BNL-101427-2014-TECH

RHIC/RF- 8;BNL-101427-2013-IR

Dynamic performance of the RHIC acceleration RF system

W. Pirki

April 1993

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC)

DISCLAIMER

RHIC/RF-8

RHIC Project

BROOKHAVEN NATIONAL LABORATORY

RHIC/RF Technical Note No. 8

Dynamic Performance of the RHIC Accelerating RF System

W. Pirkl

April 1993

DYNAMIC PERFORMANCE OF THE RHIC ACCELERATING RF SYSTEM

Werner Pirkl

Brookhaven National Laboratory
April 1993

Introduction

The RHIC accelerating rf system operates at 26.7 MHz and has to provide, as its name suggests, the power and agility to accelerate the beams from injection up to the end energy and to hand them off to the storage rf system. The machine cycle mandates beam gymnastics for ions at transition crossing, and for ions and protons alike at beam handoff.

It is therefore required that the accelerating cavities be able to perform an amplitude step from zero to full voltage in about one millisecond, and to reverse phase at full voltage within about 100 microseconds.

This note discusses methods to simulate the dynamic behavior of the accelerating cavity system and gives results for the amplitude and phase transient in response to fast changes of the reference signal. The frequency variation during ion acceleration is slow in comparison and can be taken care of by a (mechanical) tuner; implications will not be considered in this context.

Simulation by PSPICE

PSPICE is a powerful circuit simulation program that allows analysis in the frequency as well as the time domain. A simplified circuit model for the accelerating rf system was established and system responses evaluated (Annex 1). Frequency response in open-loop and in closed-loop (i.e. with rf feedback) was obtained for the overall circuit. It is well known from basic feedback theory that the possible loop gain is limited by the overall delay in the feedback chain. This delay has to be a multiple of $1/(2 * frf) = 18.73$ nsec since the nominal phase at the center operating frequency should initially be set at 180 degrees (using a phase reversal transformer if necessary).

A delay of 6 * 18.73 = 112.7 nsec has been assumed in the simulation, since this covers a 500 W ENI drive amplifier (33 nsec) plus 2 * 10 m of high-speed foam cable (2 * 36.7 nsec).

The PSPICE results confirm predictions obtained with the help of the NICHOLL's diagram, namely that a loop gain of 40 dB leads to a near-ideal overall response whereas a loop gain of 50 dB results in a peaking of 5 dB at the limits of the passband (see Annex 1).

Simulation of the transient response was less convenient for the following reasons:

- -difficulties arise to simulate adequate limiting characteristics to model saturation of the amplifier.
- -the output of the transient analysis traces the fine-grain structure of each rf cycle but fails to deliver the global parameters of interest, namely amplitude and phase. These have to be elaborated manually or by some kind of user-written post-processor that follows each individual rf cycle and evaluates its points of zero crossing and maxima/minima. The natural time constant of a cavity with Q=10000 amounts to Q/PI = 3180 cycles, so following a transient with 2 * 3 = 6 time constants means evaluation of some 20000 cycles, each with some 2600 points to assure proper PSPICE convergence. Working through megabyte output files or patching together meters of graphics output is highly inefficient to say the least.
- -it takes more than 80 minutes on a 33MHz 486 PC to simulate the response corresponding to a single cavity time constant.

Simulation of the high-Q cavity in the time domain by PSPICE has therefore been abandoned.

<u>Evaluation of the waveform envelope by direct integration</u>

The ultimately wanted information is the envelope of the rf waveform, i.e. the amplitude/phase response. In a single RLC resonator, which is a sufficiently complete model of the accelerating cavity for simulation of the long-range wake field, the time dependance of the envelope is given by the differential equation

$$dU/dt = (I - U * Y)/(2 * C)$$

with U the voltage envelope, I the driving current, Y the cavity conductance (all complex quantities) and C the cavity capacitance.

A program has been written in Q-Basic that integrates the above equation numerically using the Runge-Kutta algorithm. The code is a modification of a vintage program written in an early version of Turbo-Pascal lacking adequate graphics capabilities.

A listing of the program is given in Annex 2. It treats real and imaginary parts separately and integrates in time steps equal to one rf cycle (can also be chosen longer without sacrifice in precision). The code includes the hardware-related subroutines MODULATE (to simulate rf feedback and a limiting amplifier), GENERATOR (to provide the rf drive with amplitude and phase jump) as well as the algorithm-related subroutines RUNGEKUTTA (the differential equation solver) and VARDOT (to deliver the time-derivatives of the variables).

The program has been tested against output of another program, simulating an RLC circuit by superposition of 20000 sequential rf drive cycles whose response can be checked analytically; perfect agreement has been found.

Dynamic behavior of the accelerating system

A.General

The figures show amplitude and phase of the rf waveform for typical operating conditions, with the output organized as follows (see fig 1):

All parameters are listed in the upper left corner, defining

- Cavity (q, ruponq, frf, deltaf); the cavity is detuned by -500 Hz with respect to the driving rf frequency in all cases.
- Phase reversal: the rf drive remains at 0 degrees from 0 to start (450 microseconds), ramps linearly to +180 or -180 between start and stop (450 and 550 microseconds) and remains there until total (1000 microseconds).
- Rf drive irf (1mA throughout) together with the hard-limiting threshold imax
- Rf feedback loop gain
- Initial conditions of the cavity voltage (real and imaginary part =0)
- plotscale 1000 V (see below).

A polar plot of the voltage envelope is given in the upper right corner, extending from -plotscale to +plotscale with zero voltage in the center.

A rectangular plot of amplitude and phase is given on the bottom part. The horizontal scale is from 0 to total (1000 microseconds), the vertical scale 0 to plotscale for the amplitude and -180 to +180 for the phase.

Since the cavity shunt impedance is 700 kiloOhm in all cases, "full" response for the 1 mA drive is 700 V. This corresponds to 7 units on both the polar and on the cartesian plot. The geometric length of the amplitude response is also arranged to be equal in both displays.

B.Response without rf feedback

Fig.1 shows the response of a cavity that is detuned by -0.375 half-bandwidths, which corresponds to a steady-state phase of -20.5 degrees. The excitation is the standard amplitude step with a subsequent phase shift of -180 degrees, i.e. in the same direction as the steady-state phase. The passband of the cavity is centered slightly below the steady state signal, and the main spectral components of the FM signal that is equivalent to the phase shift lie on the same side. The cavity voltage settles first to about 95% of "full" response, droops to approximately 50% during the phase shift and resettles finally again to 95% at the inverted phase.

Fig.2 depicts the same initial conditions, but with a phase shift in the direction opposite to the steady-state phase. The droop is now much more pronounced in depth and duration, since the main spectral components of the exciting signal are offset on the other side of the steady-state line than the cavity passband. Transition crossing with this waveform would lead to large distortions of the bucket and subsequent beam losses.

C.Response with rf feedback

Fig.3 represents the cavity response with the rf feedback loop closed at a gain of 100, under the same conditions as fig.2 above, with phase shift in the positive (unfavorable) direction. Initial settlement is much faster and up to "full" response at zero phase despite of the cavity detuning.

The droop during the phase shift is reduced but still prohibitively high. The reason is saturation of the amplifier, whose maximum output current is limited to 1.5 times the nominal current for full response.

