Optimum Matched Filter (Transfer Function)

(Nested Processes ... Each Process controlled by a Solver)

Problem Description

The transfer function H(s) is the Laplace transform of the output signal $Y_{out}(s)^*$ divided by the Laplace transform of the input signal $Y_{in}(s)^*$: that is $H(s)=\frac{Y_{out}(s)}{Y_{in}(s)}$ where each signal's transform is assumed to be a ratio of polynomials. Thus, H(s) can likewise be stated in the form:

$$H(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots + a_m s^m}{b_0 + b_1 s + b_2 s^2 + \ldots + b_n s^n}$$

Equation 0.1 Generalized H(s)

Assuming the numerator and denominator can be factored, yields H(s) in the general form

$$H(s) = \frac{a_m (s - Z_1)(s - Z_2)\ldots(s - Z_m)}{b_n (s - P_1)(s - P_2)\ldots(s - P_n)}$$

Equation 0.2 Factored Transfer Function

where each Z_i is known as a "zero" and the P_i as a "pole" of the transfer function. Z_i and P_i are complex points in the Laplace domain.

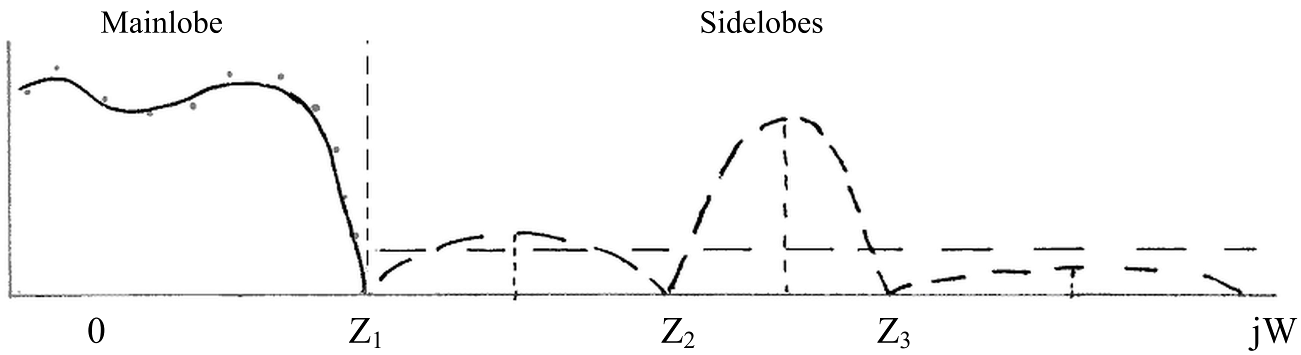
A realizable transfer function must have poles and zeros with their conjugate points. That is, poles and zeros come in pairs. If a pole or zero is located at the complex point $\sigma_i + j\omega_i$, then its conjugate is located at $\sigma_i - j\omega_i$. Thus, a generalized transfer function is stated as

$$H(s) = gain \frac{(s - Z_\sigma)(s + Z_\sigma) \prod_{p=1}^{Z\_Pairs} \{(s - Z_p)(s - Z_p^*)\} \prod_{q=1}^{Z\_Quads} \{(s - Z_q)(s - Z_q^*)(s + Z_q)(s + Z_q^*)\}}{(s - P_\sigma) \prod_{i=1}^{P\_Pairs} \{(s - \sigma_i)^2 + \omega_i^2\}}$$

Equation 0.3 Generalized Transfer Function H(s)

Given n-data points from a Bode plot (see Figure 0.1 below) that define the mainlobe of the desired transfer function, find the optimal Pole/Zero constellation such that H(s) has equal sidelobe peak amplitudes in a Bode plot and curve fits the given data in the mainlobe.

Bode Plot: Mainlobe with 3 Sidelobes



Frequency

Figure 0.1 H(s) Mainlobe & Sidelobe Plot

To help view what's going on, think of the LaPlace domain covered with a rubber matt and pinned on its corners. From the under side, in the 'mainlobe' area, place one's poles (i.e. push up the rubber matt at these locations). This should give the impression of a hill or mountain. Place your zeros on the frequency axis by pressing down from the top at your zero locations (i.e. z_1, z_2, &z_3 points as shown above on the bode plot). Now if you cross cut the rubber matt on the frequency axis and you should have the bode plot above.

