Phase Modulation Curves for Slip Stacking

John Marriner
and
Shekhar Shukla

November 12, 1997

We parameterize the low level slip stacking rf curves for the A and B cavities in the following form

$$V^A(t) = A^A(t)\cos\left[2\pi f_0 + \varphi^A(t)\right]$$

$$V^B(t) = A^B(t)\cos\left[2\pi f_0 + \varphi^B(t)\right]$$

The purpose of this document is to specify a parameterization of the phase modulation. The amplitude modulation is not specified here. The phase curve is specified piece-wise by polynomial parameterizations on 9 time segments.

TIMES

The segment lengths are specified by 9 time intervals Δt_i, with the breakpoints defined by

$$t_1 = \Delta t_1$$

$$t_2 = t_1 + \Delta t_2$$

and in general for $2 \le i \le 9$

$$t_i = t_{i-1} + \Delta t_i$$

FREQUENCY CURVES

It is convenient to specify the frequency curves and to compute the phase curves from the frequency curves. We define the frequency curves as the derivative of the phase curves

$$f^A(t) = \frac{1}{2\pi}\frac{d\varphi^A}{dt}$$

$$f^B(t) = \frac{1}{2\pi}\frac{d\varphi^B}{dt}$$

The frequency curves are specified by second order polynomials on each of 9 time segments . Except for the first and last segments, the frequency and its derivative are continuous between segments. After the frequency curves are specified, the phase curves are found by integration with the condition that the phase is continuous between segements and that the final phase is 0.

A plot of the frequency curve with the nominal parameterization is shown in Figure 1. The function of the 9 segments are given below.

Segment 1: Tune cavities before beam
Segment 2: Prepare for injection. Injection of the first batch occurs towards the end of the segment.
Segment 3: Transition to a moving bucket
Segment 4: Accelerate the first batch
Segment 5: Transition to a stationary bucket
Segment 6: Inject the second batch towards the beginning of the segment and allow the batches to slip relative to each other.

Segment 7: Transition to a moving bucket (both batches)
Segment 8: Batches experience a close encounter
Segment 9: Jump to nominal frequency and phase

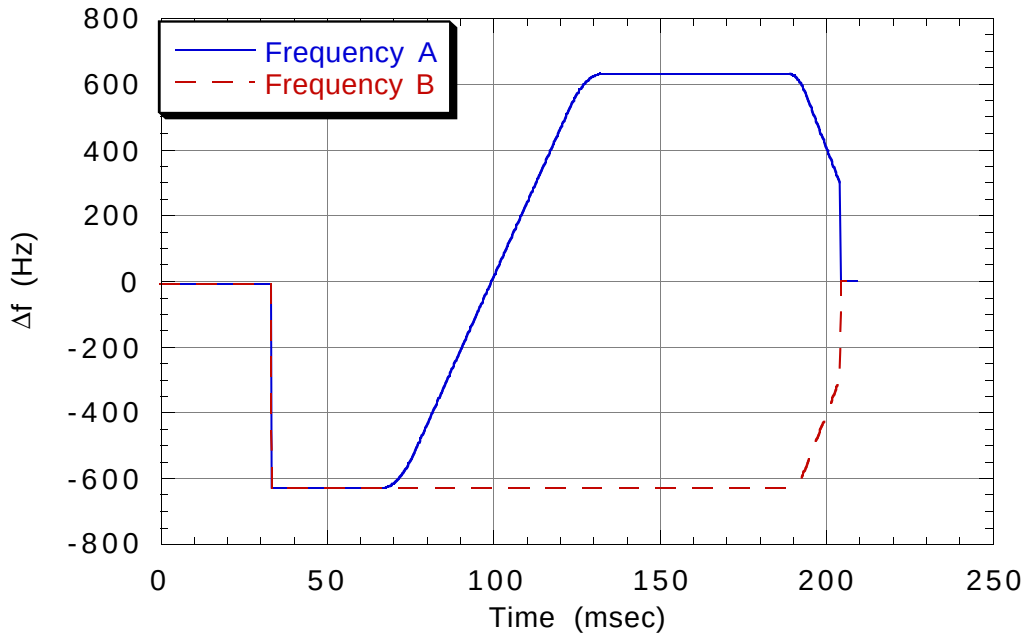

Figure 1. Nominal frequency curve.

