omega \subseteq \mathbb{N} \) and \( \Omega \triangleq [1, m] \). The choice of \( p \) will depend on how the disturbance and modelling errors influence the plant. When augmented by a Kalman state estimator, the LQ controller is often referred to as a Linear Quadratic Gaussian (LQG) controller. In the LQG Tracking (LQGT) system the current reference is

\[
i_r(k) = i_{LQGT}(k) + i_f(k) = N_i\theta_r(k) - K(\hat{x}(k|k) - N_x\theta_r(k)) + i_f(k),
\]

(B.20)

where \( \hat{x}(k|k) \) is an estimate of the present state as given by the Kalman estimator equations (Dutton et al. 1997)

\[
\begin{align*}
\hat{x}(k + 1|k) &= (A - LC)\hat{x}(k|k - 1) + (B - LD)i_{LQGT}(k) + L\theta_m(k) \quad (B.21a) \\
\hat{x}(k|k) &= (I - MC)\hat{x}(k|k - 1) - MDi_{LQGT}(k - 1) + M\theta_m(k) \quad (B.21b)
\end{align*}
\]

where \( L \) is the Kalman gain and \( M \) is the innovation gain. Again they are derived by solving a DARE.
Typically the Kalman state estimator is given by (B.21a), but this equation gives the estimate of the states based on the previous output measurement. Using the innovation gain in (B.21b), it is possible to update the state estimate based on the present output measurement, and thus improve the estimate and remove some of the latency in the feedback loop.

Figure B.3 shows the system structure of the LQGT controller. Notice that the feedback now originates from the plant output and is feedback to the Kalman state estimator.

### B.6 Integartor

Due to the nature of the grinding process (see Section 1.3) there can be considerably large and unknown disturbance torques acting on the axis. An integrator can be used to effectively reject such disturbances. Augmenting the system model state equation (B.1a) by adding the integrator state given by

\[
x_I(k+1) = x_I(k) + \theta_m(k) - \theta_r(k)
\]

(B.22a)

\[
x_I(k) + Cx(k) + D i_{LQGT}(k) - \theta_r(k)
\]

(B.22b)
produces
\[
\begin{bmatrix}
    x(k+1) \\
    x_I(k+1)
\end{bmatrix} =
\begin{bmatrix}
    A & 0 \\
    C & 1
\end{bmatrix}
\begin{bmatrix}
    x(k) \\
    x_I(k)
\end{bmatrix} +
\begin{bmatrix}
    B \\
    D
\end{bmatrix} i_{LQGI}(k) +
\begin{bmatrix}
    0 \\
    -1
\end{bmatrix} \theta_r(k).
\] (B.23)

The control law for the LQ with Integrator is
\[
\begin{align*}
    i_r(k) &= i_{LQGI}(k) + i_f(k) \\
    &= N_i \theta_r(k) - K(\dot{x}(k|k) - N_x \theta_r(k)) - K_I x_I(k) + i_f(k).
\end{align*}
\] (B.24)

To determine a value for \( K_I \), it is possible to use the LQ formulation introduced in Section B.3. First, substitute \( A \) and \( B \) in (B.14) for \( A_I \) and \( B_I \) respectively, where
\[
A_I = \begin{bmatrix} A & 0 \\ C & 1 \end{bmatrix} \quad \text{and} \quad B_I = \begin{bmatrix} B \\ D \end{bmatrix}.
\] (B.25)

Then replace \( Q \) with \( Q_I \in \mathbb{R}^{m+1 \times m+1} \) where as before, \( m \) is the order of the system. The extra elements at the end of \( Q_I \) are used to weight the new integrator state in the LQ optimisation. The result will be in the form \( [K \ K_I] \). The two elements of this vector can then be substituted into (5.20). In the implementation of the integrator (B.22a) is used to update the integrator state, since \( \theta_m \) can be measured directly. Figure B.4 shows the architecture of the system including the integrator. Note that the input to the Kalman state estimator is now \( i_{LQGI} \), which replaces \( i_{LQGT} \) in (5.19).

One potential problem with using an integrator is the possibility of it winding up if the actuator saturates. For the PMSM, this corresponds to the current limit being reached. To prevent this from occurring, an anti-windup scheme is proposed. The idea is to limit the integrator state, \( x_I \), from exceeding a value which will result in the overall current reference, \( i_r \), as given by (5.20), from exceeding the motor current limit, \( i_{\text{max}} \). Using the dynamic limits from (B.8), it then follows that the allowable range of the integrator state is given by
\[
\frac{1}{K_I} \left[ i_{LQGT}(k) - i_{\text{max}}^+(k) \right] \leq x_I(k) \leq \frac{1}{K_I} \left[ i_{LQGT}(k) + i_{\text{max}}^-(k) \right],
\] (B.26)
where $i_{LQGT}$ comes from (B.20). It is noted that this is an ad-hoc method for dealing with the constraints of the system, and importantly that this is unlikely to result in optimal performance.
APPENDIX C

