DADF

User Station Design Justification - MathCAD Files

Document Signature Table

	Name	Function	Signature	Date
Author	Dr. Paul Crawford			22-06-98
Approval				
Approval				
Release				
Eumetsat Approval				

Document Change Record

Issue/Revision	Date	DCN No.	Changed Pages/Paragraphs
3.0	22.06.98		Initial Release with other Revision 3.0 doccuments.
4.0	03.08.98		Updated following DD review meeting.marked by changebars..

Table of Contents

Table of Figures

None

Table of Tables

1 Simulation Files

This annex lists the MathCAD files used for the two technical notes on the carrier recovery PLL design, and the matched filter and bit synchronisation PLL design. These are summarised in Table 1 below:

File Name	
`biterr33.mcd`	Effect of timing errors on bit sync losses.
`bpsko66.mcd`	BPSK PLL optimisation and loss calculations for the LRIT system.
`qpsko66.mcd`	QPSK PLL optimisation and loss calculations for the HRIT system.
`phase1.mcd`	Sensitivity of PLL systems to phase noise.
`bpskdet.mcd`	BPSK phase detector (phase error estimator) performance. This writes the file `bpsk_psk_loss.prn` with the "squaring loss" for various non-linearities.
`qpskdet.mcd`	QPSK phase detector (phase error estimator) performance. This writes the file `qpsk_psk_loss.prn` with the "4th power loss" for various non-linearities.
`rcos88lrit.mcd`	FIR design file (and technology loss computation) for the LRIT system. This MathCAD file writes the data file `lrit_fir.prn` with the 22-bit coefficients for the HSP50214 programmable FIR filter.
`rcos88hrit.mcd`	FIR design file (and technology loss computation) for the HRIT system. This MathCAD file writes the data file `hrit_fir.prn` with the 22-bit coefficients for the HSP50214 programmable FIR filter.
`Viterbi2.mcd`	Visualisation of the Viterbi decoder simulation results. This computes the CCSDS error rate curve as a function of Eb/No and saves this in the file `ccsds_lut.prn`
`qpsk_edata.mcd`	Computation of carrier tracking loop for LRIT using BPSK, this with revised carrier PDF and more accurate CCSDS error rate formulae. This noe to plot as function of Es/No
`zcd_sim_bpsk.mcd`	Self-noise of ZCD timing error algorithm and *a priori* S-curve.
`zcd_sim_qpsk.mcd`	Self-noise of ZCD timing error algorithm and *a priori* S-curve.
bpsk_edata.mcd	Computation of carrier tracking loop for LRIT using BPSK, this with revised carrier PDF and more accurate CCSDS error rate formulae. This noe to plot as function of Es/No

Table 1: MathCAD files

2 HRIT Filter Coefficients

The following table lists the 22-bit values computed by `rcos7hrit.mcd` for use in the FIR filter section of the HSP50214 programmable downconverter. Copied from the file `hrit_fir.prn`

Tap Number	Value
1	-177021
2	-242790
3	-156607
4	142237
5	640167
6	1237904
7	1777163
8	2097151
9	2097151
10	1777163
11	1237904
12	640167
13	142237
14	-156607
15	-242790
16	-177021

Table 2: HRIT FIR Coefficients

NOTE: The filter is symmetric, i.e. coefficients for taps 9-16 are the same as for 8-1

3 LRIT Filter Coefficients

The following table lists the 22-bit values computed by `rcos7lrit.mcd` for use in the FIR filter section of the HSP50214 programmable downconverter. Copied from the file `lrit_fir.prn`

Tap Number	Value
1	-16479
2	-12713
3	-1131
4	13217
5	22833
6	21385
7	7383
8	-14026
9	-32421
10	-36571
11	-20341
12	12679
13	48447
14	66978
15	51188
16	-2976
17	-79402
18	-141471
19	-140183
20	-29214
21	217605
22	592507
23	1049241
24	1510521
25	1886215
26	2097151
27-52	Same as 26-1 (symmetric filter)

Table 3: LRIT FIR Coefficients

File "BITERR3.MCD"
Created using MathCAD 6.0-PLUS
Compute the sensitivity of the MSG system to errors in bit timing. This version with 'per symbol' calculation and Gaussian errors plotted w.r.t Loop SNR

$$dB(x) := if\left(|x| < 10^{-12}, -120, 10\cdot\log(|x|)\right)$$

Convert to dB
And back to numeric

$$idB(x) := 10^{\frac{x}{10}}$$

$$sinc(x) := if\left(|x| < 10^{-8}, 1.0, \frac{\sin(x)}{x}\right)$$

Safe sinc() function.

$$T := 1$$

Bit period (normalised).
Define the power spectrum of the raised cosine signal:

$$S(f,a) := if\left[|f| < \frac{1-a}{2\cdot T}, 1, if\left[|f| > \frac{1+a}{2\cdot T}, 0, \frac{1}{2}\cdot\left(1+\cos\left(\pi\cdot\frac{2\cdot f\cdot T - 1 + a}{2\cdot a}\right)\right)\right]\right]$$

Time domain version of S(f) (with check and fix for 0/0 conditions)

$$s(t,a) := if\left[\left||t| - \frac{T}{2\cdot a}\right| < 10^{-6}, \frac{\pi}{4}\cdot\frac{sinc(\pi\cdot t)}{T}, \frac{\cos\left(\pi\cdot a\cdot\frac{t}{T}\right)}{1-\left(2\cdot a\cdot\frac{t}{T}\right)^2}\cdot\frac{sinc(\pi\cdot t)}{T}\right]$$

Compute the time waveform for 10 bit (+/- 5) periods:

$$ii := 0..50$$

$$tm_{ii} := \frac{ii}{10.0}$$

$$aL := 1.0$$

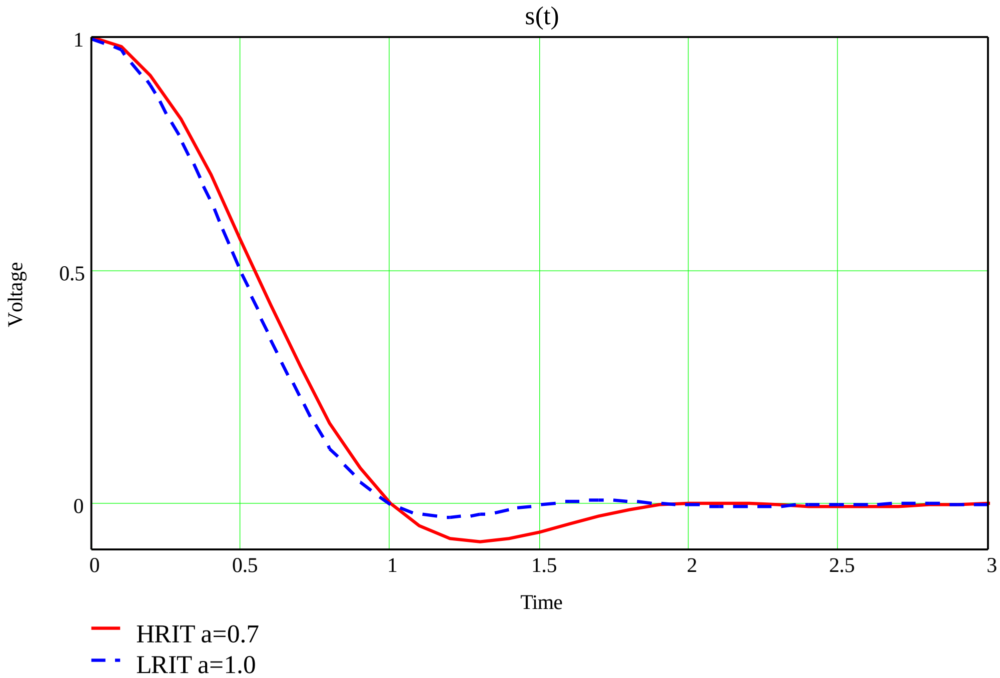
Alpha for raised cosine filter LRIT

$$stL_{ii} := s\left(tm_{ii}, aL\right)$$

Alpha for raised cosine filter HRIT

$$aH := 0.7$$

$$stH_{ii} := s\left(tm_{ii}, aH\right)$$



Compute the ISI term for adjacent bits. Use the root-sum-square formula to estimate the RMS voltage from the timing error:

$\mathrm{Kmax} := 6$

$$\mathrm{ISIfunc}(\delta, a) := \sqrt{\sum_{i=-\mathrm{Kmax}}^{\mathrm{Kmax}} \mathrm{if}\left(i = 0, 0, s(\delta + i \cdot T, a)^2\right)}$$

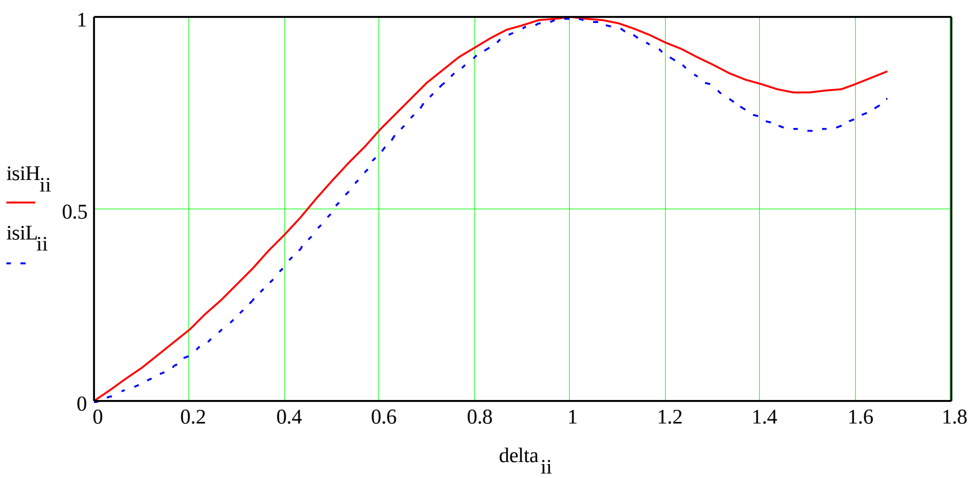
$ii := 0 .. 50$

$$\mathrm{delta}_{ii} := \frac{ii}{30.0}$$

Voltage term based on power sum

$\mathrm{isiH}_{ii} := \mathrm{ISIfunc}\left(\mathrm{delta}_{ii}, \mathrm{aH}\right)$

$\mathrm{isiL}_{ii} := \mathrm{ISIfunc}\left(\mathrm{delta}_{ii}, \mathrm{aL}\right)$



Function that uses the simulated (and saved above) Viterbi decoder results to model the CCSDS error rate versus Eb/No more accurately.

$ccsds := READPRN(ccsds\_lut)$

$Nccsds := rows(ccsds)$

$Nccsds = 32$

$icc := 0 .. Nccsds - 1$

$cc\_eb_{icc} := ccsds_{icc,0}$

$cc\_logPE_{icc} := ccsds_{icc,1}$

$cc\_vs := cspline(cc\_eb, cc\_logPE)$

$cc\_ifunc(v) := 10^{interp(cc\_vs, cc\_eb, cc\_logPE, v)}$

$vMax := ccsds_{Nccsds - 1, 0}$

$ErrMin := 10^{ccsds_{Nccsds - 1, 1}}$

$pef(v) := if(v<0, 0.5, if(v>vMax, ErrMin, cc\_ifunc(v)))$

And an inverse error rate formulae (requires care to prevent convergence failure):

$IpefLIM := vMax$

Maximum returned value.

$pefLIM := ErrMin$

Corresponding input value

$guess := idB(2.7)$

$pediff(erate, guess) := log(erate) - log(pef(guess))$

Use logs to remove slope

$ipe(erate, guess) := root(pediff(erate, guess), guess)$

$INVpef(r) := if(r>pefLIM, ipe(r, guess), IpefLIM)$

$p := pef(idB(2.7))$

$p = 2.472 \cdot 10^{-8}$

Quick test of inverse function.

$dB(INVpef(p)) = 2.700000$

Uncoded PSK (for comparason).

$peu(v) := if\left(v<0.0, 0.5, if\left(v>30, 4.718 \cdot 10^{-15}, cnorm\left(-\sqrt{2 \cdot v}\right)\right)\right)$

$peu(idB(12.15)) = 5.076 \cdot 10^{-9}$

$IpeuLIM := 25$

$peuLIM := peu(IpeuLIM)$

$peuLIM = 7.687 \cdot 10^{-13}$

$pediff(erate, guess) := log(erate) - log(peu(guess))$

$ipe(erate, guess) := root(pediff(erate, guess), guess)$

$guess := idB(8)$

$INVpeu(r) := if(r>peuLIM, ipe(r, guess), IpeuLIM)$

Run a check to see if it works OK.

$p := peu(idB(12))$

$p = 9.006 \cdot 10^{-9}$

$dB(INVpeu(p)) = 12.000000$

Decide on the error rate function.

$pe(x) := pef(x)$

$INVpe(x) := INVpef(x)$

Compute the effect of errors on the system. Kfec is fudge factor for the Forward Error Correction system's response to correlated interferance. Rc is the code rate, 1 for uncoded, 1/2 for R=1/2 Viterbi, etc.

$Kfec := 2$

$Rc := \frac{223}{512}$

$$peBIT(t, a, en) := pe\left(\frac{s(t,a)^2}{\frac{1}{en} + Kfec \cdot Rc \cdot ISIfunc(t,a)^2}\right)$$

$Imax := 20$

$ii := 0 .. Imax$

$$dt_{ii} := \frac{ii + 0.5}{5 \cdot (Imax + 0.5)}$$

Perform the calculations for the static timing error case (simply invert the Pe function):

$target\_perr := 5 \cdot 10^{-9}$

$en := INVpe(target\_perr)$

$dB(en) = 2.752$

$guess := en \cdot 1.01$

$chk := idB(0.6)$

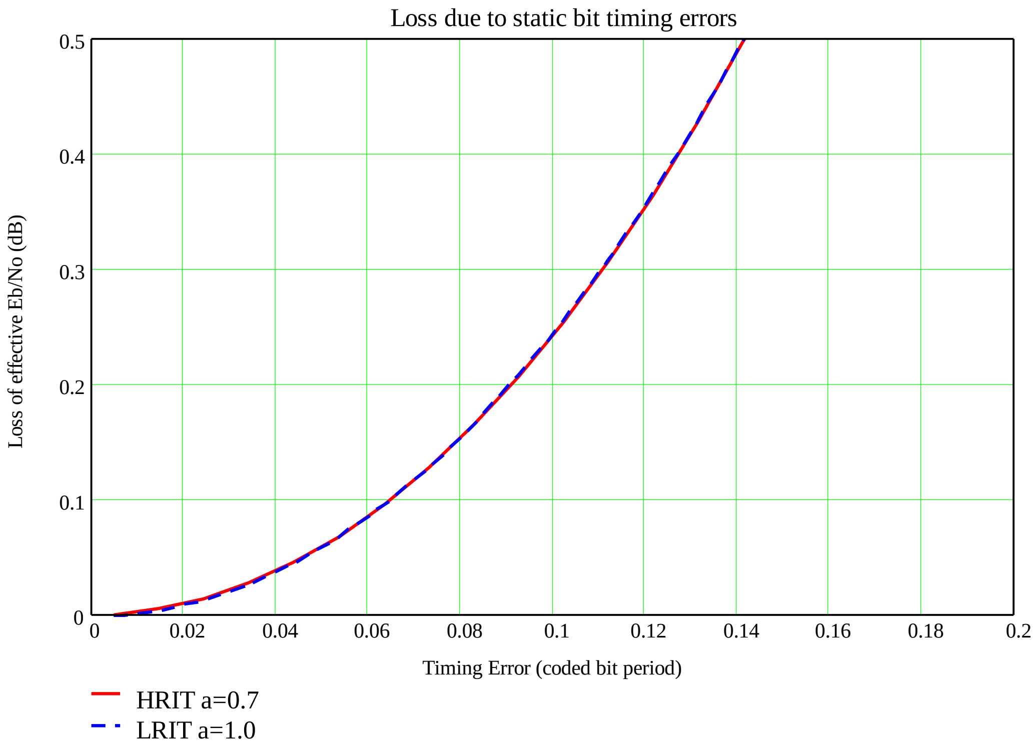
$derrF(t, a, en) := \log(peBIT(t, a, en)) - \log(target\_perr)$

$ifunc(t, a) := if(peBIT(t, a, en \cdot chk) > target\_perr, 100, root(derrF(t, a, guess), guess))$

$lossF(t, a) := dB(ifunc(t, a)) - dB(en)$

$LsH_{ii} := lossF(dt_{ii}, aH)$

$LsL_{ii} := lossF(dt_{ii}, aL)$



$$dt_{ii} := \frac{ii + 0.5}{3 \cdot (Imax + 0.5)}$$

Perform the integration for Uniform Errors.

$$UniformPerr(t, a, en) := \frac{2}{t} \cdot \int_0^{\frac{t}{2}} peBIT(\theta, a, en)\, d\theta$$

$guess := en \cdot 1.01$

$chk := idB(0.6)$

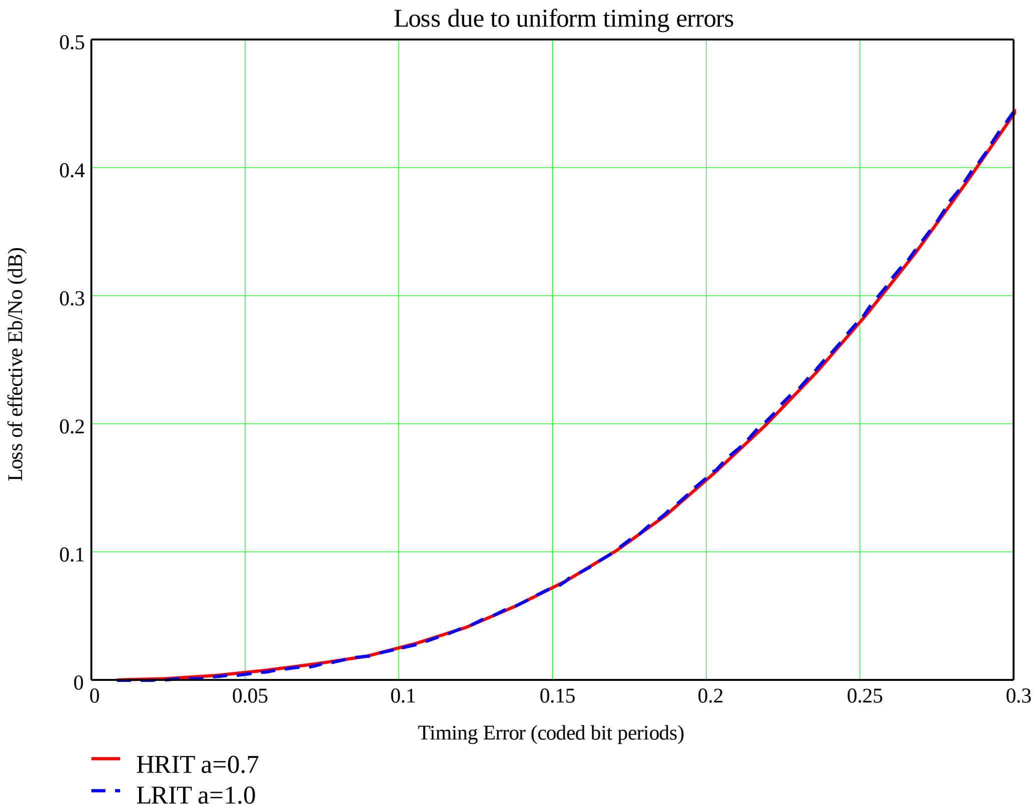
$derrF(t,a,en) := \log(target\_perr) - \log(UniformPerr(t,a,en))$

$ifunc(t,a) := if(UniformPerr(t,a,en \cdot chk) > target\_perr, 100, root(derrF(t,a,guess), guess))$

$lossF(t,a) := dB(ifunc(t,a)) - dB(en)$

$LuH_{ii} := lossF\left(dt_{ii}, aH\right)$

$LuL_{ii} := lossF\left(dt_{ii}, aL\right)$



Some checks on the system numerical range for Pb() used for Gaussian case.

$t := 0.0$

$\sigma := 0.02$

$$\alpha := \frac{1}{(\sigma \cdot 2 \cdot \pi)^2}$$

$$LoopSNR := \frac{\alpha}{2}$$

$dB(LoopSNR) = 15.006$

$$snrf(\sigma) := dB\left[\frac{1}{2 \cdot (\sigma \cdot 2 \cdot \pi)^2}\right]$$

$$I0\left[\frac{1}{(2 \cdot \pi \cdot \sigma)^2}\right] = 1.596 \cdot 10^{26}$$

$$\exp\left[\frac{\cos(2 \cdot \pi \cdot t)}{(2 \cdot \pi \cdot \sigma)^2}\right] = 3.177 \cdot 10^{27}$$

$$isnrf(snr) := \frac{1}{2 \cdot \pi} \cdot \sqrt{\frac{1}{2 \cdot idB(snr)}}$$

Finally perform the calculations for the "Gaussian" timing error. This uses the PLL probabillity distrabution function but re-scaled so that the values range from +/-0.5 in stead of +/-π. Here σ is the standard deviation, i.e. the RMS timing error.

$$Pb(t,\sigma) := \text{if}\left[\sigma<0.02, \frac{1}{\sigma\sqrt{2\cdot\pi}}\cdot\exp\left(-\frac{t^2}{2\cdot\sigma^2}\right), \exp\left[\frac{\cos(2\cdot\pi\cdot t)}{(2\cdot\pi\cdot\sigma)^2}\right]\cdot I0\left[\frac{1}{(2\cdot\pi\cdot\sigma)^2}\right]^{(-1)}\right]$$

$snrH := 15$

$snrL := 9$

$$snr_{ii} := snrL + \frac{(snrH - snrL)\cdot ii}{Imax}$$

$dt_{ii} := isnrf(snr_{ii})$

RMS timing error from the Loop SNR definition.

$Nsig := 8$

$$GaussPerr(\sigma,a,en) := 2\cdot\int_0^{\text{if}\left(\sigma<\frac{0.5}{Nsig}, Nsig\cdot\sigma, 0.5\right)} Pb(t,\sigma)\cdot peBIT(t,a,en)\,dt$$

$derrF(t,a,en) := \log(target\_perr) - \log(GaussPerr(t,a,en))$

$guess := en\cdot 1.01$

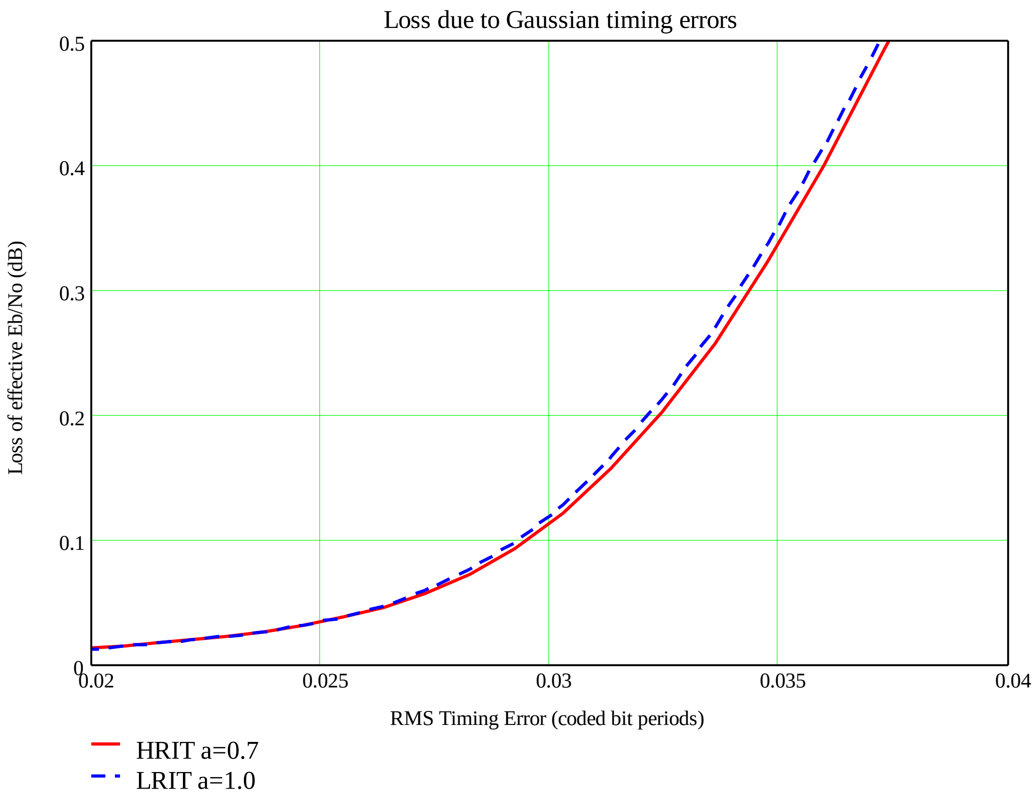
$chk := idB(1.5)$

$ifunc(t,a) := \text{if}(GaussPerr(t,a,en\cdot chk)>target\_perr, 100, root(derrF(t,a,guess), guess))$

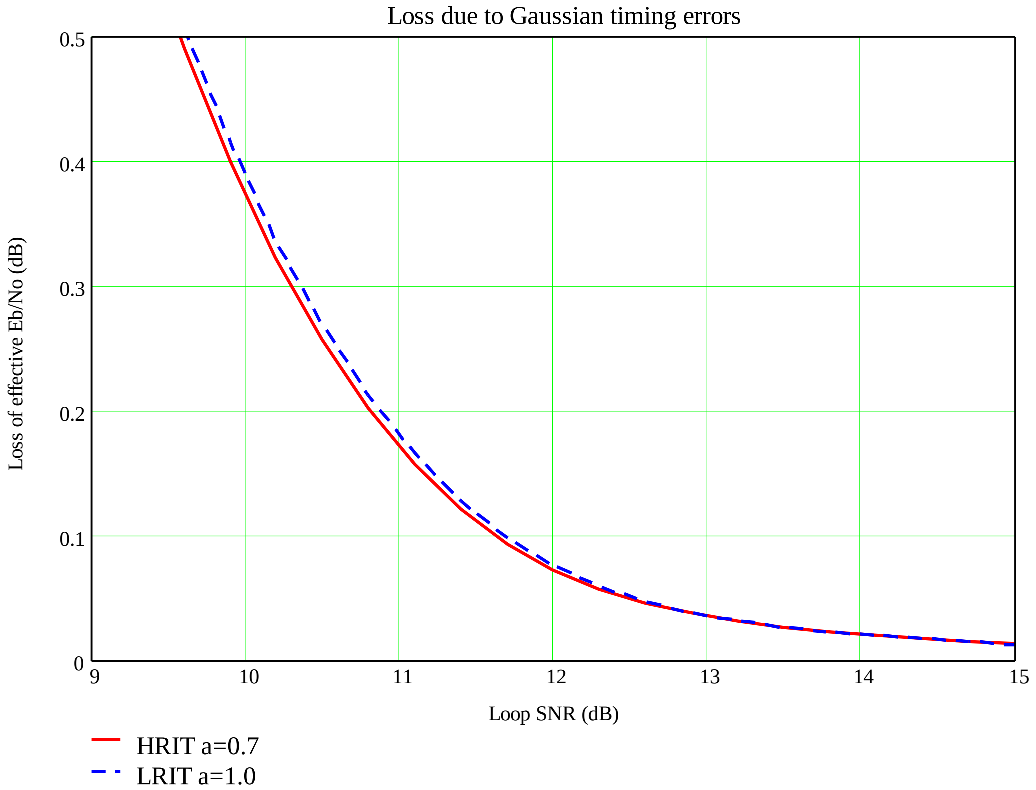
$lossF(t,a) := dB(ifunc(t,a)) - dB(en)$

$LgL_{ii} := lossF(dt_{ii}, aL)$

$LgH_{ii} := lossF(dt_{ii}, aH)$



Plot against LoopSNR for PLL discussion.



snr =

	0
5	10.5
6	10.8
7	11.1
8	11.4
9	11.7
10	12
11	12.3
12	12.6
13	12.9
14	13.2
15	13.5
16	13.8
17	14.1
18	14.4
19	14.7
20	15

dt =

	0
5	0.034
6	0.032
7	0.031
8	0.03
9	0.029
10	0.028
11	0.027
12	0.026
13	0.025
14	0.025
15	0.024
16	0.023
17	0.022
18	0.021
19	0.021
20	0.02

LgL =

	0
5	0.271
6	0.215
7	0.168
8	0.13
9	0.1
10	0.078
11	0.061
12	0.049
13	0.04
14	0.033
15	0.028
16	0.024
17	0.021
18	0.019
19	0.016
20	0.015

LgH =

	0
5	0.258
6	0.203
7	0.158
8	0.122
9	0.094
10	0.073
11	0.058
12	0.046
13	0.038
14	0.032
15	0.027
16	0.023
17	0.021
18	0.018
19	0.016
20	0.014

-- End of file --

File "BPSK_EDA.MCD"
Created with MathCAD 6.0-PLUS
Computation of carrier tracking loop for LRIT using BPSK, this with revised carrier PDF and more accurate CCSDS error rate formulae. This noe to plot as function of Es/No

$j := \sqrt{-1}$

$dB(x) := if\left(|x| < 10^{-12}, -120, 10\cdot\log(|x|)\right)$

Convert to dB
And back to numeric

$idB(x) := 10^{\frac{x}{10}}$

Convert radian^2 to degrees (RMS).

$dg(x) := \frac{180}{\pi}\cdot\sqrt{x}$

Bit rate

$BR := 128\cdot10^{3}$

Now for the transfer function of the carrier tracking PLL, 2nd order system, radian angular frequency.

$$HS2(s,\omega n,\zeta,T) := \frac{\left(2\cdot\zeta\cdot\omega n\cdot s + \omega n^{2}\right)\cdot\exp(-s\cdot T)}{s^{2} + \left(2\cdot\zeta\cdot\omega n\cdot s + \omega n^{2}\right)\cdot\exp(-s\cdot T)}$$

From Laplace 's' variable to Hz version:

$H(f,fn,\zeta,T) := HS2(j\cdot2\cdot\pi\cdot f, 2\cdot\pi\cdot fn, \zeta, T)$

The tracking error function (for power, i.e. angle^2):

$TE(f,fn,\zeta,T) := (|1 - H(f,fn,\zeta,T)|)^{2}$

Model the phase noise as flicker (f^3) and normal (f^2) over the frequency range of interest:

$L0 := idB(-88 + 60)$

Phase noise density in dBc/Hz at 1Hz offset, 60dB above 1kHz value,etc).

$f\_FL := 50$

Flicker noise corner frequency, Hz.

$$L(f,L0,f\_FL) := \frac{L0}{f^{2}}\cdot\left(1 + \frac{f\_FL}{f}\right)$$

Use the fact that CNo is approximatly constant over PLL range. Implement the fn to BR integration with a change of variable to make the points for evaluation spread in a 1/X manner to match the function shape.

$$BLoop(fn,\zeta,T) := \int_{0}^{fn} (|H(f,fn,\zeta,T)|)^{2}\,df \ldots + \int_{\frac{1}{BR}}^{\frac{1}{fn}} \frac{1}{f^{2}}\cdot\left(\left|H\left(\frac{1}{f},fn,\zeta,T\right)\right|\right)^{2}\,df$$

And the phase noise contribution:

$$MS2(fn,\zeta,L0,f\_FL,T) := 2\cdot\left[\int_{10^{-8}}^{fn} L(f,L0,f\_FL)\cdot TE(f,fn,\zeta,T)\,df\ ... + \int_{\frac{1}{BR}}^{\frac{1}{fn}} \frac{1}{f^2}\cdot L\left(\frac{1}{f},L0,f\_FL\right)\cdot TE\left(\frac{1}{f},fn,\zeta,T\right)\,df\right]$$

Add the discrete terms from the electronically despun array. The are 10 degrees peak to peak at the commutation rate (53.3Hz) and 3 degrees peak to peak at the spin rate of 100rpm (1.67Hz). These are approximate values measured at Dundee using the default MDD tracking loop (15.7Hz and d=1.0), the corrected values as power (i.e. radian^2) are:

$$K\_DIRECT\_PLL(deg\_pp,f) := \left(\frac{deg\_pp}{2\cdot\sqrt{2}}\cdot\frac{\pi}{180}\right)^2$$

$$K\_MDD\_PLL(deg\_pp,f) := \frac{K\_DIRECT\_PLL(deg\_pp,f)}{TE(f,15.7,1,0)}$$

$Kcr := K\_MDD\_PLL(10,53.3)$

$Ksp := K\_MDD\_PLL(3,1.67)$

EUMETSAT obtained 20 deg P-P as max from S/C specifications. Use this.

$Kcr := K\_DIRECT\_PLL(20,53.3)$

$Kcr = 0.015$

$Ksp = 2.738$

$MS3(fn,\zeta,Kcr,Ksp,T) := Kcr\cdot TE(53.3,fn,\zeta,T) + Ksp\cdot TE(1.67,fn,\zeta,T)$

Compute some example values for the system (before graphs for optimising the parameters).

$\zeta := 1.14$

Damping factor (dimensionless)

$T := 1\cdot 10^{-4}$

PLL loop filter time delay (seconds).

$$fn := \begin{pmatrix} 64 \\ 128 \\ 256 \end{pmatrix}$$

Natural frequency (Hz)

$imax := rows(fn)$

$ii := 0..imax - 1$

Compute the "static" parameters for varying Eb/No

$BL_{ii} := BLoop\left(fn_{ii},\zeta,T\right)$

PLL Noise bandwidth

$mm2_{ii} := MS2\left(fn_{ii},\zeta,L0,f\_FL,T\right)$

Phase Noise term

$mm3_{ii} := MS3\left(fn_{ii},\zeta,Kcr,Ksp,T\right)$

EDA phase effects.

Read the file with pre-calculated loss terms for the BPSK system. This allows us to calculate the effective C/No available to the PLL

$$Rc := \frac{223}{512}$$

Code rate.

$psk\_loss\_lut := READPRN(bpsk\_loss\_lut)$

$Nrows := rows(psk\_loss\_lut)$

$Nrows = 32$

$il := 0 .. Nrows - 1$

$idb\_EsNo_{il} := idB\left(psk\_loss\_lut_{il,0}\right)$

$db\_loss_{il} := psk\_loss\_lut_{il,3}$

Index choice is 1=DD, 2=ML, 3=V&V, 4=NthPower

$loss\_vs := cspline(idb\_EsNo, db\_loss)$

$loss\_ifunc(v) := -interp(loss\_vs, idb\_EsNo, db\_loss, v)$

$psk\_loss(v) := if\left(v<idb\_EsNo_0, 100, if\left(v>idb\_EsNo_{Nrows-1}, 0, loss\_ifunc(v)\right)\right)$

$$EffCNoFunc(ebno) := \frac{ebno \cdot BR}{idB(psk\_loss(ebno \cdot Rc))}$$

$$ms\_error(ebno, i) := \frac{BL_i}{EffCNoFunc(ebno)} + mm2_i + mm3_i$$

Set up the input Eb/No range. This excludes the 0.8dB implementation margin, i.e. this is the real demodulator input so the MSG should have 2.8+0.8 = 3.6dB normally.

$kmax := 39$

$kk := 0 .. kmax$

$Emin := -3$

$Emax := 6$

$$EbNo_{kk} := idB\left[(Emax - Emin) \cdot \frac{kk}{kmax} + Emin\right]$$

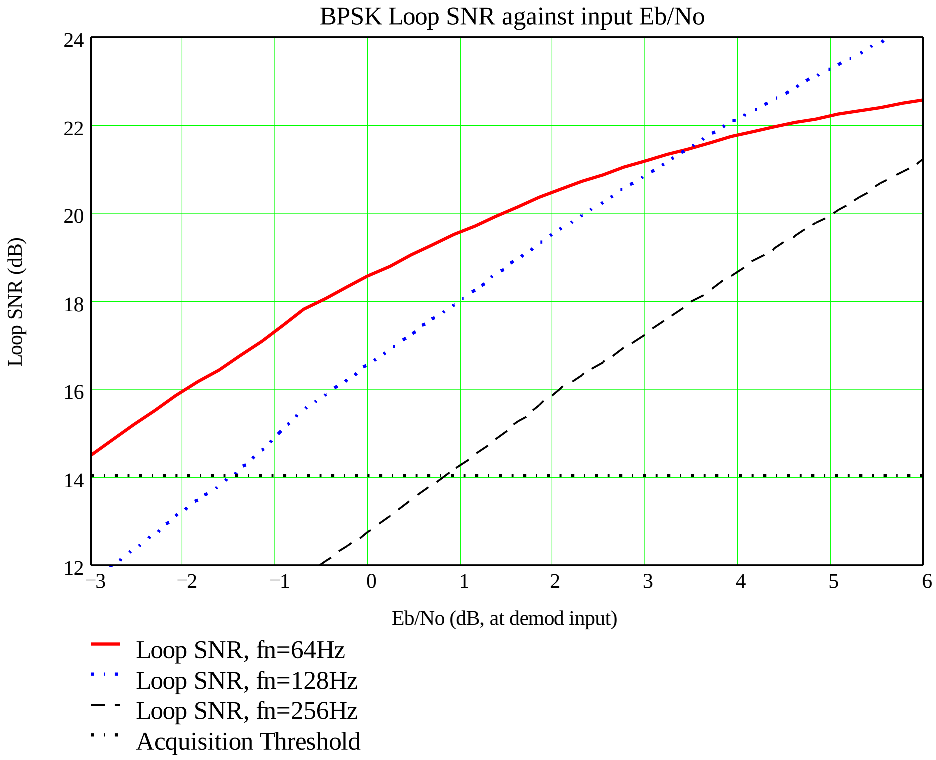
Compute the loop SNR (dB, in F.M. Gandner's notation).

$$snrL_{kk,ii} := dB\left(\frac{1}{2 \cdot ms\_error\left(EbNo_{kk}, ii\right)}\right)$$

$M := 2$

Acquisition threshold, 8dB (fastest sweep
and M^2 for M'th order recovery loop).

