T-3277

The Design of
A Multipurpose Optical Printhead Evaluation Device

By

Jeff Squier

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

T-3277

A thesis submitted to the faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Physics).

Golden, Colorado
Date 7/1/86

Signed: Jeff A. Squier
Jeff A. Squier

Approved: Frank V Kowalski
Dr. F. V. Kowalski
Thesis Advisor

Golden, Colorado
Date 7/1/86

Signed: F. D. Schowengerdt
Dr. F. D. Schowengerdt
Head, Physics Department

Abstract

A multipurpose optical printhead evaluation device was designed primairily as an instrument which is capable of quantifying the performance of a laser printhead. A Charge-Coupled Device (CCD) linear array measures the intensity profile of the laser beam at the surface normally occupied by a photoconductive drum. From these measurements various mathematical entities such as the modulation transfer function and image fidelity are calculated to help evaluate the printhead. The device is generic enough in nature that the instrument might also be used to evaluate printheads based on other technologies such as LED's.

Table of Contents

List of Figures

Acknowledgements

First and foremost I would like to thank Dr. Frank V. Kowalski without whom none of this would have been possible. He is both an inspiring teacher and researcher, and I consider myself extremely fortunate to have been able to work with him and learn from him.

Next I would like to acknowledge my thesis committee members Dr. James Brown and Dr. William Law. Their patience in having to read this thesis and suggestions on how to improve it were deeply appreciated.

I would also like to acknowledge John Udd who designed the electronics for this project. John was always willing to patiently explain things to me when they weren't quite clear. Thanks John!

Special thanks goes out to Mike Stanich of the IBM Corp. Not only was his help in this project invaluable, but he made it possible for me to see how an impressive organization like IBM operates.

Finally, I want to acknowledge the IBM corporation which supported this research.

I. Introduction

Information is being accumulated and processed today at heretofore unseen rates. In order to accommodate the tremendous output of data (in terms of written text and graphics) the high speed non-impact printing industry has developed several competing technologies. Chief among these are the laser printer using either polygonal mirrors, hologons, or galvanometers to scan the laser beam across a photoconductive surface. Other types of printheads include a linear array of LED's which can be turned on and off independently to print an entire line at one time. Consequently, a producer in this industry is forced to confront several questions in order to assure that his devices are capable of providing optimum output at competitive data rates. For instance, does the printer provide comparable quality and printing speed to that of other instruments based on the same or different technology? In addition, what improvements can be made to the present machines so that better resolution and higher speeds can be achieved?

We have developed a design tool to aid in answering these type of questions. The instrument is capable of measuring and evaluating numerous parameters which are used to determine overall printhead quality.

In addition, the device operates at sufficiently high rates so that real-time waveforms can be captured and analyzed.

The basic instrument consists of a CCD (Charge-Coupled Device) linear array mounted at the position normally occupied by the photoconductive surface. The CCD is used to measure the intensity profile of the incoming waveform as it appears at the surface of the photoconductive drum. Using these intensity measurements several important quantites can be calculated which are descriptive of printhead performance. Perhaps the most critical of these are pel placement accuracy (in both the scan and process directions) and spatial pulse width. Also included are measurements of the Modulation Transfer Function, and image fidelity.

In the analysis which follows, all measurements were performed on a laser printer which incorporates a polygonal mirror to scan the beam.

II. Types of Measurements --- Based on a Laser Printer

The relevance of the different types of measurements made for a particular system becomes especially clear with a basic working knowledge of how the printhead operates. Of particular interest here are laser printers which use a polygonal mirror to steer the laser beam across the surface of the photoconductor (1, 2, 3). The basic components comprising a laser printer are depicted in figure 1 below. (Note: The illustration is not drawn to any scale. Optical components are represented in this drawing and those in the remainder of the text simply as boxes.)

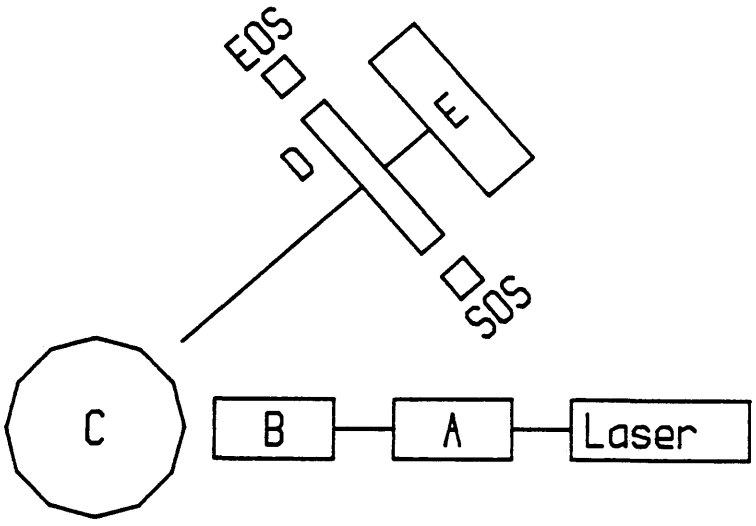

Figure 1. Basic Laser Printer Components

The first device encountered by the laser beam is an acousto-optic cell (A) which is used to modulate the intensity of the laser beam so that the appropriate series of dots, dashes, and lines appears at the surface of the photoconductive drum.

Having passed through the modulator, the beam is conditioned by some pre-scan optics (B). The optical requirements here differ from printer to printer depending on how the manufacturer decides to illuminate the facets of the polygonal mirror (C).

Since the mirror is rotating, the deflected laser beam moves in the direction of mirror rotation, producing a single scan for each facet. Ideally, the distance the beam is scanned is directly proportional to the scan angle. If the scan distance is some nonlinear function of the scanning angle, distortions are introduced (see figure 2).

After being deflected from the mirror, the beam enters an anamorphic system of post-scan optics (D). This series of lenses might be used to correct for errors in the polygonal mirror. For instance, the plane containing the facet should be parallel to a plane containing the axis of mirror rotation. For small deviations from this, any tilt may be corrected optically.

Upon leaving the post-scan optics, the beam comes to a focus on the surface of a photoconductive drum (E) which is also rotating. The direction perpendicular to the scan direction of the laser beam is known as the process direction.

Note that at the beginning and end of the scan, are positioned start-of-scan (SOS) and end-of-scan (EOS) detectors which are essential to the timing of the system. These provide translational information about the beam so that a series of pixels can be made to form lines in the process direction.

Some of the critical factors in producing letter quality text should now be evident from this brief introduction to laser printing technology. Perhaps the most obvious is the need for a uniform spotsize along the length of the scan.

Another important parameter is pel placement. (A pel is defined as the shortest interval in which the laser beam can be made to turn on and off, thus writting a single point on the photoconductor.) More explicitly, the position of a specific pel centroid from one scan to the next. This is one of the factors determining how true a line can be produced in the process direction.

Other factors affecting print quality are scanning velocity distortions and beam diameter variations (4). Scan velocity distortions is the time derivative of the

distortion, the distortion being defined as

$$\text{distortion} = (\gamma - \Gamma)/\Gamma * 100\ (\%),$$

where Γ denotes the ideal scan distance as measured from the center of the scan, γ the actual scan distance, and θ the scanning angle. With f the distance shown in figure 2, γ is related to the scanning angle by the expression,

$$\gamma = f * \tan\theta$$

The ideal scan distance is then,

$$\Gamma = f * \theta.$$

The scanning velocity distortion can now be calculated by taking the time derivative of the individual terms in the first equation and is given by

$$\text{scanning velocity distortion} = (d\gamma/dt - d\Gamma/dt)/(d\Gamma/dt).$$

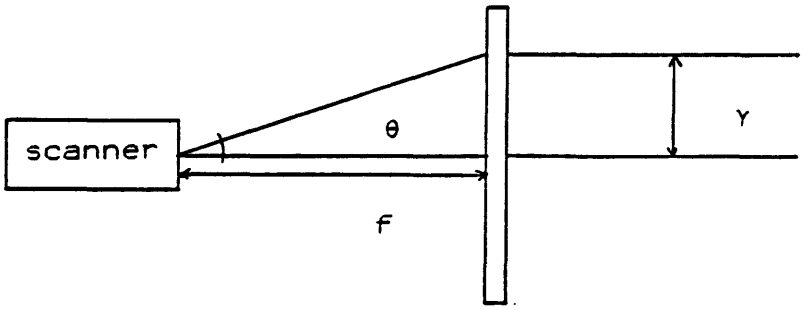

Figure 2. Scanning variables f, γ, and θ

Physically, a positive non-zero scanning velocity distortion results in the beam increasing in width in the scan direction as the angle θ increases.

The other distortion mentioned earlier, beam diameter variation, as implied by the name, refers to an elongation of the beam in the process direction.

The bench model that was developed is capable of providing measurements of all these types of distortion. Additional analysis such as image fidelity, and MTF (Modulation Transfer Function) measurements have also been provided to further scrutinize various aspects of the laser printer.

III. Instrumentation

The core of the optical printhead evaluation device is the CCD array. The array (CCD133A produced by Fairchild) consists of 1024 elements each 13 by 13 microns square located on 13 micron centers. The small photoelement size and precise geometry of the array allow for the accurate determination of the intensity profile. The array can be clocked at 20 MHz which means the entire 1024 elements can be read at rates up to 19.5 KHz. Through a combination of software and hardware support, the 1024 elements of the array are divided into 16 sectors, each containing 64 elements. Any consecutive 64 element set of blocks may be "turned on" exclusive of all the other photoelements. The purpose behind this is two-fold. First, not all measurements require 1024 data points. If only a 64 element block is needed, then data processing times can be drastically reduced, and memory requirements made less stringent. Second, if there are "dead" elements (i.e., elements whose response is extremely poor) the segments in which they occur need not be used.

The CCD array is positioned where the surface of the photoconductive drum would normally be located -- at the waist of the output laser beam. It was found impractical to mount the CCD in the actual position occupied by the drum.

Instead, a mirror is used to deflect the laser beam onto the array, so that the waist of the beam remains at the CCD sensor surface. (The mirror is of sufficiently high optical quality so that it does not introduce any distortions which could be mistaken for effects due to the optics of the laser printer.) This configuration has several advantages. First, the mirror used to deflect the laser beam onto the array can be easily translated so that a three-dimensional profile of the beam can be built up. In addition, the array can be securely fastened at one location, minimizing array vibration due to external perturbations.

Positioned underneath the array is a rotary mount which, in turn, is screwed to a translational stage. The translational stage allows the entire CCD array to be displaced along the scan direction. The rotational mount is used to align the array (when used in conjunction with the mirror translational mount) either parallel to the scan axis or perpendicular to it (i.e., along the process direction). This process could be entirely automated by using programmable actuators (such as the 855 programmable controller system available from Newport Corporation). (For a more complete description of the instrumentation see appendix C.)

The output of the CCD array is directed into a transient

recorder (LeCroy model TR8818). The recorder provides 8 bit resolution (6.2 effective bit accuracy at 5-20 MHz) of the sampled signal and is capable of 100 MHz sampling rates. Consequently, it can easily keep pace with the output of the array. The digitized waveform is stored in a 32k memory (expandable to 512k) which is transferred to an IBM AT computer for analysis.

Two interface cards were created. The first, interfaces the array to the P.C., the laser printer, and finally, the transient recorder. One of its functions, as mentioned earlier, is to provide control over which segment(s) of the array the user wishes to turn on. Another is to synchronize the operation of the device. Timing for the entire system is maintained by the pel clock produced by the laser printer. The array dumps the captured waveform to the digitizer during the interscan period. The second card is a laser printer-P.C. interface exclusively. It provides the capability to use the software to control where in the scan, and for what duration, the laser beam will be turned on. (For a complete description of the electronics see appendix B.)

IV. Array Characterization

Unfortunately, each of the 1024 elements of the CCD do not respond to the same intensity input in exactly the same manner. There seems to be a slight variation in photoelement characteristics which amounts to a constant offset from element to element in terms of response. Since this particular array has two outputs, one for the even numbered photoelements and another for the odd, additional offset occurs in the analog signal due to the summing of these output channels.

Not only was non-uniformity a problem, but the quartz window on top of the array produced interference effects. The majority of the fringes were eliminated when a wedge was placed on top of the window. A special mount was created to hold the wedge in place. Optical index matching fluid was placed in the window-wedge boundary to help match the indices of refraction between the window and wedge.

Even with the wedge, interference fringes are still visible. However, the intensity variation for an unmodulated scan across the array was shown to be less than the resolution attainable by the system. A photomultiplier tube was used to measure the scan-to-scan intensity variation. The fluctuation in intensity was slightly less than 2%. The digitizing window is set between 0 and 512mV. With an

effective bit accuracy of 6.2, there are only 73.5 levels within this window. Since the maximum output voltage of the array is approximately 1/3 volts, the system can only sense changes greater than 2%. Therefore, an unmodulated scan can be used to calibrate the response of the array. Using this scan a special algorithm calculates a characteristic gain coefficient for each of the photoelements. This gain is then divided out of all subsequent data. The effects of this routine are shown on the next two pages. The first image is the uncorrected scan, the second shows the scan with the gain divided out.

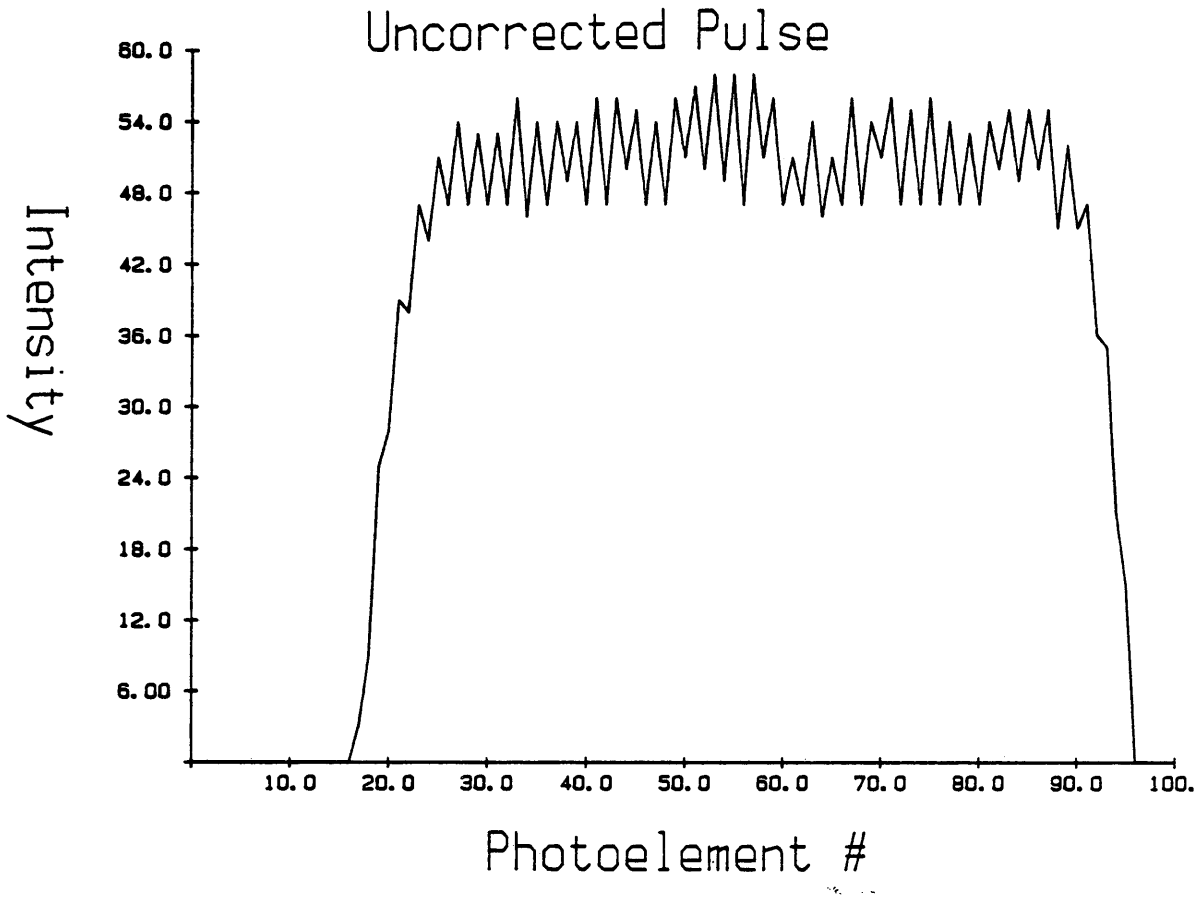

Figure 4. Uncorrected Pulse

The high frequency oscillations (1/3 cycle per data point) are due to the offset between channels. The slower intensity variation is due mostly to the interference pattern.

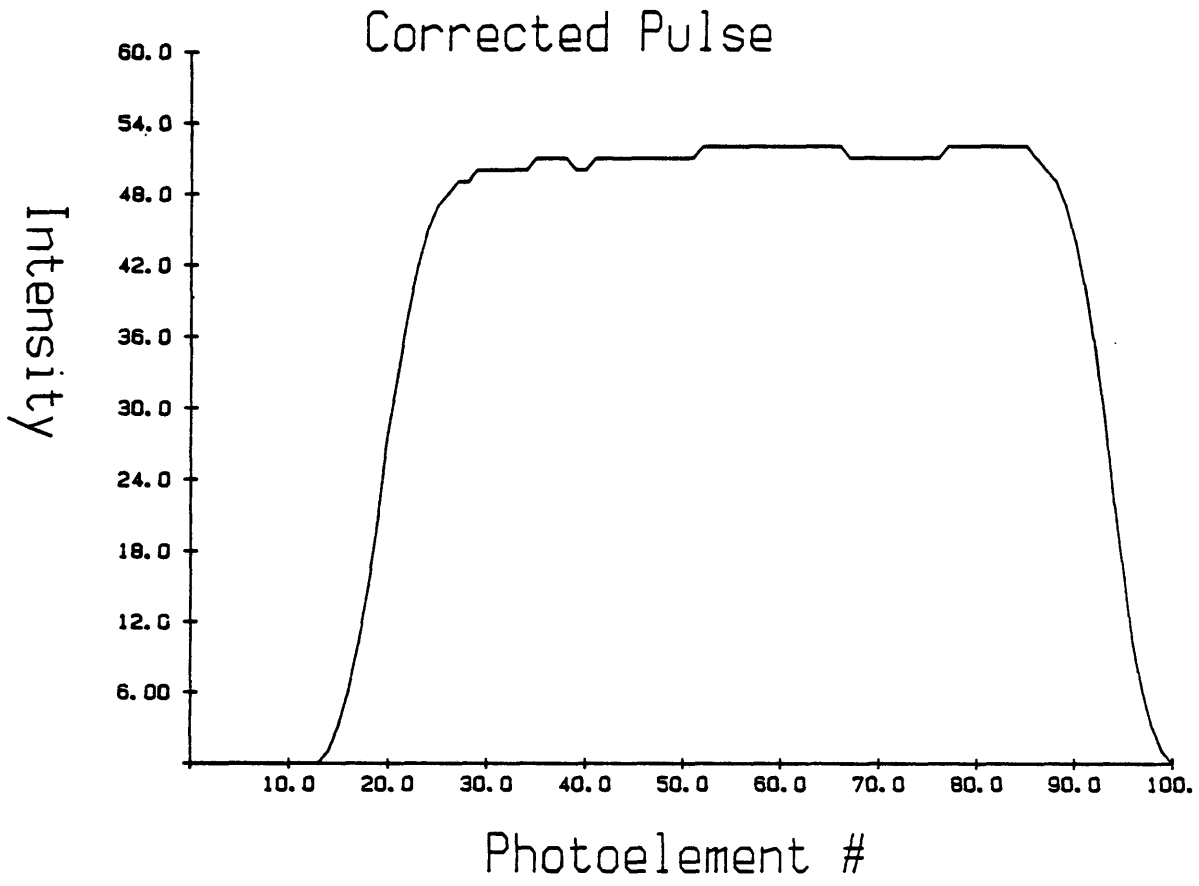

Figure 4. Corrected Pulse

he signal has now been cleaned up sufficiently so that a meaningful analysis can be performed. The interference fringes can never be completely corrected for since the facet-to-facet variation is averaged into the gain coefficients.

