Development of a Multi-dimensional Optimization Servo for Gravity Wave Detector Mirror Alignment

by Brian Elliot Lemoff

Physics 78 Project

Ronald W.P. Drever, Advisor

LIGO-T890067-00-D

I. Introduction

While most of my work this year can be applied to just about any system requiring low frequency stabilization, I shall begin with a brief discussion of the Caltech gravity wave detector. In this way, I hope to convey the original motivation for the project, as well as its possible applicability to physics research at Caltech.

The Caltech gravity wave detector consists primarily of two perpendicular forty meter optical cavities which, when all is working properly, are in resonance with a single argon ion laser. Each cavity contains two opposing mirrors, one flat and one curved, each mounted to a suspended test mass. One of the cavities is used to stabilize the laser. The other cavity is servoed to remain in resonance with the light which has been stabilized to match the first cavity. Thus, any relative change in length between the two cavities will come through as a change in the servo voltage applied to the second cavity. Because a gravity wave will cause the relative lengths of the arms to vary, the second arm servo voltage is the signal used to search for gravity waves.

Within the last two years, a third, smaller cavity has been inserted between the laser and the two main cavities. This one meter cavity, known as the *mode cleaner*, serves as an initial stage in the laser stabilization, providing a more stable light source for the two main cavities, as well as filtering out optical modes other than TEM_{00}. The mode cleaner consists of two curved mirrors which, unlike their counterparts in the main cavity, are mounted directly to fixed mirror mounts which are not isolated from the ground.

All three of the cavities have extremely high finnesse, i.e. if the laser were to suddenly be shut off, the light remaining in the cavity would make a large number of bounces before the intensity decreased to $(1/e)$ of its original value. The main cavities have finesses of order 10^4, and the mode cleaner has a finnesse of roughly 10^3. To attain such a sharp resonance, the mirrors in the cavities must be alligned to very high precision. In addition, the waist size and divergence angle of the beam must be adjusted to match the curvature and separation of the mirrors. This procedure, known as mode matching, can be accomplished with two lenses.

The parameter most often measured to determine how well the mirrors and mode matching have been alligned is called the *contrast* of the cavity. Ideally, in a perfectly resonant optical cavity, the light leaking back out of the input end of the cavity will interfere destructively with the light reflected back off of the input mirror. Thus, all of the light will be going into the cavity, and none will be reflected. In practice, however, this is impossible to acheive. If we let I_N be the intensity of the back reflected light when the cavity is not resonanting, and we let I_R be the back reflected light when the cavity is in resonance, then the contrast C is given by:

$$C = \frac{I_N - I_R}{I_N}. \tag{1}$$

Thus, a contrast of 100% would correspond to an ideally resonant cavity, and a contrast of 0% would correspond to a cavity which is not resonating.

In each of the three cavities of the gravity wave detector, the mirror alignment and mode matching is adjusted by hand to that combination of positions which maximizes the contrast. Once that has been done, the mode matching lenses, which are subject to only slow thermal drift, are left unattended, being manually adjusted whenever necessary. In the main cavities, mirror alignment is preserved by means of two servo systems, identical in nature, but different in sensitivity. Optics are adjusted so that when the mirrors are properly alligned, a helium-neon pointing laser, which reflects off of the mirror in question, will be centered on a position sensitive quadrant photodiode. A correction signal is derived from the displacement of the spot from the center of the phiotodiode and is applied to magnetic coils which control the tilt and rotation of the test masses. In this way, the test masses are held stable, and the mirror alignment is preserved. In addition to preserving mirror alignment, this servo attenuates frequencies up to several kilohertz, eliminating noise that could possibly reduce the sensitivity of the detector.

In the mode cleaner, no such alignment servo is present. Because the cavity is only one meter in length, the contrast is not nearly as sensitive to small drifts in mirror alignment as in the forty meter cavities. However, over a period of fifteen to twenty minutes, thermal effects typically cause the contrast to be degraded, and readjustment is necessary. Because the cavity mirrors are fixed, alignment of the cavity is acheived by adjusting the angle and position of the input beam. To first order, this is equivalent to adjusting the angles of the mirrors. Thus, there are four degrees of freedom to be adjusted: vertical angle, horizontal angle, height, and lateral translation of the input beam. It has been proposed to replace the one meter mode cleaning cavity with a more sensitive, ten meter mode cleaning cavity. Work to this effect is already in progress.

Because of its length, the contrast in the ten meter mode cleaner will be much more sensitive to the thermal drift of the mirrors. Thus, an alignment servo will be necessary. Because the cavity mirrors will be fixed, this servo must stabilize the four input beam parameters mentioned above. Whereas the alignment servos in the main cavities work by centering a pointing laser on a photodiode whose center has been adjusted to correspond to maximum contrast, this servo, as it was proposed, would use the contrast itself as the only signal from which to derive the four correction signals. Such a system would only be intended to compensate for low frequency ($\sim$ 10 minute period) thermal drifts, and not seismic noise and other higher frequency, lower amplitude noise sources.If such a system could be made to work, then it could, in theory, be expanded to include the two mode matching degrees of freedom as well.

Because this servo system would ultimately involve six degrees of freedom, it was decided that electronics could be minimized if a computer, equipped with DAC's and ADC's, could be substituted for the six waveform generators, lock-in amplifiers, and other electronics necessary for the system to work. I decided that the simplest way to actuate the mirror adjustment would be by using four separate mirrors, each connected to a galvanometer motor (the type of motor used to drive pens in chart recorders). I had used galvo motors as mirror drivers in a previous project, so I was already quite familiar with their performance. Because of time constraints, the incorporation of the mode matching degrees of freedom was left for the future. The details of the theory behind the servo, both one dimensional and multi- dimensional, along with a detailed description of my setup, both software and hardware, will comprise the bulk of this paper. It should be realized throughout, that a servo such as this can, with an appropriate input signal and set of actuators, be used in almost any situation in which low frequency stabilization of some system degree of freedom is desired.

II. The Idea

We will begin with the fundamental idea behind the type of maximization servo that I have been working on. Let us begin in one dimension. Suppose we have a system with a single input parameter θ, and a single output parameter, I. Furthermore, suppose there is a unique value of θ, call it Θ, where I is a maximum. The details of the system determine the functional dependence of of I on θ, but we can always find a value R, such that:

$$\frac{dI}{d\theta} \propto (\theta - \Theta) \qquad (2)$$

in the region

$$|\theta - \Theta| \leq R. \qquad (3)$$

This simply says that within range R of the optimum input parameter Θ, the output parameter $I(\theta)$ can be approximated by a parabola. From now on, unless otherwise mentioned, we will assume that we are within this range.

The task of the servo system is to maintain θ as close as possible to Θ. To accomplish this, the servo must determine the error $\theta - \Theta$, and output a correction signal, C, to compensate for it. The only input signal available to the servo, from which all of its information must be derived is the output of the system I. Hopefully, the reader will not be confused by the fact that, in the context of the servo, we will call I the input signal, while in the context of the system, we call I the output signal. Clearly, because the parabola is symmetric about $\theta = \Theta$, the value of I will not be sufficient to determine a correction signal. What we really must measure is the derivative of I with respect to θ. We can see by equation (2) that this will be proportional to the error $E = \theta - \Theta$. Thus, we need a way to measure this derivative, given only the input signal I.

Clearly, no information can be obtained on $dI/d\theta$ without varying θ and seeing how I changes. Thus, we will introduce a *dither*, that is, a small modulation applied to θ. We will have:

$$\theta(t) = \theta_0(t) + D \sin \omega_d t, \qquad (4)$$

where $\theta_0(t)$ encompasses all of the time dependence of θ not resulting from the dither. In order for the servo to work properly, θ_0 must be approximately constant over the period of ω_d. This translates into the requirement that:

$$\tilde{\theta}_0(\omega) \ll D, \quad \omega \geq \omega_0, \tag{5}$$

where $\tilde{\theta}_0(\omega)$ is the Fourier transform of $\theta_0(t)$, and $\omega_0 \ll \omega_d$ is the high frequency cutoff of our servo. We assume that D is sufficiently small such that:

$$I(\theta_0 + \alpha) \approx I(\theta_0) + \alpha \left.\frac{dI}{d\theta}\right|_{\theta_0}, \quad |\alpha| \leq D. \tag{6}$$

Thus, making use of equation (5) to treat θ_0 as a constant, gives us:

$$I(t) = I(\theta_0) + D \sin \omega_d t \left.\frac{dI}{d\theta}\right|_{\theta_0}. \tag{7}$$

Thus, introducing a dither into the system causes information about $dI/d\theta$ to be contained in the output signal $I(t)$. All we need to do to extract this information is to find the value of the Fourier transform $\tilde{I}(\omega)$ at $\omega = \omega_d$.

One way to accomplish this task is to employ a *lock-in amplifier.* A lock in amplifier uses as its inputs the modulation signal, $D \sin \omega_d t$, and the system output, $I(t)$. First, the device converts the modulation signal into a square wave:

$$S(t) = A\,\mathrm{sgn}[\sin \omega_d t], \tag{8}$$

where the function $\mathrm{sgn}(x)$ returns $\pm$ 1, depending on the numerical sign of x, and A is some constant. The product $S(t)I(t)$ will have a DC component proportional to $\tilde{I}(\omega_d)$, and hence to E. If we consider $E(t)$ to be a slowly changing function of time (slow compared to ω_d), then the Fourier transform of $S(t)I(t)$ will be proportional to that of $E(t)$ for $\omega \ll \omega_d$. Thus, the lock-in amplifier ouput is the result of passing $S(t)I(t)$ through a low pass filter that rolls off at $\omega = \omega_0 \ll \omega_d$. If we consider only the simplest low pass filter, then the output of the lock-in amplifier, V(t) is given by:

$$V(t) = N \int_0^t e^{\omega_0 (t' - t)} S(t') I(t')\, dt'. \tag{9}$$

It should be clear that equation (9) acts like a low pass filter. If it is not clear, then think of it this way. To a frequency $\omega \gg \omega_0$, the exponential looks like a constant, and hence the function is integrated. To a low frequency, $\omega \ll \omega_0$, the exponential looks like a delta function, and hence it is passed unaffected. Determining the correction signal $C(t)$ as a function of $E(t)$ is a complicated task which depends on the specific physics of the system. If no physical resonances, phase shifts, or other frequency dependent effects are present, then the simplest type of feedback is that used in an inverting op amp. That, is, we take $C(t) = -gE(t)$, where g is the frequency independent gain of the amplifier. In our situation, equation (5) puts a restriction on the allowed frequency response of the servo. This restriction, however, is already taken into account in equation (9). Thus, a reasonable choice for our correction signal is:

$$C(t) = -g \int_0^t e^{\omega_0 (t'-t)} S(t') I(t') \, dt' \,. \tag{10}$$

Equation (10) is the best method that I can think of to convert the error signal to a correction signal. As it happens, it is not quite sufficient to stabilize my servo at very high loop gain; however, it does work reasonably well for moderate loop gain.

III. Multiple degrees of freedom

Now, let us consider a servo with multiple degrees of freedom, Let us call them $\theta_1, \theta_2, \ldots \theta_n$. Let us also assume that there is a unique n-vector, $\vec{\Theta}$, such that $I(\vec{\theta})$ is maximized for $\vec{\theta} = \vec{\Theta}$. The task of our servo will be to maintain the vector $\vec{\theta}$ as close as possible to $\vec{\Theta}$, using only $I(\vec{\theta})$ as an input signal. The process is analogous to that in the previous section. In this case, we introduce simultaneous dithers to all of the degrees of freedom. Clearly, if we want our servo to be able to distinguish one degree of freedom from another, we must choose our dither frequencies to all be different. More will be said later about appropriate choices for our dither frequencies. We also make the following two assumptions:

$$\frac{\partial I}{\partial \theta_i} \propto (\theta_i - \Theta_i)\,, \tag{11}$$

and

$$D_j \frac{\partial^2 I}{\partial \theta_i \partial \theta_j} \ll \frac{\partial I}{\partial \theta_i}\,, \tag{12}$$

where the dither applied to θ_j is:

$$d_j(t) = D_j \sin \omega_{dj} t\,. \tag{13}$$

Equation (11), like equation (2) is only valid within some region R. We need this requirement so that we can derive $\vec{E} = (\vec{\theta} - \vec{\Theta})$ from the gradient $\vec{\nabla} I$. How well an approximation this is, and how large R is depend on the physical system in question. In section II, we had the requirement, see equation (5), that the dither frequency be much higher than any other frequency with a significant component in θ. This allowed us to treat θ_0 as independent of time in equation (7). In the n-dimensional case, however, $\partial I/\partial \theta_i$ is a function of n variables, each with a dither comparable in magnitude to that of θ_i. Thus, equation (12) is a necessary requirement so that we may be able to extract $\partial I/\partial \theta_i$ from the single Fourier component $\tilde{I}(\omega_{di})$. Given equations (11), (12), and (13), we can generalize equation (7) to:

$$I(t) = I(\vec{\theta}_0) + \sum_{i=1}^{n} D_i \sin \omega_{di} t \left.\frac{\partial I}{\partial \theta_i}\right|_{\vec{\theta}_0}\,, \tag{14}$$

where $\vec{\theta}_0(t)$ is the time dependence of the system input parameters not including the n dithers.

From equation (14), we see that we can extract each of the partial derivatives $\partial I/\partial\theta_i$ by finding the Fourier components $\tilde{I}(\omega_{di})$. This can be done using n lock-in amplifiers, as described in the previous section. By equation (11), the error $\vec{E}(t)$ is just proportional to the gradient $\vec{\nabla}I(t)$. Thus, to complete our n-dimensional servo, we once again need to determine our correction signal $\vec{C}(t)$. Generalizing equation (10), we will choose:

$$C_i(t) = -g_i \int_0^t e^{\omega_{0i}(t'-t)} S_i(t') I(t')\, dt' \,, \qquad (15)$$

Once again, I should mention that I cannot demonstrate this to work for high loop gain. At low loop gain, in two dimensions, however, this seems to work about as well as the one dimensional counterpart, provided the two degrees of freedom are orthogonal. Two degrees of freedom are orthogonal if:

$$\frac{\partial^2 I}{\partial\theta_i \partial\theta_j} = 0 \,, \qquad (16)$$

which is a similar, but stronger requirement than equation (12).

IV. The Software

The software which I created for this project was deliberately designed so that it could be used with just about any servo system of the type described in sections II and III with six degrees of freedom or less. A complete source listing of the program, written in Microsoft QuickBASIC, is included as an appendix. Every aspect of the servo not dependent on the physical nature of the system is handled by the computer. Thus, the computer is responsible for generating the dither signals, sampling the system output $I(t)$, acting as an n-channel lock-in amplifier to determine $\vec{E}(t)$, and outputting the correction signals $\vec{C}(t)$, given by equation (15). In fact, the program can be simply modified to generate $C(t)$ given by some function other than that in (15). The program has several parameters which are chosen by the user.

The first set of user defined parameters are the dither frequencies ω_{di}. The units of frequency are very nonstandard. The dither signals, being digital, consist of a finite number of steps. The spacing in time between steps is the same for all frequencies. Thus, the lower frequencies will have dither signals which more accurately approximate sine waves. The frequencies ω_{di} are specified in units of degrees per step. The conversion between my units and Hz depend on the speed of the computer and the ADC board, and the number of degrees of freedom. With the hardware that I have been using, with only one degree of freedom, the relation is 1 deg/step$\approx$ .8 Hz. Thus, as a conceptual aid, the reader might as well think of my units as Hz, since a factor of 20% in the dither frequency should not have serious physical implications.

The next set of user defined parameters are the time constants in the low pass filters, τ_i. These are specified in units of cycles of the dither frequency. This parameter corresponds to ω_{0i}^{-1} in equation (15). Thus, if $\omega_{di} = 1$ then $\tau_i = 100$ would be about 100 seconds, and hence ω_{0i} would be about .01 Hz.

Something called *segment lengths* are also user defined parameters. In equation (15), we have a function $S_i(t)I(t)$ which we are integrating, weighted with an exponential. We can approximate this integral by a sum (as we have to do in a computer). For time intervals $\Delta t_i \ll \tau_i$, the exponential is effectively constant. Thus, we can do an unweighted average of $S_i(t)I(t)$ over the time interval Δt_i, and then do a weighted sum of these averages. The interval, Δt_i is called the segment length, and is specified in numbers of dither cycles. The correction signal ouput from the computer is only updated after each segment has been accumulated. Proper choices of these parameters, as we will see later, can eliminate beat signals and other nuisances present in analog servos.