Figs. 4 to 6 refer to amplifier overdrive capabilities of 2.24, 3.16 and 4.5 respectively. This corresponds to rf peak driver powers of 5, 10 and 20.25 times the nominal drive power of about 100 W. The droop is reduced accordingly until the ideal response is attained.

The slew-rate of the amplitude is also reduced in proportion to the available peak current. For the overdrive capability 4.5 that assures ideal phase response for a 100-microsecond reversal time, the amplitude can be raised to 100% within less than 40 microseconds.

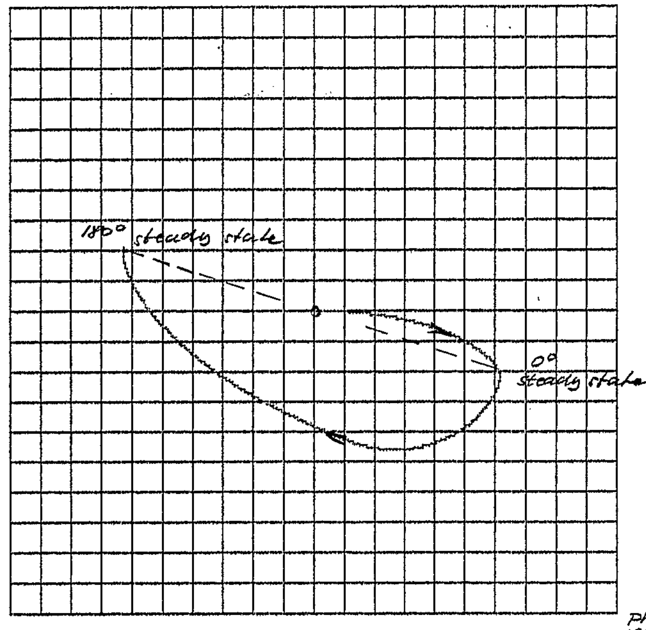
CONCLUSION

Rf feedback can be implemented with a loop gain of at least 100 which leads to fully satisfactory dynamic characteristics for low-to-medium levels. Maintaining full rf amplitude during phase reversal is however critically dependent on the available overdrive capability of the amplifier, which has to be as high as 4.5 to meet the specifications under worst case conditions.

Rf driver powers in excess of 2 kW are needed to make full use of the final tube's capabilities if class-B bias is maintained throughout. The tube can however be driven into class A for the short periods where full peak power capability is required; this saves drive power and can be achieved by pulsing the grid bias.

Amplitude requirements concerning slew-rate are less critical than those concerning phase reversal. The former are automatically covered if the latter are met.

rupong	70
q	10000
frf	2.67E+07
deltaf	-500
phistart.deg	0
phistop.deg	-180
tstart	.00045
tstop	.00055
ttotal	.001
irf	.001
imax	.0015
loopgain	1
u0R	0
u0I	0
plotscale	1000



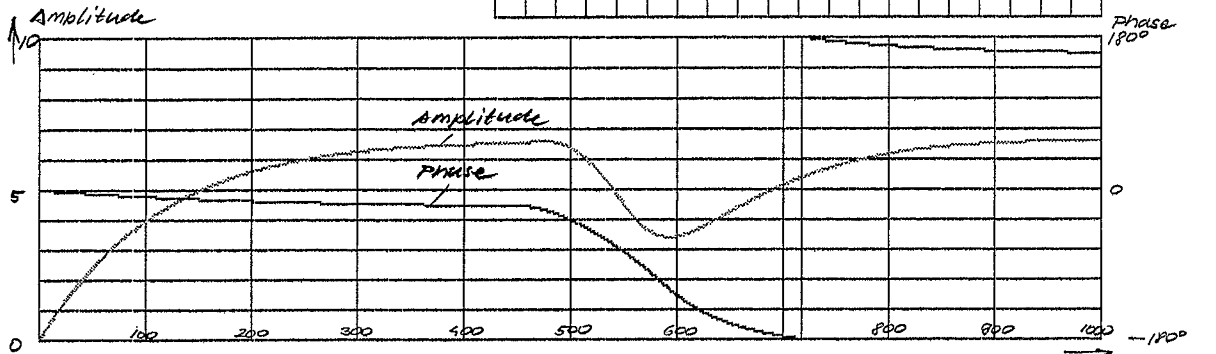
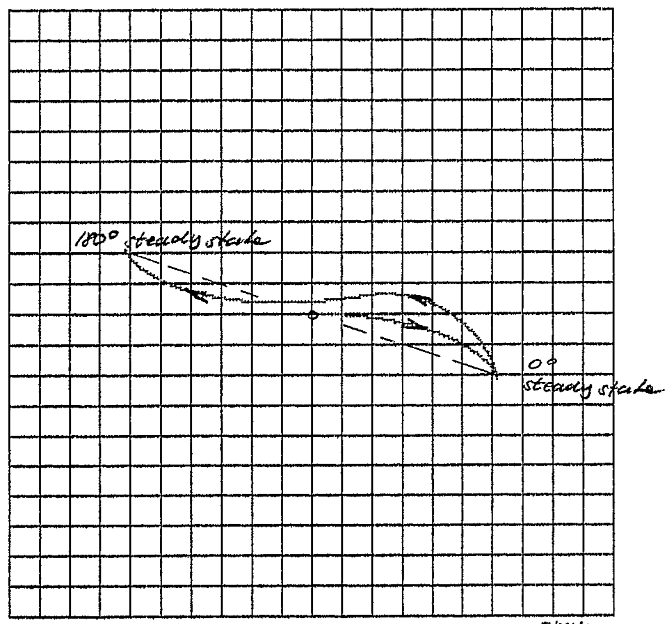


FIG 1: OPEN LOOP / Cavity phase - / Phase reversal -

rupong	70
q	10000
frf	2.67E+07
deltaf	-500
phistart.deg	0
phistop.deg	180
tstart	.00045
tstop	.00055
ttotal	.001
irf	.001
imax	.0015
loopgain	1
u0R	0
u0I	0
plotscale	1000