The objective is to keep the mainlobe 'mountain' while moving your zeros in order to get equal peak heights in your sidelobes. Sounds simple but a slight movement in those zeros changes the peak heights radically. See Figure 0.2b and note how far down these peaks are in amplitude: farther down, less noise in system.

Computer Plots

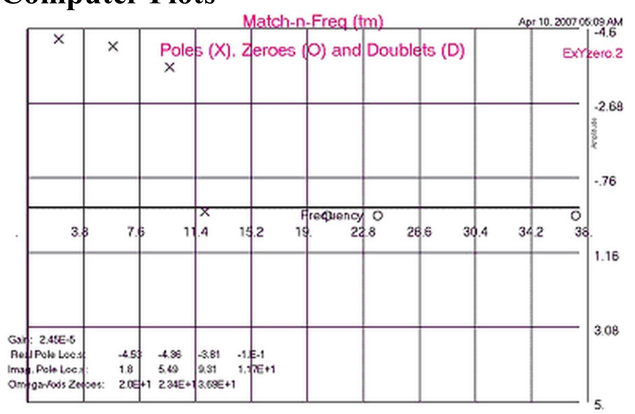


Figure 0.2a H(s) Pole/Zero Locations

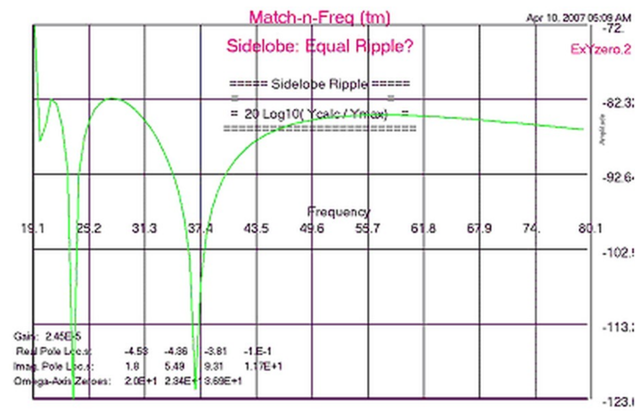


Figure 0.2b Equal Peaks in Sidelobes

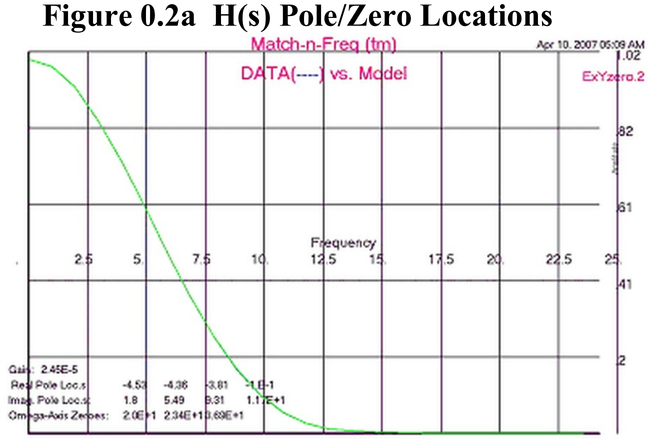


Figure 0.2c Data vs. H(s) Model Curves

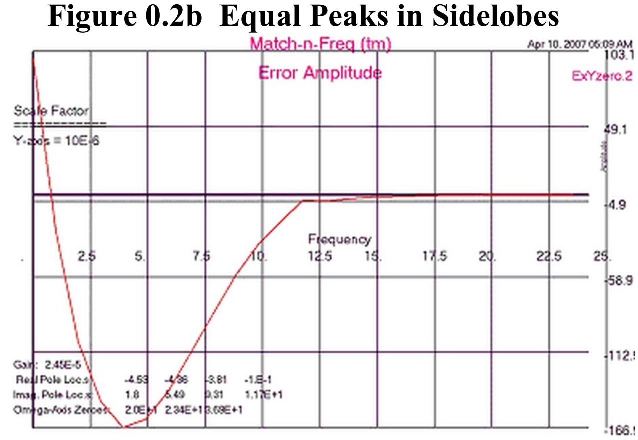


Figure 0.2d H(s) Fit Error Plot

Computer Code

The first FIND statement is used to find good values for mainlobe parameters **gain, p.real, p.imag**

FIND gain, p.real, p.imag; IN Laplace.Domain; BY AJAX; TO MATCH error

With good mainlobe parameters, then this above find statement executes two nested Find statements to find the sidelobe parameters.