The parameterization chosen for the frequency curves is

$$f^{A}(t) = -\Delta f_i \quad 0 \le t \le t_1$$
$$f^{B}(t) = -\Delta f_i$$

$$f^{A}(t) = -\Delta f_0 \quad t_1 \le t \le t_2$$
$$f^{B}(t) = -\Delta f_0$$

$$f^{A}(t) = -\Delta f_0 + \alpha(t - t_1)^2 \quad t_2 \le t \le t_3$$
$$f^{B}(t) = -\Delta f_0$$

$$f^{A}(t) = -\Delta f_0 + \alpha \Delta t_3^2 + 2\alpha \Delta t_3 (t - t_3) \quad t_3 \le t \le t_4$$
$$f^{B}(t) = -\Delta f_0$$

$$f^{A}(t) = \Delta f_0 - \alpha \Delta t_3^2 + 2\alpha \Delta t_3 (t - t_4) - \alpha (t - t_4)^2 \quad t_4 \le t \le t_5$$
$$f^{B}(t) = -\Delta f_0$$

$$f^{A}(t) = \Delta f_0 \quad t_5 \le t \le t_6$$
$$f^{B}(t) = -\Delta f_0$$

$$f^{A}(t) = \Delta f_0 - \beta (t - t_6)^2 \quad t_6 \le t \le t_7$$
$$f^{B}(t) = -\Delta f_0 + \beta (t - t_6)^2$$

$$f^{A}(t) = \Delta f_0 - \beta \Delta t_3^2 + 2\beta \Delta t_7 (t - t_7) + \gamma (t - t_7)^2 \quad t_7 \le t \le t_8$$
$$f^{B}(t) = -\Delta f_0 + \beta \Delta t_3^2 - 2\beta \Delta t_7 (t - t_7) - \gamma (t - t_7)^2$$

$$\Delta f^{A}(t) = 0 \quad t_8 \le t \le t_9$$
$$\Delta f^{B}(t) = 0$$

Continuity requires that

$$\alpha = \frac{\Delta f_0}{\Delta t_3 (\Delta t_3 + \Delta t_4)}$$

$$\Delta t_5 = \Delta t_3$$

The requirements of obtaining the final frequency offset Δf_1 determines β to be

$$\beta = \frac{\Delta f_0 - \Delta f_1 + \gamma \Delta t_8^2}{\Delta t_7 (\Delta t_7 + 2\Delta t_8)}$$

The phase offsets are given by integrating the frequency curves.

$$\phi^{A}(t) = \varphi_1^{A} + 2\pi[-\Delta f_i t] \quad 0 \le t \le t_1$$
$$\varphi^{B}(t) = \varphi_1^{B} + 2\pi[-\Delta f_i t]$$

$$\varphi^{A}(t) = \varphi_2^{A} + 2\pi[-\Delta f_0 (t - t_1)] \quad t_1 \le t \le t_2$$
$$\varphi^{B}(t) = \varphi_2^{B} + 2\pi[-\Delta f_0 (t - t_1)]$$

$$\varphi^{A}(t) = \phi_3^{A} + 2\pi\left[-\Delta f_0 (t - t_2) + \frac{\alpha}{3}(t - t_2)^3\right] \quad t_2 \le t \le t_3$$
$$\varphi^{B}(t) = \phi_3^{A} + 2\pi[-\Delta f_0 (t - t_2)]$$

$$\varphi^{A}(t) = \varphi_4^{A} + 2\pi\left[(-\Delta f_0 + \alpha \Delta t_3^2)(t - t_3) + \alpha \Delta t_3 (t - t_3)^2\right] \quad t_3 \le t \le t_4$$
$$\varphi^{B}(t) = \varphi_4^{A} + 2\pi[-\Delta f_0 (t - t_3)]$$