MODEL PREDICTIVE CONTROL FORMULATION

C.1 MPC Quadratic Optimisation Formulation

The steps to formulate the MPC problem begins by repeating the optimisation objective (6.9)

\[
\Delta U^*(k) = \arg \min_{\Delta u(k)} \{ V(k) \}, \quad (C.1)
\]

where

\[
V(k) = \sum_{i=1}^{H_t-2} \left[ \hat{y}(i|k) - r(i|k) \right]^T Q(i) \left[ \hat{y}(i|k) - r(i|k) \right]
+ \sum_{i=H_t-1}^{H_p} \left[ \hat{y}(i|k) - \hat{r}(i|k) \right]^T Q(i) \left[ \hat{y}(i|k) - \hat{r}(i|k) \right]
+ \sum_{i=0}^{H_u-1} \Delta \hat{u}^T(i|k) R(i) \Delta \hat{u}(i|k), \quad (C.2)
\]

and \(\Delta U(k) \triangleq [\Delta \hat{u}(0|k), \ldots, \Delta \hat{u}(H_u-1|k)]^T\). The prediction equation (6.10) is

\[
\begin{bmatrix}
\hat{y}(1|k) \\
\vdots \\
\hat{y}(H_p|k)
\end{bmatrix} = \Psi \hat{x}(k) + \Theta \Delta U(k) \quad (C.3)
\]

where

\[
\Psi = \begin{bmatrix} CA \\ \vdots \\ CA^{H_p} \end{bmatrix}, \quad \text{and} \quad (C.4a)
\]

\[
\Theta = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{H_u-1}B & CA^{H_u-2}B & \cdots & CB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{H_p-1}B & CA^{H_p-2}B & \cdots & CA^{H_p-H_u}B \end{bmatrix}. \quad (C.4b)
\]
MODEL PREDICTIVE CONTROL FORMULATION

With respect to

\[ R \triangleq \text{diag}\left( R(0), \ldots, R(H_u - 1) \right), \]
\[ Q \triangleq \text{diag}\left( Q(1), \ldots, Q(H_p) \right), \]
\[ Q_t \triangleq \text{diag}\left( Q(1), \ldots, Q(H_t - 2) \right), \]
\[ Q_p \triangleq \text{diag}\left( Q(H_t - 1), \ldots, Q(H_p) \right), \]

\[ T_t \triangleq \begin{bmatrix} r(1|k) \\ \vdots \\ r(H_t - 2|k) \end{bmatrix}, \]

and

\[ T_p \triangleq \begin{bmatrix} \hat{r}(H_t - 1|k) \\ \vdots \\ \hat{r}(H_p|k) \end{bmatrix}, \]

and where \( \Psi_t \) and \( \Theta_t \) are the first \( m(H_t - 2) \) rows of \( \Psi \) and \( \Theta \) respectively, and where \( \Psi_p \) and \( \Theta_p \) are the last \( m(H_p - H_t + 2) \) rows of \( \Psi \) and \( \Theta \) respectively, the cost function (C.2) is

\[
V(k) = \left( \hat{x}^T(k) \Psi_t^T + \Delta U^T(k) \Theta_t^T - T_t^T(k) \right) Q_t \left( \Psi_t \hat{x}(k) + \Theta_t \Delta U(k) - T_t(k) \right) \\
+ \left( \hat{x}^T(k) \Psi_p^T + \Delta U^T(k) \Theta_p^T - T_p^T(k) \right) Q_p \left( \Psi_p \hat{x}(k) + \Theta_p \Delta U(k) - T_p(k) \right) \\
+ \Delta U^T(k) R \Delta U(k)
\]

\[
= \left( \hat{x}^T(k) \Psi_t^T - T_t^T(k) \right) Q_t \left( \Psi_t \hat{x}(k) - T_t(k) \right) \\
+ \Delta U^T(k) \Theta_t^T Q_t \Theta_t \Delta U(k) \\
+ \left( \hat{x}^T(k) \Psi_p^T - T_p^T(k) \right) Q_p \left( \Psi_p \hat{x}(k) - T_p(k) \right) \\
+ \Delta U^T(k) \Theta_p^T Q_p \Theta_p \Delta U(k) \\
+ \Delta U^T(k) R \Delta U(k). \tag{C.5}
\]
Since
\[
\begin{align*}
(\hat{x}^T(k)\Psi_t^T - T_t^T(k))Q_t(\Psi_t\hat{x}(k) - T_t(k)) \quad \text{and} \\
(\hat{x}^T(k)\Psi_p^T - T_p^T(k))Q_p(\Psi_p\hat{x}(k) - T_p(k))
\end{align*}
\]
are not functions of $\Delta U$, they are constant in the cost function, and hence (C.5) can be simplified as
\[
V(k) = 2\Delta U^T(k)\Theta_t^TQ_t(\Psi_t\hat{x}(k) - T_t(k)) + \Delta U^T(k)\Theta_t^TQ_t\Theta_t\Delta U(k)
\]
\[
+ 2\Delta U^T(k)\Theta_p^TQ_p(\Psi_p\hat{x}(k) - T_p(k)) + \Delta U^T(k)\Theta_p^TQ_p\Theta_p\Delta U(k)
\]
\[
+ \Delta U^T(k)R\Delta U(k)
\]
\[
= 2\Delta U^T(k)((\Theta_t^TQ_t\Psi_t + \Theta_p^TQ_p\Psi_p)\hat{x}(k) - \Theta_t^TQ_tT_t(k) - \Theta_p^TQ_pT_p(k))
\]
\[
+ \Delta U^T(k)(\Theta_t^TQ_t\Theta_t + \Theta_p^TQ_p\Theta_p + R)\Delta U(k).
\]
(C.6)