FIND x.zeros **IN** .Stopband **BY HERA** **WITH BOUNDS** side.limits **TO MINIMIZE** peak.diff

```
  PROBLEM FILTER(40000, 5000, 5000)  ! Match-n-Freq (tm)
C ---------------------------------------------
C --- FORTRANCALCULUS Application: Find Pole/Zero Constellation of a ---
C    Matched Filter Transfer Function  ---
```

```
C ---------------------------------------------
  call input
C ---- Find Pole-Zero Locations ----
  nYzeros=0:  call fit   ! Don't vary yZeros yet.
  Yzeros8n= 1.6 * ylmax
  nYzeros= 3:  call fit   ! Now vary yZeros
  call output
end
model fit  ! Minimize Magnitude fit Error
C - Varying Gain & Pole/Zero locations -
  n= 1 + (nXpole+2*nPpairs) + nXzero +
2*(nZpairs+nZquads)
  allot h8low( n):   <h8low>= xlmin:  h8low(1)= 0
  allot h8hi( n):   <h8hi>= ylmax:  h8hi(1)= -1

  FIND gain,Xzero, Preal, Pimag;  &
   in Transfer;  by JOVE(contrl1);  &
   with lower h8low;  and uppers h8hi;  &
   MATCHING error; TO MINIMIZE errsum

  endif
end
model Transfer
  errsum= 0.D0
  do 50 ii= 1, npoints  !  --- CALCULATE TRANSFER
FUNCTION ----
    Y2=freq(ii)**2:  hw=xfunct(Y2)
    error( ii)= gain * hw * y8in( ii) - y8out( ii)   !
Absolute Error
    error(ii)= error(ii)/(y8out( ii)**ierrtyp)   ! Relative
    errsum= errsum + error( ii)**2
50  continue
  if( nYzeros .gt.2) call sideArea
end
Fmodel xfunct( Y2)
  real*8 num
  num= 1:    den= 1
  do 20 ij= 1, nPpairs
    den= den*factor(Y2,-Preal(ij), Pimag( ij))
20  continue
  if( nYzeros .gt. 0) then
    num= 100 * num
    do 40 ij= 1, nYzeros
      num= num * factor( Y2, 0., Yzeros(ij))
40  continue
  endif
  q= num / den
  if( q .gt. 1.D20) q= 1.D20
  xfunct= q
end
Fmodel factor( y2, sigma, omega)
  r2= sigma**2
  if( omega .eq. 0.D0) then
    factor= 1:         if( sigma .eq. 0.D0) return  ! not sure
on value
    factor= (y2 + r2) / r2  ! R2 normalizing factor
    return
  endif
  o2= omega**2:  sum=(r2+o2+y2)/10
  temp= sum*sum - 4*y2*o2/100:  factor= 0
  if( temp .eq. 0.D0) return
  temp= sqrt( temp)
  factor= temp / (r2 + o2)  ! this R2+O2 is 4
normalizing pole values
end  ! and adjusts Gain value for system.

model sideArea
  n1= nYzeros-1
  do 20 ij= 1, n1
    if(Yzeros(ij).ge.Yzeros(ij+1)) then
      tmp=Yzeros(ij):  Yzeros(ij)= Yzeros( ij+1)
      Yzeros( ij+1)= tmp
    endif
20 continue
  do 30 ij= 2, n1
    sidelims(ij-1)=Yzeros(ij+1)- Yzeros( ij)
    peakloc(ij)=(Yzeros( ij+1) + Yzeros( ij))/2  *.95
30  continue
  peakloc( 1)= (Yzeros( 2) + Yzeros( 1))/2  *.95
  peakloc( nYzeros)= (ylmax + Yzeros( nYzeros))/ 1.5
  sidelims( n1)= ABS( ylmax - Yzeros( nYzeros))
  do 40 ij= 1, n1
    sidelims( ij)= sidelims( ij)*ij/(nYzeros * 5)
    Yzeros2( ij)= Yzeros( ij+1)
40 continue

  FIND Yzeros2;  in stopband;  by Hera( contrl2); &
     with BOUNDS sidelims; & ! BANDLIM;
     TO MINIMIZE diff

  do 50 ij= 1, n1
    Yzeros( ij+1)= Yzeros2( ij)
    if( Yzeros(ij) .ge. Yzeros( ij+1)) then
      tmp= Yzeros( ij):  Yzeros( ij)= Yzeros( ij+1)
      Yzeros( ij+1)= tmp
    endif
50  continue
end
model stopband
  do 50 jj= 2, nYzeros
    Yzeros( jj)= Yzeros2( jj-1)
50 continue
  diff= 0:   sidelim= .02
  do 60 jj= 1, nYzeros
    ipeak= jj
    if( jj .gt. 1) then
      sidelim= sidelims( jj-1)
    endif

   FIND peakloc( jj);  in sidelobe;  by hera( contrl3); &
     with BOUNDS sidelim; &
     TO MAXIMIZE peakampl( jj)

    diff= diff + slope( jj)**2
    if( jj .gt. 1) then
      anorm= peakampl( jj)**2 + peakampl( jj-1)**2
```

```
    diff= diff + (peakampl( jj) - peakampl( jj-1))**2 /
anorm
   endif
60 continue
 anorm= peakampl( 1)**2 + peakampl( nYzeros)**2
 diff=diff+(peakampl(1)-peakampl( nYzeros))**2 /
anorm
 diff= diff * 1.d6
 peakave= 0:   errsumpk= 0
 do 70 jj= 1, nYzeros
   peakave=peakave+dabs(peakampl( jj))
70 continue
 do 80 jj= 1, nYzeros
   peakerr(jj)=peakave - peakampl( jj)
   errsumpk=errsumpk+(peakerr(jj) / anorm)**2
80 continue
end
model sidelobe
 peakampl(ipeak)=sideAmpl( peakloc( ipeak))
 ampl1=sideAmpl(.9999*peakloc(ipeak))
 ampl2=sideAmpl(1.0001*peakloc( ipeak))
 slope(ipeak)=1.D6*(ampl1-ampl2)/ .0002   ! Slope
approx.
end
Fmodel sideAmpl( Y)
 Y2= Y * Y:  sideAmpl= xfunct( Y2)
end
controller contrl1( JOVE)
 remax=maxit(1):  detail=ireport(1):  zero=ch8tol(1)
 stepslim=limsteps:  stepout=stepout2:
evalmax=maxeval
 accuracy=accurcy
end
controller contrl2( Hera)
 remax=maxit(2):  detail=ireport(2):
progress=ch8tol(2)
 adjust= 1:  summary= 1
end
```