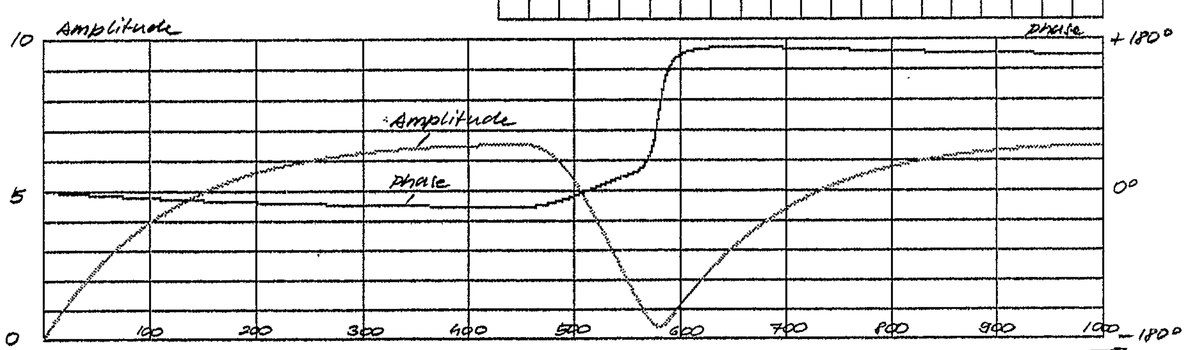
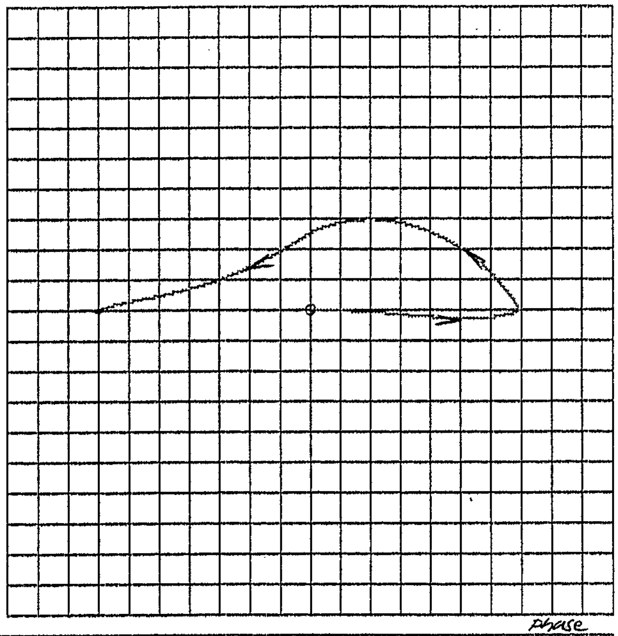


FIG 2: OPEN LOOP / Cavity phase - / Phase reversal +

ruponq	70
q	10000
frf	2.67E+07
deltaf	-500
phistart.deg	0
phistop.deg	180
tstart	.00045
tstop	.00055
ttotal	.001
irf	.001
imax	.0015
loopgain	100
u0R	0
u0I	0
plotscale	1000



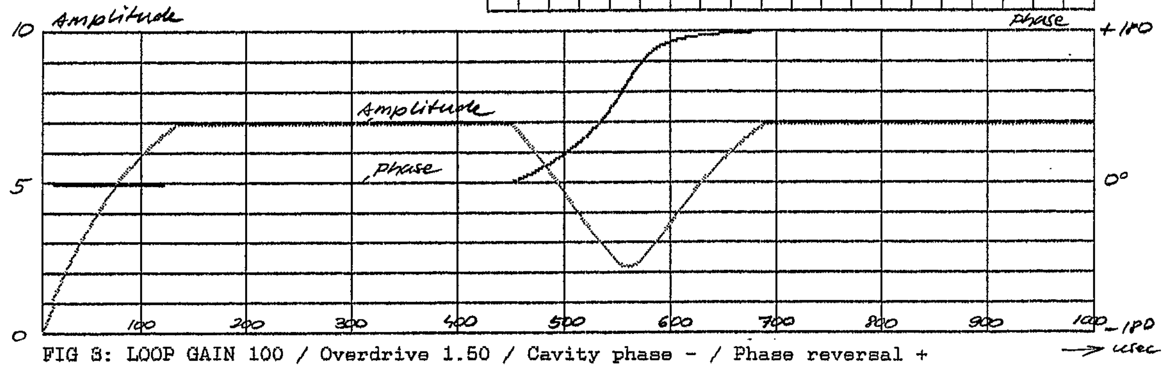
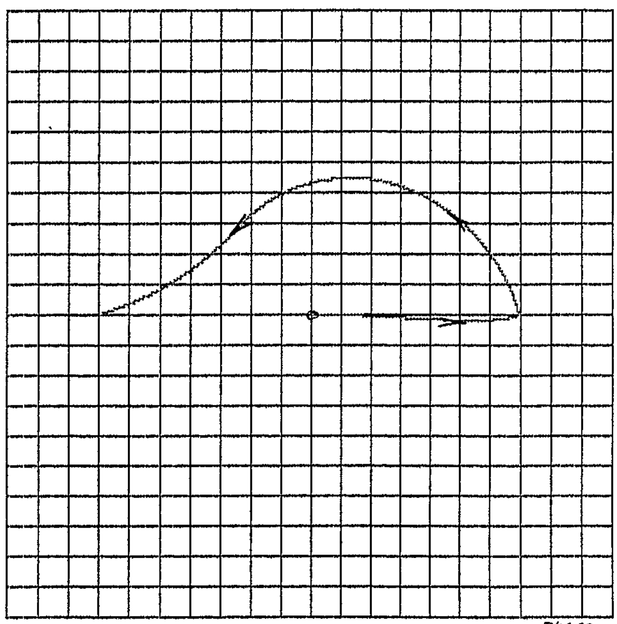


FIG 3: LOOP GAIN 100 / Overdrive 1.50 / Cavity phase - / Phase reversal +

ruponq	70
q	10000
frf	2.67E+07
deltaf	-500
phistart.deg	0
phistop.deg	180
tstart	.00045
tstop	.00055
ttotal	.001
irf	.001
imax	.00224
loopgain	100
u0R	0
u0I	0
plotscale	1000



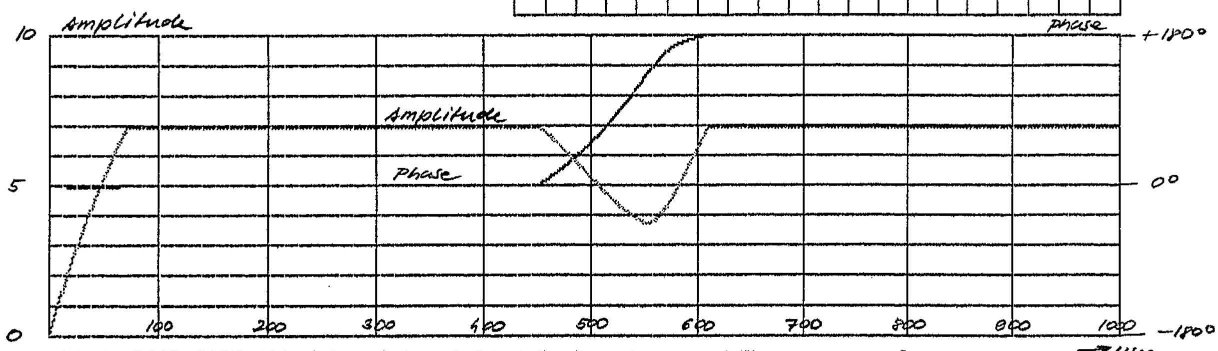
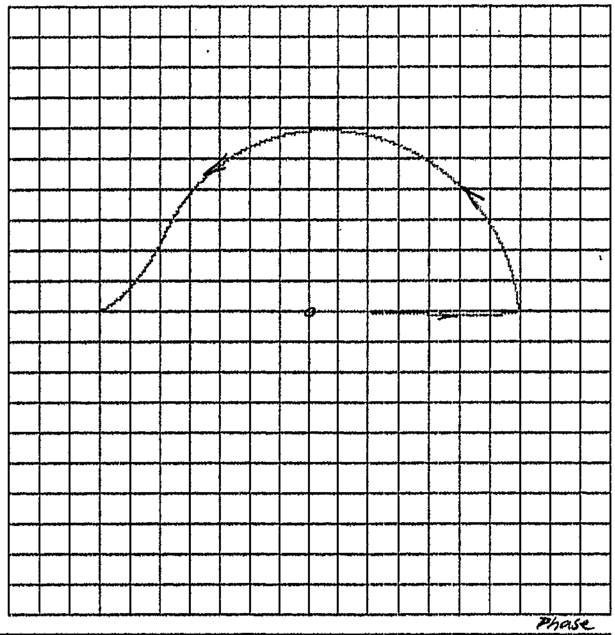


