

New Millennium Interferometer Laser Metrology Testbed

by

Julio Ortiz, Jr.  
B.S. EE '96, B.S. Mathematics '96, MIT

Submitted to the Department of Electrical Engineering and Computer Science  
in Partial Fulfillment of the Requirements for the Degrees  
of Bachelor of Science in Physics  
and Master of Engineering in Electrical Engineering and Computer Science  
at the Massachusetts Institute of Technology

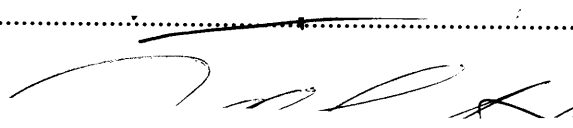
February 7, 1997

Copyright 1997 Julio Ortiz, Jr. All rights reserved.


The author hereby grants to M.I.T. permission to reproduce  
publicly paper and electronic copies of this thesis  
and to grant others the right to do so.

*ex L*

Author .....  
Department of Electrical Engineering and Computer Science  
Department of Physics  
February 7, 1997

Certified by .....  
  
George Vergheese  
Thesis Supervisor

Accepted by .....  
A. C. Smith  
Chairman, Department Committee on Graduate Theses

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
MAR 21 1997  


Eng.

New Millennium Interferometer Laser Metrology Testbed  
by  
Julio Ortiz, Jr.

Submitted to the  
Department of Electrical Engineering and Computer Science

February 7, 1997

In Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Physics  
and Master of Engineering in Electrical Engineering and Computer Science

## **ABSTRACT**

The New Millennium Interferometer Laser Metrology Testbed is a technology demonstration for a key component of the New Millennium Interferometer (NMI). For the success of NMI, NASA's proposed mission to fly a space-based interferometer, control of its position must be very precise. Laser Metrology is a powerful tool that uses interferometry to make hyper-fine distance measurements. The proposal is to use laser metrology on the NMI mission, but there is a concern that it will be difficult to implement. This experiment simulates the difficulties that the real NMI mission will have by attempting laser metrology in a two-dimensional frictionless environment. The final results are very promising, but success is heavily dependent on capabilities of the available equipment.

Thesis Supervisor: George Verghese  
Title: Professor of Electrical Engineering

## Acknowledgements

This thesis would not have been possible without generous input from many voices. My deepest thanks go to the Spatial Interferometry Group at the Jet Propulsion Laboratory, and to my most compassionate thesis advisor, Professor George Verghese.

Within the Spatial Interferometry Group, I would like to extend my deepest gratitude equally to Dr. Stuart Shaklan and Dr. Yekta Gürsel. Dr. Shaklan's inspirational leadership and keen vision for the overall direction truly kept the project on track and in line. Likewise, Dr. Gürsel helped invaluablely in making the details work. He constructed the air puck itself, but his contributions to the software and theory were just as critical. Dr. Shaklan and Dr. Gürsel were truly my full partners, and deserve at least as much credit as I do for the success of this project.

In addition, I would like to specially thank Dr. Michael Shao, Dr. Jeff Yu, and Dr. Mark Colavita for their continuous support in the group for the project. Despite their busy schedules, they always made me feel comfortable when I asked for their help and advice.

I would also like to extend many thanks to Professor George Verghese, my thesis advisor. His usually perfectly succinct advice was always very helpful. Also, I am very grateful for his patience and tolerance as the deadlines drew near.

Finally, I would like to thank the JPL Co-Operative Education Office, and in particular Ms. Linda Rodgers, for keeping me safe and sane through all difficulties.

The research described was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## Table of Contents

|  |    |
|--|----|
| Acknowledgements .....                             | 3  |
| Table of Contents .....                            | 4  |
| List of Figures .....                              | 5  |
| 1. Introduction.....                               | 6  |
| 1.1. Interferometry .....                          | 7  |
| 1.2. SSI .....                                     | 8  |
| 1.3. NMI .....                                     | 9  |
| 1.4. NMI Laser Metrology Testbed .....             | 11 |
| 1.5. Laser Metrology .....                         | 12 |
| 2. Laser Metrology Testbed Components .....        | 14 |
| 2.1. The Air Puck.....                             | 15 |
| 2.2. The Ceiling Stars.....                        | 19 |
| 2.3. The Air Table And Laser System.....           | 20 |
| 2.4. The Control Computer Communication Paths..... | 23 |
| 2.5. The Control Computer .....                    | 26 |
| 2.6. The Software .....                            | 27 |
| 3. The Control Algorithm Derivation .....          | 28 |
| 4. The Software Algorithm .....                    | 36 |
| 5. Results.....                                    | 40 |
| 6. Conclusions.....                                | 44 |
| Bibliography .....                                 | 48 |
| Appendix.....                                      | 49 |

## List of Figures

|   |    |
|---|----|
| Figure 1: SSI .....                               | 8  |
| Figure 2: NMI .....                               | 10 |
| Figure 3: The Air Puck .....                      | 14 |
| Figure 4: The Air Table and Laser System .....    | 20 |
| Figure 5: Communication Paths Block Diagram ..... | 24 |
| Figure 6: The Control Computer .....              | 25 |
| Figure 7: Sample Loop Gain and Filters .....      | 35 |
| Figure 8: X Error, Best Run .....                 | 41 |
| Figure 9: Y Error, Best Run .....                 | 42 |
| Figure 10: Angle Error, Best Run .....            | 43 |
| Figure 11: Fan Force vs. Fan Voltage .....        | 47 |

## 1. Introduction

This experiment is designed to show that an autonomous craft in a two-dimensional frictionless environment can be controlled accurately enough to perform a laser metrology experiment between the craft and stationary apparatus. This will demonstrate the technology exists for a spacecraft to perform laser metrology in the three-dimensional environment of space. Laser metrology is needed in spacecraft, such as a separated spacecraft interferometer, in order to ensure that the distance between the spacecraft is precisely known and controlled. The project's ultimate goal is to build a separated spacecraft interferometer, so this experiment is very important

The experiment was found to work in theory, but a fundamental limitation of the equipment was not discovered until insufficient time was available to correct it. Despite this, a high degree of control was achieved, such that support for the idea of laser metrology in space is warranted. Nevertheless, the high dependence on the limitations and capabilities of the equipment make this perhaps the most important issue to consider for a real separated spacecraft interferometer. The control theory, however, stands up to the challenges and should be usable (in a modified form) in the three-dimensions of space.

## 1.1 Interferometry

An interferometer is an instrument that uses light interference to determine properties of the light, the light source, or the light path. For instance, it can be used to determine the wavelength or speed of light, assemble an image of a far-away object, or measure changes in the light's path length, all depending on how the experiment is arranged and what is measured. In this experiment, only the last two applications (i.e. assembling an image and measuring changes in path lengths) will be considered. The determination of path length changes through interferometry with lasers is called laser metrology, and is discussed shortly. For now, attention is turned to the assembly of an image through interferometry.

An attempt to explain the science behind the use of interferometers as telescopes is beyond the scope of this thesis. It should suffice to say that interferometers can be made that use the light from a far-off star, looked at from different points, and assemble an image based on the light that is found at each point. The intensity of the light at each measurement point is proportional to the Fourier transform of an image of the star, which can be inverted to reveal the image. The interferometer must therefore be able to collect light from several places. The separation between two interferometer collectors is usually called the baseline, and the farther apart the baseline, the farther and smaller the objects it can see. Therefore mobility and a wide baseline are important factors in an interferometer. One can imagine that if collectors could be put in spacecraft and kept in orbit, along with a combiner to perform the interferometry, the collectors could move arbitrarily and as far apart as technology will allow. This is in fact the idea behind spatial interferometry.

## 1.2 SSI

The Separated Spacecraft Interferometer (SSI) is a proposed NASA mission to place a long baseline, many-element interferometer in space. A combiner spacecraft will accept light from many (perhaps a dozen) collector spacecraft that reflect starlight from a target star. A large number of collectors provide a more accurate look at the target star, since the amount of starlight that is received is increased. A space-based interferometer avoids the distortion of the earth's atmosphere, providing a significant sensitivity increase.

Accurate positioning of the spacecraft is necessary in order to perform the calculations of interferometry. This positioning is achieved with radio transmitters, laser metrology, and Kilometric Optical Gyros (KOG's). Laser metrology is described later in detail.

New technologies will be necessary in order to make the SSI work, and whether these technologies will function in space is unknown. For this reason, NASA would like to test many of the key technologies by actually launching into space a smaller, cheaper version of the SSI. This technology verification will be part of the New Millennium Project, under the name New Millennium Interferometer (NMI).

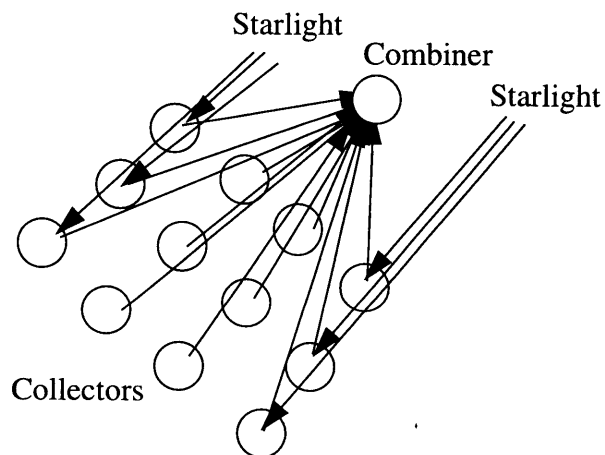


Figure 1

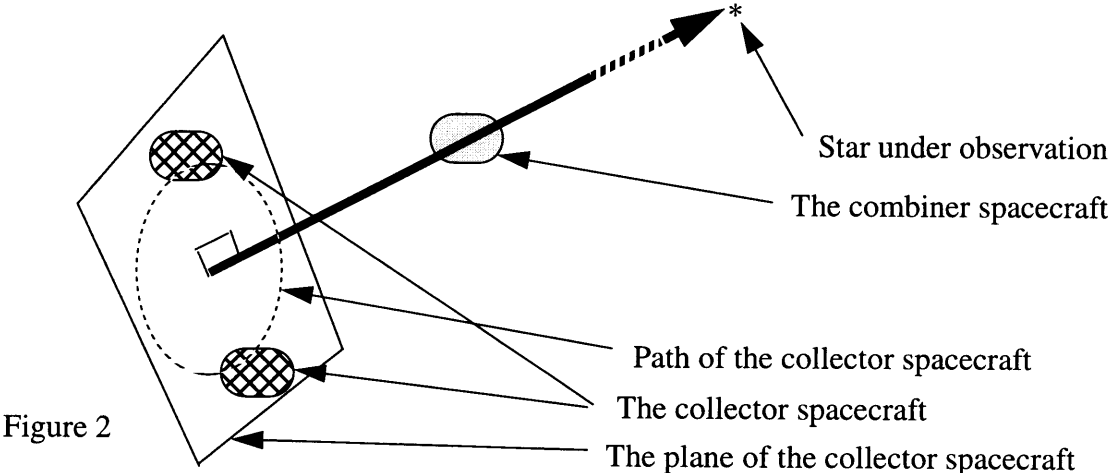


### 1.3. NMI

NASA's New Millennium Project has been tasked to launch several missions near the turn of the century as technology demonstrations for future spacecraft. One of these missions is the New Millennium Interferometer (NMI). This interferometer consists of two collector spacecraft and a combiner spacecraft. The collectors are equipped with mirrors that reflect light to the combiner spacecraft. The combiner takes the incoming light beams and allows them to interfere, yielding a certain intensity for the given separation of the collector spacecraft. The collector and combiner spacecraft are allowed to move relative to each other in certain constrained ways, allowing the combiner to accumulate different intensities for the different positions of the collectors. When enough points in the plane have been accumulated, the (signal-processed) intensity as a function of the points in the plane can be Fourier transformed, and an image generated. This image will be of whatever light-emitting object lies along the line running perpendicular to and out of the plane of the combiners, and running through the collector spacecraft. This geometry is depicted in Figure 2.

Long-range and short-range accuracy in the knowledge of the baseline separation are critical for the NMI. Long-range position is controlled by radio transmitters and receivers which send radio beacons to each other and triangulate, based on separation times. Once the spacecraft are close to their proper position, fine-tuned positioning is controlled by a process called laser metrology. The two collectors and the combiner are equipped with two sets of laser metrology equipment, each for checking the distance between it and one of the other spacecraft. In addition, a Kilometric Optical Gyro (KOG)

controls the overall rotational characteristics of the three spacecraft. The technology behind each of these will be tested by several ground-based testbeds, before NMI performs the actual technology verification in space.