$$\varphi^A(t) = \varphi_5^A + 2\pi\left[\left(\Delta f_0 - \alpha\Delta t_3^2\right)(t - t_4) + \alpha\Delta t_3(t - t_4)^2 - \frac{\alpha}{3}(t - t_4)^3\right] \quad t_4 \le t \le t_5$$

$$\varphi^B(t) = \varphi_5^B + 2\pi\left[-\Delta f_0(t - t_4)\right]$$

$$\varphi^A(t) = \varphi_6^A + 2\pi\Delta\left[f_0(t - t_5)\right] \quad t_5 \le t \le t_6$$

$$\varphi^B(t) = \varphi_6^B + 2\pi\Delta\left[-f_0(t - t_5)\right]$$

$$\varphi^A(t) = \varphi_7^A + 2\pi\left[\Delta f_0(t - t_6) - \frac{\beta}{3}(t - t_6)^3\right] \quad t_6 \le t \le t_7$$

$$\varphi^B(t) = \varphi_7^B + 2\pi\left[-\Delta f_0(t - t_6) + \frac{\beta}{3}(t - t_6)^3\right]$$

$$\varphi^A(t) = \varphi_8^A + 2\pi\left[\left(\Delta f_0 - \beta\Delta t_7^2\right)(t - t_7) + \beta\Delta t_7(t - t_7)^2 + \frac{\gamma}{3}(t - t_7)^3\right] \quad t_7 \le t \le t_8$$

$$\varphi^B(t) = \varphi_8^B + 2\pi\left[\left(\Delta f_0 - \beta\Delta t_7^2\right)(t - t_7) + \beta\Delta t_7(t - t_7)^2 + \frac{\gamma}{3}(t - t_7)^3\right]$$

$$\varphi^A(t) = 0 \quad t_8 \le t \le t_9$$

$$\varphi^B(t) = 0$$

The constants of integration are determined by continuity and the boundary condition that the phase at the end of the curve be zero, namely

$$\varphi_9^A = 0$$

$$\varphi_8^A = -\Delta\varphi_8^A$$

$$\varphi_7^A = -\Delta\varphi_7^A + \varphi_8^A$$

and in general for $1 \le i \le 7$

$$\varphi_i^A = -\Delta\varphi_i^A + \varphi_{i+1}^A$$

where $\Delta\varphi_i$ is the phase advance accross the *i*th segment. The phase advance is obtained by integrating the frequency curve. Explicit expressions for the $\Delta\varphi_i$ are given below.

$$\Delta\varphi_1^A = 2\pi\left(-\Delta f_i \Delta t_1\right)$$
$$\Delta\varphi_2^A = 2\pi\left(-\Delta f_0 \Delta t_2\right)$$
$$\Delta\varphi_3^A = 2\pi\left(-\Delta f_0 \Delta t_3 + \frac{\alpha}{3}\Delta t_3^3\right)$$
$$\Delta\varphi_4^A = 2\pi\left[\left(-\Delta f_0 + \alpha\Delta t_3^2\right)\Delta t_4 + \alpha\Delta t_3 \Delta t_4^2\right]$$
$$\Delta\varphi_5^A = 2\pi\left[\Delta f_0 \Delta t_3 - \frac{\alpha}{3}\Delta t_3^3\right]$$
$$\Delta\varphi_6^A = 2\pi\left(\Delta f_0 \Delta t_6\right)$$
$$\Delta\varphi_7^A = 2\pi\left(\Delta f_0 \Delta t_7 - \frac{\beta}{3}\Delta t_7^3\right)$$
$$\Delta\varphi_8^A = 2\pi\left[\left(\Delta f_0 - \beta\Delta t_7^2\right)\Delta t_8 - \beta\Delta t_7 \Delta t_8^2 + \frac{\gamma}{3}\Delta t_8^3\right]$$