FIG 4: LOOP GAIN 100 / Overdrive 2.24 / Cavity phase - / Phase reversal +

ruponq	70
q	10000
frf	2.67E+07
deltaf	-500
phistart.deg	0
phistop.deg	180
tstart	.00045
tstop	.00055
ttotal	.001
irf	.001
imax	.00316
loopgain	100
u0R	0
u0I	0
plotscale	1000



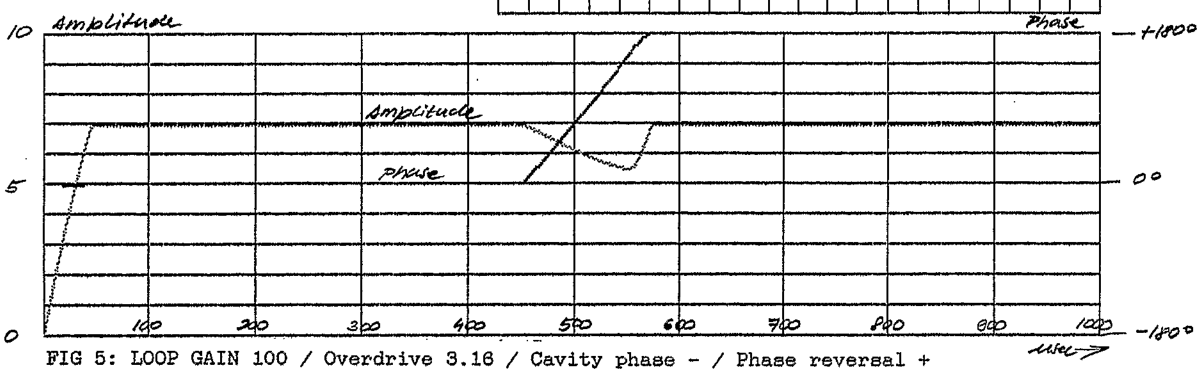
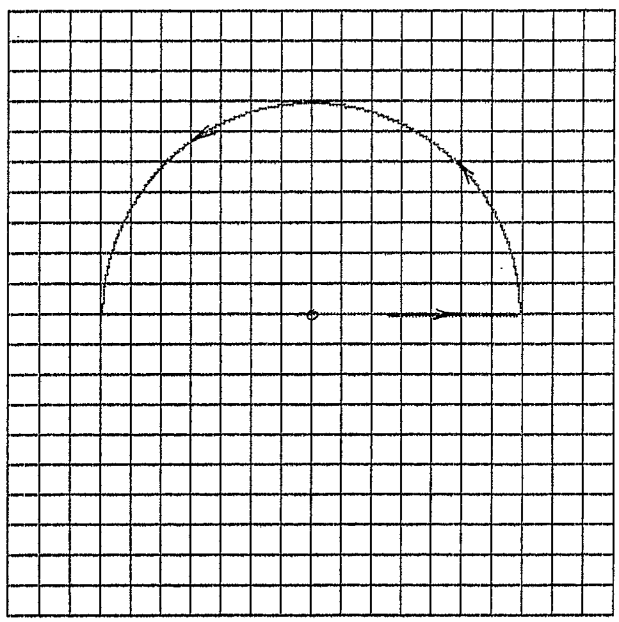


FIG 5: LOOP GAIN 100 / Overdrive 3.16 / Cavity phase - / Phase reversal +

ruponq	70
q	10000
frf	2.67E+07
deltaf	-500
phistart.deg	0
phistop.deg	180
tstart	.00045
tstop	.00055
ttotal	.001
irf	.001
imax	.0045
loopgain	100
u0R	0
u0I	0
plotscale	1000



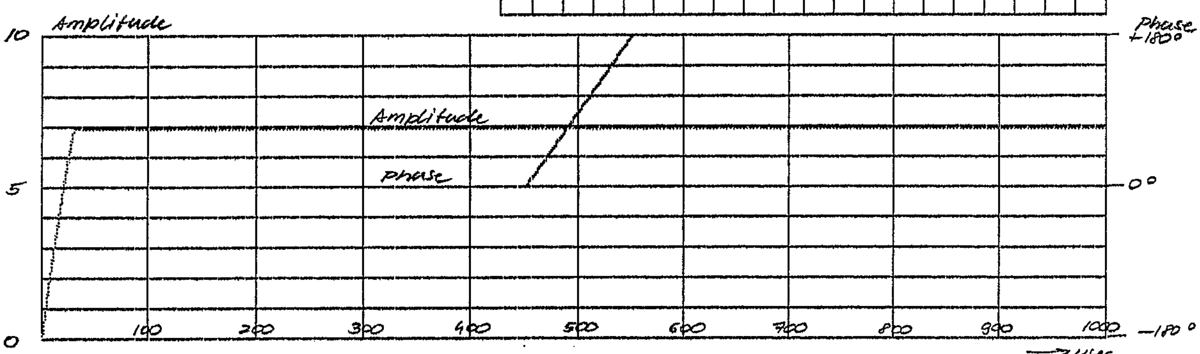
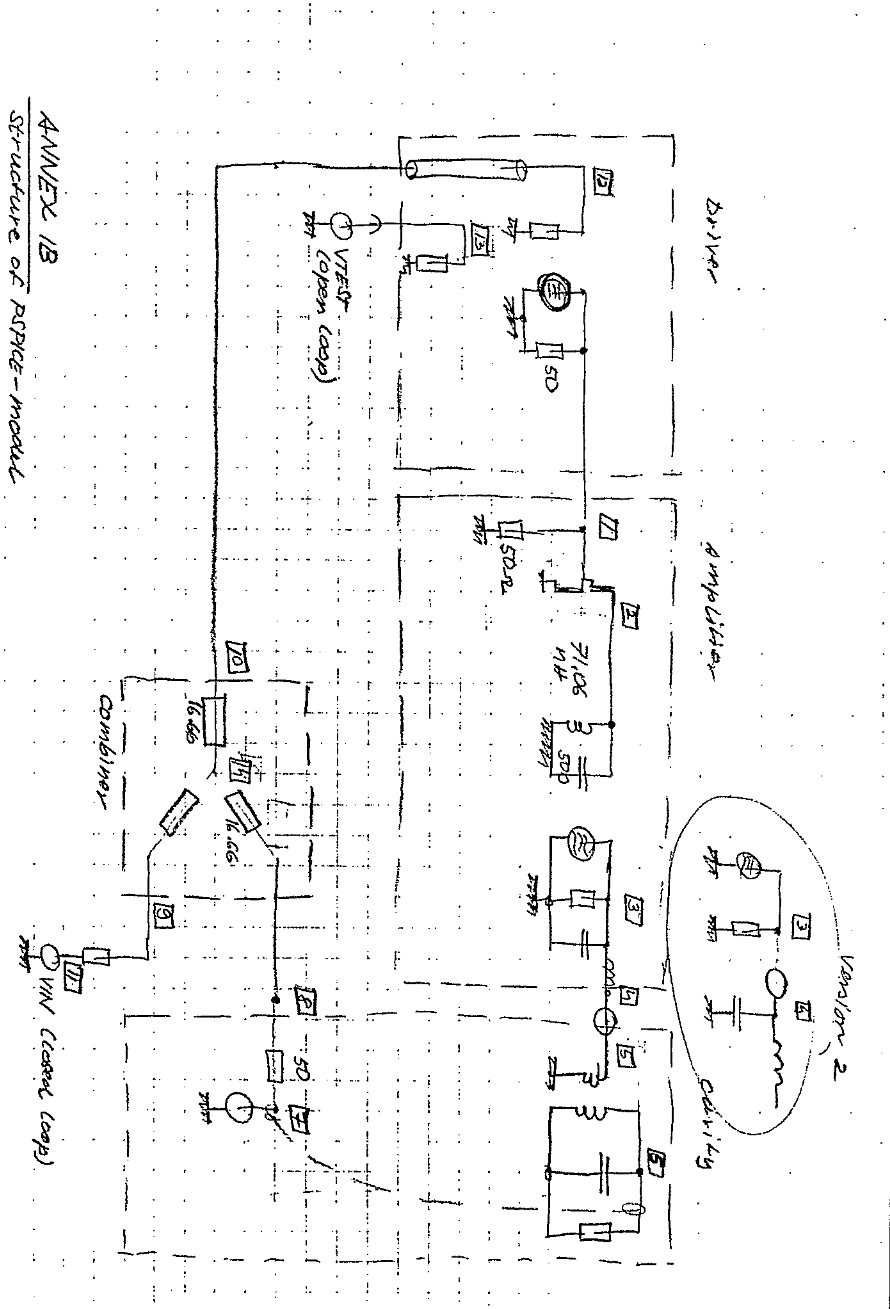