Since
\[
Q = \begin{bmatrix} Q_t & 0 \\ 0 & Q_p \end{bmatrix},
\]
equation (C.6) can be further simplified as
\[
V(k) = 2\Delta U^T(k)(\Theta^TQ\Psi\hat{x}(k) - \Theta_t^TQ_tT_t(k) - \Theta_p^TQ_pT_p(k))
\]
\[
+ \Delta U^T(k)(\Theta^TQ\Theta + R)\Delta U(k),
\]
(C.7)

and then with some rearranging become
\[
V(k) = \frac{1}{2}\Delta U^T(k)\left(2R + 2\Theta^TQ\Theta\right)\Delta U(k)
\]
\[
+ \left(2\hat{x}^T(k)\Psi^TQ\Theta - 2T_t^T(k)Q_t\Theta_t - 2T_p^T(k)Q_p\Theta_p\right)\Delta U(k)
\]
\[
= \frac{1}{2}\Delta U^T(k)\left(2R + 2\Theta^TQ\Theta\right)\Delta U(k)
\]
\[
+ \left[\begin{array}{c}
\hat{x}^T(k) \\
T_t^T(k) \\
T_p^T(k)
\end{array}\right] \left[
\begin{array}{c}
2\Psi^TQ\Theta \\
-2Q_t\Theta_t \\
-2Q_p\Theta_p
\end{array}\right] \Delta U(k).
\]
(C.8)

The ZOH-ERT is given by $\hat{r}(H_t - 1|k) = r(H_t - 1|k)$ and $\hat{r}(i|k) = r(H_t|k)$, $\forall i \in [H_t, H_p]$. In matrix form this is
\[
T_p^T(k) = \left[
\begin{array}{c}
\hat{r}(H_t - 1|k) \\
\vdots \\
\hat{r}(H_p|k)
\end{array}\right]^T = \left[
\begin{array}{c}
r(H_t - 1|k) \\
r(H_t|k)
\end{array}\right]^T \Lambda_z,
\]
(C.9)
MODEL PREDICTIVE CONTROL FORMULATION

where

\[ \Lambda_z = \begin{bmatrix} I_p & 0_p & \cdots & 0_p \\ 0_p & I_p & \cdots & I_p \end{bmatrix}, \quad (C.10) \]

and where \( I_p \) and \( 0_p \) are \( m \times m \) identity and zero matrices respectively. The FOH-ERT is given by

\[ \hat{r}(i|k) = r(H_t - 1|k) + (i - H_t + 1) \left( r(H_t|k) - r(H_t - 1|k) \right), \quad \forall i \in [H_t - 1, H_p], \]

or in matrix form

\[ T_p^T(k) = \begin{bmatrix} \hat{r}(H_t - 1|k) \\ \vdots \\ \hat{r}(H_p|k) \end{bmatrix}^T = \begin{bmatrix} r(H_t - 1|k) \\ r(H_t|k) \end{bmatrix}^T \Lambda_f, \quad (C.11) \]

where

\[ \Lambda_f = \begin{bmatrix} I_p & 0_p & -I_p & \cdots & 0_p & \cdots & (H_t - H_p)I_p \\ 0_p & I_p & 2I_p & \cdots & (H_p - H_t + 1)I_p \end{bmatrix}. \quad (C.12) \]

Introducing the parameter vector as

\[ \theta(k) = \begin{bmatrix} \hat{x}(k) \\ u(k - 1) \\ r(1|k) \\ \vdots \\ r(H_t|k) \end{bmatrix}^T \]

means that (C.8) in conjunction with (C.10) or (C.12) can be expressed as

\[ V(k) = \frac{1}{2} \Delta U^T(k) \left( 2R + 2 \Theta^T Q \Theta \right) \Delta U(k) + \theta(k) \begin{bmatrix} 2\Psi^T Q \Theta \\ 0_u \\ -2Q_t \Theta_t \\ -2\Lambda, Q_p \Theta_p \end{bmatrix} \Delta U(k) \]

\[ = \frac{1}{2} \Delta U^T(k) \mathcal{H} \Delta U(k) + \theta(k) \mathcal{F} \Delta U(k), \quad (C.14) \]

where \( 0_u \) is a zero matrix of size \( l \times lH_u \),

\[ \mathcal{H} = 2R + 2 \Theta^T Q \Theta, \quad \text{and} \]

\[ \mathcal{F} = \begin{bmatrix} 2\Psi^T Q \Theta \\ 0_u \\ -2Q_t \Theta_t \\ -2\Lambda, Q_p \Theta_p \end{bmatrix}. \quad (C.15a) \]

242
C.2 MPC CONSTRAINT FORMULATION

In addition to the constraint on the control input derived in Section 6.2.1, it is also possible to define constraints on changes to the control input and on the output. From (6.14), the constraint on changes to the control input is

\[ \Delta \hat{u}_{\min}(i) \leq \Delta \hat{u}(i|k) \leq \Delta \hat{u}_{\max}(i) \quad \forall i \in [0, H_u - 1]. \]  

(C.16)