Another set of user defined parameters are the software loop gains. These numbers simply multiply the correction signal, much as N_i do in equation (15). While these numbers contribute to the overall loop gain, they do not determine it. Additional amplifiers, filters, and optics all contribute to the total loop gain. Additional user defined parameters include ADC and DAC base addresses, which enables the user to change peripheral cards, thresholds for $I(t)$, above and below which the servo will cease operation, and flags which specify which of the six channels are on and which are off.

The final set of parameters may be the key to making a multidimensional servo like my own function properly. This is the matrix which transforms logical degrees of freedom into physical degrees of freedom. As was briefly mentioned at the end of the previous section, the multidimensional servo will function best when all of the degrees of freedom are orthogonal.

We assumed at the beginning of section III that there existed a unique vector $\vec{\Theta}$ which maximized the function $I(\vec{\theta})$. Thus:

$$\left.\frac{\partial I}{\partial \theta_i}\right|_{\vec{\theta}=\vec{\Theta}} = 0\,. \tag{17}$$

Notice that equation (17) only holds for $\vec{\theta} = \vec{\Theta}$. The single condition $\theta_i = \Theta_i$ is not sufficient for the partial derivative to vanish. Thus, a change in another degree of freedom could cause the partial maximum of θ_i to change. What we mean by orthogonality is that if all of the degrees of freedom are orthogonal, then:

$$\left.\frac{\partial I}{\partial \theta_i}\right|_{\theta_i=\Theta_i} = 0\,. \tag{18}$$

This is equivalent to requiring that the error signals E_i be independent of each other.

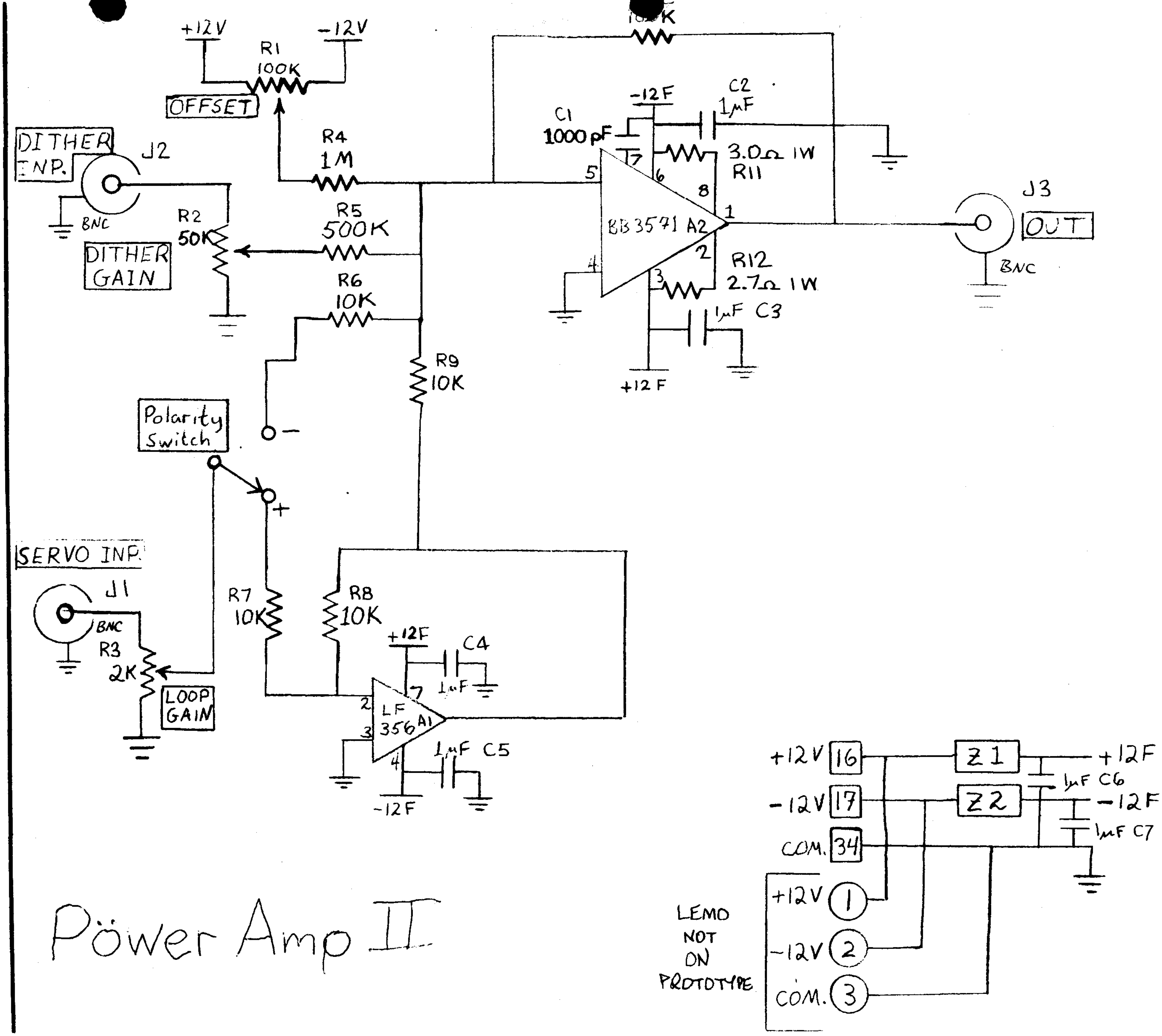
In most conceivable physical systems, we can find orthogonal degrees of freedom; however, these may not be the simplest degrees of freedom to adjust. For example, consider taking a photograph. The simplest degrees of freedom to adjust are the f-stop and the shutter speed. However, these are not orthogonal degrees of freedom, because the optimum shutter speed depends on the f-stop, and vice versa.

In the computer program, the physical degrees of freedom correspond to the various actuators being driven by the servo. The logical degrees of freedom are the orthogonal degrees of freedom, some linear combination of the physical degrees of freedom. We want each logical degree of freedom to have its own dither frequency, time constant, etc. Thus, a given actuator might be driven by some superposition of different dither signals. The program allows the user to specify the $n \times n$ matrix that transforms the logical basis into the physical. Several factors make determining the correct matrix elements a difficult process. First of all, constructing an orthogonal basis of degrees of freedom requires knowledge of the exact form of $I(\vec{\theta})$, something which is not always available. Secondly, differences in gain between the various amplifiers and actuators external to the computer must be taken into account when determining the matrix elements. Thus, to have any chance of constructing the appropriate matrix, one must do quite a bit of analysis. Even then, a great deal of trial and error will probably be unavoidable.

V. The Hardware

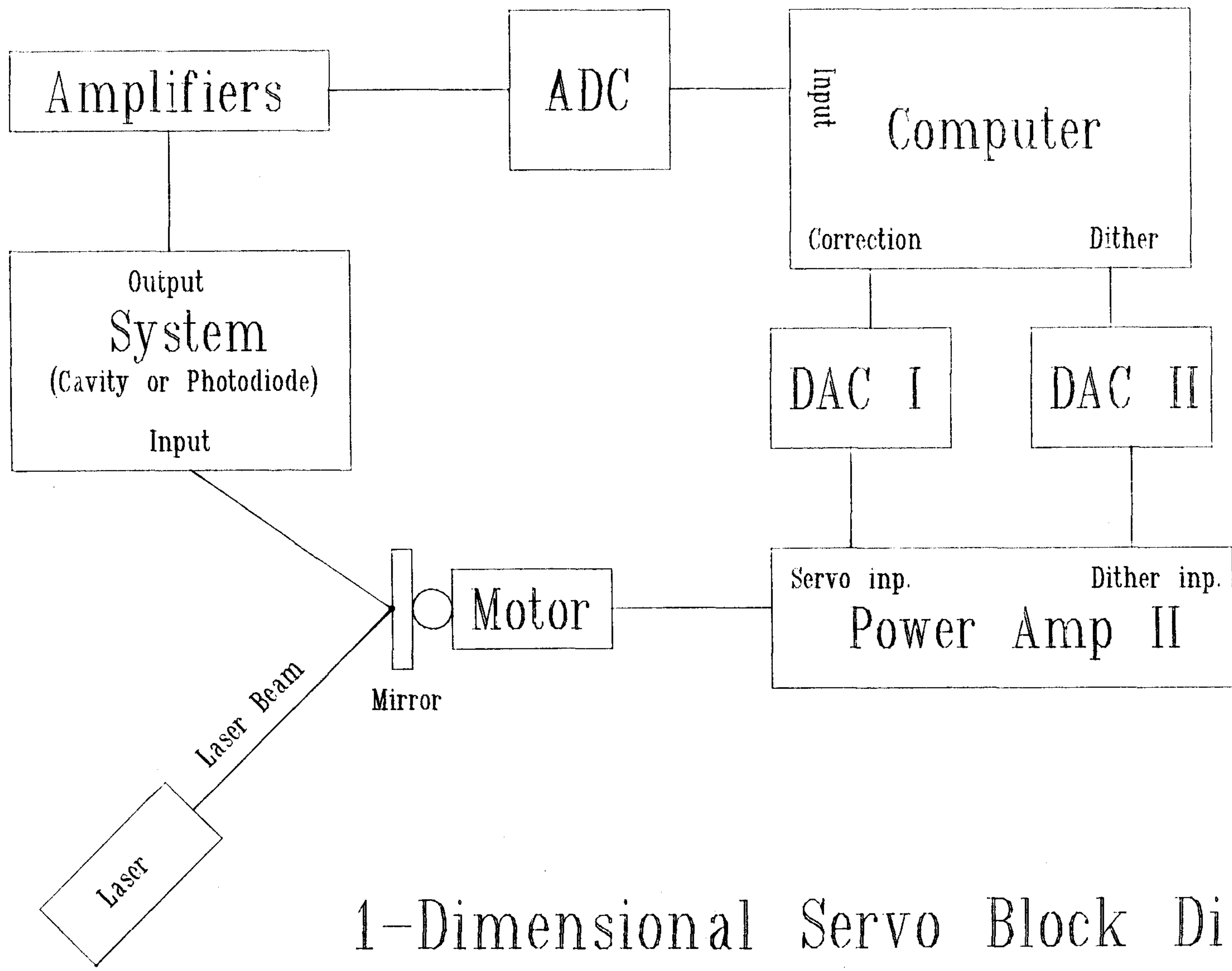
The previous section described the importance of the computer in my servo system. The type of computer which I have been using is an Everex System 1800, which is an IBM PC/AT compatible system. The analog to digital conversion is carried out with a Data Translation DT2814 A/D converter board. The DT2814 is equipped with sixteen 12-bit precision A/D channels. The computer is actually equipped with two such boards, one with a range from 0 to 10 volts, and the other with a range from -5 to +5 volts. During the year, I have used both boards, one for when my input signal $I(t)$ is strictly positive, and the other for when $I(t)$ is bipolar. The digital to analog conversion is carried out by two Metrabyte DDA-06 boards. Each board contains six 12-bit precision digital to analog converters. The boards are currently set to have ranges from -2.5 to +2.5 volts. One of the boards is used to output the dither signals, while the other board is used to output the correction signals.

The actuators in the servo (to be described later) each constitute a DC load of 28Ω. For our servo to have the maximum dynamic range, we would like to be able to supply up to 500 mA of driving current to the actuators. The D/A boards are only capable of providing a maximum of 5 mA. Thus, we must include a buffer stage between the DAC's and the actuators. We know that for our servo to be useful, the dither signal must be small compared to the expected range of our correction signal. Because, for precision, we wish to make use of the full range of our DAC, we must attenuate the dither after the computer. We also need to be free to independently adjust the gains of the dither and correction signals. The sign of the correction signal is also something we would like to be able to change, since the correct sign depends on the specific orientation of the actuators. Finally, we need to be able to adjust the DC bias going to the actuators, so that the servo can have the most range. These features are combined in the Pöwer Amp II, a dual input buffer amplifier which I designed specifically for this project. The design is based on a previous model



+12V
-12V
R1
100K
OFFSET
DITHER INP.
J2
BNC
R2
50K
DITHER GAIN
R4
1M
R5
500K
R6
10K
R9
10K
C1
1000 pF
-12F
C2
1μF
3.0Ω 1W
R11
BB3571 A2
R12
2.7Ω 1W
1μF C3
+12 F
J3
OUT
BNC
Polarity Switch
SERVO INP.
J1
BNC
R3
2K
LOOP GAIN
R7
10K
R8
10K
+12F
C4
1μF
LF 356 A1
1μF C5
-12F
+12V 16
-12V 17
COM. 34
Z1
Z2
+12F
1μF C6
-12F
1μF C7
LEMO NOT ON PROTOTYPE
+12V 1
-12V 2
COM. 3

Pöwer Amp II



1-Dimensional Servo Block Diagram

which I designed for another project involving the same type of actuators. The dither signal has an adjustable gain from 0 to .2; the correction signal gain is from 0 to 10; and the DC bias can be from -1.2 to +1.2 volts. The polarity of the correction signal can be flipped by a switch that inserts an inverting FET input op amp. The three signals (dither, correction, and bias) are added at the input of an inverting BB3571 high current amplifier, which is capable of driving 30 watts continuously, more than enough for the job at hand.

As mentioned in the introduction, the four degrees of freedom which we are trying to control are: horizontal angle, vertical angle, horizontal displacement, and vertical displacement of a laser beam. To accomplish this we need to be able to control the angles of four mirrors, two with a vertical axis, and two with a horizontal axis. The vertical and horizontal mirrors closest to the input of the optical cavity will control more angle than displacement, and the mirrors furthest from the cavity will control more displacement than angle. To obtain orthogonal degrees of freedom, linear combinations of these degrees of freedom must be used; however, it still may be possible for the servo to function using the degrees of freedom defined by the individual mirrors.

Each mirror is attached to a rectangular mirror mount, which fits over the shaft of an MFE R4-077 galvanometer. The galvanometers, described in the attached specification sheet, include two coils, each with a resistance of 15Ω. In our configuration, the coils are driven in parallel, with a single 20Ω resistor between the coils and the buffer amplifier.

VI. The Simulation

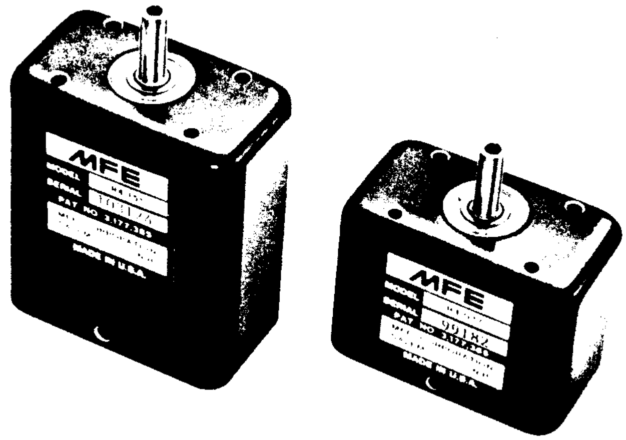
The goal of my project was to build and implement a completed four degree of freedom servo to maximize and stabilize the contrast in the ten meter mode cleaning cavity. Because the ten meter mode cleaning cavity is not yet on line, another system was needed to test my servo. The ideal system would have been the one meter mode cleaning cavity, since the functional form of $I(\vec{\theta})$, in this case the contrast as a function of the four mirror positions, is very close to that of the ten meter cavity. Unfortunately, because of other priorities, the one meter mode cleaner was not available to me. Time constraints prevented me from constructing my own stabilized optical cavity. Thus, I was left to come up with some kind of simulation

OPEN LOOP GALVANOMETERS

MFE's patented R4 Series motors are limited rotation devices that utilize the basic moving iron principle to efficiently accomplish the transition from electrical signal to mechanical force.

These motors produce an output torque proportional to input current over the angle of motion. Output rotation is tangent corrected to provide exceptional linearity in edge writing oscillographs. The tangent correction is achieved internal to the motor, thereby eliminating the need for special linearizing compensation in the drive electronics.

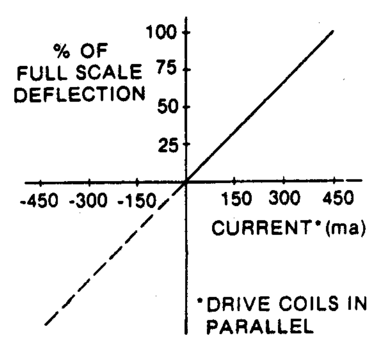
The R4-155 provides 100 Hz (-3db) and the R4-077 has 50 Hz (-3db) frequency response at 10mm amplitude when used with a total inertia load of less than 25 gm-cm².



