Metrology Techniques for Compound Rotary-Linear Motion

by

Marsette Arthur Vona, III

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2001

© Marsette Arthur Vona, III, MMI. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part.

Author

Department of Electrical Engineering and Computer Science

Certified by...........................

David L. Trumper
Associate Professor of Mechanical Engineering

Accepted by...........................

Arthur C. Smith
Chairman, Department Committee on Graduate Students
Metrology Techniques for Compound Rotary-Linear Motion

by

Marsette Arthur Vona, III

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2001, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

It is sometimes necessary to measure the motion of an machine part which both ro-
tates about and translates along a common axis. We call this compound rotary-linear
motion or cylindrical motion. Unlike simple rotary and linear motion, sensors for
cylindrical motion are not commonly available. We describe three novel methods for
measuring cylindrical motion in the context of a precision machine tool: the tilted-
mirror interferometer (theory and implementation), the helicoid-mirror interferom-
eter (theory only), and the machine-vision technique of absolute registration (theory
only).

The tilted mirror interferometer uses laser interferometry to sense the position of
a mirror that is mounted to the end of the moving part at a slight tilt with respect to
the normal plane of the axis of motion. We use interferometer optics which measure
linear displacement, and we mount three of these optics on the fixed part of the
machine so that their beams reflect against the tilted mirror.

We implemented the tilted-mirror interferometer and integrated it into a new
5DOF machine tool under development at the MIT Precision Motion Control Lab. These are the metrological specifications of our implementation:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Translation</th>
<th>Rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>3 nm (relative)</td>
<td>63 μrad (absolute)</td>
</tr>
<tr>
<td>Update Rate</td>
<td>10 kHz</td>
<td>10 kHz</td>
</tr>
<tr>
<td>Slew Rate</td>
<td>0.25 m/s</td>
<td>1800 rad/s</td>
</tr>
<tr>
<td>Range of Motion</td>
<td>±1.5 cm</td>
<td>unlimited</td>
</tr>
</tbody>
</table>

The helicoid-mirror interferometer is similar in concept to the tilted-mirror inter-
ferometer, but the tilted mirror is replaced with a mirror formed in the shape of a
precision helicoid. The technique of relative registration is based on image registra-
tion algorithms from the field of machine vision. We present the theory behind these
latter two methods, but we have not developed implementations for either.

Thesis Supervisor: David L. Trumper
Title: Associate Professor of Mechanical Engineering
Acknowledgments

The author thanks his advisor David Trumper, Associate Professor of the Mechanical Engineering Department at MIT and director of the Precision Motion Control (PMC) Lab, for his help in developing the ideas presented herein. The author also thanks Michael Liebman, a Ph.D. candidate in the PMC lab who is working on the 5DOF machine described in Section 1.3. Michael developed the configuration of the machine, the actuators, and the control system.

The author additionally thanks the following people for helpful discussions and advice related to this work: Robert J. Hocken (Director, Center for Precision Metrology, UNC Charlotte), Dr. William T. Plummer (Director of Optical Engineering, Polaroid Corporation), Dr. Carl Zanoni (Vice President of R&D, Zygo Corporation), Doug Hart (MIT d’Arbeloff Associate Professor of Mechanical Engineering and Director of the MIT Fluids Laboratory), Charles Sodini (MIT Professor of Electrical Engineering and Computer Science and President of SMaL Camera corporation), Michael Ross (Ph.D. Candidate studying Machine Vision at MIT), and John Ziegert (Professor of Mechanical Engineering at The University of Florida).

dSPACE and ControlDesk are registered trademarks of dSPACE GmbH. MATLAB, Simulink, and Stateflow are registered trademarks of The MathWorks. Windows is a registered trademark of Microsoft Corporation.

This work was performed with the support of NSF grant DMI-0084981 and an NSF Graduate Research Fellowship.
## Contents

1 Introduction .................................................. 13
   1.1 Overview .................................................. 14
   1.2 Implementation Results .................................. 15
   1.3 A New 5DOF CNC Grinding/Milling Machine ............... 15
   1.4 Target Specifications ..................................... 17
   1.5 Related Work ............................................. 19
      1.5.1 A Combined Electromagnetic/Polarimetric Technique .... 19
      1.5.2 An Optical Encoder Technique ......................... 20
      1.5.3 Work in Machine Vision and Laser Speckle .............. 22

2 Metrology Technique A: The Tilted-Mirror Interferometer .... 25
   2.1 Overview .................................................. 26
   2.2 Recovering $z$ and $\theta$ ................................ 28
      2.2.1 Two Measurement Beams in Quadrature to Measure $\theta$ .... 29
      2.2.2 Two Measurement Beams in Quadrature Plus Independent $z$ 30
   2.3 Attempting to Use Only Two Measurement Beams ............. 34
      2.3.1 Solving for $\theta$ First .............................. 36
      2.3.2 Solving for $z$ First .................................. 40
   2.4 Calibration ................................................ 42
   2.5 Resolution ................................................. 45
   2.6 Accuracy .................................................. 48
   2.7 Slew Rates ................................................. 50
   2.8 Experimental Apparatus ................................... 51
5.2 Additional Techniques ........................................... 93
5.3 Future Work .................................................. 96
5.4 Final Thoughts ................................................ 97

A The Plane Mirror Interferometer (PMI) ......................................... 99
   A.1 Operation of the PMI .......................................... 99
   A.2 Angular Tolerance ............................................. 103
   A.3 Tilting the Measurement Mirror in Order to Measure Perpendicular
       Displacements ...................................................... 105
   A.3.1 Attempting to Change the Mirror Reflection Law .......... 107
   A.3.2 Beam Centering Optics ...................................... 116
   A.4 Summary ....................................................... 116

B Software ............................................................. 119
   B.1 HP 10897A Laser Axis Board Virtual Front Panel (VFP) ....... 120
       B.1.1 User Documentation .................................... 120
       B.1.2 Architecture .............................................. 124
       B.1.3 Building .................................................. 125
   B.2 Pentek 4284 C40 DSP Board Controller ......................... 125
       B.2.1 User Documentation .................................... 126
       B.2.2 Architecture .............................................. 129
       B.2.3 Building .................................................. 131
   B.3 C40 Code ....................................................... 131
       B.3.1 User Documentation .................................... 132
       B.3.2 Architecture .............................................. 134
       B.3.3 Building .................................................. 135
   B.4 dSPACE/Simulink Code ......................................... 136

C Electrical Schematics ................................................. 139
## List of Figures

1-1 Compound Rotary-Linear Motion ................................. 13
1-2 Photo of the Tilted-Mirror Interferometer ....................... 16
1-3 Proposed Configuration of a New 5DOF CNC Grinding Machine ... 17
1-4 Reproduction of Figures From US Patent #5,982,053............... 20

2-1 Tilted Mirror Interferometer Concept ............................ 26
2-2 Layout of the Tilted-Mirror Interferometer .................... 31
2-3 Generalized Arctangent ........................................... 34
2-4 Tilted-Mirror with Two Measurement Beams ..................... 35
2-5 Tilted-Mirror Measurement Plane .................................. 35
2-6 Tilted-Mirror Two-Beam Ambiguity ............................... 39
2-7 Tilted-Mirror Two-Beam Ambiguity Data ......................... 40
2-8 Arctangent Resolution ............................................. 47
2-9 Calibration Fit Error Plots ....................................... 49
2-10 CAD Diagram of the Tilted-Mirror Interferometer ............... 51
2-11 Side-By-Side Mounting of PMI Optics ............................ 52
2-12 CAD Diagrams of the Tilted-Mirror Mounts ..................... 53
2-13 Instrumentation Structure of the Tilted-Mirror Interferometer ... 54
2-14 Screenshot of dSPACE ControlDesk UI .......................... 55
2-15 Block Diagram of the C40 to dSPACE 1103 Data Link ............. 56
2-16 Two-Beam Ambiguity Graph ...................................... 58
2-17 Practical Example of Two-Beam Ambiguity ....................... 59

3-1 Helicoid-Mirror Interferometer Concept .......................... 63
List of Tables

1.1 Actual Specifications of the Tilted-Mirror Interferometer .......................... 16
1.2 Target Specifications of the Rotary-Linear Sensor .................................. 19

2.1 Summary of the Target and Actual Specifications of the Tilted-Mirror Interferometer ................................................................. 25

4.1 HDNS2000 Essential Characteristics ...................................................... 81

A.1 Essential Characteristics of PMI Optics and Interface Electronics ........ 103
Chapter 1

Introduction

It is sometimes necessary to measure the motion of a machine component which both rotates about and translates along a common axis, as shown in Figure 1-1. We call this compound rotary-linear motion or cylindrical motion.

Figure 1-1: Compound rotary-linear motion or cylindrical motion of a machine part.

In this thesis we explore several novel techniques for measuring cylindrical motion. Unlike simple rotary and linear motion, sensors for cylindrical motion are not commonly available. We could consider mechanically transducing the cylindrical motion of our machine to make it amenable to existing sensors. However we feel this would be undesirable due to the backlash, friction, inertia, and complexity that is likely inherent in such an approach.
1.1 Overview

There are a number of different techniques that we can use to sense cylindrical motion directly, without mechanical transduction. Our options include optical encoders, capacitative sensors, electromagnetic sensors, sensors based on the polarization of light, interferometric sensors, sensors based on laser speckle, and machine vision methods.

In Section 1.5 we describe the work of other research groups that have considered the problem of measuring cylindrical motion.

We have focused mainly on interferometric techniques (Chapters 2 and 3), and also investigated a vision- and speckle-based technique (Chapter 4). We propose some other promising concepts in Section 5.2.

In Chapter 2 we describe a new interferometric technique for cylindrical metrology called the tilted-mirror interferometer. We developed an implementation of this sensor and integrated it into the new 5DOF CNC grinding machine that we describe in Section 1.3.

The tilted-mirror interferometer uses laser interferometry to sense the position of a mirror that is mounted to the end of the moving part at a slight tilt with respect to the normal plane of the axis of motion. In Chapter 2 we describe the details of this setup and the algorithm we use to recover the rotary and translational position information from the interferometer data.

In Chapter 3 we propose a second interferometric sensor, the helicoid-mirror interferometer. Chapter 4 presents a sensor based on the machine-vision technique of absolute registration. We have not experimentally implemented these latter two sensors.

The helicoid-mirror interferometer is similar in layout to the tilted-mirror interferometer, but the tilted mirror is replaced with a mirror formed in the shape of a precision helicoid. The technique of absolute registration is based on image registration algorithms from the field of machine vision.

In this effort I have worked jointly with Michael Liebman, a Doctoral student in Professor Trumper’s Precision Motion Control lab, and we have together developed
the hardware testbed shown herein. My concentration has been on the metrology system, whereas Mike has been responsible for the overall machine design, electromechanics, and control.

1.2 Implementation Results

We have only constructed a physical implementation of one technique so far: the tilted-mirror interferometer. Our implementation, shown in Figure 1-2, is intended to demonstrate concepts for a new 5DOF CNC grinding machine that we are developing (Section 1.3). We use three two-pass Plane Mirror Interferometer (PMI) optics (Appendix A), a 7.62 cm (3 inch) diameter λ/4 flat first-surface mirror, and an auxiliary 2.54 cm (1 inch) diameter λ/20 flat first-surface mirror. The interferometer system uses a standard Hewlett-Packard HeNe laser with a wavelength of λ ≈ 633 nm. The larger mirror is tilted at about 4.7 mrad with respect to the normal plane of the axis of motion. The moving mass of our sensor is about 335 g, and the inertia is about $2.54 \times 10^{-4}$ kg · m². Additional details about our apparatus are provided in Section 2.8.

The metrological specifications of our implementation are given in Table 2.1. We strived for high resolution and high speed in our implementation of the tilted-mirror interferometer. We compute the maximum slew rates that our implementation should allow in Section 2.7, and we analyze the resolution of our sensor in Section 2.5.

Time and budget constraints have prevented us from focusing on accuracy and repeatability in our implementation. We have chosen to concentrate first on the demonstration of resolution and control performance. However, we do provide an analysis of the major factors which affect accuracy in Section 2.6.

1.3 A New 5DOF CNC Grinding/Milling Machine

We are interested in cylindrical motion because it occurs in a new high-precision (0.1 μm resolution, 10 μm accuracy), high speed (1 m/s) 5 Degree Of Freedom
Figure 1-2: The combined rotary-linear testbed for our new 5DOF machine, incorporating the tilted-mirror interferometer. Photo courtesy Michael Liebman.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Translation</th>
<th>Rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>3 nm (relative)</td>
<td>63 μrad (absolute)</td>
</tr>
<tr>
<td>Update Rate</td>
<td>10 kHz</td>
<td>10 kHz</td>
</tr>
<tr>
<td>Slew Rate</td>
<td>0.25 m/s</td>
<td>1800 rad/s</td>
</tr>
<tr>
<td>Range of Motion</td>
<td>±1.5 cm</td>
<td>unlimited</td>
</tr>
</tbody>
</table>

Table 1.1: Specifications of our implementation of the tilted-mirror interferometer.

(DOF) grinding machine that is currently under development by our group at the MIT Precision Motion Control Lab [22].

Most existing 5DOF grinding or cutting machines are designed to make parts with dimensions of about 10 cm and greater. In contrast, the design of our new machine is optimized for producing parts with dimensions on the order of 1 cm and smaller. We have proposed a new machine configuration, shown in Figure 1-3, which reduces the moving inertia of the machine by combining the z and θ DOF into a single cylindrical motion. We believe the resulting reduction in inertia, will allow our machine to grind small geometries more efficiently than conventional configurations [21].

We have thus far only constructed a prototype for the z and θ DOF of this
machine, as shown in Figure 1-2. The metrology system for these DOF is the tilted-mirror interferometer (visible in Figure 1-2 but not shown in Figure 1-3), which forms the core of this thesis.

Figure 1-3: A new configuration for a 5DOF machine tool specifically designed for producing centimeter-scale parts. Each DOF is indicated by a bold double-arrow. DOF \( z \) and \( \theta \) combine to form a cylindrical motion. Figure courtesy Michael Liebman.

### 1.4 Target Specifications

The requirements of the 5DOF grinding machine dictate the performance of the \( z-\theta \) metrology system in six areas: Resolution, Update Rate, Slew Rate, Range of Motion, Repeatability, and Accuracy.

**Resolution** Resolution defines how fine a movement can be detected by the sensor.

Our goal is 0.1 \( \mu \text{m} \) for translation, which comes directly from the specification of the machine’s resolution, and 5 \( \mu \text{rad} \) for rotation. To arrive at the value for the rotational resolution we assume a 2 cm radius between the axis of motion and the point of contact of the tool with the part. At this radius a circumferential translation of about 0.1 \( \mu \text{m} \) corresponds to a rotation of 5 \( \mu \text{rad} \) (0.1 \( \mu \text{m} \)/2 cm = 5 \( \mu \text{rad} \)).
**Update Rate**  Update rate is how often the sensor makes measurements and reports them to the control system. Our goal is a 10kHz update rate for both translation and rotation. This will allow us to build a high-bandwidth digital control system for the machine.

**Slew Rate**  Slew rate is how fast the sensed object can move. Our goal is 1 m/s for translation and 50 rad/s for rotation. Again, the translation figure comes directly from the specifications for the machine. The rotation figure is based on the assumption of a radius of about 2 cm between the axis of motion and the point of contact of the tool with the part. At this distance each radian of rotation corresponds to about 2 cm of circumferential translation. Thus, to reach 1 m/s circumferential speed we need a rotational speed of about 50 rad/s \( \left( \frac{1 \text{ m/s}}{2 \text{ cm/rad}} \approx 50 \text{ rad/s} \right) \).

**Range of Motion**  Range of motion is how far the sensed object can move. Our goal is \( \pm 1.5 \text{ cm} \) for translation and \( \pm \pi \text{ rad} \) for rotation. This will give our machine the travel it needs in order to handle centimeter-scale parts.

**Repeatability**  Repeatability describes how close a set of measurements are to each other when all measurements in the set are taken with the sensed object in the same absolute position. Our goal for repeatability is 1 \( \mu \text{m} \) in translation and 50 \( \mu \text{rad} \) in rotation.

A related issue is drift, where error accumulates over time or total distance traveled. Drift is not acceptable.

**Accuracy**  Accuracy is how close the measurements of our sensor are to accepted length standards. Our machine goal for accuracy is 10 \( \mu \text{m} \) in translation and 500 \( \mu \text{rad} \) in rotation. Time and budget constraints have prevented us from addressing the accuracy issue in this thesis. We have chosen to concentrate on the demonstration of resolution and control performance.

Table 1.2 summarizes these target specifications.
### Related Work

We have found relatively few other projects which consider specifically the problem of sensing combined rotary-linear motion.

#### A Combined Electromagnetic/Polarimetric Technique

In [12], de Wit et al describe a sensor for cylindrical motion that they have incorporated into a cylindrical actuator intended for use in a pick-and-place manufacturing operation. Translation is sensed by varying the inductance of a coil, as in an LVDT. Rotation is sensed with a coaxial arrangement of two polarizing discs which change their optical transmittance as one of the disks is rotated relative to the other. Resolutions of 10 $\mu$m in translation and 0.17 mrad in rotation are claimed.

The rotary part of the sensor described by de Wit et al depends critically on the measurement of the intensity of the beam of light that is passed through the polarizing discs. There are many possible sources of noise and drift for such an intensity measurement, including fluctuations in the power of the light source, changes in the sensitivity of the detector (possibly due to thermal effects, for example), and variations in ambient light conditions. Taken together, these effects likely result in a significant amount of noise in the output of the sensor.

The Plane Mirror Interferometer (PMI) hardware that we use in the sensors described in Chapters 2 and 3 does not rely directly on any intensity measurement. As we describe in Appendix A, our PMI system uses heterodyne detection, which is
based on the measurement of frequency rather than intensity.

1.5.2 An Optical Encoder Technique

In US Patent #5,982,053 [5], Chitayat et al describe a sensor for cylindrical motion that is part of a rotary-linear actuator which has been developed by Anorad Corporation. Both translation and rotation are sensed by optically detecting the motion of a precision grid pattern printed on a cylindrical surface of the moving object. Figure 1-4 is a reproduction of two figures from the patent which give an overview of the layout of this sensor.

Figure 1-4: Two figures from US Patent #5,982,053, by Chitayat et al, showing the concept of a rotary-linear sensor developed for Anorad corporation that employs an optical encoder measurement technique.
Unfortunately, few details are provided about the implementation. The achieved resolution is not specified. It is likely a significant manufacturing challenge to print such a precision grid pattern on the cylindrical surface. However, once this is done, such a scheme can probably achieve quite good performance.

We can get some idea of the real-world performance possible from this type sensor by considering the capabilities of the Renishaw RGX planar encoder [29]. The RGX is a commercially available 2DOF optical encoder designed to measure planar motion. It is composed of a (roughly) 70 × 70 mm grid plate and a moving read head. The grid plate is imprinted with a 40 μm pitch precision optical pattern, and the read head is designed to move over the surface of the grid plate at a small spacing. Renishaw claims resolution down to 100 nm using interpolation electronics.

If we imagine wrapping the RGX grid plate onto a cylinder then the result would be a sensor for rotary-linear motion similar to the one described by Chitayat et al. This does not seem directly feasible with RGX hardware, as the grid plate is made of glass, and the read head likely functions only on a planar surface. Nevertheless, if it were possible, the resolution in translation would remain unchanged at 100 nm. The resolution in rotation would depend on the diameter of the cylinder onto which we wrap the grid plate. For example, if we assume a diameter of 3 cm then the circumference will be about 9.24 cm, so the rotary resolution would be 

\[
\frac{9.24 \text{ cm}}{100 \text{ nm}} \approx 1 \times 10^6 \text{ counts/revolution.}
\]

For comparison, the theoretical resolution of our tilted-mirror interferometer sensor (which we describe in Chapter 2) are 0.625 nm in translation and 10 μrad (about 0.6 × 10^6 counts/revolution) in rotation. However, as we describe in Section 2.5, the practical resolutions of our implementation of the tilted-mirror interferometer are significantly lower due to the effects of noise. Thus, it seems that an optical encoder technique like that described by Chitayat et al may perform better than the sensor we implemented in rotation, but our sensor would likely still have higher resolution in translation.
1.5.3 Work in Machine Vision and Laser Speckle

In Chapter 4 we propose a technique for cylindrical metrology that we call *absolute registration*. This technique builds on a significant body of work in the machine vision field of image registration and also on work in the area of laser speckle metrology.

**Image Registration**

Image registration is the problem of finding the displacement between two similar images that results in their closest overlap. This problem is often reduced to the computation of a two-dimensional autocorrelation. An important publication in this field was [38], in which Stockham showed a way to use FFT techniques (which were newly discovered at that time) to significantly reduce the running time of the autocorrelation as compared with existing algorithms. Significant improvements to autocorrelation-based techniques are also described by Rader in [31] and Rinaldo et al in [33].

However, autocorrelation is not the only way to solve an image registration problem. In [1], Barnea and Silverman point out that autocorrelation is just one instance of a broad class of algorithms that are applicable to image registration. They call algorithms in this class Sequential Similarity Detectors (SSDs), and they demonstrate that there are other SSDs which can perform as well as—or better than—autocorrelation.

In Chapter 4, we use these results to analyze the potential performance of the absolute registration algorithm that we propose as the key component of a vision-based sensor for cylindrical motion.

**Speckle Metrology**

When a beam of coherent light is scattered by a matte surface, parts of the reflected beam interfere with each other to create an image called a speckle pattern. Because the exact appearance of the speckle pattern depends on the specific microstructure of the surface, the pattern observed at a fixed location in space will change as the surface moves relative to that location. Thus, this phenomena can be viewed as a way to
extract patterned images from a surface which are suitable for input to a registration algorithm that attempts to recover the motion of the surface. The only requirement for the surface to produce a speckle pattern is that it not be perfectly smooth (i.e. mirror-like or specular). This allows us to apply image-registration techniques to images of surfaces that are not otherwise optically “interesting.”

A number of metrological techniques have been developed which exploit the speckle phenomenon to measure various types of motion [7, 39, 3], and we incorporate laser speckle into the vision-based sensor that we propose in Chapter 4.

This work that appears to be most similar to our proposed application is described in [39], where Takemori shows a speckle-based method for measuring millimeter-scale 1D displacements of a surface in a direction tangential to the incident beam. In [45], Takemori et al further state that they have implemented 1DOF planar and rotary speckle-based sensors with resolutions of 1 μm and 1 mrad.

The systems of Takemori et al seem to be based on the technique of relative image registration. As we describe in Chapter 4, such techniques are susceptible to significant drift, which, as we stated above in Section 1.4, we cannot tolerate for our application. Our proposed vision-based sensor employs a fundamentally different technique, absolute registration, which we developed specifically with the goal of eliminating drift.
Chapter 2

Metrology Technique A: The Tilted-Mirror Interferometer

In this chapter we describe an interferometric sensor for cylindrical motion called the \textit{tilted-mirror interferometer}. We developed an implementation of this sensor and integrated it into the $z$ and $\theta$ DOF of a prototype axis for the new 5DOF CNC grinding machine described in Section 1.3. The desired and actual specifications of the sensor are given in Tables 1.1 and 1.2. We summarize them in Table 2.1.