$snrACQ_{kk} := 8 + dB\left(M^2\right)$



$$\mathrm{TAve}(bl, snr) := \frac{2}{bl}\cdot\exp\left(\pi\cdot\frac{snr}{M^2}\right)$$

Approximate meant time to slip in seconds.

$$\mathrm{ta\_func}(ebno, i) := \mathrm{TAve}\left(BL_i, \frac{1}{2\cdot \mathrm{ms\_error}(ebno, i)}\right)\cdot\frac{1}{3600}$$

In hours

$$ta_{kk,ii} := \mathrm{ta\_func}\left(EbNo_{kk}, ii\right)$$

$$\mathrm{fail\_EbNo} := \mathrm{idB}(2.8 + 0.8 - 3)$$

$$\mathrm{dB}\left(\frac{1}{2\cdot \mathrm{ms\_error}(\mathrm{fail\_EbNo}, ii)}\right)$$

19.196
17.453
13.686

$$\mathrm{ta\_func}(\mathrm{fail\_EbNo}, ii)$$

$4.074\cdot 10^{22}$
$7.874\cdot 10^{12}$
31.844

Hours, but allow for 1/32 EDA time.

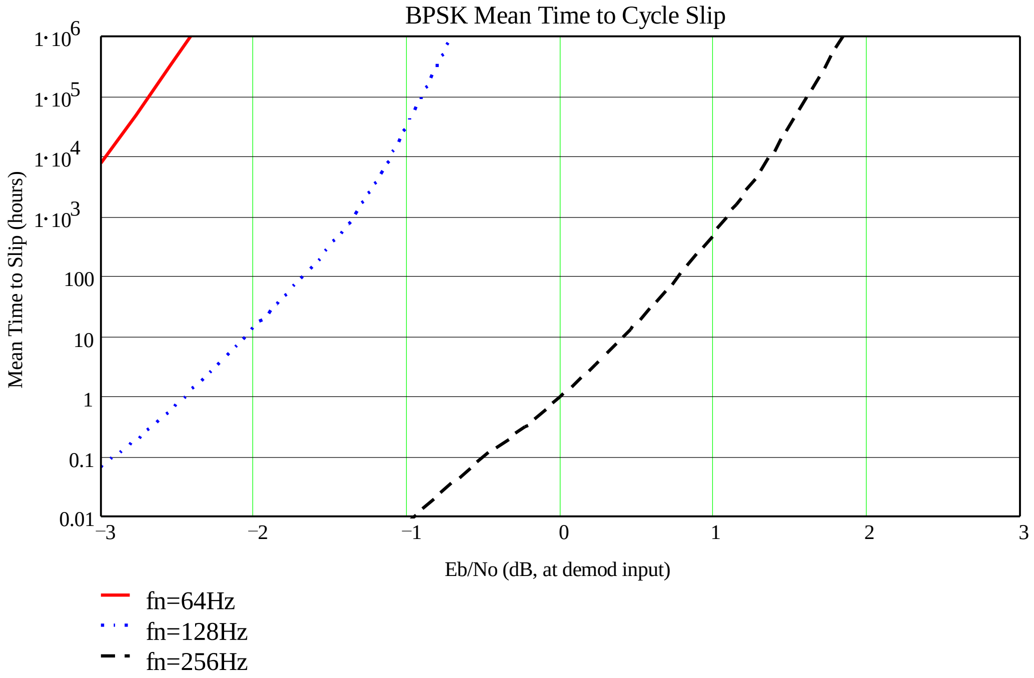
$$\mathrm{fail\_EbNo} := \mathrm{idB}(2.8 + 0.8 - 6)$$

$$\mathrm{dB}\left(\frac{1}{2\cdot \mathrm{ms\_error}(\mathrm{fail\_EbNo}, ii)}\right)$$

15.397
12.59
8.724

ta_func(fail_EbNo, ii)

$1.219 \cdot 10^{6}$
1.305
$1.186 \cdot 10^{-4}$



--End of File--

File "BPSKDET.MCD"
Created using MathCAD 6.0-PLUS
Computation of BPSK phase estimator performance.
This is VERY slow, if you want to run this file, plan a long coffee break (typically 3 minutes CPU time on 200MHz PentiumPro PC!
Do not set TOL to less than about 1E-3 for this file.

$$dB(x) := \text{if}\left(|x| < 10^{-12}, -120, 10\cdot\log(|x|)\right)$$

Convert to dB

$$idB(x) := 10^{\frac{x}{10}}$$

And back to numeric

$$efunc0(I,Q,\sigma) := I\cdot\text{if}(Q<0,-1,1)$$

Data-directed error detector.

$$efunc1(I,Q,\sigma) := I\cdot\tanh\left(\frac{Q}{\sigma^2}\right)$$

ML Approximation

$$efunc2(I,Q,\sigma) := \frac{2\cdot I\cdot Q}{\sqrt{I^2+Q^2}}$$

m(t)sin(2θ)

$$efunc3(I,Q,\sigma) := 2\cdot I\cdot Q$$

m^2(t)sin(2θ) (= 2nd power squaring loop)
Define the signal + 2-D noise PDF

$$pr(x,xx,y,yy,\sigma) := \frac{1}{2\cdot\pi\cdot\sigma^2}\cdot\exp\left[-\frac{(x-xx)^2+(y-yy)^2}{2\cdot\sigma^2}\right]$$

$$Nsig := 4.5$$

$$eps := 10^{-8}$$

For the hard-decision version divide in to regions to smooth the integration proceedure.

$$a0(xx,yy,\sigma) := \int_{eps}^{yy+Nsig\cdot\sigma}\int_{eps}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)\,dx\,dy\ ...$$
$$+\int_{yy-Nsig\cdot\sigma}^{-eps}\int_{eps}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)\,dx\,dy\ ..$$
$$+\int_{eps}^{yy+Nsig\cdot\sigma}\int_{xx-Nsig\cdot\sigma}^{-eps} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)\,dx\,dy\ ..$$
$$+\int_{yy-Nsig\cdot\sigma}^{-eps}\int_{xx-Nsig\cdot\sigma}^{-eps} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)\,dx\,dy$$

$$a1(xx,yy,\sigma) := \int_{yy-Nsig\cdot\sigma}^{yy+Nsig\cdot\sigma}\int_{xx-Nsig\cdot\sigma}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc1(x,y,\sigma)\,dx\,dy$$

$$a2(xx,yy,\sigma) := \int_{yy-Nsig\cdot\sigma}^{yy+Nsig\cdot\sigma}\int_{xx-Nsig\cdot\sigma}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc2(x,y,\sigma)\,dx\,dy$$

$$a3(xx,yy,\sigma) := \int_{yy-Nsig\cdot\sigma}^{yy+Nsig\cdot\sigma}\int_{xx-Nsig\cdot\sigma}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc3(x,y,\sigma)\,dx\,dy$$

And a set of P+1 log-spaced N/S points from "lower" to "upper".

$M := 2$

$P := 31$

$theta := \frac{2}{M} \cdot \frac{\pi}{180}$

Angle for numerical differencing.

$nn := \frac{\log(M)}{\log(2)}$

$lower := idB(-8)$

$V0 := 1$

$upper := idB(8)$

$x1 := V0 \cdot \sin(theta)$

$y1 := V0 \cdot \cos(theta)$

$kk := 0 .. P$

$x2 := V0 \cdot \sin(-theta)$

$y2 := V0 \cdot \cos(-theta)$

$a := lower$

$x1 = 0.0$

$y1 = 1$

$x2 = -0.0$

$y2 = 1$

$b := \exp\left(\frac{\ln\left(\frac{upper}{a}\right)}{P}\right)$

$efunc0(x1, y1, 1) = 0.0$

$efunc1(x1, y1, 1) = 0.0$

$en_{kk} := a \cdot b^{kk}$

Es/No values

$efunc2(x1, y1, 1) = 0.0$

$efunc3(x1, y1, 1) = 0.0$

$sigma_{kk} := \frac{V0}{\sqrt{2 \cdot en_{kk} \cdot nn}}$

Voltage standard deviation

Compute the detector shift for +/- theta

$avd0_{kk} := a0\left(x1, y1, sigma_{kk}\right) - a0\left(x2, y2, sigma_{kk}\right)$

$avd1_{kk} := a1\left(x1, y1, sigma_{kk}\right) - a1\left(x2, y2, sigma_{kk}\right)$

$avd2_{kk} := a2\left(x1, y1, sigma_{kk}\right) - a2\left(x2, y2, sigma_{kk}\right)$

$avd3_{kk} := a3\left(x1, y1, sigma_{kk}\right) - a3\left(x2, y2, sigma_{kk}\right)$

Compute the normalised phase sensitive detector values (dV/dθ)

$Kpsd0_{kk} := \frac{avd0_{kk}}{2 \cdot theta}$

$Kpsd1_{kk} := \frac{avd1_{kk}}{2 \cdot theta}$

$Kpsd2_{kk} := \frac{avd2_{kk}}{2 \cdot theta}$

$Kpsd3_{kk} := \frac{avd3_{kk}}{2 \cdot theta}$

Compute the reduction in Kpsd due to noise.

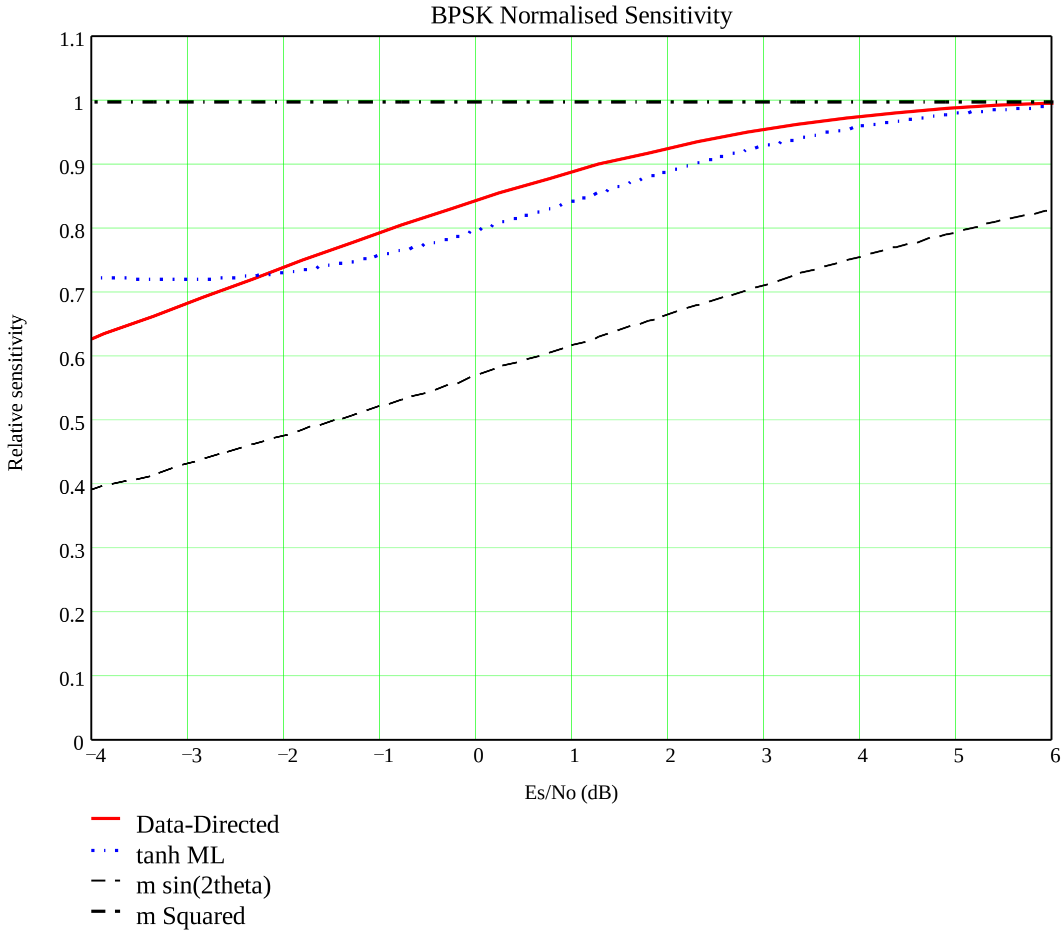
$$av0_{kk} := \frac{avd0_{kk}}{efunc0(x1, y1, sigma_{kk}) - efunc0(x2, y2, sigma_{kk})}$$

$$av1_{kk} := \frac{avd1_{kk}}{efunc1(x1, y1, sigma_{kk}) - efunc1(x2, y2, sigma_{kk})}$$

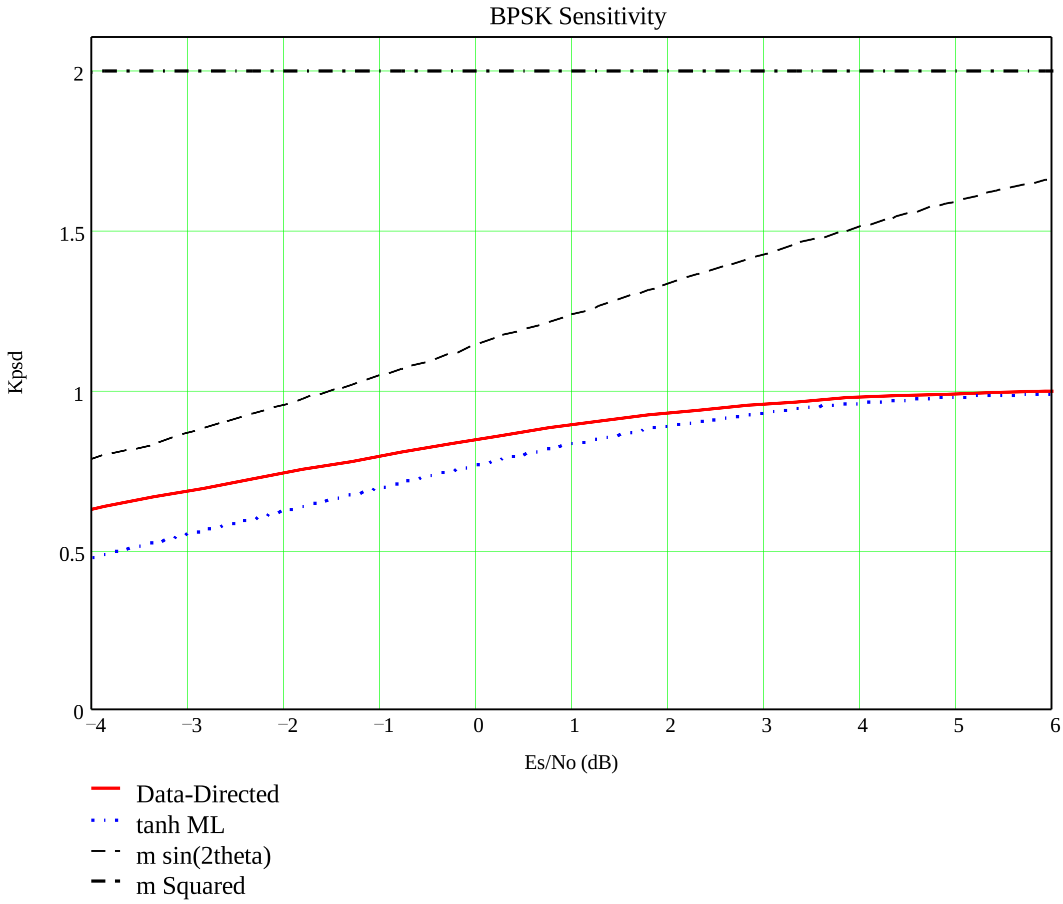
$$av2_{kk} := \frac{avd2_{kk}}{efunc2(x1, y1, sigma_{kk}) - efunc2(x2, y2, sigma_{kk})}$$

$$av3_{kk} := \frac{avd3_{kk}}{efunc3(x1, y1, sigma_{kk}) - efunc3(x2, y2, sigma_{kk})}$$

Plot out normalised detector sensitivity:



Plot out the Kpsd values for the V0 nominal signal level used.



Set up xx and yy for zero expected value, then calculate the standard deviation (mean square error) for the functions.

$theta := 0.0$

$x1 := V0 \cdot \sin(theta)$

$y1 := V0 \cdot \cos(theta)$

$$sd0(xx,yy,\sigma) := \int_{eps}^{yy+Nsig\cdot\sigma}\int_{eps}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)^2\,dx\,dy \ldots$$
$$+ \int_{yy-Nsig\cdot\sigma}^{-eps}\int_{eps}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)^2\,dx\,dy \ldots$$
$$+ \int_{eps}^{yy+Nsig\cdot\sigma}\int_{xx-Nsig\cdot\sigma}^{-eps} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)^2\,dx\,dy \ldots$$
$$+ \int_{yy-Nsig\cdot\sigma}^{-eps}\int_{xx-Nsig\cdot\sigma}^{-eps} pr(x,xx,y,yy,\sigma)\cdot efunc0(x,y,\sigma)^2\,dx\,dy$$

$$sd1(xx,yy,\sigma) := \int_{yy-Nsig\cdot\sigma}^{yy+Nsig\cdot\sigma}\int_{xx-Nsig\cdot\sigma}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc1(x,y,\sigma)^2\,dx\,dy$$

$$sd2(xx,yy,\sigma) := \int_{yy-Nsig\cdot\sigma}^{yy+Nsig\cdot\sigma}\int_{xx-Nsig\cdot\sigma}^{xx+Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc2(x,y,\sigma)^2\,dx\,dy$$

$$sd3(xx,yy,\sigma) := \int_{yy - Nsig\cdot\sigma}^{yy + Nsig\cdot\sigma} \int_{xx - Nsig\cdot\sigma}^{xx + Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc3(x,y,\sigma)^2\, dx\, dy$$

Compute vectors of mean squared values.

$$sv0_{kk} := sd0\left(x1, y1, sigma_{kk}\right)$$
$$sv1_{kk} := sd1\left(x1, y1, sigma_{kk}\right)$$
$$sv2_{kk} := sd2\left(x1, y1, sigma_{kk}\right)$$
$$sv3_{kk} := sd3\left(x1, y1, sigma_{kk}\right)$$

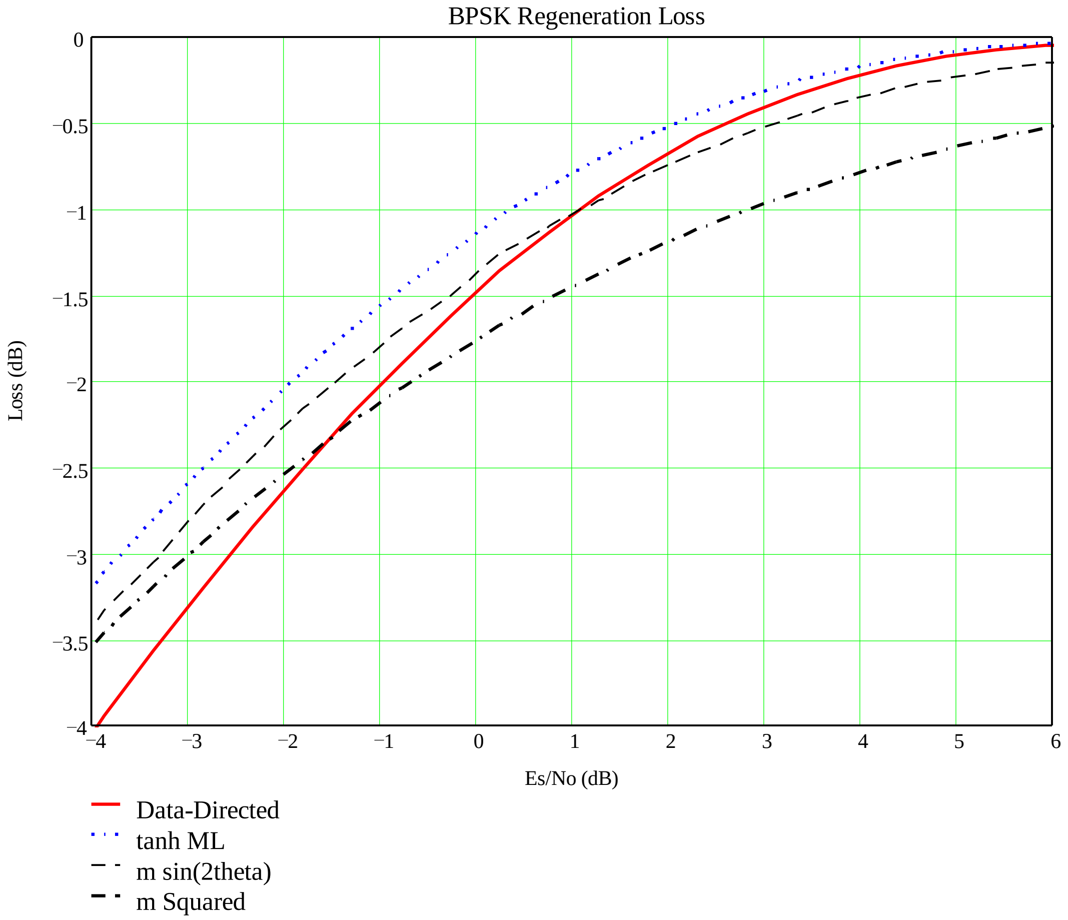
$$LossF(p) := \frac{1}{1 + \frac{1}{2\cdot p}}$$

The classical "squaring loss" function.

$$rsv0_{kk} := dB\left[\frac{\left(Kpsd0_{kk}\right)^2}{sv0_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$
$$rsv1_{kk} := dB\left[\frac{\left(Kpsd1_{kk}\right)^2}{sv1_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$
$$rsv2_{kk} := dB\left[\frac{\left(Kpsd2_{kk}\right)^2}{sv2_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$
$$rsv3_{kk} := dB\left[\frac{\left(Kpsd3_{kk}\right)^2}{sv3_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$
$$rsv4_{kk} := dB\left(LossF\left(en_{kk}\right)\right)$$
$$rsv5_{kk} := dB\left(erf\left(\sqrt{en_{kk}}\right)^2\right)$$

Analytic data-directed case.

Plot out the standard deviation relative to the detector sensitivity



$dben_{kk} := dB(en_{kk})$

Save results for later. We have the following data:

dB(Es/No) data-directed ML Msin(2θ) Squaring

$save_{kk,0} := dben_{kk}$

$save_{kk,1} := rsv0_{kk}$

$save_{kk,2} := rsv1_{kk}$

$save_{kk,3} := rsv2_{kk}$

$save_{kk,4} := rsv3_{kk}$

PRNPRECISION := 8

PRNCOLWIDTH := 15

WRITEPRN(bpsk_loss_lut) := save

-- End of file --

File "BPSKO6.MCD"
Created with MathCAD 6.0-PLUS
Computation of carrier tracking loop for LRIT using BPSK, this with revised carrier PDF and more accurate CCSDS error rate formulae.

$j := \sqrt{-1}$

$dB(x) := \mathrm{if}\left(|x| < 10^{-12}, -120, 10\cdot\log(|x|)\right)$

Convert to dB
And back to numeric

$idB(x) := 10^{\frac{x}{10}}$

Convert radian^2 to degrees (RMS).

$dg(x) := \frac{180}{\pi}\cdot\sqrt{x}$

Bit rate

$BR := 128\cdot 10^{3}$

$SR := BR\cdot\frac{512}{223}$

Symbol rate with coding

$EbNo := idB(2.7 + 0.8)$

Set up Eb/No and include tech loss margin.

$EbNo = 2.239$

(numeric)

$CNo := BR\cdot EbNo$

Carrier/Noise density ratio.

$dB(CNo) = 54.572$

Now for the transfer function of the carrier tracking PLL, 2nd order system, radian angular frequency.

$$HS2(s, \omega n, \zeta, T) := \frac{\left(2\cdot\zeta\cdot\omega n\cdot s + \omega n^{2}\right)\cdot\exp(-s\cdot T)}{s^{2} + \left(2\cdot\zeta\cdot\omega n\cdot s + \omega n^{2}\right)\cdot\exp(-s\cdot T)}$$

From Laplace 's' variable to Hz version:

$H(f, fn, \zeta, T) := HS2(j\cdot 2\cdot\pi\cdot f, 2\cdot\pi\cdot fn, \zeta, T)$

The tracking error function (for power, i.e. angle^2):

$TE(f, fn, \zeta, T) := \left(|1 - H(f, fn, \zeta, T)|\right)^{2}$

Model the phase noise as flicker (f^3) and normal (f^2) over the frequency range of interest:

$L0 := idB(-88 + 60)$

Phase noise density in dBc/Hz at 1Hz offset, 60dB above 1kHz value,etc).

$f\_FL := 50$

Flicker noise corner frequency, Hz.

$$L(f, L0, f\_FL) := \frac{L0}{f^{2}}\cdot\left(1 + \frac{f\_FL}{f}\right)$$

Some sample output values:

$dB(L(10, L0, f\_FL)) = -40.218$

$dB\left(L\left(10^{3}, L0, f\_FL\right)\right) = -87.788$

Compute integrated phase noise. This uses a change of variable so the 1/x evaluation pattern suits the function shape.

$$Ipn := 2\cdot\int_{\frac{1}{10^6}}^{\frac{1}{10}} \frac{1}{f^2}\cdot L\left(\frac{1}{f}, L0, f\_FL\right) df$$

$dg(Ipn) = 1.908$

MSG spec calls for < 2deg RMS phase noise from 10Hz to 1MHz

Compute the loss in effective C/No due to the non-linear Costas (BPSK) and extended Costas (QPSK) demodulators:

$$EsNo := EbNo\cdot\frac{223}{512}$$

Symbol signal energy in this case.

Read the file with pre-calculated loss terms for the BPSK system. This allows us to calculate the effective C/No available to the PLL

$psk\_loss\_lut := READPRN(bpsk\_loss\_lut)$

$Nrows := rows(psk\_loss\_lut)$

$Nrows = 32$

$il := 0 .. Nrows - 1$

$idb\_EsNo_{il} := idB\left(psk\_loss\_lut_{il,0}\right)$

$TypeIdx := 3$

Index choice is 1=DD, 2=ML, 3=V&V, 4=NthPower

$db\_loss_{il} := psk\_loss\_lut_{il,TypeIdx}$

$loss\_vs := cspline(idb\_EsNo, db\_loss)$

$loss\_ifunc(v) := -interp(loss\_vs, idb\_EsNo, db\_loss, v)$

$psk\_loss(v) := if\left(v<idb\_EsNo_0, 100, if\left(v>idb\_EsNo_{Nrows-1}, 0, loss\_ifunc(v)\right)\right)$

$psk\_loss(idB(2.7 - 3.6 + 0.8 - 0)) = 1.434$

$AcqLoss := 0$

Set to zero for tracking, or 3dB for acquisition to account for the non-optimum samples or to simulate EDA failed condition.

$$ImLoss := \frac{1}{idB(psk\_loss(EsNo) + AcqLoss)}$$

The "squaring loss" term.

$dB(ImLoss) = -1.439$

$dB(CNo\cdot ImLoss) = 53.133$

Effective carrier/noise density in loop.

Use the fact that CNo is approximatly constant over PLL range. Implement the fn to BR integration with a change of variable to make the points for evaluation spread in a 1/X manner to match the function shape.

$$MS1(fn,\zeta,T) := \frac{1}{CNo\cdot ImLoss}\cdot\left(\int_0^{fn}\left(\left|H(f,fn,\zeta,T)\right|\right)^2 df \ldots + \int_{\frac{1}{BR}}^{\frac{1}{fn}} \frac{1}{f^2}\cdot\left(\left|H\left(\frac{1}{f},fn,\zeta,T\right)\right|\right)^2 df\right)$$

And the phase noise contribution:

$$MS2(fn,\zeta,L0,f\_FL,T) := 2\cdot\left|\int_{10^{-8}}^{fn} L(f,L0,f\_FL)\cdot TE(f,fn,\zeta,T)\,df \ldots + \int_{\frac{1}{BP}}^{\frac{1}{fn}} \frac{1}{f^2}\cdot L\left(\frac{1}{f},L0,f\_FL\right)\cdot TE\left(\frac{1}{f},fn,\zeta,T\right)\,df\right|$$

Add the discrete terms from the electronically despun array. The are 10 degrees peak to peak at the commutation rate (53.3Hz) and 3 degrees peak to peak at the spin rate of 100rpm (1.67Hz). These are approximate values measured at Dundee using the default MDD tracking loop (15.7Hz and d=1.0), the corrected values as power (i.e. radian^2) are:

$$K\_DIRECT\_PLL(deg\_pp,f) := \left(\frac{deg\_pp}{2\cdot\sqrt{2}}\cdot\frac{\pi}{180}\right)^2$$

$$K\_MDD\_PLL(deg\_pp,f) := \frac{K\_DIRECT\_PLL(deg\_pp,f)}{TE(f,15.7,1,0)}$$

$Kcr := K\_MDD\_PLL(10,53.3)$

$Ksp := K\_MDD\_PLL(3,1.67)$

EUMETSAT obtained 20 deg P-P as max from S/C specifications. Use this.

$Kcr := K\_DIRECT\_PLL(20,53.3)$

$Kcr = 0.015$

$Ksp = 2.738$

$MS3(fn,\zeta,Kcr,Ksp,T) := Kcr\cdot TE(53.3,fn,\zeta,T) + Ksp\cdot TE(1.67,fn,\zeta,T)$

Compute some example values for the system (before graphs for optimising the parameters).

$\zeta := 1.14$

Damping factor (dimensionless)

$fn := 90$

Natural frequency (Hz)

$T := 10^{-4}$

PLL loop filter time delay (seconds).

$mm1 := MS1(fn,\zeta,T)$

Noise (thermal)

$mm2 := MS2(fn,\zeta,L0,f\_FL,T)$

Phase Noise

$mm3 := MS3(fn,\zeta,Kcr,Ksp,T)$

EDA phase effects.

$mm3a := Ksp\cdot TE(1.67,fn,\zeta,T)$

EDA spin effect only (A)

$mm3b := Kcr\cdot TE(53.3,fn,\zeta,T)$

EDA commutation only (B)

Evaluate the relative magnitude of the different effects:

$mtotal := mm1 + mm2 + mm3a + mm3b$

$dg(mtotal) = 3.149$

$$\frac{mm1}{mtotal} = 0.707$$

$$\frac{mm2}{mtotal} = 0.012$$

$$\frac{mm3a}{mtotal} = 1.073\cdot 10^{-4}$$

$$\frac{mm3b}{mtotal} = 0.28$$

Set up a function for computing the total sum square error.

$$TMS(fn, \zeta, L0, f\_FL, Kcr, Ksp, T) := MS1(fn, \zeta, T) \ldots + MS2(fn, \zeta, L0, f\_FL, T) \ldots + MS3(fn, \zeta, Kcr, Ksp, T)$$

And a set of P log-spaced frequency points from "lower" to "upper".

$P := 20$

$lower := 30$

$upper := 500$

$kk := 0 .. P$

$a := lower$

$$b := \exp\left(\frac{\ln\left(\frac{upper}{a}\right)}{P}\right)$$

$vfn_{kk} := a \cdot b^{kk}$

Now compute performance with three delay values.

$mse_{kk} := TMS\left(vfn_{kk}, \zeta, L0, f\_FL, Kcr, Ksp, 0\right)$

Zero Delay

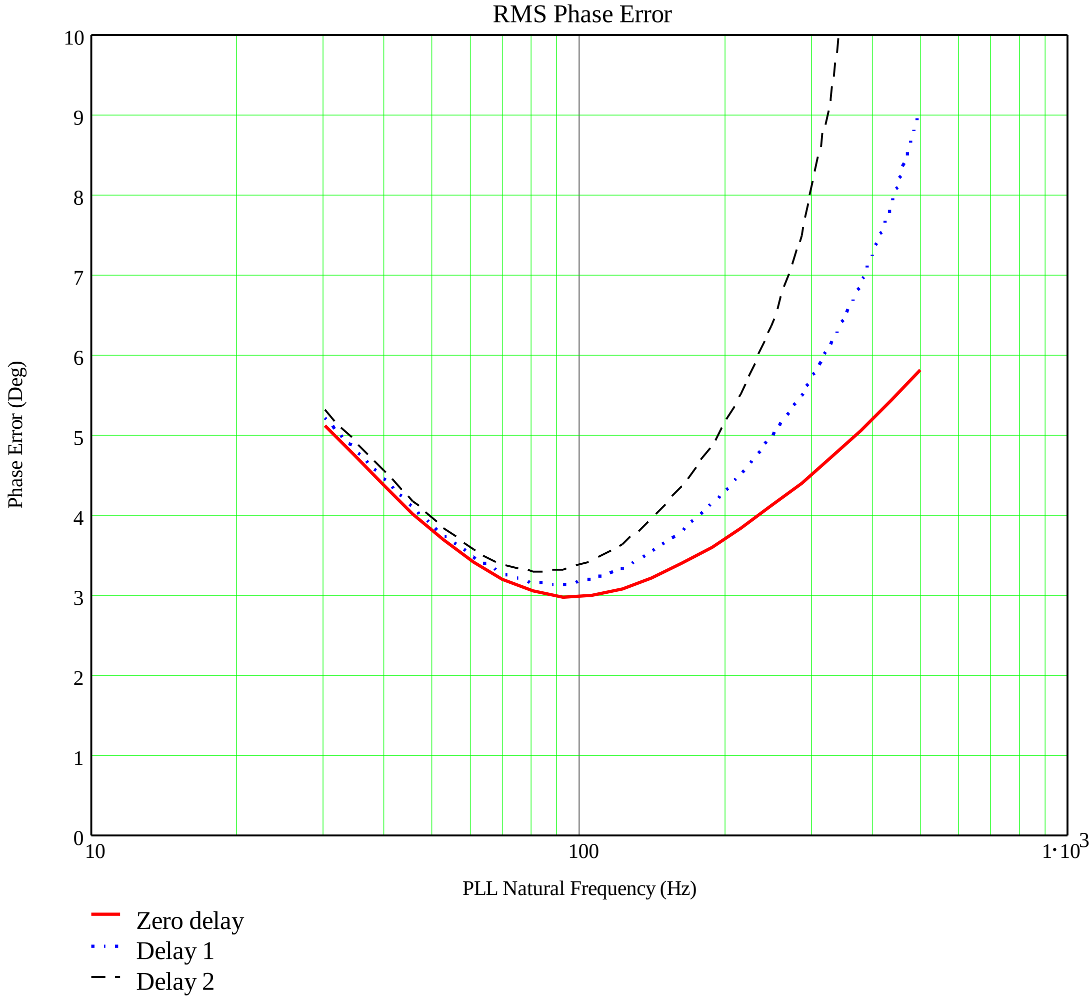
$mseD1_{kk} := TMS\left(vfn_{kk}, \zeta, L0, f\_FL, Kcr, Ksp, 10^{-4}\right)$

Delay #1

$mseD2_{kk} := TMS\left(vfn_{kk}, \zeta, L0, f\_FL, Kcr, Ksp, 2 \cdot 10^{-4}\right)$

Delay #2

Plot out these curves to find the "optimum" choice of loop natural frequency for minimum total phase error in the recovered carrier.



Compute the loop SNR (dB, in F.M. Gandner's notation).

$$snrl_{kk} := dB\left(\frac{1}{2\cdot mse_{kk}}\right)$$

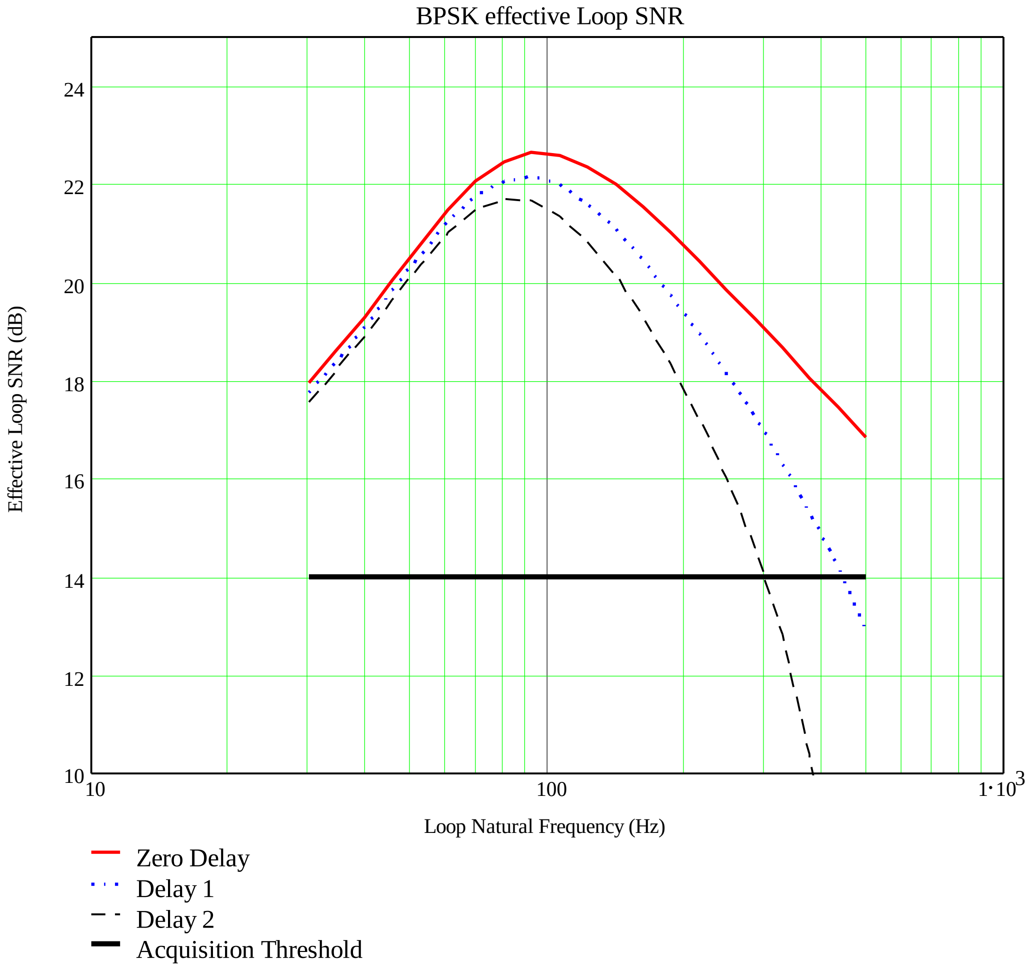
$$snrlD1_{kk} := dB\left(\frac{1}{2\cdot mseD1_{kk}}\right)$$

$$snrlD2_{kk} := dB\left(\frac{1}{2\cdot mseD2_{kk}}\right)$$

$$M := 2$$

Acquisition threshold, 8dB (fastest sweep
and M^2 for M'th order recovery loop).

