SWEPT-FREQUENCY,
8mm MICROWAVE
INTERFEROMETER FOR MMX

by

B. T. Archer, H. Meuth, and M. A. Lieberman

Memorandum No. UCB/ERL M84/104

21 December 1984

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

This work was supported by DOE Contract No. DE-AT03-76ET53059

SWEPT-FREQUENCY, 8mm MICROWAVE INTERFEROMETER FOR MMX

B. T. Archer, H. Meuth, and M. A. Lieberman

ABSTRACT

A two-channel swept-frequency, 8mm interferometer has been developed as a density diagnostic for the Ten Meter Multiple Mirror Experiment (MMX). The interferometer has been calibrated against a commercial phase shifter and against sample dielectric media. Calibration results indicate that the interferometer gives reliable density measurements for plasma densities in the 10^{11}–$10^{13} cm^{-3}$ range. The diagnostic is currently in use for obtaining time-resolved density profiles on the MMX.

1 Principle of Operation

The interferometer is illustrated in Figure 1. The microwave source signal is split into a plasma arm and a reference arm. Each arm is further split so that the signal enters each of two measurement channels. After passing through the plasma, the source arm of each channel is recombined with its corresponding reference signal in a mixer block and the relative phase is detected there. Variations in the plasma density result in a phase variation in the mixer block.

The microwave source provides a microwave output that can be swept in frequency by application of a tuning voltage. In the present case a sawtooth modulation waveform is used. The output frequency over one sweep can be represented as

$$\omega(t) = \omega_0 + \Delta\omega(t/\tau); \qquad 0 \leq t \leq \tau \tag{1}$$

where

$\Delta\omega$ = magnitude of frequency modulation

ω_0 = minimum frequency

τ = period of the sweep.

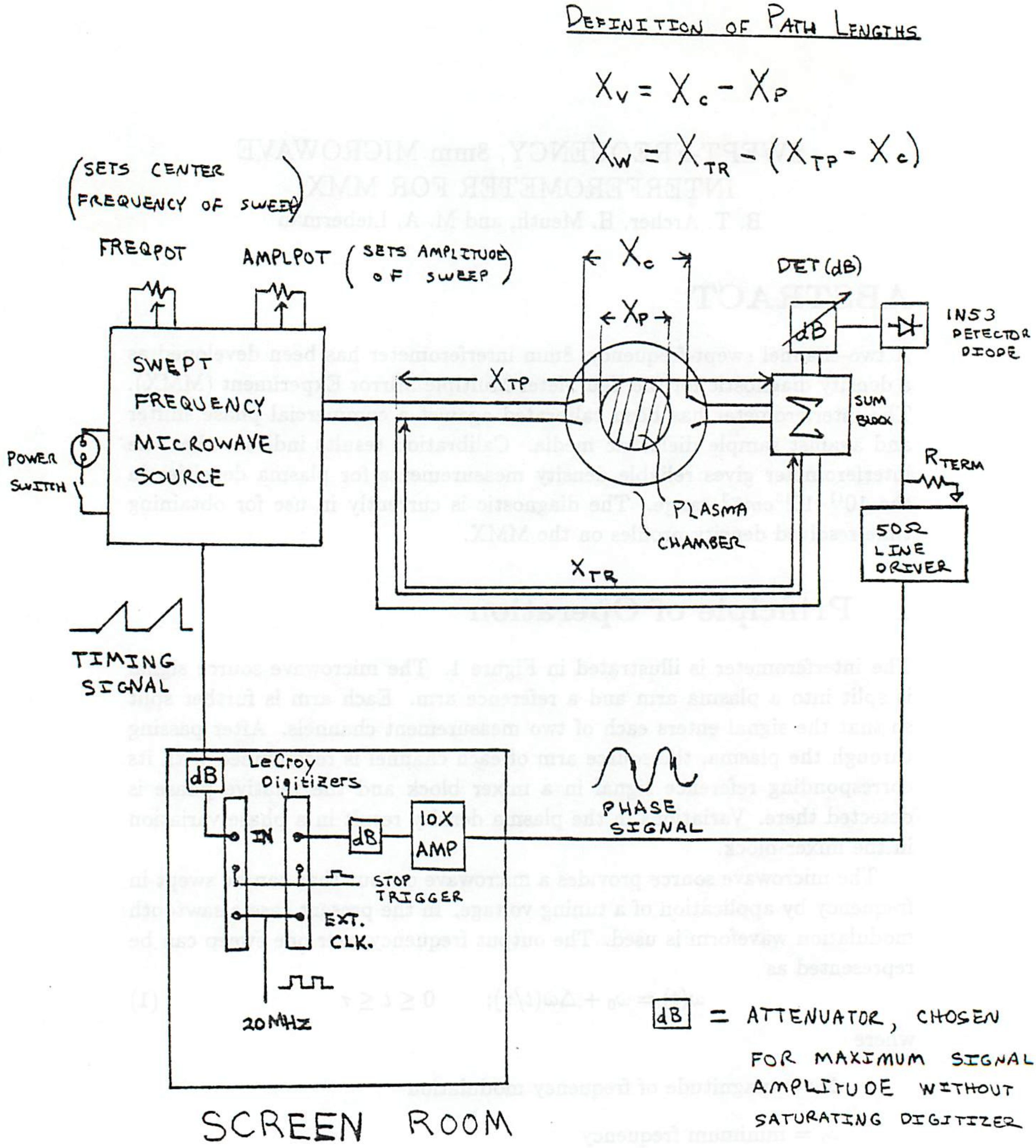


Figure 1: Interferometer block diagram

The Gunn output is fed into two channels. Each channel consists of a plasma arm and a reference arm. The signals are summed by a mixer block, and a square-law detector diode is used to detect the interference signal. As in reference [2], the output voltage is given by

$$u = k\left[U_1^2 + U_2^2 + 2U_1U_2\cos(\Omega_m t + \Phi)\right] \tag{2}$$

where

u = output voltage from detector diode

k = sensitivity of detector

U_i = amplitude of the electic field in arm i

$\Omega_m = 2\pi/\tau$

Φ = phase difference between the two arms

The AC component (third term) provides the useful signal, and it is seen that this term has a phase shift which is given by the difference in electrical path lengths between the two arms. This phase difference can be computed from the dispersion relations for waveguide, plasma, and free space:

$$\Delta\phi(t) = k_w X_w - k_p X_p - k_v X_v, \tag{3}$$

where k_w, k_p, and k_v are the wavenumbers for the microwave signal in the waveguide, the plasma, and vacuum respectively. The path differences X_w, X_p, and X_v are indicated in Fig. 1. The wavenumber k_p is related to the plasma frequency ω_p by the dispersion relation

$$\omega^2 = \omega_p^2 + k_p^2 c^2 \tag{4}$$

which applies for an unmagnetized cold plasma. The same relation holds for a magnetized plasma with $\mathbf{E} \parallel \mathbf{B}$, which is the situation for the present case. The dispersion equation for the $TE_{1,0}$ waveguide is given by

$$\omega^2 = (\pi c/a)^2 + k_w^2 c^2, \tag{5}$$

where a is the longer dimension of the waveguide cross-section. Substitution of (1), (4), and (5) into (3) yields the expression

$$\Delta\phi(t) = (X_w/c)\sqrt{[\omega_0 + \Delta\omega(t/\tau)]^2 - (\pi c/a)^2} \\ -(X_p/c)\sqrt{[\omega_0 + \Delta\omega(t/\tau)]^2 - \omega_p^2} - (X_v/c)\left[\omega_0 + \Delta\omega\,(t/\tau)\right]. \tag{6}$$

Since $\Delta\omega$ is on the order of .5 GHz and ω_0 is near 35 GHz, we may make the approximation $\Delta\omega \ll \omega_0$. With this approximation (6), and using $t \leq \tau$, becomes

$$\Delta\phi(t) = X_w\sqrt{(\omega_0/c)^2 - (\pi/a)^2} - X_p\sqrt{(\omega_0/c)^2 - (\omega_p/c)^2} - X_v\omega_0/c \\ +\Big\{(\omega_0\Delta\omega/\tau)\Big[(X_w/c^2)\left[(\omega_0/c)^2 - (\pi/a)^2\right]^{-1/2} \\ -(X_p/c)\left(\omega_0^2 - \omega_p^2\right)^{-1/2} - X_v/(\omega_0 c)\Big]\Big\}t \tag{7}$$