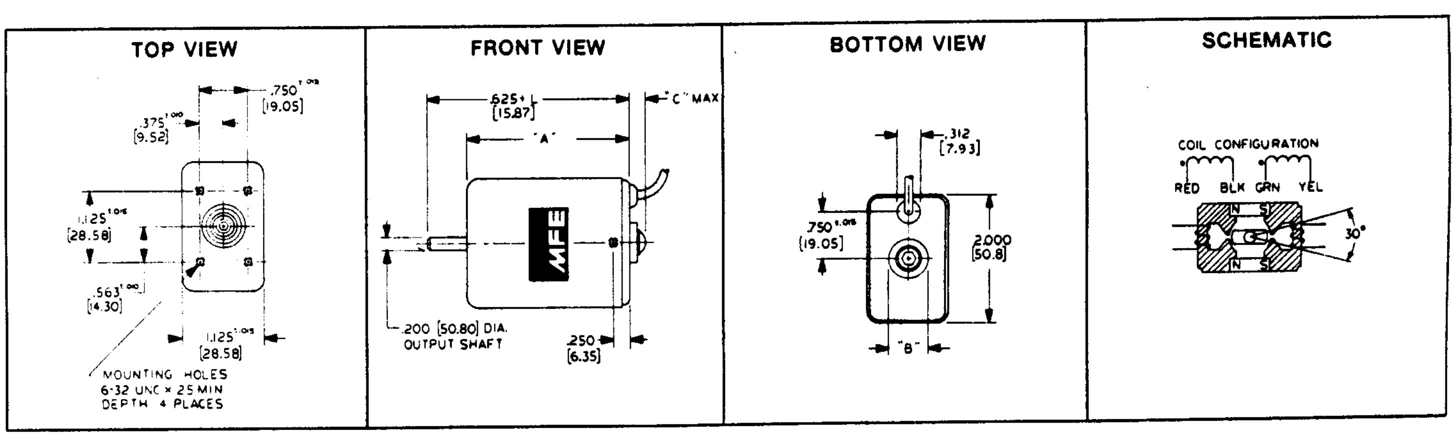
SPECIFICATIONS

	R4-077	R4-155
Rotation	±14.5 degrees	±14.5 degrees
Linearity	±0.75% of full scale	±0.5% of full scale
Hysteresis	0.75% of full scale	0.5% of full scale
Spring Stiffness	1350 ±5% gm-cm/rad	2850 ±5% gm-cm/rad
Rotor Inertia	4 gm-cm²	8 gm-cm²
Maximum Stylus Inertia	21 gm-cm²	17 gm-cm²
Natural Frequency, No Load	90 Hz	100 Hz
Back EMF Constant (Parallel Windings)	0.055 ±10% volt-second/rad	0.110 ±10% volt-second/rad
Null Torque Minimum (Full Scale Deflection)	350 gm-cm	750 gm-cm
DC Power at Full Excursion	1.5 Watts	2.0 Watts
Inductance (Parallel Winding with rotor centered)	30 ±10% mh	40 ±10% mh
Resistance*	15 ±1 ohms/coil	22 ±1 ohms/coil
Weight	9 ounces (255 grams)	15 ounces (425 grams)

CURRENT vs DEFLECTION



*Each motor is wound with two coils which may be used in series or parallel.



NOTE: ALL DIMENSION IN PARENTHESIS ARE IN MILLIMETERS.

R4-077	1.64	[41.65]	.625	[15.88]	.25	[6.35]
R4-155	2.52	[64.01]	.750	[19.05]	.15	[3.81]
DESCRIPTION	"A" DIM		"B" DIM		"C" DIM	

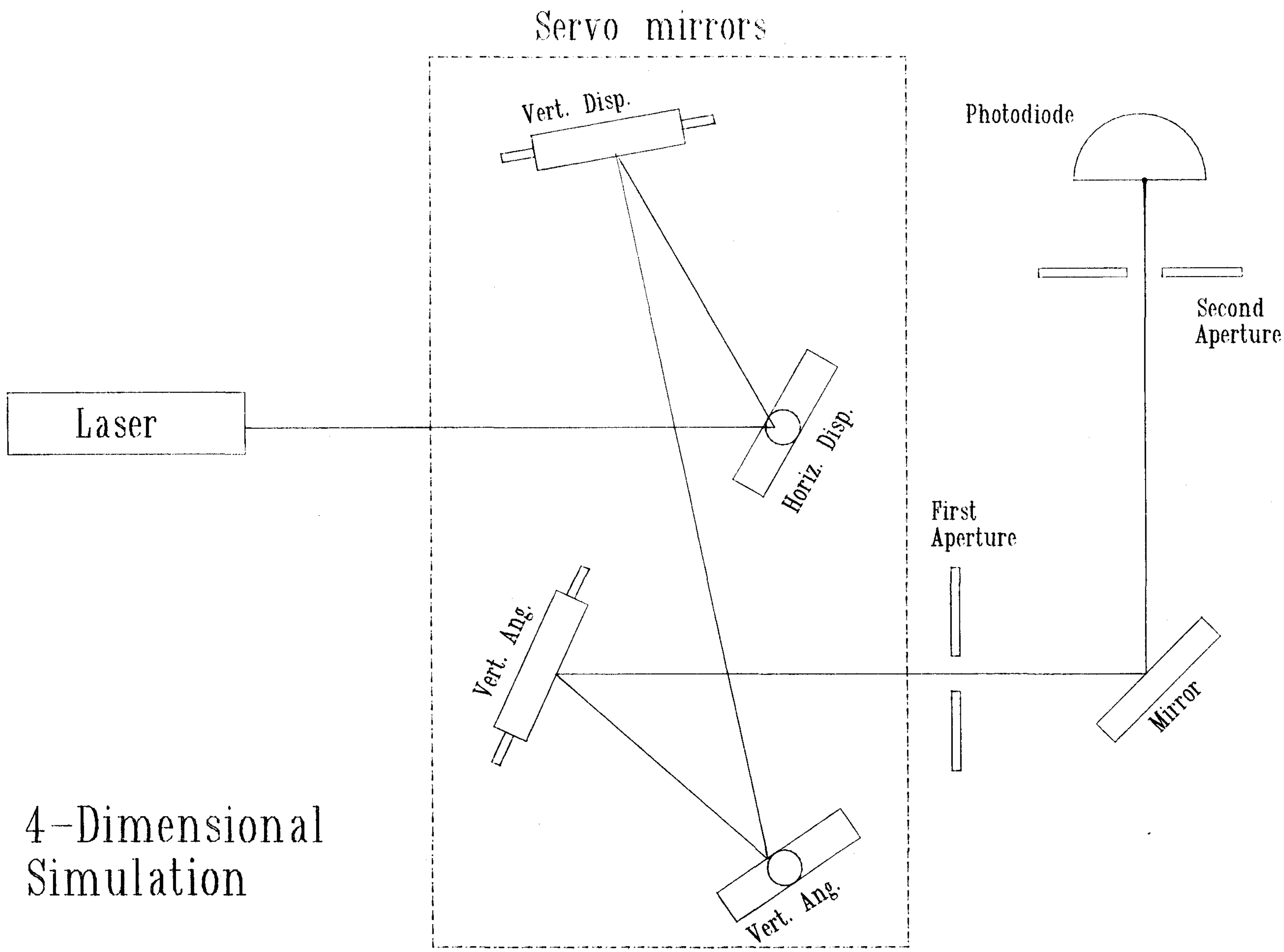
which would have an output function that depended on the four mirror positions.

The simulation I chose to use involved a laser, a photodiode, and two small apertures. In the set up, the laser is incident on the servo mirrors in the same way that it would be in the real configuration. One of the apertures is placed as close as possible to the last mirror in the servo. Thus, the position of the spot on this first aperture is mostly determined by the two earlier mirrors, which have much larger lever arms than the two later mirrors. The second aperture is placed a considerable distance after the first. The position of the spot on this second aperture is determined by the position of the first aperture, which is fixed, and by the angle of the beam, which is primarily determined by the last two mirrors in the servo. The photodiode is placed immediately behind the second aperture. Because the laser spot (ideally) is a two dimensional gaussian, the output from the photodiode should be maximized when the spot is centered on both of the apertures.

The first test is to see how well the servo works in one dimension. Since only a single mirror is involved, we can remove the first aperture. If our single degree of freedom is vertical, then we adjust the horizontal position of the spot so that it is centered on the remaining aperture. Similarly, if our degree of freedom is horizontal, we center the vertical spot position on the aperture. The exact value of ω_d is not terribly important, and several different values were used. However, all were between one and ten Herz.

Because the DC output of the photodiode, $\tilde{I}(0)$ is much larger than the component at the dither frequency $\tilde{I}(\omega_d)$, a high pass filter must be inserted between the photodiode and the A/D converter. In order for the A/D to have reasonable precision, the input signal at ω_d must be a significant fraction of the total A/D range. Without the high pass filter, the DC will saturate the amplifier, and we will get no information about $\tilde{I}(\omega_d)$. The amplifier used for this purpose is an EG&G PAR113, with a gain of 10^4, and a low frequency roll off of .1 Hz.

Another interesting point is the selection of the segment length. As mentioned earlier, this determines the interval over which $S(t)I(t)$ is averaged before the correction signal is updated. Suppose I(t) were a DC signal. In this case, $S(t)I(t)$ would just be a square wave at frequency ω_d. In fact, any part of $I(t)$ which is slowly changing compared to ω_d will have the same effect. While we do have a high pass



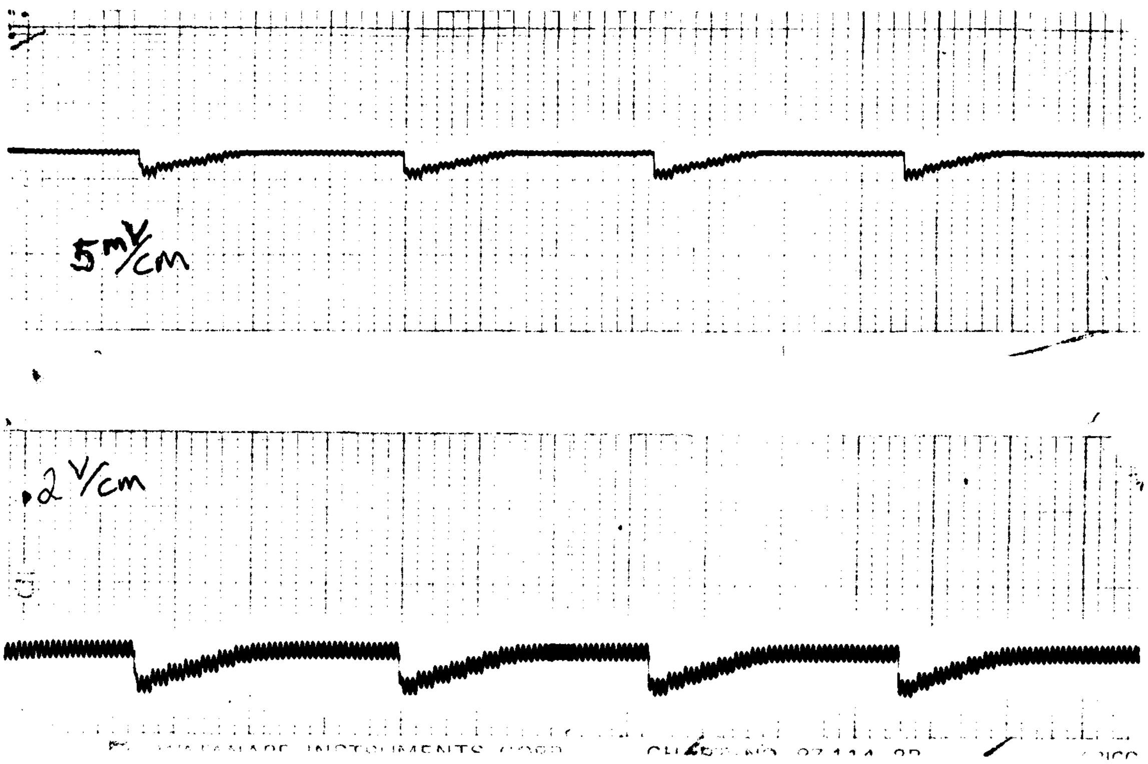
4-Dimensional Simulation

filter, there is no guarantee, particularly after sudden changes in light intensity, that the signal $I(t)$ will be centered at zero. Thus, we need some way to eliminate this effect. There is, of course, the digital low pass filter described in equation (10), which should attenuate this square wave. However, the resulting signal might still be larger than the dither signal, which would ruin the servo. We can easily eliminate this effect by requiring our segment length to be an integer times the dither period. In this case, any square wave resulting from a low frequency component of $I(t)$ will be exactly cancelled during the initial average, before any signal is output from the computer. We really do not lose anything by making such a requirement, because there is really no information contained in $I(t)S(t)$ if it is averaged over less than one cycle.

Choosing the gains is a very touchy issue. We want to choose the software gain to be high enough that the servo makes use of as much of the DAC range as possible; however, it must be low enough that the output signal never exceeds the range of the DAC. Once this is chosen, we must determine the setting of the correction signal gain knob on the Pöwer Amp II. Basically, we want this to be as high as we can make it and still have the servo be stable. Instability results when the steps in the correction signal, occurring after each segment is averaged, are too large. In this case, the servo may overcorrect, causing an oscillation to develop. We determine what this critical gain is by trial and error. The dither gain is easy to determine. We want the dither to be as small as possible, but we want enough dither so that $\tilde{I}(\omega_d)$ is large enough for the servo to derive an accurate error signal. Once all of the gains have been set, we can test our one dimensional servo.

One simple test is the step response. We find that when the gains are properly adjusted, the one dimensional servo has a smooth and consistent step response. In the example shown, $\omega_d = 3$ deg/step, segment length is 3 cycles, and the integration time constant is 300 cycles. When a 100 mV step is applied to the correction signal, we see that the servo decays back to its original output level after about 18 cycles.

Another test we can perform with the one dimensional servo is to look at how well it corrects for a low frequency large amplitude drift. The idea here is to simulate a thermal drift in the mirror position. We take the amplitude of the introduced drift signal to be roughly the diameter of the spot, and the period of the signal to be on the order of 5 minutes. Three examples are shown. In these examples,



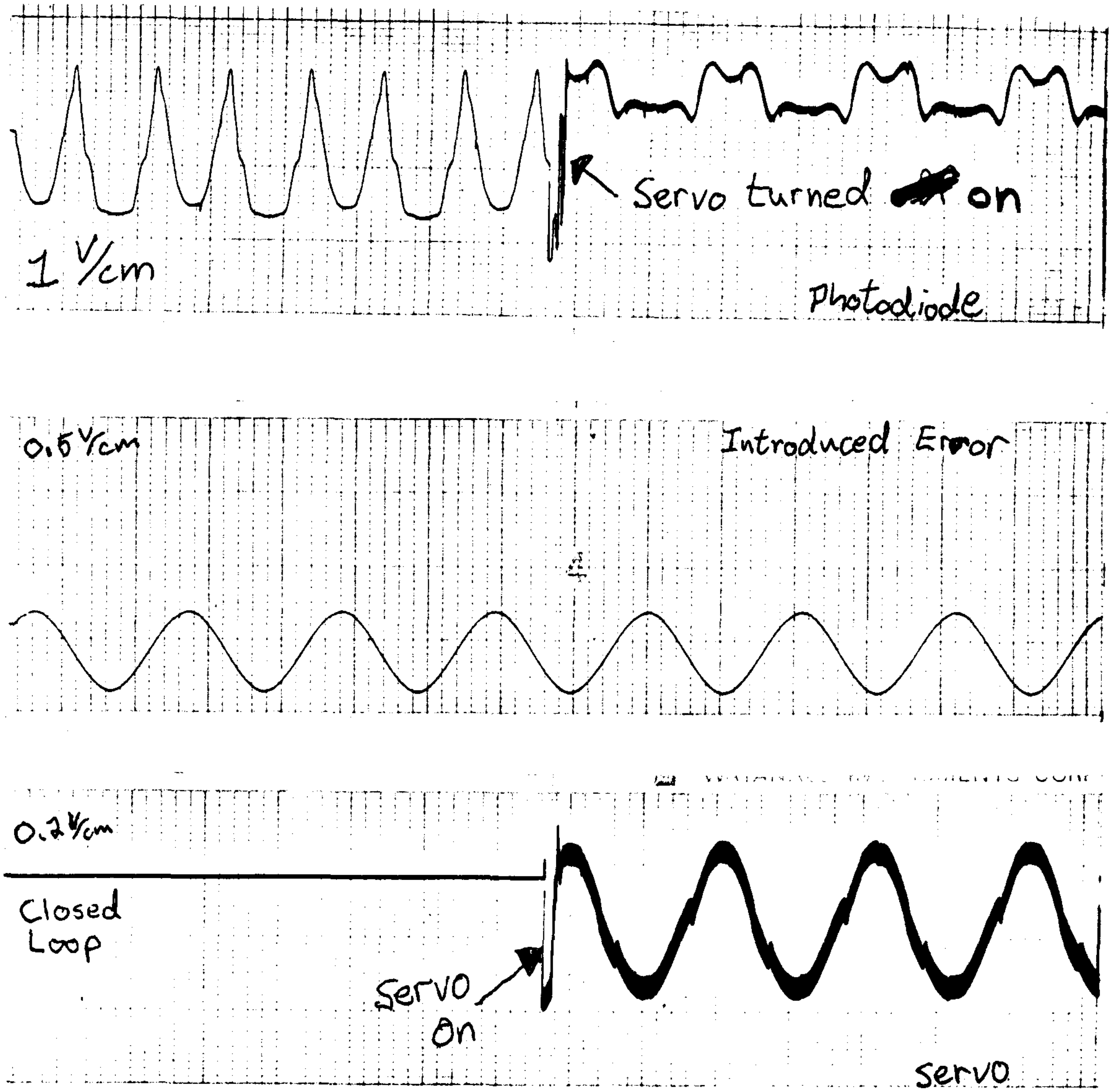
Step response of one dimensional servo: The above data shows four examples of the response of a single vertical degree of freedom to a step of 0.1 volts. The top track shows the photodiode output, and the bottom track shows the servo output. The time scale is 5 mm/sec. The settling time is about 18 cycles.

the photodiode output is shown with the servo on and with the servo off. By comparing the two cases, we can get an idea of how well the servo is correcting. In one of the examples, the servo only keeps the photodiode output to within 40% of maximum. This is an example where the servo does not have enough loop gain. In another example, the servo keeps the photodiode output essentially maximized. However, in this example, we notice that oscillations in the servo ouput occur each time the drift signal reaches a crest. This is probably an example of too much loop gain. In the third example, the servo keeps the photodiode output to within 10% of maximum. The signal appears to be quite stable. In this case, the gains are probably very well adjusted. What these examples tell us is that we must be extremely careful when adjusting the loop gain of the servo. The examples also show that the servo can indeed accomplish its intended task of drastically reducing the effect of thermal drift on the mirror alignment in one dimension.