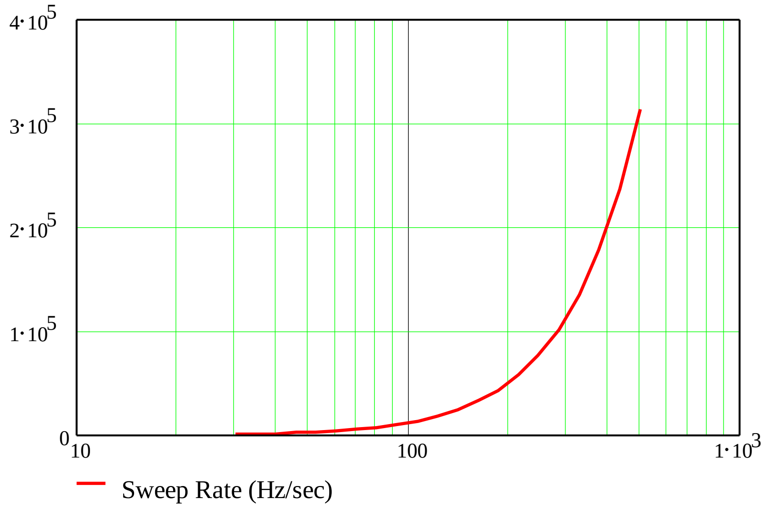
$$snrACQ_{kk} := 8 + dB\left(M^2\right)$$



Some sweep rate computations (Delay #1 only):

$$\mathrm{rate2}(p, wn) := wn^2 \cdot \mathrm{if}\left[p > \frac{9.5}{2}, 0.4, \mathrm{if}\left[p < \frac{6}{2}, 0, \left(1 - \sqrt{\frac{1}{p-2}}\right)\right]\right]$$

$$\mathrm{sweep}_{kk} := \frac{\mathrm{rate2}\left(\frac{1}{2 \cdot \mathrm{mseD1}_{kk}} \cdot \frac{1}{M^2}, \mathrm{vfn}_{kk} \cdot 2 \cdot \pi\right)}{2 \cdot \pi \cdot M}$$



Function that uses the simulated (and saved above) Viterbi decoder results to model the CCSDS error rate versus Eb/No more accurately.

$ccsds := READPRN(ccsds\_lut)$

$Nccsds := rows(ccsds)$

$Nccsds = 32$

$icc := 0 .. Nccsds - 1$

$cc\_eb_{icc} := ccsds_{icc,0}$

$cc\_logPE_{icc} := ccsds_{icc,1}$

$cc\_vs := cspline(cc\_eb, cc\_logPE)$

$cc\_ifunc(v) := 10^{interp(cc\_vs, cc\_eb, cc\_logPE, v)}$

$vMax := ccsds_{Nccsds - 1, 0}$

$ErrMin := 10^{ccsds_{Nccsds - 1, 1}}$

$pef(v) := if(v<0, 0.5, if(v>vMax, ErrMin, cc\_ifunc(v)))$

And an inverse error rate formulae (requires care to prevent convergence failure):

$IpeLIM := vMax$

Maximum returned value.

$peLIM := ErrMin$

Corresponding input value

$guess := idB(2.7)$

$pediff(erate, guess) := log(erate) - log(pef(guess))$

Use logs to remove slope

$ipe(erate, guess) := root(pediff(erate, guess), guess)$

$INVpe(r) := if(r>peLIM, ipe(r, guess), IpeLIM)$

$p := pef(idB(2.7))$

$p = 2.472 \cdot 10^{-8}$

Quick test of inverse function.

$dB(INVpe(p)) = 2.7$

And the function for the probability distrabution of the PLL phase. Here M is the number of stable points (i.e order of the PLL recovery process). With 1 for CW PLL (or residual carrier PSK), 2 = BPSK and 4 = QPSK. There is an odd problem with MathCAD and the exp()/I0() term - it is more reliable to use power -1 for some odd reason.

$\alpha(prms) := \frac{1}{prms}$

'Loop SNR' used by Spilker, = 2*SNRloop of Gardner

$$Pb(\theta,\alpha,M) := \text{if}\left[\frac{\alpha}{M^2}>50, \sqrt{\frac{\alpha}{2\cdot\pi}}\cdot\exp\left(\frac{-\alpha\cdot\theta^2}{2}\right), M\cdot\exp\left(\frac{\alpha}{M^2}\cdot\cos(M\cdot\theta)\right)\cdot\left(2\cdot\pi\cdot I0\left(\frac{\alpha}{M^2}\right)\right)^{-1}\right]$$

Now evaluate the probability of error (include phasor cross talk) and the loss involved in the typical case.

$$E1(en,\theta) := pef\left[en\cdot(\cos(\theta)+\sin(\theta))^2\right]$$

Signals add

$$E2(en,\theta) := pef\left[en\cdot(\cos(\theta)-\sin(\theta))^2\right]$$

Signals subtract

No cross talk case.

$$E3(en,\theta) := pef\left(en\cdot\cos(\theta)^2\right)$$

$$peBPSK(en,\theta) := E3(en,\theta)$$

$$peSQPSK(en,\theta) := 0.5\cdot E3(en,\theta) + 0.25\cdot E1(en,\theta) + 0.25\cdot E2(en,\theta)$$

Uncoded case

$$peQPSK(en,\theta) := 0.5\cdot E1(en,\theta) + 0.5\cdot E2(en,\theta)$$

$$Kfec := 2$$

$$Rc := \frac{223}{512}$$

$$peQPSK(en,\theta) := pef\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right)+Kfec\cdot Rc\cdot\sin(\theta)^2}\right]$$

Coded case.

Perform the integration using the assumed symmetry of Pb() and use choice of region to avoid integration convergance accuracy problems.

$$Sdev := 8$$

Number of "standard deviations" to be integrated.

Create a general integration function which uses Sdev or π/M, whichever is smaller. The calling values are the Eb/No input, the α = (1/mean square error), the number of phase values (2=BPSK, 4=QPSK) and the Probabillity of Error Function specific to the system under consideration.

$$IntFunc(en,\alpha,M,Pef) := 2\cdot\int_0^{\text{if}\left[\alpha<\left(\frac{M\cdot Sdev}{\pi}\right)^2, \frac{\pi}{M}, \frac{Sdev}{\sqrt{\alpha}}\right]} Pb(\theta,\alpha,M)\cdot Pef(en,\theta)\,d\theta$$

$$Perr(en,\alpha) := IntFunc(en,\alpha,2,peBPSK)$$

Now find the error rates, and convert this back to the equivalent loss in Eb/No for the system:

$$target\_perr := 5\cdot 10^{-9}$$

$$en := INVpe(target\_perr)$$

$$dB(en) = 2.752$$

$$guess := en\cdot 1.01$$

$$derrF(\alpha,en) := \log(target\_perr) - \log(Perr(en,\alpha))$$

Apply test to see if > 1.5dB loss, is so report 20dB (avoid convergance problems)

$$lossF(\alpha) := dB(\text{if}(Perr(en\cdot idB(1.5),\alpha)>target\_perr, 100, root(derrF(\alpha,guess),guess))) - dB(en)$$

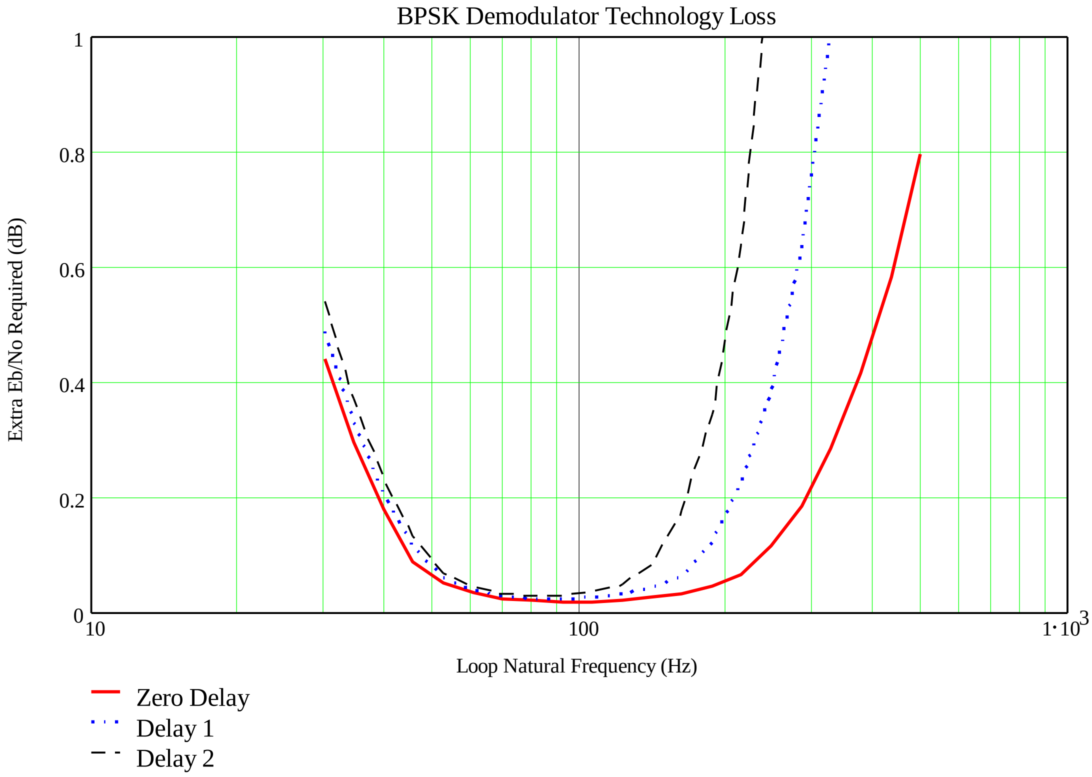
$$kk2 := 0..P$$

$$Lmse_{kk2} := lossF\left(\alpha\left(mse_{kk2}\right)\right)$$

$$LmseD1_{kk2} := lossF\left(\alpha\left(mseD1_{kk2}\right)\right)$$

$$LmseD2_{kk2} := lossF\left(\alpha\left(mseD2_{kk2}\right)\right)$$

Plot out technology loss (dB) against natural frequency:



vfn =

	0
0	30
1	34.531
2	39.747
3	45.751
4	52.661
5	60.615
6	69.771
7	80.31
8	92.44
9	106.403
10	122.474
11	140.974
12	162.267
13	186.777
14	214.989
15	247.462

LmseD1 =

	0
0	0.488
1	0.332
2	0.205
3	0.116
4	0.06
5	0.04
6	0.03
7	0.026
8	0.025
9	0.027
10	0.032
11	0.043
12	0.065
13	0.125
14	0.225
15	0.387

-- End of file --

File "PHASE1.MCD"
Created with MathCAD 6.0-PLUS
Simulation of system sensitivity to carrier phase noise, this version updated with "per symbol" computation of the QPSK crosstalk problems.

$dB(x) := if\left(|x|<10^{-12}, -120, 10\cdot\log(|x|)\right)$

Convert to dB
And back to numeric

$idB(x) := 10^{\frac{x}{10}}$

Define the function for the error probability in the antipodal (uncoded) case:

$peu(v) := if\left(v<0.0, 0.5, if\left(v>30, 4.718\cdot10^{-15}, cnorm\left(-\sqrt{2\cdot v}\right)\right)\right)$

Define the function for the error probability approximating to the coded cases, r=1/2 Viterbi convolution code fisrt:

$$logPE := \begin{bmatrix} -\log(0.5) \\ 0.529 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 7 \\ 10 \\ 12 \\ 14 \end{bmatrix}$$

$$eqEB := \begin{bmatrix} 0 \\ idB(-1) \\ idB(0.6) \\ idB(1.9) \\ idB(2.8) \\ idB(3.7) \\ idB(4.5) \\ idB(5.6) \\ idB(6.7) \\ idB(7.4) \\ idB(8) \end{bmatrix}$$

$vs := cspline(eqEB, logPE)$

$ifunc(x) := 10^{-interp(vs, eqEB, logPE, x)}$

$pev(v) := if\left(v<0, 0.5, if\left(v>idB(8), 10^{-14}, ifunc(v)\right)\right)$

Function that uses the simulated (and saved above) Viterbi decoder results to model the CCSDS error rate versus Eb/No more accurately.

$ccsds := READPRN(ccsds\_lut)$

$Nccsds := rows(ccsds)$

$Nccsds = 32$

$icc := 0 .. Nccsds - 1$

$cc\_eb_{icc} := ccsds_{icc,0}$

$cc\_logPE_{icc} := ccsds_{icc,1}$

$cc\_vs := cspline(cc\_eb, cc\_logPE)$

$cc\_ifunc(v) := 10^{interp(cc\_vs, cc\_eb, cc\_logPE, v)}$

$vMax := ccsds_{Nccsds-1,0}$

$ErrMin := 10^{ccsds_{Nccsds-1,1}}$

$pec(v) := if(v<0, 0.5, if(v>vMax, ErrMin, cc\_ifunc(v)))$

And an inverse error rate formulae (requires care to prevent convergence failure):

$IpeLIM := vMax$

Maximum returned value.

$peLIM := ErrMin$

Corresponding input value

$guess := idB(2.7)$

$pediff(erate, guess) := log(erate) - log(pec(guess))$

Use logs to remove slope

$ipe(erate, guess) := root(pediff(erate, guess), guess)$

$INVpec(r) := if(r>peLIM, ipe(r, guess), IpeLIM)$

$p := pec(idB(2.7))$

$p = 2.472 \cdot 10^{-8}$

Quick test of inverse function.

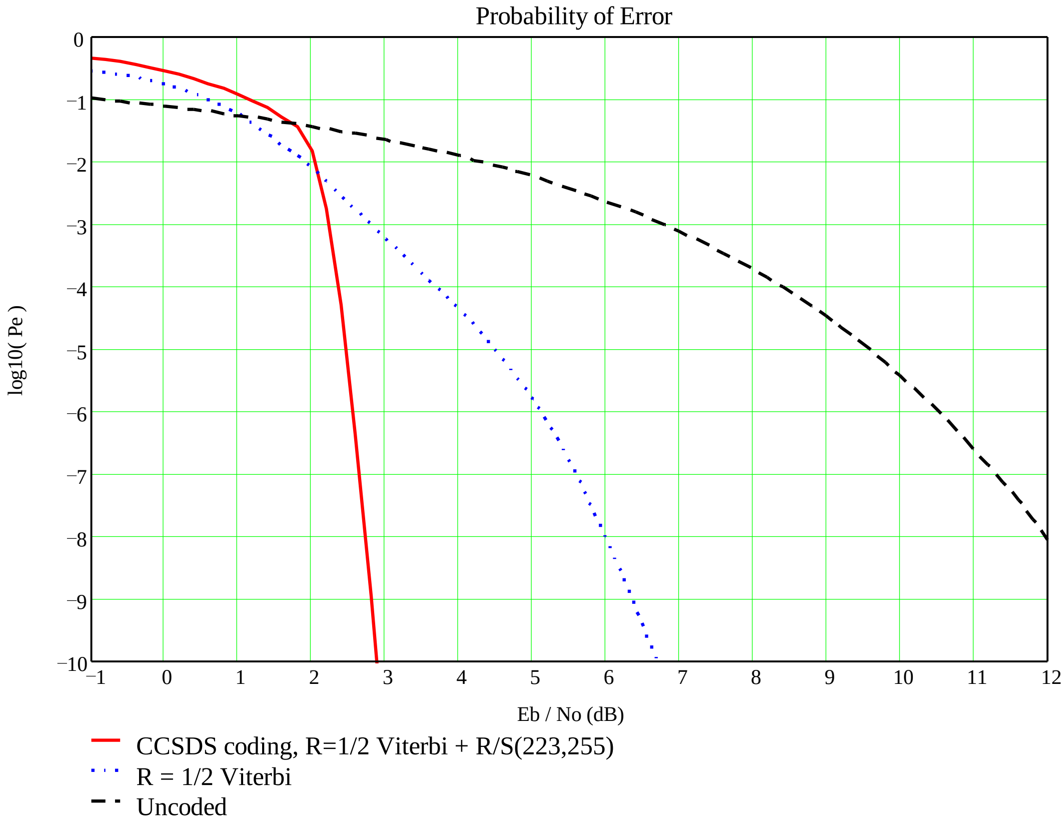
$dB(INVpec(p)) = 2.7$

Plot these function out for a typical range of Eb/No:

$jmax := 65$

$jj := 0..jmax$

$eb_{jj} := idB\left(\frac{jj}{5} - 1\right)$



And the other inverse error rate formulae (requires care to prevent convergence failure):

Uncoded case:

IpeuLIM := 25

peuLIM := peu(IpeuLIM)

$peuLIM = 7.687 \cdot 10^{-13}$

peudiff(erate, guess) := log(erate) − log(peu(guess))

ipeu(erate, guess) := root(peudiff(erate, guess), guess)

guessu := idB(8)

INVpeu(r) := if(r>peuLIM, ipeu(r, guessu), IpeuLIM)

Run a check to see if it works OK.

p := peu(idB(8))

$p = 1.909 \cdot 10^{-4}$

dB(INVpeu(p)) = 8.000000

Viterbi coded case:

IpevLIM := 5.978

pevLIM := pev(IpevLIM)

$pevLIM = 7.152 \cdot 10^{-14}$

pevdiff(erate, guess) := log(erate) − log(pev(guess))

ipev(erate, guess) := root(pevdiff(erate, guess), guess)

guessf := idB(4)

INVpev(r) := if(r>pevLIM, ipev(r, guessf), IpevLIM)

Run a check to see if it works OK.

p := pev(idB(4))

$p = 4.472 \cdot 10^{-5}$

dB(INVpev(p)) = 4.000000

And the function for the probability distrabution of the PLL phase. Here M is the number of stable points. With 1 for CW PLL (or residual carrier PSK), 2 = BPSK and 4 = QPSK. There is an odd problem with MathCAD and the exp()/I0() term - it is more reliable to use power -1 for some odd reason.

$$\alpha(prms) := \frac{1}{prms}$$

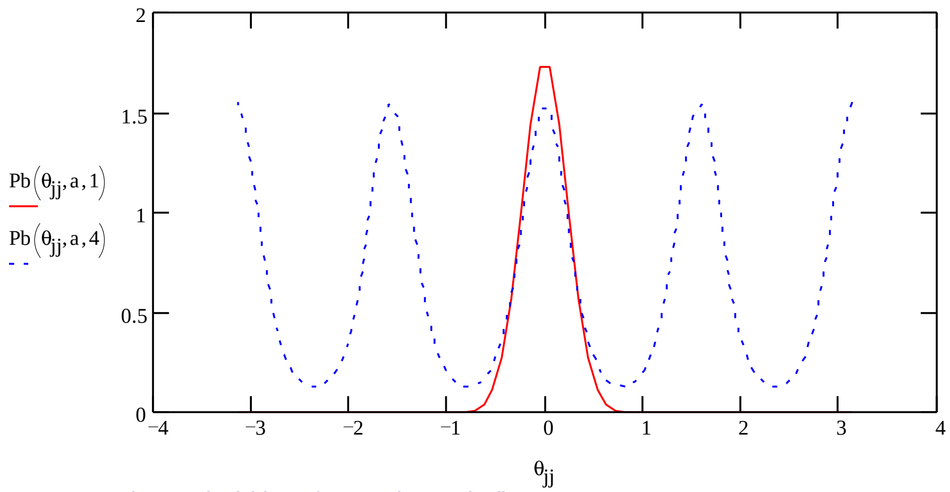
'Loop SNR' used by Spilker, = 2*SNRloop of Gardner

$$Pb(\theta,\alpha,M) := \mathrm{if}\left[\frac{\alpha}{M^2}>50, \sqrt{\frac{\alpha}{2\cdot\pi}}\cdot\exp\left(\frac{-\alpha\cdot\theta^2}{2}\right), M\cdot\exp\left(\frac{\alpha}{M^2}\cdot\cos(M\cdot\theta)\right)\cdot\left(2\cdot\pi\cdot I0\left(\frac{\alpha}{M^2}\right)\right)^{-1}\right]$$

Test plot of this (low 'a=20' for QPSK, but shows differences better.

$$a := 20$$

$$\theta_{jj} := \left(\frac{jmax}{2} - jj\right)\cdot\frac{2\cdot\pi}{jmax}$$



Now evaluate the probability of error (uncoded):

$$EbNou := idB(12.15)$$

$$peu(EbNou) = 5.076\cdot10^{-9}$$

$$E1u(en,\theta) := peu\left[en\cdot(\cos(\theta)+\sin(\theta))^2\right]$$

Signals add

$$E2u(en,\theta) := peu\left[en\cdot(\cos(\theta)-\sin(\theta))^2\right]$$

Signals subtract

$$E3u(en,\theta) := peu\left(en\cdot\cos(\theta)^2\right)$$

No cross talk case.

$$peBPSKu(en,\theta) := E3u(en,\theta)$$

$$peSQPSKu(en,\theta) := 0.5\cdot E3u(en,\theta) + 0.25\cdot E1u(en,\theta) + 0.25\cdot E2u(en,\theta)$$

$$peQPSKu(en,\theta) := 0.5\cdot E1u(en,\theta) + 0.5\cdot E2u(en,\theta)$$

Use the uncoded case to check assumptions about Kfec

$$peQPSKu2(en,\theta,K1) := peu\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right) + K1\cdot\sin(\theta)^2}\right]$$

Approximation for coded QPSK case.

$$efunc(en,\theta,guess) := \log(peQPSKu(en,\theta)) - \log(peQPSKu2(en,\theta,guess))$$

$$guess := 1$$

$$en := INVpeu\left(10^{-6}\right)$$

$\theta := 1 \cdot \frac{\pi}{180}$

$\mathrm{root}(\mathrm{efunc}(en, \theta, guess), guess) = 1.954$

Estimate Kfec from finding K1 (above)

$Kfec := 2$

Assign Kfec for use in later functions.

Now evaluate the probability of error (Viterbi coded only):

$EbNov := \mathrm{idB}(6.1)$

$\mathrm{pev}(EbNov) = 5.336 \cdot 10^{-9}$

$Rv := \frac{1}{2}$

Code rate

$\mathrm{peBPSKv}(en, \theta) := \mathrm{pev}\left(en \cdot \cos(\theta)^2\right)$

$$\mathrm{peSQPSKv}(en, \theta) := \mathrm{pev}\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right) + \frac{Kfec \cdot Rv}{2} \cdot \sin(\theta)^2}\right]$$

Extra 1/2 in front of Kfec corrects for 50%
of S-QPSK data having no crosstalk term.

$$\mathrm{peQPSKv}(en, \theta) := \mathrm{pev}\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right) + Kfec \cdot Rv \cdot \sin(\theta)^2}\right]$$

Now evaluate the probability of error (CCSDS coded):
Code rate, R=1/2 and 255,223 and sync word.

$EbNoc := \mathrm{idB}(2.7)$

$\mathrm{pec}(EbNoc) = 2.472 \cdot 10^{-8}$

$Rc := \frac{223}{512}$

$\mathrm{peBPSKc}(en, \theta) := \mathrm{pec}\left(en \cdot \cos(\theta)^2\right)$

$$\mathrm{peSQPSKc}(en, \theta) := \mathrm{pec}\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right) + \frac{Kfec \cdot Rc}{2} \cdot \sin(\theta)^2}\right]$$

$$\mathrm{peQPSKc}(en, \theta) := \mathrm{pec}\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right) + Kfec \cdot Rc \cdot \sin(\theta)^2}\right]$$

Run test to compare methods of finding Eb/No loss for static phase errors. Both loss at input to ideal system, and increase required in real system, just to compare for confidence.

$perr := 5 \cdot 10^{-9}$

$en := \mathrm{INVpec}(perr)$

$\theta := 2 \cdot \frac{\pi}{180}$

Test error

$\mathrm{dB}(en) = 2.752$

$err := \mathrm{peQPSKc}(en, \theta)$

$loss := \mathrm{dB}(en) - \mathrm{dB}(\mathrm{INVpec}(err))$

$loss = 0.014$

Loss in real compared to ideal case.

$\mathrm{dlQPSK}(\theta, en) := \log(perr) - \log(\mathrm{peQPSKc}(en, \theta))$

$dlQPSK(\theta, en\cdot 1.2) = 5.285$

$guess := 1.01\cdot en$

$reqEn\_QPSK(\theta) := root(dlQPSK(\theta, guess), guess)$

$loss := dB(reqEn\_QPSK(\theta)) - dB(en)$

$loss = 0.014$

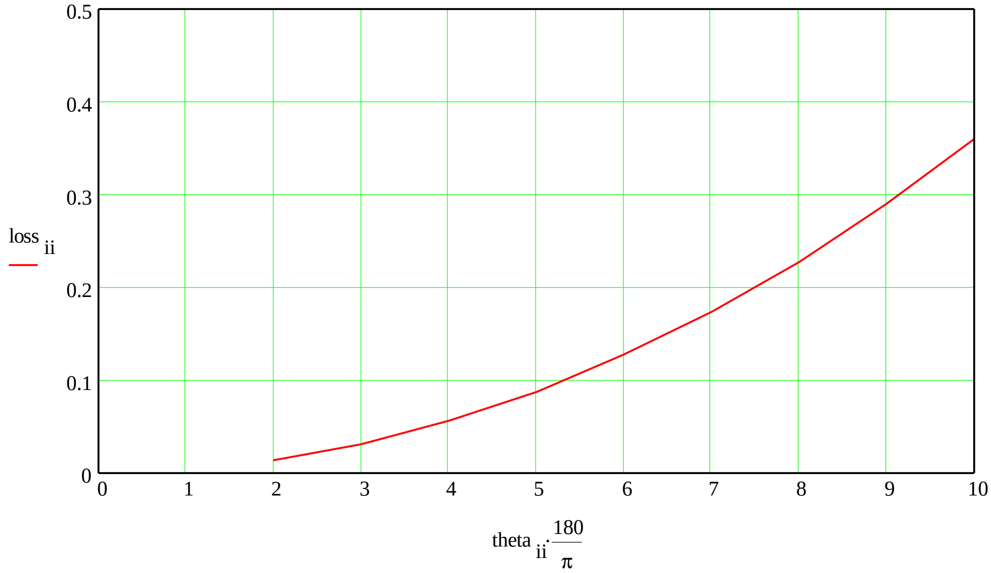
Extra Eb/No in real to match ideal case. Should be similar to "loss" shown 6 lines above.

Plot CCSDS QPSK as function of static phase error:

$ii := 2 .. 10$

$theta_{ii} := ii\cdot\frac{\pi}{180}$

$loss_{ii} := dB\left(reqEn\_QPSK\left(theta_{ii}\right)\right) - dB(en)$



Graph integrand to help with good numerical attack:

$\alpha := 2\cdot idB(23)$

$\sigma := \frac{1}{\sqrt{\alpha}}$

$Imax := 100$

$ii := 0 .. Imax$

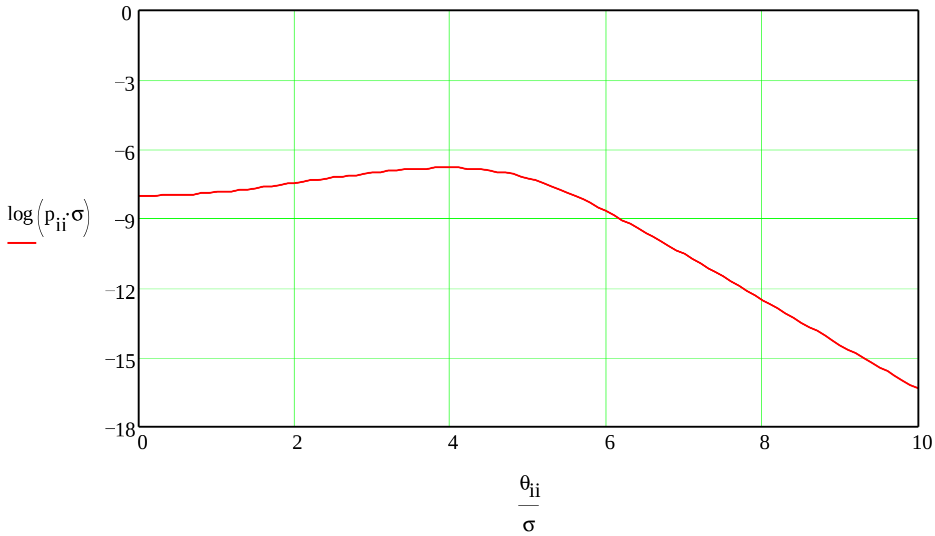
$N := 10$

$en := idB(2.7)$

$\theta_{ii} := \frac{N\cdot ii\cdot\sigma}{Imax}$

$\theta_{Imax} = 0.501$

$p_{ii} := peQPSKc\left(en, \theta_{ii}\right)\cdot Pb\left(\theta_{ii}, \alpha, 4\right)$



Perform the integration using the assumed symmetry of Pb() and use choice of region to avoid integration convergance accuracy problems.

$Sdev := 8$

Number of "standard deviations" to be integrated.

Create a general integration function which uses Ndev or π/M, whichever is smaller. The calling values are the Eb/No input, the α = (1/mean square error), the number of phase values (2=BPSK, 4=QPSK) and the Probabillity of Error Function specific to the system under consideration.

$$IntFunc(en,\alpha,M,Pef) := 2\cdot\int_0^{\text{if}\left[\alpha<\left(\frac{M\cdot Sdev}{\pi}\right)^2,\frac{\pi}{M},\frac{Sdev}{\sqrt{\alpha}}\right]} Pb(\theta,\alpha,M)\cdot Pef(en,\theta)\,d\theta$$

For the uncoded system:

$PerrBPSKu(en,\alpha) := IntFunc(en,\alpha,2,peBPSKu)$

$PerrQPSKu(en,\alpha) := IntFunc(en,\alpha,4,peQPSKu)$

For the Viterbi coded case:

$PerrBPSKv(en,\alpha) := IntFunc(en,\alpha,2,peBPSKv)$

$PerrQPSKv(en,\alpha) := IntFunc(en,\alpha,4,peQPSKv)$

For the CCSDS coded system:

$PerrBPSKc(en,\alpha) := IntFunc(en,\alpha,2,peBPSKc)$

$PerrQPSKc(en,\alpha) := IntFunc(en,\alpha,4,peQPSKc)$

And now compute for our range of loop SNR values:

$Npts := 30$

$kk := 0..Npts$

$Bl := 12$

$Bh := Bl + 10$

$Ql := 18$

$Qh := Ql + 10$

BPSK acquisition typically 8+6 = 14dB, but down to 12dB just practical.

$$alphaBPSK_{kk} := 2\cdot idB\left[\frac{kk\cdot(Bh-Bl)}{Npts}+Bl\right]$$

QPSK acquisition typically 8+12 = 20dB, but down to 18dB just practical.

$$alphaQPSK_{kk} := 2\cdot idB\left[\frac{kk\cdot(Qh-Ql)}{Npts}+Ql\right]$$

$perr := 5\cdot 10^{-3}$

$perr := 1 \cdot 10^{-6}$

$perr := 5 \cdot 10^{-9}$

= Error rate at which the sensitivity will be evaluated

Uncoded:

$en := INVpeu(perr)$

$dB(en) = 12.154$

$guess := en \cdot 1.01$

Solve for loss - use test to see if greater than 'chk' dB, if so ignore.