Comparing this result with (2) we see that it is possible by adjustment of $\Delta\omega$ to make the quantity in curly brackets equal to Ω_m, a procedure suggested by Lisitano[2]. This adjustment is made by observing the diode output on an oscilloscope and adjusting the sweep range on the Gunn source to correspond to one full sine wave output signal per sweep ramp. When the plasma enters the plasma arm there is a small correction to this frequency. The phase of the signal is given by the first three terms. In the experimental situation a vacuum shot is obtained first and its phase subtracted from the plasma shot. The phase of the vacuum shot can easily be obtained from (7) by setting $X_p = 0$. The phase difference between the two shots is then given by

$$\Phi \equiv \Delta\phi_{plasma}(t) - \Delta\phi_{vacuum}(t) \\ = X_p\sqrt{(\omega_0/c)^2 - (\omega_p/c)^2} - X_p\omega_0/c. \tag{8}$$

Given Φ, this equation can be solved for ω_p, which in turn is related to the plasma density by (in cgs units)

$$\omega_p^2 = \left(\frac{4\pi n e^2}{m}\right).$$

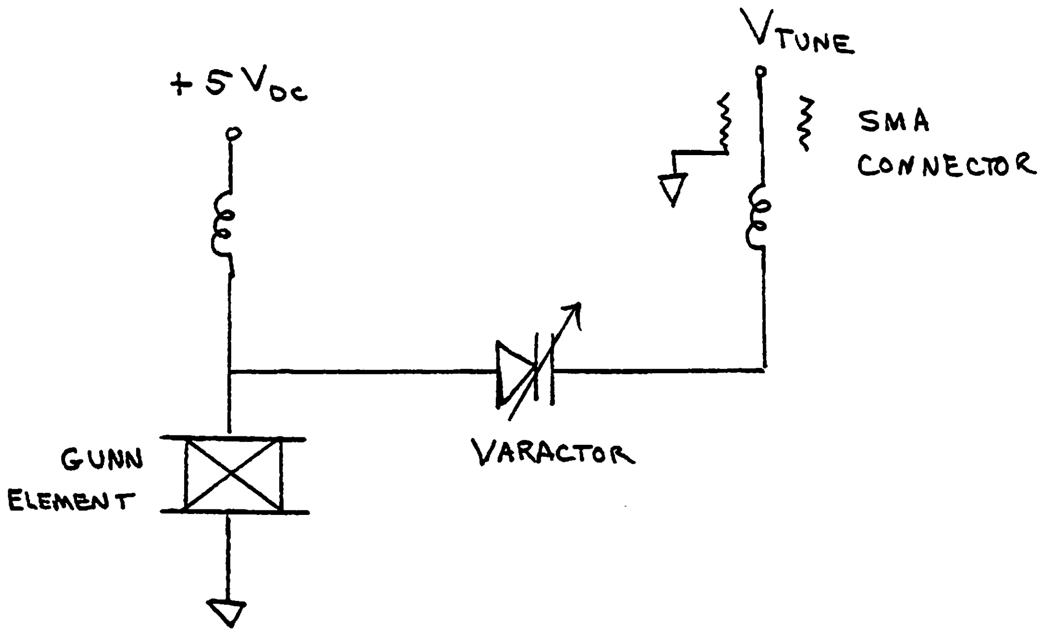


Figure 2: Gunn schematic

The explicit expression for the the density in terms of the phase shift is given by

$$n = \frac{m}{4\pi e^2}\left[\omega_0^2 - \left(\frac{c\Phi}{X_p} - \omega_0\right)^2\right].$$

2 Hardware Description

The microwave source is a tunable Gunn diode oscillator (Figure 2). Tuning is accomplished via a voltage applied to a varactor diode, which in turn alters the resonant frequency of the Gunn/varactor circuit. The Gunn was provided by Central Microwave Company. The manufacturer's specifications for the Gunn are given in Figure 3. The threshold values I_{th} and V_{th} correspond to a maximum in the current drawn by the Gunn, and the power supply must be able to source at least this much current in order to bring the source to the operating point.

$V_0 = 5.5$ $I_0 = 1.08$ $I_{th} = 1.28$ $V_{th} = 1.73$

$T = 25°$

GUNN TUNING CHARACTERISTICS		
V_{tune}	FREQ (GHz)	POWER (dBm)
0.47	34.000	17.8
0.77	.100	18.2
1.03	.200	18.6
1.30	.300	19.1
1.60	.400	19.4
1.90	.500	19.6
2.20	.600	19.7
2.50	.700	19.8
2.80	.800	19.9
3.11	.900	20.0
3.44	35.000	20.0
3.79	.100	20.1
4.19	.200	20.2
4.63	.300	20.2
5.11	.400	20.2
5.62	.500	20.2
6.16	.600	20.3
6.77	.700	20.3
7.45	.800	20.2
8.17	.900	20.2
8.93	36.000	20.1

Figure 3: Gunn diode specifications.

The Gunn drive (Figure 4) consists of three sections. The operation of each section is as follows.

I. The DC bias for the Gunn is provided by an International Series HB5-3/OVP-A DC supply.

II. The tuning waveform begins as a triangle wave output by a Dynascan 3010 Function Generator. This waveform is fed into a circuit which converts the signal into a sawtooth. The triangle wave is applied to both an inverter and a buffer. The TTL-out signal from the function generator is used to alternately throw the inverter output and then the buffer output into a final inverter. Power for this circuit is provided by an International Series HAA5-1.5/OVP-A power supply.

III. In order to protect the Gunn diode against damage from transient currents, the timing waveform is passed through an optical isolator. The TTL output from the function generator drives the isolator, and the square wave output is used to alternately turn on and off a LH0032CG op amp configured as an integrator. The time constant for the timing waveform can be changed by appropriate choice of R_I. The LH0032CG can drive a 50Ω digitizer input directly for recording the timing waveform. Power for the timing waveform is provided by an external HAA5-1.5/OVP-A power supply. The safety ground on this supply must be isolated from building ground in order to prevent ground loops. The ground for the output circuit is instead provided by the screen room ground.

Figure 5 summarizes the path lengths for the various arms of the interferometer. The diodes used for detection are 1N53-type crystals, available through Alpha TRG Microwave Group. The diodes are available in two polarities. The 1N53R diodes have red lettering, and provide a positive output voltage. The 1N53 diodes have black lettering and provide negative output