## 1.4 NMI Laser Metrology Testbed

The New Millennium Interferometer will have three separate positioning testbeds. One of these is the New Millennium Interferometer Laser Metrology Testbed. This testbed will serve as a technology demonstration, showing that a spacecraft can be controlled in a multi-dimensional frictionless environment well enough to perform laser metrology. The testbed consists of a heterodyne laser beam for laser metrology and an air table supporting an air puck that in turn supports a small laser metrology plate. The puck is basically a circular sheet of aluminum with attached propulsion, communication, laser metrology, and camera equipment. This puck is supported above the air table by air blowing through small holes in the table. The puck is controlled by a computer next to the air table. The design is such that the puck looks at a pair of LED's in the ceiling through a camera and uses them as guides to move to a desired position on the table. The heterodyne beam is waiting for the puck at that spot. When the puck arrives, the metrology plate on the puck will be directly in the path of the heterodyne beam, and laser metrology will take place between the puck and a stationary corner cube next to the air table. The experiment is successful if the puck is able to reach and hold itself at the metrology lock position on the table long enough for metrology to take place. Ideally it would hold itself there indefinitely.

Before proceeding with a detailed description of the apparatus, an explanation of laser metrology itself is in order.

## 1.5 Laser Metrology

Interferometers can be used to determine to a high degree of accuracy the changes in the path length of light. This process, which uses a laser as the light source, is called laser metrology. It is illustrated in the Laser Metrology Plate portion of Figure 3. First a special laser beam is prepared. The two rectangular orthogonal polarizations are modulated by two RF frequencies, separated by a small (acoustic range) amount in frequency. In this experiment, the RF frequencies are Citizens' Band (CB), and the acoustic frequency is 10kHz. The beam strikes a beam splitter, and half the wave is sent to a photodetector with a circular polarizer mounted in front of its lens. The circular polarizer sends equal portions of the two polarizations into the detector. The signal that is detected here is called the reference signal. The other half of the heterodyne beam continues to a polarizing beam splitter, where one polarization travels straight through to another circular-polarizer-mounted photodetector. The signal here is called the unknown signal. The other polarization is deflected 90 degrees in the cubic polarizing beam splitter, and emerges through another face of the polarizing beam splitter. At this face, and the opposite face, is a quarter wave polarizer. The deflected beam travels through this polarizer as well, and is rotated by 45 degrees in circular polarization. The deflected beam then strikes a corner cube, and returns to the polarizing beam splitter through the quarter wave polarizer. Now rotated by 90 degrees, the beam passes straight through the polarizing beam splitter, and through the other quarter wave polarizer on the opposite side. The beam then strikes another corner cube, and returns to the polarizing beam splitter again, now rotated by a full 180 degrees. The polarizing beam splitter will now deflect the beam again, and so the beam emerges

finally from the opposite face from which it first entered. It therefore also reaches the unknown detector through the circular polarizer. The unknown signal is therefore essentially identical to the reference signal, except it is out of phase by an amount proportional to the distance it has travelled. The change in distance travelled is therefore proportional to the change in phase. If the change in phase can be measured (by counting interference fringes), then likewise the change in distance can be calculated. Special hardware, like the slow laser card in this experiment, can count the number of changes in  $2\pi$  radians (fringes) that go by between two signals. Therefore the change in path length can be measured very accurately through laser metrology. In our experiment, one of the corner cubes is always stationary, so the change in path length will only occur on the axis defined by the two corner cubes. Nevertheless, this is a satisfactory test for the purposes of the experiment.

## 2. Laser Metrology Testbed Components

The New Millennium Interferometer Laser Metrology Testbed essentially consists of an air puck on an air table, propelled by fans, whose position is controlled by a camera (coarse control) and a laser metrology plate (precise control). A computer actually controls the puck by radio-controlling the fans, based on the camera's image and possibly the laser metrology information. Each component of the system is described below.

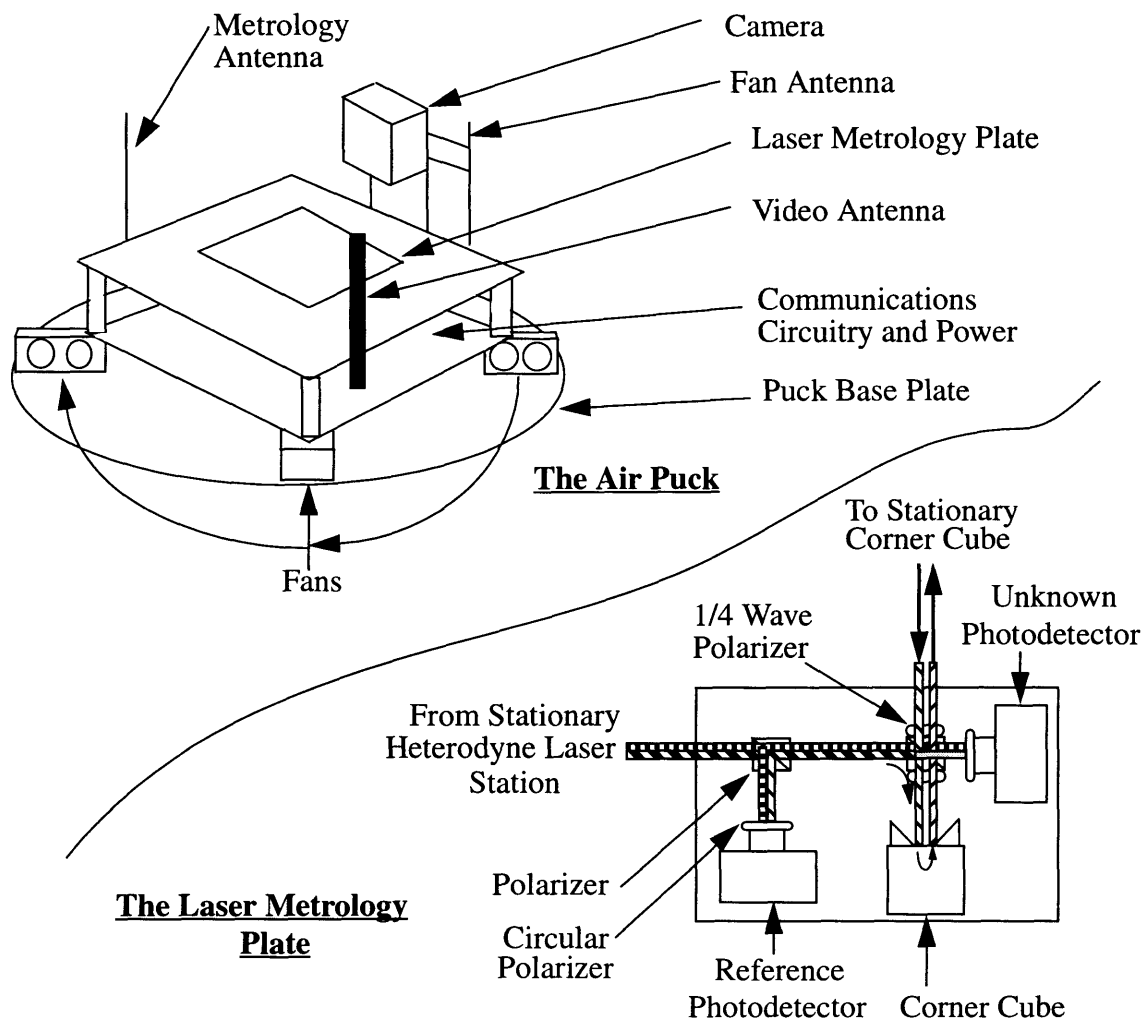


Figure 3  
The New Millennium Interferometer Laser Metrology Testbed Air Puck

## 2.1. The Air Puck

Figure 3 depicts all the main components of the air puck.

The puck can be thought of as consisting of three main bays, or levels. The lowest consists of eight radio-controlled microfans and the base plate upon which the entire structure rests. This base plate is simply a circular sheet of aluminum, fourteen inches in diameter. The fans are 1" diameter unidirectional DC fans, as conventionally used inside computers for cooling. As such, they are not designed for precise linear control from the applied DC voltage source, but nevertheless the output force increases with increased voltage. The exact manner in which it grows has been studied, and is presented later. The unidirectional nature of the fans simply means each fan generates force in only one direction. In order to simulate a bidirectional fan, two unidirectional fans are placed side by side. All fans on the base plate are arranged in such pairs, and the plate has four pairs. Referenced from the center of the base plate, the fan pairs are located 90 degrees apart from each other, on the edges of the plate, and all blow tangential to the plate. With this configuration, translation can be performed in either of two-dimensions by simply turning on like directed fans from the two fan pairs that correspond to a particular direction. Rotation is performed by turning on oppositely directed fans from all the fan pairs. At any time only one fan from each fan pair should be blowing. Each fan pair is designed to accept both positive and negative valued voltages as inputs, with positive corresponding to one of the fans, and negative corresponding to positive voltage for the other fan. The fans were found to in fact not blow air unless given a minimum voltage close to 1 volt, so all fans have a minimum voltage slightly above this threshold. This eliminates the transient lag associated

with starting the fans from a full stop. Therefore the fans are always blowing, but the force that corresponds to a fan at the threshold is very small, especially when compared to the force blown from the opposite fan when it is on.

The middle level of the puck is devoted to communications and power circuitry. Essentially all the electrical circuitry is hidden here, with wires leading to the other components of the puck. The communication equipment consists of three portions: the video, the interferometry signals, and the fan voltages. The fan voltages are inputs, while the rest are outputs. All communication is ultimately between the puck and the control computer. The video is taken from the puck-mounted CCD camera, and is a standard video signal. This is sent through an antenna on an RF signal, set to TV channel 3 or 4. The control computer receives this through a commercial RF video receiver, demodulates it simply through a standard commercial VCR, and then processes this video with the Matrox video frame grabber in the control computer's VME chassis. The interferometry signals are sent through a pair of roughly 900 MHz RF channels run through a single dedicated antenna. The control computer has a separate RF receiver for these signals as well. This receiver splits the signals, one way to a peak detector circuit and A/D card in the VME chassis, and the other to a custom laser metrology card, also in the VME chassis. Finally, after the control computer processes the video signal, it sends eight digital fan voltages through a D/A card on the VME bus, and to a specially modified radio-controlled car transmitter. These signals are sent through the antenna of the transmitter, and received by another antenna on the puck. This signal is processed in the middle level, and modified to both provide a minimum voltage above the fan threshold voltage and split the signal from bidirectional (i.e. both positive and negative) to unidirectional (i.e. positive voltage to the proper fan direc-



tion, and the minimum threshold voltage to the opposite fan).

The power supplies are also in the middle level. The puck uses two rechargeable camcorder batteries for all of its power. Each in fact can be recharged through a pair of connectors for each battery, and a switch that controls whether each battery is on, off, or charging. The batteries are wired such that either one or both of the batteries can be connected. If only one is used, then all systems except the fans will function. If both are on, all systems function. The puck must be recharged after about 4 hours of use if all systems are on and the batteries are new and fully charged. Each recharge reduces this 4 hour lifetime, however, until the batteries must in fact be replaced.

The topmost level consists of the laser metrology plate, the antennas, and the camera. The metrology plate is a square sheet of aluminum mounted less than half an inch above the roof of the middle level, with optical components glued atop it. The optics consists of two photodetectors, two polarizing beam splitters, two 1/4 wave polarizers, two circular polarizers, and a corner cube. Each photodetector has a circular polarizer mounted directly in front of it, and a beam splitter rests close to and in front of each photodetector. One photodetector is the "reference detector", while the other is called the "unknown detector". The detector/circular polarizer/beam-splitter arrangement is all that is needed for the reference detector, since it merely collects a portion of the heterodyne beam and sends its measurements back to the control computer as a reference. The unknown detector arrangement needs the remaining optical components, however. Also, the unknown detector's beam splitter is in fact a polarizing beam splitter. One polarization of the heterodyne beam travels directly into the unknown detector while the other portion is deflected 90 degrees by the polarizing beam splitter. A 1/4 wave polarizer is mounted on the side of

the beam splitter through which the beam is deflected, and on the side of the cubical beam splitter opposite to this. Just beyond the face through which the beam is deflected is a corner cube, which sends this beam back the way it came. All that is needed to “complete the circuit”, so that the deflected light will be detected by the unknown detector, is one more corner cube. The stationary corner cube, located off the puck, takes this role. The laser metrology plate is thus a complete laser metrology system, minus one corner cube.

Three antennas rise from the uppermost level. As explained above, the antennas are used for the video signal, the fan voltages, and the metrology data respectively. Each antenna is on its own side of the puck, and extends upward on the order of 2 to 3 inches. Also, a Pulnix CCD video camera mounted to a bracket points upwards towards the ceiling stars.

## 2.2. The Ceiling Stars

A large collection of black posterboards, representing the blackness of “space”, is double-stick taped to the ceiling above the air puck and air table. The collection is large enough so that no matter where the air puck is on the air table, the puck’s camera will always see only the posterboard. Two LED’s, one green and one red, are mounted on the posterboard. These LED’s represent “stars”. One star is always kept significantly brighter than the other. The separation between the lights is governed by conflicting goals. A small separation provides more mobility on the air table, as it is harder to lose both stars from the field of view. The valid area of the air table is, in fact, determined by ensuring that both stars are always visible, and the puck is physically restrained (through plastic bumpers on the air table) from travelling to areas where this is not true, A large separation, however provides the control computer with a larger angular resolution, as each pixel represents more refinement in the angle.