$chk := idB(1.5)$

$dluBPSK(\alpha, en) := \log(perr) - \log(PerrBPSKu(en, \alpha))$

$luBPSK(\alpha) := if(PerrBPSKu(en \cdot chk, \alpha) > perr, 100, root(dluBPSK(\alpha, guess), guess))$

$ulossBPSK_{kk} := dB\left(luBPSK\left(alphaBPSK_{kk}\right)\right) - dB(en)$

$dluQPSK(\alpha, en) := \log(perr) - \log(PerrQPSKu(en, \alpha))$

$luQPSK(\alpha) := if(PerrQPSKu(en \cdot chk, \alpha) > perr, 100, root(dluQPSK(\alpha, guess), guess))$

$ulossQPSK_{kk} := dB\left(luQPSK\left(alphaQPSK_{kk}\right)\right) - dB(en)$

Viterbi coded case:

$en := INVpev(perr)$

$dB(en) = 6.11$

$guess := en \cdot 1.01$

$dlvBPSK(\alpha, en) := \log(perr) - \log(PerrBPSKv(en, \alpha))$

$lvBPSK(\alpha) := if(PerrBPSKv(en \cdot chk, \alpha) > perr, 100, root(dlvBPSK(\alpha, guess), guess))$

$vlossBPSK_{kk} := dB\left(lvBPSK\left(alphaBPSK_{kk}\right)\right) - dB(en)$

$dlvQPSK(\alpha, en) := \log(perr) - \log(PerrQPSKv(en, \alpha))$

$lvQPSK(\alpha) := if(PerrQPSKv(en \cdot chk, \alpha) > perr, 100, root(dlvQPSK(\alpha, guess), guess))$

$vlossQPSK_{kk} := dB\left(lvQPSK\left(alphaQPSK_{kk}\right)\right) - dB(en)$

CCSDS coded case:

$en := INVpec(perr)$

$dB(en) = 2.752$

$guess := en \cdot 1.01$

$dlcBPSK(\alpha, en) := \log(perr) - \log(PerrBPSKc(en, \alpha))$

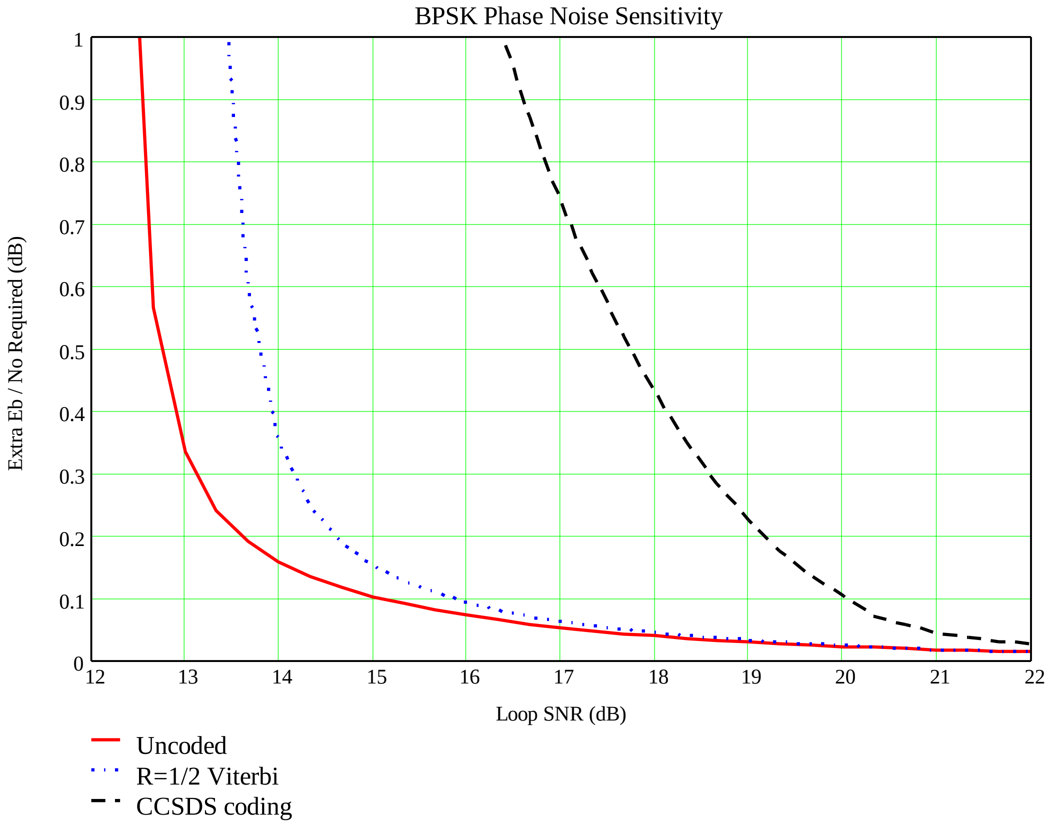
$lcBPSK(\alpha) := if(PerrBPSKc(en \cdot chk, \alpha) > perr, 100, root(dlcBPSK(\alpha, guess), guess))$

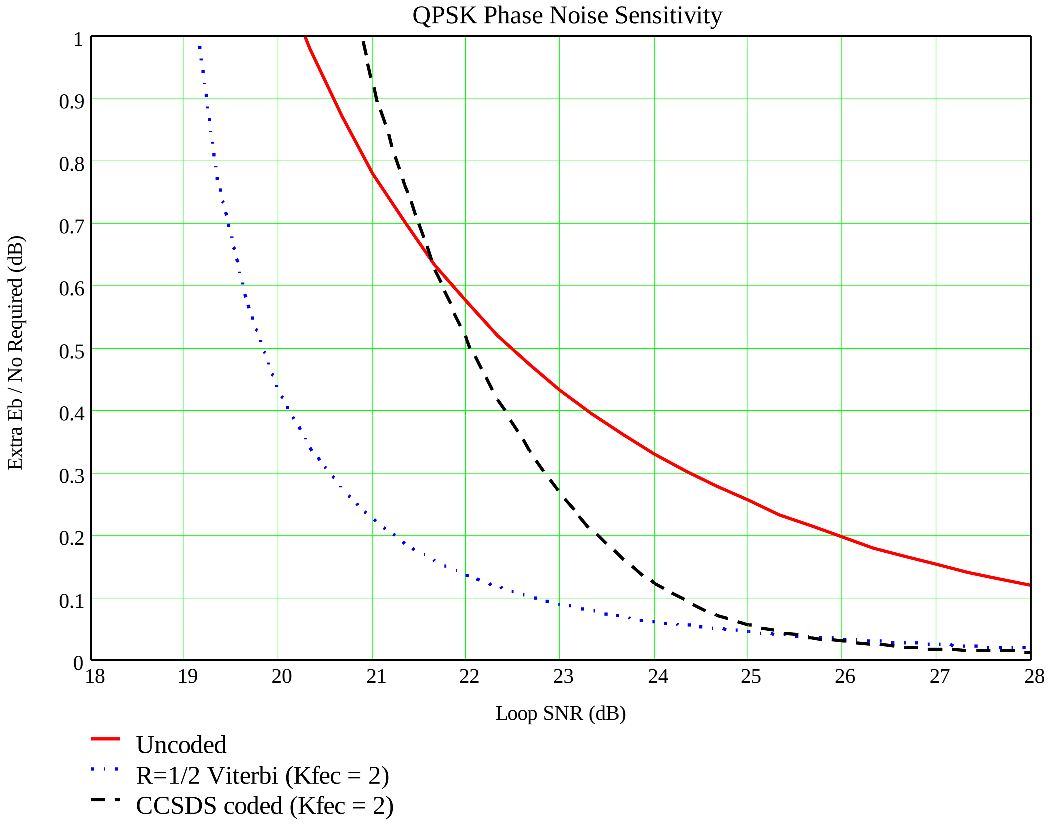
$clossBPSK_{kk} := dB\left(lcBPSK\left(alphaBPSK_{kk}\right)\right) - dB(en)$

$dlcQPSK(\alpha, en) := \log(perr) - \log(PerrQPSKc(en, \alpha))$

$lcQPSK(\alpha) := if(PerrQPSKc(en \cdot chk, \alpha) > perr, 100, root(dlcQPSK(\alpha, guess), guess))$

$clossQPSK_{kk} := dB\left(lcQPSK\left(alphaQPSK_{kk}\right)\right) - dB(en)$


BPSK Phase Noise Sensitivity
Extra Eb / No Required (dB)
Loop SNR (dB)
Uncoded
R=1/2 Viterbi
CCSDS coding


QPSK Phase Noise Sensitivity
Extra Eb / No Required (dB)
Loop SNR (dB)
Uncoded
R=1/2 Viterbi (Kfec = 2)
CCSDS coded (Kfec = 2)

-- End of file --

File "QPSK_EDA.MCD"
Created with MathCAD 6.0-PLUS
Computation of carrier tracking loop for LRIT using BPSK, this with revised carrier PDF and more accurate CCSDS error rate formulae. This noe to plot as function of Es/No

$$j := \sqrt{-1}$$

$$dB(x) := \mathrm{if}\left(|x| < 10^{-12}, -120, 10\cdot\log(|x|)\right)$$

Convert to dB
And back to numeric

$$idB(x) := 10^{\frac{x}{10}}$$

$$BR := 1\cdot10^{6}$$

Bit rate
Now for the transfer function of the carrier tracking PLL, 2nd order system, radian angular frequency.

$$HS2(s,\omega n,\zeta,T) := \frac{\left(2\cdot\zeta\cdot\omega n\cdot s + \omega n^{2}\right)\cdot\exp(-s\cdot T)}{s^{2} + \left(2\cdot\zeta\cdot\omega n\cdot s + \omega n^{2}\right)\cdot\exp(-s\cdot T)}$$

From Laplace 's' variable to Hz version:

$$H(f,fn,\zeta,T) := HS2(j\cdot2\cdot\pi\cdot f, 2\cdot\pi\cdot fn,\zeta,T)$$

The tracking error function (for power, i.e. angle^2):

$$TE(f,fn,\zeta,T) := \left(|1 - H(f,fn,\zeta,T)|\right)^{2}$$

Model the phase noise as flicker (f^3) and normal (f^2) over the frequency range of interest:

$$L0 := idB(-88 + 60)$$

Phase noise density in dBc/Hz at 1Hz offset, 60dB above 1kHz value,etc).

$$f\_FL := 50$$

Flicker noise corner frequency, Hz.

$$L(f,L0,f\_FL) := \frac{L0}{f^{2}}\cdot\left(1 + \frac{f\_FL}{f}\right)$$

Use the fact that CNo is approximatly constant over PLL range. Implement the fn to BR integration with a change of variable to make the points for evaluation spread in a 1/X manner to match the function shape.

$$BLoop(fn,\zeta,T) := \int_{0}^{fn}\left(|H(f,fn,\zeta,T)|\right)^{2}\,df \;...\; + \int_{\frac{1}{BR}}^{\frac{1}{fn}}\frac{1}{f^{2}}\cdot\left(\left|H\left(\frac{1}{f},fn,\zeta,T\right)\right|\right)^{2}\,df$$

And the phase noise contribution:

$$MS2(fn,\zeta,L0,f\_FL,T) := 2\cdot\left[\int_{10^{-8}}^{fn} L(f,L0,f\_FL)\cdot TE(f,fn,\zeta,T)\,df \;...\; + \int_{\frac{1}{BR}}^{\frac{1}{fn}}\frac{1}{f^{2}}\cdot L\left(\frac{1}{f},L0,f\_FL\right)\cdot TE\left(\frac{1}{f},fn,\zeta,T\right)\,df\right]$$

Add the discrete terms from the electronically despun array. The are 10 degrees peak to peak at the commutation rate (53.3Hz) and 3 degrees peak to peak at the spin rate of 100rpm (1.67Hz). These are approximate values measured at Dundee using the default MDD tracking loop (15.7Hz and d=1.0), the corrected values as power (i.e. radian^2) are:

$$K\_DIRECT\_PLL(deg\_pp, f) := \left(\frac{deg\_pp}{2 \cdot \sqrt{2}} \cdot \frac{\pi}{180}\right)^2$$

$$K\_MDD\_PLL(deg\_pp, f) := \frac{K\_DIRECT\_PLL(deg\_pp, f)}{TE(f, 15.7, 1, 0)}$$

$Kcr := K\_MDD\_PLL(10, 53.3)$

$Ksp := K\_MDD\_PLL(3, 1.67)$

EUMETSAT obtained 20 deg P-P as max from S/C specifications. Use this.

$Kcr := K\_DIRECT\_PLL(20, 53.3)$

$Kcr = 0.015$

$Ksp = 2.738$

$MS3(fn, \zeta, Kcr, Ksp, T) := Kcr \cdot TE(53.3, fn, \zeta, T) + Ksp \cdot TE(1.67, fn, \zeta, T)$

Compute some example values for the system (before graphs for optimising the parameters).

$\zeta := 1.14$

Damping factor (dimensionless)

$T := 10^{-4}$

PLL loop filter time delay (seconds).

$$fn := \begin{pmatrix} 64 \\ 128 \\ 256 \end{pmatrix}$$

Natural frequency (Hz)

$imax := rows(fn)$

$ii := 0 .. imax - 1$

Compute the "static" parameters for varying Eb/No

$BL_{ii} := BLoop(fn_{ii}, \zeta, T)$

PLL Noise bandwidth

$mm2_{ii} := MS2(fn_{ii}, \zeta, L0, f\_FL, T)$

Phase Noise term

$mm3_{ii} := MS3(fn_{ii}, \zeta, Kcr, Ksp, T)$

EDA phase effects.

Read the file with pre-calculated loss terms for the BPSK system. This allows us to calculate the effective C/No available to the PLL

$$Rc := \frac{223}{512}$$

Code rate.

$psk\_loss\_lut := READPRN(qpsk\_loss\_lut)$

$Nrows := rows(psk\_loss\_lut)$

$Nrows = 32$

$il := 0 .. Nrows - 1$

$idb\_EsNo_{il} := idB(psk\_loss\_lut_{il, 0})$

$db\_loss_{il} := psk\_loss\_lut_{il, 3}$

Index choice is 1=DD, 2=ML, 3=V&V, 4=NthPower

$loss\_vs := cspline(idb\_EsNo, db\_loss)$

$loss\_ifunc(v) := -interp(loss\_vs, idb\_EsNo, db\_loss, v)$

$psk\_loss(v) := if(v < idb\_EsNo_0, 100, if(v > idb\_EsNo_{Nrows-1}, 0, loss\_ifunc(v)))$

$$\text{EffCNoFunc}(\text{ebno}) := \frac{\text{ebno} \cdot \text{BR}}{\text{idB}(\text{psk\_loss}(\text{ebno} \cdot \text{Rc}))}$$

$$\text{ms\_error}(\text{ebno}, i) := \frac{BL_i}{\text{EffCNoFunc}(\text{ebno})} + \text{mm2}_i + \text{mm3}_i$$

Set up the input Eb/No range. This excludes the 0.8dB implementation margin, i.e. this is the real demodulator input so the MSG should have 2.8+0.8 = 3.6dB normally.

$\text{kmax} := 39$

$\text{kk} := 0 .. \text{kmax}$

$\text{Emin} := -3$

$\text{Emax} := 6$

$$\text{EbNo}_{kk} := \text{idB}\left[(\text{Emax} - \text{Emin}) \cdot \frac{kk}{\text{kmax}} + \text{Emin}\right]$$

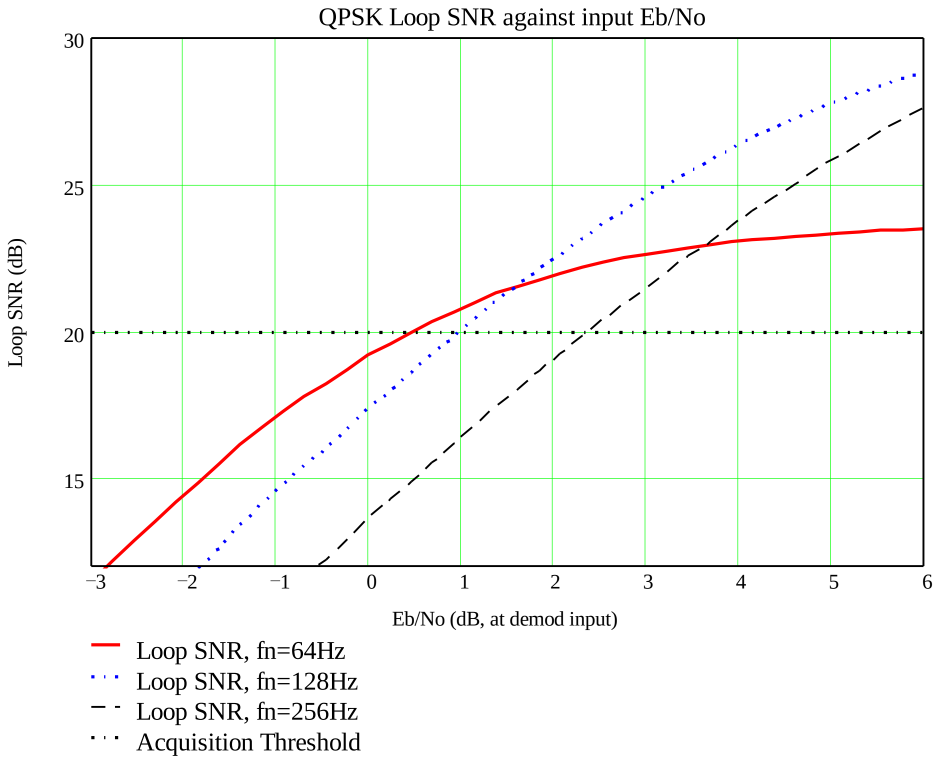
Compute the loop SNR (dB, in F.M. Gandner's notation).

$$\text{snrL}_{kk, ii} := \text{dB}\left(\frac{1}{2 \cdot \text{ms\_error}\left(\text{EbNo}_{kk}, ii\right)}\right)$$

$M := 4$

Acquisition threshold, 8dB (fastest sweep
and M^2 for M'th order recovery loop).

$$\text{snrACQ}_{kk} := 8 + \text{dB}\left(M^2\right)$$



$$\text{TAve}(bl, snr) := \frac{2}{bl} \cdot \exp\left(\pi \cdot \frac{snr}{M^2}\right)$$

Approximate mean time to slip in seconds.

$$\text{ta\_func}(\text{ebno}, i) := \text{TAve}\left(BL_i, \frac{1}{2 \cdot \text{ms\_error}(\text{ebno}, i)}\right) \cdot \frac{1}{3600}$$

In hours

$ta_{kk,ii} := ta\_func(EbNo_{kk}, ii)$

$fail\_EbNo := idB(2.8 + 0.8 - 3)$

EDA failures spec -3dB

$ta\_func(fail\_EbNo, ii)$

$1.612 \cdot 10^{3}$
5.732
$2.792 \cdot 10^{-4}$

Hours, but allow for 1/32 EDA time.

$$dB\left(\frac{1}{2 \cdot ms\_error(fail\_EbNo, ii)}\right)$$

20.206
19.039
15.338

$fail\_EbNo := idB(2.8 + 0.8 - 6)$

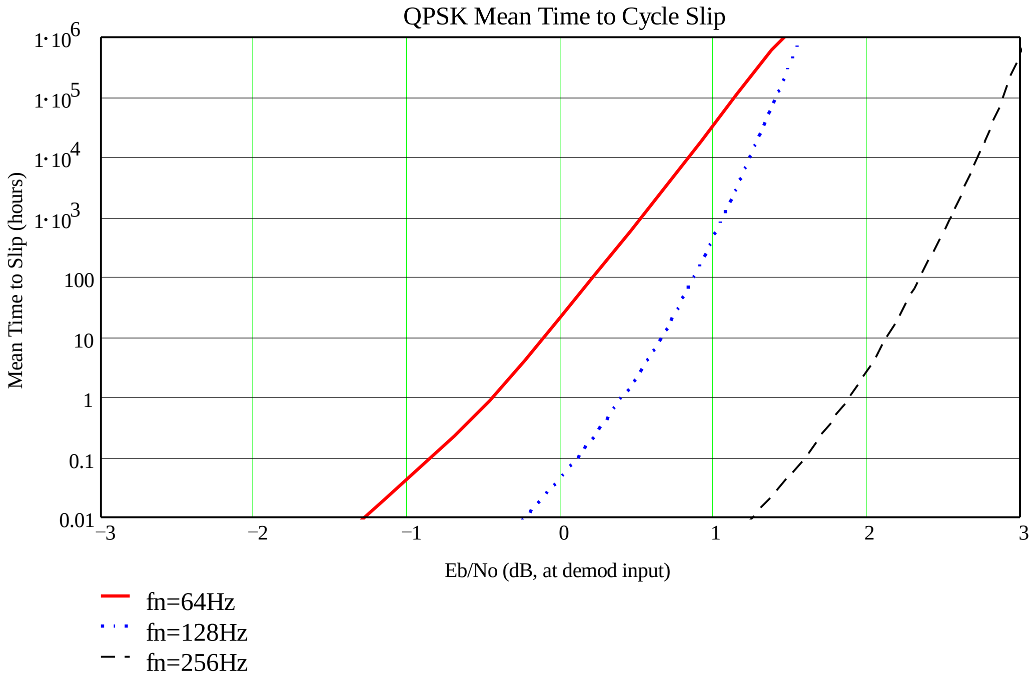
What of -6dB?

$ta\_func(fail\_EbNo, ii)$

$1.205 \cdot 10^{-4}$
$6.598 \cdot 10^{-6}$
$7.897 \cdot 10^{-7}$

$$dB\left(\frac{1}{2 \cdot ms\_error(fail\_EbNo, ii)}\right)$$

13.279
10.217
6.328



--End of File--

File "QPSKDET.MCD"
Created using MathCAD 6.0-PLUS
Computation of QPSK phase estimator performance.
This is VERY slow, if you want to run this file, plan a long coffee break (typically 3 minutes CPU time on 200MHz PentiumPro PC!   Do not set TOL to much less then 1E-3 for this file.

$$dB(x) := \mathrm{if}\left(|x| < 10^{-12}, -120, 10 \cdot \log(|x|)\right)$$

Convert to dB
And back to numeric

$$idB(x) := 10^{\frac{x}{10}}$$

$$efunc0(I,Q,\sigma) := I \cdot \mathrm{if}(Q<0,-1,1) - Q \cdot \mathrm{if}(I<0,-1,1)$$

Data-directed (hard limited)

$$efunc1(I,Q,\sigma) := I \cdot \tanh\left(\frac{Q}{\sigma^2 \cdot \sqrt{2}}\right) - Q \cdot \tanh\left(\frac{I}{\sigma^2 \cdot \sqrt{2}}\right)$$

ML function

$$efunc2(I,Q,\sigma) := 4 \cdot I \cdot Q \cdot \frac{I^2 - Q^2}{I^2 + Q^2}$$

m^2(t)sin(4θ)
m^4(t)sin(4θ) (= 4th power loop)

$$efunc3(I,Q,\sigma) := 4 \cdot I \cdot Q \cdot \left(I^2 - Q^2\right)$$

Define the 2D signal + noise PDF

$$pr(x,xx,y,yy,\sigma) := \frac{1}{2 \cdot \pi \cdot \sigma^2} \cdot \exp\left[-\frac{(x-xx)^2 + (y-yy)^2}{2 \cdot \sigma^2}\right]$$

$$Nsig := 5$$

$$eps := 10^{-8}$$

Compute the hard decision with 4 regions to help the integration routine.

$$a0(xx,yy,\sigma) := \int_{eps}^{yy+Nsig \cdot \sigma} \int_{eps}^{xx+Nsig \cdot \sigma} pr(x,xx,y,yy,\sigma) \cdot efunc0(x,y,\sigma)\, dx\, dy \;...$$
$$+ \int_{yy-Nsig \cdot \sigma}^{-eps} \int_{eps}^{xx+Nsig \cdot \sigma} pr(x,xx,y,yy,\sigma) \cdot efunc0(x,y,\sigma)\, dx\, dy \;..$$
$$+ \int_{eps}^{yy+Nsig \cdot \sigma} \int_{xx-Nsig \cdot \sigma}^{-eps} pr(x,xx,y,yy,\sigma) \cdot efunc0(x,y,\sigma)\, dx\, dy \;..$$
$$+ \int_{yy-Nsig \cdot \sigma}^{-eps} \int_{xx-Nsig \cdot \sigma}^{-eps} pr(x,xx,y,yy,\sigma) \cdot efunc0(x,y,\sigma)\, dx\, dy$$

$$a1(xx,yy,\sigma) := \int_{yy-Nsig \cdot \sigma}^{yy+Nsig \cdot \sigma} \int_{xx-Nsig \cdot \sigma}^{xx+Nsig \cdot \sigma} pr(x,xx,y,yy,\sigma) \cdot efunc1(x,y,\sigma)\, dx\, dy$$

$$a2(xx,yy,\sigma) := \int_{yy-Nsig \cdot \sigma}^{yy+Nsig \cdot \sigma} \int_{xx-Nsig \cdot \sigma}^{xx+Nsig \cdot \sigma} pr(x,xx,y,yy,\sigma) \cdot efunc2(x,y,\sigma)\, dx\, dy$$

$$a3(xx,yy,\sigma) := \int_{yy-Nsig \cdot \sigma}^{yy+Nsig \cdot \sigma} \int_{xx-Nsig \cdot \sigma}^{xx+Nsig \cdot \sigma} pr(x,xx,y,yy,\sigma) \cdot efunc3(x,y,\sigma)\, dx\, dy$$

And a set of P log-spaced N/S points from "lower" to "upper".

$M := 4$

$nn := \frac{\log(M)}{\log(2)}$

$theta := \frac{2}{M} \cdot \frac{\pi}{180}$

$P := 31$

$V0 := 1$

$lower := idB(-10)$

$upper := idB(8)$

$x1 := V0 \cdot \sin\left(\frac{\pi}{M} + theta\right)$

$x2 := V0 \cdot \sin\left(\frac{\pi}{M} - theta\right)$

$kk := 0 .. P$

$a := lower$

$y2 := V0 \cdot \cos\left(\frac{\pi}{M} - theta\right)$

$y1 := V0 \cdot \cos\left(\frac{\pi}{M} + theta\right)$

$x1 = 0.7$

$y1 = 0.7$

$b := \exp\left(\frac{\ln\left(\frac{upper}{a}\right)}{P}\right)$

$x2 = 0.7$

$y2 = 0.7$

$efunc0(x1, y1, 1) = 0.0$

$efunc1(x1, y1, 1) = 8.5$

$en_{kk} := a \cdot b^{kk}$

Es/No values

$efunc2(x1, y1, 1) = 0.0$

$efunc3(x1, y1, 1) = 0.0$

$sigma_{kk} := \frac{V0}{\sqrt{2 \cdot en_{kk} \cdot nn}}$

Voltage standard deviation

Compute the detector shift for +/- theta

$avd0_{kk} := a0(x1, y1, sigma_{kk}) - a0(x2, y2, sigma_{kk})$

$avd1_{kk} := a1(x1, y1, sigma_{kk}) - a1(x2, y2, sigma_{kk})$

$avd2_{kk} := a2(x1, y1, sigma_{kk}) - a2(x2, y2, sigma_{kk})$

$avd3_{kk} := a3(x1, y1, sigma_{kk}) - a3(x2, y2, sigma_{kk})$

Compute the normalised phase sensitive detector values (dV/dθ)

$Kpsd0_{kk} := \frac{avd0_{kk}}{2 \cdot theta}$

$Kpsd1_{kk} := \frac{avd1_{kk}}{2 \cdot theta}$

$$Kpsd2_{kk} := \frac{avd2_{kk}}{2 \cdot theta}$$

$$Kpsd3_{kk} := \frac{avd3_{kk}}{2 \cdot theta}$$

Compute the reduction in Kpsd due to noise.

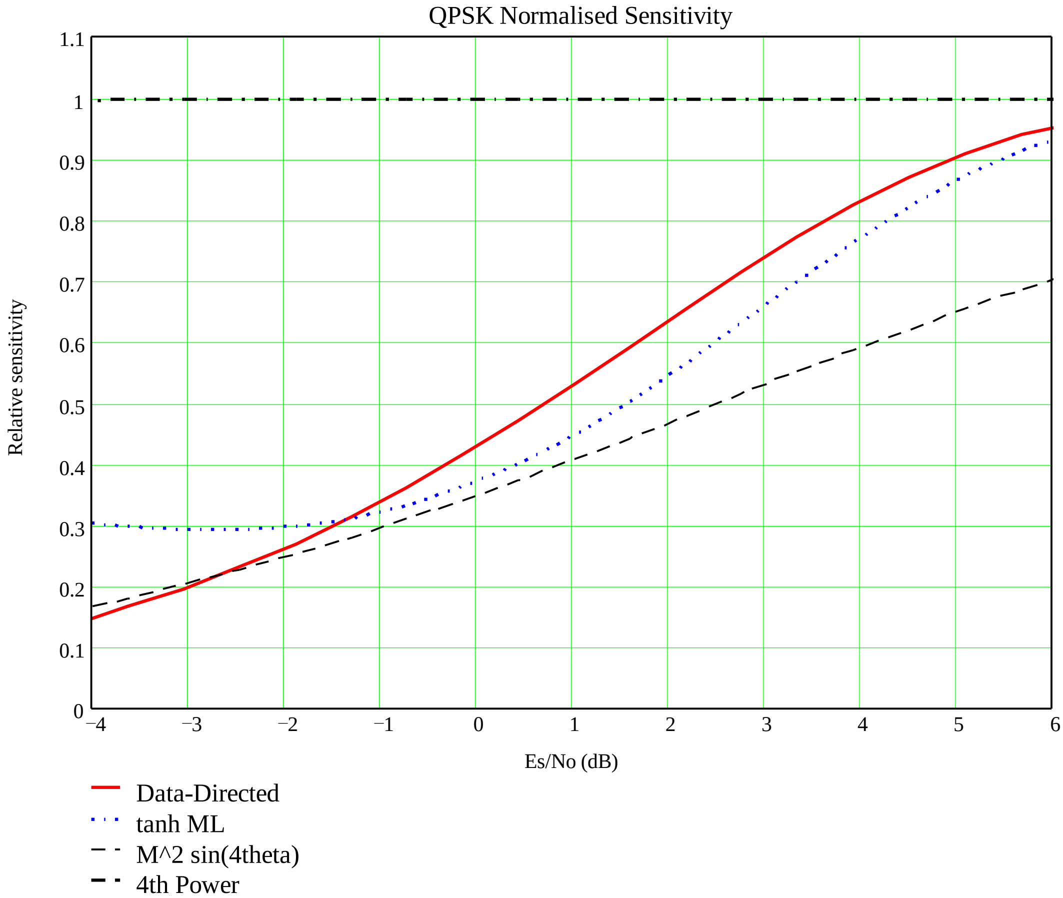
$$av0_{kk} := \frac{avd0_{kk}}{efunc0\left(x1, y1, sigma_{kk}\right) - efunc0\left(x2, y2, sigma_{kk}\right)}$$

$$av1_{kk} := \frac{avd1_{kk}}{efunc1\left(x1, y1, sigma_{kk}\right) - efunc1\left(x2, y2, sigma_{kk}\right)}$$

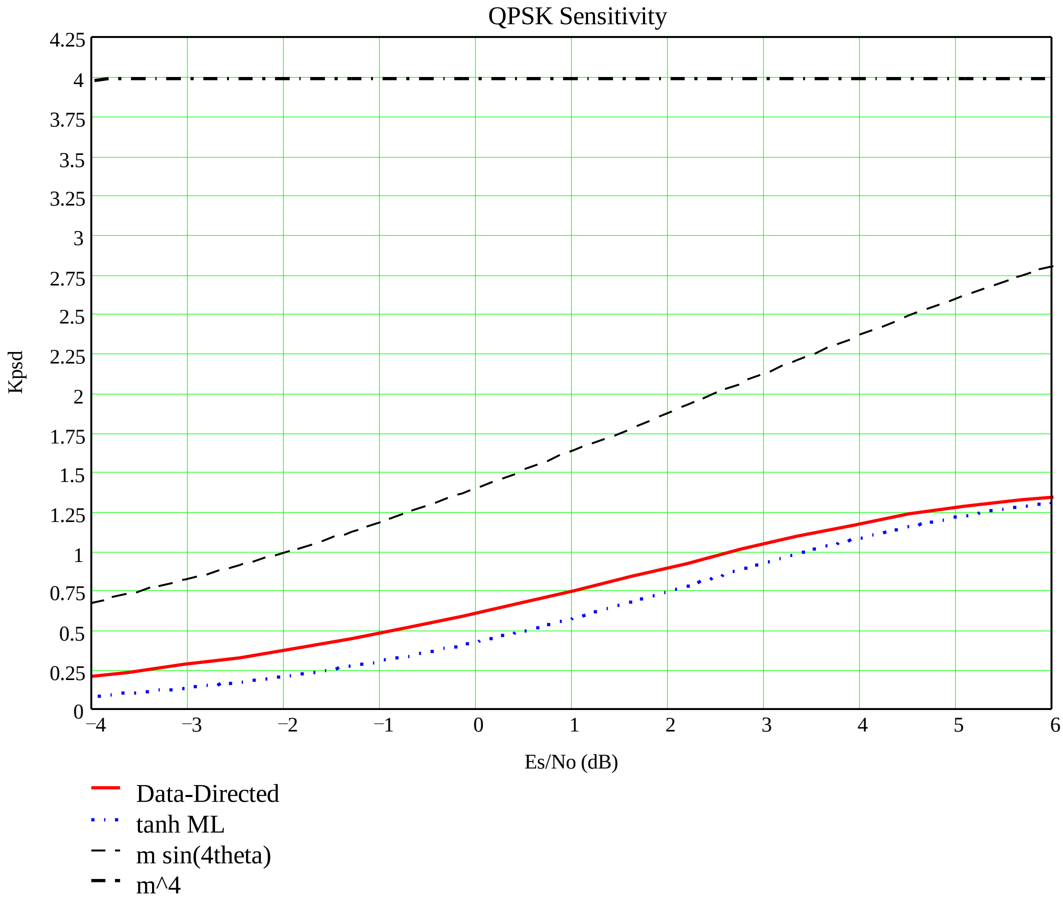
$$av2_{kk} := \frac{avd2_{kk}}{efunc2\left(x1, y1, sigma_{kk}\right) - efunc2\left(x2, y2, sigma_{kk}\right)}$$

$$av3_{kk} := \frac{avd3_{kk}}{efunc3\left(x1, y1, sigma_{kk}\right) - efunc3\left(x2, y2, sigma_{kk}\right)}$$

Plot out normalised detector sensitivity



Plot out KPSD for these



Set up for zero expected value then compute the mean square error.

$$theta := 0.0$$

$$x1 := V0\cdot\sin\left(\frac{\pi}{M} + theta\right)$$

$$y1 := V0\cdot\cos\left(\frac{\pi}{M} + theta\right)$$

$$\begin{aligned} sd0(xx, yy, \sigma) := & \int_{eps}^{yy + Nsig\cdot\sigma}\int_{eps}^{xx + Nsig\cdot\sigma} pr(x, xx, y, yy, \sigma)\cdot efunc0(x, y, \sigma)^2\,dx\,dy \ldots \\ & + \int_{yy - Nsig\cdot\sigma}^{-eps}\int_{eps}^{xx + Nsig\cdot\sigma} pr(x, xx, y, yy, \sigma)\cdot efunc0(x, y, \sigma)^2\,dx\,dy \,.. \\ & + \int_{eps}^{yy + Nsig\cdot\sigma}\int_{xx - Nsig\cdot\sigma}^{-eps} pr(x, xx, y, yy, \sigma)\cdot efunc0(x, y, \sigma)^2\,dx\,dy \,.. \\ & + \int_{yy - Nsig\cdot\sigma}^{-eps}\int_{xx - Nsig\cdot\sigma}^{-eps} pr(x, xx, y, yy, \sigma)\cdot efunc0(x, y, \sigma)^2\,dx\,dy \end{aligned}$$

$$sd1(xx, yy, \sigma) := \int_{yy - Nsig\cdot\sigma}^{yy + Nsig\cdot\sigma}\int_{xx - Nsig\cdot\sigma}^{xx + Nsig\cdot\sigma} pr(x, xx, y, yy, \sigma)\cdot efunc1(x, y, \sigma)^2\,dx\,dy$$

$$sd2(xx,yy,\sigma) := \int_{yy - Nsig\cdot\sigma}^{yy + Nsig\cdot\sigma} \int_{xx - Nsig\cdot\sigma}^{xx + Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc2(x,y,\sigma)^2 \, dx \, dy$$

$$sd3(xx,yy,\sigma) := \int_{yy - Nsig\cdot\sigma}^{yy + Nsig\cdot\sigma} \int_{xx - Nsig\cdot\sigma}^{xx + Nsig\cdot\sigma} pr(x,xx,y,yy,\sigma)\cdot efunc3(x,y,\sigma)^2 \, dx \, dy$$

Compute vectors of mean squared values.

$$sv0_{kk} := sd0\left(x1, y1, sigma_{kk}\right)$$

$$sv1_{kk} := sd1\left(x1, y1, sigma_{kk}\right)$$

$$sv2_{kk} := sd2\left(x1, y1, sigma_{kk}\right)$$

$$sv3_{kk} := sd3\left(x1, y1, sigma_{kk}\right)$$

$$LossF(p) := \frac{1}{1 + \frac{9}{p} + \frac{6}{p^2} + \frac{1.5}{p^3}}$$

Traditional 4th power loss function.

$$rsv0_{kk} := dB\left[\frac{\left(Kpsd0_{kk}\right)^2}{sv0_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$

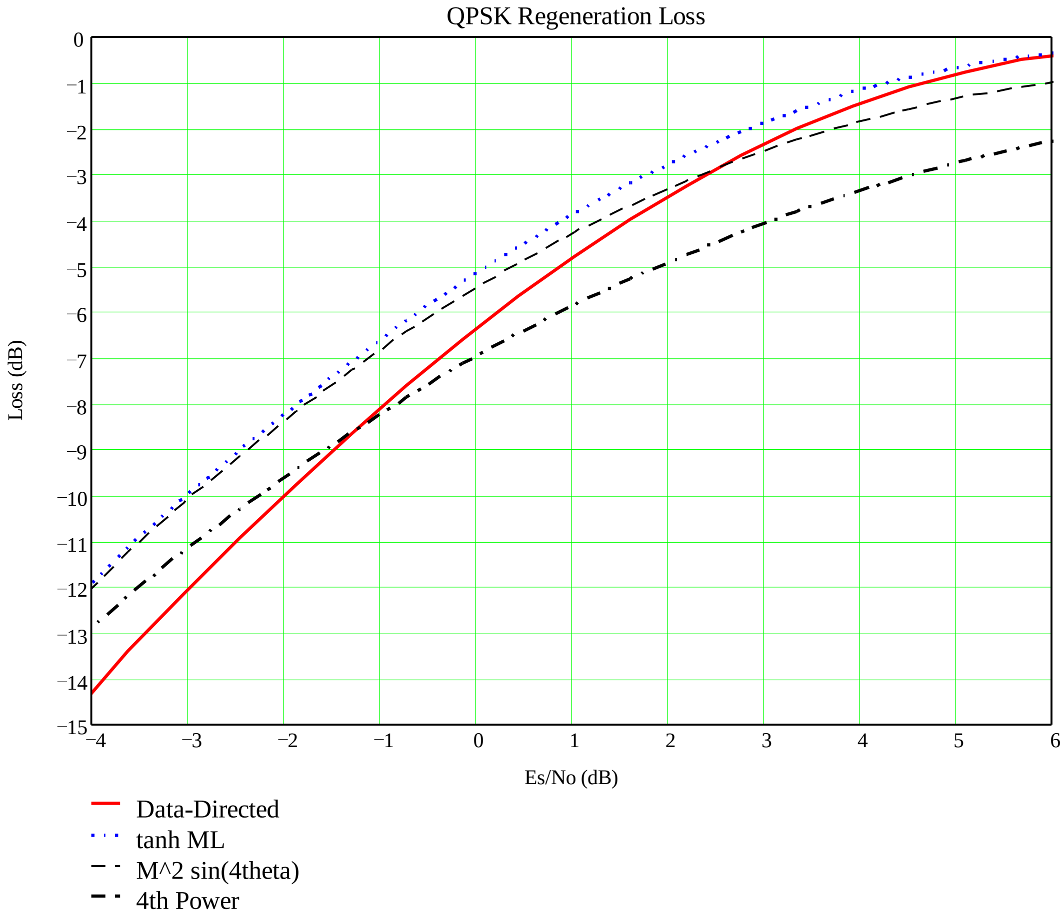
$$rsv1_{kk} := dB\left[\frac{\left(Kpsd1_{kk}\right)^2}{sv1_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$

$$rsv2_{kk} := dB\left[\frac{\left(Kpsd2_{kk}\right)^2}{sv2_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$

$$rsv3_{kk} := dB\left[\frac{\left(Kpsd3_{kk}\right)^2}{sv3_{kk}\cdot en_{kk}\cdot 2\cdot nn}\right]$$

$$rsv4_{kk} := dB\left(LossF\left(en_{kk}\cdot 2\right)\right)$$

Plot out the standard deviation relative to the detector sensitivity



$dben_{kk} := dB\left(en_{kk}\right)$

Save results for later. We have the following data:

dB(Es/No) data-directed ML M^2sin(4θ) 4th power

$save_{kk,0} := dben_{kk}$

$save_{kk,1} := rsv0_{kk}$

$save_{kk,2} := rsv1_{kk}$

$save_{kk,3} := rsv2_{kk}$

$save_{kk,4} := rsv3_{kk}$

PRNPRECISION := 8

PRNCOLWIDTH := 15

WRITEPRN(qpsk_loss_lut) := save

-- End of file --

File "QPSKO5.MCD"
Created with MathCAD 6.0-PLUS
Computation of carrier tracking loop for HRIT using QPSK, this version with 'per symbol' Kfec term.

$dB(x) := if\left(|x| < 10^{-12}, -120, 10 \cdot \log(|x|)\right)$

Convert to dB
And back to numeric

$idB(x) := 10^{\frac{x}{10}}$

Convert radian^2 to degrees (RMS).

$dg(x) := \frac{180}{\pi} \cdot \sqrt{x}$

Bit rate

$BR := 10^6$

$SR := BR \cdot \frac{512}{223}$

Symbol rate with coding

$EbNo := idB(2.8 + 0.8)$

Set up Eb/No and include tech loss margin.

$EbNo = 2.291$

(numeric)

$CNo := BR \cdot EbNo$

Carrier/Noise density ratio.