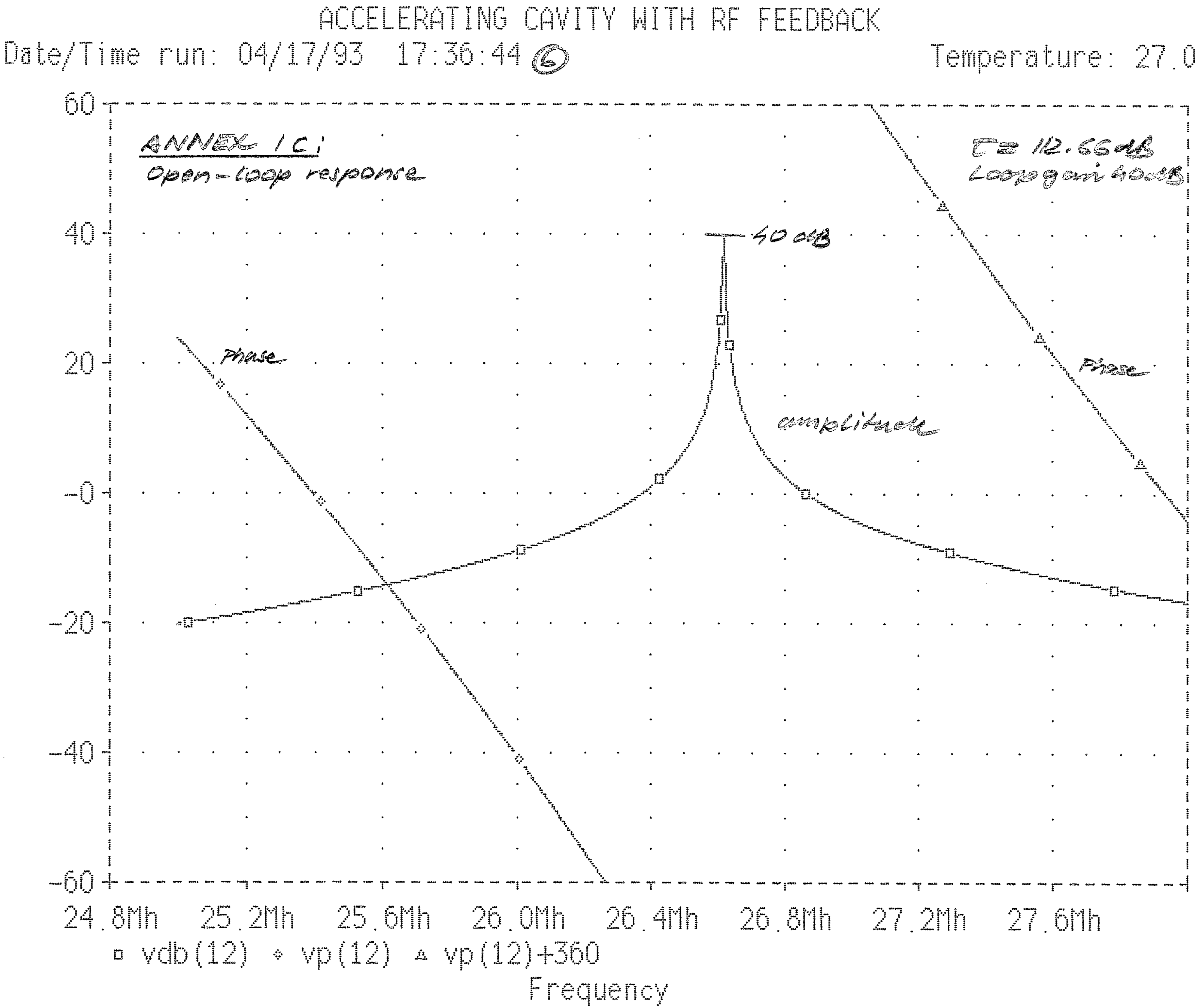
Fig 6: LOOP GAIN 100 / Overdrive 4.50 / Cavity phase - / Phase reversal +