The lower bound from (C.16) may be written in matrix form as

\[
\begin{bmatrix}
-I_l & 0_l & \cdots & 0_l \\
0_l & -I_l & 0_l & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
0_l & 0_l & -I_l & I_l \\
\end{bmatrix}\Delta U(k) \leq \begin{bmatrix}
-\Delta u_{\min}(0) \\
-\Delta u_{\min}(1) \\
\vdots \\
-\Delta u_{\min}(H_u - 1) \\
\end{bmatrix},
\tag{C.17}
\]

where \( I_l \) and \( 0_l \) are an identity matrix of size \( l \) and a zero matrix of size \( l \times l \) respectively. For the upper bound

\[
\begin{bmatrix}
I_l & 0_l & \cdots & 0_l \\
0_l & I_l & 0_l & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
0_l & 0_l & I_l & I_l \\
\end{bmatrix}\Delta U(k) \leq \begin{bmatrix}
\Delta u_{\max}(0) \\
\Delta u_{\max}(1) \\
\vdots \\
\Delta u_{\max}(H_u - 1) \\
\end{bmatrix},
\tag{C.18}
\]

The constraint on changes to the control input has been converted into a set of linear inequality constraints on \( \Delta U(k) \). The matrices in (C.17) and (C.18) can be stacked to produce the coefficients of (6.16) as

\[
G_{\Delta \hat{u}} = \begin{bmatrix}
-I_l & 0_l & \cdots & 0_l \\
-0_l & -I_l & 0_l & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
-0_l & -0_l & -I_l & I_l \\
0_l & 0_l & 0_l & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
0_l & 0_l & I_l & I_l \\
\end{bmatrix}, \quad \chi_{\Delta \hat{u}} = \begin{bmatrix}
0_x & 0_l & 0_r \\
0_x & 0_l & 0_r \\
\vdots & \ddots & \ddots \\
0_x & 0_l & 0_r \\
\end{bmatrix}, \quad \text{and} \quad W_{\Delta \hat{u}} = \begin{bmatrix}
-\Delta \hat{u}_{\min}(0) \\
-\Delta \hat{u}_{\min}(1) \\
\vdots \\
-\Delta \hat{u}_{\min}(H_u - 1) \\
\end{bmatrix}.
\]

where \( 0_x \) is an \( l \times (n + m) \) zero matrix and \( 0_r \) is an \( l \times mH \) zero matrix.

From (6.15), the constraint on the controlled output is

\[ \hat{y}_{\min}(i) \leq \hat{y}(i|k) \leq \hat{y}_{\max}(i) \quad \forall i \in [1, H_p]. \]  

(C.19)
MODEL PREDICTIVE CONTROL FORMULATION

Given that the prediction equation from (6.10) is

\[
\begin{bmatrix}
\hat{y}(1|k) \\
\vdots \\
\hat{y}(H_p|k)
\end{bmatrix} = \Psi \hat{x}(k) + \Theta \Delta U(k),
\]

(C.20)

the lower bound from (C.19) may be written in matrix form as

\[-\Theta \Delta U(k) \leq \Psi \hat{x}(k) + \begin{bmatrix}
-\hat{y}_{\min}(1) \\
-\hat{y}_{\min}(2) \\
\vdots \\
-\hat{y}_{\min}(H_p)
\end{bmatrix},
\]

(C.21)

and the upper bound as

\[\Theta \Delta U(k) \leq -\Psi \hat{x}(k) + \begin{bmatrix}
\hat{y}_{\max}(1) \\
\hat{y}_{\max}(2) \\
\vdots \\
\hat{y}_{\max}(H_p)
\end{bmatrix}.
\]

(C.22)

As was the case for (6.19) and (6.20), the terms on the right-hand side of (C.21) and (C.22) are known at time \(k\). Therefore, the constraint given by (C.19) has been converted into a set of linear inequality constraints on \(\Delta U(k)\). The matrices in (C.21) and (C.22) can be stacked to produce the coefficients of (6.16) as

\[G_\gamma = \begin{bmatrix}
-\Theta \\
\Theta
\end{bmatrix}, \quad \mathcal{X}_\gamma = \begin{bmatrix}
\Psi & 0_r^* & 0_r^* \\
-\Psi & 0_l^* & 0_r^*
\end{bmatrix}, \quad \text{and} \quad \mathcal{W}_\gamma = \begin{bmatrix}
-\hat{y}_{\min}(1) \\
-\hat{y}_{\min}(2) \\
\vdots \\
-\hat{y}_{\min}(H_p) \\
\hat{y}_{\max}(1) \\
\hat{y}_{\max}(2) \\
\vdots \\
\hat{y}_{\max}(H_p)
\end{bmatrix},
\]

where \(0_r^*\) is a zero matrix of size \(mH_p \times l\) and \(0_l^*\) is a zero matrix of size \(mH_p \times mH_t\).