$dB(CNo) = 63.6$

$EsNo := EbNo \cdot \frac{223}{512}$

The CCSDS coding rate gives the energy per SYMBOL.
Now for the transfer function of the carrier tracking PLL.

$j := \sqrt{-1}$

2nd order system, radian angular frequency.

$$HS2(s, \omega n, \zeta, T) := \frac{\left(2 \cdot \zeta \cdot \omega n \cdot s + \omega n^2\right) \cdot \exp(-s \cdot T)}{s^2 + \left(2 \cdot \zeta \cdot \omega n \cdot s + \omega n^2\right) \cdot \exp(-s \cdot T)}$$

From Laplace 's' variable to Hz version:

$H(f, fn, \zeta, T) := HS2(j \cdot 2 \cdot \pi \cdot f, 2 \cdot \pi \cdot fn, \zeta, T)$

The tracking error function (for power, i.e. angle^2):

$TE(f, fn, \zeta, T) := (|1 - H(f, fn, \zeta, T)|)^2$

Model the phase noise as flicker (f^3) and normal (f^2) over the frequency range of interest:

$L0 := idB(-88 + 60)$

Phase noise density in dBc/Hz at 1Hz offset, 60dB above 1kHz value,etc).

$f\_FL := 50$

Flicker noise corner frequency, Hz.

$$L(f, L0, f\_FL) := \frac{L0}{f^2} \cdot \left(1 + \frac{f\_FL}{f}\right)$$

Some sample output values:

$dB(L(10, L0, f\_FL)) = -40.218$

$dB\left(L\left(10^3, L0, f\_FL\right)\right) = -87.788$

Compute integrated phase noise. This uses a change of variable so the 1/x evaluation pattern suits the function shape.

$$Ipn := 2\cdot\int_{\frac{1}{10^6}}^{\frac{1}{10}} \frac{1}{f^2}\cdot L\left(\frac{1}{f}, L0, f\_FL\right) df$$

$dg(Ipn) = 1.908$

MSG spec calls for < 2deg RMS phase noise from 10Hz to 1MHz

Read the file with pre-calculated loss terms for the QPSK system. This allows us to calculate the effective C/No available to the PLL

$psk\_loss\_lut := READPRN(qpsk\_loss\_lut)$

$Nrows := rows(psk\_loss\_lut)$

$Nrows = 32$

$il := 0 .. Nrows - 1$

$idb\_EsNo_{il} := idB\left(psk\_loss\_lut_{il,0}\right)$

$TypeIdx := 3$

Index choice is 1=DD, 2=ML, 3=V&V, 4=NthPower

$db\_loss_{il} := psk\_loss\_lut_{il,TypeIdx}$

$loss\_vs := cspline(idb\_EsNo, db\_loss)$

$loss\_ifunc(v) := -interp(loss\_vs, idb\_EsNo, db\_loss, v)$

$psk\_loss(v) := if\left(v<idb\_EsNo_0, 100, if\left(v>idb\_EsNo_{Nrows-1}, 0, loss\_ifunc(v)\right)\right)$

$psk\_loss(idB(2.7 - 3.6 + 0.8 - 0)) = 5.567$

$AcqLoss := 0$

Set to zero for tracking, or 3dB for acquisition to account for the non-optimum samples or to simulate EDA failure.

$$ImLoss := \frac{1}{idB(psk\_loss(EsNo) + AcqLoss)}$$

$dB(ImLoss) = -5.452$

The "4th power" loss term.

$dB(CNo\cdot ImLoss) = 58.148$

Effective carrier/noise density in loop.

Use the fact that CNo is approximatly constant over PLL range. Implement the fn to BR integration with a change of variable to make the points for evaluation spread in a 1/X manner to match the function shape.

$$MS1(fn,\zeta,T) := \frac{1}{CNo\cdot ImLoss}\cdot\left[\int_0^{fn} \left(\left|H(f,fn,\zeta,T)\right|\right)^2 df \ldots + \int_{\frac{1}{BR}}^{\frac{1}{fn}} \frac{1}{f^2}\cdot\left(\left|H\left(\frac{1}{f},fn,\zeta,T\right)\right|\right)^2 df\right]$$

And the phase noise contribution:

$$MS2(fn,\zeta,L0,f\_FL,T) := 2\cdot\left[\int_{10^{-8}}^{fn} L(f,L0,f\_FL)\cdot TE(f,fn,\zeta,T)\,df \ldots + \int_{\frac{1}{BR}}^{\frac{1}{fn}} \frac{1}{f^2}\cdot L\left(\frac{1}{f},L0,f\_FL\right)\cdot TE\left(\frac{1}{f},fn,\zeta,T\right)\,df\right]$$

Add the discrete terms from the electronically despun array. The are 10 degrees peak to peak at the commutation rate (53.3Hz) and 3 degrees peak to peak at the spin rate of 100rpm (1.67Hz). These are approximate values measured at Dundee using the default MDD tracking loop (15.7Hz and d=1.0), the corrected values as power (i.e. radian^2) are:

$$K\_DIRECT\_PLL(deg\_pp,f) := \left(\frac{deg\_pp}{2\cdot\sqrt{2}}\cdot\frac{\pi}{180}\right)^2$$

$$K\_MDD\_PLL(deg\_pp,f) := \frac{K\_DIRECT\_PLL(deg\_pp,f)}{TE(f,15.7,1,0)}$$

$Kcr := K\_MDD\_PLL(10,53.3)$

$Ksp := K\_MDD\_PLL(3,1.67)$

EUMETSAT obtained 20 deg P-P as max from S/C doccument.

$Kcr := K\_DIRECT\_PLL(20,53.3)$

$Kcr = 0.015$

$Ksp = 2.738$

$MS3(fn,\zeta,Kcr,Ksp,T) := Kcr\cdot TE(53.3,fn,\zeta,T) + Ksp\cdot TE(1.67,fn,\zeta,T)$

Compute some example values for the system (before graphs for optimising the parameters).

$\zeta := 1.14$

$fn := 90$

$T := 10^{-4}$

$mm1 := MS1(fn,\zeta,T)$

Noise (thermal)

$mm2 := MS2(fn,\zeta,L0,f\_FL,T)$

Phase Noise

$mm3 := MS3(fn,\zeta,Kcr,Ksp,T)$

EDA phase effects.

$mm3a := Ksp\cdot TE(1.67,fn,\zeta,T)$

EDA spin effect only (A)

$mm3b := Kcr\cdot TE(53.3,fn,\zeta,T)$

EDA commutation only (B)

Evaluate the relative magnitude of the different effects.

$mtotal := mm1 + mm2 + mm3a + mm3b$

$dg(mtotal) = 2.261$

$\frac{mm1}{mtotal} = 0.432$

$\frac{mm2}{mtotal} = 0.024$

$\frac{mm3a}{mtotal} = 2.081\cdot 10^{-4}$

$\frac{mm3b}{mtotal} = 0.543$

Set up a function for computing the total sum square error.

$$TMS(fn,\zeta,L0,f\_FL,Kcr,Ksp,T) := MS1(fn,\zeta,T) \ldots + MS2(fn,\zeta,L0,f\_FL,T) \ldots + MS3(fn,\zeta,Kcr,Ksp,T)$$

And a set of P log-spaced frequency points from "lower" to "upper".

$P := 20$

$lower := 30$

$upper := 400$

$kk := 0..P$

$a := lower$

$$b := \exp\left(\frac{\ln\left(\frac{upper}{a}\right)}{P}\right)$$

$vfn_{kk} := a \cdot b^{kk}$

$mse_{kk} := TMS(vfn_{kk},\zeta,L0,f\_FL,Kcr,Ksp,0)$

Zero Delay

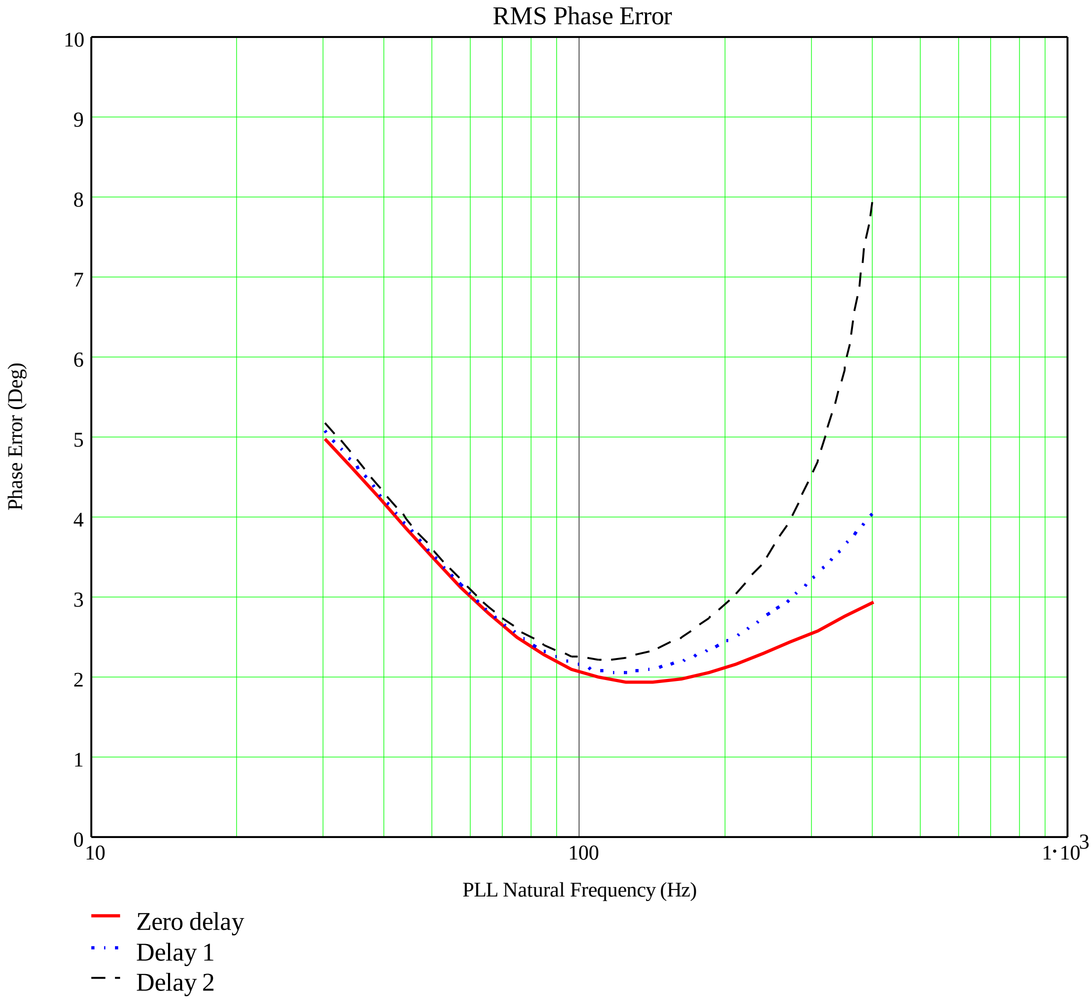
$mseD1_{kk} := TMS(vfn_{kk},\zeta,L0,f\_FL,Kcr,Ksp,10^{-4})$

Delay #1

$mseD2_{kk} := TMS(vfn_{kk},\zeta,L0,f\_FL,Kcr,Ksp,2\cdot 10^{-4})$

Delay #2

Plot out these curves to find the "optimum" choice of loop natural frequency for minimum total phase error in the recovered carrier.



Compute the loop SNR (dB, in F.M. Gandner's notation).

$$snrl_{kk} := dB\left(\frac{1}{2 \cdot mse_{kk}}\right)$$

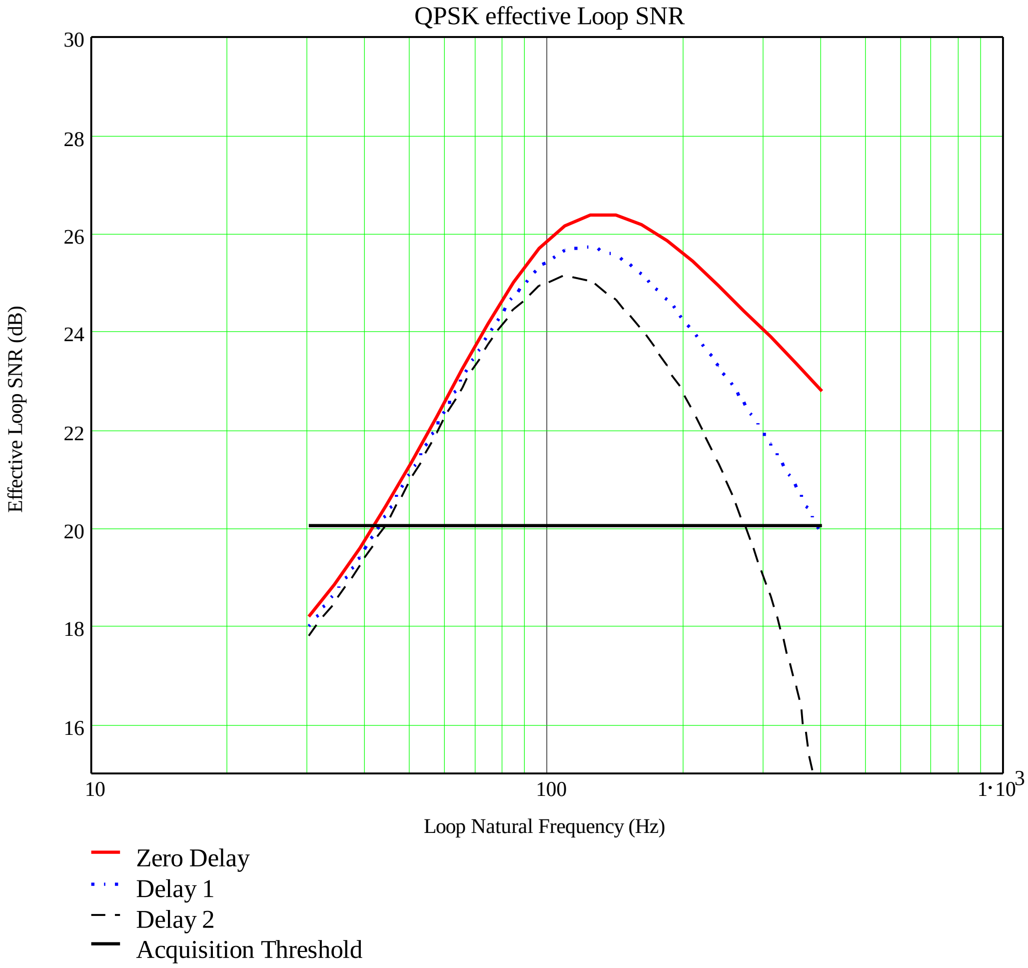
$$snrlD1_{kk} := dB\left(\frac{1}{2 \cdot mseD1_{kk}}\right)$$

$$snrlD2_{kk} := dB\left(\frac{1}{2 \cdot mseD2_{kk}}\right)$$

$$M := 4$$

$$snrACQ_{kk} := 8 + dB\left(M^2\right)$$

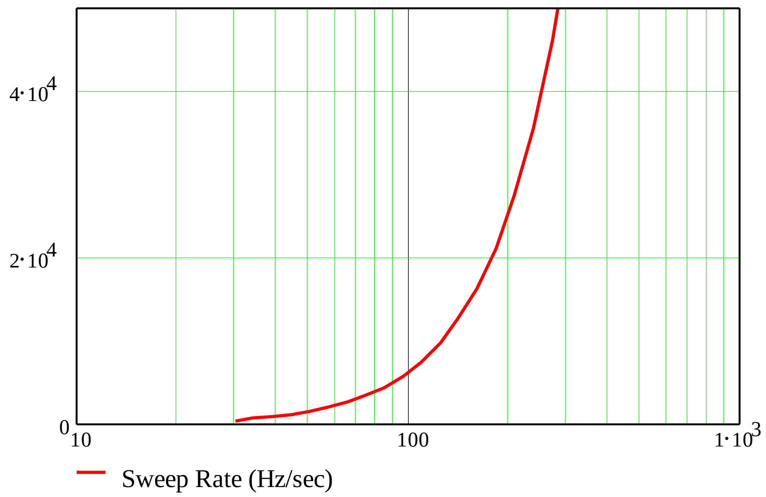
Acquisition threshold, 8dB (fastest sweep
and M^2 for Mth order recovery loop).



Sweep rate calculations (Delay #1 only):

$$\mathrm{rate2}(p, wn) := wn^2 \cdot \mathrm{if}\left[p > \frac{9.5}{2}, 0.4, \mathrm{if}\left[p < \frac{6}{2}, 0, \left(1 - \sqrt{\frac{1}{p-2}}\right)\right]\right]$$

$$\mathrm{sweep}_{kk} := \frac{\mathrm{rate2}\left(\frac{1}{2 \cdot \mathrm{mseD1}_{kk}} \cdot \frac{1}{M^2}, \mathrm{vfn}_{kk} \cdot 2 \cdot \pi\right)}{2 \cdot \pi \cdot M}$$



Function that uses the simulated (and saved above) Viterbi decoder results to model the CCSDS error rate versus Eb/No more accurately.

$ccsds := READPRN(ccsds\_lut)$

$Nccsds := rows(ccsds)$

$Nccsds = 32$

$icc := 0 .. Nccsds - 1$

$cc\_eb_{icc} := ccsds_{icc,0}$

$cc\_logPE_{icc} := ccsds_{icc,1}$

$cc\_vs := cspline(cc\_eb, cc\_logPE)$

$cc\_ifunc(v) := 10^{interp(cc\_vs, cc\_eb, cc\_logPE, v)}$

$vMax := ccsds_{Nccsds-1,0}$

$ErrMin := 10^{ccsds_{Nccsds-1,1}}$

$pef(v) := if(v<0, 0.5, if(v>vMax, ErrMin, cc\_ifunc(v)))$

And an inverse error rate formulae (requires care to prevent convergence failure):

$IpeLIM := vMax$

Maximum returned value.

$peLIM := ErrMin$

Corresponding input value

$guess := idB(2.7)$

$pediff(erate, guess) := log(erate) - log(pef(guess))$

Use logs to remove slope

$ipe(erate, guess) := root(pediff(erate, guess), guess)$

$INVpe(r) := if(r>peLIM, ipe(r, guess), IpeLIM)$

$p := pef(idB(2.7))$

$p = 2.472 \cdot 10^{-8}$

Quick test of inverse function.

$dB(INVpe(p)) = 2.7$

And the function for the probability distrabution of the PLL phase. Here N is the number of stable points. With 1 for CW PLL (or residual carrier PSK), 2 = BPSK and 4 = QPSK. There is an odd problem with MathCAD and the exp()/I0() term - it is more reliable to use power -1 for some odd reason.

$$\alpha(prms) := \frac{1}{prms}$$

'Loop SNR' used by Spilker, = 2*SNRloop of Gardner

$$Pb(\theta,\alpha,M) := \text{if}\left[\frac{\alpha}{M^2}>50, \sqrt{\frac{\alpha}{2\cdot\pi}}\cdot\exp\left(\frac{-\alpha\cdot\theta^2}{2}\right), M\cdot\exp\left(\frac{\alpha}{M^2}\cdot\cos(M\cdot\theta)\right)\cdot\left(2\cdot\pi\cdot I0\left(\frac{\alpha}{M^2}\right)\right)^{-1}\right]$$

Now evaluate the probability of error (include phasor cross talk) and the loss involved in the typical case.

$$E1(en,\theta) := pef\left[en\cdot(\cos(\theta)+\sin(\theta))^2\right]$$

Signals add

$$E2(en,\theta) := pef\left[en\cdot(\cos(\theta)-\sin(\theta))^2\right]$$

Signals subtract

No cross talk case.

$$E3(en,\theta) := pef\left(en\cdot\cos(\theta)^2\right)$$

$$peBPSK(en,\theta) := E3(en,\theta)$$

$$peSQPSK(en,\theta) := 0.5\cdot E3(en,\theta) + 0.25\cdot E1(en,\theta) + 0.25\cdot E2(en,\theta)$$

Uncoded case

$$peQPSK(en,\theta) := 0.5\cdot E1(en,\theta) + 0.5\cdot E2(en,\theta)$$

$$Kfec := 2$$

$$Rc := \frac{223}{512}$$

$$peQPSK(en,\theta) := pef\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right)+Kfec\cdot Rc\cdot\sin(\theta)^2}\right]$$

Coded case.

Perform the integration using the assumed symmetry of Pb() and use choice of region to avoid integration convergance accuracy problems.

$$Sdev := 8$$

Number of "standard deviations" to be integrated.

Create a general integration function which uses Sdev or π/N, whichever is smaller. The calling values are the Eb/No input, the α = (1/mean square error), the number of phase values (2=BPSK, 4=QPSK) and the Probabillity of Error Function specific to the system under consideration.

$$IntFunc(en,\alpha,M,Pef) := 2\cdot\int_0^{\text{if}\left[\alpha<\left(\frac{M\cdot Sdev}{\pi}\right)^2,\frac{\pi}{M},\frac{Sdev}{\sqrt{\alpha}}\right]} Pb(\theta,\alpha,M)\cdot Pef(en,\theta)\,d\theta$$

$$Perr(en,\alpha) := IntFunc(en,\alpha,4,peQPSK)$$

Now find the error rates, and convert this back to the equivalent loss in Eb/No for the system:

$$target\_perr := 5\cdot 10^{-9}$$

$$en := INVpe(target\_perr)$$

$$dB(en) = 2.752$$

$$guess := en\cdot 1.01$$

$$derrF(\alpha,en) := \log(target\_perr) - \log(Perr(en,\alpha))$$

$$lossF(\alpha) := dB(\text{if}(Perr(en\cdot idB(1.5),\alpha)>target\_perr, 100, root(derrF(\alpha,guess),guess))) - dB(en)$$

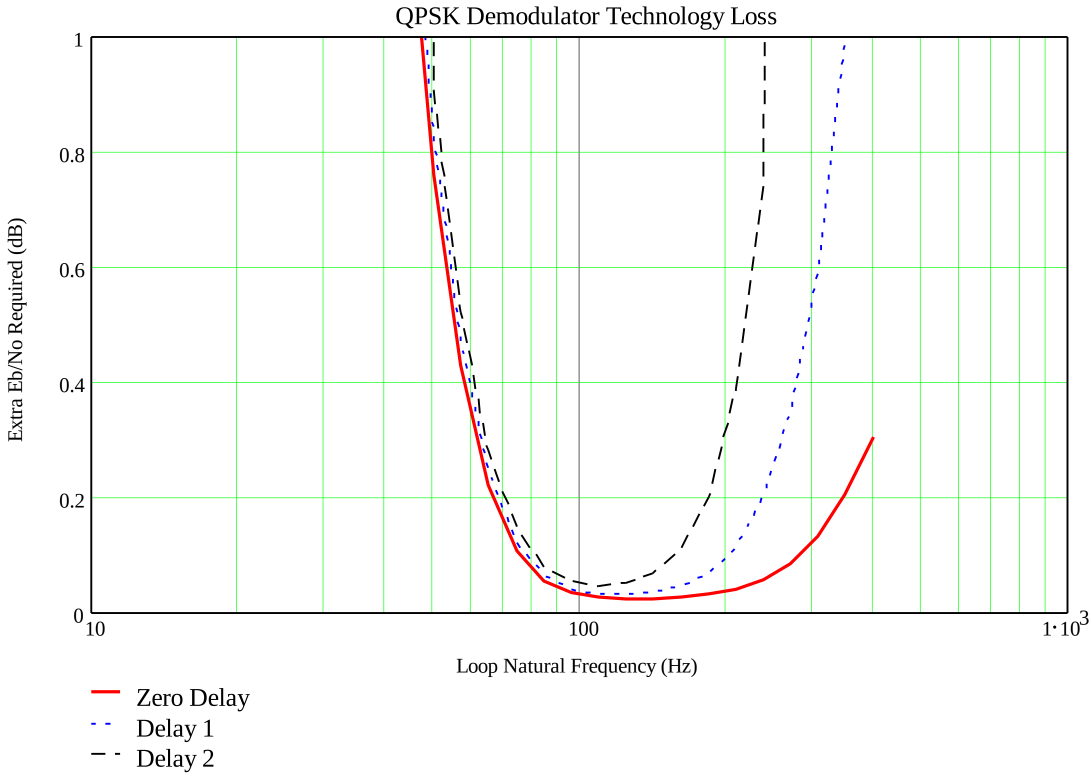
$$kk2 := 0..P$$

$$Lmse_{kk2} := lossF\left(\alpha\left(mse_{kk2}\right)\right)$$

$$LmseD1_{kk2} := lossF\left(\alpha\left(mseD1_{kk2}\right)\right)$$

$$LmseD2_{kk2} := lossF\left(\alpha\left(mseD2_{kk2}\right)\right)$$

Plot out technology loss (dB) against natural frequency:



vfn =

	0
0	30
1	34.148
2	38.87
3	44.245
4	50.363
5	57.327
6	65.253
7	74.276
8	84.547
9	96.237
10	109.545
11	124.692
12	141.933
13	161.559
14	183.899
15	209.327

LmseD1 =

	0
0	17.248
1	17.248
2	17.248
3	1.394
4	0.832
5	0.476
6	0.252
7	0.126
8	0.068
9	0.045
10	0.037
11	0.035
12	0.039
13	0.049
14	0.071
15	0.117

snrlD1 =

	0
0	18.017
1	18.686
2	19.442
3	20.278
4	21.18
5	22.124
6	23.072
7	23.971
8	24.752
9	25.342
10	25.684
11	25.758
12	25.585
13	25.211
14	24.69
15	24.062

-- End of file --

File RCOS8HRIT.MCD to compute effects of baseband filters on data detection and timing recovery.

$IF3dB := 2\cdot 10^{6}$

$IFeps := 0.0$

IF filter LPF BW and error (for component sensitivity analysis).

$dB(x) := if\left(|x| < 10^{-12}, -120, 10\cdot \log(|x|)\right)$

$sinc(x) := if\left(|x| < 10^{-8}, 1.0, \frac{\sin(x)}{x}\right)$

$idB(x) := 10^{\frac{x}{10}}$

$j := \sqrt{-1}$

$raised\_cos(f, a, T) := if\left[|f| < \frac{1-a}{2\cdot T}, 1, if\left[|f| > \frac{1+a}{2\cdot T}, 0, \frac{1}{2}\cdot\left[1 + \cos\left[\frac{\pi\cdot(2\cdot|f|\cdot T - 1 + a)}{2\cdot a}\right]\right]\right]\right]$

$vmax(x) := if(|max(x)| > |min(x)|, |max(x)|, |min(x)|)$

Vector maximum, for complex numbers

$N := 512$

Number of FFT points

$en := idB(2.8)$

Eb/No, from dB value

Correction for Forward Error Correction system handling of symbol correlated noise

$Kfec := 2$

$Rc := \frac{223}{512}$

Code rate for FEC system.

$Wpow := 0$

FFT Window, cos() power term

$Wstart := 1.0$

FFT Window, start of cos() roll-off (symbols)

$alpha := 0.7$

Roll off factor, a=0.7 HRIT, and a=1.0 LRIT

$SymbolRate := 10^{6}\cdot\frac{255}{223}$

1E6 with 255/223 rate coding HRIT, 128E3 with 2*255/223 rate coding

$T := \frac{1}{SymbolRate}$

Bit period

$T = 8.7$

$clkin := 35\cdot 10^{6}$

A/D (input) clock, max 52MHz

$procclk := 35\cdot 10^{6}$

Processor clock, max 35MHz

$Rcic := 5$

CIC decimation factor, 5=HRIT and 14=LRIT

$Fcic := \frac{clkin}{Rcic}$

$Fcic = 7\cdot 10^{6}$

CIC output rate = FIR input rate

$Rfir := 2$

FIR decimation factor, 2 for HRIT & LRIT

$$Ffir := \frac{Fcic}{Rfir}$$

$Ffir = 3.5 \cdot 10^{6}$

FIR output rate = Polyphase input rate

$$Rpoly := \frac{Ffir}{2 \cdot SymbolRate}$$

Effective decimation factor for polyphase filter. This is NOT an integer and performs the re-sampling to the 2 sample/symbol output, input to DSP software.

$Rpoly = 1.53$

$Sym := 1$

Boolean, 1 = symmetric FIR filter design

$Odd := 0$

Boolean, 1 = odd number of FIR coefficients

$$Ntap := floor\left[\frac{procclk}{\left(\frac{Fcic}{Rfir}\right)} - Rfir\right] \cdot (1 + Sym) - (Sym \cdot Odd)$$

$Ntap := if(Ntap > 255, 255, Ntap)$

$Ntap = 16$

Number of FIR filter taps available.

$$Zext := \frac{Ntap}{Fcic \cdot T}$$

$Zext = 2.614$

Bit periods for external hardware

$$M := \frac{Ntap}{Zext}$$

$M = 6.122$

Number of FIR points per symbol calculated.

$ii := 0 .. N - 1$

Data point index

$$jj := N - 1 .. \frac{N}{2}$$

Conjugate copy index

$$f_{ii} := \frac{ii}{N} \cdot \frac{M}{T}$$

Frequency array

$S_{ii} := raised\_cos\left(f_{ii}, alpha, T\right)$

$H_{ii} := \sqrt{S_{ii}}$

$H_{jj} := \overline{H_{N-jj}}$

Ideal RRC matched filter

$mm := 0 .. 3$

Set up pole positions for the analogue IF filter (LPF equivalent)

$Pole_{0} := -0.9952088 + j \cdot 1.2571058$

$Pole_{1} := -0.9952088 - j \cdot 1.2571058$

$Pole_{2} := -1.3700679 + j \cdot 0.4102497$

$Pole_{3} := -1.3700679 - j \cdot 0.4102497$

$$Kdc := \prod_{mm} Pole_{mm}$$

$Kdc = 5.2$

$$\text{Fif\_basic}(f, f3dB) := \frac{Kdc}{\prod_{mm} \left(Pole_{mm} + j \cdot \frac{f}{f3dB}\right)}$$

$\text{Fif}(f, f3dB) := \text{Fif\_basic}(f, f3dB)$

Must use magnitude only, if the symmetric filter used (always for the MUBM) since it cant cope with GD variations.

$\text{Fif}(f, f3dB) := \text{if}(Sym = 0, \text{Fif\_basic}(f, f3dB), |\text{Fif\_basic}(f, f3dB)|)$

$$GD := \frac{\arg(\text{Fif}(IF3dB, IF3dB))}{IF3dB \cdot 2 \cdot \pi}$$

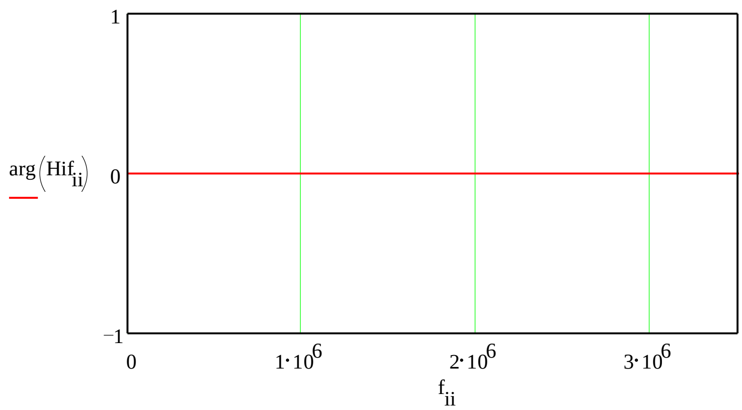
$GD = 0$

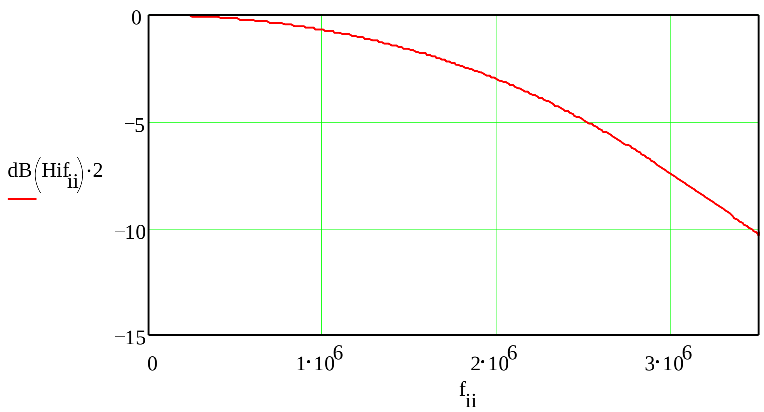
Group delay = dβ/dω

$Hif_{ii} := \text{Fif}\left[f_{ii}, IF3dB \cdot (1 + IFeps)\right] \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot GD\right)$

Correct for zero phase shift at 3dB point.

$Hif_{jj} := \overline{Hif_{N-jj}}$





$$Hcic_{ii} := \text{sinc}\left(\frac{\pi \cdot f_{ii}}{Fcic}\right)^5$$

The 5th order CIC decimation filter frequency response

$Hcic_{jj} := \overline{Hcic_{N-jj}}$

Read polyphase coeficients.

$hp := \text{READPRN}(polyphase)$

$Nhp := rows(hp)$

$Nhp = 192$

$pp := 0 .. Nhp - 1$

$pmax := max(hp)$

$$ptmax := \frac{6 \cdot Rfi}{M}$$

$ptmax = 1.96$

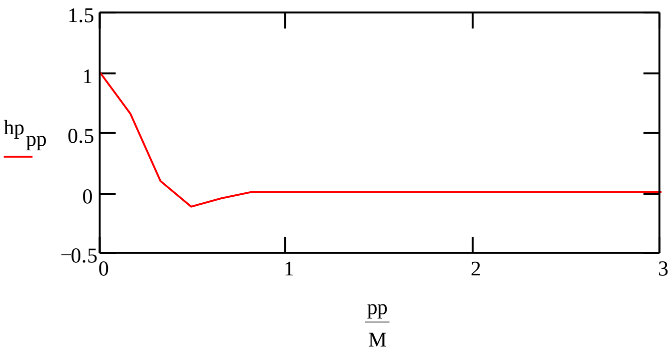
$$px_{pp} := \frac{pp}{Nhp - 1} \cdot \frac{ptmax}{1}$$

$$py := \frac{hp}{pmax}$$

$ps := cspline(px, py)$

$$pfunc(t) := if\left[(t<0) + \left(t > \frac{ptmax}{2}\right), 0, interp\left(ps, px, py, t + \frac{ptmax}{2}\right)\right]$$

$$hp_{ii} := pfunc\left(\frac{ii}{M}\right)$$

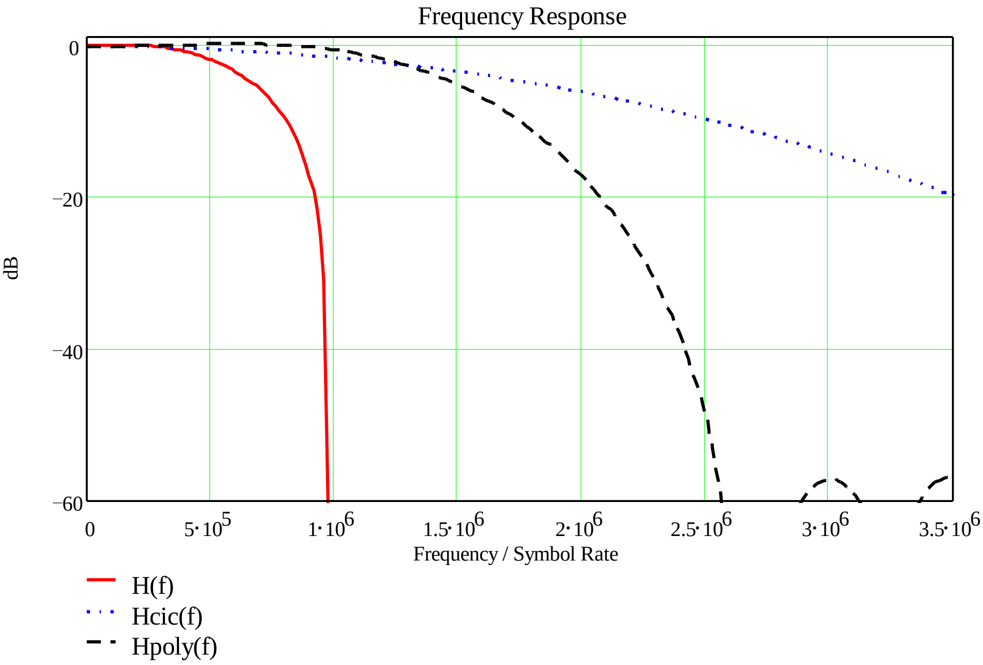


$$hp_{jj} := \overline{hp_{N - jj}}$$

$Hpoly := ICFFT(hp)$

$pmax := \left|Hpoly_0\right|$

$$Hpoly_{ii} := \frac{Hpoly_{ii}}{pmax}$$

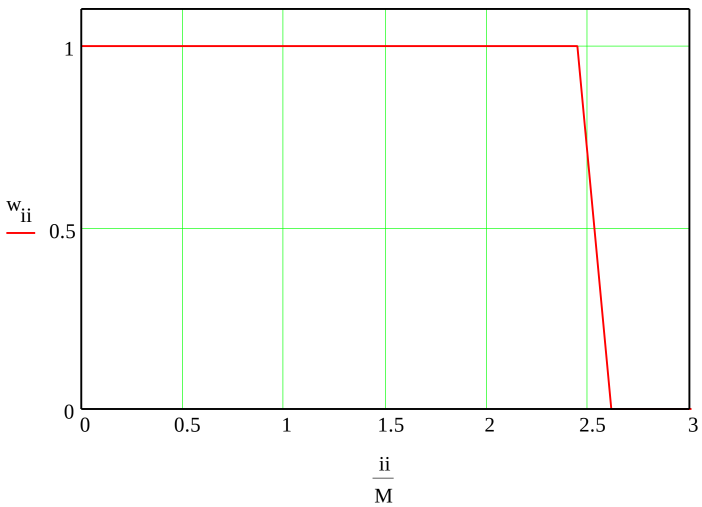


External FIR "window" function

$$imid := \frac{Ntap - 1}{2.0}$$

$$imid = 7.5$$

$$w_{ii} := \mathrm{if}\left[\left|\frac{ii - imid}{M}\right| < Wstart, 1, \mathrm{if}\left[\left|\frac{ii - imid}{M}\right| > \frac{Zext}{2}, 0, \cos\left[\frac{\left|\frac{ii - imid}{M}\right| - Wstart}{\left(\frac{Zext}{2}\right) - Wstart}\cdot\frac{\pi}{2}\right]^{Wpow}\right]\right]$$



$$H_{ii} := H_{ii}\cdot\exp\left(-j\cdot 2\cdot\pi\cdot f_{ii}\cdot\frac{imid}{M}\cdot T\right)$$

Center on mid window, delay time = imid/M.

$$H_{jj} := \overline{H_{N-jj}}$$

$$hid := \mathrm{ICFFT}(H)$$

Time domain version of ideal filter

$hmax := vmax(hid)$

$hid_{ii} := hid_{ii} \cdot \frac{1}{hmax}$

Normalised to h(t=0)=1, for on-time signal.