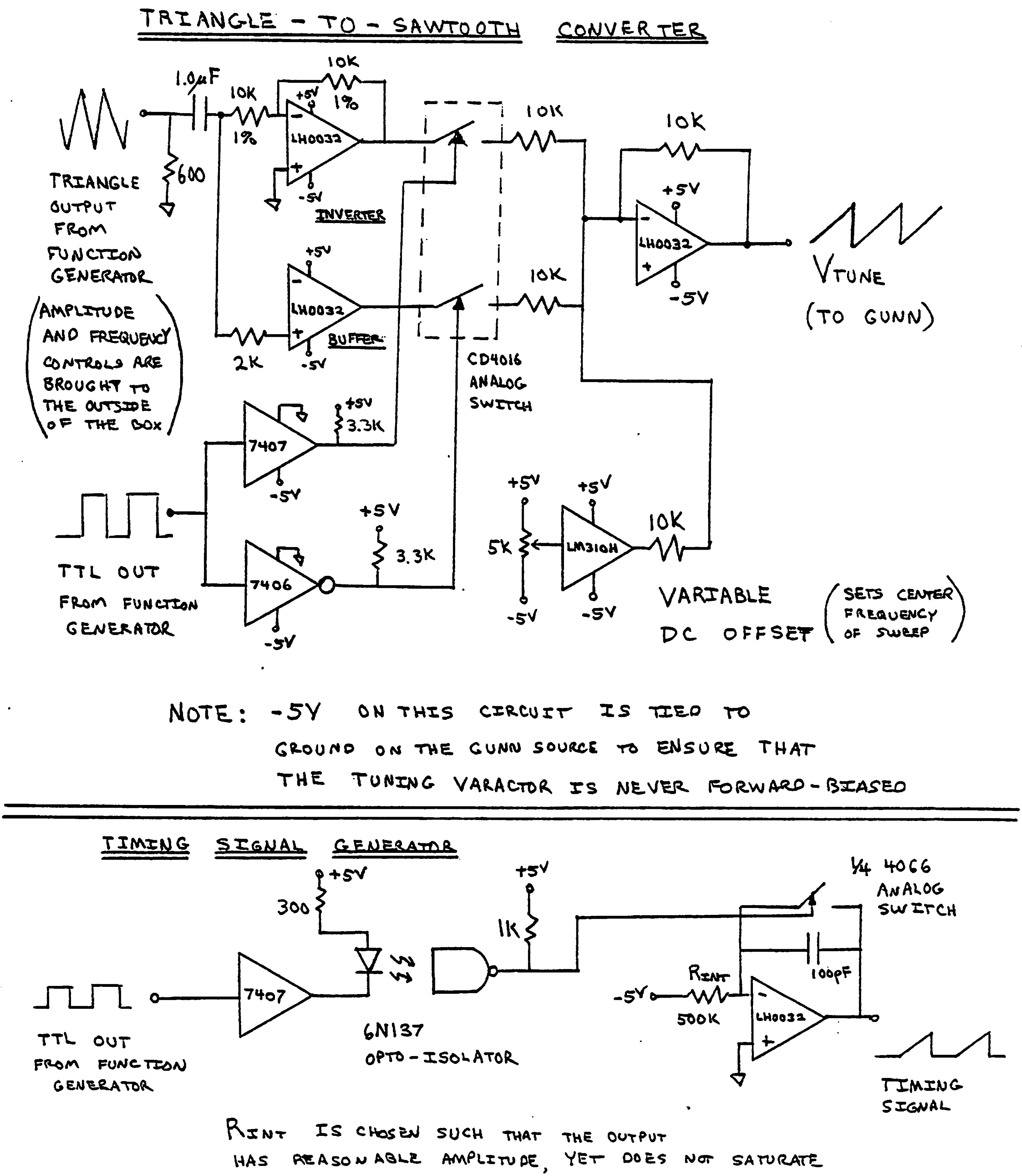


Figure 4: Gunn drive circuit.

Path	Channel	
	M_{67}	M_{78}
Reference	168	178
Plasma	381	390
Path difference	213	212

Figure 5: Interferometer path lengths (cm)

voltage. Either diode may be used for this application. The diodes have approximately a square-law response characteristic for output voltages below ≈50mV. The output from the diode is fed into a 50Ω line driver, with appropriate termination at the input of the line driver. Best results have been obtained with a 2K termination impedance. The output of the line driver is fed into an amplifer with a gain of 10 and input impedance of 1M. The output of the amplifier drives the digitizer.

3 Calibration Results

This section outlines the various techniques which were employed to verify proper operation of the interferometer and the associated data reduction code.

Two types of calibration have been performed. One type of calibration was to insert the Hughes 45751H-1000 direct reading phase shifter in the reference arm of the channel to be calibrated. The phase shifter is of the quarter-wave type described in [3]. According to Hughes engineers, the phase shifter is accurate within ±1° over the entire 26.5–40 GHz Ka band, with an insertion loss of < 1dB. Further, the phase shifter performance is optimized for the most commonly used frequency in the band (35 GHz), which happens to be the approximate frequency of operation for the interferometer. Calibrations have been performed for both channels by taking data at 10° increments.

Excellent linearity has been obtained for both channels (Figure 6). The phase shift measured with the phase shifter set at zero is a function of the exact path differences between the plasma arm and the reference arm. The different zeroes for the two channels are due to the small mismatch in the path difference for channel M_{67} as compared as compared to the channel at M_{78}. For example, a mismatch of $\lambda/2$ would effect a 180° relative phase shift between the two channels. But since the total path difference in each channel is of order 200λ, this small path mismatch is a very small source of error.

The second calibration method was to use dielectric blocks having dielectric constants which differ from that of vacuum by amounts similar to that of typical plasmas. Styrofoam and balsawood were used for the measurement. A cavity perturbation technique was used to obtain the dielectric constant for the materials at $\approx$ 2 GHz. This value was used to calculate theoretical phase shifts for the rods used. The characteristics of the dielectrics are given in the table of Figure 7. The results are plotted in Figure 8, and show reasonable consistency. Part of the scatter in the data is due to the difficulty of aligning the dielectrics in the MMX chamber.

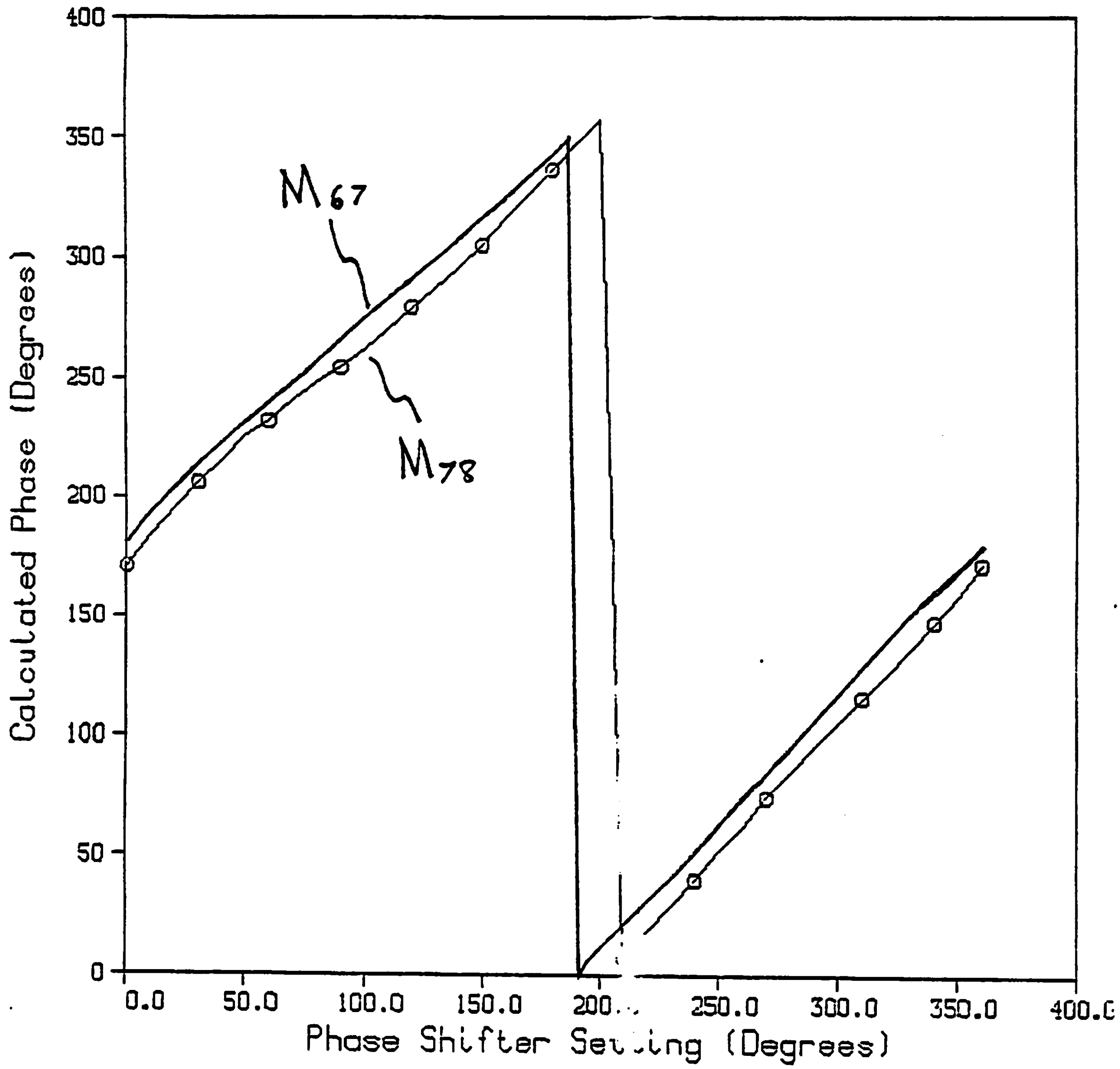


Figure 6: Phase shifter calibration data.

Material	ϵ	Cross Section	Diameter/width (cm)
Styrofoam	1.034	circle	2.6
		circle	3.4
		circle	3.9
		circle	4.5
		rectangle	3.0
		rectangle	3.7
		rectangle	5.1
Balsawood	1.29	rectangle	1.
		rectangle	2.
		rectangle	3.

Figure 7: Dimensions of dielectrics used in calibration.