C.3 DIMENSIONS OF MPC VARIABLES

Given a prediction horizon \(H_p\), a trajectory horizon \(H_t\), a control horizon \(H_u\), and if the plant has \(l\) inputs, \(n\) states, \(m\) controlled outputs, \(p\) input constraints, \(q\)
input rate change constraints, and \( h \) controlled output constraints then: Table C.1 contains the dimensions of the plant variables, Table C.2 contains the dimensions of the MPC cost function variables, and Table C.3 contains the dimensions of the MPC constraint variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_p )</td>
<td>( n \times n )</td>
</tr>
<tr>
<td>( B_p )</td>
<td>( n \times l )</td>
</tr>
<tr>
<td>( C_p )</td>
<td>( m \times n )</td>
</tr>
<tr>
<td>( A )</td>
<td>( (n + m) \times (n + m) )</td>
</tr>
<tr>
<td>( B )</td>
<td>( (n + m) \times l )</td>
</tr>
<tr>
<td>( C )</td>
<td>( m \times (n + m) )</td>
</tr>
<tr>
<td>( r / y / \hat{y} )</td>
<td>( m \times 1 )</td>
</tr>
<tr>
<td>( x_p / \hat{x}_p / \Delta \hat{x}_p )</td>
<td>( n \times 1 )</td>
</tr>
<tr>
<td>( x / \hat{x} )</td>
<td>( (n + m) \times 1 )</td>
</tr>
<tr>
<td>( u / \Delta u / \hat{u} )</td>
<td>( l \times 1 )</td>
</tr>
</tbody>
</table>

Table C.1: Dimensions of plant variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_t )</td>
<td>( m (H_t - 2) \times 1 )</td>
</tr>
<tr>
<td>( T_p )</td>
<td>( m (H_p - H_t + 2) \times 1 )</td>
</tr>
<tr>
<td>( \Psi )</td>
<td>( m H_p \times (n + m) )</td>
</tr>
<tr>
<td>( \Psi_t )</td>
<td>( m (H_t - 2) \times (n + m) )</td>
</tr>
<tr>
<td>( \Psi_p )</td>
<td>( m (H_p - H_t + 2) \times (n + m) )</td>
</tr>
<tr>
<td>( \Theta )</td>
<td>( m H_p \times l H_u )</td>
</tr>
<tr>
<td>( \Theta_t )</td>
<td>( m (H_t - 2) \times l H_u )</td>
</tr>
<tr>
<td>( \Theta_p )</td>
<td>( m (H_p - H_t + 2) \times l H_u )</td>
</tr>
<tr>
<td>( \Delta \hat{u} )</td>
<td>( l H_u \times 1 )</td>
</tr>
<tr>
<td>( \mathcal{H} )</td>
<td>( l H_u \times l H_u )</td>
</tr>
<tr>
<td>( \mathcal{F} )</td>
<td>( (n + m + l + m H_t) \times l H_u )</td>
</tr>
<tr>
<td>( \Lambda_z / \Lambda_f )</td>
<td>( 2 m \times m (H_p - H_t + 2) )</td>
</tr>
<tr>
<td>( \theta )</td>
<td>( 1 \times (n + m + l + m H_t) )</td>
</tr>
<tr>
<td>( Q )</td>
<td>( m \times m )</td>
</tr>
<tr>
<td>( Q )</td>
<td>( m H_p \times m H_p )</td>
</tr>
<tr>
<td>( Q_t )</td>
<td>( m (H_t - 2) \times m (H_t - 2) )</td>
</tr>
<tr>
<td>( Q_p )</td>
<td>( m (H_p - H_t + 2) \times m (H_p - H_t + 2) )</td>
</tr>
<tr>
<td>( \mathcal{R} )</td>
<td>( l \times l )</td>
</tr>
<tr>
<td>( \mathcal{R} )</td>
<td>( l H_u \times l H_u )</td>
</tr>
</tbody>
</table>

Table C.2: Dimensions of MPC cost function variables.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>$(hmH_p + (q + p) lH_u) \times lH_u$</td>
</tr>
<tr>
<td>$G_\Delta$</td>
<td>$pH_u \times lH_u$</td>
</tr>
<tr>
<td>$G_\Delta_\Delta$</td>
<td>$qH_u \times lH_u$</td>
</tr>
<tr>
<td>$G_\Delta_\gamma$</td>
<td>$hmH_p \times lH_u$</td>
</tr>
<tr>
<td>$X$</td>
<td>$(hmH_p + (q + p) lH_u) \times (n + m + l + mH_i)$</td>
</tr>
<tr>
<td>$X_\Delta$</td>
<td>$pH_u \times (n + m + l + mH_i)$</td>
</tr>
<tr>
<td>$X_\Delta_\Delta$</td>
<td>$qH_u \times (n + m + l + mH_i)$</td>
</tr>
<tr>
<td>$X_\Delta_\gamma$</td>
<td>$hmH_p \times (n + m + l + mH_i)$</td>
</tr>
<tr>
<td>$W$</td>
<td>$(hmH_p + (q + p) lH_u) \times 1$</td>
</tr>
<tr>
<td>$W_\Delta$</td>
<td>$pH_u \times 1$</td>
</tr>
<tr>
<td>$W_\Delta_\Delta$</td>
<td>$qH_u \times 1$</td>
</tr>
<tr>
<td>$W_\Delta_\gamma$</td>
<td>$hmH_p \times 1$</td>
</tr>
</tbody>
</table>

Table C.3: Dimensions of MPC constraint variables.
C.4 SOLVING THE MPC PROBLEM

Equation (6.12) in conjunction with (6.16) is in the standard form of a quadratic optimisation problem, where $\theta(k)$ is a parameter vector and is known \textit{a priori}. At each sample instant $k$, the optimisation is solved and the input $u(k) = u(k - 1) + \Delta \hat{u}(0|k)$ is applied to the plant. At the next sample instant the optimisation is repeated using the updated parameter vector and a shifted horizon.