ANNEX 1B
Structure of PSPICE-model

ANNEX 1A:
PSPICE circuit file

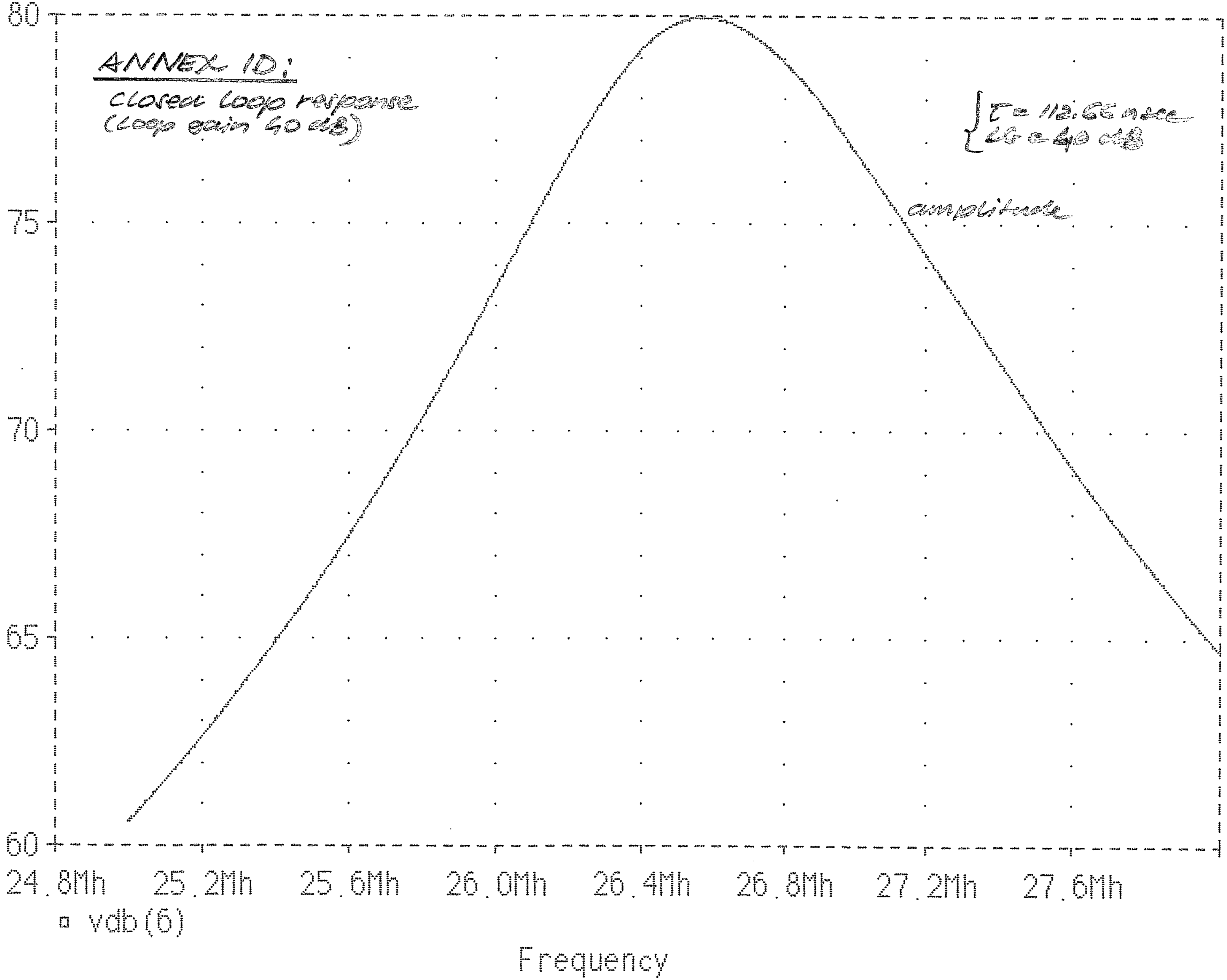
```
ACCELERATING CAVITY WITH RF FEEDBACK
*********************************************
.param ucav=200E3, pcav=50E3, uamp=15E3     ;*
.param fcav=26.7E6, r_q=70                   ;*
.param uclip=300                             ;*
*********************************************
      .param pi= 3.14159, omega={ 2 * pi * fcav}
      .param ccav={ 1 /(r_q * omega)}, lcav={r_q /omega}
 ******** Final amplifier
rina   1   0   50
   ********* input transformer 2/1
   lt11  1    100  1E-3
   lt12  2    1     1E-3
   kt1   lt11  lt12   0.9999999999
   rsep  100   0    1E-6 ; *** ground loop separation
   ******** grid circuit
   lgrid     2    0   71.06E-9
   cgrid     2    0   500E-12
   ******** grid clipper
*   dclip_p  2   200  diode
*   dclip_n  2   201  diode
*   rp_p     2   200  1E2    ; * allows bias calculation
*   rp_n     2   201  1E2
*   vclip_p .200   0    DC    { uclip}
*   vclip_n  201   0    DC    {-uclip}
   ******** power tube
   gtube     3   0   2   0    37.5E-3 ;***average tube Gm class B
   rtube     3   0    50E3
   ******** output circuit
   viprobe  3   4    AC  0
   ctube    4   0    80E-12
   lfeed    4   5    70E-9
********** cavity
lres     6   0  {lcav}
cres     6   0  {ccav}
rres     6   0  {ucav * ucav / (2 * pcav)}
lt21     5   0  {lcav /pwr(ucav/uamp , 2)}
kt2     lres  lt21  0.999999
eloop   7  0  6  0  1E-4  ;***feedback return path
rloop   7  8  50
********* combiner
rcom1   8   14  16.6666
rcom2   9   14  16.6666
rcom3   10  14  16.6666
******** driver
tdelay   10   0   12   0   z0=50   td=112.66E-9;  *** total delay
rindrive 12  0   50
****************************** open loop, output at point 12
*    gdrive  1   0    13    0   -64.03 ; * overall loop gain 50 db @ 202.5
*    vtest   13  0    ac    1
*    vin     11  0    ac    0         ; ** set to zero to terminate combiner
****************************** closed loop
   gdrive  1   0    12   0  -64.03
   vtest  13   0    ac  0
   vin    11   0    ac  1
******************************
 routd    1   0    50
 rtest    13  0    50
 rsource  9  11    50
****************************
 .AC  lin  1000  25.00E6   28.0E6
 .MODEL diode d
 .PROBE
```



ACCELERATING CAVITY WITH RF FEEDBACK
Date/Time run: 04/17/93 17:36:44
Temperature: 27.0
ANNEX 1C:
Open-loop response
60
40
20
-0
-20
-40
-60
phase
amplitude
40 dB
24.8Mh
25.2Mh
25.6Mh
26.0Mh
26.4Mh
26.8Mh
27.2Mh
27.6Mh
vdb(12)
vp(12)
vp(12)+360
Frequency