Computer Output for JOVE & HERA Solvers:

--- **JOVE** SUMMARY, INVOKED AT FIT[63] FOR MODEL FITBOTH ----

CONVERGENCE CONDITION AFTER 5 ITERATIONS
MODEL EVALUATION LIMIT EXCEEDED
OBJECTIVE CRITERION UNSATISFIED
MAXIMUM ITERATIONS PERFORMED
SPECIFIED CRITERIA UNSATISFIED

LOOP NUMBER ...	[INITIAL]	1	2
UNKNOWNS			
GAIN	1.230000E-02	2.426167E-05	2.452567E-05
PREAL(1)	2.000000E-01	7.549617E-01	7.562047E-01
PREAL(2)	2.000000E-01	7.577069E-01	7.562061E-01
PREAL(3)	2.000000E-01	7.559795E-01	7.562061E-01
PREAL(4)	2.000000E-01	5.001370E-03	5.017036E-03
PIMAG(1)	1.000000E-02	5.002709E-03	5.033684E-03
PIMAG(2)	1.000000E-01	5.000923E-03	5.011472E-03
PIMAG(3)	2.000000E-01	5.000739E-03	5.009188E-03
PIMAG(4)	3.000000E-01	3.000000E+00	2.999494E+00
OBJECTIVE			
ERRSUM	2.246747E+04	1.386396E+01	1.386382E+01
EQUALITY CONSTRAINTS			
ERROR(1)	4.664578E-03	1.812689E-06	1.912965E-06
ERROR(2)	3.036344E-03	-5.624263E-07	-4.925001E-07
ooo			
ERROR(25)	-1.157844E-18	-1.084009E-18	-1.083120E-18