<table>
<thead>
<tr>
<th>Specification</th>
<th>$z$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Goal</td>
<td>Actual</td>
</tr>
<tr>
<td>Resolution</td>
<td>0.1 $\mu$m</td>
<td>0.625 nm (theoretical)</td>
</tr>
<tr>
<td>Update Rate</td>
<td>10 kHz</td>
<td>10 kHz</td>
</tr>
<tr>
<td>Peak Slew Rate</td>
<td>1 m/s</td>
<td>0.25 m/s</td>
</tr>
<tr>
<td>Range of Motion</td>
<td>$\pm1.5$ cm</td>
<td>$\pm1.5$ cm</td>
</tr>
<tr>
<td>Repeatability</td>
<td>1 $\mu$m</td>
<td>not measured</td>
</tr>
<tr>
<td>Accuracy</td>
<td>10 $\mu$m</td>
<td>not measured</td>
</tr>
</tbody>
</table>

Table 2.1: Specifications of the tilted-mirror sensor. Rationale behind the Goal values is given in Section 1.4. Practical resolution is lower than theoretical due to non-repeatable noise in the system as described in Section 2.5. We have not attempted to approach the calculated maximum $\theta$ slew rate, nor have we measured repeatability or accuracy.

In the first part of this chapter, Section 2.1 to Section 2.7, we develop and analyze
the mathematic and algorithmic details of our implementation. In the second part, Section 2.8, we describe our experimental apparatus and instrumentation.

## 2.1 Overview

Our tilted-mirror interferometer uses laser interferometry to sense the position of a mirror that is mounted to the end of the moving part at a slight tilt (about 4.7 mrad) with respect to the normal plane to axis of motion, as shown in Figure 2-1.

![Figure 2-1: The tilted-mirror interferometer concept (figure not to scale). Two measurement beams (at arbitrary locations) are pictured here to illustrate the concept, but in general \( n \) beams might be used. In our implementation we use two beams plus an additional \( z \)-axis measurement.](image)

As the figure shows, the tilted-mirror interferometer employs multiple measurement beams aligned parallel with the axis of motion. For the sake of discussion, let's initially consider the number and locations of the beams to be variables. Each beam reports simple linear displacement, so we can consider the response of the \( i \)th measurement beam, \( d_i \), to be a measure of the position of the mirror surface along beam...
i. The vector of responses \( \mathbf{d} = [d_1, \ldots, d_n]^T \) is used as input data to an algorithm which returns the \( z \) and \( \theta \) positions of the moving machine part. We provide the details of this algorithm in Section 2.2.

In our implementation we actually use two measurement beams plus an additional independent measurement of \( z \) given by a third beam and an auxiliary mirror. We describe how this configuration works in Sections 2.2.1 and 2.2.2.

Why, though, should we require three measurements when the motion we are sensing has only two degrees of freedom? It turns out that there is a singularity in the geometry which is unavoidable using only two beams. We describe this problem, and motivate our three-beam solution, by considering potential arrangements that use only two measurement beams in Section 2.3. Later, in Section 2.9, we provide a concrete example of a singular configuration that we would observe in our apparatus if we were to remove one of the beams.

Interestingly, our implementation of the tilted-mirror interferometer is able to provide an \textit{absolute} measurement of \( \theta \) but only a relative measurement of \( z \). We explain why in Section 2.4.

As we describe below in Section 2.2, in order to recover \( z \) and \( \theta \) we need to know some parameters related to the specific way in which we built the sensor (e.g. the specific spatial relationships between the positions of the interferometers and the axis of motion). One solution to this problem is to expend great effort to construct the sensor precisely, so that parameters are known by design. Rather than attempt this mechanically challenging approach, we have developed an automatic calibration routine that measures the required parameters of the sensor after we have built it. We describe this calibration routine in Section 2.4.

We strived for high resolution and high speed in our implementation of the tilted-mirror interferometer. We analyze the resolution of our sensor in Section 2.5, and we compute the maximum slew rates that our implementation should allow in Section 2.7.

Time and budget constraints have prevented us from focusing on accuracy and repeatability in our implementation. We have chosen to concentrate first on the demonstration of resolution and control performance. However, we do provide an
analysis of the major factors which affect accuracy in Section 2.6.

2.2 Recovering $z$ and $\theta$

In this section we develop the algorithm which transforms the vector of PMI responses $\mathbf{d} \equiv [d_1, \ldots, d_n]^T$ into the $z$ and $\theta$ positions of the moving machine part. Simultaneously, we describe a good placement for the measurement beams.

We mount the PMI optics on the fixed part of the machine so that the measurement beams are located at distinct measurement points in the vertical ($xy$) world plane. As shown in Figure 2-1, if measurement beam $i$ is located at radius $R_i$ from the axis of motion and angle $\phi_i$ with respect to horizontal (i.e. the $xz$ plane), and if the tilt of the mirror is $\alpha$, then the response of this beam, $d_i$, will depend on the machine position in both $z$ and $\theta$:

$$d_i(z, \theta) = C_i + z + A_i \sin(\theta - \phi_i), \quad A_i \equiv R_i \tan(\alpha)$$

where $C_i$ is a constant that depends on where the interface electronics has been zeroed. We will refer often to equation (2.1) in the following sections. It can be easily remembered: it's just a general sinusoid in $\theta$ with DC offset ($C_i + z$), amplitude $A_i$, and phase offset $\phi_i$.

Equation (2.1) has two unknowns. Thus, we should be able to recover $z$ and $\theta$ with $n \geq 2$ measurement beams, provided we know the vectors of parameters $\mathbf{C} \equiv [C_1, \ldots, C_n]^T$, $\mathbf{A} \equiv [A_1, \ldots, A_n]^T$, and $\mathbf{\phi} \equiv [\phi_1, \ldots, \phi_n]^T$. We describe how to measure $\mathbf{C}$, $\mathbf{A}$, and $\mathbf{\phi}$ in Section 2.4.

Minimality suggests that we use as little hardware as possible, i.e. that we use exactly two measurement beams and no other information. Unfortunately, the non-linearity of (2.1) results in multiple solutions given only two equations, and it seems impossible to find an algorithm which consistently picks the correct solution given only two measurements.

The solution we implemented does use two measurement beams to measure $\theta$ but
it also includes an independent measurement of $z$ provided by a third measurement beam. This allows us to always pick the correct solution for $\theta$. We describe this arrangement in Section 2.2.2.

First, by way of introduction, in Section 2.2.1, we temporarily sidestep the issue of recovering both $z$ and $\theta$ from the input data, and consider the simpler problem of recovering $\theta$ in an ideal situation, if $z$ is already known.

### 2.2.1 Two Measurement Beams in Quadrature to Measure $\theta$

In Section 2.3 we show that if we have only two measurement beams then we can essentially only measure $\sin(\theta)$ from the information they provide. Since $\arcsin(\theta)$ has multiple solutions (even when we restrict $\theta$ to a $2\pi$ interval), there will be an unavoidable ambiguity in our knowledge of $\theta$.

How can we overcome this problem? It would be nice if, in addition to having $\sin(\theta)$, we also had $\cos(\theta)$. Then we could use the two-argument arctangent ($\arctan2$) to recover $\theta$ without ambiguity and with no singularities:

$$\theta = \arctan2(\sin(\theta), \cos(\theta)) \quad (2.2)$$

This is a common procedure in metrology (and elsewhere) called *quadrature detection*.

If we somehow know $z$ first, then we can arrange two measurement beams, call them $a$ and $b$, in specific locations so that they provide us with $\sin(\theta)$ and $\cos(\theta)$. Let's consider an ideal situation where the following constraints hold:

$$\phi_a = 0$$
$$\phi_b = -\pi/2$$
$$R_a = R_b$$

We call this situation “ideal” because it would be difficult to build a machine that satisfies these geometric constraints to a very high tolerance. Nevertheless, let's consider such a machine for now. In Section 2.2.2 we describe how we measure $z$ and
also how we handle the case where the constraints are not perfectly satisfied.

If we apply the constraints to (2.1) we get

\[ d_a(z, \theta) = C_a + z + A \sin(\theta) \]  
\[ d_b(z, \theta) = C_b + z + A \cos(\theta) \]

\[ A \equiv R_a \tan(\alpha) = R_b \tan(\alpha) \]

Given \( z \) we can compute \( \theta \) from (2.3) and (2.4) like this:

\[ \theta(d_a, d_b, z) = \arctan2(a, b) \]

\[ a \equiv d_a - C_a - z \]
\[ b \equiv d_b - C_b - z \]

Again, in order for this to work we must somehow know \( z \) already. In Section 2.3.2, we consider the possibility of finding a function that recovers \( z \) directly from \( d_a \) and \( d_b \), where interferometers \( a \) and \( b \) may be positioned in quadrature. This would allow us to use quadrature detection for \( \theta \) and still have only two measurement beams in the entire system. Unfortunately, it seems that such a function will have its own ambiguity problem.

### 2.2.2 Two Measurement Beams in Quadrature Plus Independent \( z \)

Instead of trying to recover \( z \) from \( d_a \) and \( d_b \), we can simply add a third measurement beam (beam \( c \)) to measure \( z \) directly. Then we will encounter no ambiguities or singularities as we measure \( z \), and, if we arrange measurement beams \( a \) and \( b \) in quadrature, nor will we have any problem measuring \( \theta \).

We can place the three beams so that they each reflect against the tilted mirror at a different location. However, as shown in Figure 2-2, we have chosen to place beam \( c \) at the opposite side of the moving machine part, where it reflects against an
auxiliary mirror. We explain why below.

Figure 2-2: Our implementation of the tilted-mirror interferometer employs two measurement beams (a and b, on the right) to measure $\theta$ and an additional measurement beam (c, on the left) to measure $z$.

To measure $z$ directly, we place beam c so that, ideally, $R_c = 0$. Then (2.1) gives

$$d_c(z, \theta) = C_c + z$$

(2.7)

Now it’s trivial to get $z$:

$$z(d_c) = d_c - C_c$$

(2.8)

And given this we can ideally get $\theta$ from (2.6).

Dealing With Deviations from the Ideal Constraints

Is everything perfect now? Nearly. We do require three measurement beams instead of two. But there is an additional issue. In the above analysis we made the *idealizations* $\phi_a = 0$, $\phi_b = -\pi/2$, $R_a = R_b$, which place beams a and b in quadrature, and $R_c = 0$, which allows us to measure $z$ directly. It would be difficult to actually construct a machine which satisfies all of these constraints exactly.

We deal with the last constraint, $R_c = 0$, in the following way: We position measurement beam c so that it reflects against a small auxiliary mirror attached to the opposite end of the moving machine part from the large tilted mirror. We align
beam c closely with the axis of motion, and we adjust the mount of the auxiliary mirror so that its surface is nearly perpendicular to the axis of motion. Thus the effect of any deviation of \( R_c \) from center will be minimal. It may be even better to replace the auxiliary mirror with a retroreflector so that its mounting orientation is not critical, but we have not explored this further.

What about the first three constraints, that \( \phi_a = 0, \phi_b = -\pi/2 \), and \( R_a = R_b \)? Fortunately, we can rework the math to allow deviations from these constraints if we know the parameters \( C, A, \) and \( \phi \), which we show how to measure in Section 2.4.

In our setup, the ideal quadrature constraints are only approximately satisfied, i.e. \( \phi_a \approx 0, \phi_b \approx -\pi/2 \), and \( R_a \approx R_b \). Applying these to (2.1), we get

\[
\begin{align*}
d_a(z, \theta) &\approx C_a + z + A_a \sin(\theta) \\
d_b(z, \theta) &\approx C_b + z + A_b \cos(\theta) \\
A_a &\approx A_b
\end{align*}
\]

Since we have already taken care of the constraint \( R_c = 0 \), we can use (2.8) to find \( z \). And once we have \( z \) we can get an approximation to \( \theta \) by following (2.6):

\[
\theta(d_a, d_b, d_c) \approx \arctan2(a, b) \quad (2.11)
\]

\[
a \equiv d_a - C_a - z(d_c) \\
b \equiv d_b - C_b - z(d_c)
\]

In order to turn the approximation in (2.11) into an equality, we need to account for the deviations of each of the quadrature constraints from ideal.

One of the quadrature constraints is \( R_a \approx R_b \), which leads to \( A_a \approx A_b \). We handle any deviation from this by normalizing \( a \) and \( b \):

\[
\hat{a} \equiv \frac{[d_a - C_a - z(d_c)]}{A_a}
\]

\[\text{Actually, as we discuss in Section 2.8, in our setup we really have } \phi_a \approx 3\pi/4 \text{ and } \phi_b \approx \pi/4, \text{ because that was easiest to construct. We can ignore this detail for now because we can correct for it by simply assuming a rotation of the world coordinate system by } 3\pi/4.\]
\[ \hat{b} \equiv \frac{[d_b - C_b - z(d_c)]}{A_b} \]

We accommodate the constraint \( \phi_a \approx 0 \) by assuming our calculations are done in a rotated coordinate system where the zero point of \( \theta \) coincides with \( \phi_a \) in the world coordinate system. When we are finished we can add \( \phi_a \) to the computed value of \( \theta \) to return to the world coordinate system.

Our only remaining quadrature constraint is that \( \phi_b \approx -\pi/2 \). Let’s investigate the effects of a deviation from this by reasoning about an abstract triangle. This triangle has two legs of length \( \hat{a} \) and \( \hat{b} \), and the angle between them is \( \phi_d \equiv \phi_a - \phi_b \), as shown on the right in Figure 2-3. On the left in the figure is the ideal case where \( \phi_d = \pi/2 \). In this case we have a right triangle, so the arctan2 function will give us an exact result for \( \theta \) given \( \hat{a} \) and \( \hat{b} \). In the more general case, where \( \phi_d \) may be different from \( \pi/2 \), we can construct a virtual right triangle with legs of length

\[
\begin{align*}
\hat{a}' &\equiv \hat{a}\sin(\phi_d) \\
\hat{b}' &\equiv \hat{b} - \hat{a}\cos(\phi_d)
\end{align*}
\]

The angle \( \theta \) is the same in both the original triangle and in the virtual right triangle, so (remembering to add \( \phi_a \) as described above to return to world coordinates)

\[
\theta(d_a, d_b, d_c) = \text{arctan2} \left( \hat{a}', \hat{b}' \right) + \phi_a \quad (2.12)
\]
gives an exact formula for computing \( \theta \) from \( d_a, d_b \), and \( d_c \).

Thus equations (2.8) and (2.12) compute the exact values of \( z \) and \( \theta \) from the three measurement beam responses. It should be clear that all the constants in these equations can be derived directly from the measured parameters \( C, A \) and \( \phi \).

Because we have used direct \( z \) measurement and quadrature measurement of \( \theta \), once we restrict \( \theta \) to be in a \( 2\pi \) interval, e.g. \( \theta \in (-\pi, \pi] \), there is no remaining ambiguity in (2.8) or (2.12). Neither are there any regions where \( |dz/dd_c|, |\partial\theta/\partial d_a| \),

\[\text{2Actually, we can’t measure } C_c \text{ using the procedure described in Section 2.4. This means that our sensor will only provide a relative measurement of } z.\]
Figure 2-3: Accounting for the constraint that $\phi_d \approx \pi/2$. On the left is the ideal case where $\phi_d = \pi/2$. On the right is the general case where $\phi_d$ may differ from $\pi/2$.

$|\partial \theta / \partial d_b|$, or $|\partial \theta / \partial d_c|$ approach infinity, so there are no singularities in this scheme.

2.3 Attempting to Use Only Two Measurement Beams

For completeness, in this section we'll examine the possibility of recovering both $z$ and $\theta$ from only two measurement beams. We do this in a fully general way by allowing the two beams to be at any arbitrary locations. We demonstrate that, no matter where the two beams are placed, and whether we solve for $\theta$ first or for $z$ first, we will encounter an unavoidable ambiguity in the results of our computation and several corresponding singularities.

Let the two measurement beams be $a$ and $b$. Let

\[
\begin{align*}
\mathbf{a} \equiv (a_x, a_y) &= (R_a \cos(\phi_a), R_a \sin(\phi_a)) \\
\mathbf{b} \equiv (b_x, b_y) &= (R_b \cos(\phi_b), R_b \sin(\phi_b))
\end{align*}
\]

be their coordinates in the vertical ($xy$) world plane, as shown in Figure 2-4. Then the distance between the measurement points, and hence the perpendicular distance between the measurement beams, is $l \equiv || \mathbf{b} - \mathbf{a} ||$. Define the plane which contains the two measurement beams to be the measurement plane, and define the angle this plane makes with the horizontal ($xz$) world plane $\psi$. Figure 2-5 gives a view of the
measurement plane.

Figure 2-4: Two measurement beams with measurement points a and b.

Figure 2-5: The *measurement plane*, defined as the plane which contains the two measurement beams.

Equation (2.1) leads to the following equations for the response of the measurement beams as functions of z and θ.

\[
d_a(z, \theta) = C_a + z + A_a \sin(\theta - \phi_a), \quad A_a = R_a \tan(\alpha) \quad (2.13)
\]
\[
d_b(z, \theta) = C_b + z + A_b \sin(\theta - \phi_b), \quad A_b = R_b \tan(\alpha) \quad (2.14)
\]

35
To recover \( z \) and \( \theta \) from \( d_a \) and \( d_b \) we need to invert (2.13) and (2.14). We can eliminate \( z \) quite easily from these equations by subtracting one from the other, so let’s try to solve for \( \theta \) first.

### 2.3.1 Solving for \( \theta \) First

Instead of subtracting (2.13) directly from (2.14), it will be convenient to define a new variable \( \delta \):

\[
\delta \equiv (d_b - C_b) - (d_a - C_a)
\]

Geometrically, \( \delta \) is the difference in depth of the mirror between the two measurement points in the measurement plane, as shown in Figure 2-5.

We can recover \( \theta \) directly from \( \delta \). First, consider the special case where \( a = 0 \) and \( \psi = 0 \). In this case measurement beam a is aligned with the axis of motion (since \( R_a = 0 \)) and its response degenerates to

\[
d_a(z, \theta) = C_a + z
\]

Also in this case \( R_b = l \) and \( \phi_b = 0 \), so

\[
d_b(z, \theta) = C_b + z + l \tan(\alpha) \sin(\theta)
\]

Given these, the next two equations show how to compute \( \theta \) from \( \delta \) when the conditions of this special case are satisfied:

\[
\delta = (d_b - C_b) - (d_a - C_a) = l \tan(\alpha) \sin(\theta)
\]

\[
\theta(\delta) = \arcsin \left( \frac{\delta}{l \tan(\alpha)} \right)
\]

Conveniently, it turns out that relaxing the requirements of the special case does not require us to change the result very much.

Since we are considering the surface of the mirror to be flat, and since we are only making a differential measurement of its height at two points, we can relax the
requirement that \( a = 0 \) without modifying (2.16).

Relaxing the requirement that \( \psi = 0 \) simply means that the measurement plane has a different orientation in the world, so all we need to do is add an offset of \( \psi \) to (2.16) to account for this:

\[
\theta(\delta) = \arcsin \left( \frac{\delta}{l \tan(\alpha)} \right) + \psi \quad (2.17)
\]

As there were no other constraints this equation is fully general. However, it does require two constants, \( l \tan(\alpha) \) and \( \psi \). We have claimed that we can measure the vectors of parameters \( C, A \) and \( \phi \). We can get \( \psi \) from these measured parameters as follows:

\[
\psi = \arctan \left( \frac{y_b - y_a}{x_b - x_a} \right) = \arctan \left( \frac{R_b \sin(\phi_b) - R_a \sin(\phi_a)}{R_b \cos(\phi_b) - R_a \cos(\phi_a)} \right) \tan(\alpha)
\]

\[
\psi = \arctan \left( \frac{A_b \sin(\phi_b) - A_a \sin(\phi_a)}{A_b \cos(\phi_b) - A_a \cos(\phi_a)} \right) \quad (2.18)
\]

We can derive \( l \tan(\alpha) \) from the measured parameters by applying the law of cosines:

\[
l^2 = R_a^2 + R_b^2 - 2R_aR_b \cos(\phi_b - \phi_a)
\]

\[
l^2 \tan^2(\alpha) = \left[ R_a^2 + R_b^2 - 2R_aR_b \cos(\phi_b - \phi_a) \right] \tan^2(\alpha)
\]

\[
l \tan(\alpha) = \sqrt{A_a^2 + A_b^2 - 2A_aA_b \cos(\phi_b - \phi_a)} \quad (2.19)
\]

Thus we can determine both constants in (2.17) only from the measured parameters.

Now that we have \( \theta \) we can get \( z \) quite simply. Each beam should give us an independent measure of \( z \), and we can take their average for increased robustness with respect to measurement noise:

\[
z_i(d_i) = d_i - C_i - A_i \sin(\theta - \phi_i)
\]

\[
z(d_a, d_b) = \frac{1}{2} [z_a(d_a) + z_b(d_b)]
\]
Ambiguity

Since (2.17) relies on arcsin it has multiple solutions. Clearly we can limit $\theta$ to some contiguous interval of length $2\pi$, and throw out solutions outside this interval. Mathematically it’s convenient to pick the interval $\theta \in (\psi - \pi, \psi + \pi]$. However, even after we impose this restriction there is still a binary ambiguity in $\theta$. Specifically, if $\theta \in [\psi, \psi + \pi/2)$ then there is another solution $\theta' \in (\psi + \pi/2, \psi + \pi]$, and vice-versa, and if $\theta \in (\psi - \pi/2, \psi)$ then there is another solution $\theta' \in (\psi - \pi, \psi - \pi/2)$, and vice-versa.

This ambiguity is not just a mathematical anomaly. There really are two orientations for the mirror in these cases which yield the same value for $\delta$. To get a handle on this, consider Figure 2-6. Without loss of generality, let $z$ be fixed. Let $\mathbf{n}$ be the surface normal at the point where the axis of motion intersects the surface of the mirror. Then as $\theta$ varies $\mathbf{n}$ prescribes a cone, which we’ll call the normal cone. For each value of $\theta$ there is a plane perpendicular to the measurement plane that contains $\mathbf{n}$. Let’s call this the ambiguity plane. Then in most cases the ambiguity plane intersects the normal cone in two distinct lines, one contains $\mathbf{n}$, and the other contains $\mathbf{n}'$, the surface normal that corresponds to the alternate solution $\theta'$. These two surface normals correspond to two distinct orientations of the mirror which yield the same value for $\delta$ because they are symmetric with respect to the axis of motion and they are both contained in the ambiguity plane, which is perpendicular to the measurement plane. However, $z$ is the same (by design) for these two orientations, so $\theta'$ must be different from $\theta$.