Optimisation is a very mature area, where many algorithms have been developed. As stated in the introduction, both online and offline optimisation is possible. Each has strengths and weaknesses which are briefly outlined here. The two main strengths of online optimisation as it relates to this work is that, one, it requires no additional calculations to develop an offline solution\(^1\), and two, because of reason one it is much simpler to use in an adaptive approach if the MPC problem is changed, for example, the plant model is refined online. The major drawback of MPC with online optimisation is the massive computation requirement to execute the algorithm. For systems with very fast update rates the amount of computation power required is not feasible given current day embedded devices. The main benefit of offline optimisation is that it in part solves this drawback. By putting in extra effort during the controller design phase, the online implementation can result in a much simpler, less computationally intense algorithm, for example gain scheduling using a lookup table. However, here the weakness is that typically the controller is then much less customisable after deployment. Although, if during operation the plant does not change significantly then perhaps this is not such a problem.

Section C.4.1 describes an online optimisation technique which is suitable for solving the optimisation given by the combination of (6.12) and (6.16), while Section C.4.2 describes an offline technique.

\(^1\)As will be seen shortly, the calculations to determine the offline solution are significant.
C.4.1 Hildreth’s Quadratic Programming Procedure

Hildreth’s Quadratic Programming Procedure is a Primal-Dual method which belongs to the group of optimisation techniques known as active set methods. In this approach, the active constraints need to be identified, and if the number of these constraints is large then so to is the computational requirement. A dual method is one which also identifies the non-active constraints and can then eliminate them from the solution, greatly simplifying the optimisation (Wang 2009).

The dual problem is formulated as

\[
\max_{\lambda \geq 0} \min_{\Delta U(k)} \left\{ \frac{1}{2} \Delta U^T(k) \mathcal{H} \Delta U(k) + \theta(k) \mathcal{F} \Delta U(k) + \lambda^T \left( \mathcal{G} \Delta U(k) - \mathcal{X} \theta^T(k) - \mathcal{W} \right) \right\} = \max_{\lambda \geq 0} \left\{ -\frac{1}{2} \lambda^T \mathcal{D} \lambda - \Phi^T \lambda - \frac{1}{2} \theta(k) \mathcal{F} \mathcal{H}^{-1} \mathcal{F}^T \theta^T(k) \right\} \tag{C.23}
\]

where

\[
\mathcal{D} = \mathcal{G} \mathcal{H}^{-1} \mathcal{G}^T, \text{ and} \tag{C.24a}
\]

\[
\Phi = \mathcal{X} \theta^T(k) + \mathcal{W} + \mathcal{G} \mathcal{H}^{-1} \mathcal{F}^T \theta^T(k). \tag{C.24b}
\]

The optimal input is then

\[
\Delta U = -\mathcal{H}^{-1} \mathcal{F}^T \theta^T(k) - \mathcal{H}^{-1} \mathcal{G}^T_{\text{act}} \lambda_{\text{act}} \tag{C.25}
\]

where \(\lambda_{\text{act}} = \lambda \geq 0\), and \(\mathcal{G}_{\text{act}}\) are the corresponding elements of \(\mathcal{G}\), that is, the rows of \(\mathcal{G}\) which form the active set of constraints.

Hildreth’s Quadratic Programming procedure is

\[
\lambda_i(n+1) = \max \left\{ 0, -\frac{1}{d_{ii}} \left[ \phi_i + \sum_{j=1}^{i-1} d_{ij} \lambda_j(n+1) + \sum_{j=i+1}^{g} d_{ij} \lambda_j(n) \right] \right\} \tag{C.26}
\]

where \(d_{ij}\) is the \(ij\)-th element of \(\mathcal{D}\) from (C.24a), \(\phi_i\) is the \(i\)-th element of \(\Phi\) from (C.24b), \(g\) is the number of rows in \(\mathcal{G}\), that is \(g = hmH_p + (q + p) lH_u\), and \(n\) is the iteration step. Given that \(\lambda_i(0) = 0, \forall i \in [1, g]\) then the optimal input is

\[
\Delta U = -\mathcal{H}^{-1} \mathcal{F}^T \theta^T(k) - \mathcal{H}^{-1} \mathcal{G}^T \lambda_{\text{conv}} \tag{C.27}
\]
where \( \lambda_{\text{conv}} \) is the vector of Lagrange multipliers obtained after (C.26) has converged. Convergence is determined using

\[
\left( \lambda(n) - \lambda(n+1) \right)^T \left( \lambda(n) - \lambda(n+1) \right) < \lambda_*,
\] (C.28)

where \( \lambda_* \) is some small threshold. This optimisation technique is used in this thesis whenever online optimisation is required. Table C.4 contains the dimensions of the optimisation variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D )</td>
<td>((hmH_p + (q + p) lH_u) \times (hmH_p + (q + p) lH_u))</td>
</tr>
<tr>
<td>( \Phi )</td>
<td>((hmH_p + (q + p) lH_u) \times 1)</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>((hmH_p + (q + p) lH_u) \times 1)</td>
</tr>
</tbody>
</table>

Table C.4: Dimensions of Hildreth’s QP procedure variables.