The actual coefficients are calculated using the following expression,

$$I^{\mu}_{measured} = I^{\mu}_{actual} * gain^{\mu}$$

where $\mu = 1, \ldots, 1024$.

First, a series of unmodulated scans are analyzed to determine the average intensity. This average intensity is then substituted into the above equation in place of I^{μ}_{actual} to give for the gain,

$$gain^{\mu} = I^{\mu}_{measured} / I_{average}$$

This gain is now divided out of subsequent measurements to yield I^{μ}_{actual}.

An alternate method of completely compensating for the interference effects might be to use an array of photodiodes rather than a CCD array. EG&G Reticon states in their catalogue that the interference effects due to surface electrodes in CCD arrays is not seen with photodiode arrays (5).

If the pulse to be examined is made small enough, the interference fringes are no longer a factor, and the gain

coefficients need not be used. In this case the intensity profile is smoothed by an algorithm which acts as a low pass filter and completely corrects for the odd-even channel offset. Normally, this routine is left on no matter what the duration of the pulse, so that the channel offset is always automatically accounted for.

V. Results

A. Image Fidelity

Image fidelity is a measure of how faithfully the optical system reproduces an object. If the image of the object is perfect, the fidelity Φ_f is unity (6). However, due to diffraction effects and abberations, the image is not a perfect reproduction and the fidelity lies somewhere in the range

$$0 \leq \Phi_f < 1$$

Mathematically Φ_f is defined to be

$$\Phi_f = 1 - (\iint_F (\sigma - I)^2 dxdy)/(\iint_F \sigma^2 dxdy)$$

where $\sigma(x,y) \equiv$ the ideal image intensity and $I(x,y) \equiv$ the measured intensity.

The integration is performed over the working field F of the optical system in question. No assumptions of isoplanatism (spatial-invariance) have been made. The denominator in the second term of the above expression is included for normalization purposes.

Obviously, since the CCD array is linear the above integrals reduce to the one dimensional case

$$\Phi_f = 1 - (\int_F (\sigma - I)^2 dx)/(\int_F \sigma^2 dx)$$

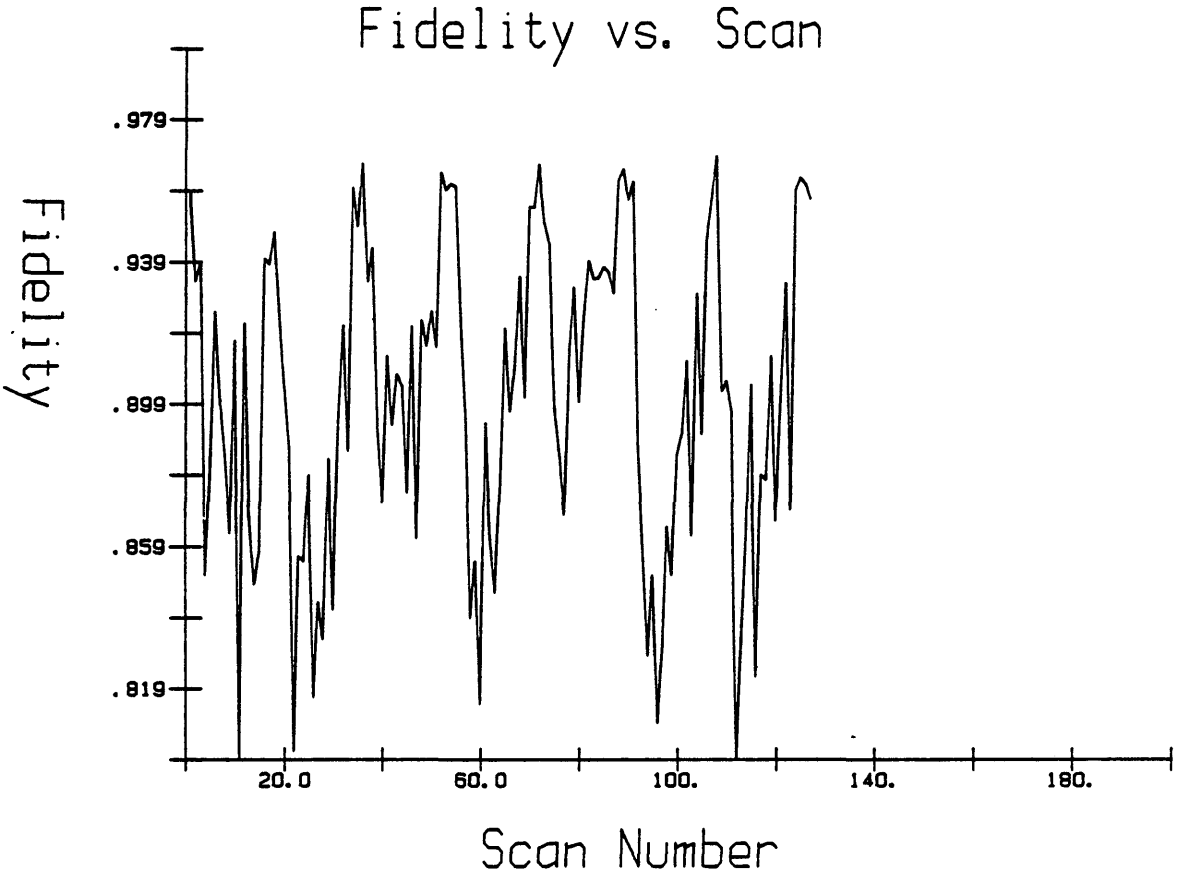

Figure 5. Image Fidelity

The image fidelity for a series of scans is plotted verses scan number in figure 5. The ideal image was a single square pulse. Due to the gaussian nature of the laser beam intensity profile, the actual measured image appeared as a convolution of a square pulse with a gaussian.

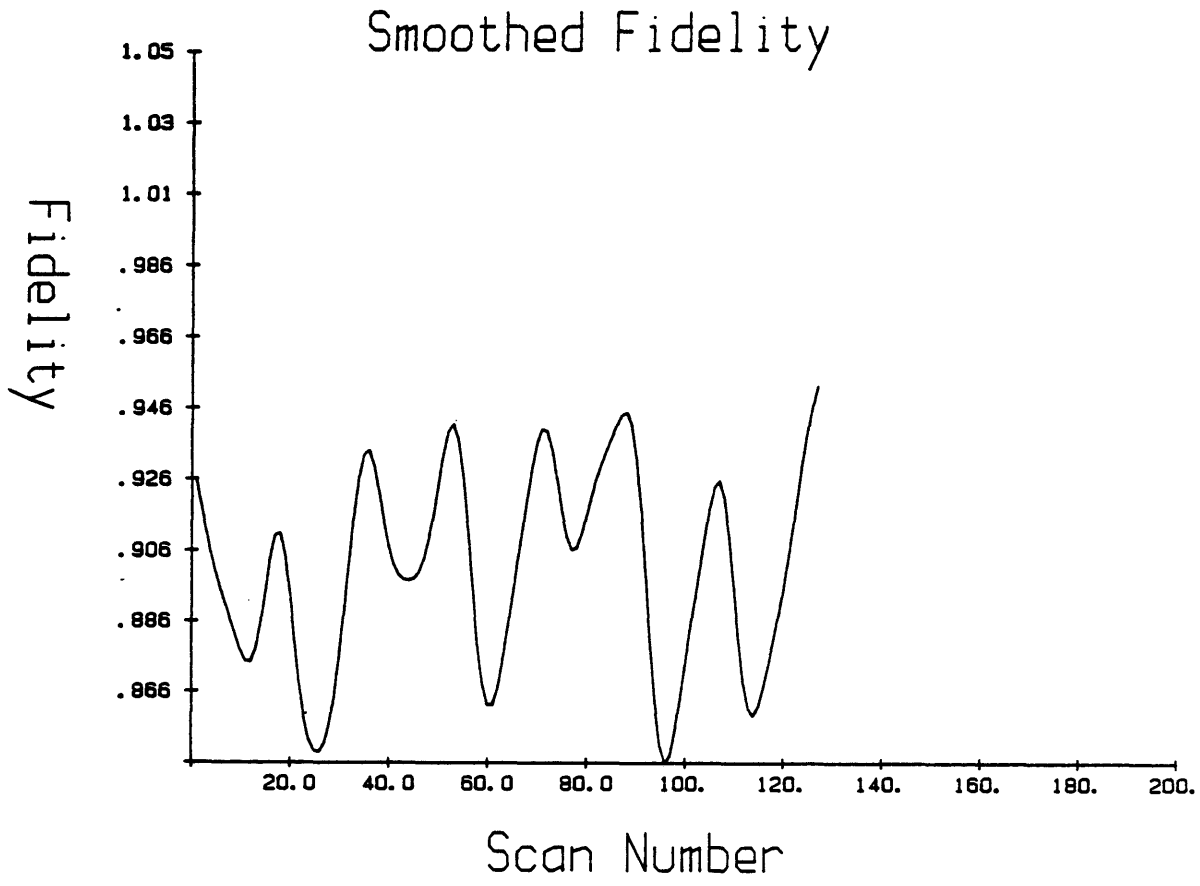

Figure 6. Smoothed Image fidelity.

It is especially interesting to remove the high frequency oscillations from the graph in figure 5. The resulting plot, as illustrated in figure 6, reveals a periodic waveform. The period of the waveform corresponds to exactly one revolution of the polygonal mirror. Thus, using

this particular type of analysis it becomes possible to distinguish each of the individual facets of the mirror.

To calculate the fidelity, information contained in the first scan is used to create an ideal image pulse. The width of the ideal pulse is given by the distance between inflection points of the measured intensity profile. The height of σ is calculated as the maximum intensity reached by I. This ideal pulse is then used in the expression for Φ_f for all consecutive scans.

B. The Optical Transfer Function

The optical transfer function (OTF) provides useful information relating to an optical systems ability to resolve spatial frequencies (7, 8, 9) . A detailed image may be thought of as being composed of a series of spatial frequencies, when appropriately weighted, sum to form this image. This is exactly analogous to the Fourier decomposition of a square pulse.

If an optical system either poorly resolves (or is simply unable to resolve) a given spatial frequency, then information about that object is lost, and the resulting image does not contain all of the structure found in the original object. Clearly then, an important parameter descriptive of lens performance is one which is indicative of how well a spatial frequency is imaged by the optical system. This is the type of information contained in the optical transfer function.

An explicit mathematical form for the OTF is found by relating the intensity distibution in the image plane $I_i(x',y')$, to that in the object plane, $I_o(x,y)$. Assuming a non-coherent, monochromatic source, the image intensity distribution is found by summing over the product of the object intensity and a factor known as the point spread function.

The point spread function accounts for the fact that a perfect point is not imaged as a point, but rather it is is blurred over a small area. For instance, in the case of a diffraction limited system, the image point is an Airy function. Thus, the point spread function is used as a weighting factor to account for diffraction and abberations in the system.

The intensity in the image plane is therefore given by the expression

$$I_i = \iint \delta(x',y';x,y;\lambda) I_o(x,y) dxdy$$

where δ is defined as the point spread function.

If the area of integration is restricted to an isoplanatic region P, then the point spread function can be rewritten as

$$\delta(x',y';x,y;\lambda) = \delta(x' - x,\ y' - y;\lambda)$$

The equation for the image intensity then becomes,

$$I_i(x',y') = \iint \delta(x' - x,\ y' - y\ ;\lambda) I_o(x,y) dxdy$$

Thus, I_i is the convolution of the object intensity with the point spread function.

Taking the Fourier Transform of both sides of the last expression,

$$F\{I_i\} = F\{\delta\} * F\{I_o\}$$

and solving for the Fourier Transform of the point spread function,

$$F\{\delta\} = F\{I_i\} / F\{I_o\} = OTF$$

This last equation, is by definition the optical transfer function.

Since the OTF is in general a complex quantity, it can be rewritten as

$$OTF = MTF * \exp(i\xi)$$

where MTF = the Modulation Transfer Function and ξ = the Phase Transfer Function.

Using the CCD array, both the MTF and ξ can be measured. The MTF (normalized by the zero spatial frequency value) is plotted on the next page for the laser printer used in this investigation.

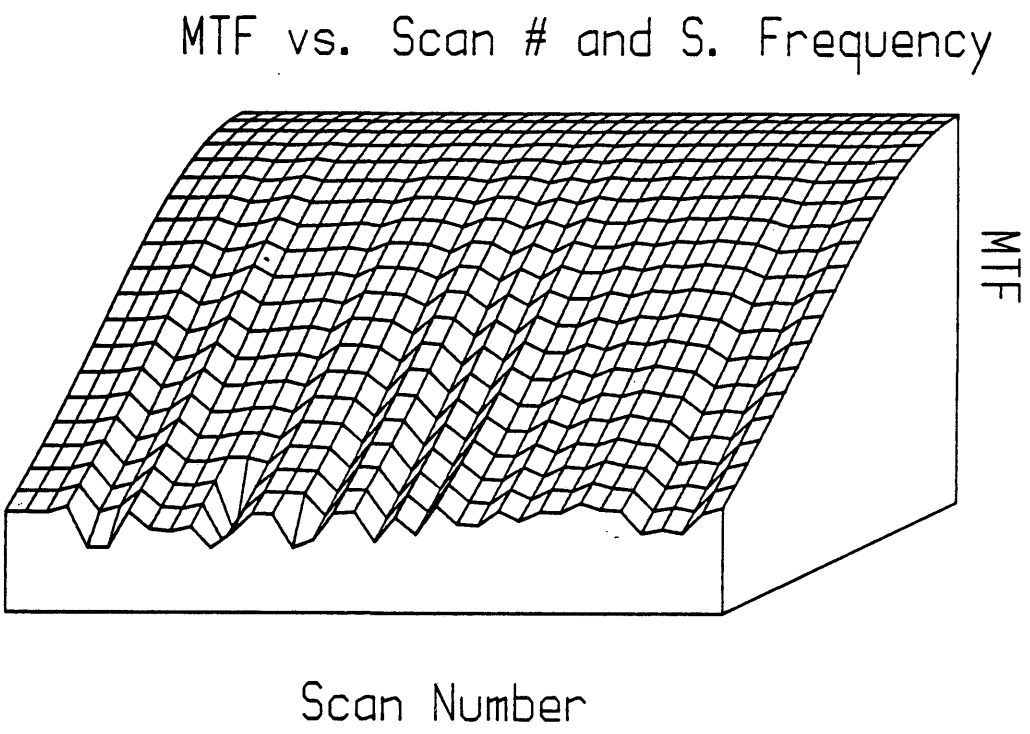

Figure 7. One dimensional MTF for the laser printer.

The one-dimensional MTF is graphed against spatial frequency and scan number producing a three-dimensional plot. This allows easy visual comparison of the MTF from one facet to the next. Shown are two complete revolutions of the polygonal mirror.

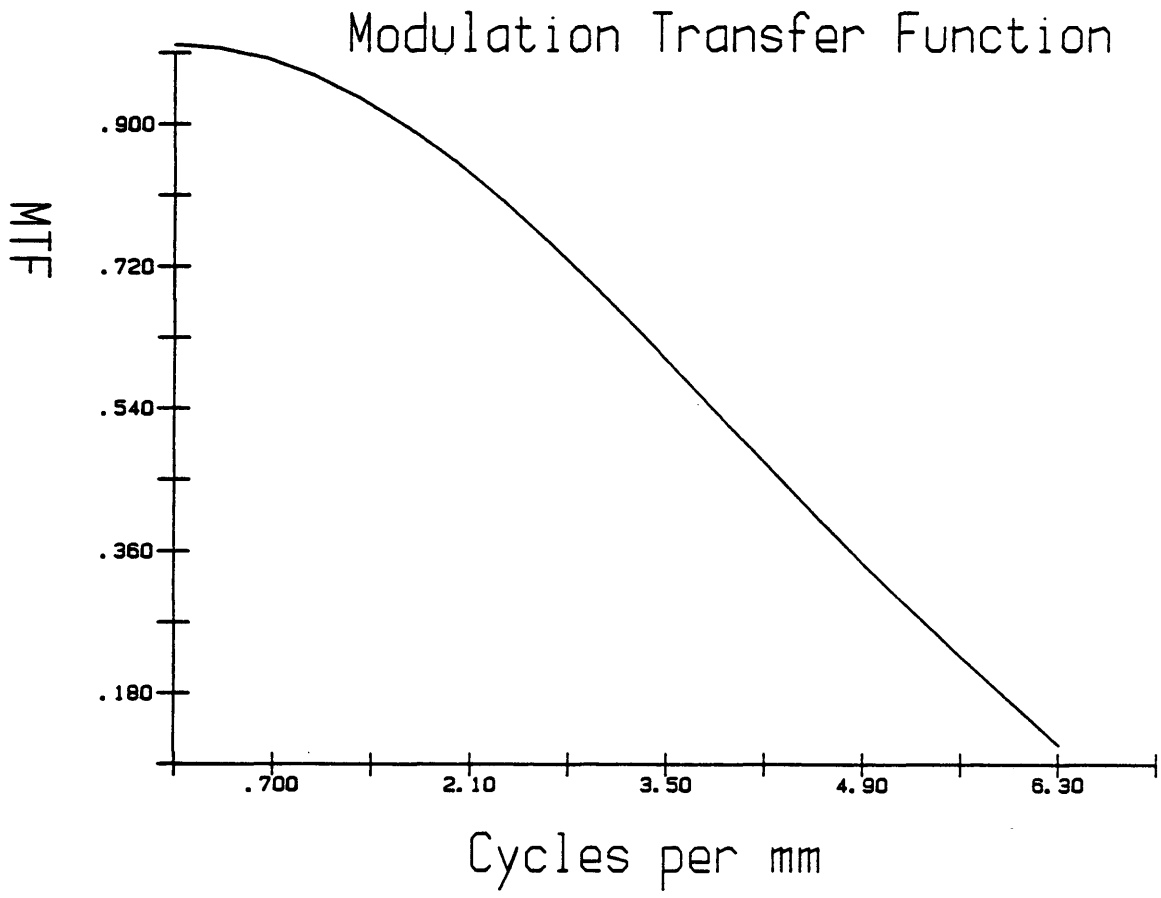

Figure 8. MTF for a single scan.

In figure 8 a single cross-section of the plot in figure 7 is shown.

Unlike more conventional methods of measuring the MTF, where a target with black and white bars of varying spatial frequency is imaged, this analysis was done by turning the laser beam on and off in a realitively short distance (thus writting a "dash" to the detector). The frequency components of this pulse were then compared to those of an ideal square pulse, yielding the MTF for the laser printer.

An alternative method might have been to modulate the beam so as to produce a train of square pulses. It was felt that this manner of measuring the MTF would result in washing out effects of abberations at a particular location. The single square pulse was thus considered the most appropriate for carrying out the analysis.