The only two values of $\theta$ which are not ambiguous are $\theta = \psi + \pi/2$ and $\theta = \psi - \pi/2$. These correspond to the situations where the ambiguity plane is tangent to the normal cone, so their intersection degenerates to a single line.

Another way to visualize the ambiguity problem is actually to consider the effects of small rotations and translations when the tilted mirror starts in one of the two non-ambiguous orientations. We provide an example of this in Section 2.9. Also in that section we derive that $\partial d_a/\partial \theta = \partial d_b/\partial \theta$ when the mirror is in such an orientation.
Figure 2-6: For a fixed value of $z$, $n$ prescribes a cone as $\theta$ varies. We call this the normal cone. We call the plane perpendicular to the measurement plane that contains $n$ the ambiguity plane. As long as they are not tangent, the ambiguity plane intersects the normal cone in exactly two lines. One contains $n$, and the other contains $n'$, the normal which corresponds to the alternate solution $\theta'$. This means that a small rotation will have the same observable effect in the values of $d_a$ and $d_b$ as a small translation, so the two will be indistinguishable.

We might attempt to overcome this ambiguity problem by imposing a continuity constraint. There is a clean boundary between the two solution regions, namely the points at which $\theta = \psi \pm \pi/2$. If we know which region $\theta$ starts in, then we might be able to track it over time, and if it approaches one of the boundary points with any momentum then we might assume that it actually passes through the boundary and into the other solution region\(^3\).

There is a problem with this scheme if $\theta$ ever approaches a boundary point slowly, i.e. it's hard to know whether it approaches the boundary and turns back, or if it actually crosses over. We might try to rectify this issue by specifying a minimum speed the machine must maintain for $\theta$ when near a boundary point. This is awkward. However, there is a compounding factor which would ruin even this plan: The

\(^3\)This is somewhat reasonable because $\theta$ is tied to the motion of a real machine part with finite inertia.
boundary points are exactly the regions where $|d\theta/d\delta| \to \infty$, i.e. they are singular points. This means that in these regions small fluctuations in the measurement $\delta$ create large disturbances in the computed value of $\theta$. This results in somewhat wild behavior for $\theta$, which makes deciding when to jump from one solution region to the other virtually impossible. Some illustrations are provided in Figure 2-7 of the boundary between solution regions (taken from real experimental data).

![Rotation (θ) vs Time](image)

**Figure 2-7:** Some illustrations of the boundary between solution regions. In this run, the moving part was rotating with a constant velocity, so the actual trajectory in $\theta$ is a sawtooth (hand-colored in magenta). We would like to recover this sawtooth from the data, but the data has a binary ambiguity. The two solutions possible from the data are shown by the blue and green plots. Unfortunately, noise makes it difficult to track $\theta$ when we need to jump from one solution to the other at the boundaries between solution regions. The plots of $\theta$ and $\theta'$ are generated from real experimental data.

### 2.3.2 Solving for $z$ First

What if we try to solve for $z$ first, and then use $z$ to compute $\theta$? For example, we might want to do the computations in this order if beams $a$ and $b$ are positioned in quadrature, as described in Section 2.2.1. We can attempt this, but we will unfortunately run into an ambiguity/singularity situation again.
Our goal is to find a function that returns $z$ given only the two measurements $d_a$ and $d_b$ and the parameters $C$, $A$, and $\phi$. We can take a shortcut to derive this function by using (2.17), which was the main result of Section 2.3.1. Without loss of generality we can assume that $\psi = 0$, because the orientation of the measurement plane doesn’t matter if we are only interested in measuring $z$. If we substitute (2.17), with $\psi = 0$, into (2.13) and simplify, we get

$$z(d_a, d_b) = d_a - C_a - \frac{A_a}{(l \tan \alpha)/(\sin \phi_a)} \left[ \frac{\delta}{(\tan \phi_a)} \pm \sqrt{l^2 \tan^2 \alpha - \delta^2} \right]$$

where

$$\delta = (d_b - C_b) - (d_a - C_a)$$

We can compute $l \tan \alpha$ from $C$, $A$, and $\phi$ by (2.19), so all the constants in (2.21) can be computed from the measured parameters.

Due to the ±, (2.21) will generally have two solutions. The only cases where it has a unique solution are where $\sqrt{l^2 \tan^2 \alpha - \delta^2} = 0$, or $\delta = \pm l \tan \alpha$. These orientations are the boundaries between the ambiguous regions. Unfortunately, they are also the orientations where $|\partial z/\partial \delta| = \infty$, i.e. they are singular points, so we are again thwarted in any attempt to apply a continuity algorithm to decide which solution to use in the ambiguous regions.

The ambiguity here is also real, and not just an artifact of the math. We can again think about it geometrically using the normal cone and ambiguity plane (Figure 2-6). This time, however, we don’t hold $z$ constant. Instead, we pivot the mirror about the line defined by the intersection of the plane of its surface and the measurement plane. If we hold this line constant $\delta$ will not change. But, except in the degenerate cases when the ambiguity plane is tangent to the normal cone (these occur when $\delta = \pm l \tan \alpha$, where the major slope of the mirror is aligned with the measurement plane), there will be exactly two such pivots where the center of the mirror lies on the axis of motion. These are two valid orientations of the mirror where $\delta$ is the same but $z$ differs.
2.4 Calibration

How can we measure the parameters $C$, $A$, and $\phi$? $A$ is the product of the vector of measurement beam radial positions with the tangent of the tilt angle of the mirror, and $\phi$ is the vector of angular positions of the measurement beams. These are both static characteristics of the specific geometry of our implementation, so it’s theoretically possible to measure them directly. We could attempt to do this with some external metrology system. However this would be inconvenient, because each time we make any adjustments to the positions of the measurement beams or the mirror we would have to repeat these measurements.

And what about $C$, the vector of zero positions of the measurement beams? $C$ is not a static geometric characteristic, it depends only on when the PMI receiver electronics were set to zero themselves. This will likely be different for each run of the machine.

To address these issues, we developed a quick and completely automated way to accurately measure $C$, $A$, and $\phi$ at the start of each run. The procedure is as follows:

1. Use open-loop control (no metrology) to move $z$ and $\theta$ to a rough home position. This position need not be exactly attained. In our implementation, we energize a single phase of the $z$ and $\theta$ motors, which causes the rotor to seek the detents which correspond to these phases.

2. When the home position has been roughly attained, zero the receiver electronics.

3. Create an initial metrology system using stored values of $C$, $A$, and $\phi$ which are chosen to be roughly correct when the receiver electronics are zeroed with the machine in the rough home position.

4. Use closed-loop control to hold $z$ constant at $z = z_0$. With the setup described above, which has beam c aligned with the axis of motion (i.e. $R_c = 0$), it’s easy to do this.

5. With $z$ held constant, spin the rotor at a constant angular velocity $\omega \equiv \dot{\theta}$. We use $\omega = 2.5 \text{ rev/s} \approx 15.708 \text{ rad/s}$. 

42
6. While the rotor is spinning, record the data values $d(t) \equiv [d_1(t), \ldots, d_n(t)]$ at each of a set of sample times $t \equiv [t_1 < \cdots < t_N]$. We record data for 5 seconds at a rate of 5 kHz.

7. If $z_0$ and $\omega$ were truly held constant, and there were no other error motions, and if the mirror was perfectly flat, then (2.1) tells us that the recorded data should ideally have the form

$$\hat{d}(t) = C + z_0 \mathbf{1} + A \sin_e((\omega t + \theta_0) \mathbf{1} + \phi)$$

$$\sin_e \equiv \text{element-wise sine}$$

$$\mathbf{1} \equiv [1 \ldots 1]^T \text{ s.t. size(1) = size(C) = size(A) = size(\phi)}$$

The caret ($\hat{}$) reminds us that $\hat{d}$ is an ideal model. $\theta_0$ is a constant that depends on the angular position of the machine at the moment we started recording data. We discuss how we deal with $z_0$ and $\theta_0$ below. For now, let's continue by expressing (2.22) as a linear combination of three basis functions

$$\hat{d}(t) = (C + z_0 \mathbf{1}) + A \sin_e(\theta_0 \mathbf{1} + \phi) \cos(\omega t) + A \cos_e(\theta_0 \mathbf{1} + \phi) \sin(\omega t)$$

$$\hat{d}(t) = D + E \cos(\omega t) + F \sin(\omega t)$$

$$D \equiv C + z_0 \mathbf{1}$$

$$E \equiv A \sin_e(\theta_0 \mathbf{1} + \phi)$$

$$F \equiv A \cos_e(\theta_0 \mathbf{1} + \phi)$$

$$\cos_e \equiv \text{element-wise cosine}$$

Since (2.23) is linear in $D$, $E$, and $F$, we can use a linear least-squares algorithm to fit the actual recorded data $d(t)$ (we use the SVD [30]). This procedure will return values for $D$, $E$, and $F$, and from these we can recover $C$, $A$, and $\phi$ as follows

$$C = D - z_0 \mathbf{1}$$

(2.24)
\[ A = \left( E^2 + F^2 \right)^{1/2} \]  
\[ \phi = \arctan2_e(E, F) - \theta_0 \]  
\[ \arctan2_e \equiv \text{element-wise two-argument arctangent} \]  

8. \( z_0 \) and \( \omega \) are of course not exactly constant, so there will be some error in our fit of (2.23) to the data \( d(t) \). There are several reasons why \( z_0 \) and \( \omega \) may vary. We explore some of the subtle ones in Section 2.6. However, a major problem is that when we spin the rotor in step 4 we use a metrology system built on only approximate guesses for \( C, A, \) and \( \phi \). We just computed new values for these parameters, and presumably these new values are better than the guesses we had used. So if the error in the fit of (2.23) to \( d(t) \) is “too large”, we should return to step 4 and repeat the spin-and-fit process, using a metrology system built with the updated parameters.

How do we define what it means for the fit error to be too large? One method is to simply wait until the absolute value of the error between the fitted model and the data \( \epsilon(t) \) is everywhere less than some threshold \( \epsilon_{\text{thresh}} \):

\[ \max_{i, t} |\epsilon_i(t)| < \epsilon_{\text{thresh}} \]

\[ \epsilon(t) = d(t) - \hat{d}(t) \]

In order to get the best fit we would like to set \( \epsilon_{\text{thresh}} \) as low as we possibly can. However, the mirror we are using is only flat to \( \Lambda/4 \), or about 158 nm, so we should not expect the data to fit the model much closer than this. Indeed, we have found that it’s feasible to set \( \epsilon_{\text{thresh}} = 150 \) nm. We typically observe about five iterations of steps 4 to 8 using this criteria.

We should point out that the above calibration procedure holds only for measurement beams \( i \) where \( R_i \neq 0 \). In our apparatus, beam c is specifically aligned with the axis of motion, so \( R_c = 0 \). Thus we cannot use the above procedure to measure \( C_c, A_c, \) or \( \phi_c \). Our computations are designed not to require \( A_c \) or \( \phi_c \). Knowing \( C_c \)
would be equivalent to knowing \( z_0 \). This would be nice, because it would allow our computations to return an absolute value for \( z \). In the absence of this information we have a relative measurement only in \( z \). Given that our sensor is based on displacement interferometer optics, which are inherently relative, we probably should not expect more than this. The relative measurement can still be readily converted to an absolute measure by the addition of an index sensor.

The situation is somewhat better for the constant \( \theta_0 \). For the purpose of fitting \( \hat{d} \) to \( d \), we can simply define \( \theta_0 \equiv 0 \). However, the parameters \( \phi \) that we then recover will be shifted by \( \phi_a \) (the angular position of measurement beam \( a \)), i.e.

\[
\phi_{\text{recovered}} = \phi_{\text{actual}} + \phi_a \]

So again, if we somehow knew \( \phi_a \) then we would have an absolute measure of \( \theta \), and without it we have only a relative measure. We can account for this if we know \( a \) priori at least one of the measurement beam angular positions in world coordinates, \( \phi_i^{\text{known}} \). Then we will always be able to compute \( \phi_a \):

\[
\phi_a = \phi_i^{\text{recovered}} - \phi_i^{\text{known}}
\]

We could measure \( \phi_i^{\text{known}} \) with some external metrology system, but this inconvenient procedure would have to be repeated if we ever changed the alignment of measurement beam \( i \). On the other hand, it may be acceptable to simply define \( \phi_i^{\text{known}} \equiv \phi_i^{\text{recovered}} \). This is the solution we are currently using. If this is not acceptable it’s of course still possible to resolve the issue by adding an index sensor.

### 2.5 Resolution

As shown in Table 2.1, we set our goals for resolution rather high. Our goal for \( \theta \), 5 \( \mu \text{rad} \), is especially difficult to attain—it’s comparable to the resolution of some of the best absolute encoders currently available [36]. The practical \( \theta \) resolution that we attained, 63 \( \mu \text{rad} \), does not quite reach our goal. As we describe below, our practical
resolution is limited by the non-repeatable noise we observe in our data. We may be able to make some refinements to our apparatus that would improve resolution, for example increasing the tilt of the mirror as described in Section A.3.

As we describe in Appendix A, the Hewlett Packard 10706A Plane Mirror Interferometer (PMI) optics [25] we use are capable of measuring displacements with a resolution of \( \lambda/1024 \), or about 0.625 nm, when combined with our HP 10780A receivers and HP 10897A axis interface electronics [24]. However, we have not yet worked hard to isolate our setup from external noise sources like vibration, temperature variations, and air currents, nor have we characterized the error motions of the measurement mirrors, so we are currently observing about 3 nm peak-to-peak noise in the data.

Since we are using beam c to measure \( z \) directly, our sensor’s resolution in this degree of freedom is theoretically about 0.625 nm. In practice the 3 nm p-p noise limits our resolution in \( z \) to the same amount, about 3 nm.

The story is not so simple for \( \theta \), which we compute according to equation (2.12). In our apparatus \( \phi_a - \phi_b \equiv \phi_d = 1.55 \text{ rad} \approx \pi/2 \), and \( A_a \approx A_b \equiv A = 135 \mu \text{m} \), so we can use the simpler equation (2.11) to reason about \( \theta \) resolution approximately. Let’s think about the two-argument arctangent function geometrically, as shown in Figure 2-8. We can consider the inputs to arctan2, \( a \) and \( b \), to be the lengths of the legs of a right triangle, and the result of arctan2(\( a, b \)), \( \theta \), to be the angle opposite the leg of length \( a \). Let \( c \) be the length of the hypotenuse. When we combine the formulas for \( a \) and \( b \) given in (2.11) with (2.9) and (2.10) we get

\[
\begin{align*}
  a &= A \sin(\theta) \\
  b &= A \cos(\theta) \\
  c &= \sqrt{a^2 + b^2} = A
\end{align*}
\]

If \( a \) and \( b \) each have uncertainty \( \varepsilon \), they define not one but an infinite set of right triangles. Then the uncertainty in arctan2(\( a, b \)) is the difference between the maximum and minimum value of \( \theta \) over all triangles in this set. The size of this
uncertainty depends not only on $\varepsilon$, but also on the range of variation in the set of triangles. The worst case for this range occurs when $\theta \approx k\pi/4$, where $k$ is any integer. As shown in the figure, the range of variation in this case is about

$$\arctan\left(\frac{\varepsilon\sqrt{2}}{c}\right) = \arctan\left(\frac{\varepsilon\sqrt{2}}{A}\right)$$  \hspace{1cm} (2.27)$$

Figure 2-8: Make a right triangle with legs of length $a$ and $b$. If the inputs to $\arctan2$ are $a$ and $b$ then we can consider the output $\theta$ to be the angle in the triangle opposite the leg with length $a$, as shown on the left. If $a$ and $b$ each have uncertainty $\varepsilon$, they define not one but an infinite set of right triangles, a few of which are shown in the center figure. The uncertainty in $\arctan2(a, b)$ is the difference between the maximum and minimum value of $\theta$ over all triangles in this set. The worst case for this difference occurs when $\theta \approx \pi/4$, shown on the right.

Since $a$ and $b$ each contain two measurement data terms, $\varepsilon$ is given by twice the measurement resolution. So in theory $\varepsilon \approx 1$ nm, but the noise in our setup limits us to $\varepsilon \approx 6$ nm in practice. Applying these values to (2.27), we get that our $\theta$ resolution is about $10 \mu$rad (about 19 bits) in theory, and $63 \mu$rad (about 17 bits) in practice.

From (2.27) it should be clear that in order to maximize $\theta$ resolution we should try to maximize $A_a$ and $A_b$. Since $A_i = R_i \tan(\alpha)$, we can maximize $A_i$ both by maximizing the radial positions of the measurement beams $R_i$ and by maximizing the mirror tilt $\alpha$. Unfortunately we are limited by design trade-offs in both of these parameters.
The larger we make $R_i$, the larger the diameter of tilted mirror we need to use. Since the inertia of the mirror is proportional to the fourth power of its radius, we would like to keep its diameter as small as possible. We are currently using a 7.62 cm (3 inch) diameter mirror, with a mass of 190 g and inertia of $1.37 \times 10^{-4}$ kg·m². With this mirror we are able to achieve $R_a \approx R_b \approx 2.87$ cm.

We are also limited in the amount that we can tilt the mirror. The interferometer optics we are using are designed to accommodate only small tilts in the measurement mirror. If the tilt is too large then the returned measurement beam will not strike the detector. We describe this situation in detail and suggest some possible work-arounds in Section A.3.

The maximum recommended tilt is an inverse function of the distance from the interferometer optics to the mirror as discussed in Section A.2. This function is not precisely specified by Hewlett Packard, but only given as a set of three sample values (plotted in Figure A-4). We extrapolate from this set that the recommended maximum tilt at our maximum working distance (about 3 cm) is about 2.8 mrad. We are currently using the somewhat larger tilt of $\alpha \approx 4.7$ mrad, which seems to work.

2.6 Accuracy

We have tried to build our sensor so that it has a high resolution, but we have not eliminated all sources of error. We discussed the non-repeatable errors (external disturbances and error motions of the measurement mirrors) above. We consider these to reduce the inherent resolution of the sensor.

However, there are also several sources of repeatable error which we have not yet mentioned. Because these errors are repeatable, we do not claim that they reduce resolution because it’s theoretically possible that we can measure and correct for them. We have not attempted to make such measurements or corrections, but as a starting point we list several repeatable error sources below.

**flatness deviations of the large tilted mirror** As we mentioned, this mirror is only specified to be flat to $\lambda/4$, or about 158 nm. There are a number of ways
that we could derive a height map of the mirror. An especially interesting idea is to consider the calibration fit error for each measurement beam \( i \), \( \epsilon_i(t) \) (defined above), combined with the angular trajectory the mirror took when that error was measured, \( \theta(t) \), to be a height map of the mirror for that measurement beam. This may be reasonable because if the model \( \hat{d}_i(t) \) is fitted very well then we can actually consider it to be a good prediction of the response a perfectly flat mirror would have. Figure 2-9 plots the calibration fit error for measurement beam \( a \), \( \epsilon_a(t) \) (defined above), across multiple runs. There seems to be some amount of correlation in these error functions from run to run. We have not explored this any further.

![Fit Error vs \( \theta \) (Two Runs)](image)

Figure 2-9: The calibration fit error for measurement beam \( a \), \( \epsilon_a(t) \), plotted for two runs. Some correlation is apparent. These plots may give us some information about the flatness deviations of the mirror, but we have not explored this further.

**deviations from the assumption that beam c measures \( z \) only** Errors of this type may arise either because beam c is not exactly aligned with the axis of motion, or because the auxiliary mirror that beam c reflects against is not exactly normal to the axis of motion, or both. As we suggested above, we
may be able to avoid these problems if we replace the auxiliary mirror with a retroreflector.

**deviation in the pointing angle of the measurement beams**  We have so far assumed that the measurement beams are aligned to be exactly parallel to the axis of motion. While it is possible to achieve very good alignment, some error is sure to remain. Such errors are common to most applications of displacement interferometry, and their effects on our sensor should not be much worse than for others.

### 2.7 Slew Rates

The PMI optics we use have a maximum rated slew rate of 254 mm/sec. Effectively, this means that the time derivatives of the responses \( d_a \) and \( d_b \) are limited:

\[
\dot{d}_{a,b} = \frac{dd_{a,b}}{dt} = \left( \frac{\partial d_{a,b}}{\partial z} \right) \dot{z} + \left( \frac{\partial d_{a,b}}{\partial \theta} \right) \dot{\theta} \leq 254 \text{ mm/sec.} 
\]  

(2.28)

By taking partial derivatives of (2.1), we get

\[
\frac{\partial d_{a,b}}{\partial z} = 1 \]

\[
\frac{\partial d_{a,b}}{\partial \theta} = A_{a,b} \cos(\theta + \phi_{a,b}) \leq A_{a,b} \approx 135 \mu \text{m/rad.}
\]

So we can simplify (2.28) to

\[
\dot{z} + (135 \mu \text{m}) \dot{\theta} \leq 254 \text{ mm/sec.} \tag{2.29}
\]

The maximum possible \( z \) slew rate will thus be

\[
\dot{z}_{\text{max}} = 0.25 \text{ m/s} \approx 254 \text{ mm/sec},
\]
which will be possible when $\dot{\theta} = 0$. Similarly, the maximum possible $\theta$ slew rate will be

$$\dot{\theta}_{\text{max}} = 1800 \text{ rad/s} \approx (254 \text{ mm/sec}) / (135 \text{ \mu m}),$$

which is possible when $\dot{z} = 0$.

### 2.8 Experimental Apparatus

As we have said, our implementation of the tilted-mirror interferometer is integrated into the $z$ and $\theta$ axes of the new 5DOF CNC grinding machine that we are building (Section 1.3).

Our implementation is shown in the photo in Figure 1-2 and in the CAD diagram in Figure 2-10. We use three two-pass Plane Mirror Interferometer (PMI) optics (Appendix A), which produce the three measurement beams, a 7.62 cm (3 inch) diameter $\lambda/4$ flat first-surface mirror, and an auxiliary 2.54 cm (1 inch) diameter $\lambda/20$ flat first-surface mirror\(^4\). The larger mirror is tilted at about 4.7 mrad with respect to the normal plane of the axis of motion. The moving mass of our sensor is about 335 g, and the inertia is about $2.54 \times 10^{-4}$ kg $\cdot$ m$^2$.

![Figure 2-10: Our implementation of the tilted-mirror interferometer.](image)

We mount PMI optics a and b side-by-side, as shown in Figure 2-11. We orient

\(^4\)We use a HeNe laser, so $\lambda \approx 633$ nm.
the measurement plane to make it parallel with the xy-plane, so ideally $\psi = 0$, and we position the interferometers symmetrically at a radius of about 2.87 cm so that, ideally, $\phi_a = 3\pi/4$ and $\phi_b = \pi/4$. Deviations in our actual apparatus from these ideals are accommodated by the calibration procedure described in Section 2.4.

Figure 2-11: We mount PMI optics a and b side-by-side, with the measurement plane parallel to the xy-plane.