Interferometer density measurements have been compared with Langmuir probe measurements on the MMX. The probes are biased in the ion saturation region (see probe biasing circuit of Figure 10), and the LaFramboise theory[4] is used to obtain the density from the measured current. The values indicated in Fig. 9 were used to calculate the probe density from the formula

$$N_{\infty+} = \frac{I_{coll}}{i_+ e^{3/2} R_p L_p \left(2\pi/m_+\right)^{1/2} T_+^{1/2}}, \tag{9}$$

where

$N_{\infty+}$ = ion density far from the probe

I_{coll} = collected ion current

e = electron charge

R_p = probe radius

L_p = probe length

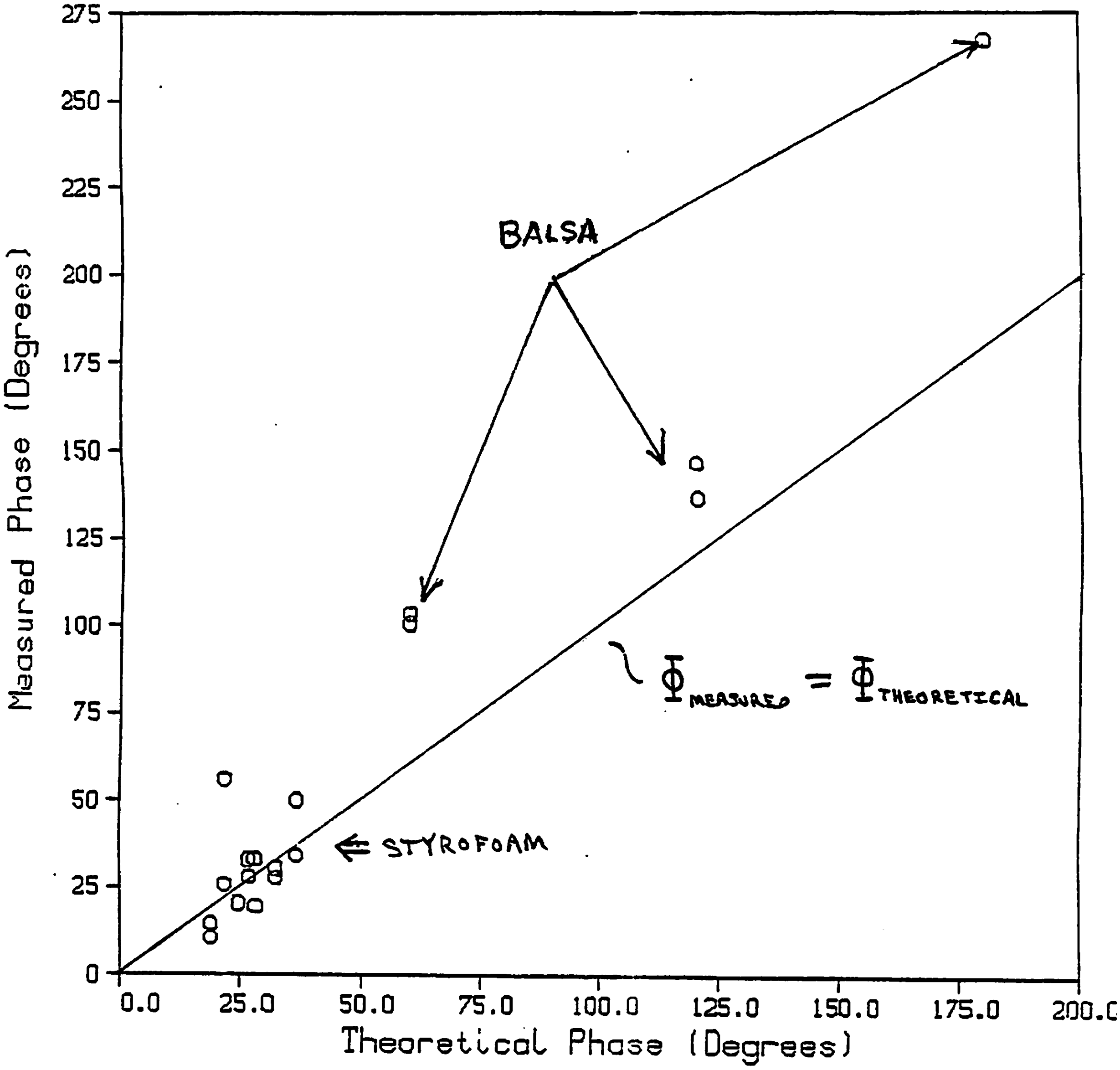


Figure 8: Dielectric rod calibration.

$$L_p = 40 \text{ mils} \quad R_p = 2.5 \text{ mils}$$
$$T_+ \approx 3.eV \quad i_+ = 3$$

Figure 9: Langmuir probe characteristics

m_+ = ion mass

T_+ = ion temperature in eV.

The quantity i_+ is a dimensionless correction factor to the Langmuir theory, which was calculated numerically by LaFramboise for the infinite cylindrical probe case.

The probes were also used to determine an effective path length $L = n_0^{-1} \int n dx$ to be used in the reduction of the interferometer data. This is accomplished by placing a movable probe at various radii and comparing this current with that of a fixed probe on axis. The fixed probe was used to normalize out shot-to-shot density variations. In this way a density profile of good accuracy is obtained. Further, the result does not depend on absolute probe calibrations, as long as probe current is linear with density. The effective path length was determined in this way to be 2.0 cm.

Data from a typical shot is indicated in Figure 11. The probes seem to consistently read a factor of ≈5 lower than the interferometer. The interferometer data was initially held suspect for the discrepancy, since it is a more complicated measurement than that of the probes. A more or less independent verification of the validity of the interferometer measurement was obtained, however, by observing the apparent probe density when the interferometer signal comes out of cutoff. The apparent probe density is consistently lower than the known cutoff density of 35 GHz microwaves ($n_e = 1.52X10^{13} cm^{-3}$) by again a factor of approximately 5 (Figure 12). This result led to a consideration of alternative explanations for the probe/interferometer discrepancy. The LaFramboise theory assumes for its model an unmagnetized collisionless Maxwellian plasma at rest. The disagreement with the interferometer may lie in the violation of these assumptions. At present, however, the most probable explanation is thought to be the presence of a cold ($T_e < 1eV$) "halo" plasma near the chamber

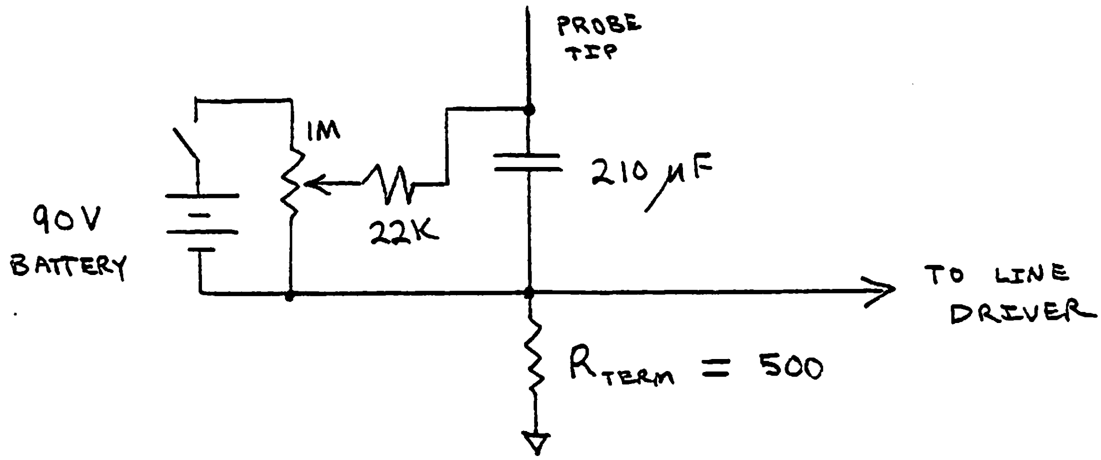


Figure 10: Langmuir probe biasing circuit.