The Nyquist frequency for the system is 38.5 mm^{-1} which turns out to be more then adequate for the analysis of the laser printer.

It should be noted that because this is a sampled image system it is possible to violate the assumption of isoplanatism (10). For example, a sufficiently small target might activate more photoelements in one position than another. The reconstructed profile would then appear to be different at both positions. This can be avoided by making sure that the ratio of pulse length to photoelement size is fairly large.

Should this ratio become small it is still possible to calculate the MTF but a more sophisticated analysis is required.

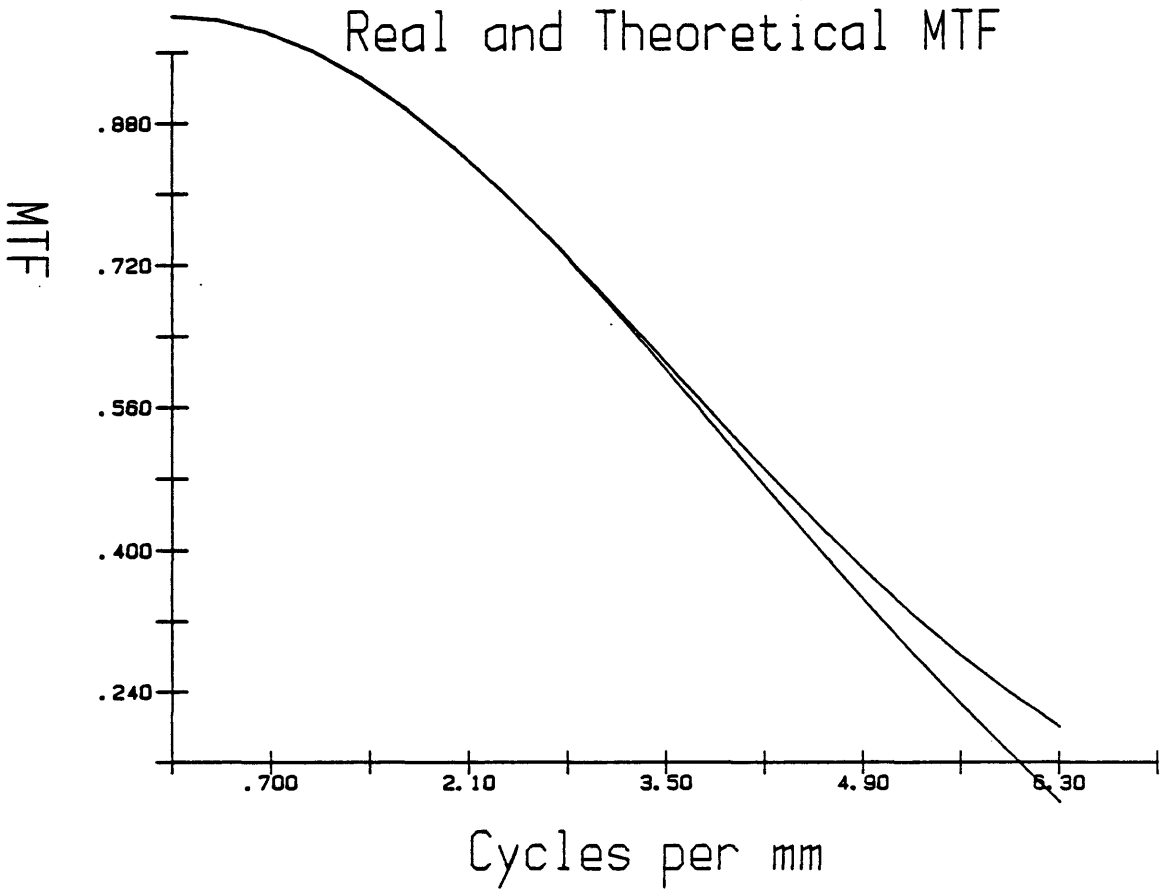

Figure 9. Theoretical and measured MTF comparison.

The theoretical MTF for the system was calculated, and in the graph shown above, compared to the measured MTF. (Theoretical values all lie on the upper curve.) There is good agreement at low spatial frequencies, the curves deviating at higher values.

Finally, the shift in position of the pulse can be measured by comparing the centroid position from one scan to the next, which is simply a measure of the phase transfer function. However, it should be noted that this measurement of the PTF provides only a relative and not absolute value for ξ. These measurements are described in the next section.

C. Pel Placement

With the laser printer programmed to produce a single pel, the placement of this pel from scan-to-scan can be measured. Each acquired scan is fit (Gauss-Newton method) to a gaussian distribution of the form

$$\text{Intensity} = B * \exp\{-C(x-A)^2\}$$

where A is the position of the centroid, B the maximum intensity, and C the gaussian modulus.

In order to determine the accuracy in position measurement, a stationary laser beam was directed on the chip. The width (at the 50 % points) of the intensity profile for this beam was adjusted to equal that of the laser printhead's as close as possible. The image of the stationary beam was captured at the same rates as that of the scanned beam. From this analysis, it was found that there is a $\pm$.59µm variability in the position of the centroid.

The reason for this error has not been completely isolated. One of the possible sources contributing to this error might be noise in the laser. It was noticed that as the laser warmed up, the error in centroid position slowly increased, and finally leveled off at the stated value.

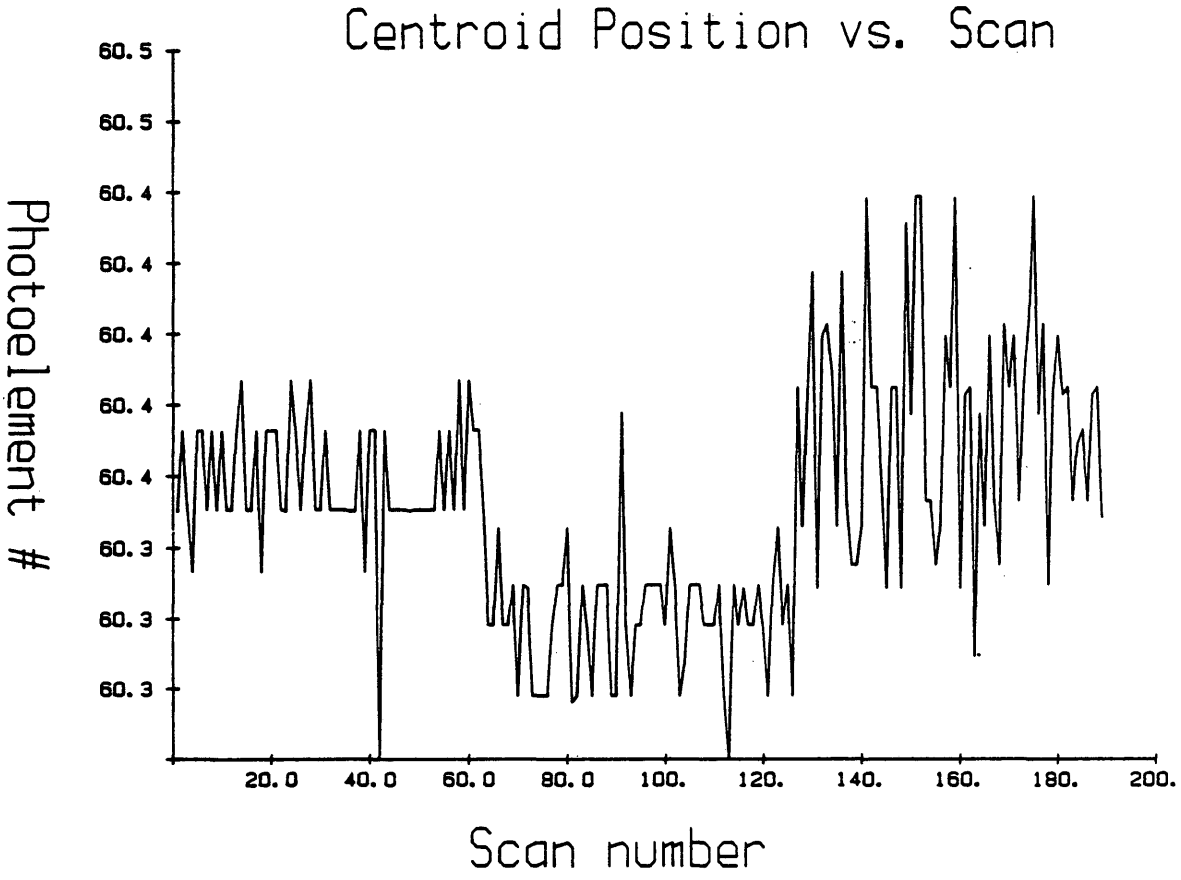

Figure 10. Centroid position as laser warms up.

Figure 10, shows the stationary beam position over a period of approximately six minutes. The data was taken in three separate intervals, two minutes between each acquisition. Notice how the position becomes less certain with each succesive acquisition.

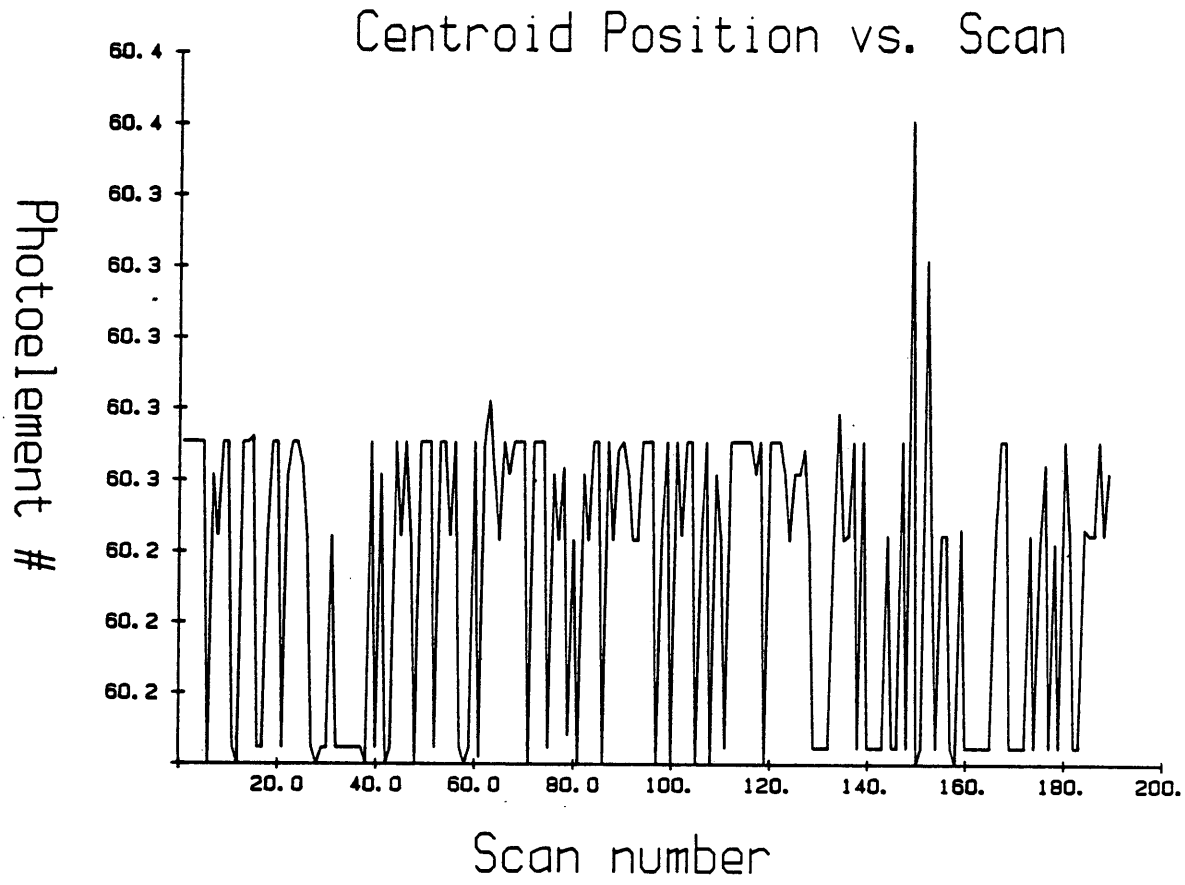

Figure 11. Centroid position calibration plot.

In figure 11, the laser has reached some sort of equilibrium, with the beam position wandering within a specific interval. Excluding a few spurious points, the range is .09 photoelements from maximum to minimum. This is used as the uncertainty in all subsequent measurements. Should laser noise be a major contributor, the actual error in centroid position may be much smaller.

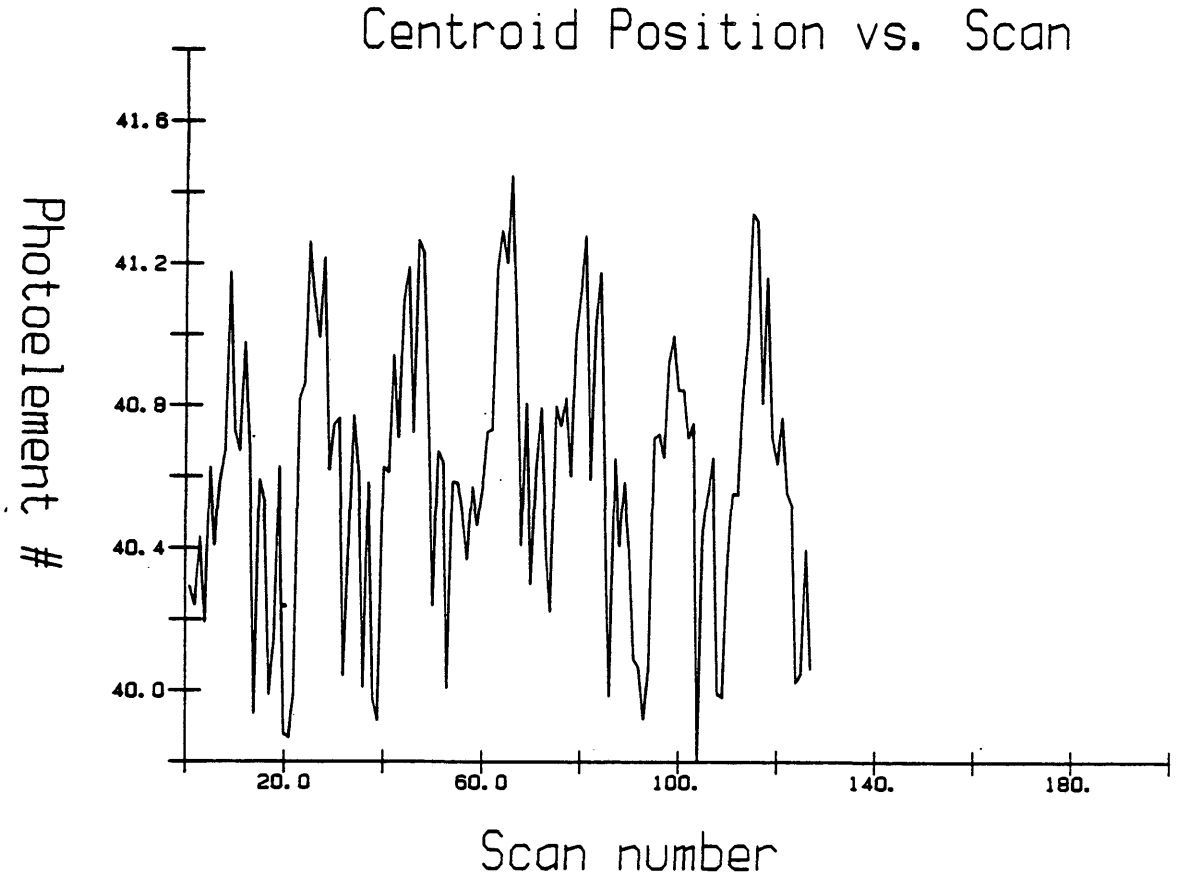

Figure 12. Centroid position for individual scans.

Figure 12 shows the change in centroid position for a series of scans taken from the laser printer. A periodic waveform is noticeable once again. Figure 13 replots the data in figure 12, but with the higher frequency oscillations stripped from the waveform.

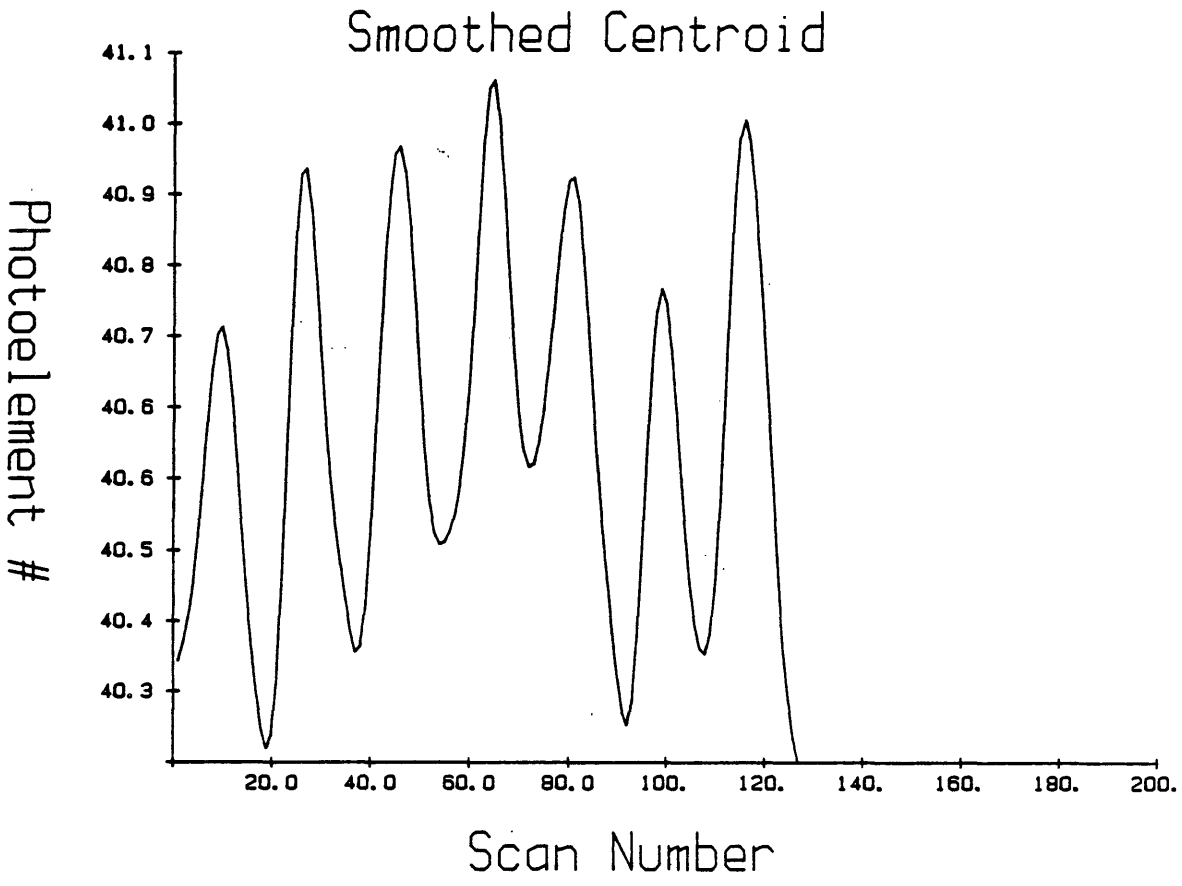

Figure 13. Smoothed centroid position.