We choose this mounting, rather than one in which the two PMI are staggered in z, so that each PMI can be as close as possible to the mirror. This minimizes deadpath. Also, this lets us achieve the maximum possible mirror tilt $\alpha$, since the maximum tilt is inversely proportional to the distance between the mirror and the farthest interferometer optics (Appendix A).

The mount for the large tilted mirror is shown in the left and center parts of Figure 2-12. This is an adjustable tilt mount with three fine-pitch thrust screws for adjustment and a central spring steel web. The adjustment screws are placed 120° apart and are mounted so that they push in the axial direction against the mirror. Thus, they kinematically define the orientation of the mirror. The radial location of the mirror (i.e. the mirror’s centering) is constrained by the way it is bolted to the spring web. The axial location of the mirror is constrained by the preload force that
the spring web provides, which pulls the mirror against the adjustment screws.

The mirror we use is made of fused silica. It weighs about 191 g and has an inertia of about $1.39 \times 10^{-4}$ kg·m², so it makes up a large part of the total moving mass and inertia of our sensor. It may be possible to use a lighter-weight mirror, for example one with a honeycomb substrate, but we have not explored this further.

![Figure 2-12: Close-up views of the mirror mounts in our implementation. The adjustable tilt mount for the large tilted mirror is shown on the left and exploded in the center. The adjustable tilt mount for the auxiliary mirror is shown on the right.](image)

The mount for the small auxiliary mirror is shown on the right in Figure 2-12. This is also an adjustable tilt mount very similar to the large mount, but with a much smaller angular travel. Because the travel of this mount is so small, we fabricated it entirely from a single piece of aluminum instead of using a separate spring steel web. Angular adjustment is allowed by a small flexure cut into the aluminum. We use six adjustment screws on this small mount, unlike the large mount, which uses only three screws. Three of the screws are mounted to push, and staggered between these are an additional three screws that are mounted to pull. This arrangement allows us to achieve better fine-tuning of the angular position.

### 2.8.1 Instrumentation

The instrumentation structure of our apparatus is shown in Figure 2-13. We use three Hewlett Packard 10897A High Resolution VMEbus Laser Axis boards to interpolate,
digitize, and accumulate the readings from the three measurement beams. This data is then relayed by a C40 DSP across a high-speed (≈ 1 MB/s) data link to a dSPACE 1103 digital control prototyping system installed in a PC. We describe this custom-built link in more detail below.

We have developed a control system for the $z$ and $\theta$ DOF of the 5DOF grinding machine using MATLAB/Simulink/Stateflow. This control system runs in real-time at 10 kHz on the dSPACE 1103 using metrology data from the tilted-mirror interferometer. Reference [21] provides details on the control system. The run-time user interface, developed using dSPACE ControlDesk, is pictured in Figure 2-14.

![Figure 2-13: The instrumentation structure of the tilted-mirror interferometer.](image)

There are also several software components to our implementation. These are described in Appendix B.

**C40 to dSPACE 1103 Data Link**

The HP 10897A laser axis boards mount in a VMEbus. We need to get the raw metrology data from these boards to the dSPACE 1103, where the machine control system runs. Each board produces a 32-bit position value, we have three boards, and our control system runs at 10 kHz, so we need a communication bandwidth of about $3 \times 32 \text{ bits} \times 10 \text{ kHz} = 960 \text{ kbps} = 120 \text{ kbps}$.

Unfortunately, the 1103 system does not come with a convenient link to the VMEbus, so we need to implement this link on our own. Since we also happen to have
Figure 2-14: Screenshot of the run-time user interface for the $z$ and $\theta$ DOF of the 5DOF grinding machine. We implemented this interface in the dSPACE ControlDesk environment.
a Texas Instruments TMS320C40 DSP installed on the VMEbus (on a Pentek 4284 board), the simplest solution is to use the C40 to gather data from the axis boards and feed it to the dSPACE system. The C40 has several communications (COM) ports which are brought out to front-panel connectors on the 4284\(^5\). These are 8-wire parallel ports designed to link multiple C40 systems together.

The dSPACE 1103 does not have a corresponding port, but we were able to develop software that runs on the 1103 and emulates a C40 COM port using the board’s general-purpose digital I/O. As shown in Figure 2-15, this reduces the hardware of the data link to a simple parallel cable connection between the C40 and the 1103. We use RS-422 balanced-line transmission on the cable to reduce noise.

![Block Diagram of the C40 to dSPACE 1103 Data Link](image)

Figure 2-15: Block Diagram of the C40 to dSPACE 1103 Data Link. The 1103 requests data from the C40 by activating the /Trigger line. Upon receiving this signal, the C40 gathers the position values from the three 10897A laser axis boards over the VMEbus. The C40 then assembles this data and a checksum into a packet, and sends the packet to the 1103 over the data lines D0–D7. A two-wire handshake, implemented with the /Strobe and /Ready lines, is used to send each byte of the packet. An additional Error signal is used by the 4284 to tell the 1103 if the laser hardware is currently reporting any error condition.

The data link operates asynchronously. The 1103 requests data from the C40 by activating the /Trigger line. Upon receiving this signal, the C40 gathers the position values from the three 10897A laser axis boards over the VMEbus. The C40 then assembles this data and a checksum into a packet, and sends the packet to the 1103 over the data lines D0–D7. A two-wire handshake, implemented with the /Strobe

\(^5\)These are not the same as the COM ports on a PC.
and /Ready lines, is used to send each byte of the packet. An additional Error signal is used by the 4284 to tell the 1103 if the laser hardware is currently reporting any error condition. Details of the software for this system are provided in Appendix B, and schematics for the RS-422 level-translation hardware are located in Appendix C.

2.9 A Concrete Example of the Two-Beam Ambiguity

In Section 2.2, we showed that there is an unavoidable binary ambiguity in any tilted mirror system that has only two measurement beams. Our implementation avoids this by using beam c in addition to beams a and b. However, now that we have described the concrete details of our implementation, it’s instructive to consider how the ambiguity would manifest itself if we had only beams a and b.

As we described in Section 2.3.1, there will in general be two possible solutions, \((z, \theta)\) and \((z', \theta')\), for any measured values of the PMI responses \(d_a\) and \(d_b\). Only one of these solutions is actually the true position of the machine. We need to pick the first solution in one operating region of the rotary travel of the machine and the second otherwise.

There is a clean boundary between the two solution regions, namely the points at which \(\theta = \psi \pm \pi/2\), or \(\theta = \pm \pi/2\) in our implementation, since \(\psi = 0\). A good way to understand the ambiguity is to consider a situation where the mirror is positioned at one of these boundaries. In these locations, a small translation-only movement of the machine is indistinguishable (given only the measurements \(d_a\) and \(d_b\)) from a small rotation-only movement.

Let’s differentiate the instance of (2.1) for each measurement beam with respect to \(\theta\), and then evaluate the results at \(\theta = \pm \pi/2\):

\[
\frac{\partial d_a}{\partial \theta} = A \cos(\theta - \phi_a)
\]

\[
\frac{\partial d_a}{\partial \theta} \bigg|_{\theta=\pm \pi/2} = A \cos(\pm \pi/2 - \pi/4)
\]
\[ \frac{\partial d_a}{\partial \theta} \bigg|_{\theta=\pm\pi/2} = \pm A \frac{\sqrt{2}}{2} \]

\[ \partial d_a = \pm A \frac{\sqrt{2}}{2} \partial \theta \]

(similarly) \[ \partial d_b = \pm A \frac{\sqrt{2}}{2} \partial \theta \]

So we see that when \( \theta = \pm \pi/2 \) the derivatives of the two responses, \( d_a \) and \( d_b \) with respect to \( \theta \) are equal, i.e. \( (\partial d_a)/(\partial \theta) = (\partial d_b)/(\partial \theta) \). Another way to see this is to consider Figure 2-16, which is a graph of \( d_a \) and \( d_b \) as \( z \) is held constant and \( \theta \) smoothly increases from 0 to \( 4\pi \). In the Figure, orientations where the slopes of the two plots are equal are marked by dotted vertical lines.

Interferometer Responses \( d_a \) and \( d_b \) (Ideal) vs \( \theta \)
(z Held Constant)

Figure 2-16: A graph of \( d_a \) and \( d_b \) as \( z \) is held constant and \( \theta \) smoothly increases from 0 to \( 4\pi \). Orientations where the slopes of the two plots are equal are marked by dotted vertical lines.

If we continue by differentiating (2.1) with respect to \( z \), we find that

\[ \frac{\partial d_a}{\partial z} = 1 \]
\[ \partial d_a = \partial z \]

(similarly) \[ \partial d_b = \partial z \]

So when \( \theta = \pm \pi/2 \) a translation-only movement of \( \partial z \) will make the same change in the observed PMI responses \( d_a \) and \( d_b \) as a rotation-only movement of \( \partial \theta = \pm \sqrt{2} \partial z \).

Figure 2-17 shows an idealized version of our apparatus, with the mirror positioned at \( \theta = \pi/2 \). If we imagine a small rotation in the positive \( \theta \) direction (clockwise) then both \( d_a \) and \( d_b \) will decrease, and by the above analysis, they will do so by the same amount in the limit where the rotation is infinitely small. Thus, this movement will be indistinguishable from a translation, so we are at a singular point. The situation is symmetric for a small rotation in the negative \( \theta \) direction.

![Figure 2-17](image)

Figure 2-17: An idealized version of our apparatus, with the mirror positioned at \( \theta = \pi/2 \). If we imagine a small rotation in the positive \( \theta \) direction (clockwise) then both \( d_a \) and \( d_b \) will decrease, and they will do so by the same amount in the limit where the rotation is infinitely small. Hence, this movement will be indistinguishable from a translation.

### 2.10 Summary

In this chapter we described an interferometric sensor for cylindrical motion called the tilted-mirror interferometer. We developed the mathematical and algorithmic details of this sensor in the first part of this chapter. In the latter part of the chapter we
described our experimental apparatus, which integrates the sensor into the $z$ and $\theta$ DOF of the new 5DOF CNC grinding machine described in Section 1.3.

The performance of our implementation, summarized in Table 2.1, appears to be reasonable. One major issue is that the practical resolution in $\theta$ is an order of magnitude lower than our goal. As we have described, it may be possible to improve this by reducing noise levels in the system and/or by applying one of the methods described in Section A.3 for increasing the allowable tilt of the mirror.

As we mention in Section 2.8, we have been able to build a control system for the $z$-$\theta$ axis of our new 5DOF grinding machine using the tilted-mirror interferometer to provide metrology data. At the time of this writing we are still working on improving this control system, and as we said, we have not yet constructed most of the other parts of the machine. This makes it difficult to predict how the final system might perform: How fast, and how accurately, will the machine be able to cut parts? We can only attempt to make the following rough prediction: If the issues of accuracy listed in Section 2.6 are addressed to a sufficient degree, then it may be possible to cut parts with an accuracy that is some not-too-small fraction (perhaps $1/10$) of the practical resolution of our implementation of the sensor.

Is the tilted-mirror interferometer a better solution for measuring rotary linear motion than the other possible approaches described in Sections 1.5 and 5.2? The answer is that it seems likely that it is better than some, but it’s far from clear that it’s the best solution. For example, an optical encoder technique like that described by Chitayat et al in [5] seems like an especially promising alternative solution for practical implementation.
Chapter 3

Metrology Technique B: The Helicoid-Mirror Interferometer

The tilted mirror interferometer described in Chapter 2 has proven to be a reasonable approach to cylindrical metrology using laser interferometers, and it is what we implemented in our experimental system. However, as we describe in Section 2.2, there are fundamental singularities in the setup which we were only able to overcome by using three measurement beams. This is somewhat unsatisfying, because the cylindrical motion we are measuring has only two degrees of freedom, so the minimum possible number of measurement beams is two. Each measurement beam adds complexity and cost to the metrology system, so we would ideally like to find an arrangement which uses as few beams as possible.

Does there exist some optical arrangement for the interferometric measurement of cylindrical motion—without singularities—that uses only two measurement beams? We have not found one. However, it seems that we can do better than the tilted mirror interferometer, the two-beam version of which has two singular orientations. In this chapter we will describe the helicoid-mirror interferometer, an optical setup for cylindrical metrology based on interferometry, which uses only two measurement beams, and which appears to have only one singular orientation.

All of the material we present in this chapter is theoretical. We have not attempted to construct an implementation of the helicoid-mirror interferometer in this work. As
we describe in Section 3.4, there are a number of issues yet to be resolved before we can begin construction. Chief among these is the fact that the helicoid surface is not locally flat, so it will induce aberrations into the measurement beams that reflect against it.

A helicoid is a surface in the shape of a helix [13, 23]. For example, a spiral staircase is a rough approximation to a helicoid. The approximation improves as the height of each step is reduced and the number of steps is simultaneously increased in order to maintain the overall height of the staircase. A helicoid can be described parametrically as follows:

\[
\vec{h} = (x_h, y_h, z_h) = (u \cos v, u \sin v, pv),
\]

(3.1)

where \( p \) is the pitch of the helicoid, \( u \) is the parameter in the radial direction, and \( v \) is the parameter in the annular (i.e. circumferential) direction. The axis of a helicoid is like the pole at the center of the spiral staircase. The assignment of coordinates in (3.1) makes \( \hat{z} \) (i.e. the \( z \) coordinate axis) the axis of the helicoid.

The pitch angle \( \alpha \) of a helicoid, taken at some radius \( u = R \), is given by

\[
\alpha = \arctan \left( \frac{p}{R} \right).
\]

(3.2)

This is the maximum angle between the surface of the helicoid and the \( xy \) plane (i.e. the plane perpendicular to the axis of the helicoid) at radius \( R \).

The helicoid-mirror interferometer is based on a first-surface mirror formed in the shape of a helicoid. Multiple laser paths then measure the \( z \)-height of the helicoid as a function of rotation and translation.

### 3.1 Overview

The basic concept of the helicoid-mirror interferometer, shown in Figure 3-1, is similar to the tilted-mirror interferometer (compare with Figure 2-1). The fundamental difference is that, instead of a plane mirror tilted at the angle \( \alpha \) with respect to the
normal plane of the axis of motion, a mirror in the form of a precision helicoid is mounted to the end of the moving machine part.

Figure 3-1: The helicoid-mirror interferometer concept (figure not to scale). A mirror in the form of a precision helicoid is mounted to the end of the moving machine part. Two measurement beams are used: beam $z$ and beam $q$. Each beam is formed by a set of Plane Mirror Interferometer (PMI) optics which are fixed to the machine frame. Beam $z$ measures against the central flat mirror, and beam $q$ measures against the helicoid mirror at radius $R$.

The helicoid is only formed in an annulus about some radius $R$, and at this radius it has ramp angle $\alpha$. A central flat mirror, mounted perpendicular to the axis of motion, is located inside the annulus. For convenience in our analysis, we will assume that the plane of the flat mirror coincides with the plane containing the low point of the helicoid surface, but this is not a requirement.

Two measurement beams are used: beam $z$ and beam $q$. Each beam is formed by a set of Plane Mirror Interferometer (PMI) optics (see Appendix A for details on the PMI) which are fixed to the machine frame. Beam $z$ measures against the central flat mirror and beam $q$ measures against the helicoid mirror at radius $R$.

As we discuss in Section 3.2, it does not appear to be very difficult to recover $z$ and $\theta$ from the responses of PMI $z$ and $q$, $d_z$ and $d_q$. We will also show in that section
that there are no singularities in this system other than at the discontinuity in height on the helicoid.

In Section 3.3, we go on to propose some parameters for what might be a practical implementation of the helicoid-mirror interferometer. We analyze this proposed implementation and derive estimates of its range of motion, resolution, and maximum slew rates.

However, we have not attempted to actually construct any physical implementation of the helicoid-mirror interferometer. There seem to be some major issues which we have not yet fully resolved, which we enumerate in Section 3.4. These include several optical aberrations, which may thwart the operation of the PMI, as well as practical issues such as the feasibility of fabricating the helicoid mirror itself.

3.2 Recovering \( z \) and \( \theta \)

In this section we show how to compute \( z \) and \( \theta \), the position and orientation of the moving machine part, from the measurements \( d_z \) and \( d_q \). We present an ideal analysis, in the sense that we assume that each beam simply measures the geometrical relationship we intend it to measure. In a real machine this may not be entirely true due to misalignments and the effects of noise sources on the measurements.

Beam \( z \) measures against a flat mirror that is perpendicular to the axis of motion, so its response will not depend on \( \theta \). According to (A.1), the response will be

\[
d_z = C_z + z,
\]

(3.3)

where \( C_z \) depends on where the interface electronics for interferometer \( z \) has been zeroed.

Beam \( q \), on the other hand, measures against the helicoid mirror at radius \( R \), so its response will depend on both \( z \) and \( \theta \):

\[
d_q = C_q + z + \theta R \tan \alpha,
\]

(3.4)
where $C_q$ depends on where the interface electronics for interferometer $q$ has been zeroed.

From (3.3) it's clear that we can recover $z$ directly:

$$z(d_z) = d_z - C_z. \quad (3.5)$$

To recover $\theta$ we subtract (3.3) from (3.4). Again, it will be convenient to define a new variable $\delta$:

$$\delta = (d_q - C_q) - (d_z - C_z). \quad (3.6)$$

Geometrically, $\delta$ is the height of the helicoid surface at the measurement point of beam $q$. This height is measured with respect to the low point of the helicoid because we have specified that the plane of the central flat mirror, which beam $z$ measures against, coincides with that low point. Given this definition,

$$\theta(d_q, d_z) = \frac{(d_q - C_q) - (d_z - C_z)}{R \tan \alpha}$$

$$\theta(\delta) = \frac{\delta}{R \tan \alpha} \quad (3.7)$$

As for the tilted-mirror interferometer of Chapter 2, we will have only relative measurements if we do not know the constants $C_q$ and $C_z$. And again, $R \tan \alpha$ is a constant that depends on the exact construction of the metrology system. Unfortunately, unlike for the tilted-mirror interferometer, we have not developed any calibration routines for the helicoid-mirror interferometer that can automatically measure $C_z$, $C_q$, or $R \tan \alpha$.

Importantly, because both (3.5) and (3.7) are linear, there will be no singularities in the recovery of $z$ and $\theta$ from $d_z$ and $d_q$.

### 3.3 A Design Feasibility Analysis

For the sake of discussion, let's consider an implementation of the helicoid-mirror interferometer with $R = 3.81$ cm (1.5 in) and $\alpha = 4.18$ mrad. This gives a pitch of
\[ p = R \tan \alpha = 0.159 \text{ mm/rad}, \] so the difference in height between the high and low points of the helicoid mirror will be \( 1 \text{ mm} = 2\pi p \).

Let’s also consider using the same PMI setups as we do in the tilted-mirror interferometer: HP 10706A PMI optics, HP 10780A receivers, and HP 10897A axis interface electronics. The details of these setups are given in Appendix A, and are summarized in Table A.1.

### 3.3.1 Range of Motion

As we described in Section 1.4, we would like our sensor to accommodate \( \pm 1.5 \text{ cm} \) travel in translation and \( \pm \pi \text{ rad} \) travel in rotation. Can we reasonably expect to achieve these travels with the implementation of the helicoid-mirror interferometer that we are considering here?

It is unlikely that we would be able to fabricate the helicoid mirror so that the low point of its ramp begins exactly where the high point left off. As shown in Figure 3-2, there will probably be a “dead zone” of some angular width \( \gamma \), say \( \gamma = 10^\circ \approx 0.175 \text{ rad} \). We cannot allow the measurement beams of PMI \( q \) to enter this dead zone, so, combined with the 6 mm diameter of the measurement beams, it will limit our range of motion.

As we describe in Appendix A, each of the PMI optics we use actually sends two measurement beams to the target mirror, spaced 12.7 mm apart. Normally, we don’t consider each beam separately because their combined operation is equivalent to the operation of a single virtual beam located halfway between them. In this situation, though, we should make sure that the two measurement beams of PMI \( q \) are oriented along a radius of the helicoid. This way we will be able to approach as close as possible to the dead zone, as shown in Figure 3-2. With this arrangement we should be able to achieve a rotary travel of about \( 2\pi - \gamma - (\text{beam diameter})/R \approx 6 \text{ rad} \approx 340^\circ \).

The \( z \) travel is limited by the fact that the angular acceptance (i.e. the acceptable tilt of the measurement mirror) of the PMI optics drops off as a function of the distance. As we discuss in Section A.2, we extrapolate from the data that Hewlett Packard specifies that the maximum allowable tilt at the maximum of our \( z \) travel
We should make sure that the two measurement beams of PMI q are oriented along a radius of the helicoid so that we will be able to approach as close as possible (given the spacing between them, and also their finite diameters) to the dead zone.

(3 cm) is about 2.8 mrad. However, in Section 2.5 we mentioned that our implementation of the tilted-mirror interferometer has shown that the PMI can be reliably used with tilts approaching 5 mrad at this distance. Thus PMI q should be able to accommodate the pitch angle of the helicoid, $\alpha = 4.18$ mrad, over the full range of $\pm 1.5$ cm travel in $z$.

### 3.3.2 Resolution

The resolution in translation of helicoid-mirror interferometer is the same as that of the PMI itself, i.e. 0.625 nm in theory, which is much better than the resolution we said we required in Section 1.4.

We can compute the resolution in rotation by dividing the total range of $\delta$, as $\theta$ varies from 0 to $2\pi$ rad, by the uncertainty in $\delta$. The range is just the difference in height of the helicoid between its highest and lowest points, which for the specific setup we are considering is 1 mm. In theory, the uncertainty is twice the PMI resolution, or about 1.25 nm, because delta depends on two PMI response measurements. Thus, the resolution in $\theta$ is $8 \times 10^5$ counts/(2$\pi$ rad) = (1 mm)/(1.25 nm), or about 8 $\mu$rad.
per count\textsuperscript{1}. This exceeds the $\theta$ resolution we said we required in Section 1.4, 5 $\mu$rad, but it’s about the same as the theoretical resolution of our implementation of the tilted-mirror interferometer, which is 10 $\mu$rad.

### 3.3.3 Slew Rates

The maximum allowable slew rate in translation will be the same as the raw PMI, or about 0.25 m/s. This is the same as the tilted-mirror interferometer, though somewhat less than the target slew rate in translation that we specified in Section 1.4 (1 m/s).

For rotation, we need to divide the maximum PMI slew rate by $\partial d_\theta/\partial \theta$. Differentiating (3.4),

$$\frac{\partial d_\theta}{\partial \theta} = R \tan \alpha.$$  

For the setup we are considering, $R \tan \alpha \approx 1.59 \times 10^{-4}$ m/rad. Thus the maximum slew rate in rotation will be about $1500$ rad/s $\approx (0.25 \text{ m/s})/(1.59 \times 10^{-4} \text{ m/rad})$. This is comparable to the capability of the tilted-mirror interferometer (1800 rad/s), and greatly exceeds the maximum slew rate we said we required in Section 1.4 (50 rad/s).