### 2.3. The Air Table And Laser System

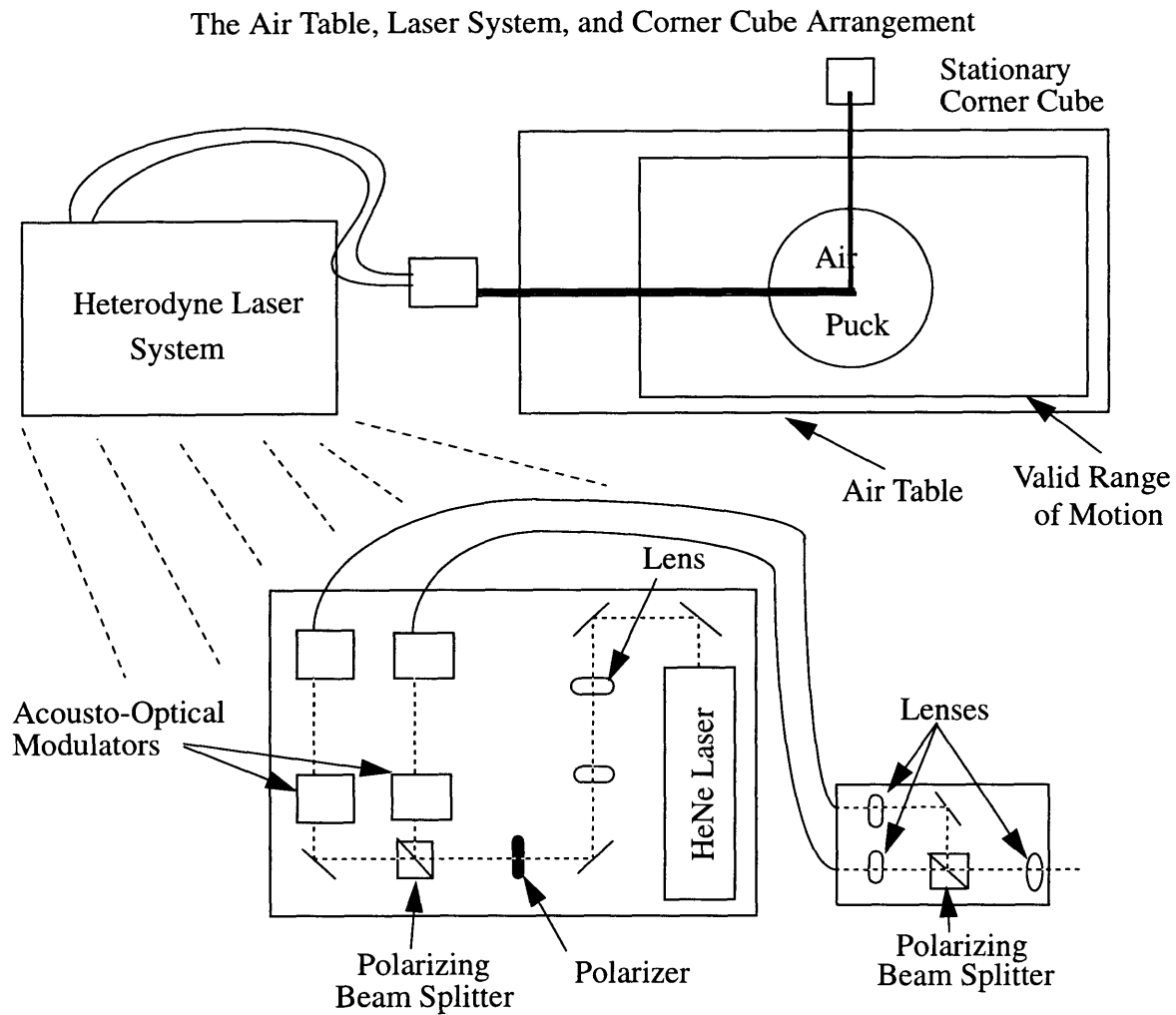


Figure 4 The Components of the Heterodyne Laser System

Figure 4 illustrates most of the relevant features of this section.

The air table itself is a hollow box with wooden top and bottom and aluminum sides. The top is a roughly one inch thick piece of wood with a grid of small holes throughout its surface, each separated about an inch apart. A ring of screws connects the aluminum siding to the wooden top. Also, three screws directly connect the wooden bottom to the wooden top. These screws are adjustable, and keep the table from inflating like

a balloon when the air is turned on, thus increasing the flatness of the air table. The table rests on three more adjustable screws, used to ensure the entire table rests horizontally. A hose extends into the aluminum side from a vacuum pump on the floor. The air pumped into the table from the pump escapes through the small holes in the table's top. This escaping air is quite forceful, and provides plenty of lift for the 14" diameter air puck. In order to ensure the air puck can always see the stars, the valid range of motion of the air puck is restricted by a barrier of plastic bumpers. The valid area represents about two-thirds of the total table area, however, so there is plenty of mobility.

A corner cube is mounted next to the table, by one of the longer sides. This is the stationary corner cube that is used for the laser metrology experiment.

Also next to the table, by one of the shorter sides, is the heterodyne laser system. This consists of two plates, the main plate and the beam launch plate. A Helium-Neon (HeNe) laser beam travels through a pair of focusing lenses to a polarizing beam splitter. Both resulting polarizations pass through their own acousto-optical modulators, set to slightly different CB Radio frequencies (usually 10 kHz apart - this is slight compared to the frequency of the light!), and to their own fiber optic wire for transport to the beam launch plate. At the launch plate, the two polarizations are recombined through another polarizing beam splitter. This is done by sending the pass polarization straight through and sending the opposite polarization perpendicularly to the pass polarization, thus deflecting it directly into the path of the pass polarization. The two beams are then shot together over the air table, for intercept when the puck reaches the lock position. This recombined laser beam, called the heterodyne beam, now consists of two polarizations modulated by slightly different frequencies. When sent through a circular polarizer, like at the reference

and unknown detectors on the puck's laser metrology plate, these frequencies will beat with each other, and the difference frequency (again, usually about 10kHz) will be detected. This is exactly what is needed for the laser metrology experiment.

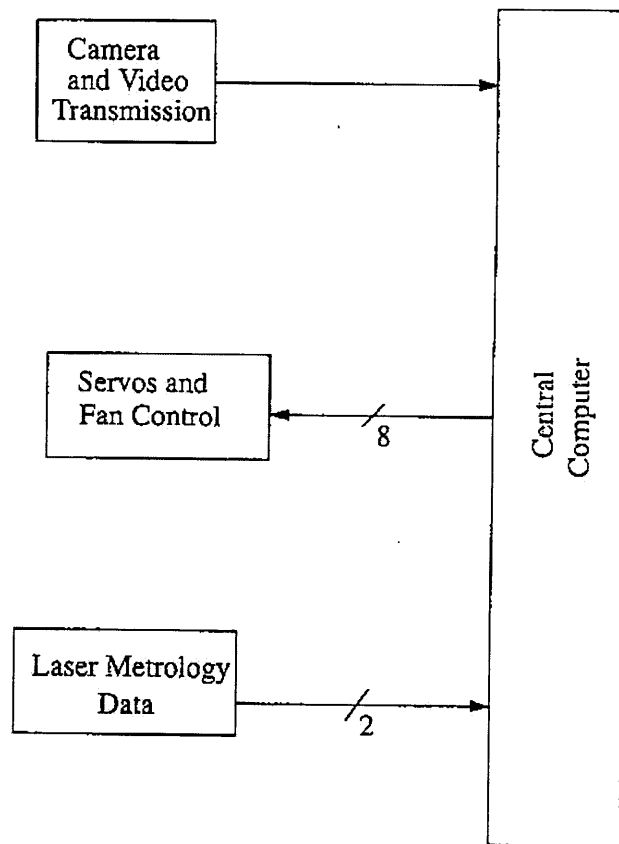
## 2.4. The Control Computer Communication Paths

The control computer and puck speak to each other through three paths - video, fan voltages, and laser metrology. Figure 5 shows a block diagram of the signal paths, while Figure 6 shows most of the apparatus in this and the next section. First, the puck sends RF modulated analog video image data from the CCD camera. This is intercepted by an RF receiver next to the control computer, and sent immediately through a VCR for demodulation. Although the VCR could in principle also record the incoming video stream, as VCR's are designed to do, its only purpose in the experiment is to demodulate the signal. The demodulated signal is then sent to the Matrox VIP-640A frame grabber in the control computer for digitization and analysis in the software. The frame grabber also has a video output, and a video monitor is attached for immediate viewing of the incoming data.

The laser metrology data are sent as audio data on a pair of roughly 900 MHz RF carriers. A receiver next to the control computer demodulates the two signals, and a post-amplifier below it amplifies the signal. The unknown signal is then sent directly to the slow laser card in the control computer, while the reference signal is split. One path also goes to the slow laser card, while the other passes through a full wave rectifier circuit to the screw panel for the Data Translation dt1401 A/D card in the control computer. The rectified signal at the A/D is used by the control computer to determine if the puck is receiving enough laser light to be considered in a lock. The slow laser card, meanwhile, uses the reference and unknown signals to count the number of fringes that pass by. It also uses a required clock signal from the Wavetek Frequency Generator.

The fan voltage values are determined in the software based on the video and

metrology inputs. The desired voltage is sent out as a 16-bit integer to the DVME626 D/A card. The resulting analog values are then sent to a screw panel, to which are attached four pairs of wires. Each pair corresponds to the voltage for a particular fan pair on the air puck. The wire pairs connect to a specially modified radio-controlled car transmitter. Four of the channels on the transmitter have been altered to transmit the DVME voltages. These voltages are received at the puck and translated to properly move the fans, as described earlier.



Block Diagram of Communication Paths Between  
The Central Computer and The Air Puck

Figure 5



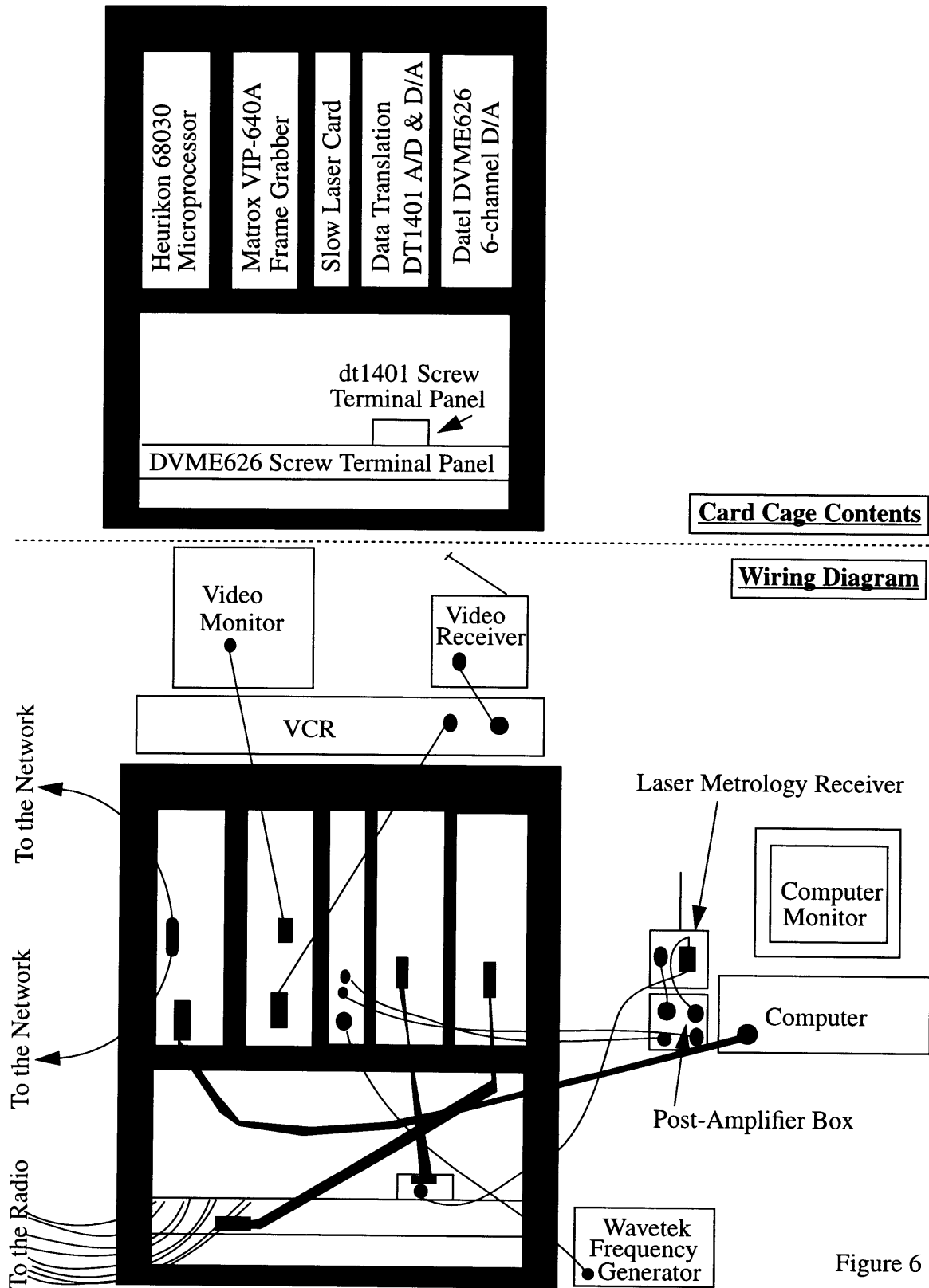


Figure 6

## 2.5. The Control Computer

Most of the control computer has been described already, but a few important elements remain. The control computer consists of: a VME bus with five cards attached; the various communication paths described above; an interface computer and monitor; and an ethernet connection to a network of computers. The five cards are a Heurikon 68030 microprocessor, a Matrox VIP640A frame grabber, a slow laser card, a Data Translation dt1401 A/D, and a Datel DVME626 D/A. The 68030 runs all of the software and controls all the other cards. Meanwhile, the VIP640A digitizes the incoming video signals, the slow laser card determines the number of fringe counts detected, the dt1401 digitizes the lock signal, and the DVME626 sends the fan voltages that the software selects out to the radio-controlled car transmitter. The interface computer, an IBM 386, simply provides a means for keyboard input to the VxWorks operating system on the 68030, with the monitor, a standard Sony black and white model, showing this input and the associated output. Finally, the ethernet connection allows for file transfer between the 68030 and the rest of the network. Files can be written to and accessed from the network. In fact, all of the software was written and compiled (with the 68030 VxWorks C compiler) on a Sun workstation, and then downloaded to the 68030. This is especially nice since the control computer has no permanent storage media of its own.