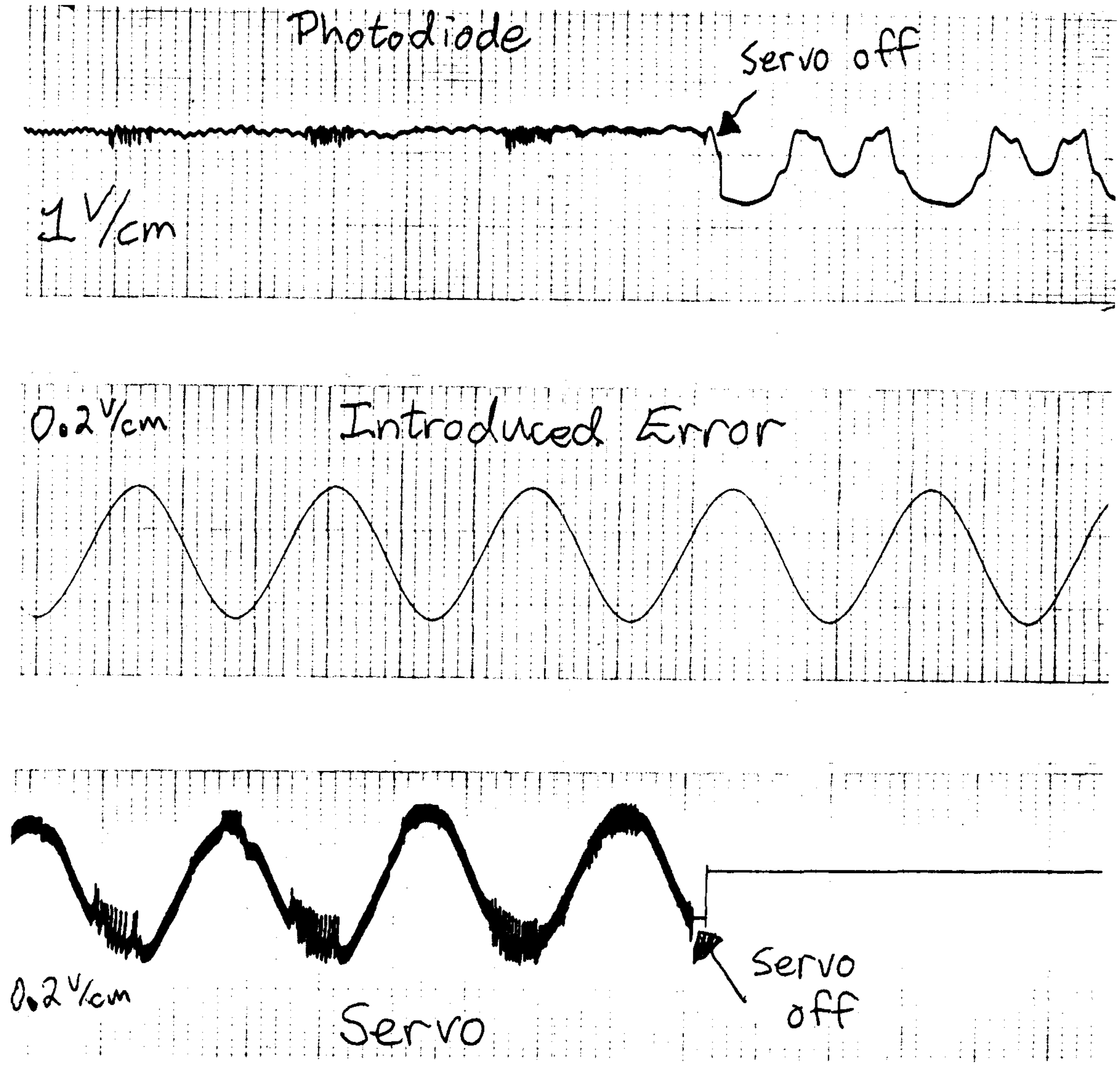
After seeing that our servo works reasonably well in one dimension, we should see how well it works in two dimensions. In choosing our two dimensions, we can either use one vertical and one horizontal degree of freedom, or we can use either both horizontal or both vertical degrees of freedom (position and angle). Intuition tells us that the former of these options will be the simpler, since horizontal and vertical spot positions should be orthogonal degrees of freedom (equation (16), where $I(x, y)$ is a two dimensional gaussian). Thus we will save the latter option until later. With only one vertical and one horizontal mirror, we do not need to distinguish between displacement and angle, thus, as in the one dimensional case, only a single aperture is necessary.

Choosing the frequencies and segment lengths have additional complications when dealing with more than one degree of freedom. One of these complications is the problem of harmonics. The only way the servo can distinguish the effects of the two degrees of freedom is by finding the Fourier components of $I(t)$ at the frequencies ω_{d1} and ω_{d2}. Because $I(\vec{\theta})$ is not a linear function, $\tilde{I}(\omega)$ will have contributions at all harmonics of the dither frequencies. If one of the dither frequencies is a harmonic of the other, then the Fourier component at that frequency will contain information about both of the degrees of freedom. Thus, in general, we require that no dither frequency be a harmonic of another.

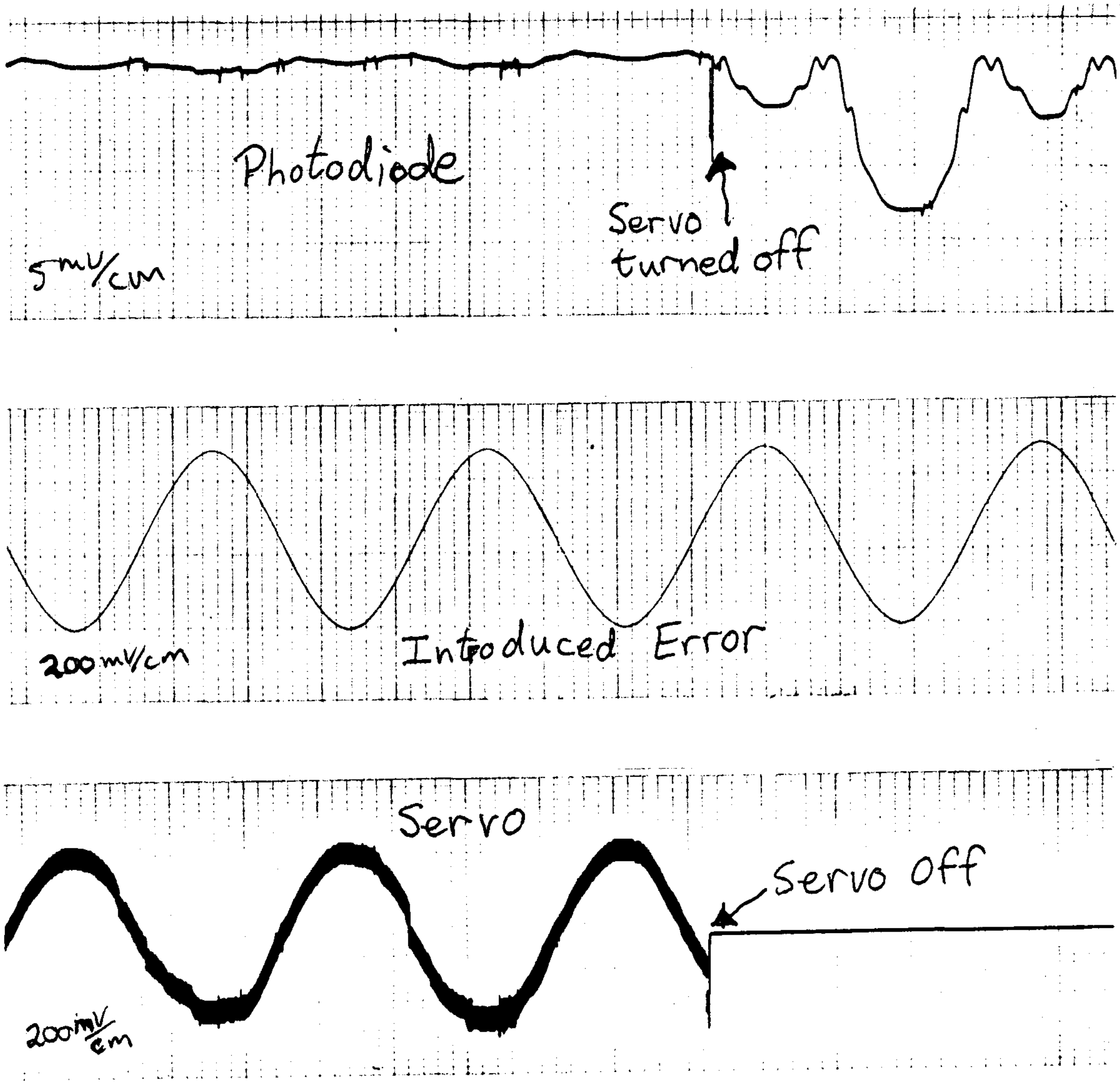
A second complication is the presence of beats. With multiple



Servo response to slow drift(insufficient loop gain): The above data show the response of a single horizontal degree of freedom when a slow error is introduced on the other horizontal mirror. The top track shows the photodiode output, the middle track shows the introduced error, and the bottom track shows the servo correction signal. Data is shown with the servo turned off for comparison.The time scale is 5 mm/min.



Servo response to slow drift(excess loop gain): The above data show the response of a single vertical degree of freedom when a slow error is introduced on the other vertical mirror. The top track shows the photodiode output, the middle track shows the introduced error, and the bottom track shows the servo correction signal. Data is shown with the servo turned off for comparison. The time scale is 5 mm/min.

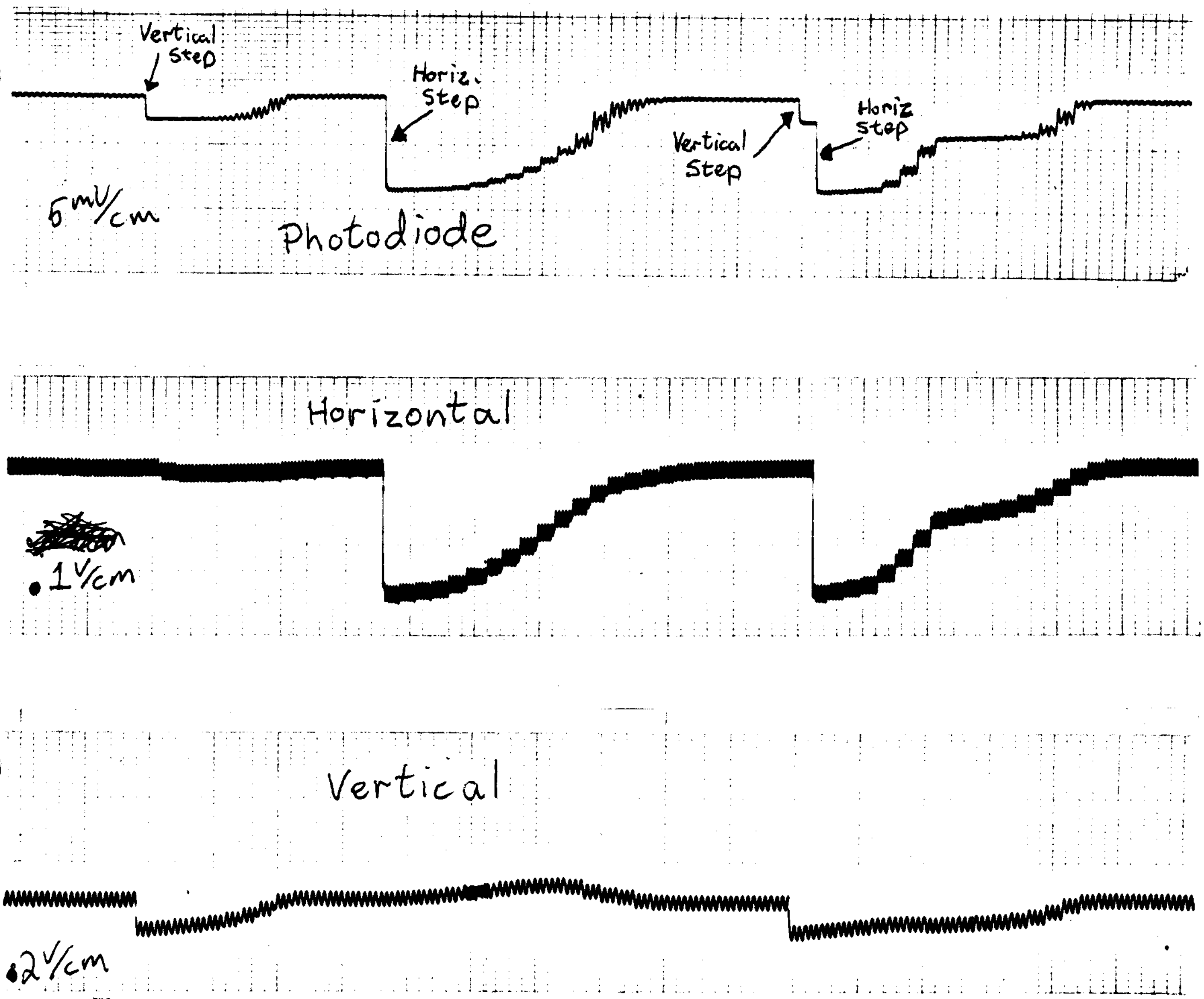


Servo response to slow drift(reasonable loop gain): The above data show the response of a single vertical degree of freedom when a slow error is introduced on the other vertical mirror. The top track shows the photodiode output, the middle track shows the introduced error, and the bottom track shows the servo correction signal. Data is shown with the servo turned off for comparison. The time scale is 5 mm/min.