Counting the number of points in one period reveals that it corresponds to one revolution of the mirror. Since the scan number is related to time, a Fourier analysis can be performed on the data, confirming that there is indeed a frequency component at the rate of mirror revolution.

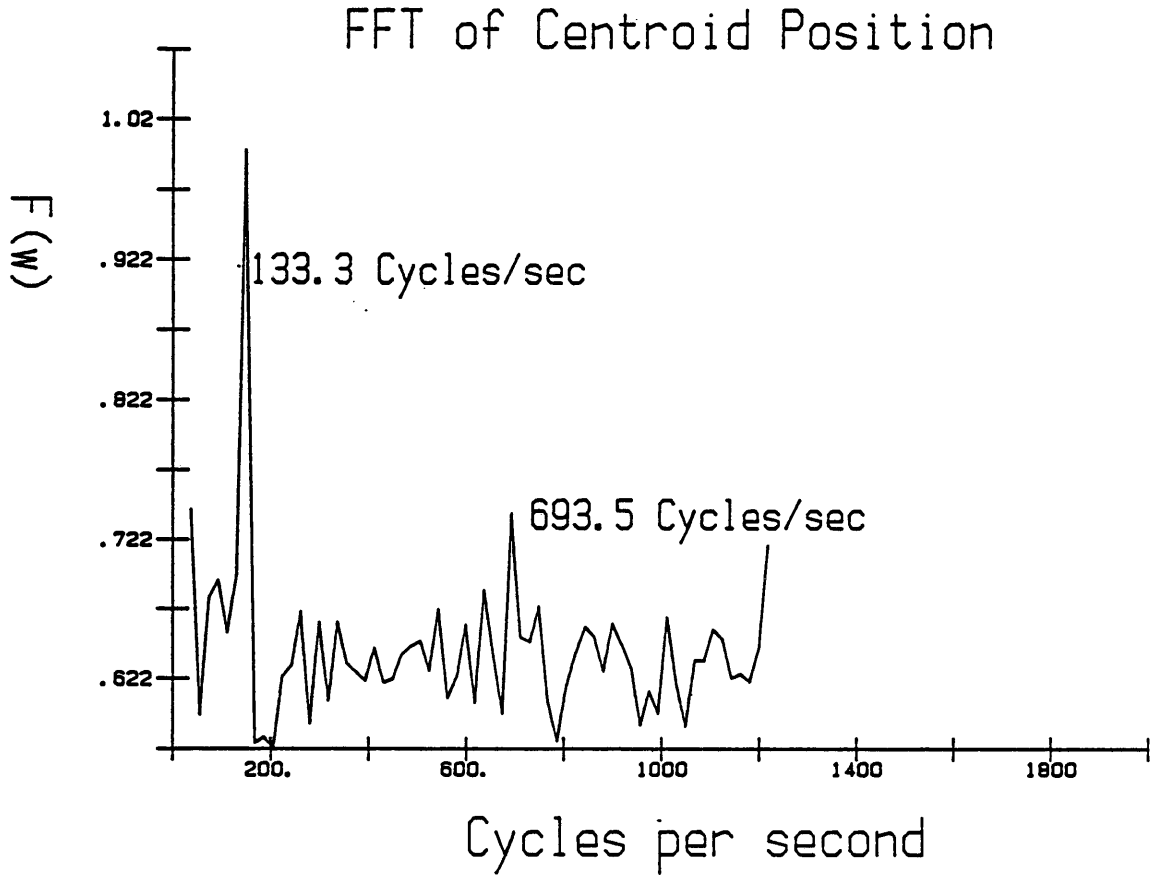

Figure 14. FFT of centroid positon.

The FFT shows two noticeable peaks. One at 133.3 cycles per second (which is the rate at which the mirror is spinning) and one at 693 cycles per second, or approximately five times the first. The reason for the second peak is unknown.

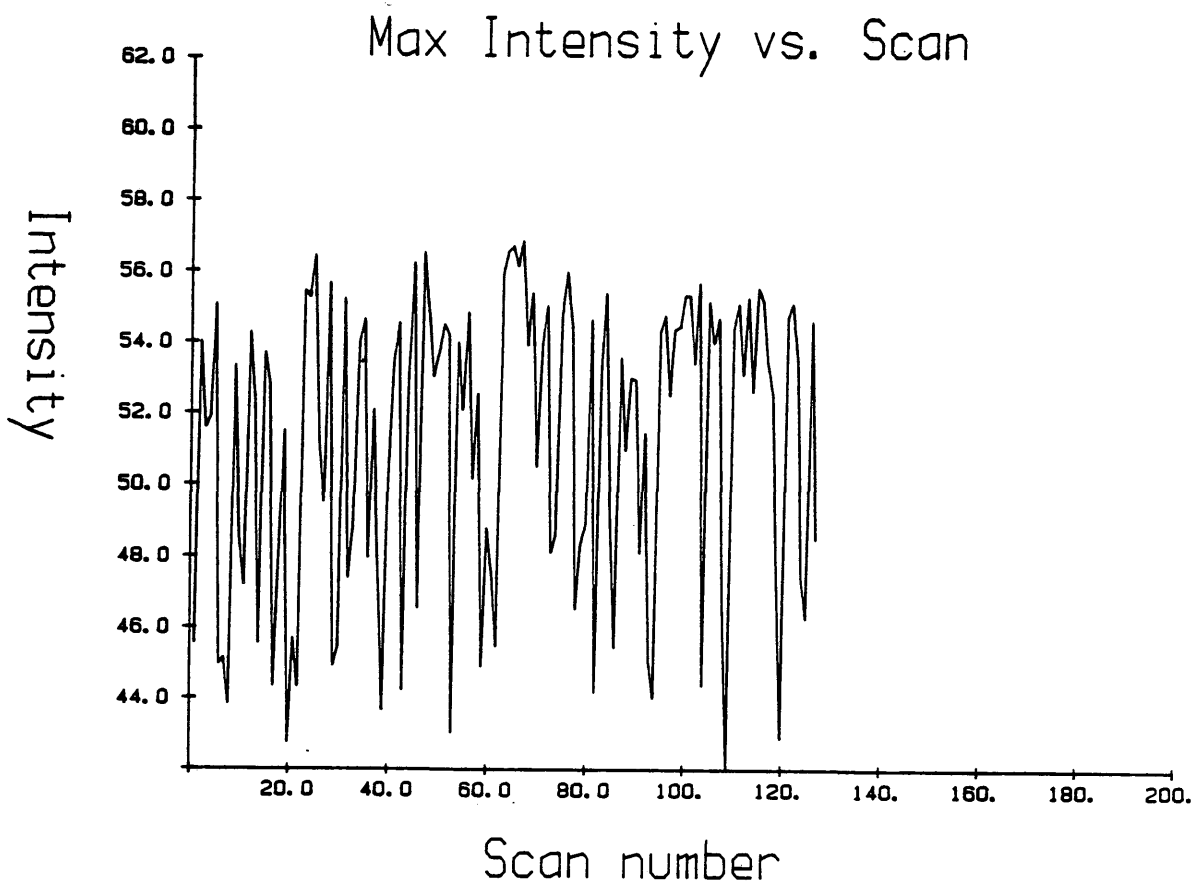

Figure 15. Change in intensity with scan.

Since a curve fitting routine is used, the change in peak intensity and waist size (FWHM) from scan-to-scan is available also. For the run shown in the centroid plots these results are graphed on this page and the following page.

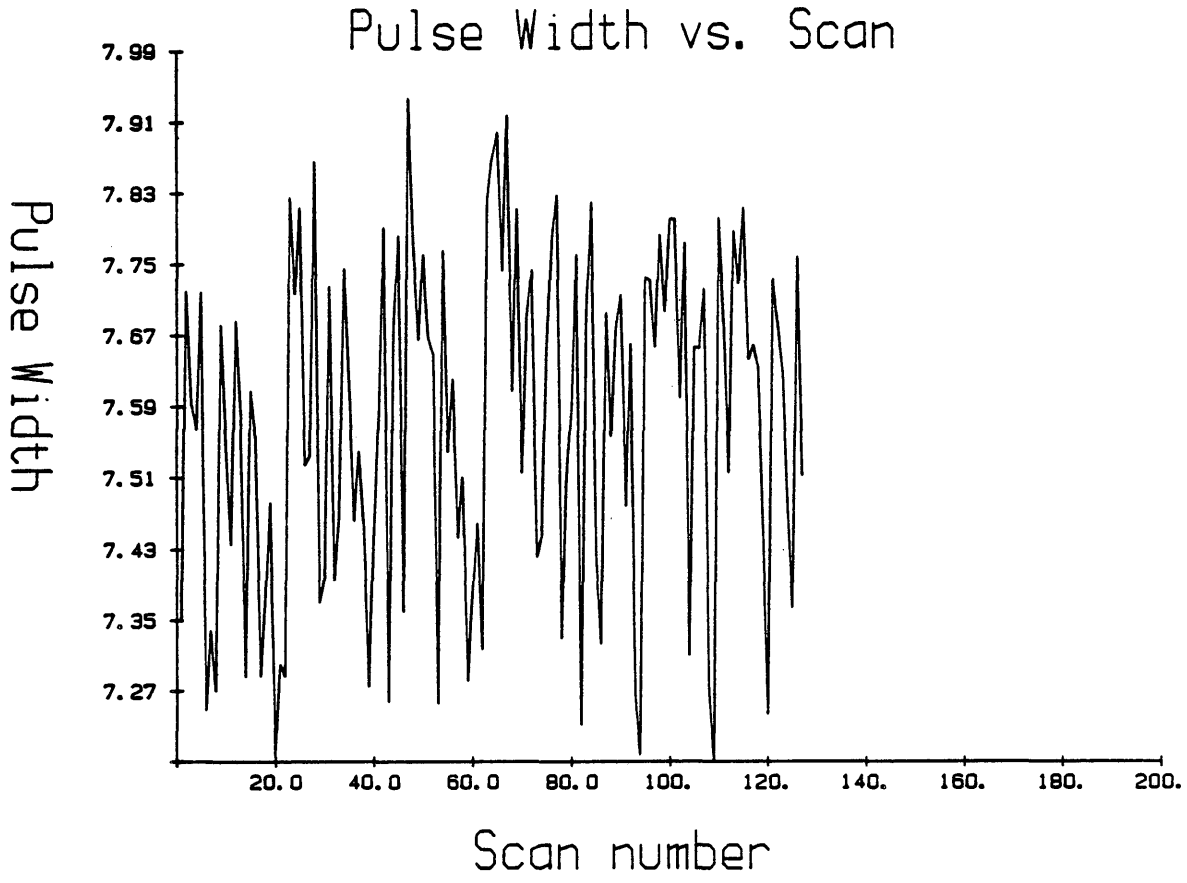

Figure 16. Change in waist size with scan.

A fourier transform of these plots reveals no distinctive frequency components, simply random peaks. (Note: the pulse width is given in units of photoelements.)

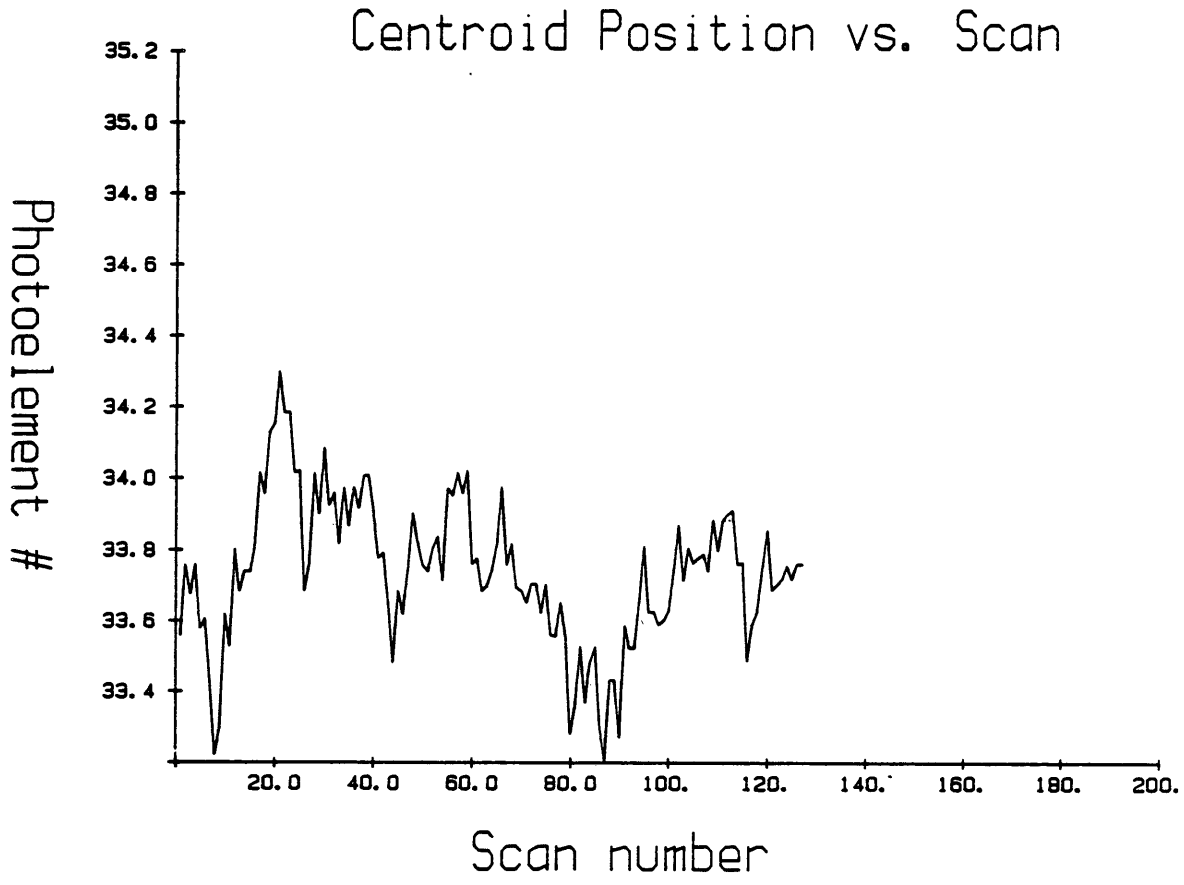

Figure 17. Centroid position in process direction.

The beam profile can also be captured in the process direction. Figure 17 illustrates how the centroid of the profile changes in position along this axis.

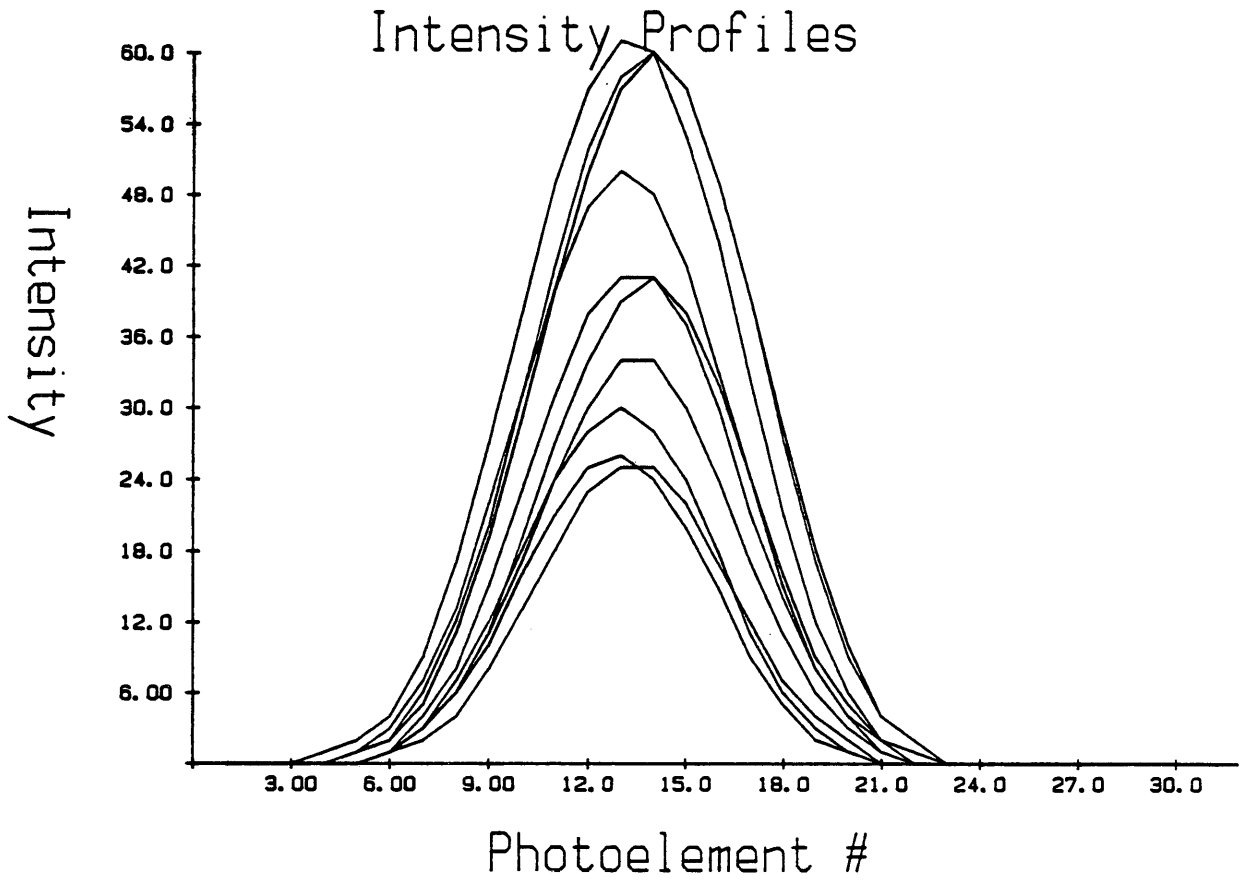

Figure 18. Typical series of intensity profiles.

To provide a more physical picture, figure 18 is included to show how the actual profile changes from scan-to-scan. Shown are 10 consecutive scans. Note how the position of the centroid changes, and how the peak intensity fluctuates.

D. Spatial Pulse Width

The final measurement concerns the width of a pulse. With the printer programmed to turn on and off within a specified interval, the spatial pulse width as measured from the points of inflection of the pulse can be determined. The results for a series of consecutive scans is presented in figure 19.