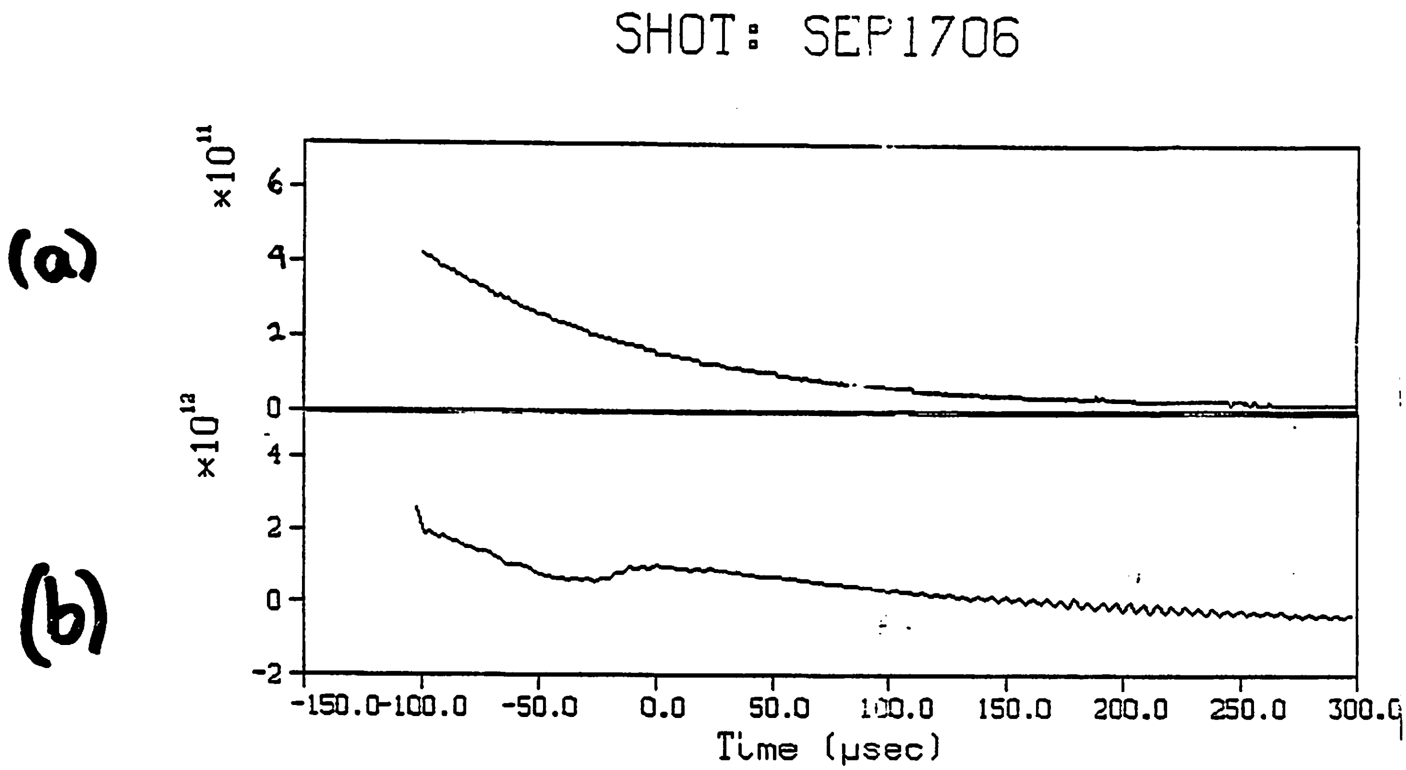
wall, to which the Langmuir probes are relatively insensitive. This explanation implies that both diagnostics are operating properly, only they measure different quantities. This hypothesis requires further investigation.

4 Data Acquisition and Reduction

Operation of the interferometer requires one digitizer for recording the timing information and one digitizer for each channel being recorded. The LeCroy 8837F digitizers are the digitizers of choice in this application, because the Transiac digitizers have been observed occasionally to miss one clock pulse after receiving a stop trigger. A missed pulse introduces an artificial phase shift between the timing channel and the data channel of $\approx 6°$ for a 170 kHz sweep rate.

Typical settings for the various operating parameters are indicated in the sample shot log entry of the following section.

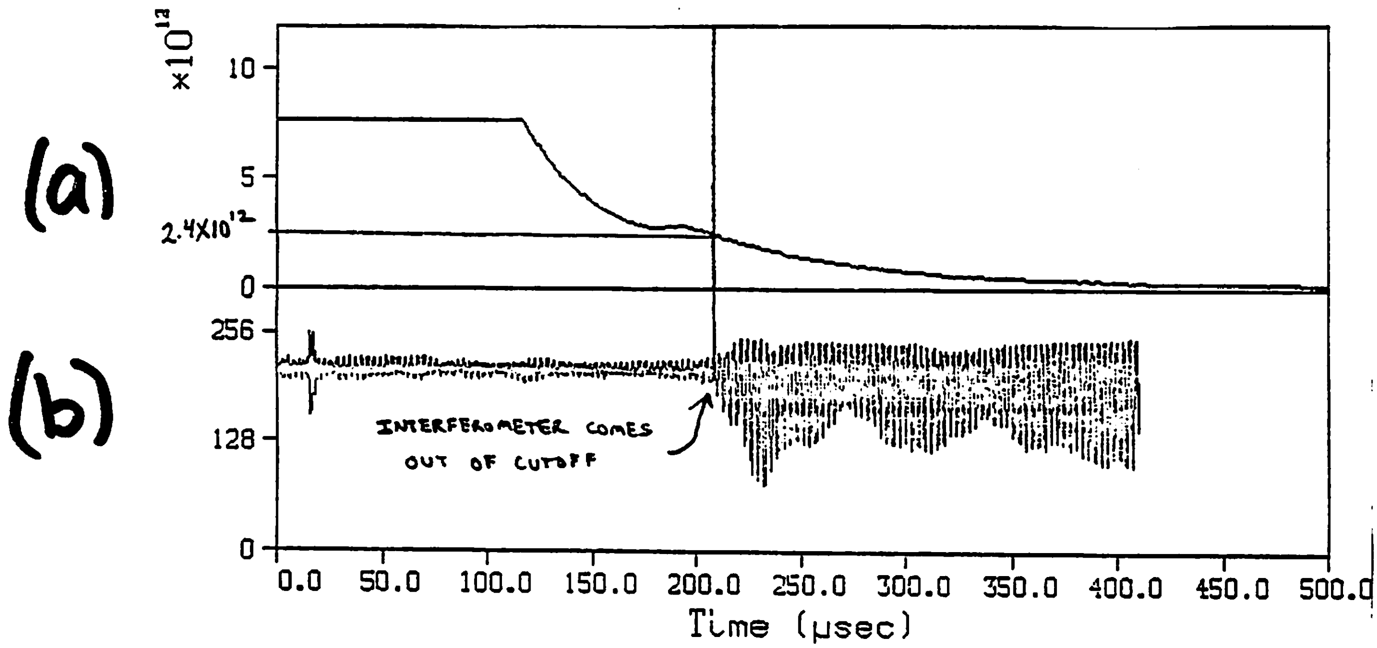
Data reduction is accomplished with the assistance of the general data reduction tools of reference [1]. The data from the interferometer is sent to



(a) Langmuir Probe M_{78} [BOTTOM]
Density (CM^{-3})
r = 1.0 cm i^+ = 3.00
Bias Voltage = -60.0 V
Termination = 500.0 Ω
Attn= 0.0dB Offset=-.233V

(b) Plasma Diameter: 4.00 cm
Center Frequency - 35.7GHz
Channel at M78

Figure 11: Comparison with Langmuir probe results.



(a) Langmuir Probe M_{78} [BOTTOM]
Density (CM^{-3})
r - 0.0 cm i^{+} - 3.C]
Bias Voltage - -60.0 V
Termination - 50.0 Ω
Attn= 0.0dB Offset=-.242V

(b) 8mm Channel at M_{78}
Raw Data

Figure 12: Apparent probe density at interferometer cutoff

NMFECC via the MFE program and analyzed there by the code **uwdnsl**. The source code is **puwdnsl**, a listing of which is given in Appendix A. Instructions for obtaining and compiling the code are included in the *listing*, along with extensive descriptive comments about the algorithms used. The present section is devoted to providing a user's guide for operating the code.

The code requires the following input files:

uwdnsin: Control file in a namelist format.

shotlog: Standard MMX shotlog.

raw data: Standard MMX raw data files.

The output file **uwd** is produced, which contains data suitable for input to the standard plotting program **mmxplt1** [1].

The input file **uwdnsin** contains a number of input variables. Most of the variables assume reasonable default values and can be omitted. A description of each variable follows.

job: This variable is used to select the type of analysis to be performed. Setting `job='plotdns'` initiates the analysis of density waveforms. `job='rods'` is used to analyze data for dielectric rods, and `job='cal'` is used to plot phase shifter calibrations.

shotbeg, shotend: These are the first and last data shots to be analyzed.

shotlog: This is the name of the shot log for the shots to be analyzed. The default shotlog is obtained by taking the first 5 letters of *shotbeg* and appending the letters "log", in conformance with the standard notation of reference [1].

timchan: This is the name of the timing channel. The default is "8MM-TIME".

timmin, timmax: The minimum and maximum times to be analyzed, in μsec.

diam: The plasma diameter to be assumed in the analysis, in cm. The default is 4.0 cm.

maxfn, acc, dfpred: These are parameters used in the IMSL fitting routine *zxcgr*, which is used to fit sine functions to the interferometer signal. Reasonable defaults are used. If convergence problems are encountered for the fit, the IMSL library outputs an appropriate error message to the terminal. Modifying these parameters may solve convergence difficulties.

midplane: This alphanumeric field indicates which midplane is to be analyzed. The default is 67.

freqmult: The frequency cutoff for the digital filter. The value is entered as a multiple fundamental frequency of the timing waveform, which is half the fundamental of the phase waveform. Good results are obtained with the default value of 3.0.