## 2.6. The Software

The 68030 runs VxWorks as its operating system, and all the software is in VxWorks. VxWorks is a “C-like” interpretive operating system, designed for real-time applications. Programs in VxWorks look almost identical to C programs, but with the power of an operating system to boot. Special VxWorks software was previously written for controlling all the cards in the VME bus, and was included in the main program as header file attachments. The details of the software algorithm, as well as the original code, are presented later.

### 3. The Control Algorithm Derivation

The air puck's position, when laser metrology is not available, must be controlled using only the known (approximately) linear behavior of the fans and the position of the stars in the sky (determined by the camera). It is simpler to manipulate one point in the sky instead of two points, so the control algorithm will concentrate on ensuring that the average position of the stars as observed by the camera matches the desired position. If this condition is met, then all that is necessary is to ensure the puck is in the right orientation. This just means the angle that the stars make when the camera views them at some point must match the angle they should make when the puck is in the position to get a metrology signal. If this additional restriction on the angle is met, then the puck must in fact be in position for metrology. So control relies solely on knowledge of the average position of the stars and the angle between them, along with knowledge of the fan behavior.

In the appendix is a description of how propellers and fan motors usually work. Unfortunately, the puck's fans did not empirically follow this theory. Experiments, discussed later, showed that the force the fans blew with were linear with the fan voltage. This actually simplifies the analysis, although the explanation is a mystery. The theory developed here will use the linear force-voltage relationship found empirically, as this is a better model for how the puck will behave.

The equation of motion for the air puck on the air table is simply  $\bar{F} = m\bar{a}$ , where  $\bar{F}$  is the force GV applied by the fans, where G is a proportionality constant. If the air table is not quite perfectly flat, however, then an additional constant term can be added to account for the pull of gravity. This constant term can also represent in general any first-order per-

turbations to the state of a perfectly flat table. We will assume, however, that it is gravity and represent it as  $mg$ , where  $g$  is the constant acceleration due to gravity. The equation of motion then becomes  $\bar{F} + m\bar{g} = GV_x\bar{x} + GV_y\bar{y} + m\bar{g} = m\bar{a}$ . The problem is now to set  $V_x$  and  $V_y$  so that the puck will go to the correct spot on the air table. In setting these voltages, the only available information is the current position of the puck. The voltages  $V_x$  and  $V_y$  are thus functions of only the position  $\bar{x}$ , as measured by the camera.

The strategy for control will be to make the force of the fans a function of the current position, velocity, and acceleration, where the latter two are approximated numerically. These derivatives are represented discretely as differences between current positions and previous positions, saved in memory. By using derivatives, an artificial friction can be introduced and used to slow down the puck. Also, integrals of the position can be introduced by representing the force  $F$  of the fans as a derivative.

If the force contains no integration, then (working in only the  $x$  direction, and continuously, and with final desired position  $x_0$ )  $V = \sum a_i(x-x_0)^{(i)}$ , so  $GV + mg = G\sum a_i(x-x_0)^{(i)} + mg = mx''$ . The steady-state solution of this is  $x = x_0 - mg/(Ga_0)$ . As this is not the desired  $x = x_0$ , this control does not work. We therefore introduce integration of the position.

The simplest case to try is

$$\frac{d}{dt}(V) = \left( \sum_{i=0}^{\infty} a_i(x-x_0)^{(i)} \right)$$

This yields:

$$GV + mg = mx''$$

or

$$G \frac{d}{dt}(V) = G \sum_{i=0}^{\infty} a_i (x - x_0)^{(i)} = mx'''$$

For the sake of simplicity, and so that we can model perturbations to the solution exactly (see below), we should eliminate terms in the sum of order  $i > 2$ . We then get

$$G \frac{d}{dt}(V) = Gc(x - x_0) + Gbx' + Gax'' = mx'''$$

or

$$x''' - \frac{Ga}{m}x'' - \frac{Gb}{m}x' - \frac{Gc}{m}(x - x_0) = 0$$

This has general solution

$$x = x_0 + Ae^{r_1 t} + Be^{r_2 t} + Ce^{r_3 t}$$

where  $r_1, r_2$ , and  $r_3$  are the three (presumed distinct) roots to the characteristic equation

$$r^3 - \frac{Ga}{m}r^2 - \frac{Gb}{m}r - \frac{Gc}{m} = 0$$

The solution to the differential equation is acceptable if all three roots have negative real parts and decay within an acceptable amount of time.

One might question whether the terms a or b coming from the voltage control equation can be set to zero, and thus simplify the control algorithm. The answer is no. If we write the cubic equation as

$$(r-r_1)(r-r_2)(r-r_3) = r^3 - (r_1+r_2+r_3)r^2 + (r_1r_2+r_1r_3+r_2r_3)r - r_1r_2r_3 = r^3 - \frac{Ga}{m}r^2 - \frac{Gb}{m}r - \frac{Gc}{m}$$

we can analyze the situation more carefully.

Given the general cubic  $f(r)=r^3+a_1r^2+b_1r+c_1$ , if  $a$  is a complex root, then so is  $\bar{a}$ , the complex conjugate of  $a$ , since  $f(\bar{a})=(\bar{a})^3+a_1(\bar{a})^2+b_1(\bar{a})+c_1=(\bar{a})^3+\bar{a}_1(\bar{a})^2+\bar{b}_1\bar{a}+\bar{c}_1=\overline{f(a)}=0$ .

Thus if  $f(r) = (r-r_1)(r-r_2)(r-r_3)$ , we require either all three are negative real numbers or (without loss of generality)  $r_3=\bar{r}_2$ ,  $r_2$  has a negative imaginary part, and  $r_1$  is a negative real

number. In the first case,  $f(r) = (r-r_1)(r-r_2)(r-r_3)=r^3-(r_1+r_2+r_3)r^2 + (r_1r_2+r_1r_3+r_2r_3)r - r_1r_2r_3=r^3 + a_1r^2 + b_1r + c_1$  means  $a_1, b_1, c_1 > 0$ , as seen by simply inserting the negative values for  $r_1, r_2$ , and  $r_3$ . In the second case, with  $r_3 = \bar{r}_2$ , then  $f(r)=r^3-$

$(r_1+r_2+\bar{r}_2)r^2+(r_1(r_2+\bar{r}_2)+r_2\bar{r}_2)r-r_1r_2\bar{r}_2=r^3-(r_1+2*\text{Re}(r_2))r^2 + (r_1*2*\text{Re}(r_2)+|r_2|^2)r - r_1|r_2|^2$ , and

since  $r_1$  and  $\text{Re}(r_2)$  are negative real numbers, and  $|r_2|^2$  is a positive real number, then again  $a_1, b_1, c_1 > 0$ . So in general, for a cubic equation to have roots with purely negative real parts, the coefficients  $a_1, b_1$ , and  $c_1$ , must be purely positive.

Applying this to our equation

$$(r-r_1)(r-r_2)(r-r_3) = r^3 - (r_1+r_2+r_3)r^2 + (r_1r_2+r_1r_3+r_2r_3)r - r_1r_2r_3 = r^3 - \frac{Ga}{m}r^2 - \frac{Gb}{m}r - \frac{Gc}{m}$$

we see that we require  $a, b, c < 0$ , so all three terms in the control equation are necessary.

Using the general solution,  $x-x_0=Ae^{r_1t}+Be^{r_2t}+Ce^{r_3t}$ , we get

$$\frac{d}{dt}(V) = A(c+br_1+ar_1^2)e^{r_1t} + B(c+br_2+ar_2^2)e^{r_2t} + C(c+br_3+ar_3^2)e^{r_3t} = A\frac{mr_1^3}{G}e^{r_1t} + B\frac{mr_2^3}{G}e^{r_2t} + C\frac{mr_3^3}{G}e^{r_3t}$$

so

$$V = -\frac{mg}{G} + A\frac{mr_1^2}{G}e^{r_1t} + B\frac{mr_2^2}{G}e^{r_2t} + C\frac{mr_3^2}{G}e^{r_3t}$$

Finally, note that  $v=dx/dt$  is:

$$v = Ar_1e^{r_1t} + Br_2e^{r_2t} + Cr_3e^{r_3t}$$

If we say  $x(0)=x_i$ ,  $V(0)=V_i$ , and  $v(0)=v_i$ , then in matrix form:

$$\begin{bmatrix} 1 & 1 & 1 \\ \frac{m}{G}r_1^2 & \frac{m}{G}r_2^2 & \frac{m}{G}r_3^2 \\ r_1 & r_2 & r_3 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} x_i - x_o \\ V_i + \frac{mg}{G} \\ v_i \end{bmatrix}$$

With some work, the inverse is found to be

:

$$\begin{bmatrix} \frac{r_2r_3}{(r_1-r_2)(r_1-r_3)} & \frac{G}{m(r_1-r_2)(r_1-r_3)} & \frac{-r_2-r_3}{(r_1-r_2)(r_1-r_3)} \\ \frac{r_1r_3}{(r_2-r_1)(r_2-r_3)} & \frac{G}{m(r_2-r_1)(r_2-r_3)} & \frac{-r_1-r_3}{(r_2-r_1)(r_2-r_3)} \\ \frac{r_1r_2}{(r_3-r_1)(r_3-r_2)} & \frac{G}{m(r_3-r_1)(r_3-r_2)} & \frac{-r_1-r_2}{(r_3-r_1)(r_3-r_2)} \end{bmatrix}$$

So the coefficients A,B,and C can be rewritten in terms of the initial values as:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \frac{(GV_i + mg)}{m(r_1-r_2)(r_1-r_3)} + \frac{r_2r_3(x_i - x_o)}{(r_1-r_2)(r_1-r_3)} + \frac{(-r_2-r_3)v_i}{(r_1-r_2)(r_1-r_3)} \\ \frac{(GV_i + mg)}{m(r_2-r_1)(r_2-r_3)} + \frac{r_1r_3(x_i - x_o)}{(r_2-r_1)(r_2-r_3)} + \frac{(-r_1-r_3)v_i}{(r_2-r_1)(r_2-r_3)} \\ \frac{(GV_i + mg)}{m(r_3-r_1)(r_3-r_2)} + \frac{r_1r_2(x_i - x_o)}{(r_3-r_1)(r_3-r_2)} + \frac{(-r_1-r_2)v_i}{(r_3-r_1)(r_3-r_2)} \end{bmatrix}$$



The initial voltage and initial velocity will probably both be zero, but this set of equations provides the flexibility to include them. Inserting the proper values into these equations and then substituting these for the coefficients A, B, and C, yields a full characterization of the motion of the air puck.

The best approach at this point for analyzing the control algorithm is to choose the roots directly, and set a, b, and c to match these roots. The solutions for various choices of the roots can be modeled on a computer. Parameters to watch for as various roots are chosen include the time for the puck to reach the desired location, and the maxima and minima of the voltage. The time to decay is determined by the smallest valued real part of the roots. The maxima and minima are more difficult to predict, but they should be kept within the maximum and minimum allowable voltages (17 Volts for each fan). Once the behavior is satisfactory, these roots will be used in the discrete time version of the control algorithm.