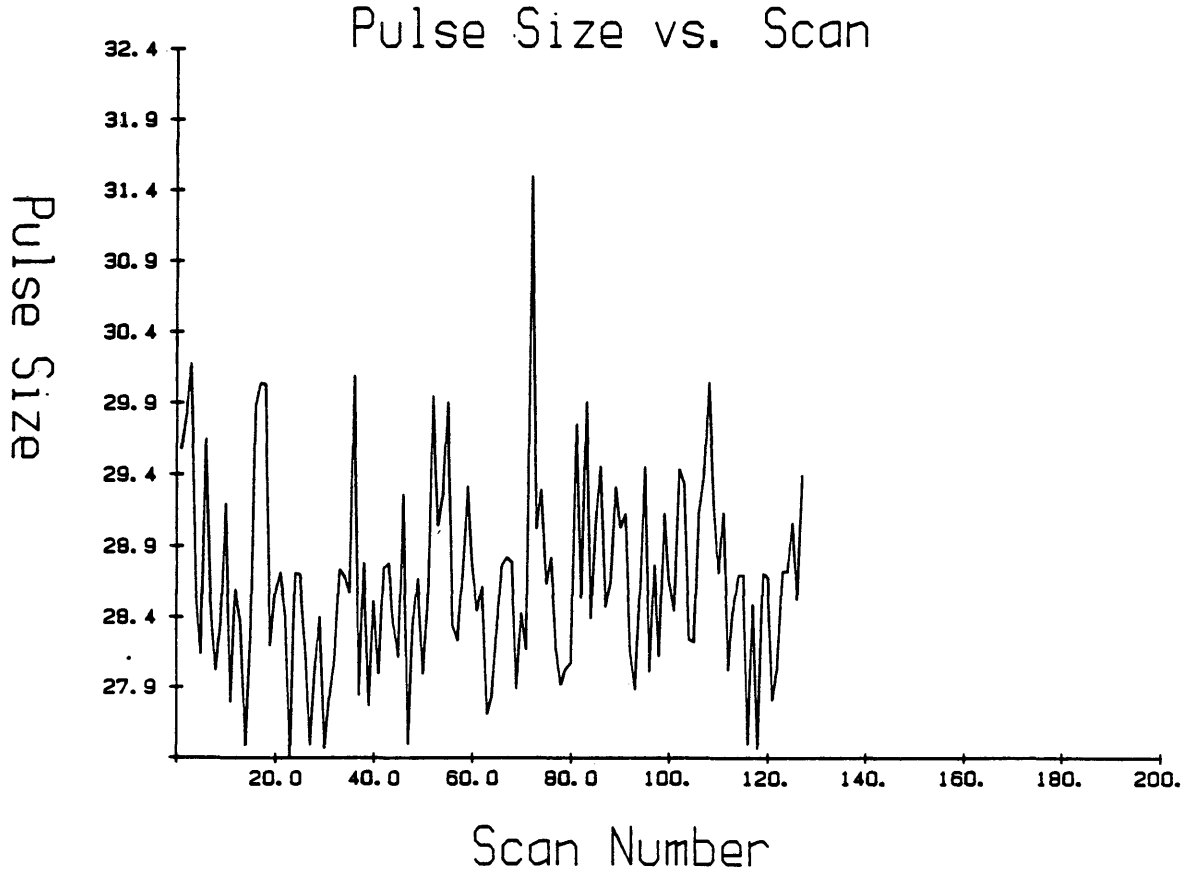

Figure 19. Spatial Pulse Width

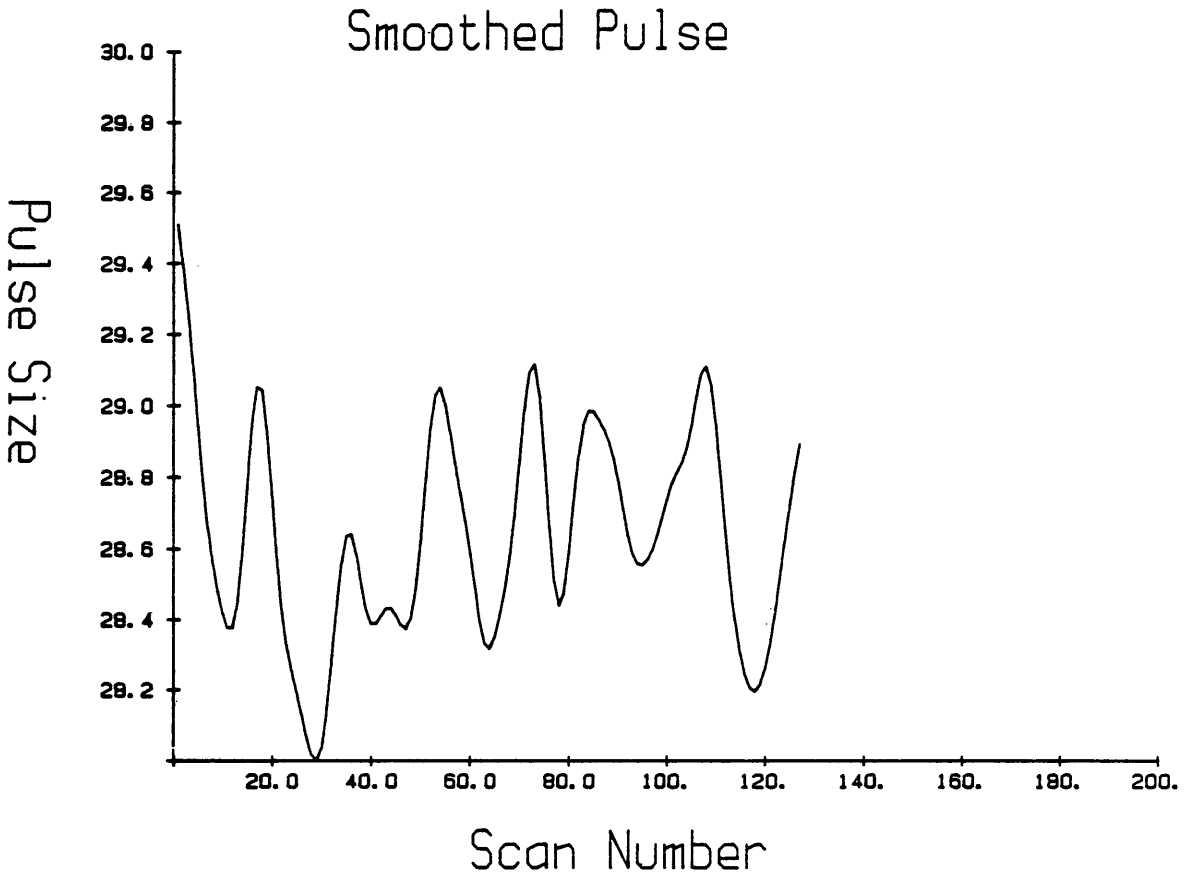

Figure 20. Smoothed spatial pulse width.

Removing the high frequency oscillations, a waveform of period 18 is revealed. Once again, it is indicative of the individual facets. (Note: the units of the pulse size are photoelements.)

VI. Exposure Predictions

Using the aforementioned techniques the intensity profile can be characterized for the entire scan distance. This information can be used to predict how the photoconductive surface of the drum would be exposed, and thus, what the resulting print should look like. Following the development of Sonnenburg (11), the measured profile is an elliptical gaussian given by,

$$I_{measured} = I_{peak}\exp\{-(\alpha^2 x^2 + \beta^2 y^2)\}$$

where the x-axis is oriented along the scan direction and the y-axis in the process direction. The peak intensity, I_{peak}, and moduli, α, and β, are all measured parameters.

The average scan velocity v_s, can be measured, leaving the rate at which the photoconductive drum rotates as the only unknown.

If the distance between scans (peak - to - peak in the process direction) is denoted by p, then the amount of energy per unit area deposited at point x_o, y_o by scan n is given by,

$$E_n(x_o,y_o) = \int I_{peak}\exp\{-(\alpha^2(x_o - np)^2 - \beta^2(y_o - y)^2\}dt$$

where the limits of integration are from t=0 to t = t_o (t_o being the time it takes to complete one scan).
This last expression can be simplified to read,

$$E_n(x_o,y_o) = -I_{peak}\exp\{-\alpha^2(x_o - np)^2\}\int\exp(-\beta^2y^2)/v_s dy$$

where l is the length of the scan, and the limits of integration are, $y=y_o$ to $y = y_o - l$.

Approximating the integral by allowing the limits to approach $\pm\infty$,

$$E_n \cong I_{peak}\exp\{-\alpha^2(x_o - np)^2\}\pi^{1/2}/(v_s\beta) \ .$$

If the speed at which the photoconductive surface crosses the scan plane is denoted by v_p, the above equation can be rewritten to read,

$$E_n = I_{peak}\exp\{-\alpha^2(x_o - np)^2\}p\pi^{1/2}/(\beta l v_p) \ .$$

The total exposure at x_o, y_o due to all scans is then,

$$E = \Sigma I_{peak}\exp\{-\alpha^2(x_o - np)^2\}\pi^{1/2}p/(\beta l v_p)$$

where the summation runs over $n = \pm\infty$.

This function now provides the exposure at any point on the photoconductive surface as a function of drum velocity. Consequently, it can be used to predict the shape of lines or pixels, and help determine optimum operating conditions.

VII. Conclusion

A close examination of the results would seem to indicate that the classical calculations of the MTF are not as appropriate in determining how well the printhead performs as are the measurements of pel placement and spatial pulse width. Of foremost importance is the intensity distribution on the surface of the photoconductor. Pel placement and spatial pulse width measurements are not only indicative of this distribution, but they can also be use to readily predict how a series of pixels, dashes, or lines in either the scan or process directions will appear.

On the other hand, the results of an MTF analysis are not as easily interpreted. By itself, it is difficult to relate how a particular value of the MTF for a given spatial frequency will affect the output of the system. The potential usefulness of the MTF lies in measuring the function for many systems and developing a sufficient database so that the relation between system performance and the MTF becomes more transparent.

It was hoped that the measurements of the OTF would allow the calculation of wave aberrations . Unfortunately, the signal-to-noise ratio is too low and the aberration retrieval algorithm too sensitve to noise to make any realistic recovery of the wave aberration conceivable (12).

Several new schemes have been proposed to improve on the accuracy and the signal-to-noise ratio . The first of these improvements would involve the recalibration of the centroid position of the stationary laser beam. An alternative method of positioning the beam on the array is being investigated which would reduce any movement of the beam due to laser noise to a second order effect. A second improvement would be to increase the gain of the signal from the array. Currently the output runs from ground to approximately 300mV. This should be increased to 512mV so that the signal spans the entire range of the digitizing window.

Another method of enhancing the system performance has already been mentioned; using a photodiode array in place of the current CCD device. However, even if the interference problems are overcome, another obstacle still remains. For example, the only photodiode arrays (of equivalent number of photoelements) capable of providing the necessary data rates are multi-channel devices. (Multi-channel meaning, rather than a single analog output for the array, there are several outputs.) Thus, many digitizers would be required, dramatically increasing the cost of the system. (An alternative might be to use a multiple track recording system. By storing the signals on magnetic tape, it is no longer necessary to have multiple high speed transient recorders, since the signal can now be

digitized at leisure.) Smaller arrays could be used, reducing the number of channels, but the photoelement size is in general larger. Even so, the smaller arrays probably offer the best possible performance. Since they have fewer elements, it takes less time to clock through the entire array so that slower rates are possible, increasing the signal-to-noise ratio and the effective bit accuracy.

Other improvements to the system do not necessairily involve increasing the signal to noise ratio, but rather, allow for easier adjustment. For instance, the array (and its rotational and translational stages) could be mounted on some type of rail system which would permit the device to be positioned anywhere in the scan distance with minimal adjustments. Additional software automating various aspects of the device (e.g., positioning, locating the beam on the array, etc.) would also help to reduce operating time.

Finally, it should be noted that while the measurements in this report were made on a laser printer, the system is quite capable of capturing the intensity profile produced by other types of instruments. A careful analysis of the different profiles might provide some useful insight as to which technology provides the optimum performance.

References Cited

1. L. Beiser, "Laser Scanning and Recording: Developments and Trends", Laser-Focus, February 1985.

2. R. Sprague, J. Urbach, T. Fisli, "Advancesin Laser and E-O Printing Technology", Laser-Focus, October 1983.

3. R. Webb, "Optics for Laser Rasters", Applied Optics, Vol. 23, No. 20, 15 October 1984.

4. T. Matsuda, F. Abe, and H. Takahashi, "Laser Printer Scanning System with a Parabolic Mirror", Applied Optics, Vol. 17, No. 6, 15 March 1978.

5. EG&G Reticon, Image Sensing Products, Catalogue, p. 69, 1985.

6. E. Linfoot, Fourier Methods in Optical Image Evaluation, Focal Press, 1960.

7. E. Hecht and A. Zajac, Optics, Addison-Wesley Publishing Company, pp.397-422, 1974.

8. W. Smith, Modern Optical Engineering, McGraw-Hill Book Company, pp.290-325, 1966.

9. F. Smith, "Optical Image Evaluation and the Transfer Function", Applied Optics, Vol.23, No.15, 1 August 1984.

10. S. Park, R. Schowengerdt, M. Kaczynski, "Modulation Transfer Function for Sampled Image Systems", Applied Optics, Vol. 23, No. 15, 1 August 1984.

11. H. Sonnenburg, "Laser-Scanning Parameters and Latitudes in Laser Xerography", Applied Optics, Vol. 21, No. 10, 15 May 1982.

12. J. Maeds and K. Murata, "Retrieval of Wave Aberration from Point Spread Function or Optical Transfer Function Data", Applied Optics, Vol.20, No.2, 15 January 1981.

Appendix A -- Software.

This section descibes the software used in the data acquisition and analysis. The entire program is written in ASYST. A dictionary is provided which describes how each word operates, what inputs are required, and what values are returned to the stack (if any). At the end of the dictionary the program is listed. The format follows that of the ASYST manuals, except for the asterisk ranking system which is defined to be:

* --- Word cannot be executed by itself. It is used internally in the program only.

** -- Word can be executed by itself, but its primary purpose is internal to the program. May require certain arrays to have data loaded into them prior to execution of this command.

*** - Always executed by itself.

Examples on how to use the word are included where appropriate, and are in general, confined to *** words.

It is recommended that the user read the ASYST -- A Brief Introduction to Asyst manual at the very least before attempting to use any of the following commands.

At this point it is also appropriate to comment on how the data files which are acquired using the

DATA.ACQUISITION.AND.STORAGE command are structured. For any given acquisition there will be n subfiles in the file. The first n-1 subfiles contain the intensity profiles from n-1 scans. The length of the subfile depends on how the user has programmed the CCD array. They can range from 64 to 1024 elements long.

The n^{th} subfile is always 5 elements long. This is the status array. This is the first subfile read when any of the programs which do analysis are executed. The five elements are read into an array named STATUS. The first element is the run number. This can be any number the user wishes. Its primary purpose is to provide a "tag" so that different runs can be identified. The second element is the number of photoelements that were used. This will range from 64 to 1024 in blocks of 64. The number of elements in the n-1 subfiles will always equal this number. The third element in the STATUS array specifies the amount of memory the TR8818 was programmed to use. In the fourth element is stored a number which determines how long the laser beam was turned on in the scan. The final element of STATUS stores how far from the waist of the output beam the CCD was when the data was acquired.

If the structure of a particular file has been forgotten, it can be printed out on the computer screen by using the following series of commands:

```
OK FILE.OPEN  (filename)
OK ?FILE.TEMPLATE
```

After issuing the ?FILE.TEMPLATE command the computer prints out the structure of that particular file.

In the examples that follow the data is acquired in the file TEST19.DAT . Any 8 letter filename with a three letter extension could have been used.

File: ANALYSIS.IBM

GAUSSIAN [X -- F(X)] **

A data array is required on the stack before execution. GAUSSIAN evaluates the function $B*\exp\{-C(x - A)^2\}$ at each point of the array and returns these values in an array of the same dimension to the top of the stack. The values for A, B, and C are provided in the word PARAMETERS.FOR.GAUSSIAN, or can be defined by the user.

FFT.OF.MEASURED.INTENSITY [--] **

Finds the FFT for the measured intensity profile which must be read into the array INTENSITY prior to execution. The FFT is stored in the array FFT.INTENSITY.

FFT.OF.IDEAL.INTENSITY [--] **

Calculates the FFT of an ideal square pulse. The Ideal FFT is stored in the array FFT.IDEAL.INTENSITY . The size of the ideal pulse is taken from the fourth element in the status array. The height of the ideal pulse is arbitrarily set at 63. This should be changed if the pulses being measured are noticeably higher.

MTF.CALCULATION [--] **

Calculates the MTF using FFT.INTENSITY and FFT.OF.IDEAL.INTENSITY . Consequently, the words FFT.OF.MEASURED.INTENSITY and FFT.OF.IDEAL.INTENSITY must be executed prior to this instruction. The MTF is stored in the array MTF.

RETRIEVE.DATA [--] *

The status array occuring in the data file named by "FILENAME1 is read and stored in the array STATUS.

PULSE.ONE [--] *

Fits a quartic polynomial to the first edge of the captured intensity profile. Stores the edge of the profile in array SUB.INTENSITY.ONE .

PULSE.TWO [--] *

Fits a quartic polynomial to the trailing edge of the captured profile. Stores the edge in the array SUB.INTENSITY.TWO.

INFLECTION.POINT.ONE [--] *

Finds the inflection point from the curve fit of the PULSE.ONE word. Stores this value in the scalar P1.

INFLECTION.POINT.TWO [--] *

Finds the inflection point for the curve found in PULSE.TWO . This value is stored in the array P2.

CALCULATE.PULSE.SIZE [--] ***

Upon execution of this word, the computer will prompt the user for the name of the data file which stores the intensity profile measurements. It then calculates the spatial pulse width and stores these values in a new file. This word MUST be used prior to the MTF.ANALYSIS word.

Example:

```
OK CALCULATE.PULSE.SIZE
Name of data file please:TEST19.DAT
Name pulse size file please: PTEST19.DAT
Did you input the requested parameters correctly (Y/N)?Y
pulse size:27.89
pulse size:28.93
```

The data file where the measured intensity distributions are stored is in TEST19.DAT . The new file which will be created is called PTEST19.DAT, and will contain the pulse size for each scan. (Note: the ". . ." at the end of the example simply signify that the program keeps running at this point

until the pulse size has been calculated for all the scans in the file TEST19.DAT.) The format of the new file is one subfile for each scan, 5 elements in each subfile. The pulse size is in the fourth element. The other elements in the subfile are the same as those in the STATUS array.

CHECK.DATA [--] *

This can be used only inside other words. It is used to prompt the user to insure that all requested information has been properly entered.

CREATE.MTF.FILE [--] *

This word is employed inside the MTF.ANALYSIS routine. It creates a file in which the MTF's for successive scans can be stored.

MTF.ANALYSIS [--] ***

Calculates the MTF for a series of scans and stores the MTF's in the file named by the second prompt. CALCULATE.PULSE.SIZE must be executed prior to using this word.

Example:

OK MTF.ANALYSIS

Name of data file please:TEST19.DAT

Name of MTF file:MTF19.DAT

OK

The data file is TEST19.DAT . The new file to be created will be called MTF19.DAT and will contain the MTF for each scan in a separate subfile. CAUTION: The number of spatial frequencies that come out of the calculation is dependent on the size that the CCD array was dimensioned to. For best results with this particular word it is recommended that the array size not go below 128.

3D.MTF.PLOT [--] ***

Creates a three-dimensional plot of the MTF. It prompts the user for the name of the MTF file, and the number of scans that the user wants plotted. The MTF.ANALYSIS word must be used prior to this command.

OK 3D.MTF.PLOT

Enter filename please:MTF19.DAT

Enter number of scans to be in plot:36

Plotter or graphics display (1 or 2):1

OK

Notice the filename entered is that created by the MTF.ANALYSIS word. The number of scans to be plotted is 36. The output will be going to a HP7470 plotter. Had a 1 been selected, the output would have gone to the screen of the computer.

GAUSSIAN.PARAMETERS [--] *

Defines the parameters A, B, and C used by the word GAUSSIAN.

CREATE.GAUSSIAN.FILE [--] *

Used internally by the word PEL.FIT.TO.GAUSSIAN only. Creates a file in which the parameters A, B, and C (see GAUSSIAN.PARAMETERS) are stored for each individual scan.

SEARCH.FOR.MAX.INTENSITY [--] *

Finds the maximum value of the intensity, and its corresponding index in the array INTENSITY. The maximum value is stored in the scalar MAX.INTENSITY, its location in the array is stored in COUNTER1.