$$\Delta\varphi_1^B = 2\pi\left(-\Delta f_i \Delta t_1\right)$$
$$\Delta\varphi_2^B = 2\pi\left(-\Delta f_0 \Delta t_2\right)$$
$$\Delta\varphi_3^B = 2\pi\left(-\Delta f_0 \Delta t_3\right)$$
$$\Delta\varphi_4^B = 2\pi\left(-\Delta f_0 \Delta t_4\right)$$
$$\Delta\varphi_5^B = 2\pi\left(-\Delta f_0 \Delta t_3\right)$$
$$\Delta\varphi_6^B = 2\pi\left(-\Delta f_0 \Delta t_6\right)$$
$$\Delta\varphi_7^B = 2\pi\left(-\Delta f_0 \Delta t_7 + \frac{\beta}{3}\Delta t_7^3\right)$$
$$\Delta\varphi_8^B = 2\pi\left[\left(-\Delta f_0 + \beta\Delta t_7^2\right)\Delta t_8 + \beta\Delta t_7 \Delta t_8^2 - \frac{\gamma}{3}\Delta t_8^3\right]$$
$$\Delta\varphi_9^B = 0$$

$$\Delta t_6 = \frac{n_b/2 - \gamma\,\Delta t_8^3/3 + \beta\Delta t_7\left(\Delta t_8^2 + \Delta t_7 \Delta t_8 + \Delta t_7^2/3\right)}{\Delta f_0} - \Delta t_7 - \Delta t_8$$