ooo

LOOP NUMBER ...	[INITIAL]	5
UNKNOWNS		
GAIN	1.230000E-02	2.451293E-05
PREAL(1)	2.000000E-01	3.134831E-01
PREAL(2)	2.000000E-01	6.712762E-01
PREAL(3)	2.000000E-01	4.565040E-01
PREAL(4)	2.000000E-01	5.250445E-02

```
  PIMAG( 1)          1.000000E-02      3.674824E-01
  PIMAG( 2)          1.000000E-01     4.785724E-02
  PIMAG( 3)          2.000000E-01     5.398629E-03
  PIMAG( 4)          3.000000E-01     7.671278E-01
 OBJECTIVE
  ERRSUM            2.246747E+04     7.207860E-01
 EQUALITY CONSTRAINTS
  ERROR( 1)          4.664578E-03     1.908127E-06
  ERROR( 2)          3.036344E-03    -4.879579E-07
  ooo
  ERROR(25)         -1.157844E-18    -1.154290E-18

---END OF LOOP SUMMARY

  ooo

---- HERA SUMMARY, INVOKED AT SIDEAREA[180] FOR MODEL STOPBAND ----

  CONVERGENCE CONDITION AFTER  2 ITERATIONS
    UNKNOWNS CONVERGED
    OBJECTIVE CRITERION UNSATISFIED
    ALL SPECIFIED CRITERIA SATISFIED

 LOOP NUMBER ...      [INITIAL]          1                 2
 UNKNOWNS
  YZEROS2(1)        1.500000E+00     1.525001E+00     1.525000E+00
  YZEROS2(2)        2.200000E+00     2.060394E+00     2.060395E+00
 OBJECTIVE
  DIFF              5.476185E+18     8.586592E+06     1.890314E+06

---END OF LOOP SUMMARY

---- HERA SUMMARY, INVOKED AT SIDEAREA[180] FOR MODEL STOPBAND ----

  CONVERGENCE CONDITION AFTER  3 ITERATIONS
    UNKNOWNS CONVERGED
    OBJECTIVE CRITERION UNSATISFIED
    ALL SPECIFIED CRITERIA SATISFIED

 LOOP NUMBER ...      [INITIAL]          1                 2
 UNKNOWNS
  YZEROS2(1)        1.525000E+00     1.483725E+00     1.483717E+00
  YZEROS2(2)        2.060395E+00     1.958406E+00     1.958428E+00
 OBJECTIVE
  DIFF              2.453228E+16     2.538071E+06     1.887044E+06

 LOOP NUMBER ...      [INITIAL]          3
 UNKNOWNS
  YZEROS2(1)        1.525000E+00     1.483717E+00
  YZEROS2(2)        2.060395E+00     1.958429E+00
 OBJECTIVE
  DIFF              2.453228E+16     1.885848E+06

---END OF LOOP SUMMARY

  O o o

--- JOVE SUMMARY, INVOKED AT FIT[63] FOR MODEL FITBOTH ----
```

```
  CONVERGENCE CONDITION AFTER  5 ITERATIONS
    OBJECTIVE CRITERION UNSATISFIED
    MAXIMUM ITERATIONS PERFORMED
    SPECIFIED CRITERIA UNSATISFIED

 LOOP NUMBER ...      [INITIAL]          1               2
 UNKNOWNS
   GAIN              2.451838E-05    2.451843E-05    2.451824E-05
   PREAL( 1)         1.903761E-01    1.900577E-01    1.898417E-01
   PREAL( 2)         2.174091E-01    2.170316E-01    2.167918E-01
   PREAL( 3)         2.261904E-01    2.258044E-01    2.255600E-01
   PREAL( 4)         6.035538E-03    6.026419E-03    5.945594E-03
   PIMAG( 1)         4.643045E-01    4.642716E-01    4.643402E-01
   PIMAG( 2)         2.736201E-01    2.736099E-01    2.736497E-01
   PIMAG( 3)         8.978034E-02    8.977338E-02    8.978452E-02
   PIMAG( 4)         5.907978E-01    5.906571E-01    5.902591E-01
 OBJECTIVE
   ERRSUM            1.604998E-01    1.604787E-01    1.604562E-01
 EQUALITY CONSTRAINTS
   ERROR( 1)         1.910196E-06    1.910216E-06    1.910143E-06
   ERROR( 2)        -4.901276E-07   -4.901482E-07   -4.901328E-07
   ooo
   ERROR(25)        -1.165042E-18   -1.165045E-18   -1.165047E-18

   ooo

 LOOP NUMBER ...      [INITIAL]          5
 UNKNOWNS
   GAIN              2.451838E-05    2.451764E-05
   PREAL(1)          1.903761E-01    1.904776E-01
   PREAL(2)          2.174091E-01    2.177901E-01
   PREAL(3)          2.261904E-01    2.266766E-01
   PREAL(4)          6.035538E-03    5.000000E-03
   PIMAG(1)          4.643045E-01    4.656372E-01
   PIMAG(2)          2.736201E-01    2.742334E-01
   PIMAG(3)          8.978034E-02    8.995914E-02
   PIMAG(4)          5.907978E-01    5.862881E-01
 OBJECTIVE
   ERRSUM            1.604998E-01    1.596411E-01
 EQUALITY CONSTRAINTS
   ERROR(1)          1.910196E-06    1.909913E-06
   ERROR(2)         -4.901276E-07   -4.890354E-07
   ooo
   ERROR(25)        -1.165042E-18 -1.165045E-18

---END OF LOOP SUMMARY
```