Sample Input

```
job='plotdns'
shotbeg='oct1401'
shotend='oct1410'
midplane='78'
diam=3.5
timmin=0.
timmax=400.
```

The shotlog is also used in the data reduction process. A sample entry for the interferometer follows.

Sample Shot Log Entry

```
SHOT=SEP2001
REFSHOT='SEP2001R'
CHAN=8MMTIME CABLE='L1' ADB=14.  RINT=510K
OFFSET=-8.9 EXTINT=.05
CHAN=8MMPM67 CABLE='L2' TENXAMP=1 ZIN=1M COLOR='BLACK'
```

```
RTERM=2K AMPLPOT=1.75 FREQPOT=17.8 LINEDRVR=1
DETDB=2.  SRCATTN=1.80 CENFREQ=35.675 PSHIFT=0.
OFFSET=0.0 EXTINT=.05 PATHCM=5.1 EPSILON=1.034
```

Some of these variables are used by the data reduction code, and others are only for documentation of the experimental conditions.

REFSHOT: The indicated reference shot is used to establish the reference phase between the timing and phase waveforms.

CABLE: The cable designator of the cable carrying the given signal.

ADB: The signal attenuation in dB.

RINT: The value for the integration resistor for the timing waveform (Figure 4).

OFFSET: The digitizer offset read on the DVM output

EXTINT: The clock rate provided to the external clock input for the digitizer (in μsec).

TENXAMP: The slot number of the gain 10 amplifier used.

ZIN: Input impedance for the amplifier.

COLOR: This describes the color of the diode used, in order to determine the polarity of the signal. It may be either `'red'` or `'black'`

RTERM: Termination resistance for the microwave detector diode.

AMPLPOT: Setting for the amplitude potentiometer on the Gunn source.

FREQPOT: Setting for the frequency potentiometer on the Gunn source.

LINEDRVR: Number of the line driver used to drive the signal cable.

DETDB: Setting of the microwave attenuator to which the detector diode is attached.

SRCATTN: Setting of the attenuator which is attached to the Gunn output.

CENFREQ: Center frequency of the swept microwave source, as measured with the FXR U410A frequency meter.

PSHIFT: Setting of the Hughes phase shifter. This value is used when a calibration run is being used. Otherwise it serves as documentation.

PATHCM, EPSILON: When running a dielectric rod calibration, these values represent the path length of the interferometer signal through the rod, and the dielectric constant of the rod.

Bibliography

[1] Archer, B. T., MMX Data Acquisition and Data Reduction Software, Electronics Research Laboratory Memorandum No. UCB/ERL M84/85, Univ. of California at Berkeley, 1984

[2] Lisitano, G., Rev. Sci. Instr. **36** (1965) 364.

[3] Collin, R. E., **Foundations for Microwave Engineering**, New York, McGraw-Hill 1966

[4] LaFramboise, J. G., Theory of Spherical and Cylindrical Langmuir Probes in a Collisionless, Maxwellian Plasma at Rest, University of Toronto Institute for Aerospace Studies Report No. 100, June 1966