$H1_{ii} := H_{ii}$

No compensation

$$H1_{ii} := \frac{H_{ii}}{Hcic_{ii} \cdot Hpoly_{ii} \cdot \left(Fif\left(f_{ii}, IF3dB \right) \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot GD \right) \right)}$$

$H1_{jj} := \overline{H1_{N-jj}}$

$hmax := \left| H1_0 \right|$

$H1_{ii} := \frac{H1_{ii}}{hmax}$

$h1 := ICFFT(H1)$

Time domain version of ideal filter

$hmax := vmax(h1)$

$h1_{ii} := h1_{ii} \cdot \frac{1}{hmax}$

Normalised to h(0)=1

$h2_{ii} := h1_{ii} \cdot w_{ii}$

h2 is windowed version

$hmax := vmax(h2)$

$h2_{ii} := h2_{ii} \cdot \frac{1}{hmax}$

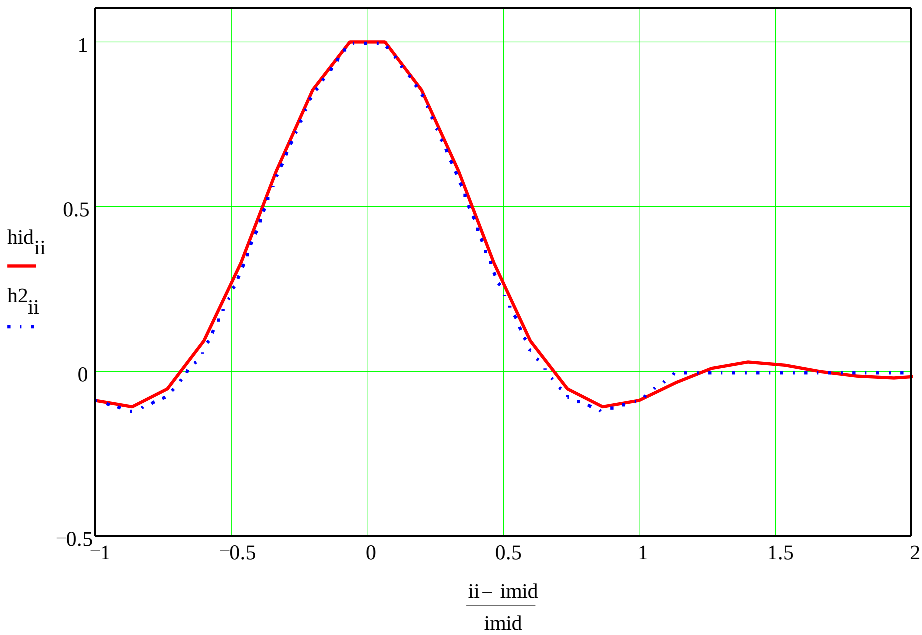
$H2 := CFFT(h2)$

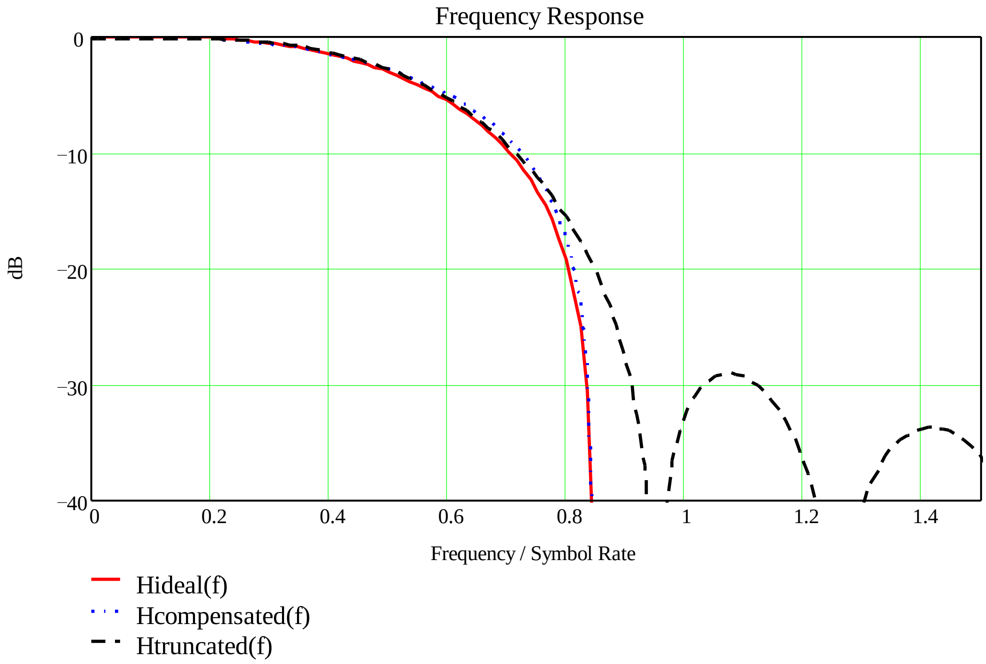
$hmax := \left| H2_0 \right|$

Normalised and with -ve frequency after window.

$H2_{ii} := H2_{ii} \cdot \frac{1}{hmax}$

H2 now has the distorted (true) FIR response





$$E_{ii} := \mathrm{if}\left(\left|H1_{ii}\right| > 0, \left|\frac{H1_{ii}}{H2_{ii}}\right|, 1\right)$$

Pre-distorted response.

$$H4_{ii} := H_{ii}\cdot \mathrm{if}\left[\left(\left|E_{ii}\right| < 2\right)\cdot\left(\left|E_{ii}\right| > 0.5\right), \left(\left|E_{ii}\right|\right)^{2}, 1\right]$$

$$H4_{jj} := \overline{H4_{N-jj}}$$

$$h4 := \mathrm{ICFFT}(H4)$$

$$h4_{ii} := h4_{ii}\cdot w_{ii}$$

$$hmax := \mathrm{vmax}(h4)$$

$$h4_{ii} := \frac{h4_{ii}}{hmax}$$

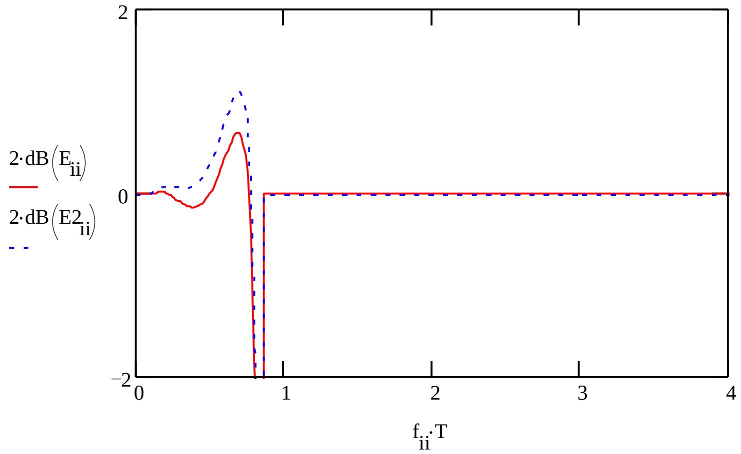
$$H4 := CFFT(h4)$$

$$hmax := \left|H4_0\right|$$

$$H4_{ii} := H4_{ii} \cdot \frac{1}{hmax}$$

Re-compute the error:

$$E2_{ii} := if\left(\left|H1_{ii}\right| > 0, \left|\frac{H1_{ii}}{H4_{ii}}\right|, 1\right)$$



$$H_{ii} := H_{ii} \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot \frac{-imid}{M} \cdot T\right)$$

Center back on t=0, delay time = imid/M.

$$H_{jj} := \overline{H_{N-jj}}$$

$$Hover2_{ii} := H2_{ii} \cdot Hcic_{ii} \cdot Hpoly_{ii} \cdot Hif_{ii}$$

$$Hover2_{ii} := Hover2_{ii} \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot \frac{-imid}{M} \cdot T\right)$$

Center back on t=0, delay time = imid/M.

$$Hover2_{jj} := \overline{Hover2_{N-jj}}$$

$$S_{ii} := H_{ii} \cdot Hover2_{ii}$$

Perfect RRC and non-predistorted designed filters in cascade.

$$S_{jj} := \overline{S_{N-jj}}$$

Copy conjugate for -ve spectrum

$$s := ICFFT(S)$$

In to time domain.

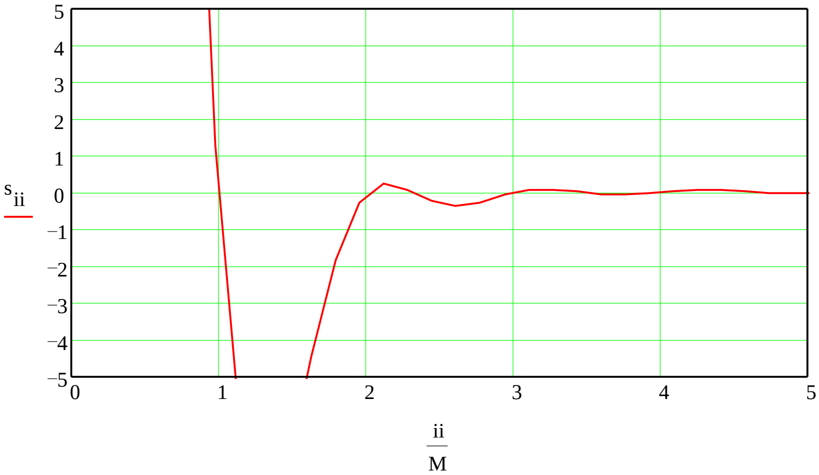
$$smax := vmax(s)$$

Normalise curve to maximum (on-time) value.

$$s_{ii} := s_{ii} \cdot \frac{100}{smax}$$

For a display of % volatage.

Plot the response (expanded to show residual error):



Compute the Root Sum Square error, assume even symmetry:

$vx_{ii} := ii$

$vy := Re(s)$

$vs := cspline(vx, vy)$

$$un\_isi2 := \frac{2 \cdot \sum_{kk=1}^{floor\left(\frac{N}{2 \cdot M}\right)} interp(vs, vx, vy, kk \cdot M)^2}{\left(\left|s_0\right|\right)^2}$$

$un\_isi2 = 6.955 \cdot 10^{-7}$

$$un\_isi\_loss := \frac{1}{1 - en \cdot Kfec \cdot Rc \cdot un\_isi2}$$

$$NNmax := floor\left(\frac{N-1}{2}\right)$$

$nn := 0 .. NNmax$

$$un\_snr\_loss := \frac{\left[\left|\sum_{nn}\left(H_{nn} \cdot H_{nn}\right)\right|\right]^2}{\sum_{nn}\left(\left|H_{nn}\right|\right)^2} \cdot \frac{\sum_{nn}\left(\left|Hover2_{nn}\right|\right)^2}{\left[\left|\sum_{nn}\left(H_{nn} \cdot Hover2_{nn}\right)\right|\right]^2}$$

$un\_snr\_loss = 1.001$

$dB(un\_snr\_loss) = 4.25 \cdot 10^{-3}$

$un\_dB\_loss := dB(un\_isi\_loss) + dB(un\_snr\_loss)$

Test overall response to evaluate ISI under optimum timing (limit of performance) for pre-compensated design:.

$Hover4_{ii} := H4_{ii} \cdot Hcic_{ii} \cdot Hpoly_{ii} \cdot Hif_{ii}$

$$Hover4_{ii} := Hover4_{ii} \cdot exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot \frac{-imid}{M} \cdot T\right)$$

Center back on t=0, delay time = imid/M.

$Hover4_{jj} := \overline{Hover4_{N-jj}}$

$S_{ii} := H_{ii} \cdot Hover4_{ii}$

Perfect RRC and designed filters in cascade.
Copy conjugate for -ve spectrum

$S_{jj} := \overline{S_{N-jj}}$

In to time domain.

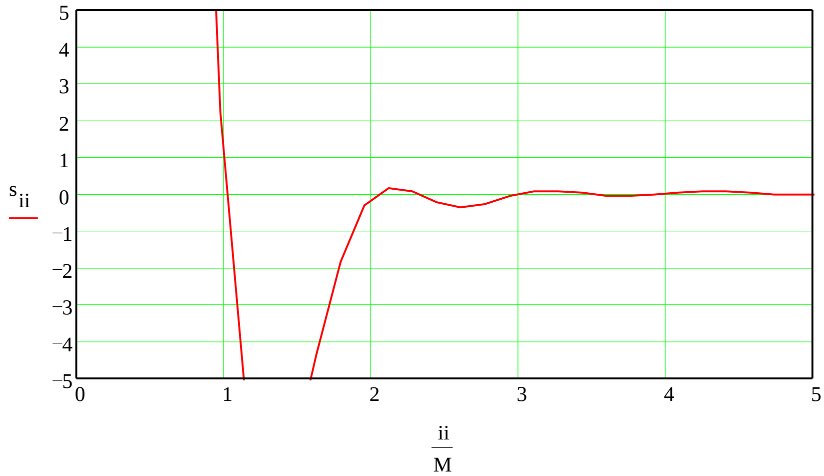
$s := \mathrm{ICFFT}(S)$

Normalise curve to maximum (on-time) value.

$smax := \left|s_0\right|$

$s_{ii} := s_{ii} \cdot \frac{100}{smax}$

For a display of % volatage.
Plot the response:



Compute the Root Sum Square error, assume even symmetry:

$vx_{ii} := ii$

$vy := \mathrm{Re}(s)$

$vs := \mathrm{cspline}(vx, vy)$

$$pre\_isi2 := \frac{2 \cdot \sum_{kk=1}^{\mathrm{floor}\left(\frac{N}{2 \cdot M}\right)} \mathrm{interp}(vs, vx, vy, kk \cdot M)^2}{\left(\left|s_0\right|\right)^2}$$

$pre\_isi2 = 1.857 \cdot 10^{-4}$

$\sqrt{pre\_isi2} = 0.014$

$\mathrm{dB}(pre\_isi2) = -37.313$

$$pre\_isi\_loss := \frac{1}{1 - en \cdot Kfec \cdot Rc \cdot pre\_isi2}$$

$$pre\_snr\_loss := \frac{\left[\left|\sum_{nn}\left(H_{nn} \cdot H_{nn}\right)\right|\right]^2}{\sum_{nn}\left(\left|H_{nn}\right|\right)^2} \cdot \frac{\sum_{nn}\left(\left|Hover4_{nn}\right|\right)^2}{\left[\left|\sum_{nn}\left(H_{nn} \cdot Hover4_{nn}\right)\right|\right]^2}$$

$pre\_snr\_loss = 1.001$

$pre\_dB\_loss := \mathrm{dB}(pre\_isi\_loss) + \mathrm{dB}(pre\_snr\_loss)$

Compare both approaches

$dB(un\_snr\_loss) = 4.25 \cdot 10^{-3}$

$dB(un\_isi\_loss) = 5.014 \cdot 10^{-6}$

$un\_dB\_loss = 4.255 \cdot 10^{-3}$

$dB(pre\_snr\_loss) = 5.49 \cdot 10^{-3}$

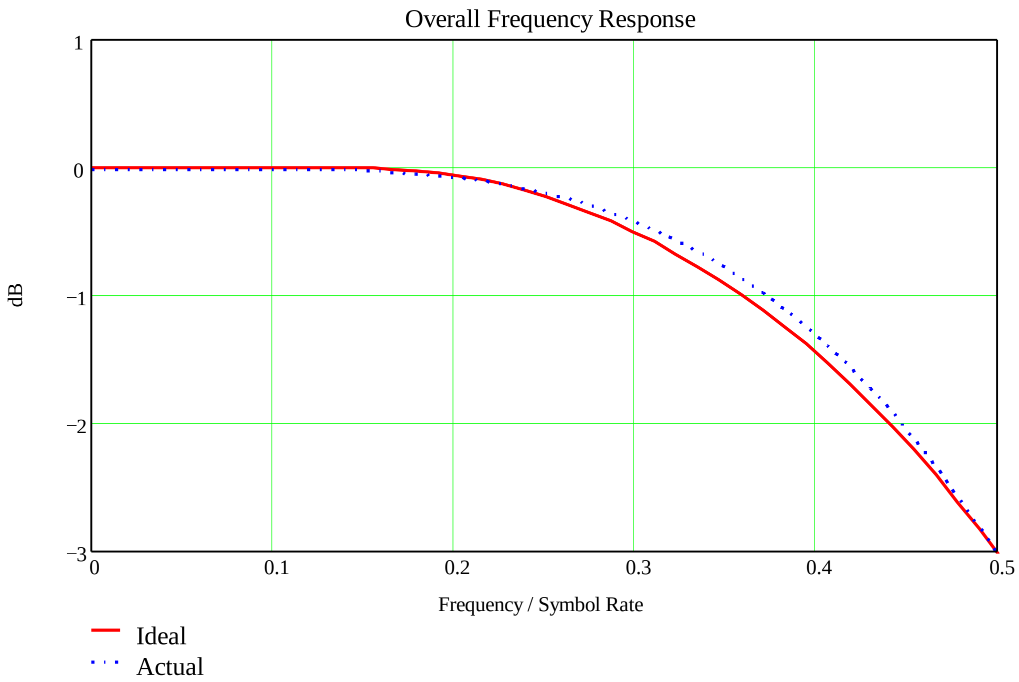
$dB(pre\_isi\_loss) = 1.338 \cdot 10^{-3}$

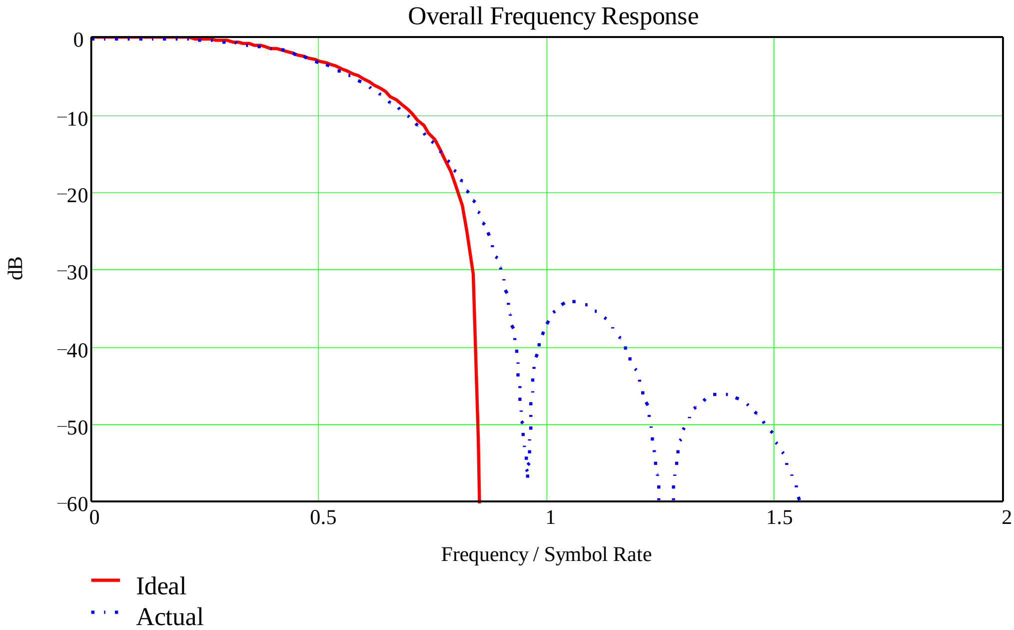
$pre\_dB\_loss = 6.828 \cdot 10^{-3}$

Pick the best performer: 2 = un_ 4 = pre_

Hused := Hover2

h_used := h2





Finally write output file with HSP50214 filter coefficients

$\text{nint}(z) := \text{if}(\text{Re}(z)<0, \text{ceil}(\text{Re}(z) - 0.5), \text{floor}(\text{Re}(z) + 0.5)) + j \cdot \text{if}(\text{Im}(z)<0, \text{ceil}(\text{Im}(z) - 0.5), \text{floor}(\text{Im}(z) + 0.5))$

cc := 0 .. Ntap − 1

Scale to 22bit coefficients

h_used =

	0
0	-0.084
1	-0.116
2	-0.075
3	0.068
4	0.305
5	0.59
6	0.847
7	1
8	1
9	0.847
10	0.59
11	0.305
12	0.068
13	-0.075
14	-0.116
15	-0.084

$h\_op_{cc} := \text{Re}\left[\text{nint}\left[h\_used_{cc} \cdot \left(2^{21} - 1\right)\right]\right]$

PRNPRECISION := 8

Set up for nice file output

PRNCOLWIDTH := 8

WRITEPRN(hrit_fil) := h_op

$pre\_dB\_loss = 6.828 \cdot 10^{-3}$

$un\_dB\_loss = 4.255 \cdot 10^{-3}$

--End of File--

File RCOS8LRIT.MCD to compute effects of baseband filters on data detection and timing recovery.

$IFeps := 0.0$

$IF3dB := 2\cdot 10^{6}$

$dB(x) := if\left(|x| < 10^{-12}, -120, 10\cdot \log(|x|)\right)$

$sinc(x) := if\left(|x| < 10^{-8}, 1.0, \frac{\sin(x)}{x}\right)$

$idB(x) := 10^{\frac{x}{10}}$

$j := \sqrt{-1}$

$raised\_cos(f, a, T) := if\left[|f| < \frac{1-a}{2\cdot T}, 1, if\left[|f| > \frac{1+a}{2\cdot T}, 0, \frac{1}{2}\cdot\left[1+\cos\left[\frac{\pi\cdot(2\cdot|f|\cdot T - 1 + a)}{2\cdot a}\right]\right]\right]\right]$

$vmax(x) := if(|max(x)| > |min(x)|, |max(x)|, |min(x)|)$

Vector maximum, for complex numbers

$N := 512$

Number of FFT points

$en := idB(2.8)$

Eb/No, from dB value

Correction for Forward Error Correction system handling of symbol correlated noise

$Kfec := 2$

$Rc := \frac{223}{512}$

Code rate for FEC system.

$Wpow := 0$

FFT Window, cos() power term

$Wstart := 1.0$

FFT Window, start of cos() roll-off (symbols)

$alpha := 1.0$

Roll off factor, a=0.7 HRIT, and a=1.0 LRIT

$SymbolRate := 128\cdot 10^{3}\cdot 2\cdot \frac{255}{223}$

1E6 with 255/223 rate coding HRIT, 128E3 with 2*255/223 rate coding

$T := \frac{1}{SymbolRate}$

Bit period

$T = 3.4$

$clkin := 35\cdot 10^{6}$

A/D (input) clock, max 52MHz

$procclk := 35\cdot 10^{6}$

Processor clock, max 35MHz

$Rcic := 14$

CIC decimation factor, 5=HRIT and 14=LRIT

$Fcic := \frac{clkin}{Rcic}$

$Fcic = 2.5$

CIC output rate = FIR input rate

$Rfir := 2$

FIR decimation factor, 2 for HRIT & LRIT

$$Ffir := \frac{Fcic}{Rfir}$$

$Ffir = 1.2$

FIR output rate = Polyphase input rate

$$Rpoly := \frac{Ffir}{2 \cdot SymbolRate}$$

Effective decimation factor for polyphase filter. This is NOT an integer and performs the re-sampling to the 2 sample/symbol output, input to DSP software.

$Rpoly = 2.1$

$Sym := 1$

Boolean, 1 = symmetric FIR filter design

$Odd := 0$

Boolean, 1 = odd number of FIR coefficients

$$Ntap := floor\left[\frac{procclk}{\left(\frac{Fcic}{Rfir}\right)} - Rfir\right] \cdot (1 + Sym) - (Sym \cdot Odd)$$

$Ntap := if(Ntap > 255, 255, Ntap)$

$Ntap = 52$

Number of FIR filter taps available.

$$Zext := \frac{Ntap}{Fcic \cdot T}$$

$Zext = 6.0$

Bit periods for external hardware

$$M := \frac{Ntap}{Zext}$$

$M = 8.5$

Number of FIR points per symbol calculated.

$ii := 0 .. N - 1$

Data point index

$$jj := N - 1 .. \frac{N}{2}$$

Conjugate copy index

$$f_{ii} := \frac{ii}{N} \cdot \frac{M}{T}$$

Frequency array

$S_{ii} := raised\_cos\left(f_{ii}, alpha, T\right)$

$H_{ii} := \sqrt{S_{ii}}$

$H_{jj} := \overline{H_{N - jj}}$

Ideal RRC matched filter

$mm := 0 .. 3$

$Pole_0 := -0.9952088 + j \cdot 1.2571058$

$Pole_1 := -0.9952088 - j \cdot 1.2571058$

$Pole_2 := -1.3700679 + j \cdot 0.4102497$

$Pole_3 := -1.3700679 - j \cdot 0.4102497$

$$Kdc := \prod_{mm} Pole_{mm}$$

$Kdc = 5.2$

$$Fif\_basic(f, f3dB) := \frac{Kdc}{\prod_{mm} \left(Pole_{mm} + j \cdot \frac{f}{f3dB}\right)}$$

Must use magnitude only, if the symmetric filter used since it cant cope with GD variations.

$Fif(f, f3dB) := if(Sym = 0, Fif\_basic(f, f3dB), |Fif\_basic(f, f3dB)|)$

$$GD := \frac{arg(Fif(IF3dB, IF3dB))}{IF3dB \cdot 2 \cdot \pi}$$

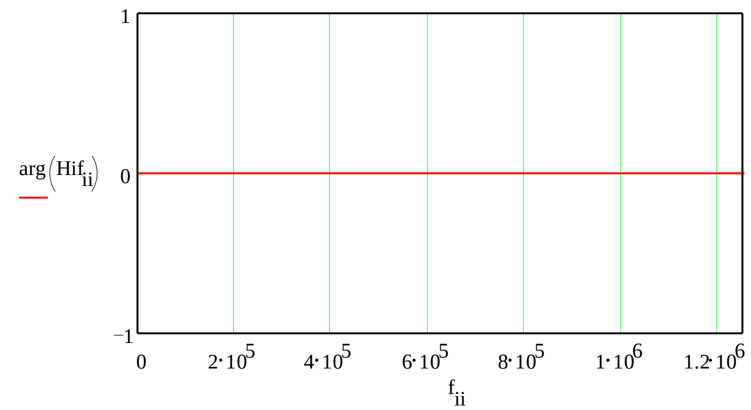
$GD = 0$

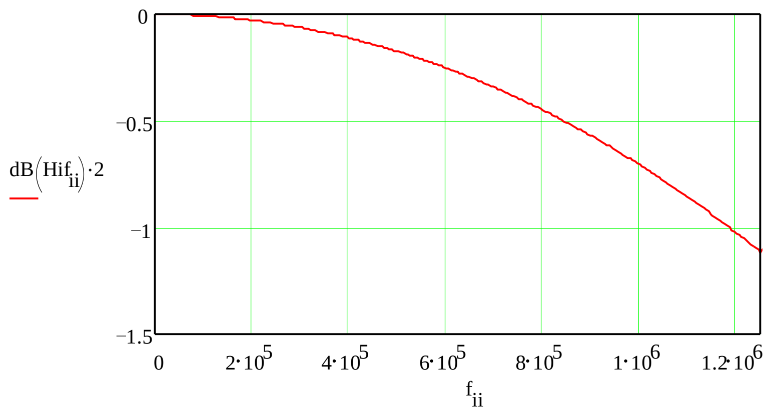
Group delay = dβ/dω

$Hif_{ii} := Fif\left[f_{ii}, IF3dB \cdot (1 + IFeps)\right] \cdot exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot GD\right)$

Correct for zero phase shift at 3dB point.

$Hif_{jj} := \overline{Hif_{N-jj}}$





$$Hcic_{ii} := sinc\left(\frac{\pi \cdot f_{ii}}{Fcic}\right)^5$$

$Hcic_{jj} := \overline{Hcic_{N-jj}}$

The CIC decimation filter frequency response

Read polyphase coeficients.

$hp := READPRN(polyphase)$

$Nhp := rows(hp)$

$Nhp = 192$

$pp := 0 .. Nhp - 1$

$pmax := max(hp)$

$ptmax := \frac{6 \cdot Rfi}{M}$

$ptmax = 1.405$

$px_{pp} := \frac{pp}{Nhp - 1} \cdot \frac{ptmax}{1}$

$py := \frac{hp}{pmax}$

$ps := cspline(px, py)$

$pfunc(t) := if\left[(t<0) + \left(t>\frac{ptmax}{2}\right), 0, interp\left(ps, px, py, t + \frac{ptmax}{2}\right)\right]$

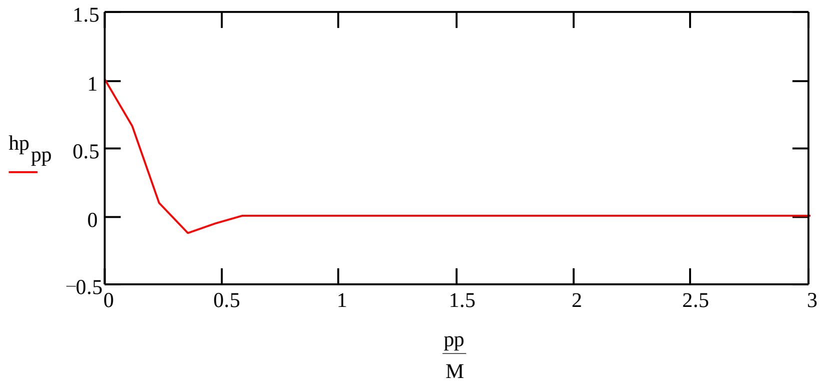
$hp_{ii} := pfunc\left(\frac{ii}{M}\right)$

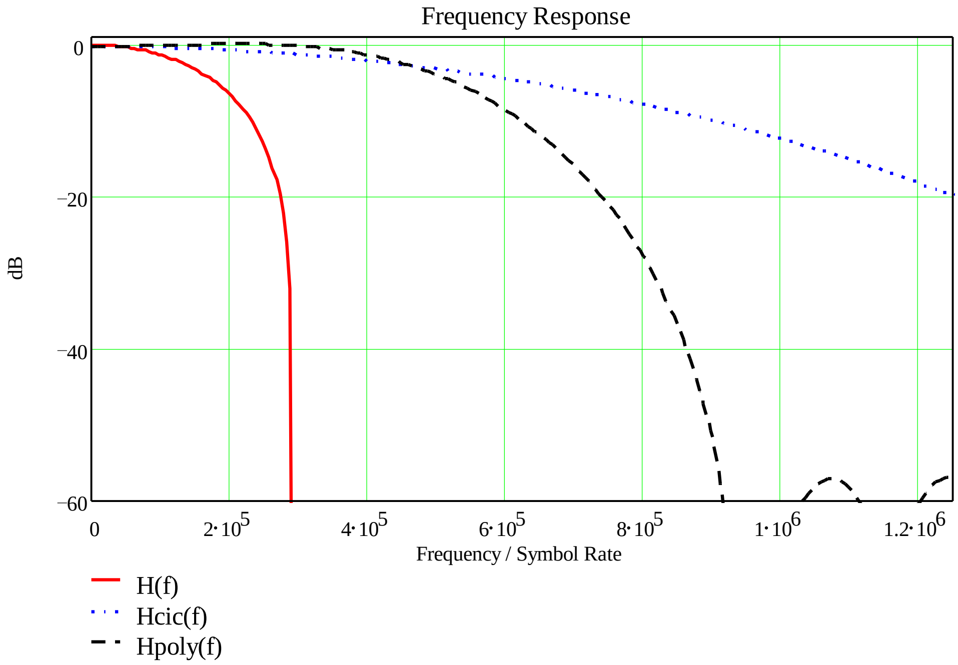
$hp_{jj} := \overline{hp_{N-jj}}$

$Hpoly := ICFFT(hp)$

$pmax := \left|Hpoly_0\right|$

$Hpoly_{ii} := \frac{Hpoly_{ii}}{pmax}$



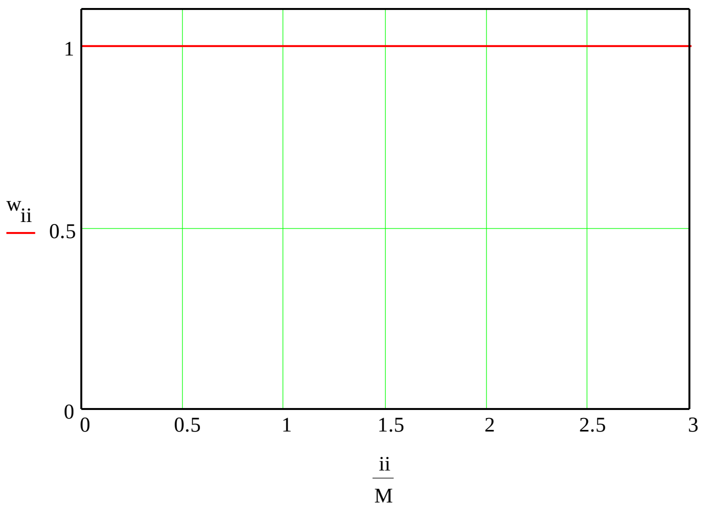


External FIR "window" function

$$imid := \frac{Ntap - 1}{2.0}$$

$imid = 25.5$

$$w_{ii} := if\left[\left|\frac{ii - imid}{M}\right| < Wstart, 1, if\left[\left|\frac{ii - imid}{M}\right| > \frac{Zext}{2}, 0, \cos\left[\frac{\left|\frac{ii - imid}{M}\right| - Wstart}{\left(\frac{Zext}{2}\right) - Wstart}\cdot\frac{\pi}{2}\right]^{Wpow}\right]\right]$$



$$H_{ii} := H_{ii}\cdot\exp\left(-j\cdot 2\cdot\pi\cdot f_{ii}\cdot\frac{imid}{M}\cdot T\right)$$

Center on mid window, delay time = imid/M.

$$H_{jj} := \overline{H_{N-jj}}$$

$hid := ICFFT(H)$

Time domain version of ideal filter

$hmax := vmax(hid)$

$$hid_{ii} := hid_{ii} \cdot \frac{1}{hmax}$$

Normalised to h(t=0)=1, for on-time signal.

$H1_{ii} := H_{ii}$

$$H1_{ii} := \frac{H_{ii}}{Hcic_{ii} \cdot Hpoly_{ii} \cdot \left(Fif\left(f_{ii}, IF3dB\right) \cdot exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot GD\right)\right)}$$

No compensation, just deconvolution.