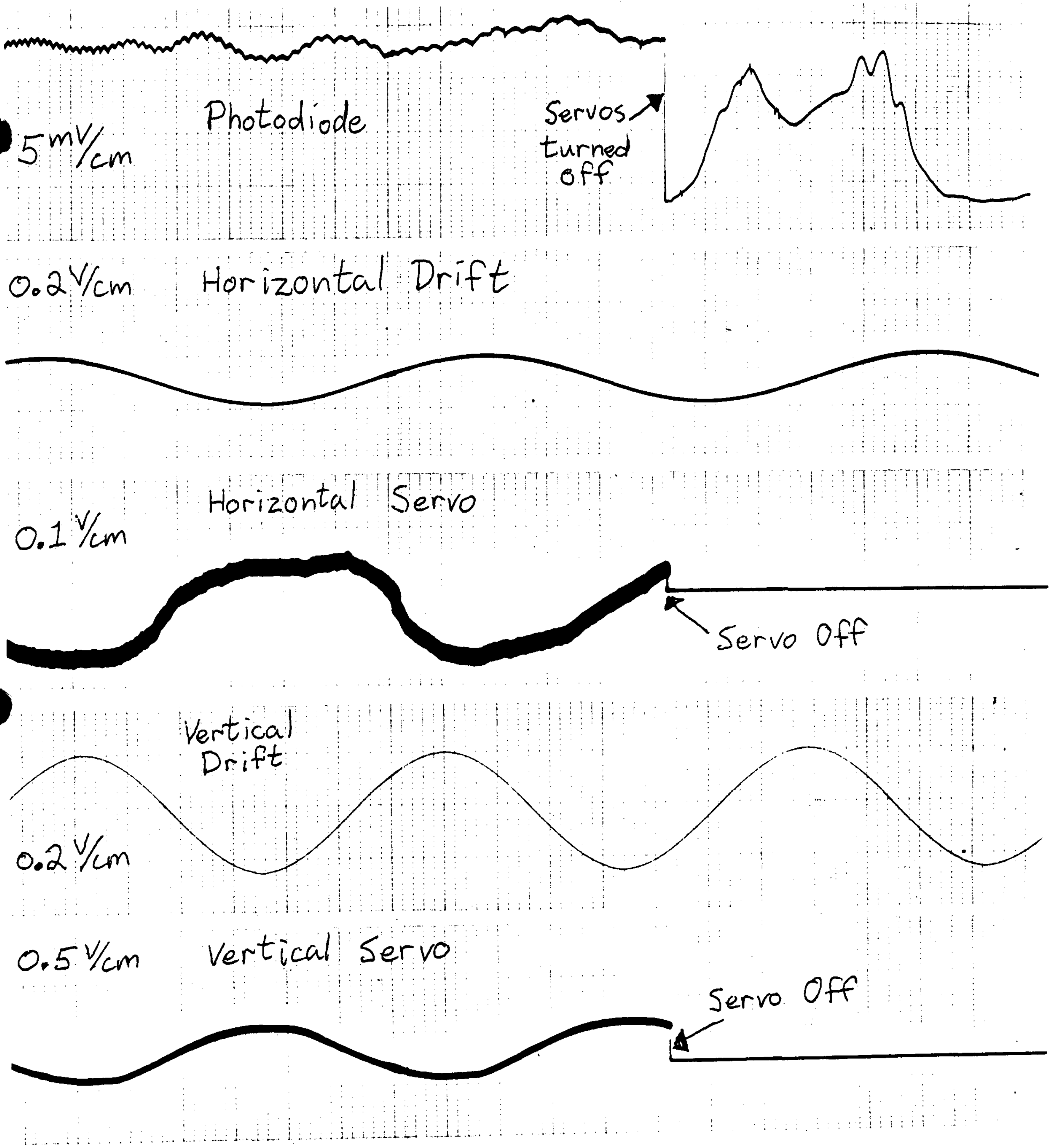
dithers, the product $S_i(t)I(t)$ will contain a beat at frequency $\omega_{di} - \omega_{dj}$ for all $j \neq i$. The amplitude of this beat signal will be comparable, and can be larger than the dither signal itself. While this is not catastrophic, it is a nuisance. We can eliminate the presence of beats if we choose our frequencies to all be multiples of a single base frequency, and then choose our segment length to be the period of that base frequency. Thus, after each complete segment, every dither signal has gone through an integer number of cycles. All of the beats exactly average out. Because no two frequencies may be harmonics, and because we want all dithers to be at comparable frequencies, it is convenient to choose our dither frequencies at prime multiples of the base frequency. In the two dimensional servo which we are considering now, the frequencies are 3 and 5 deg/step,and the segment lengths are 3 cycles and 5 cycles for vertical and horizontal respectively. Both integration lengths are 300 cycles.

One test that we can do with our two dimensional servo is the step response. With both servos operating, we first give the vertical correction signal a .1 volt step. We see that it corrects itself after 27 cycles. The only difference between this test and the one dimensional test is the presence of the other servo. All gains and other parameters are exactly the same as before. Thus, the settling time of the vertical servo is increased by 30% when the other servo is present. Next, we step the horizontal signal by .2 volts. We find that it smoothly decays back to its original level after 85 cycles. Finally, we step both signals at roughly the same time. Now, the vertical servo takes 51 cycles to recover, and the horizontal servo takes the same 85 cycles to recover as before. Probably, the reason that the vertical was more affected than the horizontal by the simultaneous stepping was that the horizontal step was twice as large. Regardless, this is a very positive result, because it means that, although we may give up a factor of two in settling time, our servo behaves qualitatively the same in two dimensions as it does in one dimension.

Another test that we can perform in two dimensions is analogous to what we did in one dimension when we applied a drift signal. Now, however, a drift signal will be applied in both horizontal and vertical degrees of freedom. Once again, the drift amplitudes will be roughly the diameter of the spot. In the example shown, the periods of the drifts are roughly fifteen minutes, with the horizontal drift having a slightly higher period. As we see, the servos manage to keep the photodiode intensity to within 25% of maximum. The fact



Step response in two dimensions: The above data show the responses of a single horizontal and a single vertical degree of freedom to an applied step in the servo output voltage. The vertical step is 0.1 volts, and the horizontal step is 0.2 volts. Each degree of freedom is stepped twice, once by itself, and once simultaneously with the other. The top track shows the photodiode output, the middle track shows the horizontal servo output, and the bottom track shows the vertical servo output. The time scale is 5 mm/sec.



Two dimensional response to slow drift: The above data show the responses of a single vertical and a single horizontal degree of freedom to slow drift signals introduced on the other two mirrors. The top track shows the photodiode output, the second track shows the introduced horizontal error, the third track shows the horizontal correction signal, the fourth track shows the introduced vertical error, and the bottom track shows the vertical correction output. Data is shown with the servos turned off for comparison. The time scale is 5 mm/min.

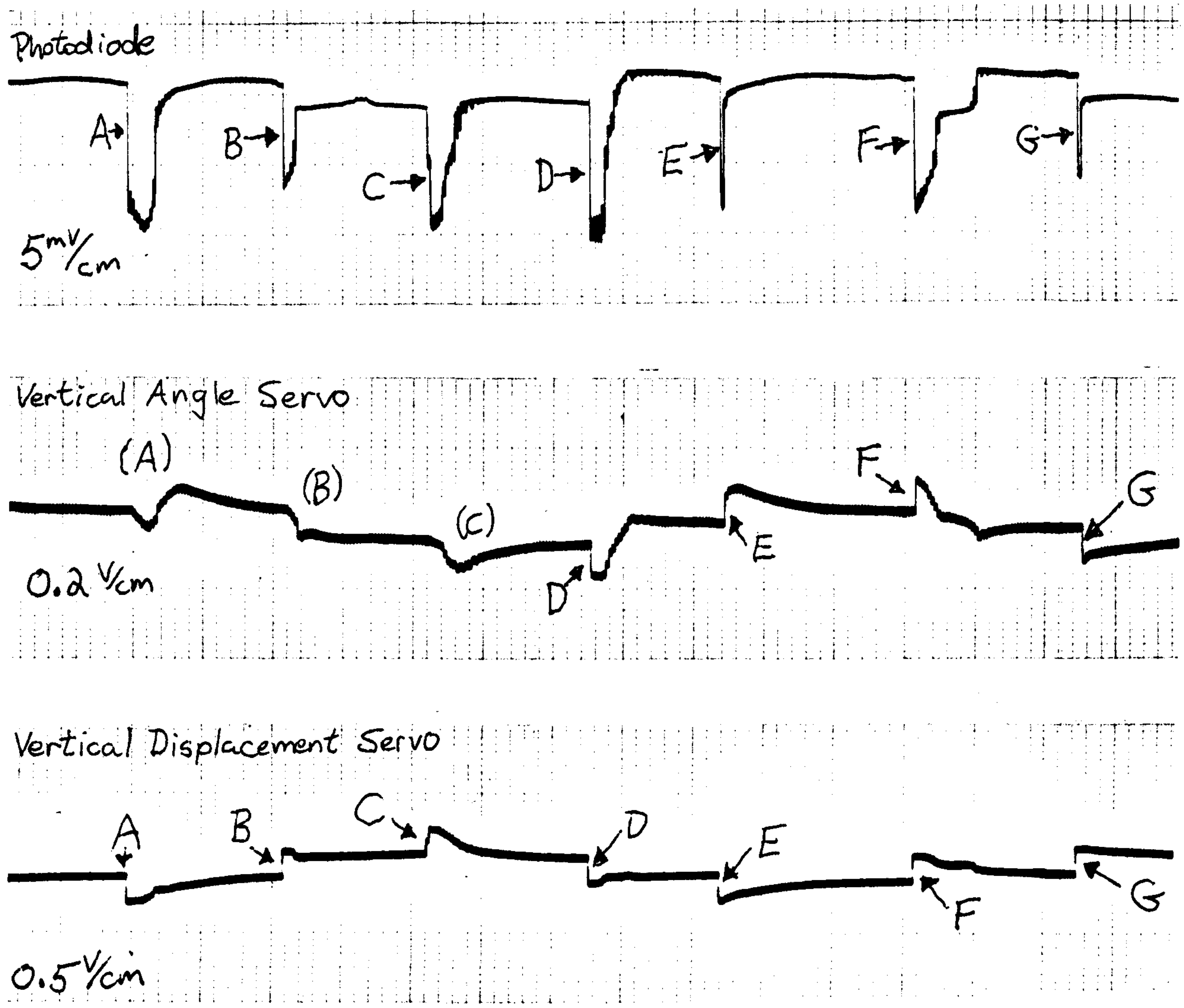
that the horizontal servo output is not a nice sine wave is probably an indication that the two degrees of freedom are not quite orthogonal. This nonorthogonalty results mainly from the fact that our laser spot is somewhat distorted, and hence is not a true two-dimensional gaussian. From these two tests, we conclude that no significant complications arise when we have two orthogonal degrees of freedom. A reasonable expectation is that when we go to an arbitrary number of orthogonal degrees of freedom, the servo will continue to perform qualitatively the same, although the settling time may increase.

We next consider the case of two coupled degrees of freedom. For this case, we must reinsert the additional aperture immediately following the last servo mirror. Because the last two mirrors in the servo are quite close to the first aperture, the spot position at this aperture is not terribly sensitive to the mirrors' orientation. The spot position at the second aperture, however, is quite sensitive to these mirrors. We will thus call these mirrors the angle degrees of freedom, because ideally, only the angles of the beam are affected at the first aperture. The two earlier mirrors have a large effect on the spot position at the first aperture, however, the spot position at the second aperture is not very sensitive to these mirrors. Thus, we call these the displacement mirrors, because ideally only the displacement of the beam is affected at the first aperture by these mirrors.

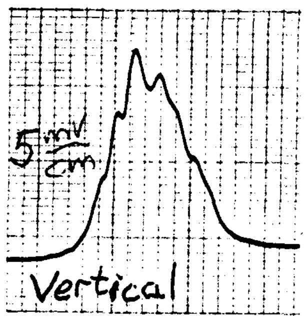
If the angle mirrors were zero distance from the first aperture, and the displacement mirrors were infinitely far away, they would represent orthogonal degrees of freedom. Because this is not even close to being the case, the angle and displacement degrees of freedom for a given axis are coupled. This means that optimum orientation for one of the mirrors is dependent on the orientation of the other mirror. For example, suppose we are dealing with the two vertical mirrors. Further, suppose that the spot is centered on the first aperture, but centered below the second aperture. The angle servo quickly responds, raising the spot at the second aperture, but also raising it slightly at the first aperture. Then the displacement servo will respond by lowering it at the first aperture, but also slightly at the second aperture. Eventually, the two should reach equilibrium, and the spot should be centered at both apertures. Of course, the correction signals could be inaccurate if equation (12) is not satisfied, but we must go under the assumption that it is satisfied if we are to hope for any positive result. Hence, we expect the servo with two coupled degrees of freedom to work, but not as efficiently as in the orthogonal case.

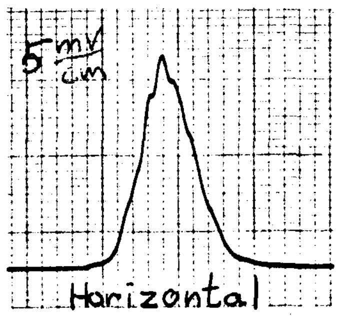
The figure shows a chart recorder record of seven events, A-G, involving the two vertical degrees of freedom. In event A, the displacement servo is given a negative step. We see that the angle servo is actually the first to respond to this step. Eventually, the two servos return to their original levels, and the system is once again stabilized at the maximum photodiode output. The settling time is about 24 seconds, roughly 8 times the settling time of the one dimensional servo. Event B corresponds to a positive step in the displacement servo. Once again, both degrees of freedom respond until equilibrium has been reached. This time, however, the equilibrium point is somewhat below the maximum photodiode output. This is the type of behavior we would expect if $I(\vec{\theta})$ had more than one local maximum. In this case, the servo will be stable at any of the local maxima, provided the dither is sufficiently small. If our step were too large, we could have thrown the spot onto another peak. Of course, we have assumed all along that the spot was gaussian. We will come back to this point later. Event C corresponds to another positive step in the displacement. This time the servos quickly settle back to the level it had before C. In event D, both servos are given negative steps. Both servos quickly settle at the levels they were at prior to B. Apparently, the steps were sufficient to throw the spot back into the neighborhood of the true maximum. Event E corresponds to a negative displacement step and a positive angle step. Once again the system settles smoothly to the true maximum. Event F is quite interesting. After both servos receive positive steps, they first settle to the equilibrium held between B and D. After about 6 seconds, they begin ascending the peak to the true maximum. Perhaps the dither was sufficiently large that it included neighborhoods of both maxima. After a sufficient number of averages, a single bit change in the correction signal may have been enough to kick the system into climbing the right hill. Event G is not terribly significant, it is simply the reverse of event E.

In the previous paragraph, we assumed that the multiple equilibrium levels could be explained by a spot with multiple local maxima. This was quite an assumption and was actually based on experiment. The simplest way to check the intensity profile of the spot is to remove the first aperture and scan the full spot across the second aperture with a ramp signal applied to one of the galvanometers. We adjust the spot so that the dimension not being ramped is centered on the aperture. What we obtain are beam profiles along the vertical and horizontal diameters of the spot. A figure showing these profiles



Step response in two coupled dimensions: The above data show the responses of the two vertical degrees of freedom to an applied step in the servo output voltage. The angle step is 0.1 volts, and the displacement step is 0.2 volts. The top track shows the photodiode output, the middle track shows the angle servo output, and the bottom track shows the displacement servo output. The time scale is 50 mm/min. An explanation of events A-G is given in the text.