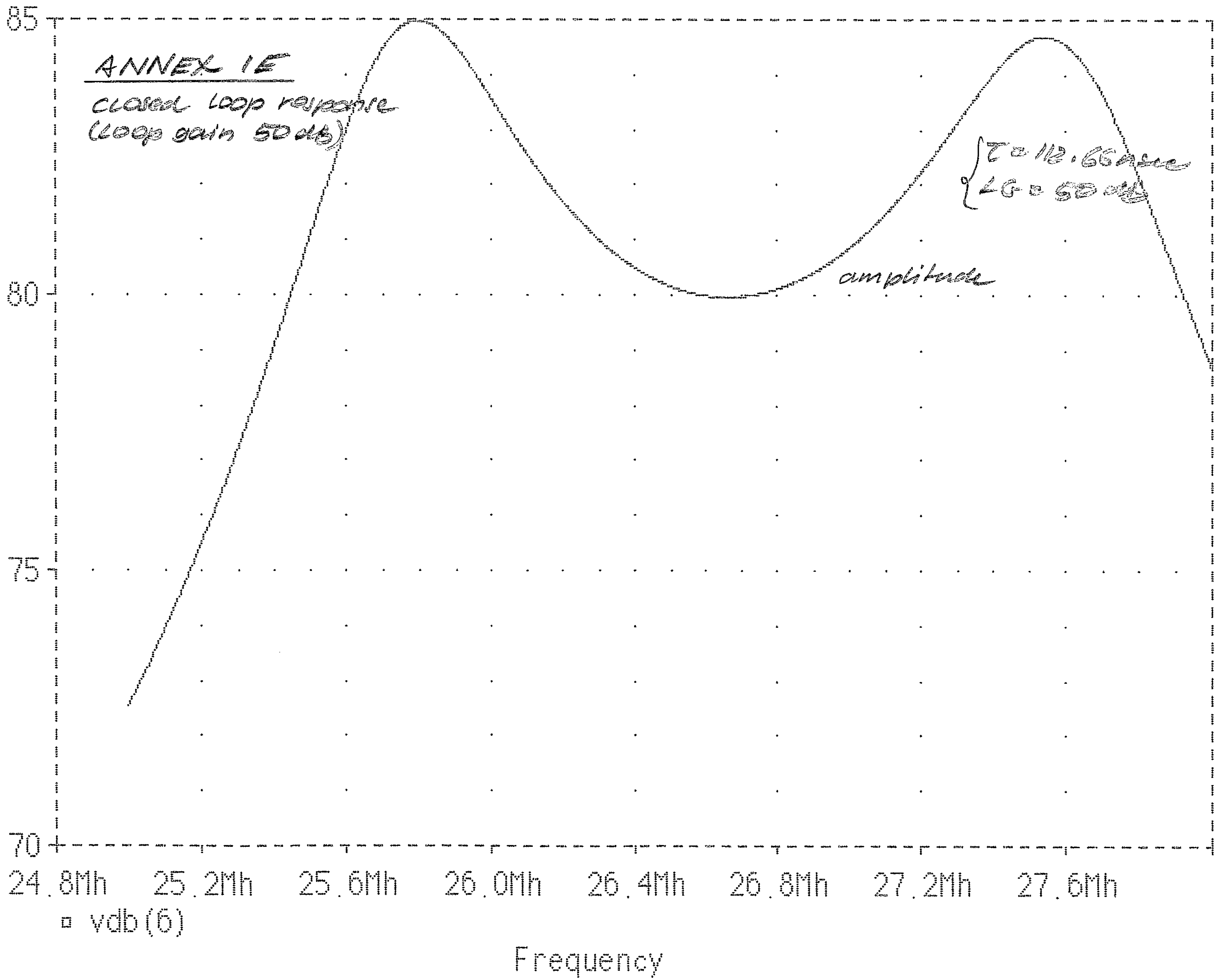
ACCELERATING CAVITY WITH RF FEEDBACK

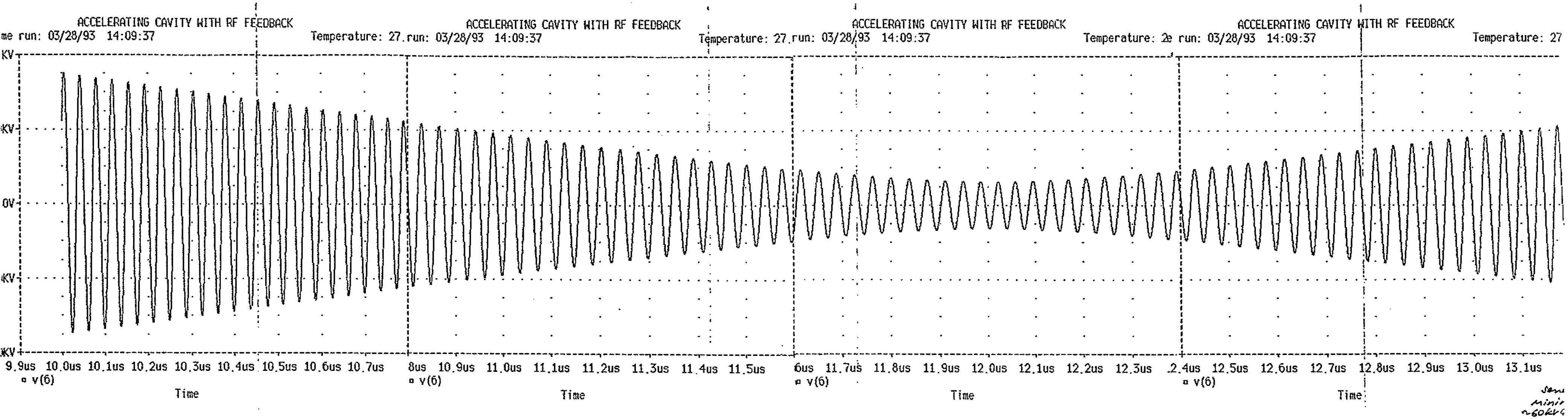
Date/Time run: 04/17/93 17:51:03 (9) Temperature: 27.0



ACCELERATING CAVITY WITH RF FEEDBACK

Date/Time run: 04/17/93 17:46:40 Temperature: 27.0





ANNEX IF!

PSPICE analysis time domain
(4 output screens combined)

ANNEX 2

Runge-Kutta Integration Program file (QBasic)

pg 1/4

```
DECLARE SUB MODULATE ()
DECLARE SUB RUNGEKUTTA (t!)
DECLARE SUB GENERATOR (t!)
DECLARE SUB VARDOT ()
DECLARE SUB CRECTOPOL (x, y, mag, ang)
'************************** PROGRAM RESRUKU *********************************
'*       Evaluates RESonatorRESponse to pulsetrain                            *
'*               using Runge-Kutta integration                                *
'*                      W.Pirkl, April93                                      *
'* Purpose: Study of RHIC accelerating system with rf feedback (envelopes)   *
'******************************************************************************
GOSUB initialize
GOSUB getdata
GOSUB conditiondata
GOSUB drawframe
GOSUB plotfunction
GOSUB resetscreen
END

initialize:
  DIM v, va, vd
  OPTION BASE 0
  arrdim = 2
'*** v...array variables, va...array temporay arguments,
'*** vd...array derivatives
'*** arrdim...dimension of array= order of DIFF.EQ.
'*** array(0) holds time, array(1)...array(arrdim) holde variables proper
  CLS
  pi = 3.141593: radian = 180 / pi
  SCREEN 12                    '** 640 pixels hor, 480 vert (GATEWAY 486)
  'VIEW (80, 0)-(559, 479)       '** 479/479 pixels in viewport
  'VIEW (120, 40)-(520, 440)     '** 400/400 pixels in viewport
  'VIEW (75, 40)-(558, 440)       '** square after printing
RETURN

getdata:
  ruponq = 70
  q = 10000
  frf = 2.67E+07
  deltaf = -500
  phistart.deg = 0
  phistop.deg = 180
  tstart = .00045
  tstop = .00055
  ttotal = .001
  irf = .001
  imax = .0045
  loopgain = 100
  u0R = 0!: u0i = 0
  plotscale = 1000!'**** in Volts +-hor, +-vert
RETURN

conditiondata:
  fcav = frf + deltaf
  trf = 1 / frf
  c = 1 / (2 * pi * fcav * ruponq)
  rres = ruponq * q
  yr = 1 / rres                            '*** real part of res. conductance
  yi = (frf / fcav - fcav / frf) / ruponq  '*** imaginary part of res. cond.
  phistart = phistart.deg / radian
  phistop = phistop.deg / radian
     FOR n = 0 TO arrdim
        v(n) = 0: va(n) = 0: vd(n) = 0
```