C.4.2 Explicit (Closed Form) Solution

As an alternative to online optimisation, recent advances in MPC have resulted in techniques specifically designed to allow extra calculations to be completed offline with the benefit that the online solution is then much simpler and less computationally intensive. This allows MPC to be applied to systems which due to their fast update rates could not previously have used MPC. These techniques when applied to MPC are often referred to as Explicit Model Predictive Control (EMPC).

While not a contribution of this thesis, for completeness, the key steps of the explicit solution presented in (Tøndel et al. 2003a) is show here. The cost function from (6.12) is

\[
V(k) = \frac{1}{2} \Delta U^T(k) \mathcal{H} \Delta U(k) + \theta(k) \mathcal{F} \Delta U(k)
\] (C.29)

Defining

\[
z(k) \triangleq \Delta U(k) + \mathcal{H}^{-1} \mathcal{F}^T \theta(T),
\] (C.30)

\[
\mathcal{S} \triangleq \mathcal{X} + \mathcal{G} \mathcal{H}^{-1} \mathcal{F}^T, \text{ and}
\] (C.31)

249
MODEL PREDICTIVE CONTROL FORMULATION

\[ V_z(k) \triangleq V(k) + \frac{1}{2} \theta(k) F H^{-1} F^T \theta^T(k), \quad (C.32) \]

(6.12) and (6.16) then become

\[ V_z(k) = \frac{1}{2} z^T(k) H z(k), \quad \text{and} \quad (C.33) \]

\[ G z(k) \leq W + S \theta^T(k). \quad (C.34) \]

The Karush-Kuhn-Tucker (KKT) conditions are given by

\[ H z(k) + G^T \lambda(k) = 0, \quad \forall \lambda \in \mathbb{R}^q, \quad (C.35a) \]

\[ \lambda_i(k) [G_i z(k) - W_i - S_i \theta^T(k)] = 0, \quad i = 1, \ldots, q, \quad (C.35b) \]

\[ -\lambda_i(k) \leq 0, \quad \text{and} \quad (C.35c) \]

\[ G z(k) \leq W + S \theta^T(k). \quad (C.35d) \]

From (C.35a) we get

\[ z(k) = -H^{-1} G^T \lambda(k) \quad (C.36) \]

and from (C.35b) we get

\[ \lambda_A(k) = 0, \quad \text{and} \quad (C.37a) \]

\[ -G_A H^{-1} G_A^T \lambda_A(k) - W_A - S_A \theta^T(k) = 0. \quad (C.37b) \]

Equation (C.37a) contains the non-active constraints, and (C.37b) contains the active constraints. From (C.37b) we get

\[ \lambda_A(k) = -\left( G_A H^{-1} G_A^T \right)^{-1} \left[ W_A + S_A \theta^T(k) \right] \quad (C.38) \]

and substituting into (C.36) gives

\[ z_A(k) = H^{-1} G_A^T \left( G_A H^{-1} G_A^T \right)^{-1} \left[ W_A + S_A \theta^T(k) \right] \quad (C.39) \]

Inserting (C.39) into (C.30) produces the optimal control sequence for the set of active constraints \(A\)

\[ \Delta U_A(k) = H^{-1} G_A^T \left( G_A H^{-1} G_A^T \right)^{-1} \left[ W_A + S_A \theta^T(k) \right] - H^{-1} F^T \theta^T(k) \quad (C.40) \]
Substituting (C.39) into (C.35d), and (C.38) into (C.35c) produces the critical region boundary

\[
\mathcal{G} \mathcal{H}^{-1} \mathcal{G}_A^T \left( \mathcal{G}_A \mathcal{H}^{-1} \mathcal{G}_A^T \right)^{-1} \left[ \mathcal{W}_A + \mathcal{S}_A \theta^T(k) \right] \leq \mathcal{W} + \mathcal{S} \theta^T(k) \tag{C.41a}
\]

\[
\left( \mathcal{G}_A \mathcal{H}^{-1} \mathcal{G}_A^T \right)^{-1} \left[ \mathcal{W}_A + \mathcal{S}_A \theta^T(k) \right] \leq 0 \tag{C.41b}
\]

When the parameter vector belongs within the boundary given by (C.41), then the optimal control sequence is given by (C.40). Thus using the explicit solution of Bemporad et al. (2002) or Tøndel et al. (2003a), the solution for \( \Delta \hat{u} \) is

\[
\Delta \hat{u}(\theta(k)) = M_i \theta^T(k) + m_i, \quad \theta(k) \in X_i, \tag{C.42}
\]

where \( M_i \in \mathbb{R}^{l \times (n + m + l + mH_t)} \) and \( m_i \in \mathbb{R}^{l \times 1} \) represent the controller gains associated with polyhedral region \( i \). Table C.5 contains the dimensions of the optimisation variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z )</td>
<td>( lH_u \times 1 )</td>
</tr>
<tr>
<td>( \mathcal{S} )</td>
<td>( (hmH_p + (q + p)lH_u) \times (n + m + l + mH_t) )</td>
</tr>
</tbody>
</table>

Table C.5: Dimensions of Explicit MPC design variables.