The computer cannot use the continuous control algorithm described above, but it can approximate it as well as possible. If we write the control algorithm in terms of time differences, instead of differentials, the equation becomes:

$$V(t + \Delta t) = V(t) + c\Delta t(x(t) - x_0) + b(x(t) - x(t - \Delta t)) + \frac{a}{\Delta t}((x(t) - x(t - \Delta t)) - (x(t - \Delta t) - x(t - 2\Delta t)))$$

or written in a more computer-friendly manner:

$$V[n + 1] = V[n] + c\Delta t(x[n] - x_0) + b(x[n] - x[n - 1]) + \frac{a}{\Delta t}((x[n] - x[n - 1]) - (x[n - 1] - x[n - 2]))$$

This along with the discretized equation of motion for a constantly accelerating mass in the constant gravity field,

$$x[n + 1] = x[n] + (x[n] - x[n - 1]) + \frac{1}{2}(\Delta t)^2(GV[n] + mg)$$

should give the desired outcome. This discrete-time version can be directly analyzed by simply having a computer perform each of the finite steps, given initial values. The behavior, if  $\Delta t$  is small enough, should almost exactly match the behavior from the continuous case. The control algorithm is then used in the program, while the puck actually acts out the motion simulated by the above equation of motion.

An example of the open-loop gain of the system is illustrated in Figure 7. This is a typical open-loop gain plot for a proportional-integral-differentiator controller, such as the system described above. There is a sharp 60 dB/decade drop until some critical frequency, after which the drop is at 20 dB/decade. From control theory, we learn that we should aim for a decay rate of 20dB/decade at the frequency for which the loop gain is unity (cross-over frequency). If the roots are picked directly, however, then the unity gain point can be controlled.

The system tends to generate a lot of noise through various means. Noise on the communication paths, quantization error, and noise from the power supplies all give rise to noise on the estimation of the star position, for instance. One way to reduce this noise is to simply low-pass filter the star position estimates in the software. The effect two of filters is shown on the open-loop gain plot. Especially important is that the filter effects do not occur until well past the open-loop unity gain point. This therefore has a small but noticeable effect of smoothing the star position estimates.

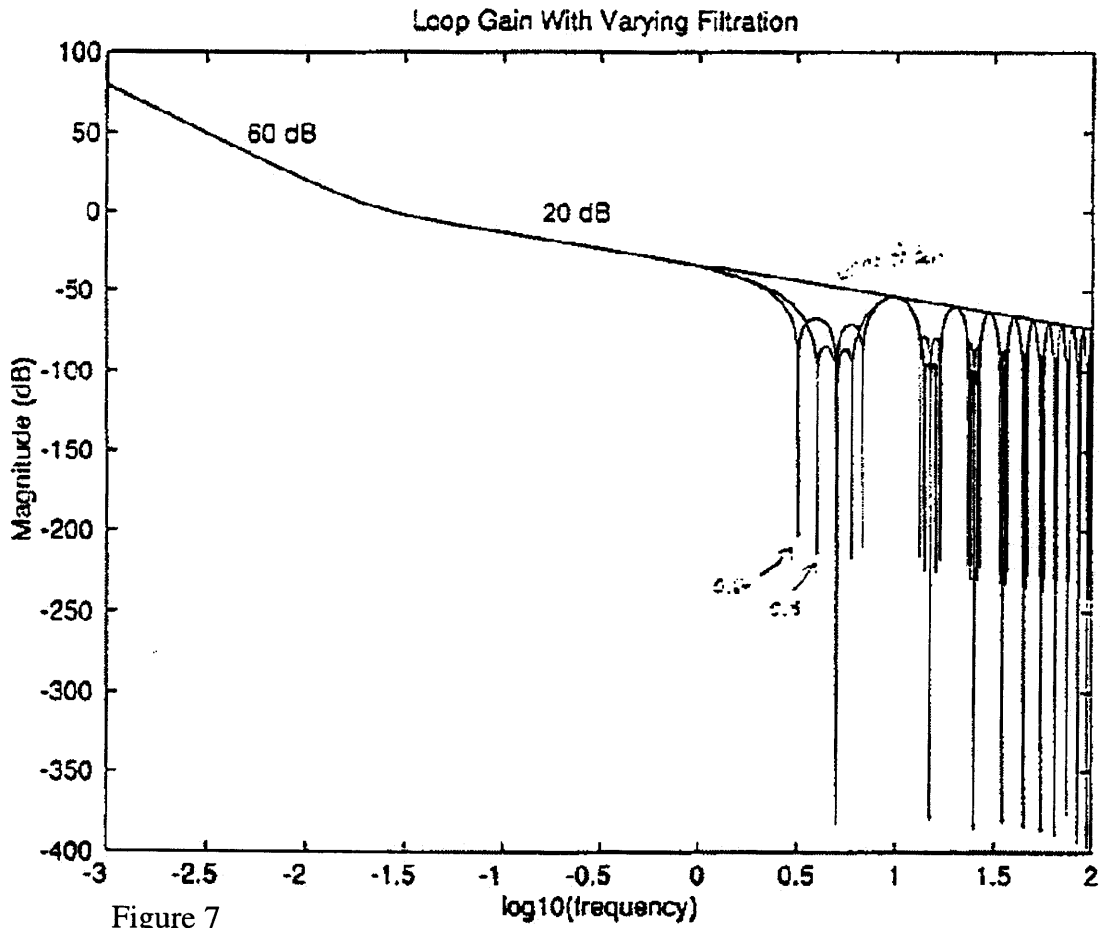


Figure 7

#### 4. The Software Algorithm

The VxWorks software is the implementation of the control system for the puck. It takes as inputs the digitized video signal from the VIP640A card, the laser metrology lock signal from the dt1401 card, and the fringe counts from the slow laser card, and outputs the necessary fan voltages through the DVME626. The video signal gives the puck's position, velocity, and acceleration, while the lock signal tells the fringe counting to begin, and the fringe counts themselves give a fine-tuned position estimate in one dimension. The fan voltages, meanwhile, tell the puck how to move to reach the lock position and angle. A copy of the actual program is located in the appendix.

The software begins by initializing all the cards on the VME bus for operation. It then initializes all variables. Next, it runs the puckcontrol routine once. If this returns successfully, then the puckresume routine is set to run at every system auxiliary clock interval. Then puckcontrol runs again. The clock interrupt is then enabled and the main program ends. Control is thus given to the puckcontrol and puckresume routines. While puckcontrol is an infinite loop routine (if there are no errors, that is), it begins each loop with a taskSuspend command, so in fact the program only runs every time puckresume orders a taskResume. The intervals at which puckresume does this is a user-defined multiple of the base clock interval, 1/60 of a second. In practice, puckresume usually would taskResume once every 5 to 7 clock ticks, so the puckcontrol routine ran at a rate between almost 9 Hz( $60 / 7$ ) and 12 Hz ( $60 / 5$ ).

Puckcontrol runs in two modes. Either way, it is the heart of the control algorithm. The first is the initialization mode, and only runs once. It is the first run that puckcontrol

performs. In this mode, the VIP640A is ordered to digitize an image and dump all of it into an array. The entire array is searched for values above a certain user-defined minimum threshold. Anything above this threshold is considered to come directly from the star, while anything below this is considered background light. When the search first discovers a starlight point in the array, it averages the array positions over all starlight points within a square of user-defined side length (in pixels), weighted by the intensity at each starlight point. This yields an estimate for the star's actual position in the array. All elements of the averaged square are then completely zeroed out, so that the search can resume for the second star. When the next starlight point is found, the process is repeated for the second star. If more or less than 2 stars are found, an error is generated and the program ends. If only 2 stars are found, then the program sorts them so that the brighter star is called star 0, and the other is star 1. The rest of the program runs identically to the second mode of puckcontrol, minus the second modes' star search.

Puckcontrol in the second mode is an infinite loop, but it begins with a taskSuspend This makes it wait until puckresume, which is set to the clock interrupt, orders a taskResume. Once this occurs, the VIP640A is ordered to digitize an image and dump only two small regions to two arrays. Each region is selected based on the last calculated position of the stars, with each array corresponding to one star. The averaging described before is performed within each small region, and the offset from center of each star is used to modify the last known star position. The regions are user-defined squares as before, and thus should have a side length large enough to take into account possible movements of the star position in the frame while between runs of puckcontrol. These small modifications to the star position are all that is required for the star search, and the

program procedures in this mode and in the initialization mode are henceforth essentially identical, and described below.

Once the star searches are completed, the program checks that the stars are not too close to the edges of the visible area to give incorrect results. If this is the case, the program notifies the user and exits. Otherwise, the program next determines the position and angle data from the star locations (in pixels) found earlier. The program determines the angle of rotation in the absolute frame of the lock position by determining the angle the stars make in the puck's current coordinate system, and offsetting this by the angle the stars should make in the lock position's frame. It converts this to an angle between  $-\pi$  and  $\pi$ . Next, it uses this angle to rotate the star positions in pixels, such that the star positions will now be off from the lock position only through a rectangular coordinate translation. It then determines the difference in pixels in the x- and y-coordinates between the average star position in the puck's coordinate system (with the rotation undone as above) and the average star position in the lock position. This difference is then multiplied by the conversion factor between pixels on screen and meters on the air table, yielding the amount of translation that remains in both of the coordinate axes.

The next stage is to use the known errors in position and angle to correct the puck's position. This is performed in a separate subroutine called `CoarseTransandRot`, which accepts these three error signals as input. It first determines the desired voltages for each of the three directions (x, y, and angle) using the exact discrete-time control algorithm described earlier. It then converts the translation (x, y in the lock position's coordinate system) fan voltages to the coordinate system on the puck, bounded between  $\pm 15$  Volts. It then undoes this conversion and uses the result as the new value for the voltages in the

software. If the bounding operation had no effect, then the voltage value should be identical to the calculated value. If the bounding did change the voltages, however, now the voltages in the algorithm are those the fans will actually use. Next, the software determines the maximum voltage that can be put on any of the four fans such that none of them reaches  $\pm 7$  Volts. If one-quarter of the rotation voltage (divided by four because four fans will be blowing this same amount) exceeds this, it simply uses this maximum value for all four fans. Otherwise the calculated rotation voltage is used. All of the voltages for each fan are added together, and the total voltage for each fan is sent to the DVME626 D/A channels. Control then returns to the puckcontrol routine.

Puckcontrol next discards the oldest coordinate data, and makes room in the position and angle arrays for the data in the next run by swapping all the current data back by one spot in the arrays. Next, the dt1401 checks whether the metrology data is ready by seeing whether the metrology lock signal exceeds the (user-defined) minimum threshold. If not, the program loops and begins anew. If there is a lock, however, the slow laser card is ordered to read its counters. This yields the number of fringe counts that have passed by. This is multiplied by a constant to determine the change in position in the metrology axis. This change can be stored in a file, being that it is the desired result of the experiment, or one could ambitiously try to use this to fine tune the position of the puck (although the fan propulsion system is not really sensitive enough to properly use the fine data from the metrology system). In any case, once these calculations are finished, the puckcontrol routine loops and waits for a new taskResume from puckresume. This ends one run of the software, and the experiment proceeds by simply running an endless loop of such runs.

## 5. Results

The experiment consists of simply turning on all the apparatus, placing the puck somewhere on the air table, letting it go, and recording the results. The ultimate test that the experiment works is to actually receive fringes from the puck. Unfortunately, although the puck itself was fully functional, a satisfactory lock could not be achieved. Another good test is that the laser metrology system is locked. Unfortunately again, this could not be reliably achieved, for the reasons below. Yet another test is to monitor the star's position in software. This was possible, and helped to reveal some of the system's problems.

Presented in Figures 8, 9, and 10 are the best runs in x, y, and angle, respectively. Given the almost 1 centimeter diameter of the incoming beam, a wiggle in either x or y position on its own of about  $\pm 3$  millimeters about the lock position could be acceptable, and if both are allowed to wiggle then both wiggles should be kept to within about  $\pm 1.5$  millimeters. In these best runs, the standard deviations are on the order of 1.1-1.3 millimeter, which should be quite acceptable. Typical runs, as opposed to these best runs, were usually close to 1.3-1.5 mm in standard deviation, which was still quite acceptable. Simply watching the puck, the casual observer could rarely see it move at all in position.

The angle requirements were more stringent than were at first realized, however, and presented the single greatest problem in the end. A simple calculation reveals the beam deflection Y for a given angle A off the lock angle and distance X between the puck and the stationary corner cube is  $Y = 2X \tan(A)$ . So, given a maximum deflection Y of  $\pm 3$  mm as before, a separation X of about 1 foot (roughly  $25 \text{ mm/inch} * 12 \text{ inches/foot}$ ), the maximum angle A that can be permitted is  $A = \arctan(Y/(2X))$ , or  $A = \arctan(\pm 3/600) =$



$\arctan(\pm 1/200)$ , which is roughly  $\pm 5\text{mRad}$ . The  $\pm 3\text{mm}$  is in fact a bit too generous, because it again ignores the wiggles in the x and y coordinates. A more realistic bound would probably be about  $\pm 1\text{mm}-1.5\text{mm}$ , which then yields a maximum angle of between  $\pm 1.5-2.5\text{ mRad}$ . The best run of the data had a standard deviation of about  $4.5\text{mRad}$ , so it would have been acceptable by this criterion, but this best was not very typical. Usual runs had a standard deviation of closer to 6 or 7 mRad, which was clearly not acceptable. The wiggles in angle were usually visible on the table, exacerbated as they were by the small position lock wiggles as well. Explanations for the angle lock failure are presented below.