----------------------------- ExYzero.1

Resulting Parameters in De-normalized form:

Pole-pairs

	PREAL	PIMAG
1.	-3.809552E+00	9.312744E+00
2.	-4.355801E+00	5.484667E+00
3.	-4.533533E+00	1.799183E+00

4. -1.000000E-01 1.172576E+01

Zeros

Zeros on Omega-axis: 0 +- 20.00 29.67 39.17

ELAPSED TIME= 5.99 SECONDS

Computer App for Solving these Problems

The Match-n-Freq App is required to solve this type of problem. Please download the free Match-n-Freq App, install it, and execute some Demo files in order to get an idea of the process that one must do to solve your problem

Findings

The resulting plots told the story according to the theory & practical application of the time. The results suggested a good to excellent solution but we wanted better results. Converting the transfer function, H(s), to the time domain, h(t), is were this project ended. h(t) was achieved but had some problem finding the y.out(t) output signal peaks. The following equations will define this problem in the time domain.

Future

Next, find a good math model for $y_{out}(t)$ using a series of Lorentzian curves as done in **Error! Reference source not found.**.1. Calculate a Pattern-Induces-Bit (PIB) shift given a bit stream of zeroes and ones with some time spacing. For example, a pattern of 3T-8T-3T-8T-3T etc. where T = 20.8 (ns) with a series of bits. The bits alternate their polarity for each '1' bit. This will generate a sinusoidal wave. Once satisfied that the PIB shifts are accurate, put this PIB shift program around the matched filter program above. The new program should have a FIND statement to find the number of poles (i.e. 'nXpole' & 'nPpairs') necessary to minimize the PIB shift. (After working some on the time domain approach, it seems that the zeroes do not need to be requested. If they are necessary, ones input & output function, $y_{in}(t)$ & $y_{out}(t)$, will bring them into play. See **Error! Reference source not found.** or **Error! Reference source not found.** for more on this.)

Minimizing the PIB shift should be the overall design objective for a read-write channel for a disc drive. If the PIB shift is to high, the data or bit pattern written will not be able to be retrieved.

A project objective is very important to say the least. Get your team to agree on one and keep it short; just a few words e.g. minimize this or maximize that. Here is where Programming Calculus really shines. If an objective changes over time, just change it in your model and rerun the problem. Without Programming Calculus one may have been playing with a numerical method. A change in objective could force one to practical start over; loosing months of time.

***Note** Error in Yin & Yout calculations when digital:

The best way to determine Yin & Yout functions is to find their desired functions in the time-domain (if digital data, then approximate function with a good Curve fit routine) and then calculate their (analog) Fourier transforms.

Yin is an 'isolated readback pulse' created with some type of disc drive head. Capture this signal digitally and then curve fit it using CurvFit with a series of Lorentzian pulses.

Yout is a desired signal with some desired features; e.g. 'thin' pulse, no pre- nor post-undershoots, etc. A (Modified) Lorentzian pulse is what we choose.

Your Turn!

Have a Filter design to solve? Please state it here and use graphs, pictures, etc. to get your problem well stated and understood by those reading it.