pg 2/4

```
     NEXT n  '*** fill SHARED arrays wity dummy values for definition
RETURN

drawframe:
  '**** text
'  WIDTH 80, 25
  LOCATE 2, 1
  PRINT "ruponq        "; ruponq
  PRINT "q             "; q
  PRINT "frf           "; frf
  PRINT "deltaf        "; deltaf
  PRINT "phistart.deg "; phistart.deg
  PRINT "phistop.deg  "; phistop.deg
  PRINT "tstart        "; tstart
  PRINT "tstop         "; tstop
  PRINT "ttotal        "; ttotal
  PRINT "irf           "; irf
  PRINT "imax          "; imax
  PRINT "loopgain      "; loopgain
  PRINT "u0R           "; u0R
  PRINT "u0I           "; u0i
  PRINT "plotscale     "; plotscale
'****** polar viewport
  pwp = plotscale
  VIEW (270, 10)-(630, 310)
  WINDOW (-pwp, -pwp)-(pwp, pwp)
    FOR n = -pwp TO pwp + .01 STEP pwp / 10   '** 2*10 horizontal lines
        LINE (-pwp, n)-(pwp, n)
    NEXT n

    FOR n = -pwp TO pwp + .01 STEP pwp / 10   '** 2*10 vertical lines
        LINE (n, -pwp)-(n, pwp)
    NEXT n
  xl1 = u0R: yl1 = u0i: PSET (xl1, yl1), 4
'****** cartesian viewport
 VIEW (0, 320)-(630, 470)
   '**** amplitude window
   WINDOW (0, 0)-(ttotal, pwp)
     FOR n = 0 TO pwp + .01 STEP pwp / 10    '*** 10 horizontal lines
        LINE (-pwp, n)-(pwp, n)
     NEXT n

     FOR n = 0 TO ttotal + .01 STEP ttotal / 10 '*** 10 vertical lines
        LINE (n, -pwp)-(n, pwp)
     NEXT n
  xl2 = 0: yl2 = SQR(u0R * u0R + u0i * u0i): PSET (xl2, yl2), 2
  '**** phase window
  WINDOW (0, -180)-(ttotal, 180)
  xl3 = 0: yl3 = 0: PSET (xl3, yl3), 1
RETURN

plotfunction:
    v(1) = u0R: v(2) = u0i: v(0) = 0
  FOR t = 0 TO ttotal STEP trf
    CALL GENERATOR(t)
    MODULATE
    CALL RUNGEKUTTA(trf)
    '***** print results on three curves
       '***polar
       uR = v(1): uI = v(2)
       VIEW (270, 10)-(630, 310)
       WINDOW (-pwp, -pwp)-(pwp, pwp)
       LINE (xl1, yl1)-(uR, uI), 4
       xl1 = uR: yl1 = uI
       '***amplitude
       CALL CRECTOPOL(uR, uI, amplitude, phase)
```

Pg 3/5

```
      VIEW (0, 320)-(630, 470)
      WINDOW (0, 0)-(ttotal, pwp)
      LINE (x12, y12)-(t, amplitude), 2
      x12 = t: y12 = amplitude
      '*** phase
      WINDOW (0, -180)-(ttotal, 180)
      LINE (x13, y13)-(t, phase), 1
      x13 = t: y13 = phase
    'LPRINT : LPRINT " main: t, uRI "; v(0), v(1), v(2)
  NEXT t
 BEEP: BEEP: BEEP
 RETURN

resetscreen:          '** holds screen for dumping until key is pressed
   DO WHILE INKEY$ = "": LOOP
   SCREEN 9: WIDTH 80, 25: VIEW: CLS
RETURN



SUB CRECTOPOL (x, y, mag, ang)
'*** rectangular to polar conversion, angle in degrees
IF x = 0 THEN
   IF y > 0 THEN
      ang = 90
   ELSEIF y = 0 THEN
      ang = 0
   ELSE ang = -90
   END IF
ELSE
    ang = 57.29577 * ATN(y / x)
END IF
IF x < 0 THEN
        IF y < 0 THEN ang = ang - 180 ELSE ang = ang + 180
        END IF
mag = SQR(x * x + y * y)
END SUB


SUB GENERATOR (t)
'*** returns real/imag generator current into SHARED Ir,Ii
'*** phase linearily ramped between tstart and tstop,
'**** between phistart AND phistop

SHARED irf, tstart, tstop, phistart, phistop, isrcR, isrcI

IF t <= tstart THEN
   phi = phistart
            ELSEIF t <= tstop THEN
   phi = phistart + (phistop - phistart) * (t - tstart) / (tstop - tstart)
            ELSE
   phi = phistop
            END IF

   isrcR = irf * COS(phi)
   isrcI = irf * SIN(phi)

END SUB

SUB MODULATE
'*** Modulates drive to resonator, e.g, in rf feedback loop

SHARED v(), isrcR, isrcI, iR, iI, rres, loopgain, imax
'     iR = isrcR: iI = isrcI   '*** testcase
   ierrR = isrcR - v(1) / rres: ierrI = isrcI - v(2) / rres   '*** comparator
   iR = loopgain * ierrR: iI = loopgain * ierrI               '*** gain block
```

Pg 4/5

```
   overdrive = SQR(iR * iR + iI * iI) / imax
   IF overdrive > 1 THEN                                      '*** limiter
                iR = iR / overdrive
                iI = iI / overdrive
            END IF
END SUB

SUB RUNGEKUTTA (t)
'*** calculates one time step t of differential equation
'*** uses v...variables, va...temp.arguments, vd...derivatives
'*** time in v(0), va(0); derivatives generated by SUB VARDOT

SHARED v(), va(), vd(), arrdim
DIM k1(10), k2(10), k3(10)

FOR n = 1 TO arrdim
          va(n) = v(n)
      NEXT n
   va(0) = v(0)
   VARDOT

      FOR n = 1 TO arrdim
          k1(n) = vd(n) * t
          va(n) = v(n) + k1(n) / 2
      NEXT n
   va(0) = v(0) + t / 2
   VARDOT

      FOR n = 1 TO arrdim
          k2(n) = vd(n) * t
          va(n) = v(n) + k2(n) / 2
      NEXT n
   VARDOT

      FOR n = 1 TO arrdim
          k3(n) = vd(n) * t
          va(n) = v(n) + k3(n)
      NEXT n
   va(0) = v(0) + t
   VARDOT

      FOR n = 1 TO arrdim
          v(n) = v(n) + (k1(n) + 2 * k2(n) + 2 * k3(n) + vd(n) * t) / 6
      NEXT n
          v(0) = v(0) + t
END SUB

SUB VARDOT
'*** returns derivatives
'*** aguments in SHARED array VA, results in shared array VD
'*** time in VA(0) is not considered, taken care of by SUB GENERATOR
'LPRINT "ir,ii,yr,yi,c"; : LPRINT USING "   #.###^^^^"; ir; ii; yr; yi; c
SHARED va(), vd(), iR, iI, yr, yi, c
   vd(1) = (iR - va(1) * yr + va(2) * yi) / (2 * c)
   vd(2) = (iI - va(1) * yi - va(2) * yr) / (2 * c)
END SUB
```