### C.4.3 Binary Search Tree

In the paper by Tøndel et al. (2003b), the authors propose an approach to construct a binary search tree from an EMPC solution. The idea is to construct the binary search tree such that for a given \( \theta \in X \), at each node a single affine function is evaluated and the branch condition is based on the sign of the result. Figure C.1 shows an example of a EMPC binary search tree.

The hyperplanes defining the polyhedra in the partition are given by the equality relations in (C.41). They can be combined and represented as

\[
\mathcal{Y} \theta^T = \mathcal{Z}, \tag{C.43}
\]
MODEL PREDICTIVE CONTROL FORMULATION

where \( Y \in \mathbb{R}^{L \times (n + m + l + mH_t)} \) and \( Z \in \mathbb{R}^{L \times 1} \) and \( L \) is the number of unique hyperplanes.

The algorithm for constructing the binary search tree is:

1. Generate the sets \( \mathcal{I}(j^+) \) and \( \mathcal{I}(j^-) \) for all \( j \in [1, L] \), which are the sets containing every polyhedral region in \( X \) that is on the positive and negative sides of the \( j \) hyperplane, respectively. The polyhedral region \( X_i \) is on the positive side of hyperplane \( j \) if \( Y_j \theta^T - Z_j > 0 \), for any \( \theta \in X_i \), and is on the negative side if \( Y_j \theta^T - Z_j < 0 \), for any \( \theta \in X_i \). Of course it is possible for a polyhedral region to be on both sides of a hyperplane.

2. Define \( N_k \) to be a node in the tree that consists of \((\mathcal{I}_k, J_k, C_k, j_k)\), where \( \mathcal{I}_k \) is the set of polyhedral regions that are still candidates to contain the parameter vector \( \theta \), \( J_k \) is the set of hyperplanes (including signs) which have been evaluated to reach node \( N_k \), \( C_k \) is the explicit affine control law (if the node is a leaf) defined in (C.42), and \( j_k \) is the next hyperplane which is to be evaluated (if the node is not a leaf). Here \( \mathcal{I}_k = \mathcal{I}(\mathcal{J}_k) \) and \( C_k = C(\mathcal{I}_k) \).

The root node \( N_1 \) is initialised with \((\{1, \ldots, n_r\}, \emptyset, 0, 0)\), where \( n_r \) is the total number of polyhedral regions.

3. The set of unexplored nodes is initialised as \( N \leftarrow \{N_1\} \).

4. Select any unexplored node \( N_k \in N \), and then \( N \leftarrow N \setminus N_k \).

5. Compute \( \tilde{\mathcal{I}}_k^+ = \mathcal{I}(\mathcal{J}_k) \cap \mathcal{I}(j^+) \) and \( \tilde{\mathcal{I}}_k^- = \mathcal{I}(\mathcal{J}_k) \cap \mathcal{I}(j^-) \) for all \( j \). Sort the hyperplanes by \( \max \) \( \left( \text{length}(C(\tilde{\mathcal{I}}_k^+)), \text{length}(C(\tilde{\mathcal{I}}_k^-)) \right) \), where \( C \) is the set of unique controllers (associated with the specified set of polyhedral regions).

6. Compute \( \mathcal{I}_k^+ = \mathcal{I}(\mathcal{J}_k \cup j^+) \) and \( \mathcal{I}_k^- = \mathcal{I}(\mathcal{J}_k \cup j^-) \) for a limited number of the first elements of the sorted list from step 5. Select \( j_k \) among these as \( j_k = \arg \min_j \max \left( \text{length}(C(\mathcal{I}_k^+)), \text{length}(C(\mathcal{I}_k^-)) \right) \).
SOLVING THE MPC PROBLEM

Figure C.1: EMPC binary search tree. Reproduction of an image presented in Tøndel et al. (2003b).

7. Create two new child nodes $N_{k}^{+} \leftarrow (I_{k}^{+}, J_{k} \cup j_{k}^{+}, 0, 0)$ and $N_{k}^{-} \leftarrow (I_{k}^{-}, J_{k} \cup j_{k}^{-}, 0, 0)$.

8. If $\text{length}(C(I_{k}^{+})) > 1$, add $N_{k}^{+}$ to $N$, else $N_{k}^{+}$ is a leaf node, $N_{k}^{+} \leftarrow (I_{k}^{+}, J_{k} \cup j_{k}^{+}, 0, C(I_{k}^{+}))$. If $\text{length}(C(I_{k}^{-})) > 1$, add $N_{k}^{-}$ to $N$, else $N_{k}^{-}$ is a leaf node, $N_{k}^{-} \leftarrow (I_{k}^{-}, J_{k} \cup j_{k}^{-}, 0, C(I_{k}^{-}))$.

9. If $N \neq \emptyset$, goto step 4, else terminate.

The algorithm for searching the binary search tree is:

1. Let the current node $N_{k}$ be the root node of the tree.

2. while $N_{k}$ is not a leaf node
3. Evaluate $\mathcal{V}_{j_k} \theta^T - Z_{j_k}$.

4. Let $N_k$ be the child node according to the sign of the result from step 3.

5. end (while)

6. Evaluate the control input $\Delta \hat{u}(\theta)$ corresponding to $C_k$ from $N_k$. 