### 3.4 Unresolved Issues

Thus, it seems that an implementation of the helicoid mirror interferometer could potentially meet most of the performance requirements we specified in Section 1.4. Unfortunately all is not perfect with the helicoid-mirror interferometer. There are several issues which remain to be resolved before we could attempt to actually construct it.

#### 3.4.1 Optical Aberrations

A major concern is that, since the helicoid surface is not flat, it may induce optical aberrations into the measurement beams. If these aberrations are severe enough,
they could interfere with the proper operation of PMI q. Here, again, we need to think about both beams of PMI q, rather than considering a single virtual beam (Appendix A).

There seem to be two main issues:

**the helicoid surface normals at each beam are not coincident** The beams from PMI q hit the helicoid surface at different radii ($R \pm 6.35$ mm, as shown in Figure 3-2). Thus they will be reflected in slightly different directions. This is because the surface normal of the helicoid is not constant with respect to the radial surface parameter $u$. If the difference is large enough, then the measurement beam may fail to overlap sufficiently with the reference beam at the PMI detector, which would cause the PMI to fail.

**the helicoid surface is not flat over the area of each beam** We are using the helicoid specifically because it is not globally flat. However, because it is a smooth surface, this means that the helicoid will neither be flat locally over the 6 mm diameter of the measurement beams. Thus, the wavefronts of the reflected measurement beams will not be planar. This may cause intra-beam fringes to appear when interference occurs between the measurement and reference beams at the PMI detector. These fringes could be confused by the detector with fringes that occur because of legitimate path-length differences between the measurement and reference beams (which the detector is supposed to measure), resulting in faulty displacement readings.

It may be possible to combat these problems using some combination of the following techniques:

**decreasing the PMI beam separation** As we state in Appendix A, the PMI that we use has a nominal separation of 12.7 mm between the two passes of the measurement beam. It should be possible to reduce this separation by introducing some small periscope-type prisms into the beam path (or by other means). By doing so, we could minimize the difference in surface normal between the two beams at the helicoid surface.
decreasing the beam diameter It should also be possible to reduce the diameter of the measurement beam from its current value of 6 mm. This would lessen the effects of the local curvature of the helicoid, simply because it would reduce the area of the curved surface that the beams intersect. We could possibly achieve this by purchasing a new laser source that produces a smaller beam and/or using some optical arrangement to reduce the beam diameter.

making \( R \) as large as possible If we simultaneously hold the pitch \( p \) constant, this will result in a shallower pitch angle \( \alpha \), and the helicoid surface will be closer to flat in the local region that interacts with the PMI beams.

using wavefront correction optics Like glasses or contact lenses for a person, it may be possible to fit some type of corrective optics to the PMI that would further reverse the effect on the beam wavefronts of the curvature of the helicoid surface.

3.4.2 Other Problems

Besides the above optical issues, there are some additional problems with the helicoid-mirror interferometer.

A major concern is the fabrication of the helicoid mirror itself. It seems possible to build it using machines typically used to fabricate aspheric optics. However, it seems this would be expensive due to the machine operation time that would likely be required. We investigated this with Dr. William Plummer, Director of Optical Engineering at Polaroid Corporation, who confirmed that building such a precision optic would be expensive. An additional possibility is to approximate the helicoid surface by bending a split ring of some material, but we have not explored this further.

We should also be concerned with the mass and inertia of the moving part of the sensor. Just as in the tilted-mirror interferometer, we have a design trade-off: we are encouraged to make the radius of the helicoid \( R \) as large as possible to minimize the effects of the optical aberrations (as mentioned above), but by doing so we would also increase the inertia of the sensor.
One concept that we have come up with to significantly reduce the inertia of the helicoid-mirror interferometer is shown in Figure 3-3. We call this a “periscope” arrangement, because a system of two small mirrors is mounted to the end of the moving part instead of the helicoid mirror itself. This system bends measurement beam $q$ 180° and displaces it in the radial direction, so its operation is similar in principle to that of a periscope. In this arrangement, the helicoid mirror is mounted rigidly to the frame of the machine, and the periscope optics cause beam $q$ to scan along the mirror. Since only the periscope optics are mounted to the moving machine part, only their mass and inertia will be important, and it’s likely that these figures will be much lower than for an implementation that mounts the helicoid mirror directly to the moving part. As we show in the Figure, it may even be possible to work beam $z$ into this arrangement with a minimal addition of optics.

### 3.5 Summary

In this chapter we proposed the **helicoid-mirror interferometer**, an interferometric technique for measuring cylindrical motion. The helicoid-mirror interferometer is similar to the tilted-mirror interferometer (Chapter 2). The fundamental difference is that, instead of a plane mirror that is tilted with respect to the normal plane of the axis of motion, a mirror in the form of a precision helicoid is mounted to the end of the moving machine part.

The geometry of the two-beam helicoid-mirror interferometer seems to have only one singularity, whereas the two-beam version of the tilted-mirror interferometer has two. Thus, the helicoid-mirror interferometer offers a larger singularity-free range of motion in rotation than the two-beam tilted-mirror interferometer would. Additionally, as we depicted in Figure 3-3, we have found a possible implementation of the helicoid-mirror interferometer that may potentially offer a much lower moving mass/inertia than the straightforward implementation of the helicoid-mirror interferometer. Since the latter is so similar to the implementation of the tilted-mirror interferometer, the alternate implementation of the helicoid-mirror interferometer will
Figure 3-3: The “periscope” concept for reducing the moving mass and inertia of the helicoid-mirror interferometer. Instead of mounting the helicoid mirror directly to the end of the moving machine part, in this configuration a small, low-mass/inertia periscope arrangement of two mirrors is attached. The helicoid mirror is mounted to the fixed part of the machine, and the periscope optics are used to scan measurement beam $q$ (orange) along its surface. We have depicted beam $q$ as a single-beam PMI in this drawing. Beam $z$ is also incorporated into this arrangement (green), and it shares most of the same optics that are used by beam $q$.

We have also likely be lower in mass and inertia than the tilted-mirror interferometer.

While we have not constructed any physical implementation of the helicoid-mirror interferometer, we presented a design feasibility analysis which proposed some parameters that may be reasonable for implementation. We analyzed the theoretical performance of a helicoid-mirror interferometer built according to these parameters, and this analysis shows that such an implementation would meet most of our requirements (given in Section 1.4) at least as well as our implementation of the tilted-mirror interferometer.

There do seem to be some major issues that remain to be resolved before we can attempt to construct the helicoid mirror interferometer, which we discuss at the end of the chapter. These include optical aberrations that may affect the performance of
the PMI, as well as practical considerations like the fabrication of the helicoid-mirror itself.
Chapter 4

Metrology Technique C: Machine Vision

We have thus far described two interferometric techniques for cylindrical metrology: the tilted-mirror interferometer and the helicoid-mirror interferometer. In this chapter we step back from interferometry and consider using techniques from the field of machine vision to build a sensor for cylindrical motion. Most of the material we present is theoretical, though we do describe a hardware experiment in Section 4.2.1.

We first review several vision algorithms which can be applied to cylindrical metrology. These algorithms have been used in the past to measure general 2DOF motion. They are all based on computing the relative registration between successive frames of the surface of the moving object, and are subject to measurement drift as errors accumulate. We continue by outlining a possible scheme—absolute registration—for avoiding such drift for our machine motion-control application.

4.1 Basic Topology

A basic topology for applying vision to cylindrical metrology is a camera held in a fixed position to capture images of the moving machine part, as shown in Figure 4-1.

We will assume the moving part is cylindrical (as it is in our machine). In this case the imaged surface has symmetry such that it will not vary geometrically as
Figure 4-1: A basic topology for applying vision techniques to the cylindrical metrology problem is to position a camera to capture images of the moving machine part. Surface patterns are needed to detect motion because the visible surface of the cylindrical part does not vary geometrically as it rotates and translates.

it rotates and translates. In order to sense motion visually, we consider using an optically patterned surface. Such patterns may already be available at some scale due to the machining operations that produced the surface, or we may be able to use coherent illumination to produce a speckle pattern as described in Section 4.3.

We also assume that we know the radius, $R$, of the cylindrical surface, and that the circumference of the surface is large relative to the aperture of the camera. Thus, when the machine part rotates by the amount $\theta$ and translates by $z$, the image in the camera will translate by the amount $t$, where

$$t = (x_t, y_t) = (z, R\theta). \quad (4.1)$$

If we know $t$ then we can recover $z$ and $\theta$ like this:

$$t = (x_t, y_t)$$

$$z(t) = x_t \quad (4.2)$$

$$\theta(t) = \frac{1}{R} y_t. \quad (4.3)$$

One obvious way to measure $t$ is to compute and accumulate the translations which give the best registrations between pairs of successive frames. We refer to this
class of approaches as *relative registration*, and we discuss it in Section 4.2. A major drawback of relative registration is drift, the potentially large accumulated error due to small errors in each of the computed inter-frame translations.

For our application it seems possible to develop a global map of the entire range of object surface that our sensor will see, because the moving machine part will have a well-defined range of motion. This would allow us to match successive frames not against their predecessors, but against the global map. Thus we should be able to avoid drift. We call this approach *absolute registration* and we discuss it in Section 4.4.

### 4.2 Relative Registration

One obvious way to recover displacement using a topology like that shown in Figure 4-1 is to compute and accumulate the translations $t_i$ which give the best registrations between pairs of successive frames ($I_{i-1}, I_i$). Then the position of the moving machine part at frame $j$ is $\sum_{i \leq j} (z(t_i), \theta(t_i))$. Immediately we can see a major drawback of this relative approach: if each $t_i$ has some small error $\epsilon_i$, which is unavoidable, then the accumulated position will have error $\sum_i \epsilon_i$. This accumulated error—drift—will eventually become greater than our desired resolution.

A common way to compute $t_i$ is to find the maximal *autocorrelation* of $(I_{i-1}, I_i)$. If the frame size is $N^2$ and the search area for the autocorrelation is $M^2$, then this can be expressed as

$$\max_{t_i \in [-\frac{M}{2}, \frac{M}{2}]^2} \sum_{x \in [\frac{M}{2}, N-\frac{M}{2}-1]^2} I_{i-1}(x) I_i(x - t_i), \quad (4.4)$$

where we consider all possible shifts $t_i$ up to $\pm \frac{M}{2}$ in each dimension, and compute the correlation only over the central $(N-M)^2$ area of the frames.

In [1], Barnea and Silverman point out that normalization is required to avoid spurious results, i.e. instead of using (4.4), we need to use something like

$$\max_{t_i \in [-\frac{M}{2}, \frac{M}{2}]^2} \frac{[\sum_{x \in [\frac{M}{2}, N-\frac{M}{2}-1]^2} I_{i-1}(x) I_i(x - t_i)]^2}{[\sum_{x \in [\frac{M}{2}, N-\frac{M}{2}-1]^2} I_{i-1}(x)^2] [\sum_{x \in [\frac{M}{2}, N-\frac{M}{2}-1]^2} I_i(x - t_i)^2]}, \quad (4.5)$$

77
For example, consider a frame pair \((I_{i-1}, I_i)\) where the true optimal displacement is \(\hat{t}_i\). If there is some \((N - M)^2\) sub-image of \(I_i\) at a displacement \(t_i \neq \hat{t}_i\) in which the intensity of every pixel is greater than the intensity of the brightest pixel in the sub-image at \(\hat{t}_i\), then the un-normalized autocorrelation result for \(t_i\) will be greater than that for \(\hat{t}_i\).

The running time of the straightforward implementation is \(O(M^2(N - M)^2)\), because each of \(M^2\) possible translations is considered, and for each translation the entire \((N - M)^2\) central area of the frames needs to be multiplied and accumulated. In [38], Stockham demonstrates that FFT techniques can be employed to reduce the running time to \(O((N - M)^2 \log M)\), and further refinements are described by Rader in [31] and Rinaldo et al in [33]. However, the relative constant factors associated with the FFT techniques are significant, about 200 ([1]). The relative constant factor for the straightforward (normalized) implementation is about 4 ([1]), so the straightforward implementation will be faster for \(M < M_B\) where \(200 \log M_B \approx 4M_B^2\), or \(M_B \approx 14\). With optimized implementations on modern machines these relative constant factors may be slightly more similar, but there will likely still be a significant break-even threshold \(M_B\).

Computing the autocorrelation as in (4.4) or (4.5) is not the only way to arrive at \(t_i\), though. In [1], Barnea and Silverman present a very general class of algorithms they call Sequential Similarity Detectors (SSDs). An SSD is a four-tuple \((O_1, O_2, D, Q)\), where \(O_1\) is an ordering algorithm that specifies in what order the translations \(t_i \in [-\frac{M}{2}, \frac{M}{2}]^2\) are tried, \(O_2\) is an ordering algorithm that specifies the order in which the pixels \(x \in [\frac{M}{2}, N - \frac{M}{2} - 1]^2\) are compared for each candidate translation, \(D\) is a metric used to compare pairs of pixels, and \(Q\) is a measuring algorithm that maps a subset of the metric values for the \((N - M)^2\) pixel pairs of a given candidate translation onto a common inspection surface \(S_i\), which allows comparison of the results among all candidate translations. The optimal translation is given by the location of a global extremum of \(S_i\).

\(^1\)Relative constant factors in [1] are given as the equivalent number of integer adds on an IBM 360/65.
Autocorrelation is an SSD where $O_1$ and $O_2$ are arbitrary total orders on their respective sets, $D$ is multiplication, $Q$ is summation, and $S_i$ is searched for a maximum. Barnea and Silverman present a different SSD where $O_1$ and $O_2$ are random and $D$ is absolute difference. They point out that, if the optimal translation is $\hat{t}_i$, then the accumulated match error (i.e. the running sum of absolute pixel differences) will grow faster for a translation $t_i \neq \hat{t}_i$ than it will for $\hat{t}_i$. As shown in Figure 4-2, faster-growing accumulated error curves can be detected because fewer pixel pairs need to be compared before the accumulated error reaches a threshold. Barnea and Silverman use the number of necessary pixel pair comparisons for the mapping $Q$ (so $S_i$ is searched for a maximum), and they experiment with both constant and monotone increasing thresholds.

![Figure 4-2: Accumulated error (i.e. running sum of absolute pixel differences) for several translations $t_{i1} \neq t_{i2} \neq \hat{t}_i$, where $\hat{t}_i$ is optimal (adapted from [1]).](image-url)

Barnea and Silverman present the results of experiments on real-world noisy data which indicate that their proposed SSD algorithm is accurate to one pixel and several orders of magnitude faster than FFT-based autocorrelation. They describe how to precompute a monotone increasing threshold curve which yields a given average
number of pixel comparisons $\tilde{n} \ll (N - M)^2$ for each candidate translation. This is the reason for their improved performance: because they stop comparing pixels when the threshold has been reached, their algorithm is early-out. Standard auto-correlation approaches must compare every pixel pair for each translation. Barnea and Silverman suggest that $\tilde{n} = 10\sqrt{(N - M)/32}$ is a "reasonable" number, so the (average case) running time of their algorithm is $O(M^2 \tilde{n}) = O(M^2 \sqrt{N - M})$, with a relatively small relative constant factor, about 2 [1].

There are still more ways to arrive at the translation $t_i$. For example, we can search for the translation which gives the highest mutual information between the two frames [43, 16]. A statistically-based algorithm for computing autocorrelations is presented by Schils in [35]. Another class of methods is based on the joint transform correlation described by Rau in [32], which is a purely optical method to compute the correlation among images [20]. Unfortunately we don’t have the space to detail these methods here.

4.2.1 The Agilent HDNS2000

It’s not hard to imagine how we could implement some of the above algorithms using off-the-shelf hardware, at least for low speeds (more on high speed implementation in Section 4.5). We could couple an image sensor to a frame grabber and a processor with some memory. However, we would ideally like to have all of these components built into one compact package, with a lens on one end and position outputs in some standard electrical format at the other end. Amazingly, such a component is now commercially available in a 16-pin DIP form factor: the HDNS2000 from Agilent Technologies. Agilent does not document the algorithm used in the chip, but it must be operating according to some relative registration scheme similar to those described above.

The HDNS2000 is a monolithic solution for building surface-independent optical mice. It’s at the heart of a new breed of optical mice available from Microsoft, Apple, and others.

Agilent describes the electrical interface and application of the HDNS2000 in [41].
The essential characteristics of this chip are listed in Table 4.1.

<table>
<thead>
<tr>
<th>Process</th>
<th>CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Power</td>
<td>18 MIPS</td>
</tr>
<tr>
<td>Frame Rate</td>
<td>1.5 kHz</td>
</tr>
<tr>
<td>Allowable Slew Rate</td>
<td>0.3 m/sec</td>
</tr>
<tr>
<td>Position Output Resolution</td>
<td>64 μm</td>
</tr>
<tr>
<td>Position Output Drift</td>
<td>1% of travel</td>
</tr>
</tbody>
</table>

Table 4.1: Vital characteristics of the HDNS2000.

**Experimental Apparatus**

We informally verified the specified output resolution and drift characteristics of the HDNS2000 with an experimental setup. As shown in Figure 4-3, we used an HDNS2000 that was already built into an optical mouse (Model MUT1 from Inland\(^2\)). We removed most of the packaging on the mouse, and we added an electrical interface between the HDNS2000 and our dSPACE 1102 prototyping system.

The HDNS2000 can be configured to provide output in both PS/2 serial mouse protocol and as two quadrature-encoded positions. The latter is a standard format compatible with a wide variety of control system electronics. However, it is unused in mice, so we were able to connect to this interface without disrupting the rest of the mouse circuitry.

As a mouse chip, the HDNS2000 is primarily intended to operate in a planar motion situation. However, since we are interested in rotary-linear motion, we tested the chip in a cylindrical configuration similar in concept to Figure 4-1. We mounted a 3 inch diameter by 1.125 inch long cylinder on a shaft which allowed it to rotate and translate freely. We patterned the outer surface of the cylinder by coloring it unevenly with a black marker, and we mounted the HDNS2000 (still attached to the remainder of the mouse we bought it in) on a rigid arm, with about a 1/16 inch gap above the cylinder.

\(^2\)Inland/Ben Cole International, Oxford MI 48371.
Figure 4.23: Apparatus for an experiment to test the HDNS2000 in a cylindrical motion.
We rotated and translated the moving cylinder by hand and observed the $x$ and $y$ output counts of the HDNS2000. By comparing the accumulated counts with a hand computation of the approximate distances traveled (given the way we moved the cylinder and the cylinder’s dimensions), we informally verified that the resolution of the HDNS2000 is about 64 $\mu$m with about 1% drift.

Summary

The update (frame) rate and slew rate of the HDNS2000 are reasonably good. Unfortunately, the resolution is not as high as we would like. But the killer for using the HDNS2000 for real-time motion control is the drift, which is not acceptable.

4.3 Laser Speckle Techniques

If we look for other existing implementations of relative registration-based 2DOF metrology, we quickly arrive at laser speckle-based techniques. Laser speckle is a phenomenon which occurs when a matte (diffusely reflecting) surface is illuminated with coherent light. As shown in Figure 4-4, rays of the incident beam which reflect off different micro-facets of the surface may intersect each other somewhere in space. At such intersections, the two rays will interfere, and thus a three-dimensional pattern of “cigar-shaped” ([14]) light and dark areas will be produced in the space surrounding the illuminated spot on the surface. An image sensor placed somewhere in this region will capture a 2D slice of the speckle pattern, which appears as a granular image of light and dark blobs.

Importantly, when the surface is displaced relative to the incident beam, the speckle pattern will also shift accordingly. Thus it is possible to recover the displacement of the surface by measuring the displacement of the speckle pattern, and many systems have been developed based on this concept [7, 39, 3].

The average diameter of the blobs in a speckle image depend only on the geometry of the imaging system. Surprisingly, the exact microstructure of the matte surface has little effect [6]. It is not necessary to use a lens—speckle formed with a bare
Figure 4-4: A matte surface illuminated by a coherent beam will produce a three-dimensional speckle pattern of "cigar-shaped" ([14]) light and dark areas where different reflected rays intersect.

...
the techniques, which were originally implemented with chemically developed photographic plates and manual inspection (speckle photography), first reported by Burch and Tokarski in [4]. Adaptation to electronic analysis (Electronic Speckle Pattern Interferometry, ESPI) apparently began with the work of Robinson in [34]. In [39], Takemori shows (surprise) that larger displacements, on the order of several mm, can be measured if the reference image is updated periodically. In [45], Takemori et al state that they have implemented 1DOF planar and rotary speckle-based sensors with resolutions of 1 μm and 1 mrad, respectively, but they unfortunately do not report on the drift characteristics of their systems.

4.4 Absolute Registration

We need to solve the problem of drift if we are to successfully use vision techniques to provide feedback in machine control systems. Fortunately, in the methods of relative registration described above, we have not yet played all our cards. There are two properties of our machine control application that we can exploit to eliminate drift:

1. Our moving machine part has a well-defined range of motion: ±1.5 cm in translation and unlimited in rotation.

2. The controller for our machine can presumably make near-term predictions about the machine’s motion.

The first of these properties allows us to consider using a global image of the entire surface area that our sensor will see over all possible motions. Then instead of comparing successive frame pairs against each other to detect relative translations $t_i$, we compare each new frame against the global image to recover the absolute position of the object at each frame, $p_i$. We call this technique absolute registration.

The same basic registration algorithms described in the previous section can also be applied in this case. However it should not be necessary to register each successive frame $I_i$ against the entire global image. The second property listed above allows us to make a prediction of the inter-frame translation $\hat{t}_i$, so we only
need to consider a small target area of the global image, centered on the predicted new position \( \hat{p}_i = p_{i-1} + \hat{t}_i \), against which we need to register each frame\(^3\). If the prediction has \( C\% \) uncertainty, then we should consider candidate translations \( t_i \in \{(\hat{t}_i + (C/100\%)\|\hat{t}_i\|d) \forall d \text{ s.t. } \|d\| \leq 1\} \), i.e. all translations \( t_i \) where \( p_{i-1} + t_i \) ends within a circle of radius \((C/100\%)\|\hat{t}_i\|\) centered on \( \hat{p}_i \). Thus the search area for each registration will be roughly \( \pi((C/100\%)\|\hat{t}_i\|)^2 \), and the size of the images to be matched will be the frame size, i.e. \( N^2 \).

How can we generate the global image? One way is to scan the entire range of motion slowly, using a temporary high-precision auxiliary metrology system to align the data. We can also consider incrementally updating the global image over time, which could possibly compensate for slow changes in alignment.