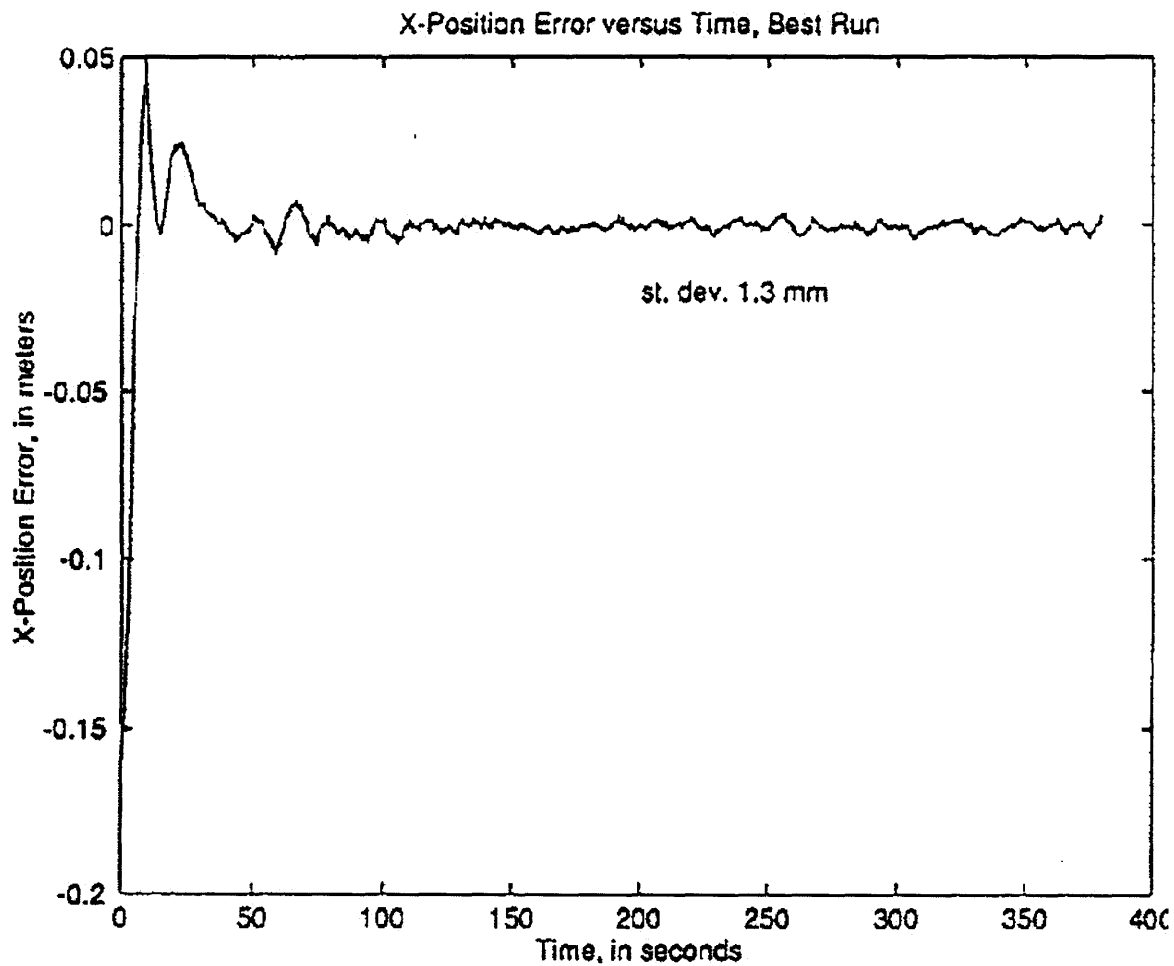
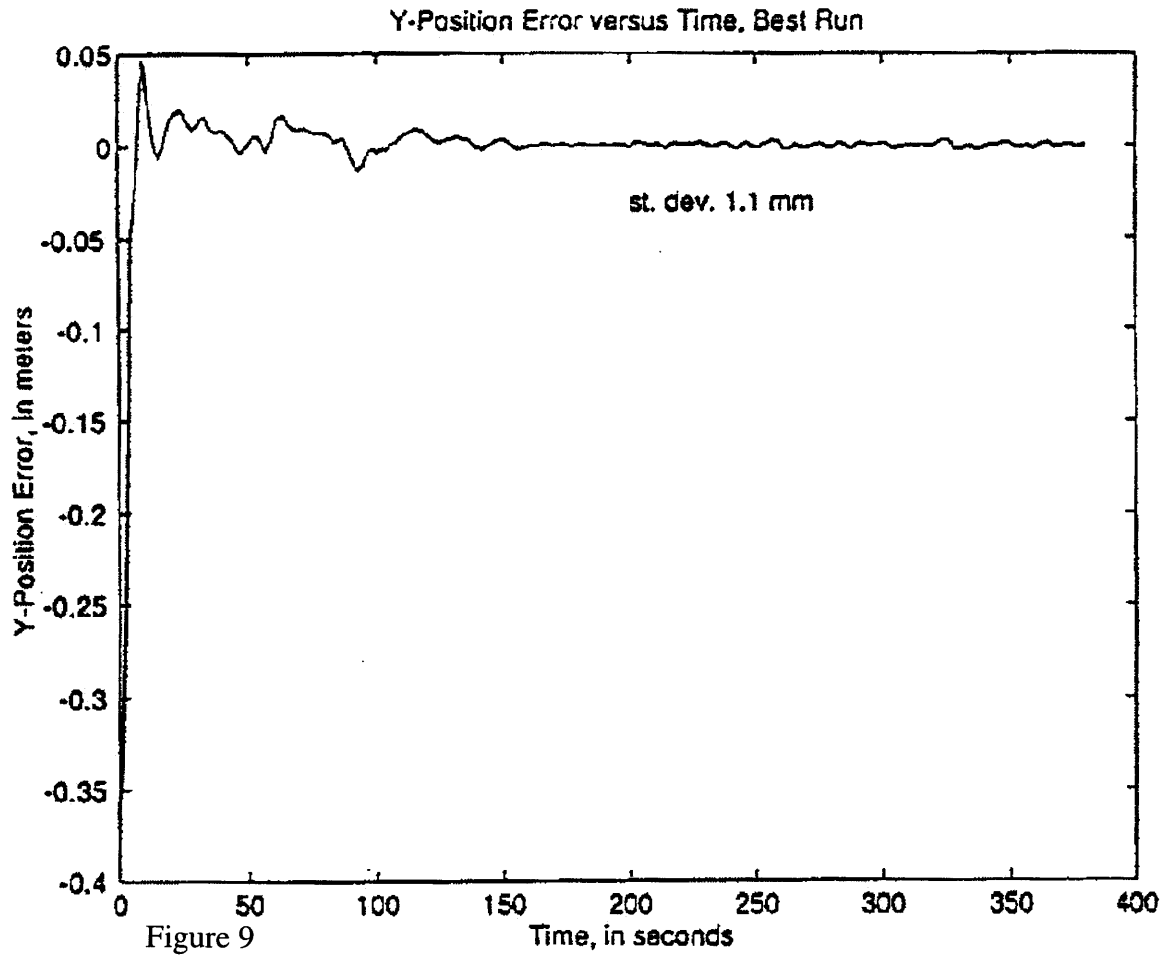


Figure 8



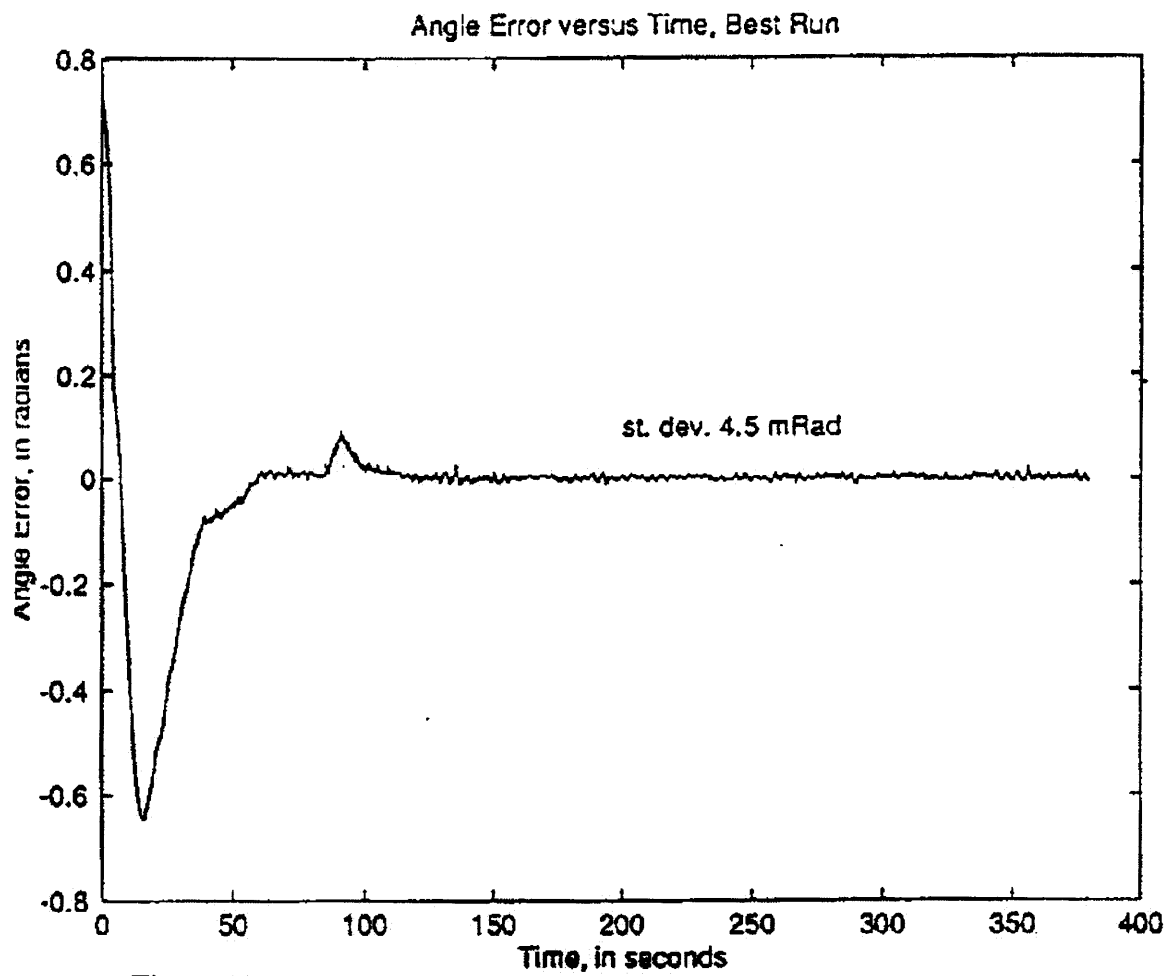


Figure 10

## 6. Conclusions

The experiment seems to have been successful, despite the missing angle lock. All the equipment was fully functional, as verified by the near lock that was achieved. The laser metrology apparatus was tested simply by turning off the air table's vacuum pump and placing the puck at the desired spot. Fringes were detected and counted, as expected in the actual experiment. The locks in position seemed perfectly adequate. The only problem, in fact, seems to have been the lack of a lock in angle. This is not to say that the angle performed extremely poorly. It simply was not good enough for the experiment.

Several factors could account for the lack of a lock in angle. Primarily, the blame is placed on a lack of angular resolution, a direct consequence of the camera resolution, the star placement, and the separation distance between the puck's lock position and the stationary corner cube. To explain, the camera resolution was found to be about 3mm on the ceiling per pixel in x and y. The stars themselves were separated by about 2 feet, or roughly 600 mm. The smallest angle that can be detected is represented by a one pixel change in position, so this is about  $\arctan(3/600)$ , which is roughly 5 mRad once again. Therefore the angle cannot be distinguished much better than this amount, but since we require a bit better than this for our experiment, it cannot work. To make it work, the stars must be spread out more, or the camera must have improved resolution, or the puck must get closer to the stationary corner cube. Unfortunately, when this problem in the system was discovered, it was too late to make the necessary changes. Nevertheless, the successful locks in the rectangular coordinates verify that the control algorithms do work, which is really what the experiment is about.

Other significant problems in the system include signal noise and real-world fan dynamics. The experimental apparatus in the laboratory was, due to space concerns for other experiments, tightly collected in one corner of a large lab table. This meant all the communications signals in the experiment were transmitting physically near to each other. Also, the experimental apparatus consisted of a large quantity of metal, including the VME card cage, the puck itself, the air table's aluminum sides, the lab table's metal surface, and the various special boxes that were constructed for the experiment. This means that a lot of the radiated energy reflected numerous times in the vicinity of the transmitters and receivers. This has the potential to create a lot of noise, and indeed the noise level was noticeable in almost all the communication paths, including the video. Thus the noise always played a real factor in the experiment. To compensate, low-pass filtering was performed on the star position and a slow decay rate towards the correct spot was used (yielding a lower bandwidth and less noise sensitivity). Nevertheless, if the noise could actually be lessened, through better equipment, greater separation between the transmitters, less metal in the area, etc., a more stable performance could be expected.