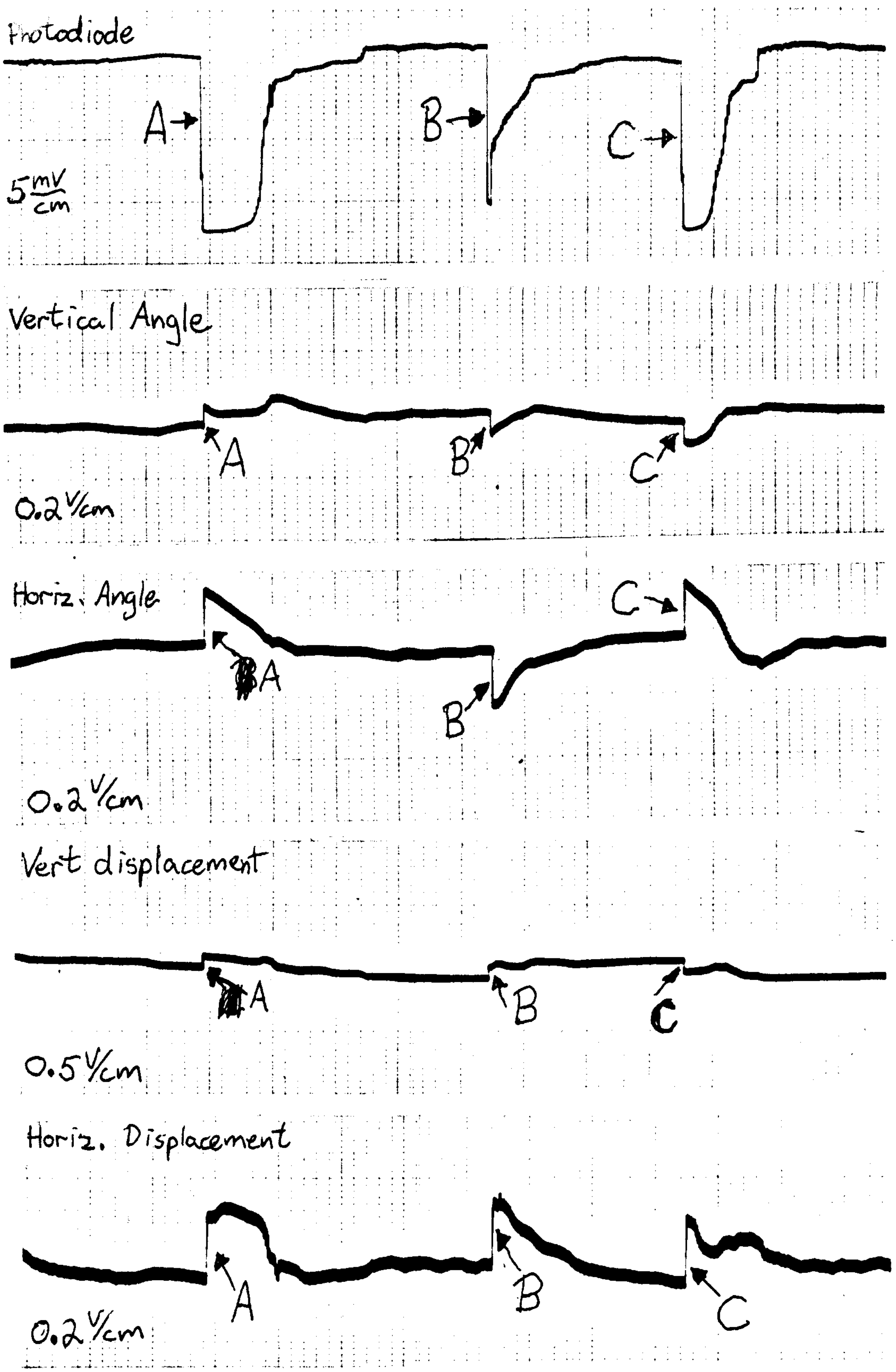




Vertical and Horizontal beam profiles: Above are shown the vertical (top) and horizontal (bottom) laser beam profiles. Each profile was generated with a ramp signal applied to an appropriate galvo motor. Notice the multiple stationary points on the vertical profile and the shoulders on the horizontal profile.

has been included. From the profiles it is quite clear that the spot has at least 3 local maxima along the vertical axis. Along the horizontal axis, shoulders are present in roughly the same places as the maxima were on the vertical. This suggests a sort of bullseye pattern characteristic of a diffraction pattern. Hence, we conclude that in all probability, diffraction off of the apertures is causing multiple maxima in the beam profile; this in turn causes the servo to occasionally stabilize itself at an intensity level lower than the true maximum. We should not be concerned by this fact. In fact, we should be happy that our servo is working as it should, i.e. stabilizing itself to stationary points in $I(\vec{\theta})$. Because this is only a simulation, we needn't bother fixing the problem, as long as we understand what is going on. In the mode cleaner, the contrast as a function of mirror angles also has many stationary points. These correspond to the various modes TEM_{ij}. Because TEM_{00}, the true maximum, has much higher contrast than the next lowest modes, we expect the servo to always be in the neighborhood of the true maximum. As always, a sudden step introduced by some mechanical or electrical disturbance could cause the servo to jump to another mode. In this case, the alignment will have to be corrected manually.

Now that we have analyzed the two dimensional servo with both orthogonal and coupled degrees of freedom, we are ready to put it all together into a four degree of freedom servo. We really do not expect anything new to happen. In fact, when we run the servo with four degrees of freedom, we see all of the effects described in the previous two paragraphs occurring between coupled servos, and we see very few effects occurring between orthogonal servos. Hence, the problem essentially decouples to that of two orthogonal pairs of coupled degrees of freedom. The step response of the four dimensional servo is shown in a figure. In this figure, event A is one in which all of the dimensions are given positive steps. The system first settles to a local maximum, and then to the true maximum. In B, the angles are given negative steps, and the displacements are given positive steps. This time, the system settles to a local maximum. In event C, the horizontals are given positive steps, and the verticals are given negative steps. After a brief plateau at the local maximum, the system settles to the true maximum. The four dimensional settling time is about 24 seconds. Thus, we find no complications in four dimensions that were not already present in two dimensions. Our simulation of the four dimensional servo, therefore seems to be a success.



Step response in four dimensions: The above data show the responses of all four mirror degrees of freedom to simultaneous step impulses. The top track shows the photodiode output, the second track shows vertical angle, the third track shows the horizontal angle, the fourth track shows the vertical displacement, and the bottom track shows the horizontal displacement. The time scale is 25 mm/min. Events A, B, and C are described in the text.

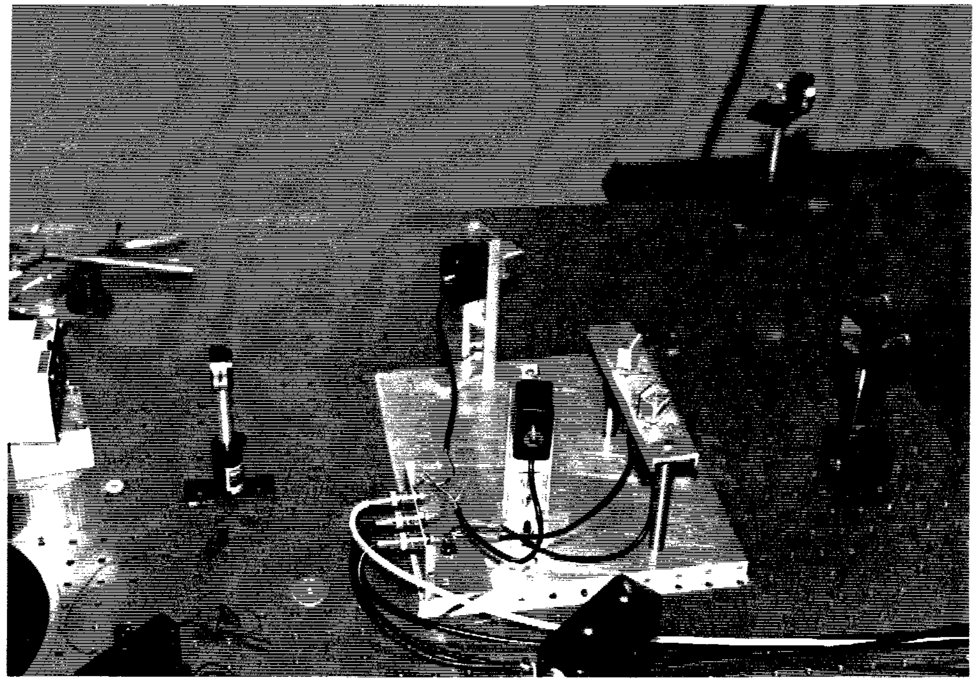
VII. Conclusion

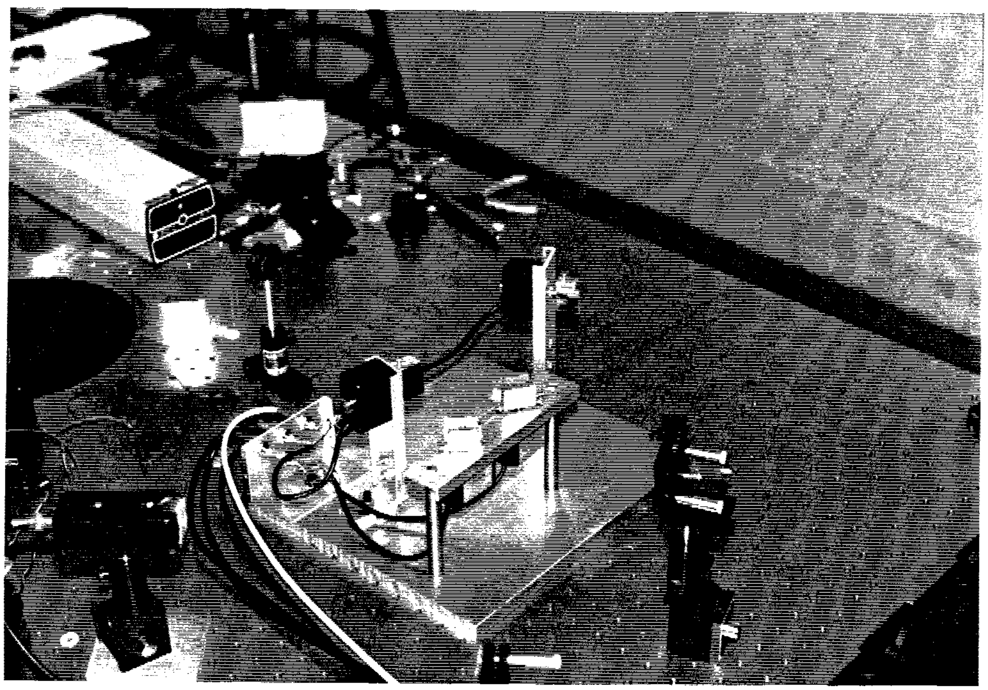
The original goal of this project was to design, build, test, and implement a computer based servo system to simultaneously optimize four mirror orientations and two lens configurations for maximum contrast in the ten meter mode cleaning cavity. Because of time and equipment constraints, the lens degrees of freedom were left for the future. One possible way of handling the lens degrees of freedom may be to make use of the lens adjustment mechanism taken from an autofocus camera; however, this is for a future project. Because the ten meter mode cleaning cavity is not yet operational, and because the one meter mode cleaning cavity is in constant use, as the never ending battle to increase the sensitivity of the gravity wave detector continues, implementation of my system has not been possible. I believe that the remainder of the project has been a success.

Because the focus of my project has been to complete a user friendly, working model of my system, capable of being implemented by those who follow, the design and construction aspect of the project have taken the better part of the year. In its current form, the computer program has simple menus by which all of the servo parameters can be changed, thus requiring no modifications to the actual program. The correction signals output by the computer, as well as the contrast (or I(t) in an arbitrary system) are displayed in real time by easy to read bar graphs. All of the required electronics outside of the computer conform to the NIM standard, and fit conveniently in a single NIM rack. The entire optical apparatus, including the four galvanometer motors and mirrors, are solidly mounted to an aluminum base. Thus, the system can be transported anywhere without requiring realignment of the mirrors. A connector panel is also mounted on the aluminum base, providing the inputs for the four galvo motors. All of the connectors, from the computer to the motors are standard low voltage BNC. Thus, all one needs to do to implement the system is to move it to the desired location, connect several BNC cables, adjust the optical path so that the beam enters the apparatus at the appropriate height and is directed to the mode cleaner after exiting the aparatus, and adjust the servo parameters for maximum loop gain and stability. This last task is by far the hardest. The servo parameters appropriate for the mode cleaner will no doubt be quite different than those in my simulation. It took me several months before I had found the correct parameters. Hopefully, my experiences will enable the process to be much shorter the next time around.

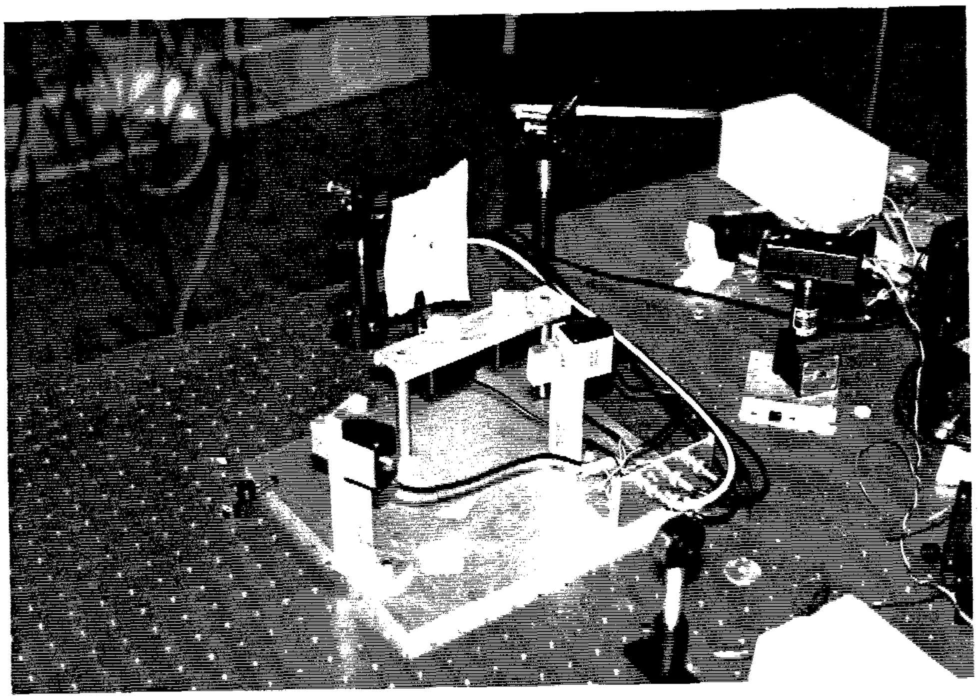
The tests included in this paper represent the most in depth that I have conducted. The results indicate that, in the simulation, the four dimensional servo should be sufficient to keep the system within 25% of its maximum output. Quantitatively, this tells us little about how it will perform when implemented with the mode cleaning cavity. Qualitatively, however, the tests tell us a lot. There was a great deal of concern, before the project was begun, that four degrees of freedom could not be optimized with only one signal from which to derive information. The fear was that the servo would go in endless circles, analagous to a cat chasing its tail. We have found that this is not the case. The four dimensional servo is stable, and it is succesful in remaximizing the output of the system after the system input degrees of freedom have been perturbed away from optimum. Furthermore, we have gained some insight into how the different degrees of freedom interact with one another. We have seen that orthogonal degrees of freedom behave very independently, and coupled degrees of freedom interact. More importantly, we have demonstrated that the interactions of the coupled degrees of freedom, while increasing the settling time, do not prevent the servo from reaching equilibrium and remaining stable.

There is little more that can be learned from the simulation which I have set up. Typically one measures the frequency response, both amplitude and phase, of a servo at many different error frequencies. This is an essential test to do before a system such as this is implemented, because it tells us what frequencies of noise will be attenuated by the servo, and it tells us what frequencies we must attenuate before the servo input so as not to feed back phase shifted corrections that can result in instability. In our simulation, we were more interested in whether or not the servo could be made to work. If we had measured the transfer function of the servo, it would have given us information which is quite dependent on the simulation and would probably not be terribly useful when the servo was implemented on the mode cleaner. Such a measurement can easily be done once the system is in its final configuration. The next step, therefore, seems to be to test the four dimensional system, if not on the mode cleaner itself, on a stabilized optical cavity, using the contrast as the servo input parameter.

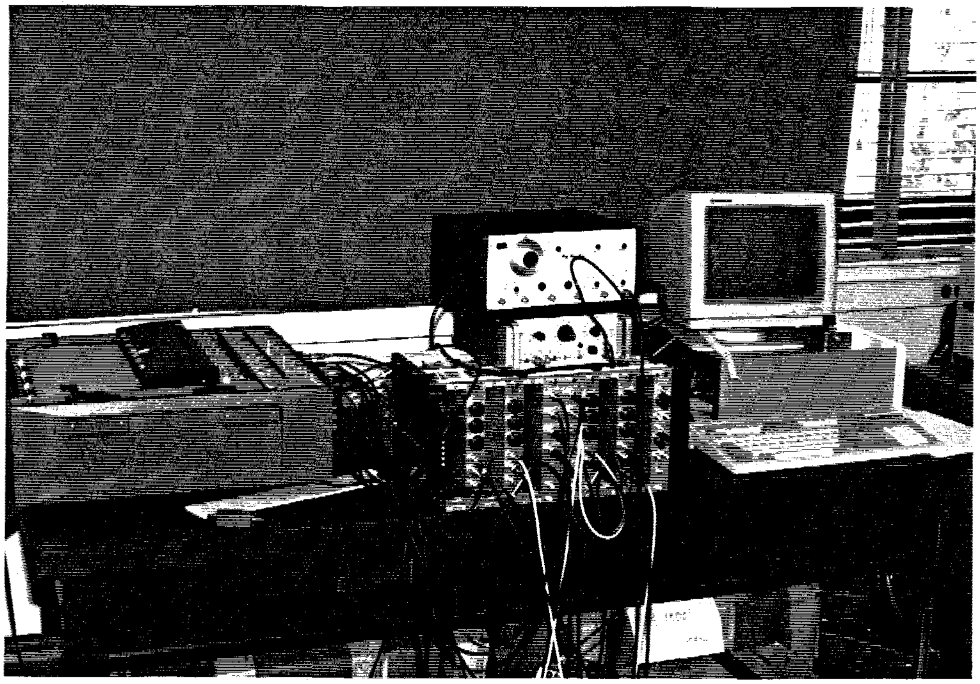


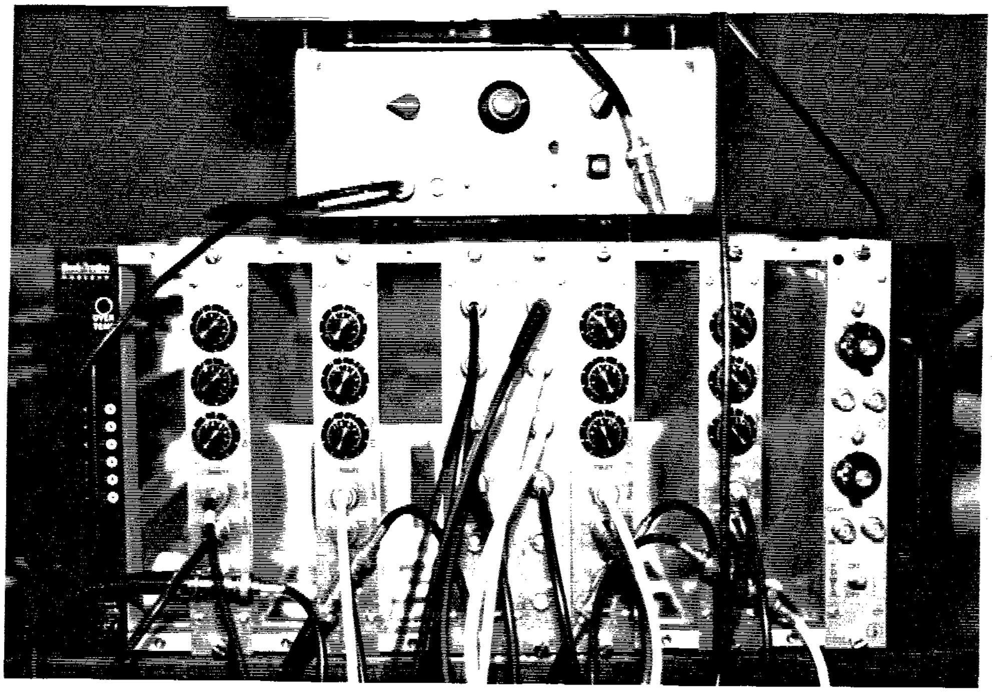


Above: Two views of the optical setup for the simulation with the first aperture removed.