How large will the global image be? We should make sure that it can be stored in a reasonable amount of memory. If the magnification of the imaging system is \( A \), each pixel has area \( \epsilon^2 \), and the range of motion is \( L^2 \), then the size of the global image will be \( G^2 \) pixels, where

\[
G^2 = \frac{AL^2}{\epsilon^2}
\]

For example, if we have 100MB of memory, each pixel is stored as 1 byte, no compression is used, the magnification is 10, and \( \epsilon = 10\mu\text{m} \), then the largest acceptable range of motion is given by

\[
L_{\text{max}} = \frac{\epsilon G}{\sqrt{A}} = \frac{(10 \mu\text{m})\sqrt{100(220)}}{\sqrt{10}} \approx 3.2 \text{ cm}
\]

This is in the ballpark of the range of motion required for our machine. For larger machines we can consider compressing the global image, for example if we compress by a factor of 10 then \( L_{\text{max}} = 10.24 \text{ cm} \). We can also consider re-sampling the global image at a coarser resolution than \( \epsilon \) to reduce storage requirements. Then we can use a sub-pixel registration algorithm for matching, for example that described by Frischholz et al in [15]. We may be able to resample as coarsely as \( 10\epsilon \) ([15]). Com-

\(^3\)System initialization (i.e. determination of \( p_0 \)) can be achieved through limit switches or other sensors.
bining this with 10:1 compression, we might achieve \( L_{\text{max}} \approx 1 \text{ m} \), which is reasonable for medium-sized machines. On the other hand, we can apply these techniques to our small machine and require less than 100MB memory: 10MB for 10:1 compression, 100kB for 10:1 compression combined with 10\( \epsilon \) resampling.

4.5 Practical Considerations

Memory requirements are not the only practical issue that we need to think about. A prime concern is getting a camera which can support the 1 to 10 kHz framerates we require. Several such high-speed cameras do exist, but as Turko et al describe in [42], interfacing to them is not trivial. Because we are only interested in registration, we should be willing to sacrifice resolution for increased speed. We cite the HDNS2000, described in Section 4.2.1 above, as an illustration that high speed (presumably low-resolution, though Agilent does not report on this) imagers with integrated DSP can be built as commodity items.

In addition to the issues already mentioned, we should also consider things like the practicality of the lens system we require and the computational power of the processor that we will use. We present the following rough analysis of all the issues combined. We have not attempted any experiments, so this analysis has not been verified.

4.6 A Design Feasibility Analysis

Let’s assume that we have a low-resolution imager with \((N = 50)^2\) square pixels, each of area \((\epsilon = 20 \mu\text{m})^2\), so the total sensor size is 1 mm\(^2\). Let’s also assume that our lens system is a microscope with a 2x objective (100 mm focal length, clear aperture = 12 mm) and a 10x eyepiece (focal length = 25 mm), both available from [27], with a separation of 160 mm (which is typical [2]). Then the total magnification will be 20x, the combined focal length will be \(|1/100 + 1/25 - 160/2500|^{-1}\text{[mm]} \approx 70 \text{ mm}\), and the effective \(f/\) number will be 70/12 \approx 6. If we use speckle effects produced by a
HeNe laser (\( \lambda \approx 630 \text{ nm} \)) the average “blotch” size will be 
\[1.22(1 + 20)(630 \text{ nm})(6) \approx 100 \mu m = 5 \epsilon\]
according to the formula given in Section 4.3. Thus the imager should have sufficient resolution to resolve most blotches, and several blotches should always be in view.

### 4.6.1 Resolution

Let’s assume that the registration algorithm we use gives results correct to 1 pixel. Then our position-sensing resolution will be 
\[\epsilon/A = (20 \mu m)/20 = 1 \mu m,\]
which is exactly what we want.

### 4.6.2 Range of Motion and Memory Requirements

As we described above, the amount of memory required is determined by range of motion of the machine. Let’s consider to consider a small machine where the range of motion is \( \pm 1.5 \text{ cm} \) in translation and unlimited in rotation. For convenience, we’ll assume that the circumferential surface area that the camera views would be square if it were “unwrapped” from the moving cylinder. Let the side-length of this square be \( L = 3 \text{ cm} \), which accounts for the total travel in translation. If we assume that the radius of the moving surface is 
\[R = (3 \text{ cm})/(2\pi) \approx 4.8 \text{ mm},\]
then we will also be able to accommodate the total travel in rotation. For the situation we are describing here, assuming 1 byte/pixel, no compression, and no resampling, the required memory is 
\[A(L/\epsilon)^2 = 20(0.03/0.0002)^2 = 45 \times 10^6 \approx 43\text{MB}.\]

### 4.6.3 Update Rate, Slew Rates, and Processor Speed Requirements

Both the update (i.e. frame) rate and the allowable slew rate will be determined indirectly by the computational power of the processor. Let’s consider using the SSD algorithm that Barnea and Silverman propose in [1]. The per-frame running time of
this algorithm will be about

\[ K_f 10 \sqrt{(\text{image dimension})/32} \times (\text{search area}) = K_f \frac{10 \sqrt{N/32}}{32} \left( \pi \left( \frac{(C/100\%)}{\|\hat{t}_i\|_{\text{max}}} \right)^2 \right) \]

based on the results in Section 4.2 and Section 4.4, where \( K_f \) is a constant factor, \( C \) is the percent uncertainty in predicting the inter-frame translation, and \( \|\hat{t}_i\|_{\text{max}} \) is the maximum allowed inter-frame translation. Let’s guess \( K_f = 4 \) and \( C = 25\% \), so that the per-frame running time reduces to \( (4)(10) \sqrt{50/32} \times (\text{search area}) = 50 \times (\text{search area}) = 50\pi((25\%/100\%))\|\hat{t}_i\|_{\text{max}})^2 \approx 10\|\hat{t}_i\|_{\text{max}}^2 \). If \( P \) [FLOPS] is the processor power, \( U \) [Hz] is the desired update rate, and \( V_{\text{max}} \) [m/s] is the maximum slew rate in translation, then

\[
10\|\hat{t}_i\|_{\text{max}}^2 = (\text{FLOPS per frame}) = \frac{P}{U} \\\ne\|\hat{t}_i\|_{\text{max}} = \frac{V_{\text{max}}}{U} \\
\Rightarrow V_{\text{max}} \approx 0.3e\sqrt{PU}
\]

So for \( P = 10 \) MFLOPS and \( U = 10 \) kHz the maximum allowable slew rate in translation will be about 2 m/s. Since the radius of the moving surface is \( R \approx 4.8\text{mm} \), its circumference is 3 cm, and the corresponding maximum slew rate in rotation will thus be \( (2 \text{ m/s} \times (2\pi \text{ rad/3 cm}) \approx 419 \text{ rad/s}) \).

### 4.7 Summary

In this chapter we considered sensors for cylindrical motion based on machine vision algorithms. First we described existing systems for general 2DOF metrology based on relative registration techniques like autocorrelation. After reviewing the algorithmic foundations of these techniques, we described the HDNS2000, an integrated circuit that incorporates a miniature camera to perform relative registration-based 2DOF metrology. We described an experiment which we performed to evaluate the performance of the HDNS2000 in a cylindrical metrology situation. The results of this experiment confirmed that, while the chip basically functions as specified, the measurements provided by the HDNS2000 have too much drift and too low a resolution
to be useful in our application.

Next we described the phenomenon of laser speckle and reviewed its use in metrology. Laser speckle is a convenient way to produce an optically patterned image from a surface that is not specular (i.e. mirror like) but that need not be optically differentiated otherwise. All of the speckle metrology systems we reviewed also seemed to rely on relative registration algorithms.

Relative registration-based systems inherently suffer from measurement drift. In the final part of this chapter we suggested a refinement called *absolute registration* that might be used in eliminate drift in machine-control applications. We presented an analysis which suggests that an implementation based on absolute registration, and also incorporating laser speckle, may be feasible.

We have not performed any experiments to verify our analysis. It’s clear that several questions need to be answered before we can endorse an absolute registration-based machine vision approach to rotary-linear metrology. A major issue is whether the laser speckle phenomena will prove reliable and repeatable over the relatively long travels we require. Another major issue is speed: How can we incorporate a very high-speed camera and its associated processing electronics into our sensor in a cost-effective way?

Until these questions are answered, it seems more promising to consider an interferometric technique like we describe in Chapter 2 or 3, or one of the other techniques we describe in Section 1.5 or 5.2.
Chapter 5

Conclusion

In this thesis we have explored several novel techniques for measuring cylindrical motion. We have focused mainly on interferometric techniques: the tilted-mirror interferometer (Chapter 2) and the helicoid-mirror interferometer (Chapter 3). We also presented a vision-based technique we call absolute registration (Chapter 4).

We have developed a physical implementation of only one of these techniques: the tilted-mirror interferometer, which we integrated into a new 5DOF CNC grinding machine that we are beginning to construct. With 63 μrad resolution in rotation and 3 nm resolution in translation, our implementation has very good performance for a first prototype and enables the testing and refinement of our axis concept.

As we described in Section 2.8, we have begun to implement a control system for the prototype z-θ DOF of our new 5DOF grinding machine that uses our tilted-mirror interferometer to provide metrology data. This work is only a start—we still have a long way to go before we can begin to cut parts with the new machine.

5.1 Results

The resolution that our tilted-mirror interferometer achieves in θ is about an order of magnitude poorer than our target of 5 μrad. As we described in Section 2.5, it should be possible to improve the practical resolution of our implementation by reducing noise. This will only be feasible to a certain extent, though. Even if we
could reduce all noise, the theoretical resolution that we could achieve, 10 μrad, does not quite reach our goal. Thus, it seems that if we are to improve the tilted-mirror interferometer so that it reaches the resolution goals we set we will have to consider implementing an enhancement to our setup, for example the beam centering optics described in Section A.3.2.

In addition to building a sensor for our 5DOF machine, we also have the larger goal of developing methods that can be adapted to other situations where the measurement of rotary-linear motion is required. Is the tilted-mirror interferometer that we implemented the best solution in all such situations? Probably not. In what situations is it likely to be a good choice? The tilted-mirror interferometer uses only standard, off-the-shelf PMI hardware, which some labs may have on-hand, plus several mounts which are relatively simple to fabricate using common machine tools. These properties make the tilted-mirror interferometer an attractive option to those who already have some PMI hardware and who require a sensor for cylindrical motion that is otherwise quick and cheap to implement. We fit into this category, which is why we chose this solution.

What about the helicoid-mirror interferometer? Since we have only developed this concept in theory, we are not in a position to draw too many conclusions about it. As we described in Section 3.4, there are several major issues yet to be resolved before we can consider implementing the helicoid-mirror interferometer. Especially troubling are optical aberrations that may be induced by the local curvature of the helicoid. Additionally, the fabrication of the helicoid mirror is potentially difficult.

The helicoid-mirror interferometer does seem to offer a larger singularity-free range of motion, using only two measurement beams, than the two-beam implementation of the tilted-mirror interferometer would. Also, as we illustrated in Figure 3-3, we have identified a possible implementation of the helicoid-mirror interferometer that has the potential for significantly lower moving mass/inertia than the tilted-mirror interferometer. So, for those who have only enough PMI optics for two measurement beams on-hand, or for those who require a lower mass/inertia than the tilted-mirror interferometer provides, the helicoid-mirror interferometer may be a good option,
provided that the issues described in Section 3.4 can be resolved.

PMI hardware costs thousands of dollars, so, for those who do not already have this equipment available, both the tilted-mirror interferometer and the helicoid-mirror interferometer will be expensive prospects.

Much work also remains to be done before we can consider the machine vision technique of absolute registration that we propose in Chapter 4 to be a feasible solution for measuring cylindrical motion. As we mentioned in Section 4.5, one major question is how to inexpensively incorporate a high-speed camera and its associated electronics into the sensor. However, a higher-level issue is that the concept of absolute registration seems to be somewhat novel, and we have not verified its performance experimentally. So we can only recommend this technique to those who are prepared for significant development effort and uncertainty.

At this time, some form of cylindrical encoder as in Chitayat [5] is the technique which is most likely to robustly achieve our desired performance specifications. We have focused our research efforts elsewhere though, since the cylindrical encoder issues are primarily developmental.

5.2 Additional Techniques

There are other techniques for measuring cylindrical motion in addition to the interferometric and vision-based concepts we have proposed. For completeness, here are a few of them:

**An Optical Encoder Technique** Standard linear and rotary optical encoders are 1-dimensional devices. However, there do exist some 2-dimensional planar encoders, for example the Renishaw RGH42 XY Planar Encoder [29]. It should be possible to adapt a technique like this to our cylindrical geometry, effectively wrapping the planar encoder onto the surface of the moving cylinder. As we discussed in Chapter 1.5, Chitayat et al at Anorad Corporation have implemented this sort of a technique [5].
A Capacitance Gauge Technique Capacitance gauges can be used to precisely measure small distances, up to a range of about ±1000 μm [40]. It may be possible to use such gauges as the basis for a translation-tolerant rotation sensor (the translation itself would have to be sensed in some other fashion). As illustrated in Figure 5-1, we could modify the surface of the moving part so that it is not perfectly cylindrical, but rather so that it undulates as a function of θ in some controlled way (though it should still have zero curvature in the z direction so the system is unaffected by translational motions). As long as these undulations are lower in magnitude than the range of the capacitance gauges, we should be able to mount one or more of the gauges on the fixed part of the machine so that they measure the radial gap (i.e. the difference from cylindrical) caused by the undulations. From this information we should be able to recover the rotation of the moving machine part.

Figure 5-1: A concept for measuring the rotary part of rotary-linear motion with capacitance gauges. The gauges are attached to the fixed part of the machine, and measure a gap to the moving part. The moving part is specifically non-cylindrical, so the gap spacings will change as the part rotates. The part has zero curvature in the direction of translational motion, so the gap widths will ideally not vary with translation.
A Rotating Half-Waveplate Technique ¹ A half-waveplate is an optical device commonly used to rotate the polarization angle of a plane-polarized beam. If, on the other hand, circularly polarized light is directed through the half-waveplate, then circularly polarized light will also emerge ([17]), but with a phase-shift that depends on the amount of rotation of the waveplate. A rotation of $\pi/4$ results in a phase shift of $\pi/2$ radians according to reference [44], which is a website that includes some information on this concept. It should be possible to detect this phase shift with an interferometer, and thus to recover the rotation of the half-waveplate. Furthermore, the phase shift should be independent of where along the beam the waveplate is located, so it would be insensitive to translation of the waveplate. As shown in Figure 5-2, by attaching the waveplate to the moving machine part, this phenomena might be used as the basis of a sensor for the rotary part of a cylindrical motion. However, due to time constraints, we have not investigated this idea further.

Figure 5-2: A concept for measuring the rotary part of rotary-linear motion with a half-waveplate and an interferometer. As the half-waveplate rotates it causes a phase shift in the returned measurement beam that depends on the amount of rotation.

Electromagnetic Methods Electromagnetic resolvers and Linear Variable Differential Transformers (LVDT's) are common sensors for independent rotary and

¹This idea was proposed to us by John Ziegert, Professor of Mechanical Engineering at the University of Florida.
linear motion, respectively. It may be possible to design a resolver that is insensitive to translation of its rotor, and similarly to design an LVDT that is insensitive to the rotation of its armature. Perhaps combined into one unit, these could form the basis of a sensor for cylindrical motion. In [12] de Wit et al use an electromagnetic sensor to measure translation but not rotation, as we mentioned in Chapter 1.5.

**Polarimetric Methods** It is possible to manipulate the polarization plane of a beam of light according to the rotation of the moving machine part, as in de Wit et al. However, this technique may be prone to sensitivity to absolute intensity problems as well as to the homogeneity of the polarizer material.

### 5.3 Future Work

Besides exploring these additional techniques, there are many possible improvements and extensions to the specific techniques described in Chapters 2, 3, and 4:

**improve the practical resolution of the tilted-mirror interferometer** We see two possibilities here: reducing the interferometer noise, and increasing the mirror tilt. It should be possible to reduce the interferometer noise by more carefully isolating the machine from vibration, temperature fluctuations, and air currents. As we describe in Appendix A, the tilt in the mirror only displaces the returned measurement beam at the detector, and does not deviate it in angle. It may be possible to install passive beam-centering optics (Section A.3.2) which would correct this shift to some extent, thus allowing a larger mirror tilt.

**reduce the moving mass/inertia of the tilted-mirror interferometer** As we stated in Section 2.8, a major portion of the moving mass and inertia of the tilted-mirror interferometer is accounted for by the mirror itself. There are several ways we could lighten the mirror; for example we could mill pockets out of its glass substrate, or we could replace the substrate entirely with some type of light, stiff material (e.g. a honeycomb-type material).
make the tilted-mirror interferometer accurate We discussed several issues related to accuracy in Section 2.6. However, much remains to be done in order to quantify and enhance the accuracy of our implementation of the tilted-mirror interferometer.

implement the helicoid-mirror interferometer We have not yet performed any hardware experiments related to the helicoid-mirror interferometer concept proposed in Chapter 3. The next major steps for the development of this concept are enumerated in Section 3.4. Of primary concern are fabrication of the helicoid mirror and analysis and resolution of the optical aberrations that may be present in the system.

implement a vision system with absolute registration We have neither had time to perform many experiments related to the machine-vision concept proposed in Chapter 4. In preparation for some initial experiments we recently purchased a simple CCD camera, a video-capture board, a microscope lens assembly, and a HeNe laser. The first experiment performed with this hardware should be designed to verify the spatial repeatability of the speckle pattern that we require for the absolute registration algorithm. A simple experimental procedure might involve setting up a precision axis (e.g. an air-bearing spindle) with the laser and the camera so that it produces a speckle image. Will we reliably observe the same speckle pattern if we move the axis and then return it to the original position?

5.4 Final Thoughts

The development of the ideas presented in this thesis has been somewhat frustrating. We came up with the concept of the tilted-mirror interferometer very quickly. However, we also immediately saw the design trade-offs that it would entail, especially that increasing its resolution seems to require also increasing its moving inertia (Section 2.5). We were quite excited about several ideas that we had for getting around
this trade-off, but as we describe in Section A.3.1, most of these ideas are fatally flawed. Furthermore, as we fleshed out the design of the tilted-mirror interferometer, we discovered the singular configurations and the related measurement ambiguity (Section 2.3.1) that are unavoidable using only two laser beams.

We expended significant effort searching for additional concepts that would be simpler to implement, cheaper, or better in performance than the tilted mirror interferometer and the techniques that other groups have implemented (described in Section 1.5). We came up with the helicoid-mirror interferometer, which does seem to have a small singularity issue than the tilted-mirror interferometer. However, we were not able to solve the problem of actually getting the helicoid mirror fabricated.

We discovered the machine vision techniques related to relative registration, and developed the idea of absolute registration, while taking a course in machine vision (6.866 at MIT, taught by Berthold Horn). Much of the content of Chapter 4 is derived from a term paper that we developed for that course. Initially we hoped that this approach would be lower in cost and simpler to fabricate than other techniques, but we quickly realized that this idea is not without its own problems, especially the cost of the high-speed camera system that seems to be required.

Since our latter two ideas seemed to have more serious problems than our initial concept of the tilted-mirror interferometer, we decided to implement this method rather than attempt one of the others. As we have stated, the resulting sensor achieves reasonable performance, but it does not meet all of our goals.

We have recently purchased some hardware to begin experimenting with the vision-based technique proposed in Chapter 4, but we have run out of time to do this, so this work remains as an exciting project for future students!
Appendix A

The Plane Mirror Interferometer (PMI)

There are a number of different optical configurations in common use for interferometrically measuring displacement [26]. The tilted-mirror interferometer and helicoid-mirror interferometer cylindrical metrology techniques that we describe in Chapters 2 and 3, respectively, are each based on a specific configuration of optics for interferometric displacement measurement called the two-pass Plane Mirror Interferometer (PMI). In [26], Musinski gives a good overview of modern displacement-measuring interferometer technologies including the PMI. Reference [25] from Hewlett Packard is the official reference manual for the specific PMI hardware that we use.

In this Appendix we describe the PMI in detail, and we elaborate on several potential enhancements to the standard PMI optics that might be useful in the tilted-mirror interferometer and helicoid-mirror interferometer.

A.1 Operation of the PMI

One configuration of a PMI, shown in Figure A-1, is a rigid assembly of four optical components: a polarizing beam-splitter, a quarter-waveplate, and two retroreflectors.

As its name states, the PMI optics are designed to measure the displacement of a
Figure A-1: The two-pass Plane Mirror Interferometer (PMI). For clarity the measurement and reference beams are drawn adjacent to each other, but in reality they are coincident. We can consider the operation of the two passes of the measurement beam to be the same as that of a single virtual beam centered between them. We use HP 10706A PMI optics, HP 10780A receivers, and HP 10897A axis interface electronics.

planar mirror, which we will call the measurement mirror. In the ideal arrangement, the measurement mirror always remains perpendicular to the measurement beams, and only the displacement of the mirror in the direction parallel to the beams is measured. Components of the displacement that are tangent to the beams are ignored. However, as we describe in Section A.2, the PMI optics allow small (mrad-scale) deviations in angle of the measurement mirror from the perpendicular. We make critical use of this property in both the tilted-mirror interferometer and the helicoid-mirror interferometer measurement techniques.

An important feature of the PMI system that we use is that it employs heterodyne detection [26, 25]. The measurement information it provides is derived from the frequency of the recombined measurement and reference beam, rather than directly from the intensity of the recombined beam. The system is designed so that frequency of the recombined beam will vary as the Doppler effect causes the phase of the returned measurement beam to change with changes in the position of the measurement mirror.

The use of heterodyne detection is important because, compared with an intensity-based system, it offers greatly reduced noise levels [26]. Changes in ambient light,
fluctuations in laser power, variations in detector sensitivity, and movement of the recombined beam across the aperture of the detector would all contribute significant noise to an intensity-based system. However, these factors have a far smaller effect the frequency-based system that we use.

There are several sub-types of PMI optics. One variable is the number of passes the measurement beam makes between the optics and the measurement mirror. We use two-pass optics, where the measurement beam travels twice to the mirror and back. However, we rarely need to explicitly consider both beams independently because their combined operation is indistinguishable from that of a single virtual beam. As shown in Figure A-1, this virtual beam would be exactly centered between the two actual beams. When we speak of the measurement beam (singular) we are referring to this virtual beam.

The number of passes, the wavelength of the laser light, and the interpolation capacity of the interface electronics combine to give the theoretical resolution of the PMI system. However, there will be some non-repeatable noise in our measurement data unless we work hard to isolate the PMI setup from noise sources like vibration, temperature variations, and air currents. Random deviations in the pointing angle of the laser beam will also couple into the distance measurements that the PMI report, and such deviations can originate from within the source laser cavity.

We perform our experiments on an optical table with passive pneumatic isolator legs, but we have not taken any additional steps to minimize noise sources. Figure A-2 shows a plot of the PMI response for a motionless target mirror, sampled at 5 kHz over 1 second with a deadpath (distance between PMI and measurement mirror) of about 3 cm. We consider the roughly 3 nm peak-to-peak high-frequency noise to be non-repeatable and essentially random. We account for this noise in our analyses by considering it to determine the practical resolution of the interferometer optics.

Table A.1 lists the theoretical and practical resolutions and other vital characteristics of the HP 10706A PMI optics, HP 10780A receivers, HP 10897A axis interface electronics, and HP5517B laser source that we use.

The PMI, as with all interferometric displacement measuring optics, returns only
Figure A-2: The PMI response for a motionless target mirror, sampled at 5 kHz over 1 second with a deadpath of (distance between PMI and measurement mirror) of about 3 cm. We consider the roughly 3 nm peak-to-peak high-frequency noise to be non-repeatable and essentially random.