In the end, the fans worked but were somewhat unreliable. For one, the fan characteristics were determined to actually not follow the propeller's voltage-square law force behavior, but produced a quasi-linear force for a linear voltage instead. This was determined by tethering the air puck to a spring and measuring the spring's displacement when the fan (and its complement on the other side of the puck) blew at different DC values. By Hooke's Law, this displacement should be proportional to the applied force. The results of these experiments show that the fan force is more linear than anything else with respect to applied DC voltage, but this linearity is certainly rough (see Figure 11). The control algo-

rithm was rewritten for such a linear relationship as a result of these experiments, but again that linearity is questionable. This also meant the principle of superposition that allowed the same fans to be used for both translation and rotation was just as questionable. Nevertheless, clearly it was no total disaster either. A second problem with the fans is that they had a significant start-up time when ordered to change DC voltages. For instance, from a complete stop, the fans took 4 seconds to reach their maximum force. Although the fans were always kept on to minimize this, they still could take several clock interrupt cycles to reach a desired force. To further minimize this, the time it took the control algorithm to reach the lock position was made long, so that movements were slow and the slow fan transition times had a smaller effect. Nevertheless, a better propulsion system could have produced smoother results, reducing noise and possibly achieving a better lock in position and angle.

As a general principle, better equipment could also have improved system performance, but resources were limited. Still, with the above improvements, the lock in angle might have been achieved and might have held well.

The experiment shows that control of a spacecraft can be achieved in a two-dimensional frictionless environment. It relies heavily on the capabilities of the equipment, however, and a space quality system, like the New Millennium Interferometer, would require much more from its propulsion, communication, and video systems than were available on the puck. In fact, just about every component of the puck would need to be of a much better quality, with far better performance capabilities, than was available for this experiment. Nevertheless, the theory does seem to work, and at least in principle, this technology can fly.

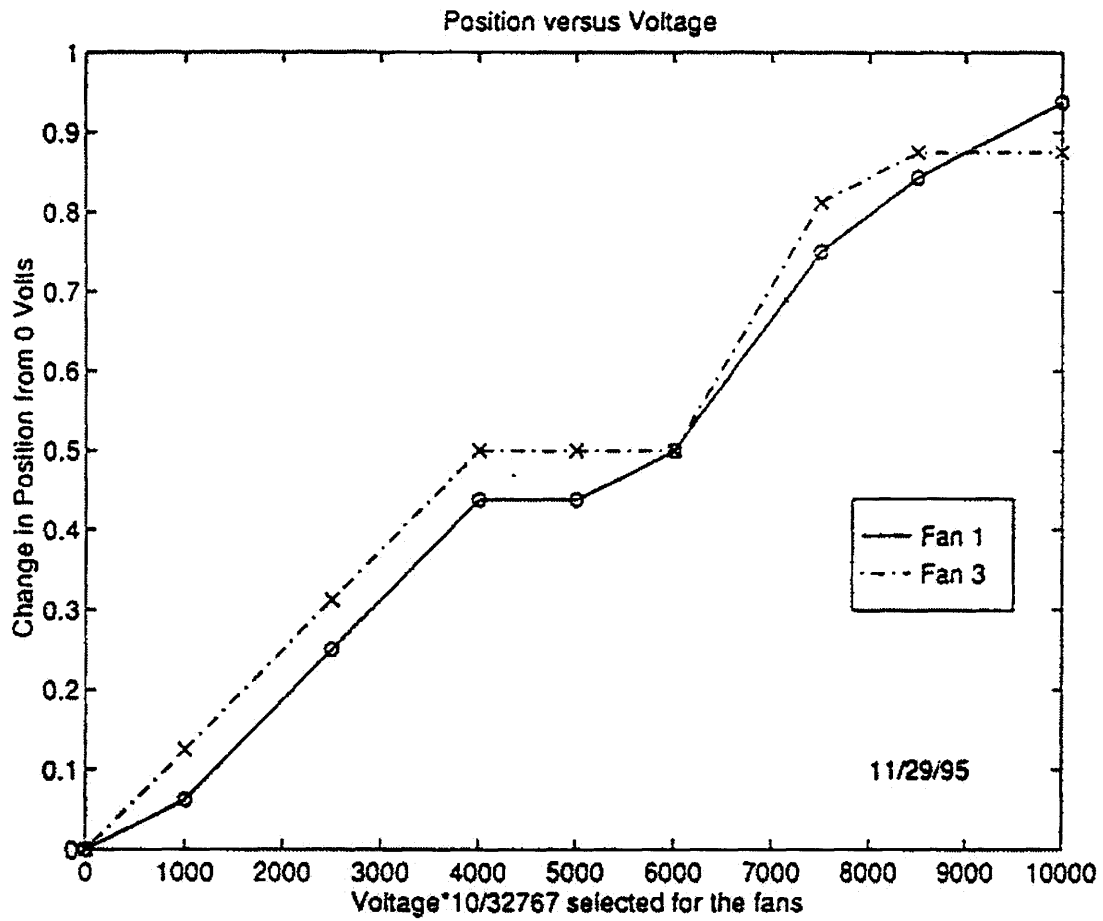


Figure 11

## **Bibliography**

- 1) Glauert, H. "Airplane Propellers", Aerodynamic Review, A General Review of Progress, Volume IV, Section L, pp. 169-360, Dover Publications, Inc., New York. 1963.
  
- 2) Kulkarni, S.R. "Separated Spacecraft Interferometry". JPL Internal Memo. 1994.
  
- 3) Colavita, M. & McGuire J. "New Millennium Interferometer: Preliminary Design". JPL Internal Memo. 1995.



## Appendix

### Propeller Theory:

Each fan produces a flow of mass moving through the blades at a fixed velocity. This gives a momentum of  $\Delta p = (\Delta m)v$ , where  $m$  is the constant mass of the air and  $v$  is the velocity of the fans. The fan speed was found empirically to increase linearly with the fan's input voltage, so  $\Delta p = (\Delta m)GV$ , where  $G$  is a proportionality constant between voltage at the fan and angular velocity of the fans. Also, as time goes by, the displaced mass is  $\Delta m = M\Delta t$ , where  $M$  is another fixed constant, depending on the geometry of the fan, the density of the air, etc. Taking the limit as the time interval decreases leaves:

$$F = \frac{dp}{dt} = \frac{dm}{dt}GV = MGV$$

(The component to the derivative of the momentum coming from time changes in the voltage were ignored because voltage transitions will be very quick, and then the voltage will be left fixed for the duration of each chopping period.) So the force from each fan is directly proportional to the voltage applied to that fan.

Each fan produces a flow of mass moving through the blades at a fixed velocity. If a fan rotating at constant angular velocity  $\omega$  is thought of as subtending a certain angle  $\Theta_0$  in the plane of rotation and having a height  $d$  for each fixed radius  $r$  from the center of rotation (so  $d$  and  $\Theta_0$  are actually written  $d(r)$  and  $\Theta_0(r)$ ), then particles travelling at one end of the fan must reach the other end of the fan (a total distance  $d(r)$ ) in a time  $\Theta_0(r)/\omega$ . This means the velocity of particles pushed by the fans equals  $\omega d(r)/\Theta_0(r)$ . Incidentally,

the function describing a fan's geometry is presumed to be monotonic in its  $\phi(\theta)$  component, so that no air is ever caught in the fan. In a time  $t$ , the fans blow a mass through its blades equaling the air density  $\rho$  times

$$\int_0^{|\omega|t} \int_0^{R_o} d(r)rdrd\Theta = |\omega|t \int_0^{R_o} d(r)rdr$$

a term representing the component of the cross-sectional area of the fan blade perpendicular to the plane of rotation ( $R_o$  represents the maximum radius of the fan) times the angle that has been subtended through rotation from time 0 to time  $t$ . This angle is simply equal to  $\omega t$ . The absolute values are included because  $\omega$  can be reversed in sign, but the volume swept out should not change (presuming equal magnitudes for  $\omega$ ). So the volume swept out in time  $t$  is some constant  $V_o$  (based solely on the geometry of the fan, constant in time) times  $|\omega|t$ . The momentum of a fan blade is thus the velocity at each point multiplied by the volume at each point, or

$$\int_0^{|\omega|t} \int_0^{R_o} \rho \frac{\omega d(r)}{\theta_o} d(r)rdrd\theta = \rho |\omega| \frac{\omega t}{\theta_o} \int_0^{R_o} [d(r)]^2 r dr = \rho H |\omega| \omega t$$

where  $H$  is simply the collection of the constant (in time) terms that depend only on the geometry of the fan. The force is the time derivative of the momentum, so the force on each fan blade is

$$F = \rho H |\omega| \omega$$

Experimentally, it was determined that the angular velocity of each fan varies very linearly with voltage, so finally

$$F = G|V|V$$

where the constant  $G$  is proportional to the density  $\rho$ , the geometry constant  $H$ , and the proportionality factor between voltage and angular momentum.

VxWorks Software:

```
#include <stdio.h>
#include <math.h>
#include <vxWorks.h>
#include <vme.h>
#include <sysLib.h>

#ifdef VW_502
#include <68k/iv.h>
#include <vxProto.h>
#else
#include <iv.h>
#endif /* VW_502 */

#include "/home/hafnir/startrack/stsimdyn2.h"
#include "/home/yekta/MetrologySoftware/MatroxLib/vip640aLib.h"
#include "/home/yekta/MetrologySoftware/MatroxLib/vip640aDumpLib.h"
#include "/home/yekta/MetrologySoftware/dt1401Lib/dt1401Lib.h"
#include "/home/yekta/MetrologySoftware/dvme626Lib/dvme626Lib.h"
#include "/home/yekta/MetrologySoftware/slowlasPcbLib/slowlasPcbLib.h"
#include "/home/hafnir/startrack/myrecLibB.h"
/* #include "/home/palomar/drivers/slowlasPcbLib.h" */

#define ROUND(a) (int)((a)<0. ? ((a)-.5) : ((a)+.5))
#define ABS(a) ((a)<0 ? -(a) : (a))
#define MAX(a,b) ((a)>(b) ? (a) : (b))
#define MIN(a,b) ((a)<(b) ? (a) : (b))
#define BOUND(a,b,c) (((c) > (a) ? ((a) > (b) ? (a) : ((b) > (c) ? (c) : (b))) : ((c) > (b) ? (c) : ((b) > (a) ? (a) : (b))))

/* Big assumptions: 1) Moves slow. Recommend about 10 degrees per second.
2) Star small. Recommend about 10 pixel diameter.
3) Stars always in the middle of the screen. Recommend
at least 50 pixels from any end of the screen.
4) Bottom left hand corner is point 0,0. */

int state = 0;

unsigned char image[640*480];
unsigned char simage[52*52];
unsigned char thresh = 130, thresho;
```

```

int a=0,b=0,c=0,d=0,sn=0,ns=0,tmp1,tmp2,lockmin=1000,tim=0,mytaskID,arcpbr,jim;
int pheight=480,pwidth=640,tau,tcpt,rad=12,rad1=14,tr,tr1,trs,f1cc=70,f2cc=28;
int f1t[2], f2t[2], g[2],f1s[2],f2s[2],phasedata=0,lock=0;
double f1[2],f2[2];
double mperpixx = .003, mperpixy = .003;
double thact,aveNS,aveEW,alpha,beta=0,dalpha,cao,sao,cap,sap,tmp1f,tmp1f;
double x[3], y[3], argum[3];
/* double xo=435.931335, yo=-32.173363, thdes = -2.454324; */
double xo=10.960155, yo=39.951531, thdes = -2.319941;

/* double m = 3, I = .01, G= 0.001, GArg = 0.0003, aa=0.04, bb=0.25, cc=0.5; */

double m = 3, I = 0.005, aa=0.04, bb=0.25, cc=0.5;
double aax = 0.04, bbx = 0.1, ccx = 1.5, Gx = 0.002;
double aay = 0.04, bby = 0.1, ccy = 2, Gy = 0.002;
double aaA = 0.04, bbA = 0.1, ccA = 10, GArg = 0.0003;

double dt= 0.117, convers=(32767/17),NSt1,NSt2,NSt3,EWt1,EWt2,EWt3,At1,At2,At3;
double vNS=0,vEW=0,vArg=0,vArgSent=0,vNS1=0,vNS2=0,vEW1=0,vEW2=0;
double kNS,kEW,kArg, lNS, lEW, lArg, pNS, pEW, pArg;

STATUS goforit(void)
{
    remove("MetroData");
    vip640aContGrabOff();
    vip640aUnZoom();
    vip640aSelectBuf(0);
    dvme626FullInit(TRUE);
    dt1401FullInit();
    dt1401SelSoftTrigger();
    dt1401Set(1,0,0);
    slowlasFullInit();
    slowlasClearCounters();

    recOpen("MetroData");
    printf("Initialized.\n");

    vNS=0,vEW=0,vArg=0,vArgSent=0,vNS1=0,vNS2=0,vEW1=0,vEW2=0,state=0;
    f1[0]=0, f2[0]=0, f1[1]=0, f2[1]=0;
    f1s[0]=0, f2s[0]=0, f1s[1]=0, f2s[1]=0;
    tr=2*rad, tr1=2*rad1, trs=tr+tr1, thresho = thresh/3;
/* jim=-1; */
    gains();
    fullstop();
    puckcontrol();
}