PEL.FIT.TO.GAUSSIAN [--] ***

Prompts the user for the name of the data file in which the intensity measurements are stored. The user is also asked to supply a name for the file in which the parameters

A, B, and C for each scan will be stored. The program then proceeds to fit each scan in the file to a gaussian curve.

Example:

```
OK PEL.FIT.TO.GAUSSIAN
Name of data file please:TEST19.DAT
Name of Gaussian file:GTEST19.DAT
Calculating scan:1
Parameters for the scan are:29.90  63.34 .05
```

Once again the name of the data file TEST19.DAT is entered. The file to be created is named GTEST19.DAT. The structure of this file is such that for every scan, an array of dimension 3 is written to the file. The first element is A, then B, and finally C. The values are written to the screen as they are calculated and in the same order. CAUTION: This program can only be used on data which is gaussian in nature. Consequently, use on files which have only a single pel (of the laser printer) for the intensity profile.

PLOT.PEL.DATA [--] ***

The data file created by the word PEL.FIT.TO.GAUSSIAN is

first asked for. The user then is asked to chose the desired plot. WARNING: Make sure the array PEL.POSITION is dimensioned so that the number of array elements equals the number of subfiles found in the current file.

Example:

```
OK PLOT.PEL.DATA
Enter file name please:GTEST19.DAT
Type : 1 - for pel position 2 - for intensity change
3 - for pulse width: 1
Plotter or Graphics Display ( 1 or 2 ):2
OK
```

Notice that the file entered is that created by the PEL.FIT.TO.GAUSSIAN word. At the second prompt a 1 was entered so that a plot of pel position verses scan number would appear on the CRT.

DATA.TABLE [--] ***

Creates a text file from the file which was created by the PEL.FIT.TO.GAUSSIAN word.

Example:

```
OK DATA.TABLE
```

File name please: GTEST19.DAT

Name Text File:TTEST19.DAT

OK PRINT TTEST19.DAT

Once again the file created from the PEL.FIT.TO.GAUSSIAN word is used. A new file is created by the name of TTEST19.DAT. This file is then printed out, providing the gaussian parameters for each scan in tabular form.

CREATE.FIDELITY.FILE [--] *

Used internaly by the word IMAGE.FIDELITY. Makes a file in which to store the values of the fidelity.

FIDELITY.INTEGRAL [--] *

Performs the integral as defined by the fidelity.

IMAGE.FIDELITY [--] ***

Prompts the user for the data file, and a name for a new file which will be created to store the value of the fidelity for each scan. Proceeds to calculate the image fidelity.

Example:

OK IMAGE.FIDELITY

Enter data filename please:TEST19.DAT

Name image fidelity fidelity file:FID19.DAT
For run 1 phi is .93
For run 2 phi is .94

The new file is named FID19.DAT. The scan number and the value for the fidelity is srored in a two element array for each succesive scan.

FIDELITY.PLOT [--] ***

Allows the user to plot the values of the fidelity calculated by the IMAGE.FIDELITY word.

Example:

OK FIDELITY.PLOT
Enter fidelity filename please: FID19.DAT
Plotter or graphics display (1 or 2)? 1
OK

The file FID19.DAT is now plotted.

PULSE.SIZE.PLOT [--] ***

Plots the spatial pulse width verses scan number. Be sure to enter the name of the pulse data file. CALCULATE.PULSE.SIZE must be executed prior to using this

command.

SINGLE.MTF.PLOT [--] ***

Plots any given cross-section of the 3D MTF plot. Be sure to enter the name of the MTF file, not the original data filename. MTF.ANALYSIS must be used prior to this command if an MTF file for a set of data does not exist.

THEORETICAL.MTF [--] *

Calculates the theoretical MTF for the system.

THEO.PLOT [--] ***

Plots the theoretical MTF and the measured MTF (any one particular scan) on the same graph for easy visual comparison.

File: AQUISITION.IBM

ARRAY.SCAN.PARAMETERS [--] **

Prompts the user to specify which 64 element segments of the CCD should be used. There are sixteen 64 element blocks numbered from zero to fifteen. The entire array is also enabled or disabled by the last input.

Example:

```
OK ARRAY.SCAN.PARAMETERS
Start value:2 End value:5 Enable/disable scan( 1/0 ):1
```

In this particular example blocks 2 through 5 (inclusive) of the CCD array have been "activated". Had a 0 been entered at the last prompt the entire array would have been "deactivated".

ENABLE.SCAN [--] **

Enables the CCD array. (At least those blocks chosen by the ARRAY.SCAN.PARAMETERS command.)

DISABLE.SCAN [--] **

Disables the entire CCD array.

START.SCAN.BYTE [--] *

Used internally only by the word SCAN.LENGTH . Determines the point where the laser printer will be turned on. This number is written to the laser printer -- P.C. interface card by the word SCAN.LENGTH .

END.SCAN.BYTE [--] *

Used internally only by the word SCAN.LENGTH . Determines the poinjt where the laser printer will be turned off.

SCAN.LENGTH [--] **

Prompts the user for the point where the laser should be turned on, and the final point where the laser will be turned off. It returns the total length of the scan. There are 3360 positions in which to turn the beam on in the 14" scan length.

Example:

```
OK SCAN.LENGTH
Enter scan starting point (max - 3360): 1650
Enter scan end point (max - 3360):1660
The total length of the scan is: 11
OK
```

ACTUATOR.PARAMETERS [--] **

Allows the user to define the step size and the velocity of any of the actuators connected to the Newport Controller.

Example:

```
OK ACTUATOR.PARAMETERS
Step size: S1.1
Velocity: V1.01
OK
```

The velocity and step size of the actuator in port one have now been set.

MOVE.ACTUATOR.ONE [--] **

Steps actuator in port one a distance predetermined by the ACTUATOR.PARAMETERS word.

MOVE.ACTUATOR.TWO [--] **

Steps actuator in port two a distance predetermined by the ACTUATOR.PARAMETERS word.

OUT [command scalars --] **

This command transfers data byte(s) to 8901A then sends the unlisten, untalk commands. Note: The bytes are written

in reverse order to the 8901A, and all series of bytes must be terminated by a -1 . For example, to the start the

transient recorder digitizing it takes the commads 9 0 16. Using the OUT word this would be written,

OK -1 6 0 9 OUT (carriage return)

EXECUTE.CAMAC [--] **

Tells the 8901A to execute the CAMAC cycle previously loaded by using the OUT command.

START.SAMPLING [--] **

Starts the TR8818 digitizing.

BLOCK.READ [--] **

Reads the data stored by the TR8818 into the buffer array DATA. Note that all this data is packed into 16 bit words.

DIGITIZER.WINDOW.OFFSET [--] ***

Sets the the window over which digitizing takes place.

Example:

OK DIGITIZER.WINDOW.OFFSET
Amount of offset:255
OK

Writting an offset of 255 allows signals between 0 and +511mV to be digitized. This is stored in the Lecroy's memory (even when turned off) so it need only be performed once, unless a new digitizing window is needed.

CAMAC.Z [--] **

Sends the CAMAC z command which results in clearing the 8901A. The memory in the TR8818 is not disturbed.

STOP.SAMPLING [--] **

Stops the transient recorder from digitizing.

ACQUISITION.PARAMETERS [--] *

Used internally by DATA.ACQUISITION.AND.STORAGE only. Determines the file size and name for a particular run.

CREATE.DATA.FILE [--] *

Used internally by DATA.ACQUISITION.AND.STORAGE only. Creates a file in which the intensity profiles from the CCD

array will be stored.

SELECT.SCAN [--] *

Used internally by DAT.ACQUISITION.AND STORAGE only. Unpacks the data received from the TR8818.

DATA.ACQUISITION.AND.STORAGE [--] ***

Acquires the data, unpacks, and writes the individual scans to a file named by the user.

Example:

```
OK DATA.ACQUISITION.AND.STORAGE
Run number:345
Amount of data per run:8192
Enter array size:256
Enter amount of defocus:0
Name of file: TEST19.DAT
Enter scan starting point (max - 3360):1650
Enter scan end point (max - 3360):1660
The total length of the scan is:11
Scan number:1
```

WEIGHT.FUNCTION [--] *

Used internally by ARRAY.CHARACTERIZATION only. Sorts the intensity values from smallest to largest. Counts the number of like intensity readings which are stored in the array WEIGHT.

AVERAGE.INTENSITY [--] *

Computes the average intensity and the standard deviation.

INTENSITY.ADJUST [--] *

Throws out all values of the intensity for which the intensity-average difference is greater than the standard deviation.

MEAN.INTENSITY.CALCULATION [--] *

An iterative loop which calculates average intensity, until the standard deviation is constant.

ARRAY.CHARACTERIZATION [--] ***

Prompts the user for the name of a file containing the characterization data, and the filename of the data to be corrected. The corrected waveforms are stored in the original data file.

Example:

```
OK ARRAY.CHARACTERIZATION
Enter filename of characterization data please: CTEST19.DAT
Enter data filename please: TEST19.DAT
OK
```