Appendix A.
Listing of PUWDNS1, the interferometer data reduction code.

```
*    Program uwdnsl                   Branch T. Archer III
* Written: 8/10/84                    Revised: 10/3/84
* Compilation:
*     rcft p=puwdnsl,x=uwdnsl,lib=(mmxutl,bevdr,imsl) / t v
* To retrieve sources and executable files from filem:
*     filem rds 1235 .mmxinp  puwdnsl uwdnsl
*     filem rds 1235 .datared fbevdr  bevdr
* Note: imsl is a system library

*     Logical Unit Number designations:
*         1   -   tty      terminal i/o
*         2   -   uwdnsin namelist format control file
*         3   -   mmxerr  message file if msglun<>0
*         4   -   shot     raw data file
*         5   -   uwd      mmxplt-digestible output file

*             STANDARD RAW DATA HEADER
      cliche header
      integer datim(6),filnam,filext,name(32),slot(32),smpcod(32),
     $        reclen(32),pretrg(32),devtyp(32),crate(32),reserv(64)
      common/header/
     $    datim,filnam,filext,name,slot,smpcod,
     $    reclen,pretrg,devtyp,crate,reserv
      endcliche header

      cliche vars
      parameter (maxpnt=8192,maxswp=200,maxcal=100)
      integer isumarm(maxpnt),itime(maxpnt),shot,
*             NAMELIST INTEGERS
     $    job,shotlog,shotbeg,shotend,refshot,plotswp,maxfn,
*             MMXPLT PLOTTING FLAGS
     $    undwadv,plttyp,mrkflg,isym,timflg,frmadv,
*             LABELS
     $    lstrng(8),label(5,25),lblarr(9,10),
*             MISCELLANEOUS
     $    nmpnts,numswp,numeval,iwk(maxcal),period,ramplen,
     $    color,ibegin,iend,calind,nmpt,prvref,signeps,
     $    freqpos,nd2p1,ipower2,timchan,phschan,found
      real sumarm(maxpnt),time(maxpnt),phase(maxswp),swpbeg(maxswp),
     $    caldat(2,maxcal),wk(3*maxpnt+6),smpint,a,refphase,pttime,
*             PHYSICAL VALUES
     $    omega0,omsqr,scfctr,cenfreq,pathcm,epsilon,k0,c,rad2deg,
*             NAMELIST REALS
     $    timmin,timmax,diam,acc,dfpred,freqmult,
*             MMXPLT PLOTTING VARIABLES
     $    xmin,xmax
      complex fourdat(maxpnt/2+1)
      common/vars/shot,
     $    job,shotlog,shotbeg,shotend,refshot,plotswp,maxfn,
     $    undwadv,plttyp,mrkflg,isym,timflg,frmadv,
     $    lstrng,label,lblarr,
     $    nmpnts,numswp,numeval,iwk,period,ramplen,
     $    color,ibegin,iend,calind,nmpt,prvref,signeps,
     $    freqpos,nd2p1,ipower2,timchan,phschan,found,
```

```
       $     sumarm,time,phase,swpbeg,
       $     caldat,wK,smpint,a,refphase,pttime,
       $     omega0,omsqr,scfctr,cenfreq,pathcm,epsilon,K0,c,rad2deg,
       $     timmin,timmax,diam,acc,dfpred,freqmult,
       $     xmin,xmax,
       $     pi,twopi,
       $     fourdat

        equivalence (isumarm,sumarm),(itime,time)
        endcliche vars

        use vars
        use header
        data pi,twopi/3.141592654,6.283185308/c/3.e10/rad2deg/57.29577951/
       $     scfctr/3.15e-10/diam/4./prvref/0/cenfreq/35.7/
       $     maxfn/100/acc/1.e-3/dfpred/500./
       $     undwadv/1/plttyp/1/mrkflg/0/isym/1/timflg/0/frmadv/1/
       $     midplane/'67'/timchan,phschan/'8MMTIME','8MMPM67'/
       $     freqmult/3./
*                        PLOTTING LABELS
        data label/ '$                                            ',     1
       $             '(F)requency ((MH)z)$                         ',     2
       $             '(F)ourier (T)ransform$                       ',     3
       $             '(R)eal (P)art$                               ',     4
       $             '(I)maginary (P)art$                          ',     5
       $             '(I)nterferometer (C)alibration (D)ata$       ',     6
       $             '(P)hase (S)hifter (S)etting ((D)egrees)$',     7
       $             '(D)ensity (()cm\r)-3\rx))$                   ',     8
       $             '(C)alculated (P)hase ((D)egrees)$            ',     9
       $             '8mm (I)nterferometer (R)esults$              ',     10
       $             '(C)enter (F)requency = xx.x (GH)z$           ',     11
       $             '(P)lasma (D)iameter: x.xx cm$                ',     12
       $             '(T)ime ((lm)sec)$                            ',     13
       $             '(F)requency (C)utoff = xx.x$                 ',     14
       $             '(T)heoretical (P)hase ((D)egrees)$           ',     15
       $             '(C)hannel at (M)xx$                          '/     16

* lblarr determines the sequence of labels that will appear
* on various types of plots output by this code.
*         1 - Real part of Fourier Transform of timing waveform
*         2 - Imaginary part of Fourier Transform of timing waveform
*         3 - Phase shifter calibration
*         4 - Density profile
*         5 - Dielectric rod calibration
        data lblarr/10,1,1,3,4,1,1,1,1,
       $             10,1,1,3,5,1,1,1,1,
       $             6,7,9,11,14,1,1,1,1,
       $             10,13,8,12,11,16,1,1,1,
       $             6,15,9,11,14,1,1,1,1/
        namelist/input/job,shotlog,shotbeg,shotend,maxfn,timchan,
       $     timmin,timmax,diam,acc,dfpred,midplane,freqmult
        call link('unit1=tty,unit2=uwdnsin,unit5=(uwd,create,text)//')
 10     read (2,input)
        call tolower(job,40)
```

```
      if (job.eq.'end') goto 800
      if (shotlog.eq.0) then
          shotlog=shotbeg
          call zmovechr(shotlog,5,'log',0,3)
      endif
      call logopn(shotlog)
      calind=0
      shot=shotbeg
      encode(29,1000,label(1,12)) diam
*     36-15 + 4 + 4 = 29       (Count characters for encode statment)
1000  format ('(P)lasma (D)iameter: ',f4.2,' cm$')
      call zmovechr(phschan,5,midplane,0,2)
      call zmovechr(label(1,16),16,midplane,0,2)
      signeps=-1.
      if (job.eq.'rods') signeps=1.
*         Get reference shot.
100   write (1,1100) shot
1100  format ('Processing file ',a8)
      if ((job.eq.'plotdns').or.(job.eq.'rods')) then
*         Calculate reference phase.
      ier=ixtract(shot,phschan,'REFSHOT','ALPHA',refshot,len)
      if (refshot.eq.0) then
          call mmxerr(221,'REFSHOT',8)
          goto 910
      endif
      call tolower(refshot,8)
      if (refshot.ne.prvref) then
*             Open reference shot and read digitizer data.
          call open(4,refshot,0,len)
          if (len.eq.-1) goto 900
          call readdat(timchan,time,1)
          ier=icompare(timchan,phschan)
          if (ier.ne.0) then
              call mmxerr(ier,0,0,'icompare')
              call exit
          endif
          call getswps
          call readdat(phschan,sumarm,0)
          call calcphs(1)
          call close(4)
          numswp2=(numswp+1)/2
          refphase=ssum(numswp,phase,1)/float(numswp)
          write (1,1111) refshot,refphase
1111      format ('refshot =',a8,' refphase = ',e11.4)
          call getfreq
          prvref=refshot
      endif
      endif
*         Open plasma shot and read digitizer data.
      call tolower(shot,8)
      call open(4,shot,0,len)
      if (len.eq.-1) goto 400
      call readdat(timchan,time,1)
      if (found.eq.0) goto 390
```

```
      ier=icompare(timchan,phschan)
      if (ier.ne.0) then
          call mmxerr(ier,0,0,'icompare')
          call exit
      endif
      call getswps
      call readdat(phschan,sumarm,0)
      if (found.eq.0) goto 390
      if ((job.eq.'cal').or.(job.eq.'rods')) then
          calind=calind+1
          if (calind.eq.1) call getfreq
          call calcphs(0)
          avephs=ssum(numswp,phase,1)/float(numswp)
          write (1,1112) shot,avephs
 1112     format ('shot =',a8,' avephs = ',e11.4)
          if (job.eq.'cal') then
              ier=ixtract(shot,phschan,'PSHIFT','REAL',
     $                caldat(1,calind))
              caldat(2,calind)=avephs*rad2deg
          else
              ier=ixtract(shot,phschan,'PATHCM', 'REAL',pathcm)
              ier=ixtract(shot,phschan,'EPSILON','REAL',epsilon)
*     Calculated expected phase shift, based on epsilon & pathcm.
              caldat(1,calind)=(sqrt(epsilon)-1)*K0*pathcm*rad2deg
              caldat(2,calind)=avephs*rad2deg
          endif
      else
          call calcdns
      endif
 390  call close(4)
 400  call incfil(shot)
      call tolower(shot,8)
      if (shot.le.shotend) goto 100
      if (job.eq.'rods') then
          mrkflg=-1
          labnum=5
      endif
      if (job.eq.'cal') then
          mrkflg=1
          labnum=3
      endif
      if ((job.eq.'rods').or.(job.eq.'cal')) then
          shot=shotbeg
          plttyp=3
          call wrtlab(labnum,calind,'DETERMIN')
          write (5,5000) (caldat(1,i),i=1,calind)
          write (5,5000) (caldat(2,i),i=1,calind)
 5000     format (6e11.4)
      endif
      goto 10
 800  write (1,8000)
 8000 format ('Normal exit.')
      call exit
 900  write (1,9000)
```

```
9000  format ('No reference shot found.')
      goto 990
910   call mmxerr(ier)
990   write (1,9900)
9900  format ('Abnormal exit.')
      end

*-----------------------------------------------------------------
*                   GET TIMING INFORMATION
*      This subroutine gets the timing information from the
* channel labelled '8MMTIME'.  The real array 'swpbeg' is filled in
* with the beginning points for each sweep.
* METHOD:
* (1) - Fourier transform the timing waveform.
* (2) - Extract the DC level and the period of the fundamental.
*          Take the length of the ramp as 1/3 of the fundamental period.
* (3) - Fitting ramps over two full periods, find the fit which gives the
*          greatest positive slope. Take the reference positions as
*          the crossing of these ramps with the DC level.
* (4) - Continue fitting ramps, advancing 1 cycle for each fit.
* (5) - Interpolate between cycles to get even numbered sweeps.
*-----------------------------------------------------------------
      subroutine getswps
      use vars
      if (popcnt(nmpnts).ne.1) then