```

```

printf("Back from puckcontrol.\n");

if(state == 0)
{
    state = 1;
    mytaskID=taskSpawn("taskpuckcontrol",10,VX_FP_TASK,8192,(FUNCPTR)
puckcontrol,0,0,0,0,0,0,0,0,0,0);
    if(sysAuxClkConnect((FUNCPTR) puckresume,0) == ERROR) return(ERROR);
    sysAuxClkEnable();
    printf("Puckcontrol spawned and engaged.\n");
}
return(OK);
}

void puckresume(void)
{
    tim++;
    if(tim==7)
    {
        taskResume(mytaskID);
        tim=1;
    }
}

void puckcontrol(void)
{
    do
    {
/*        tickSet(0);
        beta=tickGet(); */

/* State = 1 is the normal loop, while state = 0 is the first/set-up loop. */

        if (state == 1)
        {
            taskSuspend(0);
            vip640aGrab();
            while(vip640aWaitGrab());

/* Average over the points. The algorithm here is to know the number of stars
that are out there, to find a point with intensity greater than the threshold,
take all the points around the first point within a certain neighborhood and
find their weighted average. This weighted average is the location of the star.
*/

            for(sn=0,ns=0; sn<2; sn++)

```

```

    {
/*   tmp1 = ROUND(f1[sn])-rad1-rad;
    tmp2 = ROUND(f2[sn])-rad1-rad; */
    tmp1 = f1s[sn]-rad1-rad;
    tmp2 = f2s[sn]-rad1-rad;
    vip640aDumpROI(0,tmp2,480-tmp1-trs,trs,trs,(UWORD *) simage);
    setROISize(trs,trs);
    setROIStart(tmp2,480-tmp1-trs);
/*   if(sn==0 && jim==0) dumpROI(0,"jim0pic");
    if(sn==0 && jim==1) dumpROI(0,"jim1pic");
    if(sn==0 && jim==2) dumpROI(0,"jim2pic");
    if(jim==4) state=2; */
    for(a=0; a<tr1; a += 2)
    {
        for(b=0; b<tr1; b += 2)
        {
            if(simage[(rad+a)*trs+rad+b] > thresh)
            {
                for(c=0,f1t[sn]=0, f2t[sn]=0, g[sn]=0; c<tr; c++)
                {
                    tcpt = (a+c)*trs+b;
                    for(d=0,tau=0; d<tr; d++)
                    {
                        if(simage[tcpt+d]>thresho)
                        {
                            f2t[sn] += ((int) simage[tcpt+d])*d;
                            tau += ((int) simage[tcpt+d]);
                        }
                    }
                    if(tau>0)
                    {
                        f1t[sn] += tau*c;
                        g[sn] += tau;
                    }
                }
                f1[sn] = ((double) (tmp1+a))+((double) f1t[sn])/((double) g[sn]); /* y-
position */
                f2[sn] = ((double) (tmp2+b))+((double) f2t[sn])/((double) g[sn]); /* x-
position */
                recFloat((float) a);
                recFloat((float) b);
                f1s[sn] += a-rad1;
                f2s[sn] += b-rad1;
                a=tr1, b=tr1;
                ns++;
            }
        }
    }

```

```

    }
  }
}
if(ns != 2)
{
printf("Error. Did not find two stars. Found %d stars.\n",ns);
state = 2;
}
}

if (state == 0)
{

```

/\* Average over the points. The algorithm here is to know the number of stars that are out there, to find a point with intensity greater than the threshold, take all the points around the first point within a certain neighborhood and find their weighted average. This weighted average is the location of the star.  
\*/

```

    vip640aGrab();
    while(vip640aWaitGrab());
    vip640aDumpPage(0,(UWORD *) image);
    for(a=0,sn=0;a<pheight;a+=2)
    {
for(b=0;b<pwidth;b+=2)
{
    if(image[a*pwidth+b] > thresh)
    {
if(sn > 1) sn=1, state=2;
for(c=0,f1t[sn]=0,f2t[sn]=0,g[sn]=0;c<tr;c++)
{
    arcpbr=(a-rad+c)*pwidth+b-rad;
    for(d=0,tau=0;d<tr;d++)
    {
if(image[arcpbr+d]>thresho)
{
    f2t[sn] += ((int) image[arcpbr+d])*d;
    tau += ((int) image[arcpbr+d]);
    image[arcpbr+d] = (unsigned char) 0;
}
}
    if(tau>0)
    {
f1t[sn] += tau*c;
g[sn] += tau;

```



```

    }
}
    f1[sn]= ((double) (a-rad))+((double) f1t[sn])/((double) g[sn]); /* y-position
*/
    f2[sn]= ((double) (b-rad))+((double) f2t[sn])/((double) g[sn]); /* x-position
*/
    f1s[sn] = a;
    f2s[sn] = b;
    sn++;
    recFloat((float) a);
    recFloat((float) b);
    printf("a = %d, b = %d\n",a,b);
    }
}
}

```

```

if (state == 2) printf("Error. Found more than 2 stars.\n");

```

```

    if(sn<2)
    {
state=2;
printf("Only found %d stars.\n",sn);
    }

```

/\* At this point, the stars should be distinguished, and f1&f2 should be sorted so that star 0 is in f1[0]&f2[0], etc. \*/

```

    if(g[1]<g[0])
    {
tmp1f = f1[0];
f1[0] = f1[1];
f1[1] = tmp1f;
tmp1f = f2[0];
f2[0] = f2[1];
f2[1] = tmp1f;
tmp1 = f1s[0];
f1s[0] = f1s[1];
f1s[1] = tmp1;
tmp1 = f2s[0];
f2s[0] = f2s[1];
f2s[1] = tmp1;
tmp1f = g[1];
g[1] = g[2];
g[2] = tmp1f;
printf("Switch\n"); */

```

```

    }
}

for(a=0; a<2; a++)
{
    if(f1[a]-rad-rad1<0)
    {
        printf("f1[%d] out of bounds.\n",a);
        f1[a]=rad+rad1;
    }
    if(f1[a]+rad+rad1>pheight)
    {
        printf("f1[%d] out of bounds.\n",a);
        f1[a]=pheight-rad-rad1;
    }
    if(f2[a]-rad-rad1<0)
    {
        printf("f2[%d] out of bounds.\n",a);
        f2[a]=rad+rad1;
    }
    if(f2[a]+rad+rad1>pwidth)
    {
        printf("f2[%d] out of bounds.\n",a);
        f2[a]=pwidth-rad-rad1;
    }
}

```

```

/* Determine the angle and position of the stars in absolute coordinates */
/* Set the initial values as well.                                     */

```

```

if(f2[1]!=f2[0]) thact = atan((f1[1]-f1[0])/(f2[1]-f2[0]));
else if(f1[1]>f1[0]) argum[0] = 3.1415925/2;
else if(f1[1]<f1[0]) argum[0] = -3.1415925/2;
else
{
    state = 2;
    printf("f1 and f2 are the same coordinate.\n");
}

if(f2[0] > f2[1])
{
    if(thact>0) thact -= 3.1415927;
    else thact += 3.1415927;
}

```

```

    argum[0] = thact-thdes;
    if(argum[0] > 3.1415927) argum[0] -= 2*3.1415927;
    if(argum[0] < -3.1415927) argum[0] += 2*3.1415927;

    aveNS = 0.5*(f1[0]+f1[1])-f1cc-240;
    aveEW = 0.5*(f2[0]+f2[1])+f2cc-320;
    cap = cos(argum[0]-3.1415927/4);
    sap = sin(argum[0]-3.1415927/4);
    x[0]= -cap*aveEW-sap*aveNS;
    y[0]= -sap*aveEW+cap*aveNS;
    x[0] = (x[0]-xo)*mperpixx;
    y[0] = (y[0]-yo)*mperpixy;

    if(state==0)
    {
        argum[1] = argum[0];
        argum[2] = argum[0];
        x[1]=x[0];
        x[2]=x[0];
        y[1]=y[0];
        y[2]=y[0];
    }

/* Do the actual translation based on the position and angle data */

    if(state != 2) CoarseTransandRot(x,y,argum);

/* Swap out the current coordinates into the past coordinates */

    x[2] = x[1];
    y[2] = y[1];
    argum[2] = argum[1];
    x[1] = x[0];
    y[1] = y[0];
    argum[1] = argum[0];

/* Check whether we can digitize fringes now, and if so then do so. */
/* Save the output to a file. */

    dt1401ADStart();
    while(!dt1401ADDone());
    dt1401AD(0,&lock);
    if(lock>lockmin)
    {

```

```

        slowlasFreeze();
        slowlasReadCounter(0,&phasedata);
        slowlasUnfreeze();
        recInt(phasedata);
    }
/*    alpha=tickGet();
    dalpha = (double) (alpha-beta)/60;
    printf("Elapsed time for the frame grab:%f\r",dalpha);
    recFloat((float) dalpha); */
/*    jim++; */
}
while(state==1);
if(state==2)
{
    sysAuxClkDisable();
    fullstop();
    recClose();
    dumpPage(0,"problempic");
    printf("Error. Giving up now.\n");
}
}

```

```
void CoarseTransandRot(x,y,argum)
```

```
double x[3],y[3],argum[3];
```

```
{
    gains();
    vNS += NSt1*y[0]-NSt2*y[1]+NSt3*y[2];
    vEW += EWt1*x[0]-EWt2*x[1]+EWt3*x[2];
    vArg += At1*argum[0]-At2*argum[1]+At3*argum[2];

```

```

    cao = cos(argum[0]);
    sao = sin(argum[0]);
    vEW2 = BOUND(-15,cao*vEW/2+sao*vNS/2,15);
    vEW1 = -vEW2;
    vNS1 = BOUND(-15,-sao*vEW/2+cao*vNS/2,15);
    vNS2 = vNS1;

```

```

    vEW=2*(cao*vEW2-sao*vNS1);
    vNS=2*(sao*vEW2+cao*vNS1);

```

```
printf("At1=%f,x[0]=%f,y[0]=%f, argum[0]=%f\r",At1,x[0],y[0],argum[0]);
```

```

recFloat((float) f1[0]);
recFloat((float) f2[0]);
recFloat((float) f1[1]);
recFloat((float) f2[1]);

```

```

recFloat((float) vNS1);
recFloat((float) vNS2);
recFloat((float) vEW1);
recFloat((float) vEW2);
recFloat((float) vNS);
recFloat((float) vEW);
recFloat((float) x[0]);
recFloat((float) y[0]);
recFloat((float) vArg);

vArgSent= 4*MIN(MIN((vNS1+17),(vEW1+17)),MIN((17-vNS2),(17+vEW2)));
if(vArg < 0) vArgSent = -vArgSent;
if(ABS(vArgSent) > ABS(vArg)) vArgSent = vArg;
else vArg = vArgSent;
vNS1 -= vArgSent/4;
vEW1 -= vArgSent/4;
vNS2 += vArgSent/4;
vEW2 -= vArgSent/4;

/* printf("vArg= %f, vArgSent= %f, argum[0]= %f\r",vArg,vArgSent,argum[0]); */

recFloat((float) vArgSent);
recFloat((float) argum[0]);

dvme626DA(0,ROUND(BOUND(-17,vNS1,17)*convers));
dvme626DA(1,ROUND(BOUND(-17,vEW1,17)*convers));
dvme626DA(2,ROUND(BOUND(-17,vNS2,17)*convers));
dvme626DA(3,ROUND(BOUND(-17,vEW2,17)*convers));
}

void fullstop()
{
dvme626DA(0,0);
dvme626DA(1,0);
dvme626DA(2,0);
dvme626DA(3,0);
}

void as()
{
sysAuxClkDisable();
fullstop();
recClose();
dumpPage(0,"lastpic");
}

```

```

void gains()
{
  kNS= -m/Gy*(aay*bby*ccy);
  lNS= -m/Gy*(aay*bby+bby*ccy+aay*ccy);
  pNS= -m/Gy*(aay+bby+ccy);
  kEW= -m/Gx*(aax*bbx*ccx);
  lEW= -m/Gx*(aax*bbx+bbx*ccx+aax*ccx);
  pEW= -m/Gx*(aax+bbx+ccx);
  kArg= -I/GArg*aaA*bbA*ccA;
  lArg= -I/GArg*(aaA*bbA+bbA*ccA+aaA*ccA);
  pArg= -I/GArg*(aaA+bbA+ccA);
  NSt1 = kNS*dt+lNS+pNS/dt;
  NSt2 = lNS+2*pNS/dt;
  NSt3 = pNS/dt;
  EWt1 = kEW*dt+lEW+pEW/dt;
  EWt2 = lEW+2*pEW/dt;
  EWt3 = pEW/dt;
  At1 = kArg*dt+lArg+pArg/dt;
  At2 = lArg+2*pArg/dt;
  At3 = pArg/dt;
}

```