Both CTEST19.DAT and TEST19.DAT are files created when the DATA.ACQUISITION.AND.STORAGE command were used. However, in CTEST19.DAT the scan was unmodulated. This file is used to calculate the gain coefficients. The gain is then divided out of each scan in TEST19.DAT and the corrected scan rewritten to this file.

```
INTEGER SCALAR COUNTER1
INTEGER SCALAR PLOTTER.OR.GRAPHICS
INTEGER SCALAR ORIGIN

REAL SCALAR ARRAY.SIZE 256  ARRAY.SIZE :=
REAL SCALAR PULSE.SIZE
REAL SCALAR A
REAL SCALAR B
REAL SCALAR C
REAL SCALAR A1
REAL SCALAR B1
REAL SCALAR C1
REAL SCALAR A2
REAL SCALAR B2
REAL SCALAR C2
REAL SCALAR MAX.INTENSITY
REAL SCALAR WINDOW.FIT      10 WINDOW.FIT :=
REAL SCALAR HALF.INTENSITY
REAL SCALAR P1
REAL SCALAR P2
REAL SCALAR MAX.IDEAL.INTENSITY
63. MAX.IDEAL.INTENSITY :=

REAL DIM[ 2 ] ARRAY PHI
REAL DIM[ 5 ] ARRAY STATUS
REAL DIM[ 3 ] ARRAY PARAMETERS
REAL DIM[ ARRAY.SIZE ] ARRAY DIFF.INTENSITY
REAL DIM[ WINDOW.FIT ] ARRAY FITTED.INTENSITY
REAL DIM[ WINDOW.FIT ] ARRAY X.DISTANCE
REAL DIM[ ARRAY.SIZE 2 / 1 + ] ARRAY WAVENUMBER
REAL DIM[ ARRAY.SIZE 2 / 1 + ] ARRAY CUM.MTF
REAL DIM[ 127 ] ARRAY PEL.POSITION

DP.REAL DIM[ ARRAY.SIZE 2 / 1 + ] ARRAY MTF
DP.REAL DIM[ ARRAY.SIZE ] ARRAY GAUSS
DP.REAL DIM[ ARRAY.SIZE ] ARRAY INTENSITY
DP.REAL DIM[ ARRAY.SIZE ] ARRAY IDEAL.INTENSITY

DP.COMPLEX DIM[ ARRAY.SIZE 2 / 1 + ] ARRAY FFT.INTENSITY
DP.COMPLEX DIM[ ARRAY.SIZE 2 / 1 + ] ARRAY FFT.IDEAL.INTENSITY
DP.COMPLEX DIM[ ARRAY.SIZE 2 / 1 + ] ARRAY OTF

20 STRING "FILENAME1
20 STRING "FILENAME2

RAD

: GAUSSIAN

A - DUP * C * -1.0 *
EXP
B *


ASYST Version 1.5
Page 1    c:analysis.ibm   01/04/80   01:04
```

```
;

: CHECK.DATA

CR ." Did you input the requested parameters correctly (Y/N) ?"
"INPUT  " Y" "=

;

: FFT.OF.MEASURED.INTENSITY

DP.REAL.FFT
INTENSITY FFT
SUB[ 1 , STATUS [ 2 ] 2 / 1 + ]
FFT.INTENSITY :=

;


: FFT.OF.IDEAL.INTENSITY

0. IDEAL.INTENSITY :=

MAX.IDEAL.INTENSITY IDEAL.INTENSITY SUB[ 5 , STATUS [ 4 ] ] :=

IDEAL.INTENSITY FFT
SUB[ 1 , STATUS [ 2 ] 2 / 1 + ]

FFT.IDEAL.INTENSITY :=

;


: MTF.CALCULATION

STATUS [ 2 ] STATUS [ 4 ] / DUP 1 MODULO - COUNTER1 :=
20 COUNTER1 <
IF 20 COUNTER1 := THEN

COUNTER1 0 DO

FFT.IDEAL.INTENSITY [ I 1 + ] ZMAG 0. <>

IF FFT.INTENSITY [ I 1 + ]
   FFT.IDEAL.INTENSITY [ I 1 + ] /
   OTF [ I 1 + ] :=

ELSE 0 OTF [ I 1 + ] :=
THEN

LOOP

OTF ZMAG DUP [ 1 ] /
MTF + MTF :=


ASYST Version 1.5
Page 2    c:analysis.ibm   01/04/80   01:04
```

```
;

: RETRIEVE.DATA

"FILENAME1 DEFER> FILE.OPEN

1 SUBFILE
INTENSITY FILE>ARRAY
FILE.CONTENTS SUBFILE
STATUS FILE>ARRAY

"FILENAME1 DEFER> FILE.CLOSE

;

: PULSE.ONE

0 COUNTER1 :=

BEGIN
COUNTER1 1 + COUNTER1 :=
INTENSITY [ COUNTER1 ] 0 <>
UNTIL

COUNTER1   ORIGIN :=

BEGIN
COUNTER1 1 + COUNTER1 :=
INTENSITY [ COUNTER1 ] INTENSITY [ COUNTER1 1 + ] >=
UNTIL

INTENSITY SUB[ ORIGIN , COUNTER1 ORIGIN - 1 + ]
" CREATE.COPY SUB.INTENSITY.ONE" "EXEC
" REAL DIM[ COUNTER1 ORIGIN - 1 + ] ARRAY X.1" "EXEC
" ORIGIN COUNTER1 X.1 []FILL" "EXEC
" X.1 SUB.INTENSITY.ONE 4 LEASTSQ.POLY.FIT" "EXEC
DUP DUP
[ 1 ] A1 :=
[ 2 ] B1 :=
[ 3 ] C1 :=

;

: PULSE.TWO

ARRAY.SIZE 1 + COUNTER1 :=

BEGIN
COUNTER1 1 - COUNTER1 :=
INTENSITY [ COUNTER1 ] 0 <>
UNTIL

COUNTER1  ORIGIN :=


ASYST Version 1.5
Page 3    c:analysis.ibm   01/04/80   01:05
```

```
BEGIN
COUNTER1 1 - COUNTER1 :=
INTENSITY [ COUNTER1 1 - ] INTENSITY [ COUNTER1 ] <=  ,
UNTIL


INTENSITY SUB[ COUNTER1 1 + , ORIGIN COUNTER1 - ]
" CREATE.COPY SUB.INTENSITY.TWO" "EXEC
" REAL DIM[ ORIGIN COUNTER1 - ] ARRAY X.2" "EXEC
" COUNTER1 1 + ORIGIN X.2 []FILL" "EXEC
" X.2 SUB.INTENSITY.TWO 4 LEASTSQ.POLY.FIT" "EXEC
DUP DUP
[ 1 ] A2 :=
[ 2 ] B2 :=
[ 3 ] C2 :=
;


: INFLECTION.POINT.ONE

B1 DUP * 9. * 24. A1 * C1 * - .5 **
-3. B1 * +
12. / A1 /
P1 :=

;

: INFLECTION.POINT.TWO

B2 DUP * 9. * 24. A2 * C2 * - .5 **
-3. B2 * +
12. / A2 /
P2 :=

;


: CREATE.PULSE.FILE

FILE.TEMPLATE
  REAL DIM[ 5 ] SUBFILE
  STATUS [ 3 ] STATUS [ 2 ] / 1 - TIMES
END

"FILENAME2 DEFER> FILE.CREATE

;


: CALCULATE.PULSE.SIZE

BEGIN


ASYST Version 1.5
Page 4     c:analysis.ibm    01/04/80    01:05
```

```
CR ." Name of data file please:" "INPUT "FILENAME1 ":=
CR ." Name pulse size file please:" "INPUT "FILENAME2 ":=
CHECK.DATA
UNTIL

RETRIEVE.DATA
CREATE.PULSE.FILE

STATUS [ 3 ] STATUS [ 2 ] / 1 -  0 DO

"FILENAME1 DEFER> FILE.OPEN
I 1 + SUBFILE
INTENSITY FILE>ARRAY
FILE.CLOSE

PULSE.ONE
PULSE.TWO
INFLECTION.POINT.ONE
INFLECTION.POINT.TWO

P2 P1 -  STATUS [ 4 ] :=

I 1 + 1 =
IF "FILENAME1 DEFER> FILE.OPEN
    FILE.CONTENTS SUBFILE
    STATUS ARRAY>FILE
    FILE.CLOSE

THEN

"FILENAME2 DEFER> FILE.OPEN
 I 1 + SUBFILE
STATUS ARRAY>FILE
FILE.CLOSE

CR ." Pulse size:" STATUS [ 4 ] .
" FORGET SUB.INTENSITY.ONE" "EXEC
STACK.CLEAR

LOOP

;


: CREATE.MTF.FILE

FILE.TEMPLATE
  REAL DIM[ STATUS [ 2 ] 2 / 1 + ] SUBFILE
  STATUS [ 3 ] STATUS [ 2 ] / 1 -
  TIMES
END

"FILENAME2 DEFER> FILE.CREATE


ASYST Version 1.5
Page 5    c:analysis.ibm   01/04/80   01:05
```

```
;

: MTF.ANALYSIS

BEGIN
CR ." Name of Data file please: " "INPUT
"FILENAME1 ":=

CR ." Name of MTF file: " "INPUT
"FILENAME2 ":=

CHECK.DATA
UNTIL
RETRIEVE.DATA
FFT.OF.IDEAL.INTENSITY

CREATE.MTF.FILE

STATUS [ 3 ] STATUS [ 2 ] / 1 -  0 DO

0 MTF :=  0 OTF :=

"FILENAME1 DEFER> FILE.OPEN
I 1 + SUBFILE
INTENSITY FILE>ARRAY
 FILE.CLOSE

FFT.OF.MEASURED.INTENSITY
MTF.CALCULATION

"FILENAME2 DEFER> FILE.OPEN
I 1 + SUBFILE
MTF ARRAY>FILE
FILE.CLOSE

LOOP

;

: 3D.MTF.PLOT

BEGIN
CR ." Enter filename please:" "INPUT "FILENAME1 ":=
CR ." Enter number of scans to be in plot:" #INPUT P1 :=
CR ." Plotter or Graphics Display ( 1 or 2 ):" #INPUT
PLOTTER.OR.GRAPHICS :=
CHECK.DATA
UNTIL

"FILENAME1 DEFER> FILE.OPEN
1 SUBFILE
MTF FILE>ARRAY
0 COUNTER1 :=


ASYST Version 1.5
Page 6    c:analysis.ibm   01/04/80   01:06
```

```
BEGIN
COUNTER1 1 + COUNTER1 :=
MTF [ COUNTER1 ] 0 =
UNTIL

MTF SUB[ 1 , COUNTER1 1 - ]

F1  1 DO
I 1 + SUBFILE
MTF FILE>ARRAY
MTF SUB[ 1 , COUNTER1 1 - ]  LAMINATE
LOOP

TRANS[ 1 , 2 ]  REV[ 1 ]
DUP
PLOTTER.OR.GRAPHICS 1 =
IF HP7470
ELSE GRAPHICS.DISPLAY THEN

VERTICAL AXIS.OFF
NO.LABELS
AXON.PLOT

.0170 1.25 * .0644 1.25 * CHAR.SIZE

NORMAL.COORDS
.3 .8 POSITION
" MTF vs. Scan # and S. Frequency" LABEL
.4 .05 POSITION
" Scan Number" LABEL
270 LABEL.DIR
.95 .6 POSITION
" MTF" LABEL

FILE.CLOSE
AXIS.DEFAULTS
;


: PARAMETERS.FOR.GAUSSIAN

MAX.INTENSITY B :=
COUNTER1 1. -  A :=
.0338 C :=

;


: CREATE.GAUSSIAN.FILE

FILE.TEMPLATE
   REAL DIM[ 3 ] SUBFILE
   STATUS [ 3 ] STATUS [ 2 ] / 1 -
   TIMES


ASYST Version 1.5
Page 7    c:analysis.ibm   01/04/80   01:06
```

```
END

"FILENAME2 DEFER> FILE.CREATE

;

: SEARCH.FOR.MAX.INTENSITY

1 COUNTER1 :=

BEGIN

INTENSITY [ COUNTER1 ] MAX.INTENSITY >=

IF INTENSITY [ COUNTER1 ] MAX.INTENSITY :=

THEN

INTENSITY [ COUNTER1 ] MAX.INTENSITY <
COUNTER1 1 + COUNTER1 :=

UNTIL

COUNTER1 1 - COUNTER1 :=

;

: PEL.FIT.TO.GAUSSIAN

BEGIN
CR ." Name of data file please:" "INPUT "FILENAME1 ":=
CR ." Name of Gaussian file:" "INPUT "FILENAME2 ":=
CHECK.DATA
UNTIL

RETRIEVE.DATA
CREATE.GAUSSIAN.FILE

STATUS [ 3 ] STATUS [ 2 ] / 1 - 0 DO

CR ." Calculating scan:" I 1 + .

"FILENAME1 DEFER> FILE.OPEN
I 1 + SUBFILE
INTENSITY FILE>ARRAY
FILE.CLOSE

0 MAX.INTENSITY :=
SEARCH.FOR.MAX.INTENSITY
PARAMETERS.FOR.GAUSSIAN

INTENSITY SUB[ COUNTER1 WINDOW.FIT 2 / 1 -  - , WINDOW.FIT  ]
FITTED.INTENSITY :=


ASYST Version 1.5
Page 8    c:analysis.ibm   01/04/80   01:06
```

```
COUNTER1 WINDOW.FIT 2 / 1 - -  COUNTER1 WINDOW.FIT 2 / + X.DISTANCE ]FILL

GAUSS-NEWTON.FIT
CURVE.FIT[ X.DISTANCE , FITTED.INTENSITY ; GAUSSIAN ; A , B , C ]

PARAMETERS @[ 1 ] ENTER[ A , B , C ]

CR ." Parameters for the scan are:" A . B . C .

"FILENAME2 DEFER> FILE.OPEN
I 1 + SUBFILE
PARAMETERS ARRAY>FILE
"FILENAME2 DEFER> FILE.CLOSE

STACK.CLEAR

LOOP

;


: PLOT.PEL.DATA

BEGIN
CR ." Enter file name please:" "INPUT  "FILENAME1 ":=
CR ." Type : 1 - for pell position 2 - for intensity change "
CR ." 3 - for pulse width " #INPUT  A :=
CR ." Plotter or Graphics Display ( 1 or 2 ):" #INPUT
PLOTTER.OR.GRAPHICS :=
CHECK.DATA
UNTIL

"FILENAME1 DEFER> FILE.OPEN

FILE.CONTENTS  0 DO

I 1 + SUBFILE
PARAMETERS FILE>ARRAY

PEL.POSITION @[ I 1 + ] ENTER[ PARAMETERS [ A ] ]
STACK.CLEAR

LOOP

PLOTTER.OR.GRAPHICS 1 =
IF HP7470
ELSE GRAPHICS.DISPLAY THEN

FILE.CLOSE
VERTICAL GRID.OFF
HORIZONTAL GRID.OFF

A 1. =    .


ASYST Version 1.5
Page 9    c:analysis.ibm   01/04/80   01:06
```

```
IF    PEL.POSITION Y.AUTO.PLOT
      .0170 1.3 * .0644 1.3 * CHAR.SIZE
      NORMAL.COORDS
      .3  .974 POSITION
      " Centroid Position vs. Scan " LABEL
      .4  .05  POSITION
      " Scan number" LABEL
      270 LABEL.DIR
      .025 .8 POSITION
      " Photoelement #" LABEL
      .15  .15 POSITION

THEN

A 2. =

IF    PEL.POSITION Y.AUTO.PLOT
      .0170 1.3 * .0644 1.3 * CHAR.SIZE
      NORMAL.COORDS
      .3  .974 POSITION
      " Max Intensity vs. Scan " LABEL
      .4  .05 POSITION
      " Scan number" LABEL
      270 LABEL.DIR
      .025 .8 POSITION
      " Intensity" LABEL
      .15  .15 POSITION

THEN

A 3. =

IF    .5 LN -1.0 * PEL.POSITION /
      .5 ** 2.0 * PEL.POSITION :=
     PEL.POSITION Y.AUTO.PLOT
      .0170 1.3 * .0644 1.3 * CHAR.SIZE
     NORMAL.COORDS
     .3  .974 POSITION
     " Pulse Width vs. Scan " LABEL
     .4 .05 POSITION
     " Scan number" LABEL
     270 LABEL.DIR
     .025 .8 POSITION
     " Pulse Width" LABEL
     .15  .15 POSITION

THEN

;

: DATA.TABLE

BEGIN
CR ." File name please:" "INPUT "FILENAME1 ":=


ASYST Version 1.5
Page 10    c:analysis.ibm   01/04/80   01:07
```

```
CR ." Name text file:" "INPUT "FILENAME2 ":=
CHECK.DATA
UNTIL

"FILENAME2 DEFER> OUT>FILE
"FILENAME1 DEFER> FILE.OPEN
CONSOLE.OFF

CR CR
." Data from:" "FILENAME1 "TYPE
CR
." Scan #     Position of Centroid     Max Intensity        Waist Size"     CR
."               ( in pells )                              ( in microns )"  CR
." ------------------------------------------------------------------------"
CR

0 A :=

FILE.CONTENTS 0 DO

I 1 + SUBFILE
PARAMETERS FILE>ARRAY

4 1 FIX.FORMAT ."       " I 1 + .
9 5 FIX.FORMAT ."           " PARAMETERS [ 1 ] .
."           " PARAMETERS [ 2 ] .
."           "
     .5 LN -1.0 * PARAMETERS [ 3 ] /
     .5 ** 2.0 * 13. * .
CR

PEL.POSITION @[ I 1 + ] ENTER[ PARAMETERS [ 1 ] ]
STACK.CLEAR


LOOP


CR ." Mean Centroid Position:" PEL.POSITION MEAN .
CR ." Standard deviation:"  PEL.POSITION VARIANCE .5 ** .
"FILENAME1 DEFER> FILE.CLOSE
OUT>FILE.CLOSE

;

: CREATE.FIDELITY.FILE

FILE.TEMPLATE
 REAL DIM[ 2 ] SUBFILE
 STATUS [ 3 ] STATUS [ 2 ] / 1 -
 TIMES
END

"FILENAME2 DEFER> FILE.CREATE


ASYST Version 1.5
Page 11    c:analysis.ibm   01/04/80   01:07
```

```
;

: FIDELITY.INTEGRAL


INTENSITY IDEAL.INTENSITY - DUP *
INTEGRATE.DATA [ STATUS [ 2 ] ]

INTENSITY SORT 1 ROTATE [ 1 ] DUP *
P2 P1 - * / 1. SWAP - PHI [ 2 ] :=

COUNTER1 PHI [ 1 ] :=

;

: IMAGE.FIDELITY

BEGIN
CR ." Enter data file name please:" "INPUT
"FILENAME1 ":=
CR ." Name image fidelity file:" "INPUT
"FILENAME2 ":=
CHECK.DATA
UNTIL

RETRIEVE.DATA
CREATE.FIDELITY.FILE
0 COUNTER1 :=

0 IDEAL.INTENSITY  :=

"FILENAME1 DEFER> FILE.OPEN
COUNTER1 1 + SUBFILE
INTENSITY FILE>ARRAY
FILE.CLOSE

INTENSITY DIFFERENTIATE.DATA DIFF.INTENSITY :=
DIFF.INTENSITY LOCAL.MAXIMA
SWAP 2 RAMP CATENATE  [ 1 ] P1 :=

DIFF.INTENSITY  LOCAL.MINIMA
SWAP 2 RAMP CATENATE 3 ROTATE  [ 1 ] P2 :=

1. IDEAL.INTENSITY SUB[ P1 , P2 P1 - 1 + ] :=
INTENSITY SORT 1 ROTATE [ 1 ] IDEAL.INTENSITY *
IDEAL.INTENSITY :=

STATUS [ 3 ] STATUS [ 2 ] / 1 - 0 DO

COUNTER1 1 + COUNTER1 :=

"FILENAME1 DEFER> FILE.OPEN
COUNTER1 SUBFILE


ASYST Version 1.5
Page 12    c:analysis.ibm   01/04/80   01:07
```

```
INTENSITY FILE>ARRAY
FILE.CLOSE
FIDELITY.INTEGRAL

CR ." For run" PHI [ 1 ] . ." Phi is:" PHI [ 2 ] .

"FILENAME2 DEFER> FILE.OPEN
COUNTER1 SUBFILE
PHI  ARRAY>FILE
FILE.CLOSE

STACK.CLEAR
LOOP
;

: FIDELITY.PLOT

BEGIN
CR ." Enter fidelity filename please:" "INPUT "FILENAME1 ":=
CR ." Plotter or Graphics Display ( 1 or 2 ):" #INPUT
PLOTTER.OR.GRAPHICS :=
CHECK.DATA
UNTIL

"FILENAME1 DEFER> FILE.OPEN
FILE.CONTENTS P1 :=
" REAL DIM[ P1 ] ARRAY FIDELITY" "EXEC

P1 0 DO
STACK.CLEAR
I 1 + SUBFILE
PHI FILE>ARRAY
" FIDELITY @[ I 1 + ] ENTER[ PHI [ 2 ] ]" "EXEC
LOOP

PLOTTER.OR.GRAPHICS 1 =
IF HP7470
ELSE GRAPHICS.DISPLAY THEN

VERTICAL GRID.OFF
HORIZONTAL GRID.OFF

" FIDELITY Y.AUTO.PLOT" "EXEC
      .0170 1.3 * .0644 1.3 * CHAR.SIZE
NORMAL.COORDS
.3 .974 POSITION
" Fidelity vs. Scan" LABEL
.4 .05 POSITION
" Scan Number" LABEL
270 LABEL.DIR
.025 .8 POSITION
" Fidelity" LABEL
.15 .15 POSITION


ASYST Version 1.5
Page 13    c:analysis.ibm   01/04/80   01:08
```

```
FILE.CLOSE
;

: PULSE.SIZE.PLOT

BEGIN
CR ." Enter data file name:" "INPUT "FILENAME1 ":=
CR ." Plotter or Graphics Display ( 1 or 2 ):" #INPUT
PLOTTER.OR.GRAPHICS :=
CHECK.DATA
UNTIL

"FILENAME1 DEFER> FILE.OPEN
FILE.CONTENTS P1 :=

" REAL DIM[ P1 ] ARRAY PULSES" "EXEC

P1 0 DO
STACK.CLEAR
I 1 + SUBFILE
STATUS FILE>ARRAY
" PULSES @[ I 1 + ] ENTER[ STATUS [ 4 ] ] " "EXEC
LOOP

FILE.CLOSE

PLOTTER.OR.GRAPHICS 1 =
IF HP7470
ELSE GRAPHICS.DISPLAY THEN

VERTICAL GRID.OFF
HORIZONTAL GRID.OFF

" PULSES Y.AUTO.PLOT" "EXEC
      .0170 1.3 * .0644 1.3 * CHAR.SIZE
NORMAL.COORDS
.3 .974 POSITION
" Pulse Size vs. Scan" LABEL
.4 .05 POSITION
" Scan Number" LABEL
270 LABEL.DIR
.025 .8 POSITION
" Pulse Size" LABEL
.15 .15 POSITION

;

: SINGLE.MTF.PLOT

BEGIN
CR ." Name of MTF file please : " "INPUT "FILENAME1 ":=
CR ." Scan Number : " #INPUT P1 :=
CR ." Plotter or Graphics Display ( 1 or 2 ):" #INPUT
PLOTTER.OR.GRAPHICS :=


ASYST Version 1.5
Page 14    c:analysis.ibm   01/04/80   01:08
```

```
CHECK.DATA
UNTIL

"FILENAME1 DEFER> FILE.OPEN
P1 SUBFILE
MTF FILE>ARRAY
FILE.CLOSE

0 COUNTER1 :=

BEGIN
COUNTER1 1 + COUNTER1 :=
MTF [ COUNTER1 ] 0 =
UNTIL

PLOTTER.OR.GRAPHICS 1 =
IF HP7470
ELSE GRAPHICS.DISPLAY THEN

VERTICAL GRID.OFF
HORIZONTAL GRID.OFF
" REAL DIM[ COUNTER1 1 - ] ARRAY X.AXIS" "EXEC
0 COUNTER1 2. - .013 / ARRAY.SIZE / " X.AXIS []FILL" "EXEC
" X.AXIS MTF SUB[ 1 , COUNTER1 1 - ] XY.AUTO.PLOT" "EXEC
.0170 1.3 * .0644 1.3 * CHAR.SIZE
NORMAL.COORDS
.3 .974 POSITION
" Modulation Transfer Function" LABEL
.4 .05 POSITION
" Cycles per mm" LABEL
270 LABEL.DIR
.025 .8 POSITION
" MTF" LABEL
.15 .15 POSITION

" FORGET X.AXIS" "EXEC

;

: THEORETICAL.MTF

STATUS [ 2 ] 2. / A :=
MAX.IDEAL.INTENSITY B :=
.0338 C :=
0. MTF :=
0. INTENSITY :=

FFT.OF.IDEAL.INTENSITY

1. STATUS [ 2 ] GAUSS []FILL
GAUSS GAUSSIAN IDEAL.INTENSITY
CONV.APER SUB[ 1 , STATUS [ 2 ] ]
INTENSITY :=
INTENSITY SORT [ STATUS [ 2 ] ] INTENSITY SWAP /


ASYST Version 1.5
Page 15    c:analysis.ibm    01/04/80    01:08
```

```
MAX.IDEAL.INTENSITY * INTENSITY :=

FFT.OF.MEASURED.INTENSITY
MTF.CALCULATION

;

: THEO.PLOT

BEGIN
CR ." Enter data filename please:" "INPUT "FILENAME1 ":=
CR ." Enter MTF filename please:" "INPUT "FILENAME2 ":=
CR ." Enter scan number:" #INPUT P1 :=
CR ." Plotter or Graphics Display ( 1 or 2 ):" #INPUT
PLOTTER.OR.GRAPHICS :=
CHECK.DATA
UNTIL

"FILENAME1 DEFER> FILE.OPEN
FILE.CONTENTS 2 / 1 + SUBFILE
STATUS FILE>ARRAY
FILE.CLOSE

THEORETICAL.MTF

0 COUNTER1 :=

BEGIN
COUNTER1 1 + COUNTER1 :=
MTF [ COUNTER1 ] 0 =
UNTIL

PLOTTER.OR.GRAPHICS 1 =
IF HP7470
ELSE GRAPHICS.DISPLAY THEN

VERTICAL GRID.OFF
HORIZONTAL GRID.OFF
" REAL DIM[ COUNTER1 1 - ] ARRAY X.AXIS" "EXEC
0 COUNTER1 2. -  .013 / ARRAY.SIZE / " X.AXIS []FILL" "EXEC
" X.AXIS MTF SUB[ 1 , COUNTER1 1 - ] XY.AUTO.PLOT" "EXEC


"FILENAME2 DEFER> FILE.OPEN
P1 SUBFILE
MTF FILE>ARRAY
FILE.CLOSE

.01 .005 .01 .005 DASHED
" X.AXIS MTF SUB[ 1 , COUNTER1 1 - ] XY.DATA.PLOT" "EXEC
     .0170 1.3 * .0644 1.3 * CHAR.SIZE
NORMAL.COORDS
.3 .974 POSITION
" Real and Theoretical MTF" LABEL


ASYST Version 1.5
Page 16    c:analysis.ibm   01/04/80   01:09
```

```
8255.PORT                          \ This section of the program initializes
                                   \ the Metrabyte and DataTranslation cards.

4 DIGITAL.TEMPLATE OUT.A           \ For the Metrabyte card : Ports A and
5 DIGITAL.TEMPLATE OUT.B           \ B are write only ports. Port A is defined
                                   \ as OUT.A , while Port B is defined as OUT.B
OUT.A DIGITAL.INIT                 \ Bits 0-3 of Port C is a write port known as
OUT.B DIGITAL.INIT                 \ OUT.C . Bits 4-7 of Port C is a read port
                                   \ known as READ.C
IO.PORT

0 DIGITAL.TEMPLATE READ.C
4 DIGITAL.TEMPLATE OUT.C

READ.C DIGITAL.INIT
OUT.C DIGITAL.INIT

DT2800                             \ For the DataTranslation card : Port 1
                                   \ is a write only port known as OUT.1
4 DIGITAL.TEMPLATE OUT.0           \ Port 0 is also a write only port
OUT.0 DIGITAL.INIT                 \ denoted by OUT.0
5 DIGITAL.TEMPLATE OUT.1
OUT.1 DIGITAL.INIT

20 STRING "DATA.IN                \ Define all pertinent variables
20 STRING "DATA.OUT
20 STRING "FILENAME1
20 STRING "FILENAME2

INTEGER SCALAR ARRAY.START
INTEGER SCALAR ARRAY.END
INTEGER SCALAR SCAN
INTEGER SCALAR BYTE1  112 BYTE1 :=   \ Byte1 is initialized at 112 so that
INTEGER SCALAR BYTE2  0 BYTE2 :=     \ the TR8818 default parameter is :
INTEGER SCALAR VOLTAGE.OFFSET        \ 0/8 of memory pre-trigger, external
INTEGER SCALAR BUFFER                \ clock.
INTEGER SCALAR A 8200 BUFFER :=
INTEGER SCALAR B
INTEGER SCALAR ARRAY.SIZE 64   ARRAY.SIZE :=
INTEGER SCALAR START.SCAN
INTEGER SCALAR END.SCAN
INTEGER SCALAR BYTE3
INTEGER SCALAR BYTE4
INTEGER SCALAR COUNTER1
INTEGER SCALAR ORIGIN
INTEGER SCALAR LSB

INTEGER DIM[ 5 ] ARRAY STATUS

REAL SCALAR AVERAGE
REAL SCALAR S.DEV
REAL SCALAR OLD.S.DEV  40 OLD.S.DEV :=
REAL SCALAR FILE.LENGTH


ASYST Version 1.5
Page 1    c:acquire.ibm   01/04/80   02:00
```

```
REAL DIM[ ARRAY.SIZE ] ARRAY INTENSITY
REAL DIM[ ARRAY.SIZE ] ARRAY WEIGHT
REAL DIM[ ARRAY.SIZE ] ARRAY I.VALUE
REAL DIM[ ARRAY.SIZE ] ARRAY ADJUST
REAL DIM[ ARRAY.SIZE ] ARRAY S.INTENSITY
REAL DIM[ ARRAY.SIZE ] ARRAY GAIN

SEND.INTERFACE.CLEAR
REMOTE.ENABLE.ON

1 GPIB.DEVICE 8901A
2 GPIB.DEVICE LINEAR.ACTUATOR
13 EOS.CHARACTER
EOS.ON

INTEGER DIM[ BUFFER ] ARRAY DATA      \ Sets default buffer size of 8901A
8901A DATA []GPIB.BUFFER              \ to BUFFER 16 bit.

: CHECK.DATA                .
CR ." Did you input the requested parameters correctly (Y/N)?"
"INPUT " Y" "=
;


: ARRAY.SCAN.PARAMETERS                \ Used to enable array. Also ,
                                       \ determines number of blocks of
CR                                     \ the array which will be digitized
BEGIN                                                              .
." Start value:" #INPUT ARRAY.START :=
." End value:" #INPUT ARRAY.END :=
." Enable/disable scan( 1 / 0 ) :" #INPUT SCAN :=
CHECK.DATA
UNTIL

OUT.B
ARRAY.START 16 * ARRAY.END + DIGITAL.OUT
OUT.C
SCAN DIGITAL.OUT

;

: ENABLE.SCAN             \ This command enables the array clock.

OUT.C 1 DIGITAL.OUT

;

: DISABLE.SCAN            \ This command disables the array clock

OUT.C 0 DIGITAL.OUT

;


ASYST Version 1.5
Page 2    c:acquire.ibm   01/04/80   02:00
```

```
: START.SCAN.BYTE

START.SCAN 2048 - DUP 0 >=
IF START.SCAN :=  8 BYTE3 := THEN
START.SCAN 1024 - DUP 0 >=
IF START.SCAN :=  BYTE3 4 + BYTE3 := THEN
START.SCAN  512 - DUP 0 >=
IF START.SCAN :=  BYTE3 2 + BYTE3 := THEN
START.SCAN  256 - DUP 0 >=
IF BYTE4 :=  BYTE3 1 + BYTE3 :=
ELSE START.SCAN BYTE4 := THEN
BYTE3 16 + BYTE3 :=

;

: END.SCAN.BYTE

END.SCAN 2048 - DUP 0 >=
IF END.SCAN := 8 BYTE3 := THEN
END.SCAN 1024 - DUP 0 >=
IF END.SCAN := BYTE3 4 + BYTE3 := THEN
END.SCAN 512 - DUP 0 >=
IF END.SCAN := BYTE3 2 + BYTE3 := THEN
END.SCAN 256 - DUP 0 >=
IF BYTE4 := BYTE3 1 + BYTE3 :=
ELSE END.SCAN BYTE4 := THEN

;

: SCAN.LENGTH

CR ." Enter scan starting point ( max - 3360 ):"
#INPUT START.SCAN :=

CR ." Enter scan end point ( max - 3360 ):"
#INPUT END.SCAN :=

END.SCAN START.SCAN - 1. + STATUS [ 4 ] :=

CR ." The total length of the scan is:" STATUS [ 4 ] .

0 BYTE3 :=  0 BYTE4 :=

START.SCAN.BYTE
OUT.0
BYTE3 DIGITAL.OUT
OUT.1
BYTE4 DIGITAL.OUT
OUT.0
BYTE3 16 - DIGITAL.OUT

0 BYTE3 := 0 BYTE4 :=

END.SCAN.BYTE


ASYST Version 1.5
Page 3    c:acquire.ibm   01/04/80   02:01
```

```
OUT.0
BYTE3 DIGITAL.OUT
OUT.1
BYTE4 DIGITAL.OUT

;

: ACTUATOR.PARAMETERS      \ Allows one to define the step size and the
                           \ velocity of any of the actuators connected
LINEAR.ACTUATOR            \ to the Newport Controller.
CR
." Step size:" "INPUT GPIB.WRITE
." Velocity:" "INPUT GPIB.WRITE

;

: MOVE.ACTUATOR.ONE          \ Steps actuator in port one a distance
                             \ predetermined by the actuator.parameters
LINEAR.ACTUATOR              \ word.
" I1" GPIB.WRITE

;

: MOVE.ACTUATOR.TWO         \ Performs the same function as above
                            \ except the controller in port two is
LINEAR.ACTUATOR             \ incremented.
" I2" GPIB.WRITE

;

: OUT                        \ This marks the beginning of words
                             \ defined to operate the LeCroy system.
ME TALKER 8901A LISTENER

BEGIN
DUP 0 < ?DUP
IF
ELSE ASCII" TALK             \ The OUT command transfers data byte(s)
THEN                         \ to 8901A then sends the unlisten ,
UNTIL                        \ untalk commands.
UNTALK UNLISTEN
DROP

;


: EXECUTE.CAMAC           \ Tells the 8901A to execute camac cycle
                          \ previously loaded by using the OUT
8901A TALKER UNTALK       \ command.

;


ASYST Version 1.5
Page 4     c:acquire.ibm    01/04/80    02:01
```

```
: START.SAMPLING          \ Starts the TR8818 digitizing

-1 BYTE2 BYTE1 6 0 16 OUT
EXECUTE.CAMAC

-1 6 0 9 OUT
EXECUTE.CAMAC

;

: SAMPLING.PARAMETERS       \ Sets the TR8818 control parameters.

CR '
." Pre-trigger / Sampling period:" #INPUT
BYTE1 :=

." Memory Size :"  #INPUT
BYTE2 :=

;

: BLOCK.READ                \ Reads the data stored by the TR8818
                            \ into the buffer array DATA. Note
                            \ that all this data is packed into 16
-1 122 OUT                  \ bit words.

-1 6 0 17 OUT

EXECUTE.CAMAC

-1 6 0 2 OUT

ME LISTENER 8901A TALKER
BUFFER.LISTEN

UNTALK UNLISTEN

;

: DIGITIZER.WINDOW.OFFSET     \ Sets the window over which digitizing
                              \ takes place.
CR
." Amount of offset:" #INPUT
VOLTAGE.OFFSET :=


-1 6 0 19 VOLTAGE.OFFSET OUT

EXECUTE.CAMAC

;


ASYST Version 1.5
Page 5    c:acquire.ibm   01/04/80   02:01
```

```
: CAMAC.Z              \ Sends the CAMAC Z command.

-1 49 OUT
EXECUTE.CAMAC

;

: STOP.SAMPLING              \ Computer stop trigger.

-1 25 OUT
EXECUTE.CAMAC

;

: 8901A.BUFFER.SIZE          \ Sets the 8901A buffer size.

CR
." Enter buffer size (8k to 16k , 4k increments ) :"
#INPUT BUFFER :=
INTEGER DIM[ BUFFER ] ARRAY DATA
8901A DATA []GPIB.BUFFER

;


: ACQUISITION.PARAMETERS     \ Determines the file size and name for
                             \ a particular run .

CR ." Run Number:" #INPUT STATUS [ 1 ] :=
CR ." Amount of data per run:" #INPUT STATUS [ 3 ] :=
CR ." Enter array size:" #INPUT STATUS [ 2 ] :=
CR ." Enter amount of defocus:" #INPUT STATUS [ 5 ] :=
CR ." Name of file ( 20 characters or less ):" "INPUT
"FILENAME1 ":=

;

: CREATE.DATA.FILE

FILE.TEMPLATE
   INTEGER DIM[ STATUS [ 2 ] ] SUBFILE
   STATUS [ 3 ] STATUS [ 2 ] / 1 -  TIMES
   REAL DIM[ 5 ] SUBFILE
END

"FILENAME1
DEFER> FILE.CREATE

"FILENAME1
DEFER> FILE.OPEN

;

: SELECT.1.SCAN


ASYST Version 1.5
Page 6    c:acquire.ibm   01/04/80   02:02
```

```
LSB 0 =

IF DATA SUB[ ORIGIN , STATUS [ 2 ] 2 / ]
UNPACK 255 SWAP -
 INTENSITY :=

ELSE DATA SUB[ ORIGIN , STATUS [ 2 ]  2 / 1 + ]
UNPACK SUB[ 2 , STATUS [ 2 ] / ]  255 SWAP -
 INTENSITY :=

THEN

;

: DATA.ACQUISITION.AND.STORAGE

DISABLE.SCAN
BEGIN
ACQUISITION.PARAMETERS
CHECK.DATA
UNTIL
CREATE.DATA.FILE

0 COUNTER1 :=

SCAN.LENGTH
START.SAMPLING
STOP.SAMPLING
ENABLE.SCAN
500 MSEC.DELAY
BLOCK.READ

DATA SUB[ 1 , 2 ] UNPACK
[ 1 ] LSB :=


STATUS [ 2 ] 2 / 2 + ORIGIN :=


STATUS [ 3 ] STATUS [ 2 ] / 1 - 0 DO


CR ." Scan number:" I 1 + .

SELECT.1.SCAN
I 1  + SUBFILE
INTENSITY ARRAY>FILE

STACK.CLEAR
ORIGIN STATUS [ 2 ] 2 / + ORIGIN :=
LOOP


ASYST Version 1.5
Page 7    c:acquire.ibm   01/04/90   02:02
```

```
FILE.CONTENTS SUBFILE
STATUS ARRAY>FILE
DISABLE.SCAN

FILE.CLOSE

;


: WEIGHT.FUNCTION

INTENSITY SORT S.INTENSITY :=              \ Sorts the intensity values from
1 A :=      1 B :=                         \ smallest to largest.

BEGIN
S.INTENSITY [ A ] S.INTENSITY [ A 1 +  ] -
0 =

IF WEIGHT [ B ] 1 + WEIGHT [ B ] :=
                                           \ Counts the number of like intensity
ELSE                                       \ readings. These are stored in WEIGHT.
    WEIGHT [ B ] 1 + WEIGHT [ B ] :=        \ The corresponding value of the
   S.INTENSITY [ A ] I.VALUE [ B ] :=       \ intensity is stored in I.VALUE
    B 1 + B :=

THEN A 1 + A :=
     A ARRAY.SIZE =

UNTIL

S.INTENSITY [ A ] I.VALUE [ B ] :=
WEIGHT [ B ] 1 + WEIGHT [ B ] :=

;

: AVERAGE.INTENSITY

WITH.FREQUENCIES

I.VALUE WEIGHT MEAN AVERAGE :=
                                           \ Computes the average intensity
I.VALUE WEIGHT SAMPLE.VARIANCE .5 **       \ and the standard deviation.
S.DEV :=

;

: INTENSITY.ADJUST
                                  \ Throws out all values of the intensity
I.VALUE AVERAGE - ABS             \ for which the intensity average difference
S.DEV [<] ADJUST :=               \ is greater than the standard deviation.

ARRAY.SIZE 0 DO

I.VALUE [ I 1 + ] ADJUST [ I 1 + ] * I.VALUE [ I 1 + ] :=
```

```
WEIGHT  [ I 1 + ] ADJUST [ I 1 + ] * WEIGHT  [ I 1 + ] :=

LOOP

;

: MEAN.INTENSITY.CALCULATION

       AVERAGE.INTENSITY
       INTENSITY.ADJUST                \ An iterative loop which calculates
                                       \ average intensity , until the
S.DEV OLD.S.DEV <                      \ standard deviation is constant.

IF S.DEV OLD.S.DEV := MYSELF
THEN

;

: ARRAY.CHARACTERIZATION

0 GAIN :=   1 COUNTER1 :=  0 WEIGHT :=  0 I.VALUE :=

BEGIN
CR ." Enter filename of characterizaton data please:" "INPUT "FILENAME1 ":=
CR ." Enter data filename please:" "INPUT "FILENAME2 ":=
CHECK.DATA
UNTIL

"FILENAME1 DEFER> FILE.OPEN

FILE.CONTENTS 1 - 0 DO

COUNTER1 SUBFILE
INTENSITY FILE>ARRAY
INTENSITY SMOOTH INTENSITY :=
WEIGHT.FUNCTION
MEAN.INTENSITY.CALCULATION
INTENSITY AVERAGE / GAIN + GAIN :=
0. WEIGHT :=   COUNTER1 1 + COUNTER1 :=

STACK.CLEAR

LOOP
FILE.CONTENTS 1 - FILE.LENGTH :=
GAIN FILE.LENGTH / GAIN :=
FILE.CLOSE

"FILENAME2 DEFER> FILE.OPEN

FILE.CONTENTS 1 - 0 DO

I 1 + SUBFILE
INTENSITY  FILE>ARRAY  .
INTENSITY SMOOTH INTENSITY :=


ASYST Version 1.5
Page 9    c:acquire.ibm   01/04/80   02:03
```

```
INTENSITY GAIN /  INTENSITY :=
I 1 + SUBFILE
INTENSITY ARRAY>FILE

LOOP

FILE.CLOSE

;
```