The relative position of the measurement mirror. This means that there is some arbitrary constant $C$ such that the response of the measurement beam, $d$, is

$$d(z) = C + z,$$  \hspace{1cm} (A.1)

where $z$ is the position of the measurement mirror in the direction parallel to the measurement beam (the positive direction of $z$ is oriented from the PMI optics towards the measurement mirror). Practically, $C$ depends on the state of the interface electronics at the time it is commanded to zero itself.
Theoretical Resolution 0.625 nm
Noise Amplitude\(^a\) 3 nm p-p
Practical Resolution 3 nm
Range ±10.6 m
Maximum Slew Rate 254 mm/sec
Wavelength 633 nm (HeNe)
Interpolation 1024
Maximum Update Rate\(^b\) 100kHz
Beam Diameter 6 mm
Two-Pass PMI Beam Separation 12.7 mm

\(^a\)The noise amplitude (which determines the practical resolution) is not an intrinsic characteristic of our interferometer optics or electronics, but rather arises due to the inadequacy of the isolation that we have implemented.

\(^b\)Maximum update rate is specified for connection to the 10897A boards via VMEbus, which we use. Faster rates are possible with other interface techniques.

Table A.1: Vital characteristics of the HP 10706A PMI optics, HP 10780A receivers, and HP 10897A axis interface electronics, and HP5517B laser source that we use. Note that these are the characteristics of the raw optics and electronics—the properties of the cylindrical sensors we describe in subsequent chapters will depend on these but will not necessarily be identical.

### A.2 Angular Tolerance

As we mentioned above, the two-pass PMI optics are designed to operate with the measurement mirror nominally perpendicular to the measurement beam. Importantly, some amount of angular deviation α of the measurement mirror is allowed. As shown on the right in Figure A-3, the effect of any tilt in the measurement mirror does not deviate the returned measurement beam in angle, but only displaces it parallel to the reference beam. Thus the wavefronts of the returned measurement and reference beams will remain parallel even when the measurement mirror is tilted, so no intra-beam fringes will be created at the detector, and the response of the PMI will remain an accurate measurement of the displacement of the measurement mirror.

Though Figure A-3 only shows a 2D arrangement, where the tilt of the measurement mirror coincides with the plane containing the measurement beams, the angular tolerance of the two-beam PMI is the same for any 3D tilt of the measurement mirror [25].
Figure A-3: On the left is the normal PMI arrangement, where the measurement mirror is perpendicular to the measurement beam. For clarity the measurement and reference beams are shown adjacent to each other, but in reality they start out coincident. On the right the measurement mirror has been tilted to the angle $\alpha$ w.r.t. the perpendicular plane to the measurement beam. Notice that in this case the measurement beam is displaced horizontally when it returns to the receiver, but it remains parallel to the reference beam. This figure shows a 2D arrangement where the tilt of the measurement mirror coincides with the plane containing the measurement beams, but the angular tolerance of the two-beam PMI is the same for any 3D tilt of the measurement mirror [25].

The amount we tilt the measurement mirror and the distance of the measurement mirror from the PMI optics determine the amount of displacement in the returned measurement beam. This sets a limit on the maximum allowed tilt for a given working distance, because some part of the measurement beam must coincide with the reference beam at the detector to allow measurement of their interference.

The maximum recommended tilt is an inverse function of the distance from the interferometer optics to the mirror. This function is not precisely specified by HP for the 10706A PMI optics we use, but only given as a set of three sample values. We plot these in Figure A-4 along with a cubic interpolating curve, which we will use later to extrapolate other values.
Figure A-4: The maximum recommended tilt of the measurement mirror for the HP 10706A PMI optics is an inverse function of the distance from the optics to the mirror. HP provides three data points on this curve.

A.3 Tilting the Measurement Mirror in Order to Measure Perpendicular Displacements

For the tilted-mirror interferometer and helicoid-mirror interferometer, which we describe in Chapters 2 and 3, respectively, we intentionally tilt the measurement mirror because by doing so we can coerce the PMI response to depend not only on the displacement of the measurement mirror parallel to the measurement beam, but also on its displacement in directions perpendicular to the beam. As shown in Figure A-5, the PMI response to such a perpendicular displacement would be maximal when the direction of the displacement is aligned with the direction of maximum slope of the tilted measurement mirror.
Figure A-5: If we tilt the measurement mirror then the PMI response will depend on both the parallel displacement $z$ and the perpendicular displacement $y$ of the measurement mirror.

In this case the response of the measurement beam will be

$$d(z, y) = C + z - y \tan \alpha,$$

(A.2)

where $\alpha$ is the tilt of the measurement mirror, $z$ is the position of the measurement mirror in the direction parallel to the measurement beam, and $y$ is the position of the mirror in the plane perpendicular to the measurement beam in the direction of the mirror’s maximal slope. The response will vary trigonometrically for perpendicular displacements in other directions—this occurs in the tilted-mirror interferometer (details in Chapter 2) but not in the helicoid-mirror interferometer.

When we tilt the measurement mirror to allow measurement of perpendicular displacements we will in general desire the largest possible mirror tilt $\alpha$. This is because the more we tilt the mirror, the more sensitive the PMI will be to perpendicular displacements. For this reason, we have explored several ways to extend the maximum tilt angle beyond that which is allowed by the raw PMI optics. In the next few sections we describe three such ideas. We show that the first two of these, which attempt to change the mirror reflection law, will not work. In retrospect, this seems obvious, but these ideas deceptively appear correct and are commonly suggested. The third idea we describe, which adds optics to passively re-center the returned measurement
beam, does seem to have some potential, but we have not attempted to implement it.

### A.3.1 Attempting to Change the Mirror Reflection Law

In this section we describe two ideas we had for increasing the allowable tilt of the measurement mirror. We show that, while both of these ideas do allow the tilt of the mirror to be increased, they unfortunately each have additional effects which destroy their potential usefulness to us. This was somewhat frustrating to discover, because each of these ideas initially excited us by the prospect of offering a higher mirror tilt, and hence a higher resolution for measuring perpendicular displacements.

The first idea we describe entails placing a prism in front of the measurement mirror, and the second explores replacing the mirror with a reflective diffraction grating. While we were able to detect the unfortunate side effects in each of these approaches on paper, we felt that it would be instructive to carry out an experiment to verify our prediction for the grating method. We describe this experiment, which did verify our calculations, in the final part of this section.

Each of these ideas is based on the concept of correcting for the change in angle that the tilted mirror imparts to the reflected measurement beam. This change in angle is what gives rise to the displacement of the measurement beam when it returns to the detector. Again, if this displacement becomes too large then the detector fails to function. The change in angle arises simply because the interaction of the beam with the measurement mirror follows the well known law

\[
\text{angle of incidence} = -\text{angle of reflection}.
\]

That is, the angle between the reflected beam and the mirror surface normal will have the same magnitude as the angle between the incident beam and the surface normal, but will be opposite in direction (Figure A-6, left). What if we could somehow coerce the measurement mirror to violate this law? Specifically, if we could change the
mirror's behavior so that

angle of incidence = angle of reflection,

then the angle between the reflected beam and the mirror surface normal would be equal in magnitude and direction with the angle between the incident beam and the surface normal. Thus, the reflected beam would be coincident with the incoming beam, and the returned measurement beam would not be displaced relative to the reference beam at the detector. The operation of such a “mirror” is shown on the right in Figure A-6.

![Mirror Diagram](image)

Figure A-6: On the left, a normal mirror is shown exhibiting the normal mirror reflection law. On the right a “modified” mirror is shown which reflects the return beam in the same direction as the incident beam, even though the mirror is not perpendicular to the incident beam. We discuss several optical arrangements below which seem to achieve this optical geometry, but none of them are suitable for our purposes.

Changing the mirror’s geometric reflection property may seem like a futile thing to attempt. However, as we describe in the next two sections, there are at least a few optical systems which seem to achieve this. That is, they successfully modify the mirror’s operation so that the returned beam is coincident with the incoming beam. Unfortunately, each of these ideas also has an additional effect which nullifies the measurement behavior that we desire of the tilted mirror, namely a response which varies with perpendicular displacements.
Correction Prism

As shown in Figure A-7, we can use a prism to turn the measurement beam before it strikes the mirror. If $\beta$ is the angle of the prism and $n$ the index of refraction, then we can select $\beta$ so that the prism always turns the beam so it is exactly perpendicular to the mirror by applying Snell’s law:

\[
\begin{align*}
    n \sin \beta &= \sin(\alpha + \beta) \\
    \beta &= \arctan \left( \frac{\sin \alpha}{n - \cos \alpha} \right).
\end{align*}
\]  \hspace{1cm} (A.3)

![Figure A-7: We can arrange a prism in front of the tilted mirror so that it turns the incoming beam to always be normal to the mirror surface. Here we assume that the prism and mirror are rigidly attached to each other and move as a single unit. Unfortunately, as we show in the text, the response of the measurement beam (shown in red) will not depend on $y$.](image)

If we attach a prism to the mirror to form a rigid assembly, then the incoming measurement beam will always coincide with the exiting beam, as shown in the Figure. This achieves our geometric goal of changing the mirror’s behavior so that the reflected beam coincides with the incident beam, even though neither is actually perpendicular to the mirror.
Unfortunately, there are additional effects which simultaneously cancel all dependence of the PMI response on the perpendicular displacement $y$. Let $\text{glass}(y)$ be the distance the measurement beam travels through the prism glass, and let $\text{air}(y)$ be the distance the beam travels through the air between the prism and the mirror. Then, to within a constant factor,

$$\text{glass}(y) = c + y \tan \beta,$$

where $c$ is the constant part of the path through the prism (as shown in the Figure), and

$$\text{air}(y) = q \sin(\alpha + \beta),$$

also to within a constant factor. From the Figure,

$$q = \frac{D - y}{\cos \beta}$$

and from (A.3)

$$\cos \beta = \frac{\sin(\alpha + \beta)}{n \tan \beta}$$

so

$$q = \frac{D - y}{\cos \beta} = \frac{D - y}{\sin(\alpha + \beta)} \frac{n \tan \beta}{n \tan \beta}$$

$$\text{air}(y) = q \sin(\alpha + \beta)$$

$$\text{air}(y) = (D - y) n \tan \beta.$$  

The PMI response, $d$, will be (to within a constant factor)

$$d(z, y) = C + z + n\text{glass}(y) + \text{air}(y)$$

$$= C + z + ny \tan \beta + (D - y) n \tan \beta$$

$$d(z, y) = C + z + D n \tan \beta.$$

where $C$ is a constant factor that depends on where the interface electronics was
zeroed, as described above. Since (A.7) does not depend on \( y \), the prism-and-tilted-mirror system will not tell us anything about the perpendicular displacement of the mirror.

**Littrow Mount Diffraction Grating**

Diffraction gratings are another type of optical device which can be used to turn light. As shown on the left in Figure A-8, a reflective diffraction grating will produce multiple orders of diffracted beams for an incoming beam. If the incoming beam makes the angle \( \alpha \) with the grating surface normal, then the angle \( \beta \) of diffracted beam \( m \) is given by the grating equation:

\[
\sin \alpha + \sin \beta = m \lambda
\]

or

\[
\beta = \arcsin((m/d)\lambda - \sin \alpha),
\]

where \( d \) is the spacing between the rulings of the grating and \( \lambda \) is the wavelength of the light. Interestingly, we can add the constraint

\[
\alpha = \beta
\]

to (A.8) to get

\[
2d \sin \alpha_t = m \lambda
\]

or

\[
\alpha_t = \arcsin \left( \frac{m \lambda}{2d} \right).
\]

Thus, if we arrange the angle of the incoming beam to be \( \alpha_t \), the order-\( m \) refracted beam will coincide with it. The angle \( \alpha_t \) is called the *Littrow angle* [18]. A diffraction grating oriented at the Littrow angle for the first-order diffracted beam \( (m = 1) \) is shown on the right in Figure A-8.

Can we simply use a diffraction grating mounted at the Littrow angle instead of the tilted mirror? Indeed, this arrangement solves the geometric problem of getting the return beam to coincide with the incoming beam. Furthermore, the Littrow
Figure A-8: On the left is a diffraction grating with an incident beam making angle $\alpha$ with the grating surface normal. The grating surface is perpendicular to the paper and is ruled in the direction normal to the paper. On the right the grating has been oriented at the first-order Littrow angle $\alpha_l$, so the incoming beam is coincident with one of the first-order diffracted beams. In the magnified view we show the individual rulings of the grating, which are blazed (i.e. cut asymmetrically) so that when the grating is mounted at the Littrow angle the large faces of the rulings are perpendicular to the incoming beam. This has the effect of directing most of the incoming power back along the first-order diffracted beam.

angle for $m = 1$, $\lambda = 633$ nm, and a reasonable choice of $d$ is much larger than the maximum tilt the PMI allows a regular mirror. For example, for $d = (1/6) \times 10^{-5}$ m (600 lines/mm), $\alpha_l \approx 0.1911$ rad $\approx 10.95^\circ$.

Unfortunately, there is an additional effect, called the grating interferometer effect, which will again nullify all dependence of the PMI response on the perpendicular translation of such a tilted grating [18, 36]. This effect states that when an incident beam is displaced by $\delta y$ across the rulings of a grating there is a phase shift $\phi$ induced in the order-$m$ diffracted beam given by

$$\phi(\delta y) = 2\pi m \frac{\delta y}{d}. \quad (A.10)$$

To the PMI receiver, this phase shift is indistinguishable from a phase shift due to a
change in path length of the measurement beam, so it will be added in to the PMI response \( d \).

Because the grating is mounted at the Littrow angle \( \alpha_l \), a perpendicular translation of \( y \) in the direction of the maximum slope of the grating will actually cause the measurement beam to scan across the grating surface by the amount

\[
\delta_y(y) = \frac{y}{\cos \alpha_l}.
\]

Thus the grating interferometer effect, as a function of the perpendicular translation, will be

\[
\text{grating-interferometer}(y) = \phi(\delta_y(y)) \left( \frac{\lambda}{2\pi} \right)
\]

\[
\text{grating-interferometer}(y) = y \left( \frac{m\lambda/d}{\cos \alpha_l} \right). 
\] (A.11)

Combining this with (A.9), we get

\[
\text{grating-interferometer}(y) = 2y \tan \alpha_l .
\] (A.12)

Thus, for each pass of the measurement beam, (A.12) gives the apparent change in path distance as a function of the perpendicular displacement \( y \) in the direction of maximum slope of the grating.

Our original intention was that the tilted grating would act as a “special” tilted mirror, differing from a normal tilted mirror only in that the returned beam would be coincident with the incoming beam. The change in path length for each pass of the measurement beam against such a tilted mirror would be

\[
\text{tilted-mirror}(y) = -2y \tan \alpha_l
\] (A.13)

where the factor of 2 derives from the fact that the path length changes both going to mirror and returning from it, and the factor of \(-1\) from because we have oriented \( y \) so that the mirror surface becomes closer to the PMI optics as \( y \) increases.
Unfortunately, \((A.13)\) is equal in magnitude and opposite in sign from \((A.12)\), so the net change in optical path distance as a function of \(y\) per pass of the measurement beam will be null. Formally,

\[
\text{path-distance-per-pass}(y) = \text{tilted-mirror}(y) + \text{grating-interferometer}(y)
\]

\[
= -2y \tan \alpha_l + 2y \tan \alpha_l
\]

\[
\text{path-distance-per-pass}(y) = 0.
\]

We found this result a little surprising, so we set up an experiment to verify it. As shown in the CAD diagram in Figure A-9 and the photo in Figure A-10, we arranged a diffraction grating at the Littrow angle in front of one PMI, and simultaneously arranged a plane mirror perpendicular to the measurement beam of a second PMI. The grating and the mirror were both rigidly affixed to a \(y\) translation stage. The experiment consisted of moving \(y\) in increments of about 0.050 inch and recording the responses of the two PMI for each increment. If our analysis is correct, then we should see no significant difference between the slopes of the two responses, because the grating interferometer effect should exactly cancel the tilted-mirror effect of the grating.

We used a 24 mm by 52 mm duplicated rectangular reflection grating, number 23-04-BK-267, from Richardson Grating Laboratory\(^1\). The line spacing on this grating is \(d = (1/6) \times 10^{-5} \text{m}\), or 600 lines/mm, which gives a Littrow angle of \(\alpha_l \approx 0.1911 \text{rad} \approx 10.95^\circ\) for the \(\lambda = 633\) nm laser that we use. As illustrated in Figure A-8, this grating is blazed\(^2\) at the angle \(11^\circ 21'\), which roughly maximizes the power of the first-order diffracted beam (transmittance under our operating conditions is specified at 70–80\%).

The data from the experiment, plotted in Figure A-11, support our hypothesis that the response from the grating would be basically the same as the response from the mirror. It’s apparent that both responses are roughly linear with about the same

\(^1\)Richardson Grating Laboratory. 705 St. Paul Street, Rochester NY 14605.

\(^2\)I.e. the rulings are asymmetrically cut.
Figure A-9: CAD model of the experimental setup we used to verify that (A.13) is equal and opposite to (A.12). A diffraction grating and a plane mirror (mostly obscured in the drawing by the grating) are mounted side-by-side on the same motion stage. In our setup the plane mirror is mounted to the end of an air-bearing spindle which is used in other experiments. As the stage is translated in $y$ the response of the two measurement beams is recorded.

slope. This common slope is likely due to a misalignment of the $y$ stage relative to its ideal perpendicular configuration, which couples a small component of the $y$ movement into detectable $z$ translation. The small remaining difference in slope between the two responses is probably there because the grating was likely not set at exactly the Littrow angle, so its response has an additional small component that depends on perpendicular translation, just as a regular tilted mirror would.
A.3.2 Beam Centering Optics

Both of the ideas we presented in the previous section for increasing the allowable tilt of the measurement mirror seem to be failures. Is there any other way to solve this problem? The answer may be yes. We have come up with a third method which seems promising, but we did not have time to attempt to implement it.

Since the tilt in the mirror only displaces the returned measurement beam, it may be possible to install passive beam-centering optics which would correct this shift to some extent. This would extend the allowed mirror tilt beyond the inherent capabilities of the raw PMI optics. One possibility for the beam centering optics is a simple telescope arrangement of a convex and concave lens placed directly in front of the detector, as shown in Figure A-12. As we said, we have not explored this any further.

A.4 Summary

In this Appendix we described the Plane Mirror Interferometer optics (PMI). The PMI optics are a fundamental component of both the tilted-mirror interferometer that we describe in Chapter 2 and the helicoid-mirror interferometer that we describe in Chapter 3. We described all of the details of the PMI relevant to the analyses in Chapters 2 and 3. We also described several potential enhancements to the PMI optics that might be useful in our applications.
Figure A-10: Photo of the experimental setup we used to verify that (A.13) is equal and opposite to (A.12). A diffraction grating and a plane mirror (mostly obscured in the photo by the grating) are mounted side-by-side on the same motion stage. In our setup the plane mirror is mounted to the end of an air-bearing spindle which is used in other experiments. As the stage is translated in $y$ the response of the two measurement beams is recorded.
Figure A-11: Data from an experiment comparing the response of a PMI directed at a Littrow-mount diffraction grating \(d_1\), red) with that of a second PMI directed at a perpendicular plane mirror \(d_2\), green). Both the grating and mirror are attached to the same stage, which is moved perpendicular to the PMI \(y\). We would expect a large difference between \(d_1\) and \(d_2\) if our analysis (in the text) is incorrect. Only a small difference appears though, and this difference is likely due to some deviation in the angle of the grating from exactly \(\alpha_1\). The fact that neither curve has zero slope is probably caused by some deviation in the angle of the \(y\) stage from exactly perpendicular to the measurement beams.

Figure A-12: It may be possible to allow larger measurement mirror tilts by installing some optics that re-center the returned measurement beam. One concept is to use a telescope arrangement of a convex and concave lens, as shown here. We have not explored this further.
Appendix B

Software

We developed four major pieces of software for our implementation of the tilted-mirror interferometer:

- HP 10897A Laser Axis Board Virtual Front Panel (VFP)
- Pentek 4284 C40 DSP Board Controller
- C40 Code
- dSPACE/Simulink Code

We will describe the details of each of these, at both the user and programmer levels, in the following sections.

As we describe in Section 2.8.1, the instrumentation for the tilted-mirror interferometer consists of three HP 10897A Laser Axis Boards [24], a Pentek EPC-5A 486 VME PC [10, 8, 9], a Pentek 4284 C40 DSP board [28], and a dSPACE 1103 prototyping system installed in another PC. The first three of these, the 10897A’s, the EPC-5A, and the 4284, are all installed on a common VME bus. The 4284 communicates with the 1103 over a custom high-speed parallel link (described in Section 2.8.1).

The main control system for the machine runs on the 1103. The C40 is used mainly to manage the high-speed transfer of data from the three 10897A boards to
the 1103. The EPC-5A 486 VME PC is used to configure the 10897A boards and the 4284 board.

The 10897A VFP and the 4284 controller both run on the EPC-5A 486 VME PC under Windows 3.1. Figure B-1 shows a screenshot of the GUI for these programs on the EPC-5A. The C40 code runs stand-alone on the 4824 C40, and has no GUI of its own (the 4284 controller is used to interface with it). The dSPACE/Simulink code runs on the 1103 embedded PowerPC; it has both a development interface, in the form of Simulink blocks (documented below), and a run-time interface through dSPACE ControlDesk. Figure 2-14 shows the ControlDesk interface.

B.1 HP 10897A Laser Axis Board Virtual Front Panel (VFP)

This application allows the user to control an HP 10897A Laser Interferometer Axis board installed on the VME bus. It is designed to run only under Windows 3.1 on a RadiSys EPC-5 with EPConnect software drivers installed.

All associated files are located on the EPC-5A 486 VME PC in the PMC lab.

B.1.1 User Documentation

Each successive concurrent invocation of the application will control a different axis. For example, if three copies of the application are run at the same time, the first copy will control the first axis board, the second copy will control the second board, and the third copy will control the third board. The user may override this behavior by using the “Set Axis...” system menu item, as described below. The VME base address (address space A24) for each axis board is settable in the INI file.

The axis board under control is indicated in the title bar of the application’s main window with a zero-based integer index. The VME base address of the controlled axis board is also provided in the title bar as a C-style hexadecimal integer.

The position and velocity are read from the 10897A board using the board’s Auto
Figure B-1: Screenshot of the tilted-mirror interferometer software running on the EPC-5A 486 VME PC. Three copies of the 10897A VFP are running to control the three axis boards. The 4284 control software is also running, with the Upload and Command dialogs visible.
Sample feature. Only the low 32 bits of the position are currently read, giving a usable range of about 2.68 meters. The source code could be modified with relative ease to read the additional high bits [32...34] if necessary.

The current state of the application is saved when the application is closed and restored at startup. This includes the state of all the dialog box controls, whether auto-update is enabled, and the position of the dialog box on the screen. Separate state is maintained for each axis board.