$H1_{jj} := \overline{H1_{N-jj}}$

$hmax := \left|H1_0\right|$

$$H1_{ii} := \frac{H1_{ii}}{hmax}$$

$h1 := ICFFT(H1)$

Time domain version of ideal filter

$hmax := vmax(h1)$

$$h1_{ii} := h1_{ii} \cdot \frac{1}{hmax}$$

Normalised to h(0)=1

$h2_{ii} := h1_{ii} \cdot w_{ii}$

h2 is windowed version

$hmax := vmax(h2)$

$$h2_{ii} := h2_{ii} \cdot \frac{1}{hmax}$$

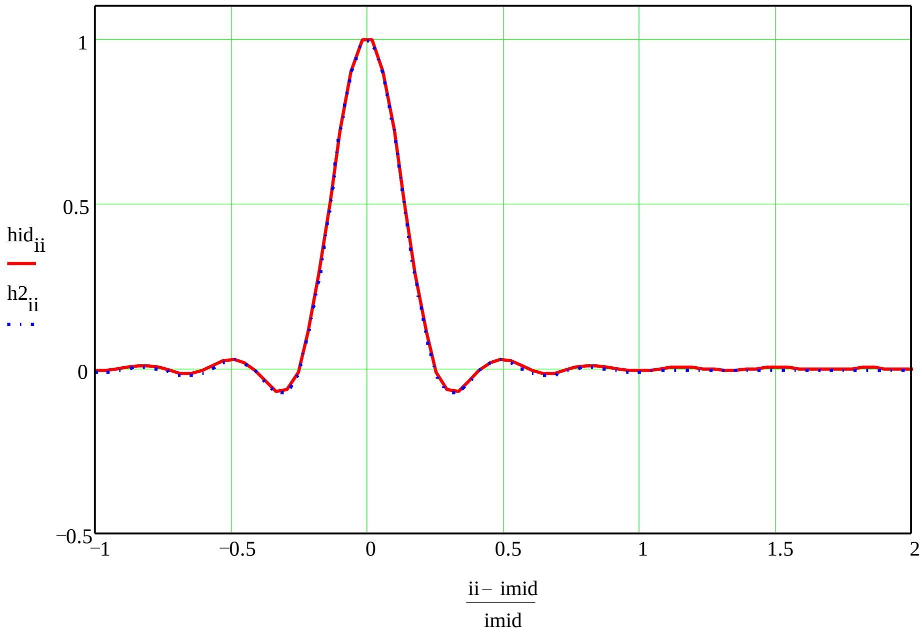
$H2 := CFFT(h2)$

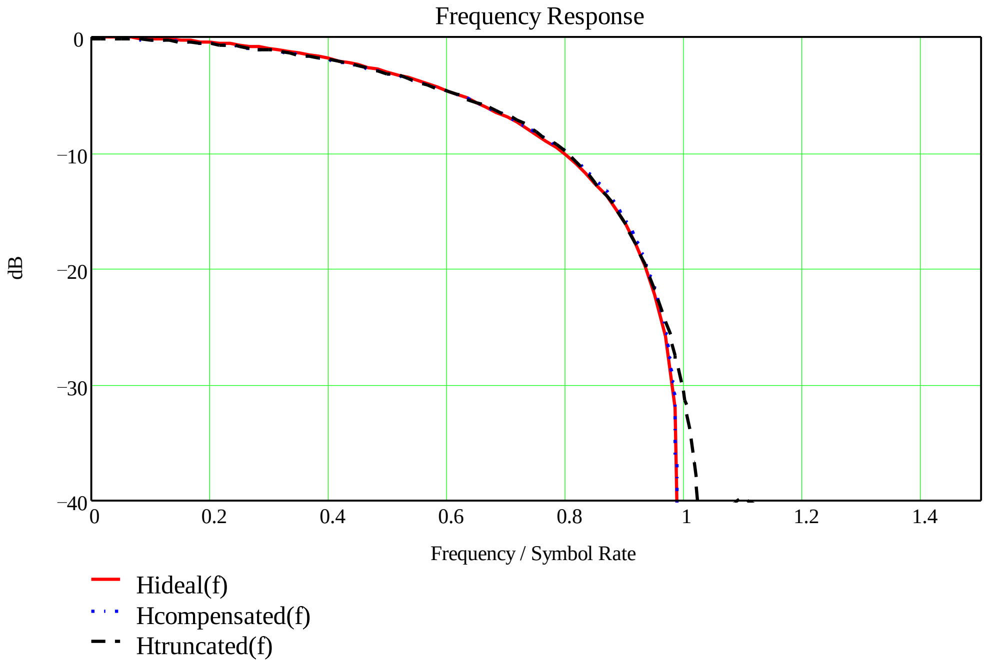
$hmax := \left|H2_0\right|$

Normalised and with -ve frequency after window.

$$H2_{ii} := H2_{ii} \cdot \frac{1}{hmax}$$

H2 now has the distorted (true) FIR response





Compute the frequency domain error due to time domain window:

$$E_{ii} := \mathrm{if}\left(\left|H1_{ii}\right| > 0, \left|\frac{H1_{ii}}{H2_{ii}}\right|, 1\right)$$

Pre-distorted response:

$$H4_{ii} := H_{ii} \cdot \mathrm{if}\left[\left(\left|E_{ii}\right| < 2\right) \cdot \left(\left|E_{ii}\right| > 0.5\right), \left(\left|E_{ii}\right|\right)^{2}, 1\right]$$

$$H4_{jj} := \overline{H4_{N-jj}}$$

Back through the time domain window:

$$h4 := \mathrm{ICFFT}(H4)$$

$$h4_{ii} := h4_{ii} \cdot w_{ii}$$

$hmax := vmax(h4)$

$h4_{ii} := \frac{h4_{ii}}{hmax}$

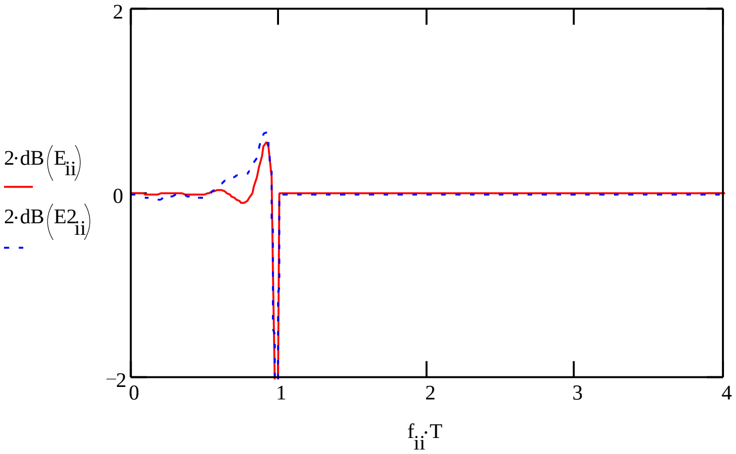
$H4 := CFFT(h4)$

$hmax := \left|H4_0\right|$

$H4_{ii} := H4_{ii} \cdot \frac{1}{hmax}$

Re-compute the error to see if improved:

$E2_{ii} := if\left(\left|H1_{ii}\right| > 0, \left|\frac{H1_{ii}}{H4_{ii}}\right|, 1\right)$



$H_{ii} := H_{ii} \cdot exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot \frac{-imid}{M} \cdot T\right)$

Center back on t=0, delay time = imid/M.

$H_{jj} := \overline{H_{N-jj}}$

Compute overall cascaded filter performance, first for the normal case (deconvolved, but not pre-distorted):

$Hover2_{ii} := H2_{ii} \cdot Hcic_{ii} \cdot Hpoly_{ii} \cdot Hif_{ii}$

The FIR filter, CIC, polyphase and IF bandpass filter.

$Hover2_{ii} := Hover2_{ii} \cdot exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot \frac{-imid}{M} \cdot T\right)$

Center back on t=0, delay time = imid/M.

$Hover2_{jj} := \overline{Hover2_{N-jj}}$

$S_{ii} := H_{ii} \cdot Hover2_{ii}$

Perfect RRC and non-predistorted designed filters in cascade.

$S_{jj} := \overline{S_{N-jj}}$

Copy conjugate for -ve spectrum

$s := ICFFT(S)$

In to time domain.

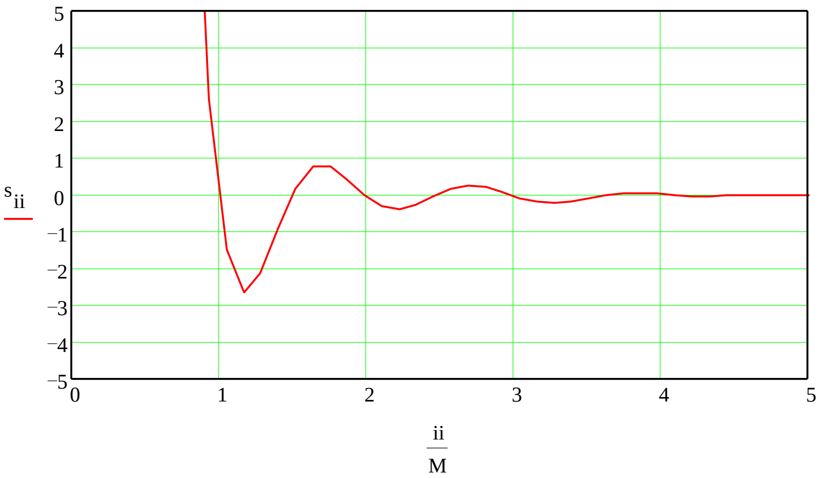
$smax := vmax(s)$

Normalise curve to maximum (on-time) value.

$s_{ii} := s_{ii} \cdot \frac{100}{smax}$

For a display of % "voltage" at ISI points.

Plot the response (expand to show residual error):



Compute the Root Sum Square error using cubic spline interpolation (simpler in MathCAD than Fourier approach). Assume even symmetry in summation:

$vx_{ii} := ii$

$vy := \mathrm{Re}(s)$

$vs := \mathrm{cspline}(vx, vy)$

$$un\_isi2 := \frac{2 \cdot \displaystyle\sum_{kk=1}^{\mathrm{floor}\left(\frac{N}{2 \cdot M}\right)} \mathrm{interp}(vs, vx, vy, kk \cdot M)^2}{\left(\left|s_0\right|\right)^2}$$

$un\_isi2 = 3.258 \cdot 10^{-7}$

$$un\_isi\_loss := \frac{1}{1 - en \cdot Kfec \cdot Rc \cdot un\_isi2}$$

$$NNmax := \mathrm{floor}\left(\frac{N-1}{2}\right)$$

$nn := 0 .. NNmax$

Next compute the loss due to frequency domain errors (with respect to the ideal "matched filter" performance):

$$un\_snr\_loss := \frac{\left[\left|\displaystyle\sum_{nn}\left(H_{nn} \cdot H_{nn}\right)\right|\right]^2}{\displaystyle\sum_{nn}\left(\left|H_{nn}\right|\right)^2} \cdot \frac{\displaystyle\sum_{nn}\left(\left|Hover2_{nn}\right|\right)^2}{\left[\left|\displaystyle\sum_{nn}\left(H_{nn} \cdot Hover2_{nn}\right)\right|\right]^2}$$

$un\_snr\_loss = 1$

$un\_dB\_loss := \mathrm{dB}(un\_isi\_loss) + \mathrm{dB}(un\_snr\_loss)$

$un\_dB\_loss = 2.866 \cdot 10^{-4}$

Same again for pre-distorted response:

$Hover4_{ii} := H4_{ii} \cdot Hcic_{ii} \cdot Hpoly_{ii} \cdot Hif_{ii}$

$$Hover4_{ii} := Hover4_{ii} \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot f_{ii} \cdot \frac{-imid}{M} \cdot T\right)$$

Center back on t=0, delay time = imid/M.

$Hover4_{jj} := \overline{Hover4_{N-jj}}$

$S_{ii} := H_{ii} \cdot Hover4_{ii}$

Perfect RRC and designed filters in cascade.
Copy conjugate for -ve spectrum

$S_{jj} := \overline{S_{N-jj}}$

In to time domain.

$s := ICFFT(S)$

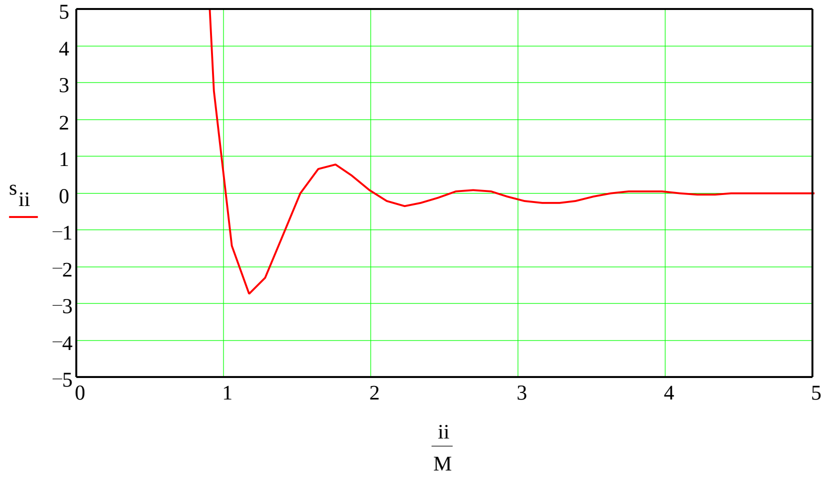
Normalise curve to maximum (on-time) value.

$smax := \left|s_0\right|$

For a display of % "voltage".

$s_{ii} := s_{ii} \cdot \frac{100}{smax}$

Plot the response:



$vx_{ii} := ii$

$vy := Re(s)$

$vs := cspline(vx, vy)$

$$pre\_isi2 := \frac{2 \cdot \sum_{kk=1}^{floor\left(\frac{N}{2 \cdot M}\right)} interp(vs, vx, vy, kk \cdot M)^2}{\left(\left|s_0\right|\right)^2}$$

$pre\_isi2 = 8.004 \cdot 10^{-6}$

$\sqrt{pre\_isi2} = 2.829 \cdot 10^{-3}$

$$pre\_isi\_loss := \frac{1}{1 - en \cdot Kfec \cdot Rc \cdot pre\_isi2}$$

$$pre\_snr\_loss := \frac{\left[\left|\sum_{nn}\left(H_{nn} \cdot H_{nn}\right)\right|\right]^2}{\sum_{nn}\left(\left|H_{nn}\right|\right)^2} \cdot \frac{\sum_{nn}\left(\left|Hover4_{nn}\right|\right)^2}{\left[\left|\sum_{nn}\left(H_{nn} \cdot Hover4_{nn}\right)\right|\right]^2}$$

$pre\_snr\_loss = 1$

$pre\_dB\_loss := dB(pre\_isi\_loss) + dB(pre\_snr\_loss)$

$pre\_dB\_loss = 6.365 \cdot 10^{-4}$

Compare the two approaches to chose best method:

$dB(un\_snr\_loss) = 2.843 \cdot 10^{-4}$

$dB(un\_isi\_loss) = 2.348 \cdot 10^{-6}$

$un\_dB\_loss = 2.866 \cdot 10^{-4}$

$dB(pre\_snr\_loss) = 5.788 \cdot 10^{-4}$

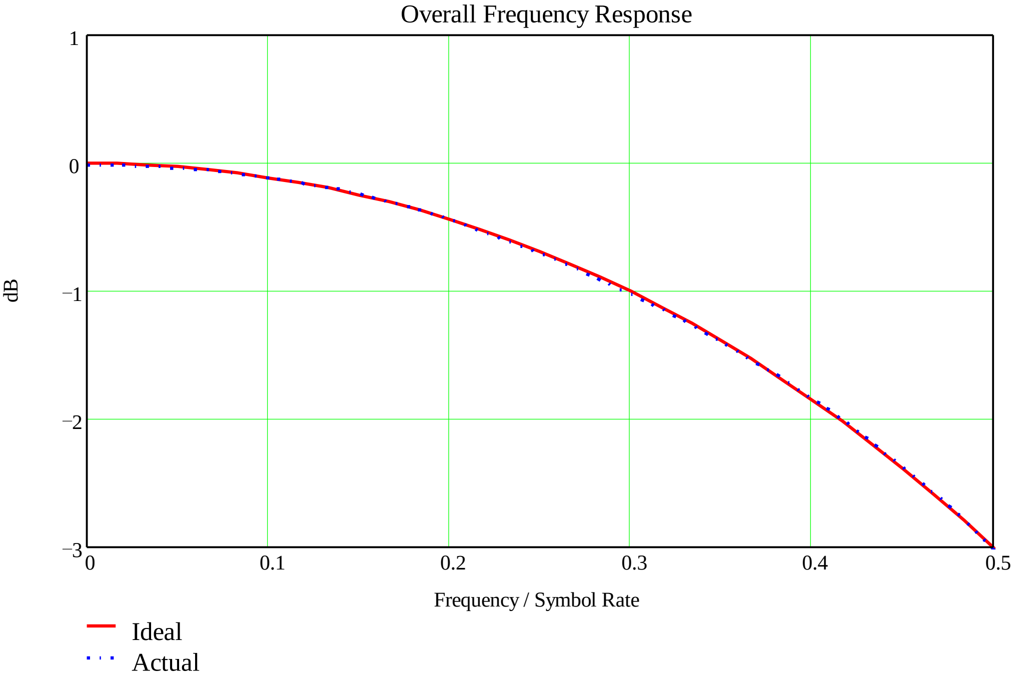
$dB(pre\_isi\_loss) = 5.769 \cdot 10^{-5}$

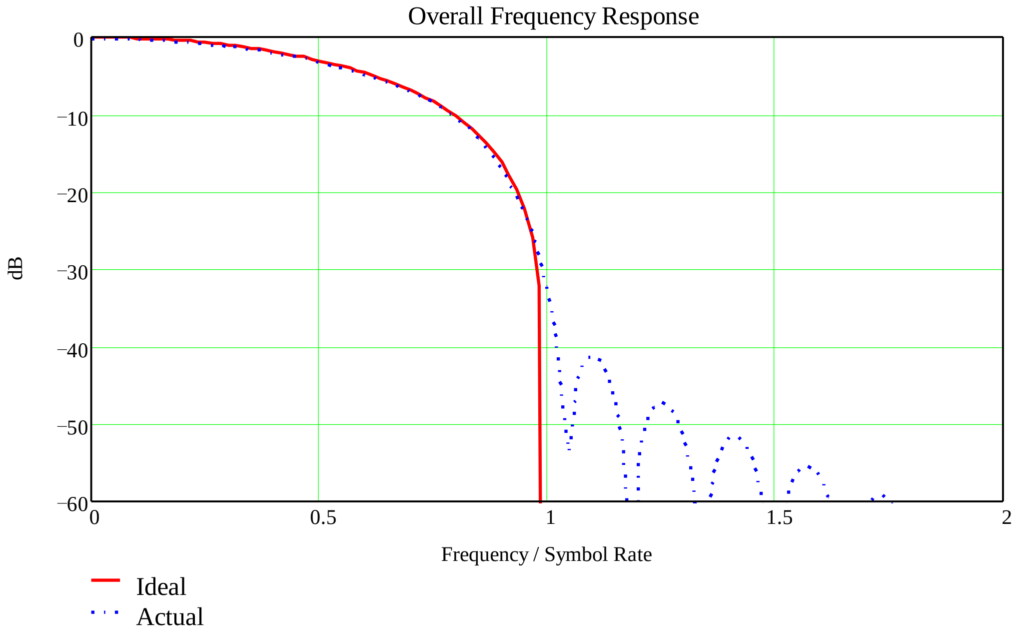
$pre\_dB\_loss = 6.365 \cdot 10^{-4}$

Pick the best overall performer: 2 = un_ 4 = pre_

$Husted := Hover2$

$h\_used := h2$





Finally scale to HSP50214 filter word size (integer) and write to file.

$\text{nint}(z) := \text{if}(\text{Re}(z)<0, \text{ceil}(\text{Re}(z) - 0.5), \text{floor}(\text{Re}(z) + 0.5)) + j\cdot\text{if}(\text{Im}(z)<0, \text{ceil}(\text{Im}(z) - 0.5), \text{floor}(\text{Im}(z) + 0.5))$

$cc := 0 .. Ntap - 1$

22bit coefficients

$h\_op_{cc} := \text{Re}\left[\text{nint}\left[h\_used_{cc}\cdot\left(2^{21} - 1\right)\right]\right]$

h_used =

	0
0	$-7.858\cdot10^{-3}$
1	$-6.062\cdot10^{-3}$
2	$-5.394\cdot10^{-4}$
3	$6.302\cdot10^{-3}$
4	0.011
5	0.01
6	$3.521\cdot10^{-3}$
7	$-6.688\cdot10^{-3}$
8	-0.015
9	-0.017
10	$-9.699\cdot10^{-3}$
11	$6.046\cdot10^{-3}$
12	0.023

PRNPRECISION := 8

Set up for nice file output

PRNCOLWIDTH := 8

WRITEPRN(lrit_fir) := h_op

$pre\_dB\_loss = 6.365\cdot10^{-4}$

$un\_dB\_loss = 2.866\cdot10^{-4}$

$Ntap = 52$

--End of File--

File "viterbi2.mcd"
Created with MathCAD 6.0-PLUS
Visulisation and further processing of simulated Viterbi decoder output file. This version for the R=1/2 decoder option (as used in the normal CCSDS case).

$dB(x) := if\left(|x| < 10^{-12}, -120, 10 \cdot \log(|x|)\right)$

Convert to dB
And back to numeric

$idB(x) := 10^{\frac{x}{10}}$

$vtxt := READPRN(vitres)$

File is normally "vitres.prn" but the .prn is assumed by MathCAD.

$Nvt := rows(vtxt)$

$Nvt = 189$

Data in viterbi.prn file is line-by-line with each line containing:
0) Eb/No (dB)
1) Number of bits simulated
2) Bit error rate (Pe), numeric.
3) Word error rate, numeric
4) Bit errors per word
5) Phase error, degrees.
6) Interleaved (1=yes/0=no)
These run 5 deg increments (-10, -5, 0, 5, 10), alternating between non-interleaved and interleaved (except for the 0 deg case where interleaved & non are identical). Total is 5 sets of non-interleaved
and 4 interleaved data results then the Eb/No value increments by 0.25dB, thus there are 9 rows per Eb/No value.

$Nump := 9$

$Ntrials := floor\left(\frac{Nvt}{Nump}\right)$

Floor to truncate partial file.

$Ntrials = 21$

Number of Eb/No entries.

$ii := 0 .. Nump - 1$

Index each simulated case.

$jj := 0 .. Ntrials - 1$

Index each Eb/No value.

$safe\_log(x) := if\left(|x| < 10^{-14}, -14, \log(x)\right)$

$$
\begin{aligned}
vfunc(i) := \;& eqEB_0 \leftarrow 0 \\
& \text{for } j \in 0 .. Ntrials - 1 \\
& \quad eqEB_{1+j} \leftarrow idB\left(vtxt_{i+j \cdot Nump, 0}\right) \\
& logPE_0 \leftarrow -\log(0.5) \\
& \text{for } j \in 0 .. Ntrials - 1 \\
& \quad logPE_{1+j} \leftarrow -safe\_log\left(vtxt_{i+j \cdot Nump, 2}\right) \\
& cspline(eqEB, logPE)
\end{aligned}
$$

MathCAD wont allow use of matrix as array
of vectors, hence must use proceedure to
unpack in to vectors to do an array of functions.

$vs_{ii} := vfunc(ii)$

Do all splines in one go, save for next function/proceedure.

$$\text{pefg}(i,v) := \left| \begin{array}{l} eqEB_0 \leftarrow 0 \\ \text{for } j \in 0..Ntrials-1 \\ \quad eqEB_{1+j} \leftarrow idB\left(vtxt_{i+j\cdot Nump,0}\right) \\ logPE_0 \leftarrow -\log(0.5) \\ \text{for } j \in 0..Ntrials-1 \\ \quad logPE_{1+j} \leftarrow -safe\_log\left(vtxt_{i+j\cdot Nump,2}\right) \\ 0.5 \text{ if } v<0 \text{ otherwise} \\ 10^{-logPE_{Ntrials}} \text{ if } v>eqEB_{Ntrials} \text{ otherwise} \\ 10^{-interp\left(vs_i, eqEB, logPE, v\right)} \end{array} \right.$$

Again unpack into local vectors, then apply the interpolation on the logaritmic data (after chacking it is within a reasonable range).
Zero phase error case corresponds to index=4 case.

$\text{pef}(v) := \text{pefg}(4,v)$

Define the function we wish to model the QPSK degradation:

$Rc := \frac{1}{2}$

Code rate (i.e. bit rate/symbol rate)

$$\text{peQPSKf2}(en,\theta,K1) := \text{pef}\left[\frac{\cos(\theta)^2}{\left(\frac{1}{en}\right) + K1\cdot Rc\cdot \sin(\theta)^2}\right]$$

Then use root-finding to solve for the Kfec term (guess = K1 in above).

$guess := 1.5$

$$\text{efunc}(en,i,guess) := \log(\text{pefg}(i,en)) - \log\left(\text{peQPSKf2}\left(en, vtxt_{i,5}\cdot\frac{\pi}{180}, guess\right)\right)$$

Must exclude the zero phase error case (i.e. pef() itself) otherwise no convergence.

$\text{Kfunc}(en,i) := \text{if}(i = 4, 0, \text{root}(\text{efunc}(en,i,guess), guess))$

Set up and plot for all of the simulated data points.

$kkmax := Ntrials - 1$

1=normal, >1 to stop upper Eb/No values?

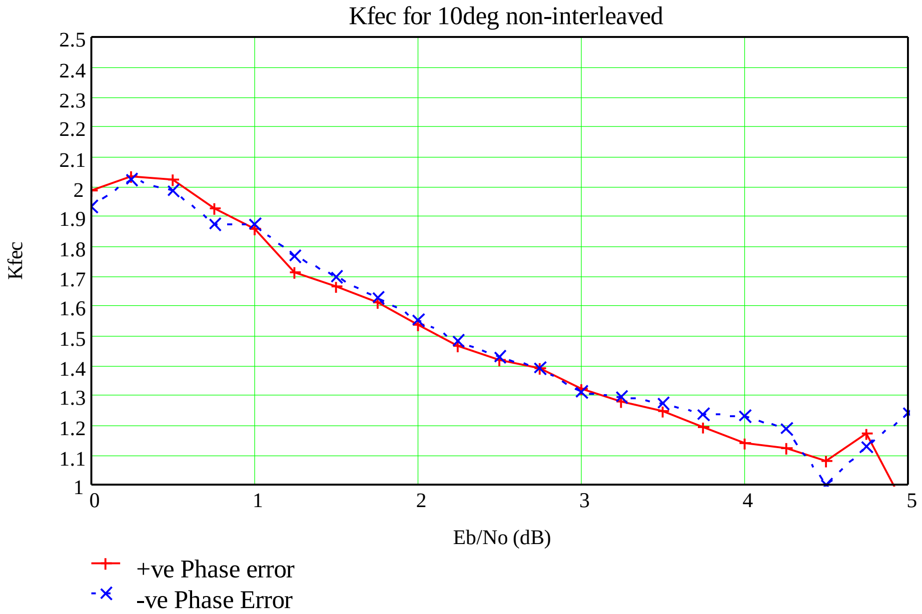
$kk := 0..kkmax$

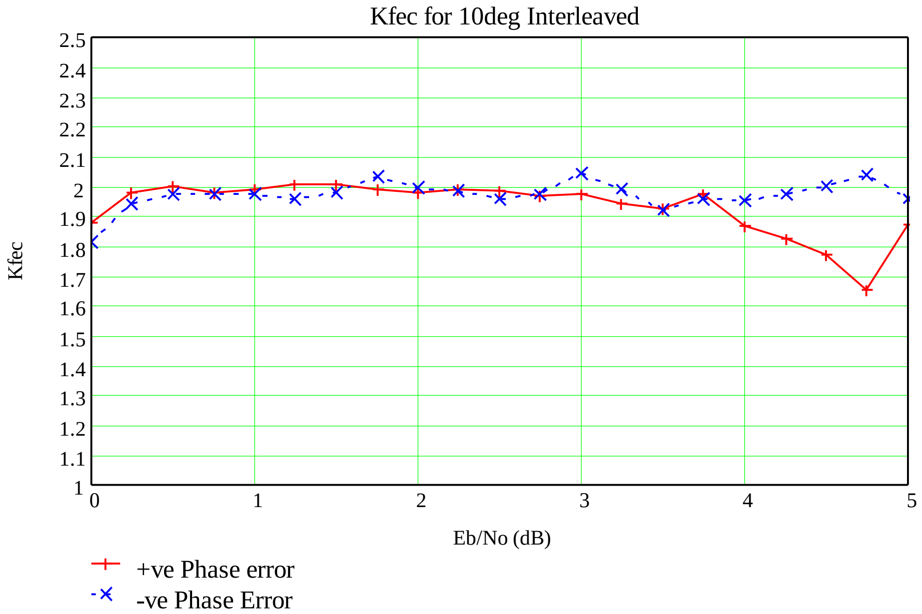
The ii index runs through the following cases:
0) -10deg, non-interleaved
1) -10deg, interleaved
2) -5deg, non
3) -5deg, int
4) 0deg
5) 5deg, non
6) 5deg, int
7) 10deg, non
8) 10deg, int

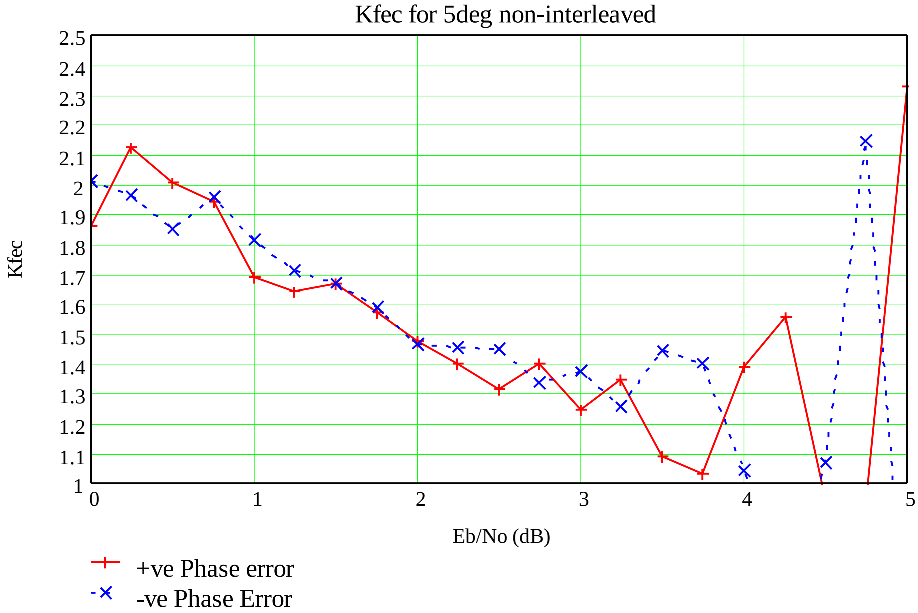
$en_{kk} := idB\left(vtxt_{kk\cdot Nump,0}\right)$

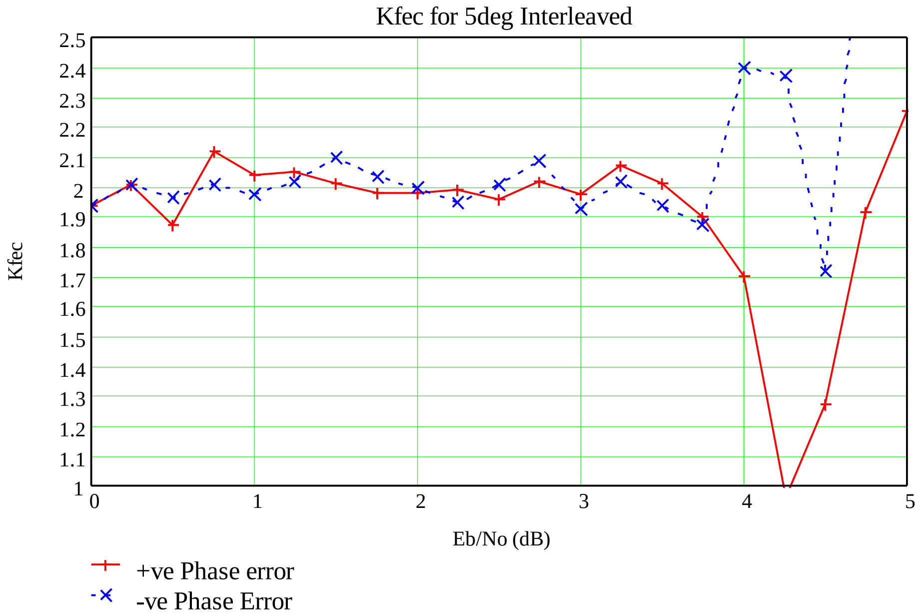
$dB\left(en_{kkmax}\right) = 5$

$Kfec_{ii,kk} := \text{Kfunc}\left(en_{kk}, ii\right)$


Kfec for 10deg non-interleaved
Kfec
2.5
2.4
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0
1
2
3
4
5
Eb/No (dB)
+ve Phase error
-ve Phase Error


Kfec for 10deg Interleaved
Kfec
2.5
2.4
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0
1
2
3
4
5
Eb/No (dB)
+ve Phase error
-ve Phase Error





Compare the Viterbi decoder bit-errors/word-errors to the "uncoded case" (all bits independant) operating at the same AVERAGE ERROR RATE as the Viterbi decoder.

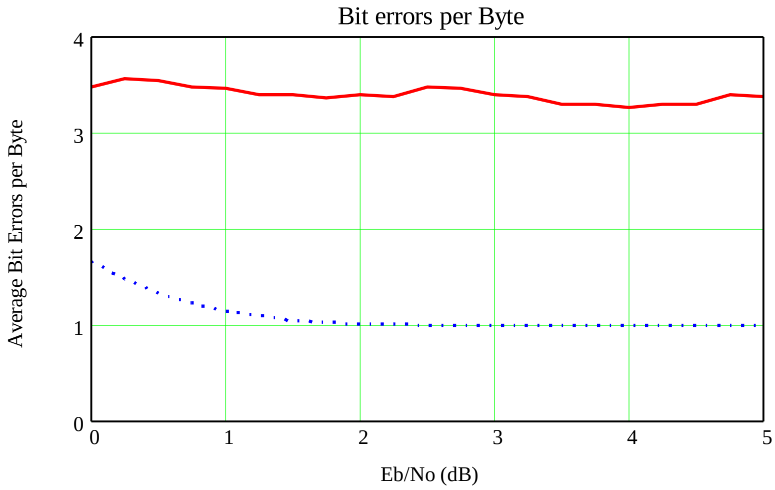
$idx := 4$

$en_{jj} := idB\left(vtxt_{jj \cdot Nump + idx, 0}\right)$

"Average bit errors per byte" from sum of probable errors compared to the probability there is any errors.

$$nerr(p) := \frac{\sum_{k=1}^{8} k \cdot dbinom(k, 8, p)}{1 - dbinom(0, 8, p)}$$

$nrandom_{jj} := nerr\left(vtxt_{jj \cdot Nump + idx, 2}\right)$



Now set up interpolation functions for word error rate and bits/word to allow CCSDS system modeling. First entry is the limiting case of Eb/No -> 0 where PE=0.5, so find probability of a word error (i.e. any bits wrong) and set up limitng values.

$logPW_0 := -\log(1 - dbinom(0, 8, 0.5))$

$eqEB_0 := 0$

$bit\_per\_word_0 := 0.5$

Now read/convert the simulated data (idx = 4 = zero phase error):

$idx := 4$

$logPW_{1+jj} := -safe\_log\left(vtxt_{jj \cdot Nump + idx, 3}\right)$

$eqEB_{1+jj} := idB\left(vtxt_{jj \cdot Nump + idx, 0}\right)$

$bit\_per\_word_{1+jj} := \dfrac{vtxt_{jj \cdot Nump + idx, 4}}{8}$

logPW =

	0
0	$1.7 \cdot 10^{-3}$
1	0.483
2	0.593
3	0.721
4	0.869
5	1.035
6	1.22
7	1.425
8	1.643
9	1.88
10	2.132
11	2.399
12	2.679
13	2.969
14	3.277

eqEB =

	0
0	0
1	1
2	1.059
3	1.122
4	1.189
5	1.259
6	1.334
7	1.413
8	1.496
9	1.585
10	1.679
11	1.778
12	1.884
13	1.995
14	2.113
15	2.239

bit_per_word =

	0
0	0.5
1	0.47
2	0.461
3	0.453
4	0.442
5	0.432
6	0.42
7	0.408
8	0.396
9	0.384
10	0.371
11	0.359
12	0.347
13	0.337
14	0.327
15	0.321

$vsw := cspline(eqEB, logPW)$

$vsw2 := cspline(eqEB, bit\_per\_word)$

$$nCCSDS(en) := \left(\sum_{k=17}^{255} \frac{k}{255} \cdot dbinom\left(k, 255, 10^{-interp(vsw, eqEB, logPW, en)}\right) \right) \cdot interp(vsw2, eqEB, bit\_per\_word, en)$$

$pec(v) := if\left(v<0, 0.5, if\left(v>1.71, 1.796\cdot 10^{-14}, nCCSDS(v)\right)\right)$

Plot these function out for a typical range of Eb/No:

$jmax := 30$

$jj := 0 .. jmax$

$minE := -1$

$maxE := 3.2$

$eb_{jj} := idB\left[minE + \frac{jj}{jmax}\cdot(maxE - minE)\right]$

Compute the frame failure rate from probabillity that one or more of the 4 interleaved frames has greater than 16 errors in the 255 bytes involved.

$pFrameError(en) := 1 - pbinom\left(16, 255, 10^{-interp(vsw, eqEB, logPW, en)}\right)^4$

$pframe(en) := if\left(en<1, 1, if\left(en>1.77, 7.994\cdot 10^{-15}, pFrameError(en)\right)\right)$

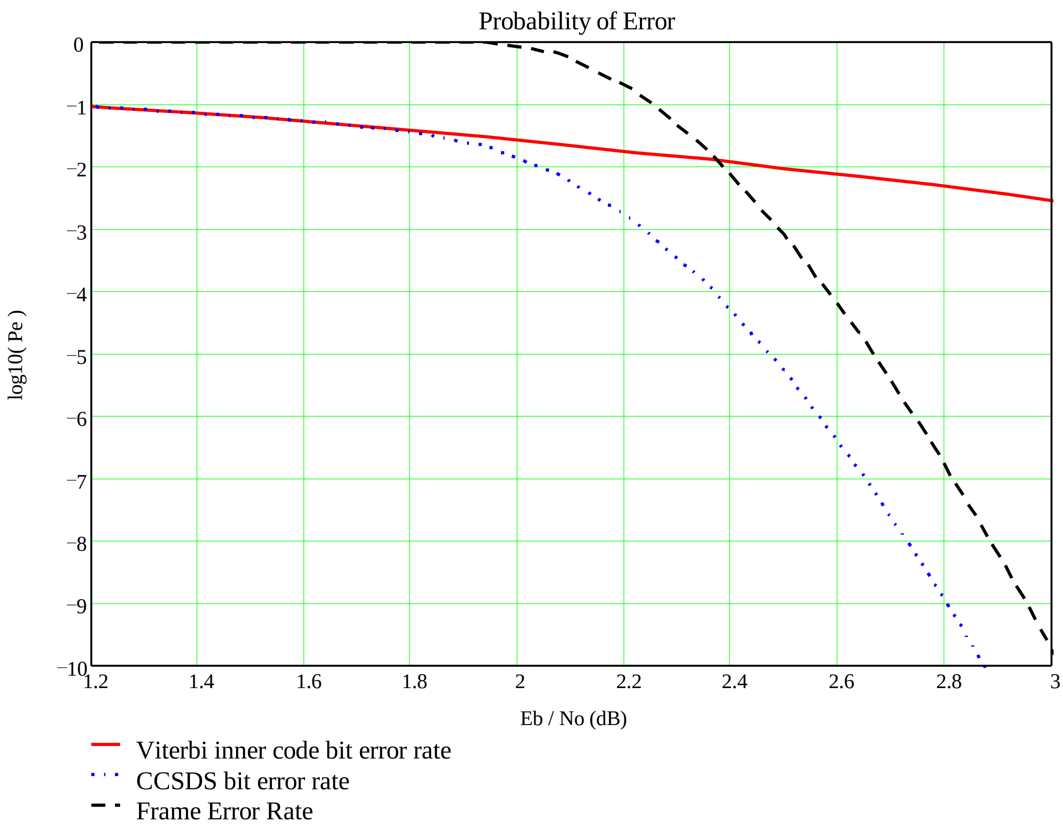
Prepare to plot results. The 223/255 corrects for the R/S overhead. The -0.2dB corrects for 8-bit rather than 3-bit decisions.