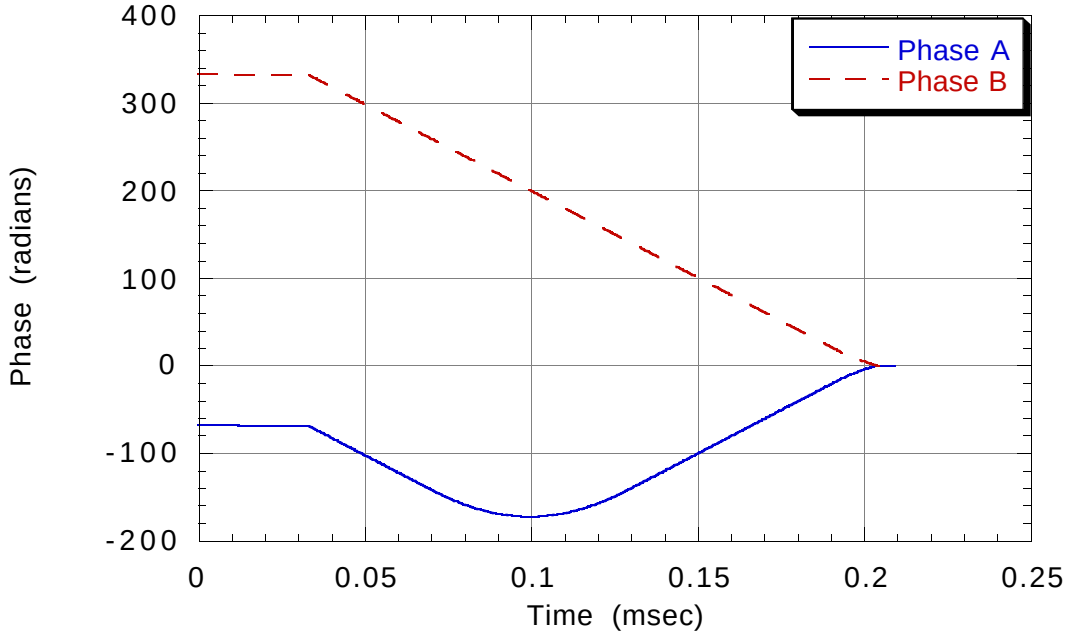

Figure 2. Nominal Phase Curve

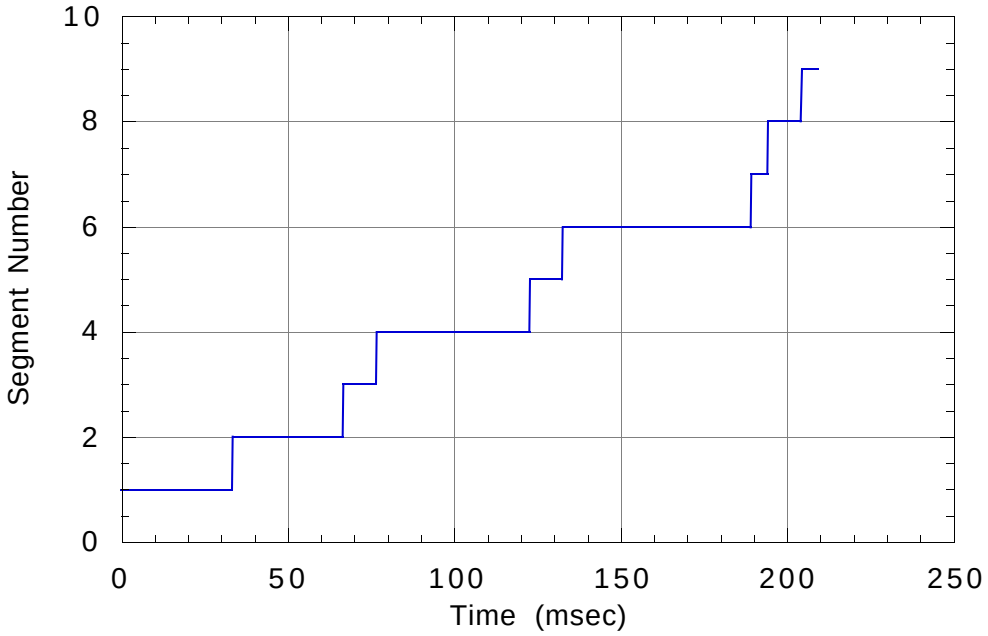

Figure 3. Segment number vrs time.

LIST OF INDEPENDENT PARAMETERS

Parameters	Nominal	Units

	Value	
Δt_1	33.3	msec
Δt_2	33.3	msec
Δt_3	10.0	msec
Δt_4	56.7	msec
Δt_7	5	msec
Δt_8	10	msec
Δt_9	1	msec
Δf_0	630	Hz
Δf_1	300	Hz
n_b	86	
γ	0	sec^{-2}

COMPUTED PARAMETERS

Parameters	Computed Value	Units
α	1.12×10^6	sec^{-2}
β	2.64×10^6	sec^{-2}
Δt_5	10	msec
Δt_6	56.57	msec