The following variables should be defined to the correct dimension or value before loading the program into memory.

File: ANALYSIS.IBM

ARRAY.SIZE --- Initialize at the length of the CCD array the user wishes to use.

MAX.IDEAL.INTENSITY --- This is currently set at 63. It determines the height of the ideal pulse in both the MTF and FIDELITY calculations. The value of 63 is a common maximum intensity when the system is correctly aligned. If the signal is spanning the full range of the digitizing window, this should probably be changed to a higher value.

PEL.POSITION --- Initialize to the number of scans in a file.

File: ACQUIRE.IBM

BYTE1 --- Currently initialized at 112 so that 0/8 of memory is pre-trigger samples, and the clock is external.

BYTE2 --- Currently set at 0 set that memory size is 8K.

ARRAY.SIZE --- Initialize to the CCD array size to be used.

BUFFER --- Set to 1/2 of the amount of memory that BYTE2 programs the LeCroy system for. For example, if the LeCroy memory is at 8K this should be at 4100. Always make slightly larger than 1/2 of LeCroy memory, so that this buffer can also accept any status bytes the system sends over.

Appendix B -- Instrumentation.

This appendix describes the instrumentation of the system. The basic system configuration is as shown in figure B-1.

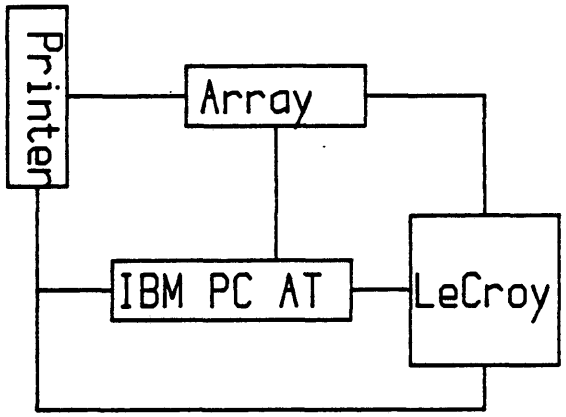

Figure B-1. System configuration.

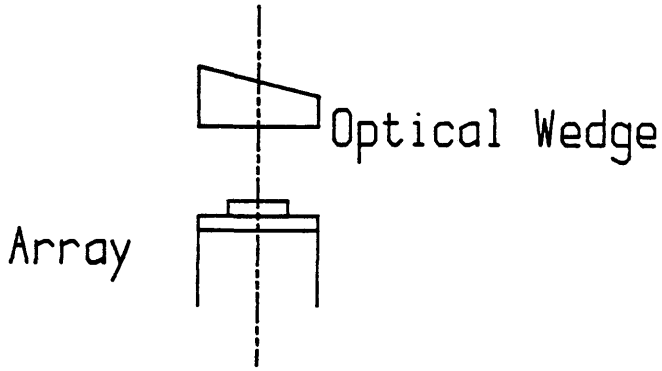

Figure B-2. Exploded end view of wedge and array.

Figure B-2 shows the orientation of the optical wedge with respect to the array. The wedge used was purchased from Melle Griot, product number 02PRW005. The wedge angle is 7.67 degrees. The index matching fluid placed in the wedge-array boundary is from Cargille, Type B.

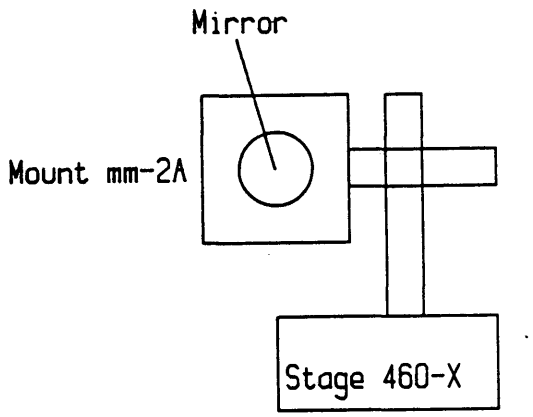

Figure B-3. Mirror holder.

The apparatus designed to hold the mirror which deflected the beam onto the array is shown in figure B-3. The mirror, purchased from Melle Griot has model number 02MFG0016/001 Max-Bright coating. The mirror holder is Newport corporations model number mm-2A. The translational mount is also from Newport, model 460-X. The translational axis for the device is out of the paper. The rods used to connect the device were 10" long.

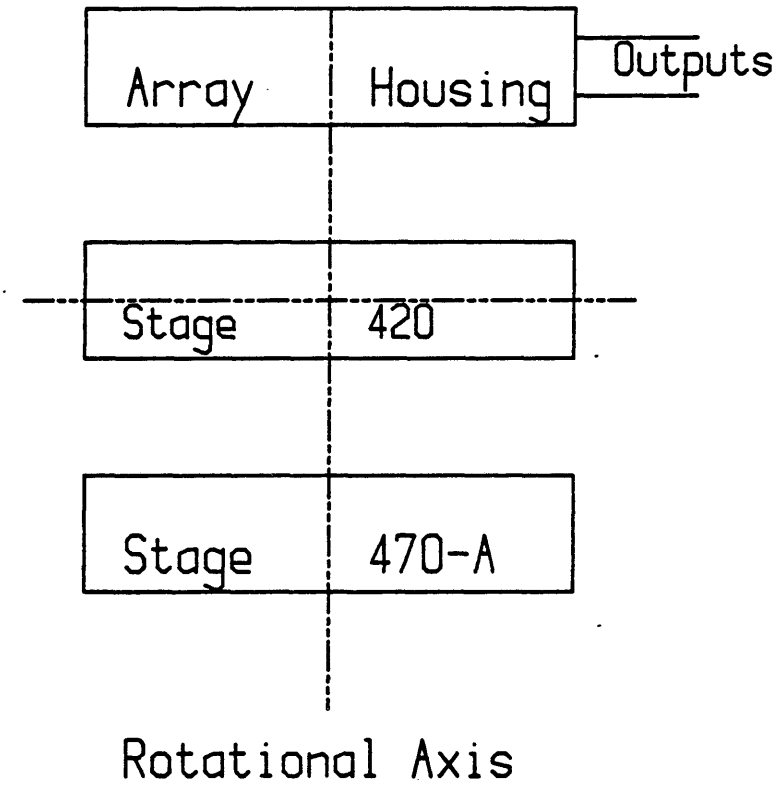

Figure B-4. Array mount.

The final illustration shows how the Newport rotational mount (model 470-A), translational mount (model 420), and the array are stacked together. The rotational axis and translational axis are also shown.