$pV_{jj} := log\left(pef\left(eb_{jj}\cdot\frac{223}{255}\cdot idB(-0.2)\right)\right)$

$pCCSDS_{jj} := log\left(pec\left(eb_{jj}\cdot\frac{223}{255}\cdot idB(-0.2)\right)\right)$

$pF_{jj} := log\left(pframe\left(eb_{jj}\cdot\frac{223}{255}\cdot idB(-0.2)\right)\right)$

$dbEB_{jj} := dB\left(eb_{jj}\right)$



$save_{0,0} := 0$

$save_{0,1} := log(0.5)$

$save_{jj+1,0} := eb_{jj}$

$save_{jj+1,1} := pCCSDS_{jj}$

$PRNPRECISION := 8$

$PRNCOLWIDTH := 15$

$WRITEPRN(ccsds\_lut) := save$

Function that uses the simulated (and saved above) Viterbi decoder results to model the CCSDS error rate versus Eb/No more accurately.

$ccsds := READPRN(ccsds\_lut)$

$Nccsds := rows(ccsds)$

$Nccsds = 32$

$icc := 0 .. Nccsds - 1$

$cc\_eb_{icc} := ccsds_{icc,0}$

$cc\_logPE_{icc} := ccsds_{icc,1}$

$cc\_vs := cspline(cc\_eb, cc\_logPE)$

$cc\_ifunc(v) := 10^{interp(cc\_vs, cc\_eb, cc\_logPE, v)}$

$vMax := ccsds_{Nccsds - 1, 0}$

$ErrMin := 10^{ccsds_{Nccsds - 1, 1}}$

$pe\_ccsds(v) := if(v<0, 0.5, if(v>vMax, ErrMin, cc\_ifunc(v)))$

--End of File--

File ZCD_SIM_BPSK.MCD to analyse BPSK (LRIT) case Zero Crossing Detector simulations.

$dB(x) := \text{if}\left(|x| < 10^{-12}, -120, 10 \cdot \log(|x|)\right)$

$h1 := READPRN(bk8)$

$h2 := READPRN(bk0)$

$Nhp := rows(h1)$

$Nhp = 33$

$ii := 0 .. Nhp - 1$

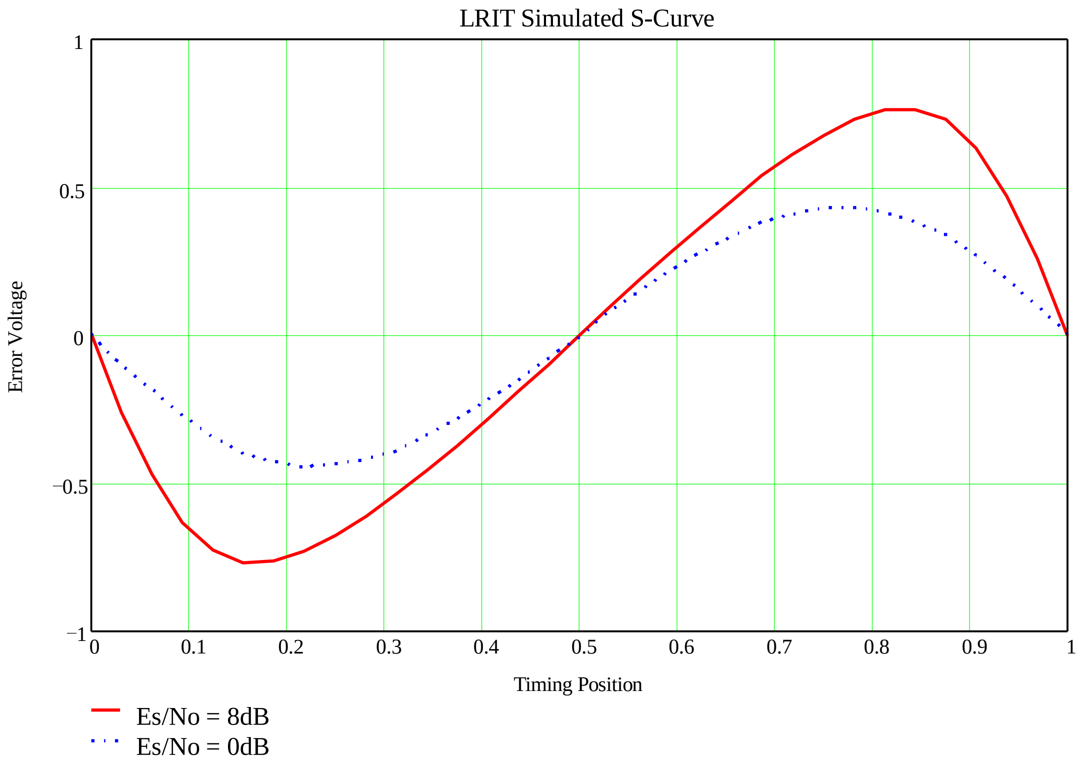
$d := 1$

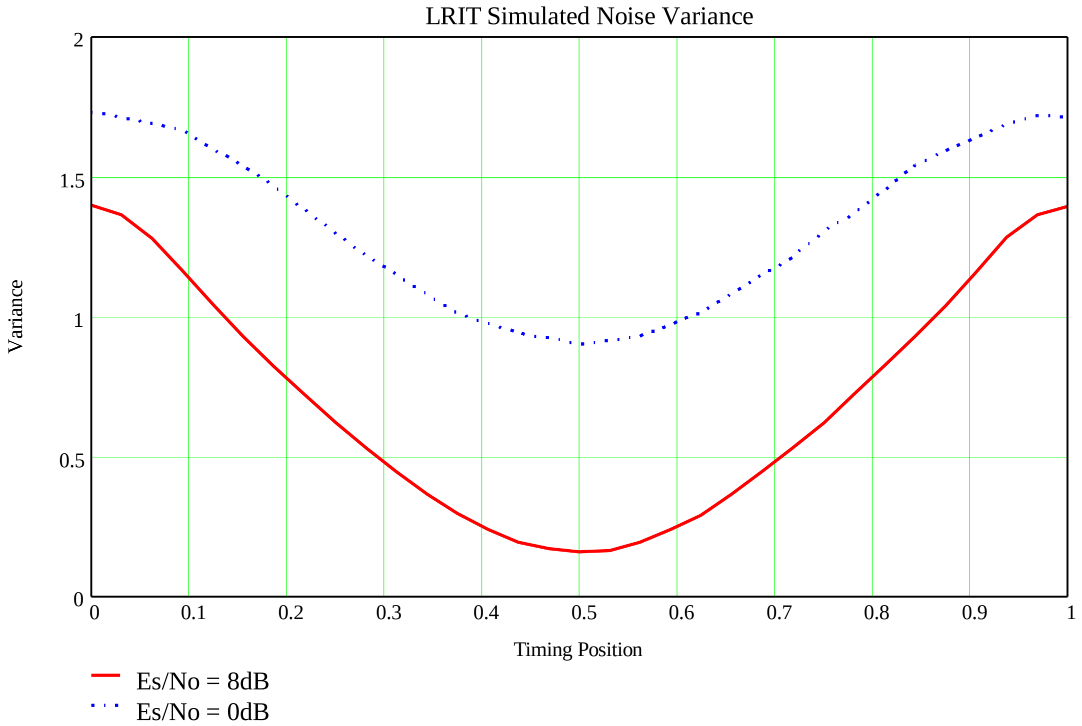
$$slope1 := \frac{h1_{\frac{Nhp-1}{2} - d, 1} - h1_{\frac{Nhp-1}{2} + d, 1}}{h1_{\frac{Nhp-1}{2} - d, 0} - h1_{\frac{Nhp-1}{2} + d, 0}}$$

$$slope2 := \frac{h2_{\frac{Nhp-1}{2} - d, 1} - h2_{\frac{Nhp-1}{2} + d, 1}}{h2_{\frac{Nhp-1}{2} - d, 0} - h2_{\frac{Nhp-1}{2} + d, 0}}$$

$slope1 = 3.065$

$slope2 = 2.514$





$ii := d .. Nhp - 1 - d$

$h1_{16,2} = 0.161$

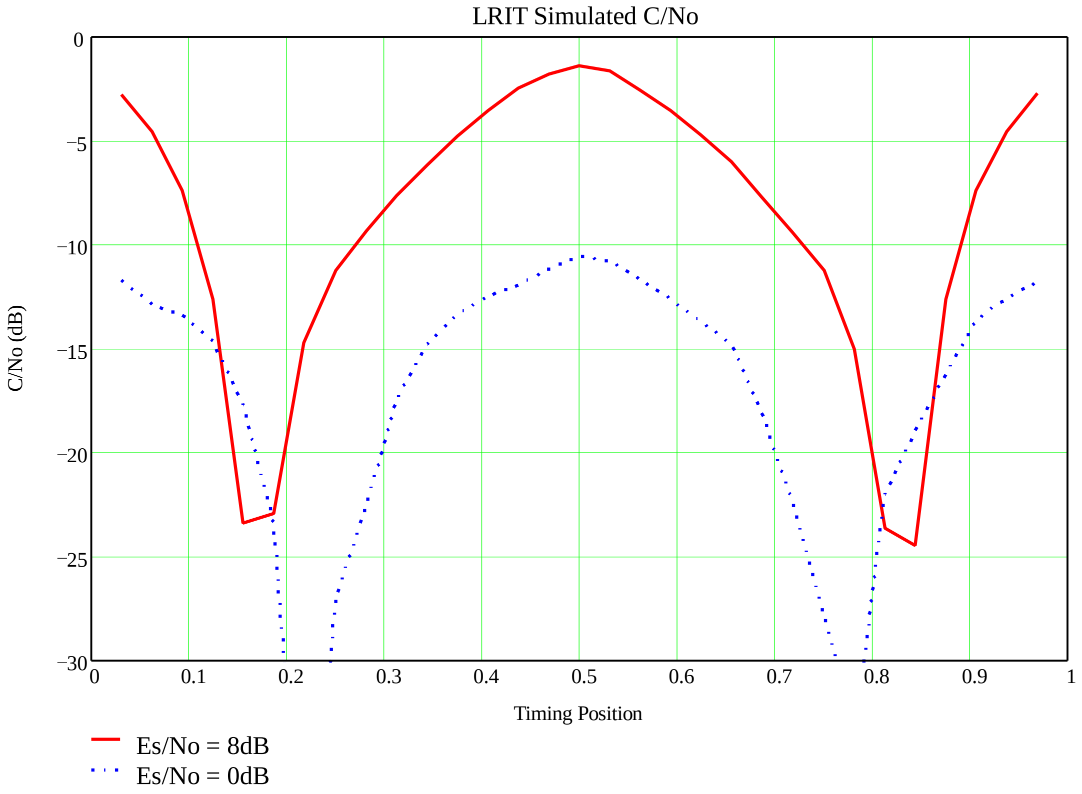
$h2_{16,2} = 0.904$

$K := 1$

Use 1 for BPSK results to get per symbol

$$h3_{ii,2} := \frac{\left(\frac{h1_{ii-d,1} - h1_{ii+d,1}}{h1_{ii-d,0} - h1_{ii+d,0}}\right)^2}{2 \cdot K \cdot \left(0.8 \cdot h1_{ii,2} + 0.1 \cdot h1_{ii-d,2} + 0.1 \cdot h1_{ii+d,2}\right) \cdot 4 \cdot \pi^2}$$

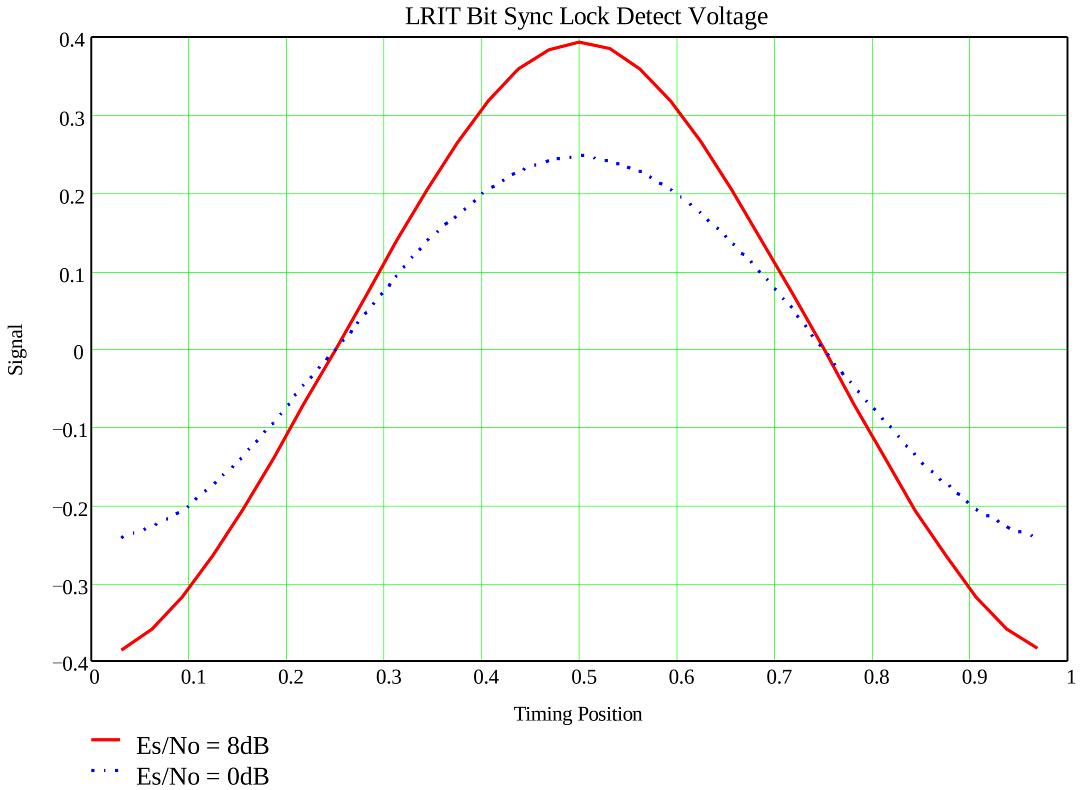
$$h4_{ii,2} := \frac{\left(\frac{h2_{ii-d,1} - h2_{ii+d,1}}{h2_{ii-d,0} - h2_{ii+d,0}}\right)^2}{2 \cdot K \cdot \left(0.8 \cdot h2_{ii,2} + 0.1 \cdot h2_{ii-d,2} + 0.1 \cdot h2_{ii+d,2}\right) \cdot 4 \cdot \pi^2}$$

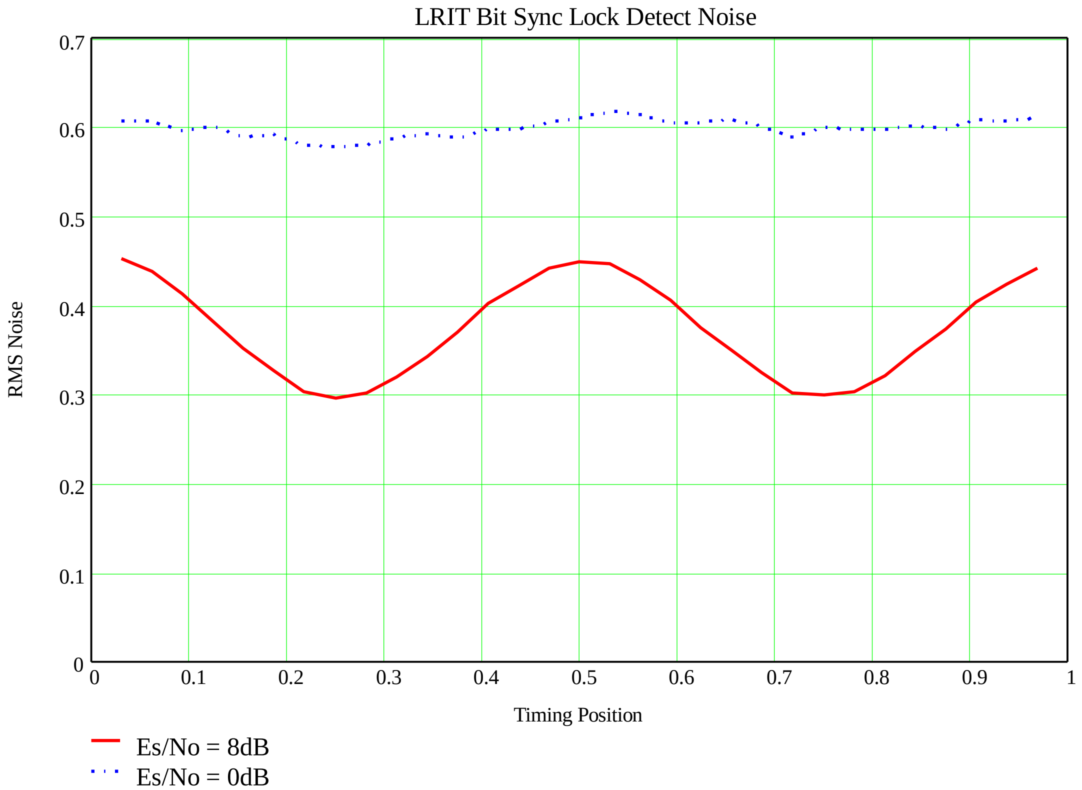


Array index values are:
0=timing
1=PSD voltage
2=PSD variance
3=Symbol Perror
4=Technology loss on Pe(sym) (dB)
5=LD voltage
6=LD variance

h2 =

	0	1	2	3	4	5	6
0	0	0.00371	1.73308	0.2919	8.23903	-0.24896	0.37681
1	0.03125	-0.10439	1.71151	0.2646	7.03414	-0.24265	0.36803
2	0.0625	-0.18509	1.68905	0.2404	6.04671	-0.22748	0.3688
3	0.09375	-0.26854	1.66843	0.2151	5.07042	-0.20685	0.35497
4	0.125	-0.33897	1.59858	0.192	4.2126	-0.17369	0.3612
5	0.15625	-0.3994	1.54072	0.1691	3.38474	-0.13656	0.34709
6	0.1875	-0.42869	1.4708	0.1518	2.76487	-0.09367	0.35016
7	0.21875	-0.44481	1.38352	0.1354	2.17425	-0.04604	0.33576
8	0.25	-0.43324	1.29534	0.1225	1.70332	$9.7\cdot10^{-5}$	0.33489
9	0.28125	-0.41623	1.21581	0.1098	1.22884	0.0474	0.3365
10	0.3125	-0.38753	1.15083	0.1004	0.86986	0.09403	0.34663
11	0.34375	-0.3356	1.08139	0.09456	0.64377	0.13897	0.34982
12	0.375	-0.28179	1.01188	0.08858	0.40651	0.17183	0.34514
13	0.40625	-0.21504	0.97723	0.08426	0.23178	0.20441	0.35659
14	0.4375	-0.15059	0.94085	0.0816	0.12262	0.22715	0.35705


LRIT Bit Sync Lock Detect Voltage
Signal
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Timing Position
Es/No = 8dB
Es/No = 0dB


LRIT Bit Sync Lock Detect Noise
RMS Noise
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Timing Position
Es/No = 8dB
Es/No = 0dB

File ZCD_SIM_QPSK.MCD to analyse QPSK (HRIT) case Zero Crossing Detector simulations.

$dB(x) := if\left(|x| < 10^{-12}, -120, 10 \cdot \log(|x|)\right)$

$h1 := READPRN(gk8)$

$h2 := READPRN(gk0)$

$Nhp := rows(h1)$

$Nhp = 33$

$ii := 0 .. Nhp - 1$

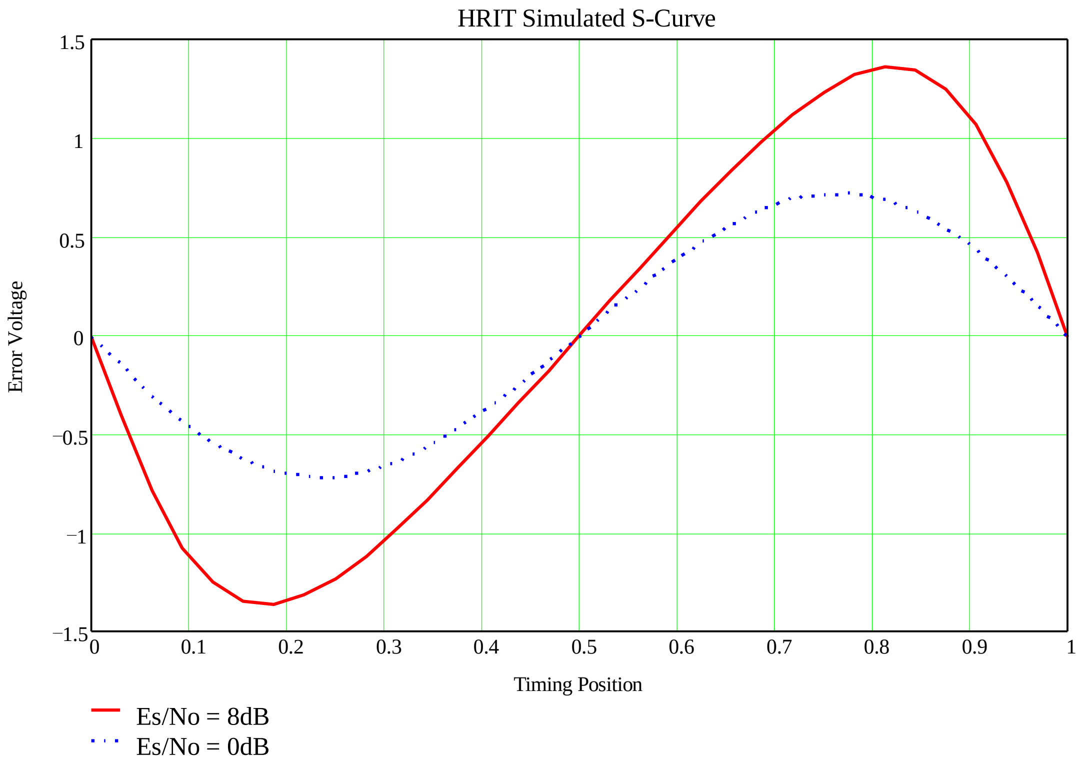
$d := 1$

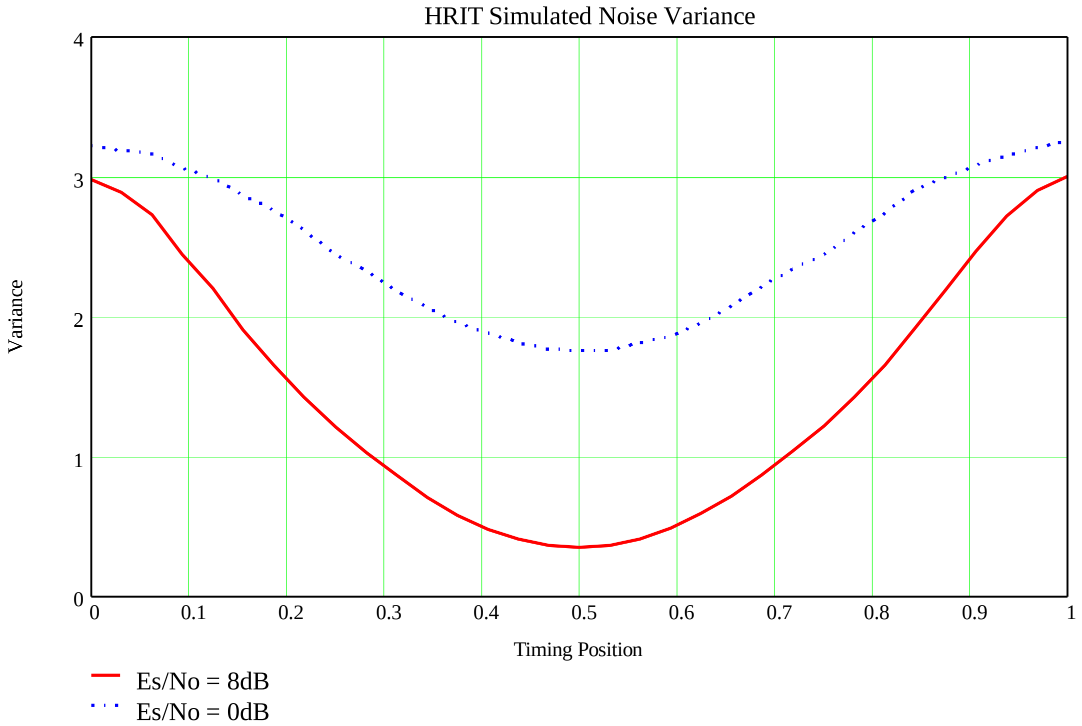
$$slope1 := \frac{h1_{\frac{Nhp-1}{2}-d,1} - h1_{\frac{Nhp-1}{2}+d,1}}{h1_{\frac{Nhp-1}{2}-d,0} - h1_{\frac{Nhp-1}{2}+d,0}}$$

$$slope2 := \frac{h2_{\frac{Nhp-1}{2}-d,1} - h2_{\frac{Nhp-1}{2}+d,1}}{h2_{\frac{Nhp-1}{2}-d,0} - h2_{\frac{Nhp-1}{2}+d,0}}$$

$slope1 = 5.634$

$slope2 = 4.197$





$ii := d .. Nhp - 1 - d$

$h1_{16,2} = 0.35$

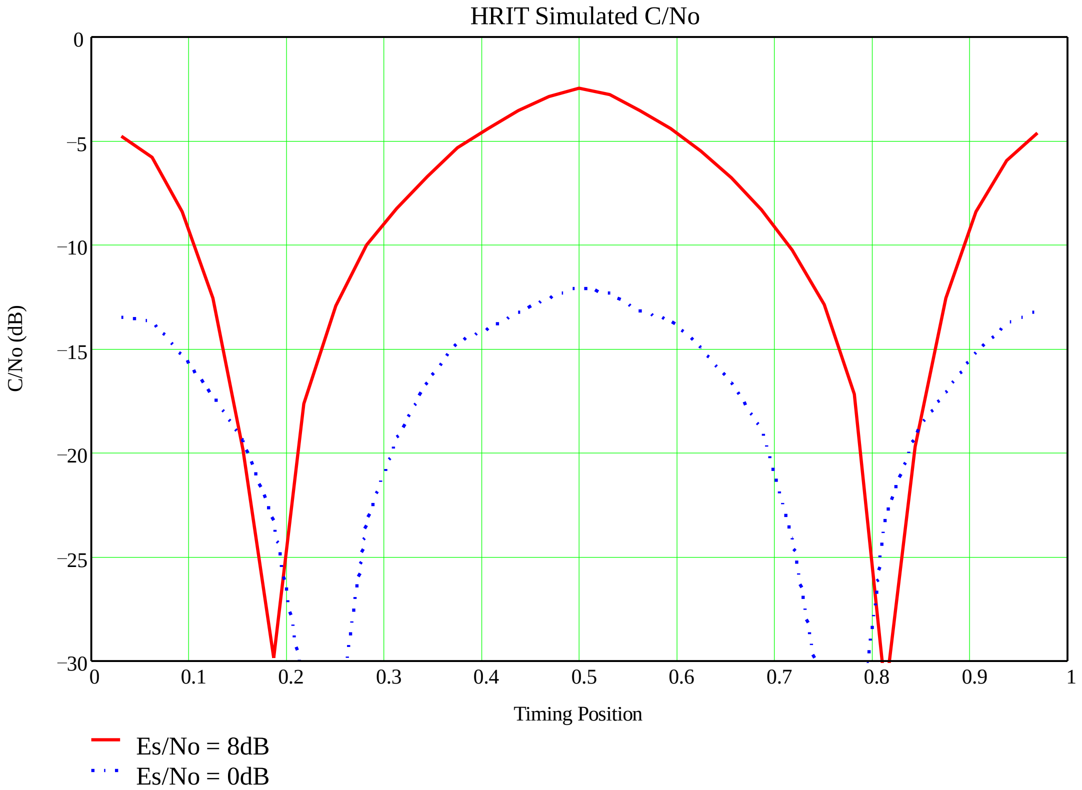
$h2_{16,2} = 1.763$

$K := 2$

Use 2 for QPSK results to get per symbol

$$h3_{ii,2} := \frac{\left(\frac{h1_{ii-d,1} - h1_{ii+d,1}}{h1_{ii-d,0} - h1_{ii+d,0}}\right)^2}{2 \cdot K \cdot \left(0.8 \cdot h1_{ii,2} + 0.1 \cdot h1_{ii-d,2} + 0.1 \cdot h1_{ii+d,2}\right) \cdot 4 \cdot \pi^2}$$

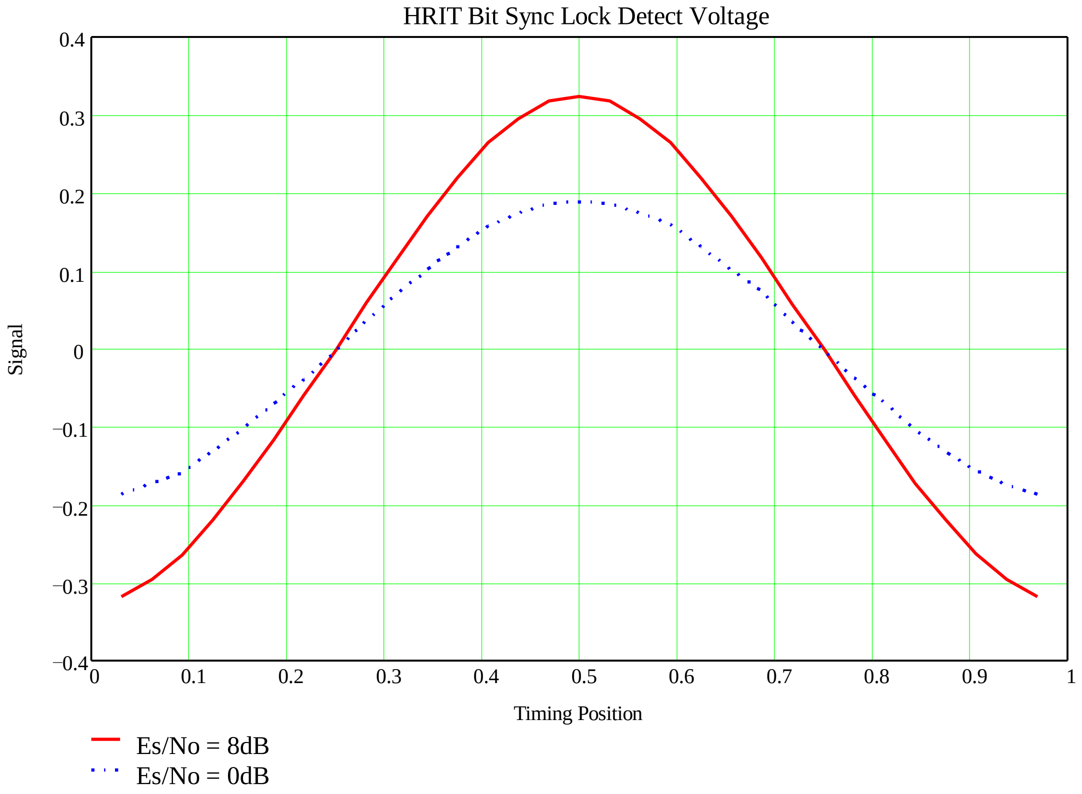
$$h4_{ii,2} := \frac{\left(\frac{h2_{ii-d,1} - h2_{ii+d,1}}{h2_{ii-d,0} - h2_{ii+d,0}}\right)^2}{2 \cdot K \cdot \left(0.8 \cdot h2_{ii,2} + 0.1 \cdot h2_{ii-d,2} + 0.1 \cdot h2_{ii+d,2}\right) \cdot 4 \cdot \pi^2}$$

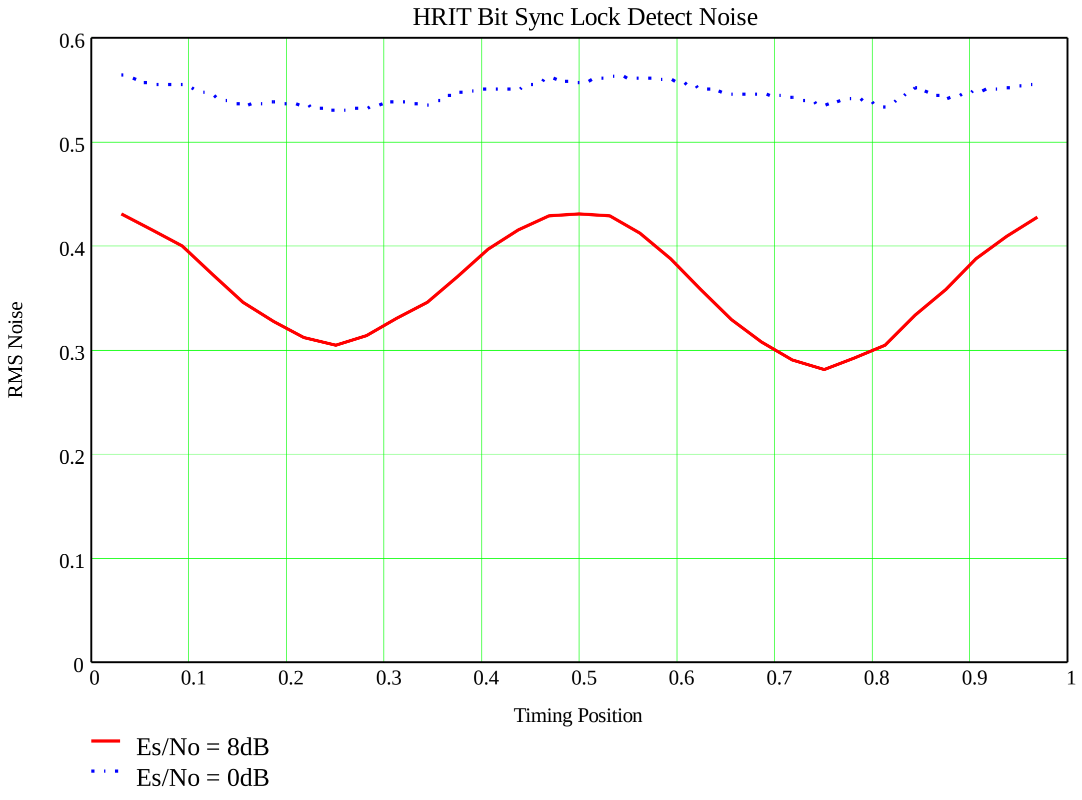


Array index values are:
0=timing
1=PSD voltage
2=PSD variance
3=Symbol Perror
4=Technology loss on Pe(sym) (dB)
5=LD voltage
6=LD variance

h2 =

	0	1	2	3	4	5	6
0	0	-0.00821	3.22665	0.2607	6.871	-0.18819	0.31638
1	0.03125	-0.14792	3.18798	0.252	6.51124	-0.18631	0.31962
2	0.0625	-0.30609	3.16349	0.2329	5.75104	-0.17167	0.30907
3	0.09375	-0.43898	3.06488	0.2089	4.83948	-0.15831	0.30921
4	0.125	-0.54307	2.99466	0.188	4.06786	-0.13181	0.29592
5	0.15625	-0.6259	2.87126	0.1662	3.28193	-0.10319	0.28708
6	0.1875	-0.68563	2.76847	0.15	2.70091	-0.0707	0.28983
7	0.21875	-0.71596	2.61428	0.1345	2.13943	-0.03845	0.28629
8	0.25	-0.71885	2.44926	0.1212	1.65284	-0.00216	0.28092
9	0.28125	-0.69074	2.33164	0.1084	1.17714	0.03757	0.28379
10	0.3125	-0.63664	2.17858	0.1004	0.87102	0.06983	0.2915
11	0.34375	-0.56511	2.06621	0.0935	0.60207	0.10277	0.28745
12	0.375	-0.46916	1.95277	0.08828	0.39447	0.13161	0.30033
13	0.40625	-0.36258	1.88852	0.08304	0.18187	0.15805	0.304
14	0.4375	-0.25475	1.81162	0.08097	0.09657	0.17484	0.30408


HRIT Bit Sync Lock Detect Voltage
Signal
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Timing Position
Es/No = 8dB
Es/No = 0dB


HRIT Bit Sync Lock Detect Noise
RMS Noise
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Timing Position
Es/No = 8dB
Es/No = 0dB