Appendix

The FORTRAN source code to generate slip stacking curves shown in this paper is shown below.

```
      program slip
      implicit real*8 (a-h)
      implicit real*8 (o-z)
      real*8 nbucket
c
      twopi = 4.*datan(1.d0)
      rfharm = 7.0*84.
c

      deltat_1 = 0.0333
      deltat_2 = 0.0333
      deltat_3 = 0.010
      deltat_4 = 0.046
      deltat_7 = 0.005
      deltat_8 = 0.010
      deltat_9 = 0.005
c
      deltaf_i = 10.0
      deltaf_0 = 630.
      deltaf_1 = 300.
      nbucket = 86.0
      gamma = 0.0
c
      deltat_5 = deltat_3
c
      alpha = deltaf_0/(deltat_3*(deltat_3+deltat_4))
      beta = (deltaf_0-deltaf_1+gamma*deltat_8**2)
     .            /(deltat_7**2+2.*deltat_8*deltat_7)
      deltat_6=(nbucket/2.0 - gamma*deltat_8**3/3.0
     .      + beta*deltat_7*(deltat_8**2+deltat_7*deltat_8+deltat_7**2/3.0))
      deltat_6 = deltat_6/deltaf_0
      deltat_6 = deltat_6 - deltat_7 - deltat_8
c
      t1 = deltat_1
      t2 = deltat_2 + t1
      t3 = deltat_3 + t2
      t4 = deltat_4 + t3
      t5 = deltat_5 + t4
      t6 = deltat_6 + t5
      t7 = deltat_7 + t6
      t8 = deltat_8 + t7
      t9 = deltat_9 + t8

      deltaphi_A_1 = twopi*(-deltaf_i*deltat_1)
      deltaphi_A_2 = twopi*(-deltaf_0*deltat_2)
```

```
      deltaphi_A_3 = twopi*(-deltaf_0*deltat_3+alpha*deltat_3**3/3.0)
      deltaphi_A_4 = twopi*((-deltaf_0+alpha*deltat_3**2)*deltat_4
     .            +  alpha*deltat_3*deltat_4**2)
      deltaphi_A_5 = twopi*(deltaf_0*deltat_3-alpha*deltat_3**3/3.0)
      deltaphi_A_6 = twopi*(deltaf_0*deltat_6)
      deltaphi_A_7 = twopi*(deltaf_0*deltat_7-beta*deltat_7**3/3.0)
      deltaphi_A_8 = twopi*((deltaf_0-beta*deltat_7**2)*deltat_8
     .            - beta*deltat_7*deltat_8**2 + gamma*deltat_8**3/3.0)
      deltaphi_B_1 = deltaphi_A_1
      deltaphi_B_2 = deltaphi_A_2
      deltaphi_B_3 = twopi*(-deltaf_0*deltat_3)
      deltaphi_B_4 = twopi*(-deltaf_0*deltat_4)
      deltaphi_B_5 = twopi*(-deltaf_0*deltat_5)
      deltaphi_B_6 = -deltaphi_A_6
      deltaphi_B_7 = -deltaphi_A_7
      deltaphi_B_8 = -deltaphi_A_8

      phi_A_8 = -deltaphi_A_8
      phi_A_7 = -deltaphi_A_7 + phi_A_8
      phi_A_6 = -deltaphi_A_6 + phi_A_7
      phi_A_5 = -deltaphi_A_5 + phi_A_6
      phi_A_4 = -deltaphi_A_4 + phi_A_5
      phi_A_3 = -deltaphi_A_3 + phi_A_4
      phi_A_2 = -deltaphi_A_2 + phi_A_3
      phi_A_1 = -deltaphi_A_1 + phi_A_2

      phi_B_8 = -deltaphi_B_8
      phi_B_7 = -deltaphi_B_7 + phi_B_8
      phi_B_6 = -deltaphi_B_6 + phi_B_7
      phi_B_5 = -deltaphi_B_5 + phi_B_6
      phi_B_4 = -deltaphi_B_4 + phi_B_5
      phi_B_3 = -deltaphi_B_3 + phi_B_4
      phi_B_2 = -deltaphi_B_2 + phi_B_3
      phi_B_1 = -deltaphi_B_1 + phi_B_2

      print 1, t1, t2, t3, t4, t5, t6, t7, t8, t9
    1 format (' ti-s',5f10.6)
      print 2, alpha, beta, deltat_6
    2 format (' alpha=',e15.6,' beta=',e15.6,' dt6=',f10.6)
      npt = 1000
      dt = t9/npt
      do i=0,npt
      t = i*dt
      if (t.lt.0.0) then
         nstep = 0
         phi_A = 0.0
         f_A=0.0