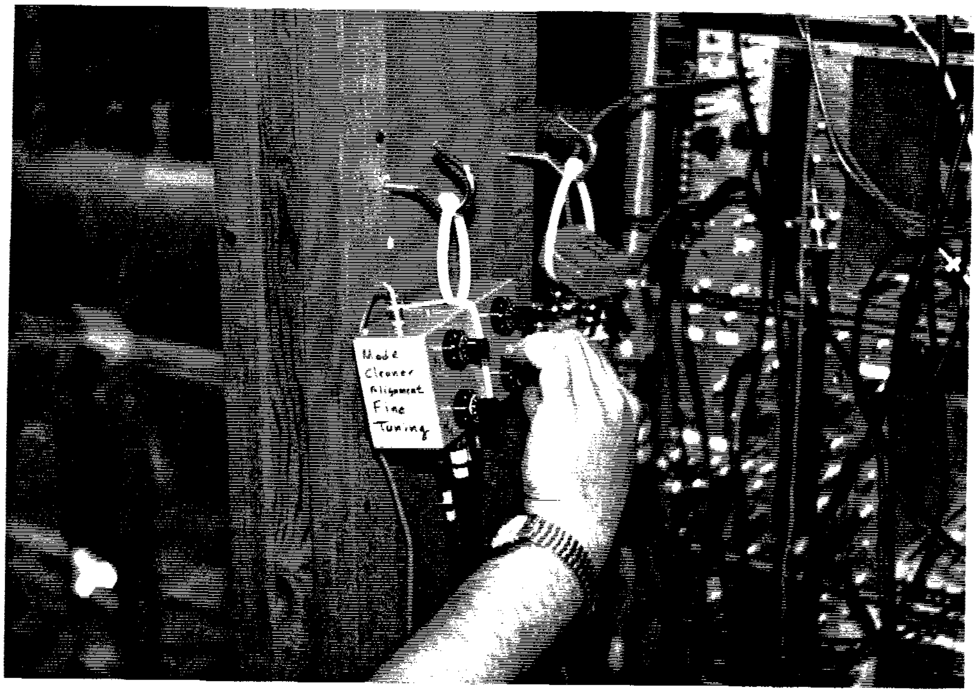


Above: Optical setup for the four dimensional simulation. Note the presence of the first aperture.

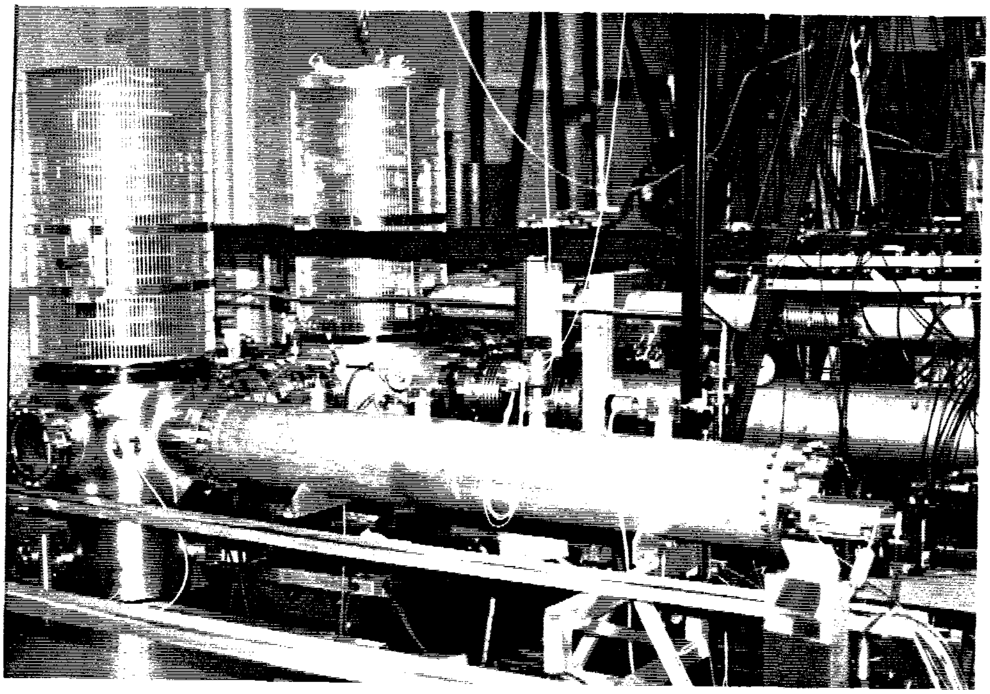


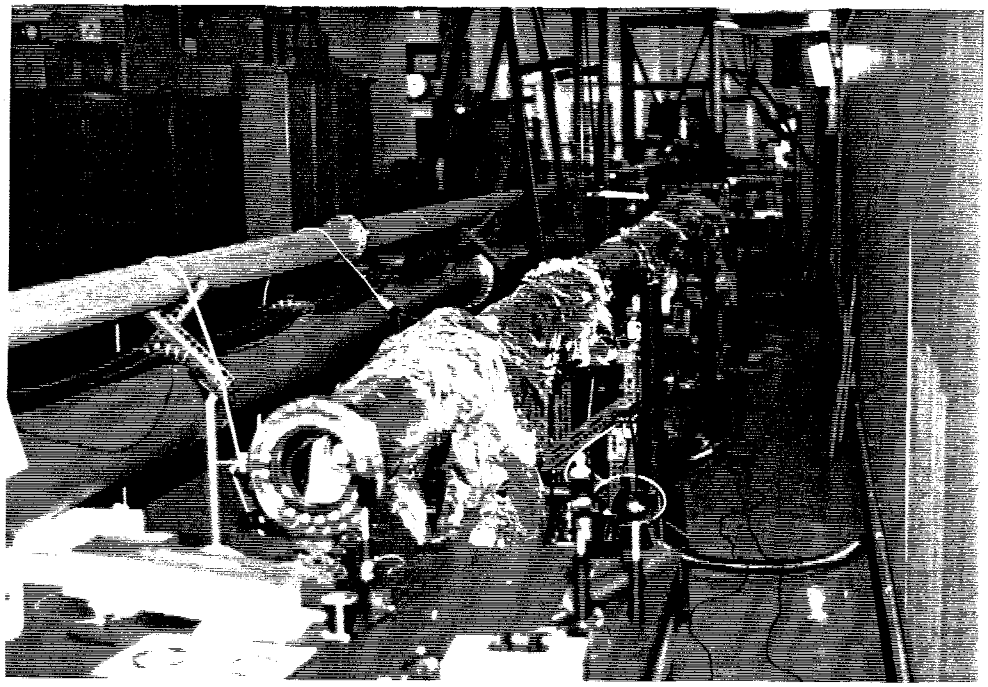


Above: The non-optical electronics involved in the servo. The modules with the three vertical knobs are the Pöwer Amp II's.



Above: The present method for maintaining mode cleaner mirror allignment. The hand belongs to Dr. Yekta Gürsel.