Controls

All controls are presented in a single dialog-box style window. The window can be minimized using the standard downward-pointing arrow button control at the upper-right corner. A system menu is provided through the standard “-” button control at the upper-left corner. The system menu “close” item should be used to exit the application (alternately, the application can be closed by double-clicking the system menu button, or by exiting Windows).

“Help...” and “About...” options have been added to the system menu which provide more information about the application.

The axis board under control may be changed at any time by selecting “Set Axis...” from the system menu.

Position and velocity are displayed in several choices of units. If the “Auto Update” check is set in the system menu, the position, velocity, and error conditions will be automatically updated every $n$ milliseconds. $n$ is configurable in the INI file. Adding or changing the available position/velocity units requires modifying the source code.

If “Auto Update” is not on then new values can still be read manually by clicking the “Update Axis” button. Note that when auto-update is disabled, the position, velocity, and error indicators will not necessarily reflect the current state of the axis board, but will show the state of the board at the time of the last (manual or automatic) update.

Turning auto update off guarantees that VME bus access cycles will only be
generated to the axis board when necessary in response to the user's manipulation of the controls. It is recommended that auto update be disabled while the real-time metrology code is running on the 4284 C40 so that all possible bus cycles are available to the 4284.

Changes to any dialog-box control take immediate effect. For example, the filter is turned on immediately after the user checks the corresponding box. No reset or update is necessary.

The 10897A filter feature is enabled by setting the “Filter” checkbox. If the filter is enabled then the various filter parameters may be changed by manipulating their respective checkboxes.

The User LED on the front panel of the 10897A may be controlled by manipulating the corresponding checkbox. The 10897A may drive any of the four available COM lines from either or none of the front-panel inputs, as configured by the “Com Drive” controls. The 10897A may also take its A and B measurement inputs from a variety of sources, as configured by the “Source” controls.

In the event of an error or any other malfunction, the 10897A may be reset to the currently displayed settings by clicking the “Reset Axis” button.

The 10897A may report the following errors: Glitch, Unlock, Overflow, Lost B, -Vb, +Vb, -Va, and +Va. If an error is detected the corresponding indicator will light (turn from gray to black). The position and velocity displays will be blanked whenever an error is detected (and will unblank at the first update after the error is cleared).

To clear an error, first fix whatever is causing the error. Most errors are due to blockage or misalignment of the measurement beam. Once the problem has been fixed, click the “Reset Axis” button one or more times until the error LED on the corresponding axis board goes out and the error indicator(s) turn back to gray.

INI File Format

Various settings are stored in the file

c:\vona\axis\axis.ini

123
This filename is hardcoded, changing it would require modification of the source code.

Global settings are stored in the [Global] section of the initialization file. They are shown in the following table.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumAxes</td>
<td>decimal integer</td>
<td>number of installed axis boards</td>
</tr>
<tr>
<td>msUpdate</td>
<td>decimal integer</td>
<td>milliseconds between display auto updates</td>
</tr>
</tbody>
</table>

These are intended as user settings and are read but never written by the application.

Settings local to each axis a are stored in [Axis a] sections. These may be set by the user, however they are written by the application upon shutdown to reflect the current state. These settings are shown in the following table.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMEBase</td>
<td>hexadecimal integer</td>
<td>VME A24 base address of axis board</td>
</tr>
<tr>
<td>AutoUpdate</td>
<td>0 or 1</td>
<td>state of auto update (1 = on)</td>
</tr>
<tr>
<td>PositionUnits</td>
<td>decimal integer</td>
<td>item index of position units to use</td>
</tr>
<tr>
<td>VelocityUnits</td>
<td>decimal integer</td>
<td>item index of velocity units to use</td>
</tr>
<tr>
<td>Filter</td>
<td>0 or 1</td>
<td>state of filter</td>
</tr>
<tr>
<td>Kp1</td>
<td>0 or 1</td>
<td>state of filter parameter Kp1</td>
</tr>
<tr>
<td>Kp0</td>
<td>0 or 1</td>
<td>state of filter parameter Kp0</td>
</tr>
<tr>
<td>Kv</td>
<td>0 or 1</td>
<td>state of filter parameter Kv</td>
</tr>
<tr>
<td>UserLED</td>
<td>0 or 1</td>
<td>state of user LED</td>
</tr>
<tr>
<td>ComDrive1</td>
<td>decimal integer</td>
<td>item index of Com Drive 1</td>
</tr>
<tr>
<td>ComDrive2</td>
<td>decimal integer</td>
<td>item index of Com Drive 2</td>
</tr>
<tr>
<td>ComDrive3</td>
<td>decimal integer</td>
<td>item index of Com Drive 3</td>
</tr>
<tr>
<td>ComDrive4</td>
<td>decimal integer</td>
<td>item index of Com Drive 4</td>
</tr>
<tr>
<td>SourceA</td>
<td>decimal integer</td>
<td>item index of Source A</td>
</tr>
<tr>
<td>SourceB</td>
<td>decimal integer</td>
<td>item index of Source B</td>
</tr>
<tr>
<td>GeometryLeft</td>
<td>decimal integer</td>
<td>screen location of left side of window</td>
</tr>
<tr>
<td>GeometryTop</td>
<td>decimal integer</td>
<td>screen location of top of window</td>
</tr>
</tbody>
</table>

B.1.2 Architecture

The 10897A VFP software is composed of four C modules (main, HP, preferences, delay), one resource script, and one shared header file (common.h).
main (main.c) Contains the application entry point, WinMain(). Also contains all
Windows-related code to manage the dialog box. The lists of position and
velocity units are located at the top of this file.

HP (hp.c, hp.h) All code related to the interface with the HP 10897A board is
included in this module. See the header file for the public interface.

preferences (prefs.c, prefs.h) This module provides functions to get and set prefer-
ences in the INI file. See the header file for the public interface.

delay (delay.c, delay.h) This module provides a crude delay function. See the header
file for the public interface.

resource script (axis.rc, axis.rh) Defines the layout of the dialog box. Use the
Borland Resource Workshop tool to edit this file.

shared header (common.h) Defines application globals and macros.

B.1.3 Building

The 10897A VFP is built using Borland C++ 4.5 under Windows 3.1 on the EPC-5A
486 VME PC. All files are located in the directory
c:\vona\axis\

To inspect or modify the code, open the project file
c:\vona\axis\axis.ide

in the Borland IDE.

B.2 Pentek 4284 C40 DSP Board Controller

This application allows the user to control a Pentek 4284 TMS320C40 DSP board
installed on the VME bus. It is designed to run only under Windows 3.1 on a RadiSys
EPC-5 with EPConnect software drivers installed.

All associated files are located on the EPC-5A 486 VME PC in the PMC lab.
B.2.1 User Documentation

Four main functions are supported: uploading code to the 4284, dumping memory from the 4284, issuing commands to a 4284 running a user command loop, and tracing to file from a 4284 running a user command loop.

Upon execution, the upload dialog box is presented. This is the main interface for the program. Dialog boxes for the other three functions are accessible through the system menu of the upload dialog.

"Help..." and "About..." options have been added to the system menu of the upload dialog (accessible by clicking the "-" button in the upper left corner of the dialog).

The current state of the application is saved when the application is closed and is restored at startup. This includes the visibility of the four dialog boxes and their positions on the screen, as well as the history list of C40 COFF files uploaded.

Uploading code to the 4284

The upload dialog allows the user to choose a COFF format TMS320C40 executable file that will be uploaded to the 4284 [11]. A drop-down history list of recently uploaded files is provided, along with a "Browse..." button which allows the user to select new files. Once a file has been selected, the "Upload" button will load the code in the COFF file onto the 4284. Execution of the code will not commence, however, until the "Go" button is pressed. Execution may be aborted by hitting the "Reset" button, which issues a Host Control Reset ([28], Section 4.2.3) to the 4284.

Upload is implemented using the look/set functions provided in the 4284 firmware. Code to be uploaded must be in the form of a single COFF format executable file. The file should have been compiled with the TI TMS320C40 C compiler version 5.0, and linked with the TI v5.0 linker. See Section B.3.3 for details.

Dumping memory from the 4284

Any of the four (Dual-Port (DPRAM), Global, Local, Internal) memory blocks may be dumped to a file. Dumping the DPRAM occurs without disturbing the C40. Dumping
the other blocks will incur a 4284 Host Control Reset (i.e. user code running on the 4284 C40 will be halted) and will trash some parts of DPRAM. The 4284 firmware look/set functions are used to implement global, local, and internal memory dumps.

The memory dump functionality is accessed through the system menu of the main (upload) dialog box. A new dialog will appear that allows the user to select the memory block to be dumped. When the “Dump to file...” button is clicked, a file selection box appears so that the user can name the dump file. Previously existing filenames can be re-used, and an overwrite warning message will be presented. As described above, unless the dump is from DPRAM, a C40 reset will be incurred (halting any currently running user code) and some DPRAM will be trashed. In this case, a confirmation dialog will be displayed before the reset is issued.

**Issuing commands to the 4284**

When the 4284 is running a user command loop, commands may be issued to it using a dialog box interface. This functionality is accessed through the system menu on the main dialog box, “Command Interface...”, which will bring up a new dialog box.

The user types the command to be issued into the “Command:” control and clicks “send” to issue a command. The C40 may respond to the command in the listbox below the command control, either by providing a status indication, a requested data value, or an error message.

The available commands are set by the programmer’s implementation of the C40 code that is currently running. The command “help” should always be present, and will return a list of the other available commands. For example, the response to “help” may look like this:

```plaintext

group1
  command1
  command2  [int]
group2
  command1  [float]
  command2
```

Then the available commands would be
Tracing to file

When the 4284 is running a user command loop, a trace may be setup and logged to file. A trace consists of a sequence of trace frames taken from the C40 at a regular interval. Each trace frame consists of a sequence of IEEE 32-bit floats, as specified by the constant TRACE_VALUES_PER_FRAME in

c:\vona\4284\ptkcodes.h.

The specific contents of the trace frames (i.e. what the meaning of each float in the frame is) is set by the trace implementation in the C40 user code. The programmer should modify this code to provide whatever trace data is desired.

The trace functionality is also accessed through the system menu on the main dialog, “Trace...”, which brings up an additional dialog box. The trace period can be set (in terms of the number of C40 interrupts between trace frames), as well as the maximum number of trace frames to take before ending the trace. When the “Start Trace...” button is clicked, the user will be prompted for a filename to which the trace data will be directed. Existing files may be overwritten and the user will be prompted to confirm if this is the case. As the trace executes, the current number of frames taken will be displayed.

If there is some error during a trace, for example if the disk becomes full, the trace will be aborted. The user can also click the “Stop Trace” button at any time during a trace to manually abort.

Due to disk buffering, the size of the trace file on disk may not accurately represent the progress of a running trace.

INI file format

Various settings are stored in the file

c:\vona\4284\4284.ini
This filename is hardcoded, changing it would require modification of the source code.

There are three entries in the [Settings] section:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A16Base</td>
<td>hexadecimal integer</td>
<td>base offset in VME A16 space of the 4284 A16 registers (IACK and Host Control registers)</td>
</tr>
<tr>
<td>DPRAMBase</td>
<td>hexadecimal integer</td>
<td>base offset in VME A24 address space of the 4284 DPRAM</td>
</tr>
<tr>
<td>MaxLoad</td>
<td>hexadecimal integer</td>
<td>maximum code size (in bytes) that can be uploaded to the 4284</td>
</tr>
</tbody>
</table>

The [DefaultFiles] section contains the entries in the COFF filename list.

The [*-geometry] sections record the visibilities and locations of the various dialog boxes.

**B.2.2 Architecture**

The 4284 controller software is divided into two components: the front-end user interface, and the back-end interface to the Pentek 4284.

**Front-End User Interface**

The front-end is composed of 6 C modules (main, command dialog, dump dialog, trace dialog, preferences, delay), one resource script, and one shared header.

**main** (main.c) Contains the application entry point, WinMain(). Also contains all Windows-related code to manage the upload dialog box.

**command dialog** (dlgcmd.c, dlgcmd.h) Contains all Windows-related code to manage the Command dialog box. See the header file for the public interface.

**dump dialog** (dlgdump.c, dlgdump.h) Contains all Windows-related code to manage the Dump dialog box. See the header file for the public interface.

**trace dialog** (dlgtrace.c, dlgtrace.h) Contains all Windows-related code to manage the Trace dialog box. See the header file for the public interface.
preferences (prefs.c, prefs.h) This module provides functions to get and set preferences in the INI file. See the header file for the public interface.

delay (delay.c, delay.h) This module provides a crude delay function. See the header file for the public interface.

resource script (4284.rc, 4284.rh) Defines the layout of the dialog boxes. Use the Borland Resource Workshop tool to edit this file.

shared header (common.h) Defines application globals and macros.

Back-End Pentek Library

The back-end interface is compiled as a library called “pentek.lib”. The public interface for this library is provided in pentek.h. The library is composed of 5 C modules (main, high-level command interface, low-level communication interface, control, firmware) and one shared header.

main (ptkmain.c, pentek.h, ptkcodes.h) Contains entry and exit code for the library. pentek.h contains the public interface for the entire library. ptkcodes.h, included by pentek.h, defines the codes used for communication between the 4284 C40 and the 4284 control software running on the EPC-5A 486 VME PC. ptkcodes.h is thus designed to be included both in the 4284 control software and in the C40 software that will talk to it.

high-level command interface (ptkhigh.c) Implements commands for controlling and interacting with the 4284, including miscellaneous, trace-related, and I/O-related commands. The functions in this module are implemented in terms of the low-level communication interface defined in ptklow.c.

low-level communication interface (ptklow.c) Implements a low-level interface for communication between the 4284 control software running on the EPC-5A 486 VME PC and the user code running on the 4284 C40. The protocol is register-based, using four locations in the 4284 DPRAM to implement command, parameter, status, and data registers.
control (ptkctl.c, ptkctl.h) Implements control functions for the 4284. These include functions related to uploading code, resetting the board, and other miscellaneous functions.

firmware (ptkfrmwr.c, ptkfrmwr.h) Implements functions that interact with the firmware on the 4284.

shared header (ptkcommn.h) Defines library globals and macros.

B.2.3 Building

The 4284 controller is built using Borland C++ 4.5 under Windows 3.1 on the EPC-5A 486 VME PC. All files are located in the directory
c:\vona\4284\

To inspect or modify the code, open the project file
c:\vona\4284\4284.ide

in the Borland IDE. Note that this project file contains the Pentek library and the C40 code as sub-projects.

B.3 C40 Code

The Pentek 4284 board on the VME bus is built around a Texas Instruments TMS320C40 floating-point DSP. We use this processor to manage the transfer of data from the 10897A laser axis boards to the dSPACE 1103, where the machine control system runs.

Upon VME power-up, the C40 boots into a firmware environment (loaded from ROM on the 4284). The 4284 controller program is then run on the EPC-5A VME PC and used to upload user software to the 4284, as described in the previous section. The user code file used to run the tilted-mirror interferometer is located at
c:\vona\4284\target\c40.x86
B.3.1 User Documentation

The main function of the C40 code is to manage the transfer of data from the three 10897A laser axis boards to the dSPACE 1103. Physically, the data travels over the VME bus from the 10897A boards to the 4284 (which contains the C40). Then the C40 passes the data on to the 1103 system via a custom-made parallel data link. Basically this link connects one of the COM\(^1\) ports on the 4284 to some of the digital I/O pins on the 1103. More details of the link are provided in Section 2.8.1.

Upon execution, the C40 code performs some initialization and then enters a command loop, where user commands are interpreted and serviced. When the 1103 makes a request (via the link hardware), the C40 enters an Interrupt Service Routine (ISR). The ISR gathers the current position data from the 10897A boards, assembles the data into a packet along with some error detection information, and then sends the packet to the 1103. If any unrecoverable errors are detected during transmission, the C40 will halt with the error message

Error ... Disabling com.

To recover, the user must issue the command

`isr reset`

to the C40, as described below.

Commands

The C40 code provides a minimal run-time user interface. Executing the command

`help`

in the 4284 controller Command interface yields the following list of available commands:

---

\(^1\)Unlike on a PC, where a COM port is typically a serial device, on the C40 a COM port is an 8-bit wide parallel communication link.
isr
  enable
disable
reset
misc
  led
    enable
disable
toggle
trace
get
  period
set
  period [int]

As described above, this means that, for example,

misc led toggle

is a valid command. The functions of each of these commands are as follows:

isr enable Enables the ISR. If the ISR is disabled, the C40 will not respond to
  requests from the 1103.

isr disable Disables the ISR.

isr reset Resets the ISR. This is by far the most often used command, because its
  effect is to re-initialize the ISR if it has been disabled due to a communications
  error. The ISR will disable itself whenever it detects that an unrecoverable
  error has occurred in data transmission to the 1103. This could happen, for
  example, because of a synchronization error due to stopping and re-starting the
  code running on the 1103. During development, this is a frequent occurrence.

misc led enable Turn on the LED on the 4284 front panel. Useful for debugging.

misc led disable Turn the LED off.

misc led toggle Change the state of the LED.
**trace get period** Get the period (measured as the number of ISRs between trace frames) of the current trace (trace is an optional function, as described in Section B.2.1).

**trace set period [int]** Set the trace period (measured as the number of ISRs between trace frames).

### B.3.2 Architecture

The C40 code is composed of 8 C modules (main, ISR, laser, command, I/O, interpreter, trace, miscellaneous) and one shared header file.

**main** (c40main.tic) contains the application entry point, main(). Initializes the 4284, sets up the other software modules, then runs the command loop.

**ISR** (c40isr.tic, c40isr.h) implements the ISR and associated functions.

**laser** (c40laser.tic, c40laser.h) encapsulates the code for communication with the 10897A boards.

**command** (c40cmd.tic, c40cmd.h) implements the low-level register-based command interface between the C40 and the 4284 control program that runs on the EPC-5A 486 VME PC.

**I/O** (c40io.tic, c40io.h) implements an I/O system for user-level interaction. The I/O system is built on top of the services provided by the command module. Effectively, this module implements the back-end for standard printf() and scanf(), so the rest of the C40 application can use these functions for user-level communication with the 4284 control program.

**interpreter** (c40intrp.tic, c40intrp.h) built on top of the I/O module, this module implements functions that other modules use to build their user-level command-line interfaces.

**trace** (c40trace.tic, c40trace.h) implements the trace subsystem.
miscellaneous (c40misc.tic, c40misc.h) functions for managing the LED and other things.

shared header (c40commn.h, lsram.h) shared application header. c40commn.h includes lsram.h, which defines how the C40 code uses the LSRAM present on the 4284.

B.3.3 Building

The C40 code is built using Borland C++ 4.5 under Windows 3.1 on the EPC-5A 486 VME PC. The Borland IDE is used to manage the project, but the Texas Instruments C compiler tools (version 5.00) are actually used to compile and link the code. All source files are located in the directory
c:\vona\4284\target

To inspect or modify the code, open the project file
c:\vona\4284\4284.ide

in the Borland IDE. The C40 code is a subproject of the 4284 controller called “target”.

To compile one of the C modules, open it in the Borland IDE and make sure its window is selected. Then select “TIC Compile” from the “Tool” menu. A DOS window will come up and show the process of the compile, including any error messages. When the compile has finished this window must be manually closed before the IDE will respond.

To re-link the C40 code, right click on the file
target\c40\.x84

in the Borland IDE Project window, then select “C40 Make” from the pop-up menu. Again, a DOS window will appear showing the progress of the link, and this window must be closed before the IDE will respond.

The “TIC Compile” and “C40 Make” IDE tools were custom-developed for this project. They are only available in the Borland IDE when the 4284.ide project is open.
B.4 dSPACE/Simulink Code

As described in Section 2.8.1, the raw PMI data is delivered to a dSPACE 1103 digital control prototyping system via a high-speed (~ 1 MB/s) data link. We have developed a control system for the 5DOF CNC grinding machine using MATLAB/Simulink/Stateflow that runs in real-time on the 1103 (see [21] for details on the control system).

Figure B-2 shows a simplified version of the Simulink layout of the tilted-mirror sensor subsystem. Data from the interferometers enters the system via the com4284 C-coded S-function block. The raw data is conditioned in the PMI block and then sent in parallel to the Mirror Tilt Parameters and Three Beam Transformation blocks. The Mirror Tilt Parameters block selects either a set of fixed Initial Parameters (for use before the auto-calibration data is available) or the parameters derived from auto-calibration, which is performed by the Trace Fit block (Section 2.4).

The Trace Fit subsystem is shown in Figure B-3. External blocks (not shown in the Figure) get the machine spinning at constant speed, and then enable trace_store (by bringing trace_store–Reset low). trace_store accumulates the raw metrology data for a period of time determined by a constant in trace_store.c. When that period elapses, trace_store raises trace_store–Ready, which causes trace_fit to execute, performing the fit. The results of the fit are then read as necessary by trace_results and transformed into the tilted-mirror parameters \( C \), \( A \), and \( \phi \) according to equations (2.24), (2.25), and (2.26).
The contents of Trace Pt are shown in Figure B-3. The contents of Trace Pt are derived from auto-calibration, which is performed by a subsidiary trace. The Mirror Tilt Parameters block selects either a set of fixed initial parameters or the data from the interferometer data. The system via the a-coding S-function called the Mirror Tilt Parameters block selects a set of fixed initial parameters or the data from the interferometer data. The top-level layout of the main tilted-mirror methodology system in Simulink.
Figure B-3: Layout of the Trace Fit subsystem, which implements the auto-calibration procedure described in Section 2.4. The trace-store and trace-results are C-coded S-functions. The tall skinny sub-block is a wrapper around a third C-coded S-function called trace-fit. This allows trace-fit to be called only when trace-store is activated. External blocks (not shown) get the machine spinning at constant speed, and then enable trace-store (by bringing trace-store-Reset low). trace-store accumulates the raw metrology data for a period of time determined by a constant in trace-store.c. When that period elapses, trace-store raises trace-store-Ready, which causes trace-store to execute, performing the fit. The results of the fit are then read as necessary by trace-results and transformed into the tilted-mirror parameters $C$, $A$, and $\phi$ according to equations (2.24), (2.25), (2.26), and (2.27).
Appendix C

Electrical Schematics
Pentek 4284 to dSPACE 1103 RS-422 Data Link Schematic
(Sheet 1 of 2: 4284 Unit)

NOTE: XXX is one pair from a 3M 1700 series twisted pair flat ribbon
cable that connects this RS-422 unit with the corresponding unit at the 1103.
Figure C-2. Pentek 4284 to dSPACE 1103 Data Link: Sheet 2 of 2 (1103 Unit)

Pentek 4284 to dSPACE 1103 RS-422 Data Link Schematic
(Sheet 2 of 2: 1103 Unit)

NOTE: XXX is one pair from a 3M 1700 series twisted pair flat ribbon cable that connects this RS-422 unit with the corresponding unit at the 4284.
Bibliography