         df_A_dt = 0.0
         phi_B = 0.0
         f_B=0.0
         df_B_dt = 0.0
      elseif (t.le.t1) then
         nstep = 1
         phi_A = phi_A_1 + twopi*(-deltaf_i*t)
```

```
      f_A = -deltaf_i
      df_A_dt = 0.0
      phi_B = phi_B_1 + twopi*(-deltaf_i*t)
      f_B = -deltaf_i
      df_B_dt = 0.0
   elseif (t.le.t2) then
      nstep = 2
      phi_A = phi_A_2 + twopi*(-deltaf_0*(t-t1))
      f_A = -deltaf_0
      df_A_dt = 0.0
      phi_B = phi_B_2 + twopi*(-deltaf_0*(t-t1))
      f_B = -deltaf_0
      df_B_dt = 0.0
   elseif (t.le.t3) then
      nstep = 3
      phi_A = phi_A_3 +twopi*(- deltaf_0*(t-t2) + alpha*(t-t2)**3/3.0)
      f_A = -deltaf_0 + alpha*(t-t2)**2
      df_A_dt = 0.0
      phi_B = phi_B_3 + twopi*(-deltaf_0*(t-t2))
      f_B= -deltaf_0
      df_B_dt = 0.0
   elseif (t.le.t4) then
      nstep = 4
      phi_A = phi_A_4 + twopi*((-deltaf_0+alpha*deltat_3**2)*(t-t3)
     .         + alpha*deltat_3*(t-t3)**2)
      f_A = -deltaf_0 + alpha*deltat_3**2 + 2.0*alpha*deltat_3*(t-t3)
      df_A_dt = 0.0
      phi_B = phi_B_4 + twopi*(-deltaf_0*(t-t3))
      f_B = -deltaf_0
      df_B_dt = 0.0
   elseif (t.le.t5) then
      nstep = 5
      phi_A = phi_A_5 + twopi*((deltaf_0-alpha*deltat_3**2)*(t-t4)
     .         + alpha*deltat_3*(t-t4)**2 - alpha*(t-t4)**3/3.0)
      f_A = deltaf_0 - alpha*deltat_3**2 + 2.0*alpha*deltat_3*(t-t4)
     .         - alpha*(t-t4)**2
      df_A_dt = 2.*alpha*deltat_3-2.*alpha*(t-t4)
      phi_B = phi_B_5 + twopi*(-deltaf_0*(t-t4))
      f_B = -deltaf_0
      df_B_dt = 0.0
   elseif (t.le.t6) then
      nstep = 6
      phi_A = phi_A_6 + twopi*(deltaf_0*(t-t5))
      f_A = deltaf_0
      df_A_dt = 0.0
      phi_B = -phi_A
      f_B = -f_A
      df_B_dt = -df_A_dt
   elseif (t.le.t7) then
      nstep = 7
      phi_A = phi_A_7 + twopi*(deltaf_0*(t-t6)-beta*(t-t6)**3/3.0)
      f_A = deltaf_0 - beta*(t-t6)**2
      df_A_dt = -2.0*beta*(t-t6)
      phi_B = -phi_A
```

```
         f_B = -f_A
         df_B_dt = -df_A_dt
      elseif (t.le.t8) then
         nstep = 8
         phi_A = phi_A_8 + twopi*((deltaf_0-beta*deltat_7**2)*(t-t7)
     .         - beta*deltat_7*(t-t7)**2+gamma*(t-t7)**3/3.0)
         f_A = deltaf_0 - beta*deltat_7**2 - 2.*beta*deltat_7*(t-t7)
     .         + gamma*(t-t7)**2
         df_A_dt = -2.*beta*deltat_7 + 2.0*gamma*(t-t7)
         phi_B = -phi_A
         f_B = -f_A
         df_B_dt = -df_A_dt
      elseif (t.le.t9) then
         nstep = 9
         phi_A = 0.0
         f_A=0.0
         df_A_dt = 0.0
         phi_B = 0.0
         f_B=0.0
         df_B_dt = 0.0
      else
         nstep = 10
         phi_A = 0.0
         f_A=0.0
         df_A_dt = 0.0
         phi_B = 0.0
         f_B=0.0
         df_B_dt = 0.0
      endif
      write(1,3) nstep, t, phi_A, f_A, df_A_dt
      write(2,3) nstep, t, phi_B, f_B, df_B_dt
    3 format (1x,i5,f10.6,f10.2,f10.2,e15.6)
c     print 5, t, phi_A, f_A, df_A_dt, phi_B, f_B, df_B_dt
    5 format (1x,f10.6,3e15.6/11x,3e15.6)
      enddo
      stop
      end
```