Above: The one meter mode cleaner (top) currently used in the system, and the ten meter mode cleaning cavity (bottom, covered in foil) currently being used to test mirrors.

```
DECLARE SUB addr ()    'Generate dither and correction DAC addresses
DECLARE SUB finish () 'Save parameters and shut down program
DECLARE SUB config () 'Allows user to change the servo parameters
DECLARE SUB renorm () 'Renormalizes contrast (not needed with contrastometer)
DECLARE SUB getip ()   'Reads first ADC channel
DECLARE SUB contrast () 'Derives contrast
DECLARE SUB getil ()    'Reads second ADC channel
DECLARE SUB bar (r%)    'Updates bar graph
DECLARE SUB graph (r%)  'Sets up bar graph graticules
DECLARE SUB start ()    'The actual servo loop
'this is the program which will control the six degrees of freedom of the
'ten meter mode cleaning cavity. There are two mode cleaning degrees of
'freedom four input beam degrees, two angular and two position.

'The following is a list of variables:
' cad(i) = DAC address for channel i correction
' dad(i) = DAC address for channel i dither
' adc = ADC base address
' e(i) = 1 if channel i enabled; 0 if channel i disabled
' w(i) = angular frequency of channel i
' t!(i) = period of integration for filter i
' s(i) = length in time of segments in filter i
' n(i) = t!(i)/s(i) = number of segments in filter i
' p(i) = +1 if dither positive, -1 if dither negative
' vp(i) = correction voltage on physical channel i
' vl!(i) = correction voltage on logical channel i
' con = contrast
' dith(i,j) = cosine lookup table
' hh(x) = high byte in DAC look up table
' ll(x) = low byte in DAC look up table
' u(i,j) = matrix element between logical and physical channels
' m(i) = number of elements in look up table for channel i
' j(i) = current index in look up table for channel i
' l(i) = current element in segment i
' cg(i) = loop gain of channel i
' g = contrast normalization constant
' hthresh  = high contrast threshold
' lthresh  = low contrast threshold
' il = laser meter value
' il0 = laser meter offset
' ip = locking photodiode value
' dc = locking photodiode offset


        DEFDBL C, G, N, W
        DEFINT A-B, D-F, H-M, O-V, X-Z
        DIM SHARED k(6), s(6), p(6), u(6, 6) AS INTEGER
        DIM SHARED e(6), m(6), j(6), hthresh, lthresh AS INTEGER
        DIM SHARED cad(6), dad(6), l(6), bs(6), adc AS INTEGER
        DIM SHARED dad, cad, il, ip, il0, dc AS INTEGER
        DIM SHARED ll(4095), hh(4095) AS INTEGER
'       $DYNAMIC
        DIM SHARED dith(6, 3000)  AS INTEGER
        DIM SHARED t!(6), vp(6), vl!(6), n(6), g, con, cg(6), w(6) AS DOUBLE
```

```
' Load configuration file
        CLS
        INPUT "Configuration file :", f$
        IF f$ = "" THEN f$ = "default.dat"
        OPEN f$ FOR INPUT AS #1
        FOR i = 1 TO 6
        INPUT #1, w(i), t!(i), s(i), cg(i), e(i)
        n(i) = t!(i) / s(i)
        FOR j = 1 TO 6
        INPUT #1, u(i, j)
        NEXT j
        NEXT i
        INPUT #1, g, il0, dc, hthresh, lthresh, adc, cad, dad
        CLOSE #1
        CALL addr

' Initialize ADC
        OUT adc, 0
        FOR i = 1 TO 10: NEXT i
        i = INP(adc) + INP(adc + 1)

' Main Menu

main:   CLS
        FOR i = 1 TO 6
        OUT dad(i), 0       '
        OUT dad(i) + 1, 8   '
        OUT cad(i), 0       '  Resets all DAC's
        OUT cad(i) + 1, 8   '
        NEXT i

        LOCATE 2, 10: PRINT "Type number of option"
        LOCATE 4, 15: PRINT "1. Begin Control Loop"
        LOCATE 6, 15: PRINT "2. Normalize Contrast"
        LOCATE 8, 15: PRINT "3. Adjust Parameters"
        LOCATE 10, 15: PRINT "4. Quit Program"

kloop:  a$ = INKEY$
        IF a$ = "" THEN GOTO kloop
        IF a$ = "1" THEN CALL start: GOTO main
        IF a$ = "2" THEN CALL renorm: GOTO main
        IF a$ = "3" THEN CALL config: GOTO main
        IF a$ = "4" THEN CALL finish: GOTO done
        GOTO kloop
done:   END

REM $STATIC
        SUB addr

'This subroutine creates the arrays which contain the addresses to be written
' to for the DAC outputs.

        FOR i = 1 TO 6
```

```
        cad(i) = cad + (i - 1) * 2
        dad(i) = dad + (i - 1) * 2
        NEXT i


        END SUB

       '
        SUB bar (r)
' This subroutine updates the bar graphs displayed during servo operation.
' r is the channel (1-6)
' l is the row on the screen
' bs(r) is the common logarithm of the multiplier of the displayed data
' v is the length of the bar graph (i.e. it is the scaled voltage)



        l = 3 * r + 2
bscal:  'Autoscaling routine for bar graph display

        wo = (vl!(r) - 2047) / 81.92 * (10 ^ bs(r))  'scales voltage for graph
        IF ABS(wo) > 25 THEN          ' Bar graph out of range
         bs(r) = bs(r) - 1            ' Decrease multiplier
         LOCATE l, 70: PRINT bs(r) ' Print new scale
         GOTO bscal
        END IF
        IF ABS(wo) < 2 AND bs(r) < 5 THEN   'Bar graph below minimum for scale
         bs(r) = bs(r) + 1            ' Increase multiplier
         LOCATE l, 70: PRINT bs(r) ' Print new scale
         GOTO bscal
        END IF

        v = INT(wo)
        IF v > 0 THEN    'Display the bar to the right of zero
           LOCATE l, 14
           PRINT STRING$(25, 32); STRING$(v, 219); STRING$(25 - v, 32)
           EXIT SUB
        END IF

        LOCATE l, 14     'Display the bar to the left of zero
        PRINT STRING$(25 + v, 32); STRING$(-v, 219); STRING$(25, 32)

        END SUB

       '
        SUB config
' This subroutine allows the user to specify all of the program parameters

        DIM mc(7), mr(7) AS INTEGER
conf:   CLS
'Display menu
        LOCATE 2, 10: PRINT "Type number of parameter to change"
        LOCATE 4, 15: PRINT "1. Dither Frequencies"
        LOCATE 6, 15: PRINT "2. Total integration lengths"
        LOCATE 8, 15: PRINT "3. Loop Gains"
        LOCATE 10, 15: PRINT "4. Base Addresses"
```

```
        LOCATE 12, 15: PRINT "5. Contrast Parameters"
        LOCATE 14, 15: PRINT "6. Enable/Disable Channels"
        LOCATE 16, 15: PRINT "7. Adjust Matrix Elements"
        LOCATE 18, 15: PRINT "8. Adjust Segment Length"
        LOCATE 20, 15: PRINT "9. Save Configuration"
        LOCATE 22, 10: PRINT "Type <ENTER> to return to main menu"

klopc:  a$ = INKEY$
       IF a$ = "" THEN GOTO klopc
        IF a$ = "1" THEN GOTO op1
        IF a$ = "2" THEN GOTO op2
        IF a$ = "3" THEN GOTO op3
        IF a$ = "4" THEN GOTO op4
        IF a$ = "5" THEN GOTO op5
        IF a$ = "6" THEN GOTO op6
        IF a$ = "7" THEN GOTO op7
        IF a$ = "8" THEN GOTO op75
        IF a$ = "9" THEN GOTO op8
        IF a$ = CHR$(13) THEN EXIT SUB
        GOTO klopc

op1: 'Change dither frequencies
        CLS
        LOCATE 1, 10: PRINT "Channel", "Frequency"
        FOR i = 1 TO 6
        LOCATE 2 + 2 * i, 10
        PRINT i, w(i)
        NEXT i
        LOCATE 16, 10
        PRINT "Channel to change (Type <ENTER> when done):";
kl1:    a$ = INKEY$: IF a$ = "" THEN GOTO kl1
        IF a$ = CHR$(13) THEN GOTO conf
        IF ASC(a$) < 49 OR ASC(a$) > 54 THEN BEEP: GOTO kl1
        i = VAL(a$): PRINT i
        LOCATE 18, 10
        PRINT "New value for w("; a$; "): ";
        INPUT v$
        IF v$ = "" THEN GOTO op1
        w(i) = ABS(VAL(v$))
        GOTO op1

op2: 'Total integration lengths
        CLS
        LOCATE 1, 10: PRINT "Channel", "Integration Length (cycles)"
        FOR i = 1 TO 6
        LOCATE 2 + 2 * i, 10
        PRINT i, t!(i) * w(i) / 360!
        NEXT i
        LOCATE 16, 10
        PRINT "Channel to change (Type <ENTER> when done):";
kl2:    a$ = INKEY$: IF a$ = "" THEN GOTO kl2
        IF a$ = CHR$(13) THEN GOTO conf
        IF ASC(a$) < 49 OR ASC(a$) > 54 THEN BEEP: GOTO kl2
        i = VAL(a$): PRINT i
        LOCATE 18, 10
        PRINT "New value for t("; a$; "): ";
```

```
        INPUT v$
        IF v$ = "" THEN GOTO op2
        t!(i) = ABS(VAL(v$)) * 360! / w(i)
        n(i) = t!(i) / s(i)
        GOTO op2

op3: 'Loop Gains
        CLS
        LOCATE 1, 10: PRINT "Channel", "Loop Gain"
        FOR i = 1 TO 6
        LOCATE 2 + 2 * i, 10
        PRINT i, cg(i)
        NEXT i
        LOCATE 16, 10
        PRINT "Channel to change (Type <ENTER> when done):";
kl4:    a$ = INKEY$: IF a$ = "" THEN GOTO kl4
        IF a$ = CHR$(13) THEN GOTO conf
        IF ASC(a$) < 49 OR ASC(a$) > 54 THEN BEEP: GOTO kl4
        i = VAL(a$): PRINT i
        LOCATE 18, 10
        PRINT "New value for cg("; a$; "): ";
        INPUT v$
        IF v$ = "" THEN GOTO op3
        cg(i) = ABS(VAL(v$))
        GOTO op3

op4:    CLS
        PRINT "ADC base address ("; adc; ") ";
        INPUT a$
        IF a$ <> "" THEN adc = VAL(a$)
        PRINT "Dither DAC base address ("; dad; ") ";
        INPUT a$
        IF a$ <> "" THEN dad = VAL(a$)
        PRINT "Correction DAC base address ("; cad; ") ";
        INPUT a$
        IF a$ <> "" THEN cad = VAL(a$)
        CALL addr
        GOTO conf

op5:    CLS
        PRINT "g ("; g; ") ";
        INPUT a$
        IF a$ <> "" THEN g = VAL(a$)
        PRINT "il0 ("; il0; ") ";
        INPUT a$
        IF a$ <> "" THEN il0 = VAL(a$)
        PRINT "dc ("; dc; ") ";
        INPUT a$
        IF a$ <> "" THEN dc = VAL(a$)
        PRINT "hthresh ("; hthresh; ") ";
        INPUT a$
        IF a$ <> "" THEN hthresh = VAL(a$)
        PRINT "lthresh ("; lthresh; ") ";
        INPUT a$
        IF a$ <> "" THEN lthresh = VAL(a$)
        GOTO conf
```

```
op6: 'Enable/disable channels

        CLS
        LOCATE 1, 10: PRINT "Channel", "Flag (1=on ; 0=off)"
        FOR i = 1 TO 6
        LOCATE 2 + 2 * i, 10
        PRINT i, e(i)
        NEXT i
        LOCATE 16, 10
        PRINT "Channel to toggle (Type <ENTER> when done):";
k17:    a$ = INKEY$: IF a$ = "" THEN GOTO k17
        IF a$ = CHR$(13) THEN GOTO conf
        IF ASC(a$) < 49 OR ASC(a$) > 54 THEN BEEP: GOTO k17
        i = VAL(a$): PRINT i
        e(i) = 1 - e(i)
        GOTO op6

op7:    'Adjusts the matrix elements u(i,j)
        FOR i = 0 TO 6
        mc(i) = 10 * i + 5
        mr(i) = 3 * i + 3
        NEXT i
st7:     CLS
        LOCATE 1, 20: PRINT "Physical Channel"
        a$ = "Logical Channel"
        FOR r = 5 TO 19
        LOCATE r, 1
        PRINT MID$(a$, r - 4, 1);
        NEXT r
        FOR i = 1 TO 6
        LOCATE mr(0), mc(i): PRINT i;
        LOCATE mr(i), mc(0): PRINT i:
        NEXT i
        FOR i = 1 TO 6
        FOR j = 1 TO 6
        LOCATE mr(i), mc(j): PRINT u(i, j);
        NEXT j
        NEXT i

        LOCATE 23, 2
        INPUT "Change element"; a$
        IF a$ = "" THEN GOTO conf
        IF ASC(UCASE$(a$)) = ASC("Y") THEN GOTO dum7 ELSE GOTO conf

dum7:   LOCATE 23, 2: PRINT "                                                            "
        LOCATE 23, 2
        INPUT ; "Logical"; i: PRINT "  "; : INPUT ; "Physical"; j: PRINT "  ";
        IF i > 6 OR i < 1 OR j > 6 OR j < 1 THEN BEEP: GOTO dum7
dum71:  INPUT ; "Coefficient"; u
        IF ABS(u) > 20 THEN BEEP: GOTO dum71
        u(i, j) = u
        GOTO st7

op75: 'Change segment lengths
        CLS
```

```
        LOCATE 1, 10: PRINT "Channel", "Segment Length"
        FOR i = 1 TO 6
        LOCATE 2 + 2 * i, 10
        PRINT i, w(i) * s(i) / 360!
        NEXT i
        LOCATE 16, 10
        PRINT "Channel to change (Type <ENTER> when done):";
k175:    a$ = INKEY$: IF a$ = "" THEN GOTO k175
        IF a$ = CHR$(13) THEN GOTO conf
        IF ASC(a$) < 49 OR ASC(a$) > 54 THEN BEEP: GOTO k175
        i = VAL(a$): PRINT i
        LOCATE 18, 10
        PRINT "New value for s("; a$; "): ";
        INPUT v$
        IF v$ = "" THEN GOTO op75
        s(i) = 360! * ABS(VAL(v$)) / w(i)
        n(i) = t!(i) / s(i)
        GOTO op75


op8:  'Save new configuration

       LOCATE 22, 10
       INPUT "filename: "; f$
       OPEN f$ FOR OUTPUT AS #1
       FOR i = 1 TO 6
       PRINT #1, w(i), t!(i), s(i), cg(i), e(i)
       NEXT i
       PRINT #1, g, il0, dc, hthresh, lthresh, adc, cad, dad
       CLOSE #1
       GOTO conf


        END SUB

       '
        SUB contrast
' This is the subroutine to calculate the contrast.
' In its present state, the routine simply reads a single ADC and sets the
' contrast proportional to that value. This is fine if an external
' contrastometer is connected to the ADC.

'If the commented program lines are activated, then the routine will calculate
' the contrast using two ADC values, one for the locking photodiode intensity,
' and one for the laser intensity monitior voltage.

        CALL getip: CALL getil     'read both photodiodes
'        IF il < il0 + 5 THEN con = 0: EXIT SUB'laser is off
'        con = (il - il0 - g * (ip - dc)) / (il - il0) * 100
'        con = 50
'        IF con < 0 THEN con = 0
        con = (ip - 2048) / 500    ' delete line if no external contrastometer
'        c% = INT(con / 2)
'        LOCATE 23, 14: PRINT STRING$(c%, 219); STRING$(50 - c%, 32)
        END SUB
```

```
        '
        SUB finish
' Save all of the program parameters

        OPEN "default.dat" FOR OUTPUT AS #1
        FOR i = 1 TO 6
        PRINT #1, w(i), t!(i), s(i), cg(i), e(i)
         FOR j = 1 TO 6
         PRINT #1, u(i, j)
         NEXT j
        NEXT i
        PRINT #1, g, il0, dc, hthresh, lthresh, adc, cad, dad
        CLOSE #1

' Reset all DAC's to zero volts
        FOR i = 1 TO 6
        OUT dad(i), 0
        OUT dad(i) + 1, 8
        OUT cad(i), 0
        OUT cad(i) + 1, 8
        NEXT i
        END SUB

        SUB getil
        OUT adc, 0                                   'Read Laser meter ADC
finl:   IF INP(adc) < 128 THEN GOTO finl
        il = INP(adc + 1) * 16 + INP(adc + 1) / 16
        END SUB

        '
        SUB getip
        OUT adc, 1                                   'Read reflected ADC
fin2:   IF INP(adc) < 128 THEN GOTO fin2
        ip = INP(adc + 1) * 16 + INP(adc + 1) / 16
        END SUB

'
        SUB graph (r)

        l = 3 * r + 1
        FOR c = 14 TO 64 STEP 5
        LOCATE l, c: PRINT CHR$(197);     'Long hash mark
        IF c < 64 THEN PRINT STRING$(4, 193) '4 short hash marks
        NEXT c
        LOCATE l + 1, 1
        IF r < 7 THEN
           PRINT "Channel"; r
           LOCATE l - 1, 39: PRINT CHR$(197)
           LOCATE l + 1, 70: PRINT 0
           EXIT SUB
        END IF
        PRINT "Contrast"

        END SUB

        '
```

```
        SUB renorm
' If no external contrastometer is present, then this routine is necessary
' to properly normalize the contrast.


        LOCATE 14, 10: PRINT "Renormalize Contrast (Y/N)"
klopr:  a$ = UCASE$(INKEY$)
        IF a$ = "" THEN GOTO klopr
        IF a$ = "N" THEN EXIT SUB
        IF a$ <> "Y" THEN BEEP: GOTO klopr
' Check if laser is on
        CALL getip: CALL getil
        IF ip = 0 OR il = 0 THEN BEEP: EXIT SUB
        LOCATE 16, 10: PRINT "Block laser, then press <ENTER>"
klopr1: IF INKEY$ <> CHR$(13) THEN GOTO klopr1
        CALL getip: CALL getil
        dc = ip: il0 = il
        LOCATE 18, 10: PRINT "Unblock green light, then press <ENTER>"
klopr2: IF INKEY$ <> CHR$(13) THEN GOTO klopr2
        LOCATE 20, 10
        PRINT "Press <ENTER> when Mode Cleaner out of lock"
klopr3: IF INKEY$ <> CHR$(13) THEN GOTO klopr3
        CALL getip: CALL getil
        g = (il - il0) / (ip - dc)
        PLAY "C16C16C16"

        END SUB

'
'************************************************************************
        SUB start
' This is the important subroutine. This routine is the one that actually
' runs the servo.

' cc(i) is the current segment, i.e. the average of the input signal times
'       the sign of the dither signal.
' p(i) is the sign of the dither signal
' ck(i) is the decay factor which creates the time constant of the filter
' vl!(i) is the voltage for logical channel i
' vp(j) is the voltage for physical channel j
' u(i,j) is the matrix relating physical channels to logical channels

        DIM cc(6), ck(6)  AS DOUBLE
        delay = 2
        s = 0                         'servo off
' Generate look up table for cosines
        FOR i = 1 TO 6
        m(i) = INT(720 / w(i))        '2 cycles
        FOR j = 0 TO m(i)
        dith(i, j) = INT(300! * COS(w(i) * j * 3.14159 / 180!) + 2048!)
        NEXT j
        NEXT i

'Generate DAC look up tables
        FOR x = 0 TO 4095
        h = INT(x / 256)
```

```
        ll(x) = x - (h * 256)      ' Low byte
        hh(x) = h                  ' High byte
        NEXT x

' Set initial conditions for servo variables
       FOR i = 1 TO 6: vl!(i) = 2048: vp(i) = 2048
       ck(i) = EXP(-1 / n(i)): bs(i) = 0
       NEXT i


' Display screen
        CLS
        LOCATE 1, 10: PRINT "Type <ENTER> to end control loop"
        LOCATE 2, 35: PRINT "Servo: OFF"
        LOCATE 2, 68: PRINT "Scale"
        FOR r = 1 TO 7
        CALL graph(r)  'sets up bar graphs
        NEXT r


klop1:  a$ = INKEY$
        IF a$ = CHR$(13) THEN EXIT SUB
        CALL contrast      'Checks contrast and updates bar graph
        IF con < hthresh THEN GOTO klop1  'cavity out of lock

' The servo is now turned on
        s = 1
        FOR i = 1 TO 6
        j(i) = 0: l(i) = 0: cc(i) = 0: vl!(i) = 2048: vp(i) = 2048
        NEXT i

        LOCATE 2, 42: PRINT "ON "

' *********       Main Servo Loop      ************

' Update dithers
servo:
        FOR i = 1 TO 6
        IF e(i) = 1 THEN                'servo enabled
        j(i) = (j(i) + 1) MOD m(i)      'element in cosine table
        FOR j = 1 TO 6
        IF u(i, j) = 0 THEN GOTO nj  ' zero matrix element
        x = (dith(i, j(i)) - 2048) * u(i, j) + 2048  'output voltage
        IF x < 0 OR x > 4095 THEN
           BEEP
           PRINT "Matrix element"; i; ","; j; "too large"
klqz:      IF INKEY$ <> "" THEN EXIT SUB ELSE GOTO klqz
        END IF

        OUT dad(j), ll(x)         '
        OUT dad(j) + 1, hh(x)    ' Output dither signal
nj:     NEXT j
        IF hh(x) > 7 THEN p(i) = 1 ELSE p(i) = -1  'sign of dither
        END IF
        NEXT i
```

```
' Sample

        CALL contrast
        IF con < lthresh THEN      ' cavity out of lock
           s = 0
           LOCATE 2, 42: PRINT "OFF"
           GOTO klop1
        END IF

' average sample into segments

        FOR i = 1 TO 6
        IF e(i) = 1 THEN           ' servo enabled
        l(i) = l(i) + 1            ' segment length increased by one sample
        cc(i) = cc(i) + con * p(i) 'add new data to segment
        IF l(i) = s(i) - 1 THEN GOSUB correct   'Update correction signal i
        END IF
        NEXT i


' check keyboard

        a$ = INKEY$
        IF a$ = "" THEN GOTO servo
        IF a$ = "1" THEN cc(1) = cc(1) + 400 '
        IF a$ = "!" THEN cc(1) = cc(1) - 400 '
        IF a$ = "2" THEN cc(2) = cc(2) + 800 '  Introduce a step in the
        IF a$ = "@" THEN cc(2) = cc(2) - 800 ' correction signal to check
        IF a$ = "3" THEN cc(3) = cc(3) + 800 ' servo response to a step.
        IF a$ = "#" THEN cc(3) = cc(3) - 800 '   (purely diagnostic)
        IF a$ = "4" THEN cc(4) = cc(4) + 300 '
        IF a$ = "$" THEN cc(4) = cc(4) - 300 '
        IF a$ = CHR$(13) THEN s = 0: EXIT SUB
        GOTO servo

' Update filter and correction signal

correct: l(i) = 0       'reset segment length to zero
         vltmp = vl!(i) 'remember current correction voltage
         vl!(i) = (vl!(i) - 2048) * ck(i) + 2048 + cc(i) * cg(i) / n(i)
                        'add current segment to weighted sum (lp filter)
        IF vl!(i) < 0 THEN
           BEEP
           vl!(i) = 0
         END IF
        IF vl!(i) > 4095 THEN
           BEEP
           vl!(i) = 4095
        END IF
        cc(i) = 0
csout:
        FOR j = 1 TO 6
        IF u(i, j) = 0 THEN GOTO nj2   'zero matrix element
        vp(j) = vp(j) + u(i, j) * (vl!(i) - vltmp) 'physical voltage
        IF vp(j) < 0 THEN vp(j) = 0
        IF vp(j) > 4095 THEN vp(j) = 4095
```

```
        OUT cad(j), ll(vp(j))
        OUT cad(j) + 1, hh(vp(j)) 'update correction output
        LOCATE j, 1: PRINT ck(j), vl!(j), vp(j)

nj2:
        NEXT j
        CALL bar(i)                'update bar graph
        RETURN

        END SUB
```