          call mmxerr(999,'Nmpnts not a power of 2.$')
          call exit
      endif
*         Step 1: Fourier Transform the timing waveform.
      do 100 i=1,nmpnts
100   time(i)=itime(i)
      call fftrc(time,nmpnts,fourdat,iwk,wk)
      nd2p1=nmpnts/2+1
      xmin=0.
      xmax=1/(2.*smpint)
      if (plotswp.eq.1) then
          call wrtlab(1,nd2p1,'AUTO')
          write (5,1000) (real(fourdat(i)),i=1,nd2p1) .
1000      format (6e11.4)
          frmadv=0
          call wrtlab(2,nd2p1,'AUTO')
          write (5,1000) (aimag(fourdat(i)),i=1,nd2p1)
      endif
*         Step 2: Extract DC level, period, and ramp length.
      dclevel=real(fourdat(1))/nmpnts
      amax=cabs(fourdat(2))
      ipos=2
      do 200 i=3,nmpnts/2+1
      amag=cabs(fourdat(i))
      if (amag.gt.amax) then
          amax=amag
          freqpos=i
      endif
200   continue
```

```
      period=nmpnts/(freqpos-1)
      if (ipos.gt.maxswp) then
          call mmxerr(999,'Too many sweeps$')
          call exit
      endif
      lenramp=period/3
*         Step 3: Fit ramps over 2 cycles.
      rampmax=0.
      do 210 i=1,nmpnts
210   wk(i)=i
      do 300 i=1,period*2
      call linfit
     $    (wk(i),time(i),sigmay,lenramp,0,a,sigmaa,b,sigmab,r)
      if (b.gt.rampmax) then
          xint=a
          rampmax=b
      endif
300   continue
      swpbeg(1)=(dclevel-xint)/rampmax
      numswp=1
*         Step 4: Continue fitting ramps, advancing 1 cycle for each fit.
400   i=int(swpbeg(numswp))+period
      if (i+period.gt.nmpnts) goto 500
      numswp=numswp+2
      call linfit
     $    (wk(i),time(i),sigmay,lenramp,0,a,sigmaa,b,sigmab,r)
      swpbeg(numswp)=(dclevel-a)/b
      goto 400
*         Step 5: Fill in the even numbered sweeps by interpolation.
500   do 510 i=2,numswp,2
510   swpbeg(i)=(swpbeg(i+1)+swpbeg(i-1))/2.
      period=nmpnts/(2*(freqpos-1))
      return

900   write (1,9000)
9000  format (/5x,'Error on timing waveform.')
      errflg=1
      pause
      return
      end

*-----------------------------------------------------------------
*         CALCULATE DENSITY
*-----------------------------------------------------------------
      subroutine calcdns
      use vars
      call calcphs(0)
*         Convert phase differences into plasma density results.
      do 100 i=1,numswp
      deltaw=c*(K0-phase(i)/diam)
100   phase(i)=scfctr*(omsqr-deltaw*deltaw)
      call wrtlab(4,numswp,'DETERMIN')
      do 200 i=1,numswp
200   swpbeg(i)=swpbeg(i)*smpint-pttime
```

```
      write (5,2000) (swpbeg(i),i=1,numswp)
      write (5,2000) (phase(i),i=1,numswp)
2000  format (6e11.4)
      return
      end

*-----------------------------------------------------------------
*         READ DIGITIZER DATA FROM CHANNEL chnnam INTO array
*-----------------------------------------------------------------
      subroutine readdat(chnnam,array,hdrflag)
      use header
      use vars
      integer chnnam,array(maxpnt),hdrflag
      found=1
      if (hdrflag.eq.0) goto 100
      read (4,1000) datim,filnam,filext,name,
     $    slot,smpcod,reclen,pretrg,devtyp,crate,reserv
1000  format (6i2,/,a8,a3,/,4(8a8,/),7(32i2,/),32i2)
100   ier=iclcprm(chnnam,0,smpint,nmpnts,pttime,nmpt)
      if (ier.ne.0) goto 910
      ier=ipsnfile(4,chnnam,0)
      if (ier.ne.0) goto 910
      read (4,1010) (array(j),j=1,nmpnts)
1010  format (16i4)
      return
*         MMXERR exit
910   if (ier.eq.302) then
          found=0
          return
      endif
      call mmxerr(ier)
      call exit
      end

*-----------------------------------------------------------------
*         CALCULATE THE PHASES FOR THE SWEEPS
*-----------------------------------------------------------------
      subroutine calcphs(refflag)
      use vars
      integer refflag
      real fitarr(3),fitdrv(3)
      external calcx2
      color='black'
      ier=ixtract(shot,phschan,'COLOR','ALPHA',color,len)
      call tolower(color,8)
      if (color.eq.'black') then
          do 110 i=1,nmpnts
110       sumarm(i)=256.-float(isumarm(i))
      else
          do 120 i=1,nmpnts
120       sumarm(i)=isumarm(i)
      endif
*             INITIALIZE GUESSES FOR PHASE FITTING ROUTINE
*     fitarr(1)=0.
```

```
      fitarr(2)=twopi/(period*smpint)
      fitarr(3)=ssum(nmpnts,sumarm,1)/nmpnts
      x2=fitarr(2)
      x3=fitarr(3)
      nn=3
      call filter
      do 200 i=1,numswp
      ibegin=swpbeg(i)
      iend=ibegin+period-1
      numeval=0
*         Invoke IMSL minimization routine to minimize chi-squared.
      call zxcgr(calcx2,nn,acc,maxfn,dfpred,fitarr,fitdrv,chisq,wk,ier)
      if (ier.eq.33) call exit
      if (ier.ne.0) then
          write (1,2000) i,ibegin
2000      format('i,ibegin',2i5)
          fitarr(2)=x2
          fitarr(3)=x3
          phase(i)=phase(i-1)
      else
      phase(i)=fitarr(1)+fitarr(2)*(swpbeg(i)-float(ibegin))*smpint
      endif
      if (a.lt.0) phase(i)=phase(i)+pi
200   continue
      if (refflag.eq.1) return
*         Force continuity of phase, beginning with the last sweep.
* No sweep is allowed to be more than pi away from the previous sweep.
      phase(numswp)=signeps*amod(refphase-phase(numswp),twopi)
      if (phase(numswp).lt.-pi) phase(numswp)=phase(numswp)+twopi
      do 390 i=numswp-1,1,-1
      phase(i)=signeps*(refphase-phase(i))
310   delphase=phase(i)-phase(i+1)
      if (delphase.gt.pi) then
          phase(i)=phase(i)-twopi
          goto 310
      endif
      if (delphase.lt.-pi) then
          phase(i)=phase(i)+twopi
          goto 310
      endif
390   continue
      return
      end

*-------------------------------------------------------------
*         COMPUTE CHI-SQUARED AND ITS DERIVATIVES
*     Chi-squared is the sum of the differences squared
*     between the data and a sine wave of the form
*         A sin(wt + phi) + D
*     where A is the amplitude, D is the DC offset,
*     w is the frequency, and phi is the phase.
*     By minimizing chi-squared, we obtain a best fit for phi.
*-------------------------------------------------------------
      subroutine calcx2(numret,fitarr,chisq,fitdrv)
```

```
      use vars
      real fitarr(3),fitdrv(3),chisq
*     Fitarr and fitdrv provide storage for the following three
*     fitting variables and the derivative of chi-squared with
*     respect to these variables:
*         fitarr(1), fitdrv(1) --- Phi, d(X**2)/d(Phi)
*         fitarr(2), fitdrv(2) --- Omega, d(X**2)/d(Omega)
*         fitarr(3), fitdrv(3) --- D, d(X**2)/d(D)
*     Count the number of evaluations required.
      numeval=numeval+1
      phi=amod(fitarr(1),twopi)
*     The amplitude A can be computed directly from the data
*     and the guesses for the other variables.
      anum=0.
      aden=0.
      do 100 i=ibegin,iend
      thetai=fitarr(2)*(i-ibegin)*smpint
      sinphi=sin(thetai+phi)
      anum=anum+(sumarm(i)-fitarr(3))*sinphi
100   aden=aden+sinphi*sinphi
      a=anum/aden
*         INITIALIZE CHISQ AND THE DERIVATIVES.
      chisq=0.
      fitdrv(1)=0.
      fitdrv(2)=0.
      fitdrv(3)=0.
      do 200 i=ibegin,iend
*         COMPUTE INTERMEDIATE RESULTS
      thetai=fitarr(2)*(i-ibegin)*smpint
      sinphi=sin(thetai+phi)
      cosphi=cos(thetai+phi)
      delta=sumarm(i)-a*sinphi-fitarr(3)
*         COMPUTE DERIVATIVES
      fitdrv(1)=fitdrv(1)+delta*cosphi
      fitdrv(2)=fitdrv(2)+delta*float(i-ibegin)*cosphi
      fitdrv(3)=fitdrv(3)+delta
*         COMPUTE CHISQ
200   chisq=chisq+delta*delta
*         PUT IN THE FINAL FACTORS
      fitdrv(1)=-2.*a*fitdrv(1)
      fitdrv(2)=-2.*a*smpint*fitdrv(2)
      fitdrv(3)=-2.*fitdrv(3)
      return
      end

*------------------------------------------------------------------
*     FILTER PHASE DATA
* The phase data is Fourier transformed.  The components
* with frequency higher than freqmult times the fundamental
* of the timing waveform frequency are set to zero, and the
* inverse transform is then performed.
*------------------------------------------------------------------
      subroutine filter
      use vars
```

```
      call rcfft2(1,1,nmpnts,sumarm,wk,fourdat)
      call rcfft2(0,1,nmpnts,sumarm,wk,fourdat)
      do 100 i=freqmult*freqpos,nd2p1
100   fourdat(i)=cmplx(0.,0.)
      call crfft2(1,-1,nmpnts,fourdat,wk,sumarm)
      call crfft2(0,-1,nmpnts,fourdat,wk,sumarm)
      call sscal(nmpnts,1./(2.*float(nmpnts)),sumarm,1)
      return
      end

*-----------------------------------------------------------------
*       EXTRACT & CALCULATE uWAVE FREQUENCY-DEPENDENT VALUES
*-----------------------------------------------------------------
      subroutine getfreq
      use vars
      ier=ixtract(shot,phschan,'CENFREQ','REAL',cenfreq)
      encode(33,1000,label(1,11)) cenfreq
*     38-15+4+6=33
1000  format ('(C)enter (F)requency = ',f4.1,'(GH)z$')
      encode(28,1010,label(1,14)) freqmult/2.
*     38-15+4+1=28
1010  format ('(F)requency (C)utoff = ',f4.1,'$')
*        Compute the vacuum wavenumber, cgs units.
      omega0=twopi*cenfreq*1.e9
      K0=omega0/c
      omsqr=omega0*omega0
      return
      end

*-----------------------------------------------------------------
*     WRITE MMXPLT1 LABELS
* This subroutine writes labels to the output file in the standard
* form for pre-processed data.
*-----------------------------------------------------------------
      subroutine wrtlab(labnum,numplt,ixtype)
      use vars
      write (5,1000) shot
1000  format (a8)
      do 100 i=1,3
100   write (5,1010) (label(j,lblarr(i,labnum)),j=1,5)
1010  format (5a8)
      write (5,1020) numplt,ixtype,xmin,xmax,0.,0.
1020  format (i8,a8,2e12.5,'DETERMIN',2e12.5)
      write (5,1030) wndwadv,plttyp,mrkflg,isym,timflg,frmadv
1030  format (6i2)
      do 200 i=4,9
200   write (5,1010) (label(j,lblarr(i,labnum)),j=1,5)
*        4 RESERVED LINES
      write (5,1110)
1110  format (///)
      return
      end
```