Step 1 - Make a dynamic spectrum to detect the major tone - Carrier line

Input data file name DataFileName := "Data3.NoSpin.NoPhNoise.byte"

A priori Doppler Polynomial model file name in this case it is 0 DpcFileName0 := "Data3.Det.Pass0.tone0.fitcoeffs.txt"

Data set and processing essential parameters

Band Width $BW := 250 \cdot kHz$

Scan length (in samples) $Nt := 8 \cdot 1024 \cdot 1024$

N of points to FFT $Nps := 8 \cdot 1024$ For a second iteration, the FFT can be longer

Overlapping of FFT segments $Ovlp := 2$

Number of spectra to average (not accounting for overlap) $Nav := 8$

Padding coeff and array .Padding 2 means that input array will be papped with the same length of zeroes, so making it two times longer. padding is helpful, although cpu time consuming.
Padding 1 means no padding. $Padd := 2$

Derived parameters (some of then are just for illustration)

Sampling rate $Sr := 2 \cdot BW$

And sampling interval $dt := Sr^{-1}$ Dimensionless $dtn := dt \cdot s^{-1}$

Total time span $Tspan := Nt \cdot dt$ $Tspan = 16.777216\ s$ Dimensionless $Tspann := Tspan \cdot s^{-1}$

Input Time grid $jt := 0 .. Nt - 1$ $tt_{jt} := jt \cdot dt$

Binning within FFT segment $jps := 0 .. Nps - 1$

Window function in time domain $Win_{jps} := \cos\left[\frac{\pi}{Nps} \cdot (jps - 0.5 \cdot Nps + 0.5)\right]^2$

Time span of the FFT $tw := dt \cdot Nps$ $tw = 0.016384\ s$

Native frequency resolution $df := tw^{-1}$ $df = 61.03515625\ s^{-1}$

Derived parameters , Continue

Number of spectra to average (accounting for overlap) $Nspav := Nav \cdot Ovlp - (Ovlp - 1)$ $Nspav = 15$

shift (in samples) between overlapping segments $Oshift := \frac{Nps}{Ovlp}$

Number of resulting spectra $Nspec := \frac{Nt}{Nps \cdot Nav}$ $Nspec = 128$

Indexing the spectra $jspec := 0 .. Nspec - 1$

Input data Block length for a single averaged output spectrum $Bav := Nps \cdot Nav$

Time stamps for resulting spectra $tspec_{jspec} := (jspec + 0.5) \cdot Bav \cdot dt$

Output sampling interval $Bav \cdot dt = 0.131072\ s$ $tspec_0 = 0.065536\ s$

Padding array $Npadd := Nps \cdot (Padd - 1)$ $Npadd := if(Padd = 1, 1, Npadd)$ $jpad := 0 .. Npadd - 1$ $dpadd_{jpad} := 0$

Single-sided Output spectrum binning for 2-sided FFT $Nfp := \frac{Nps \cdot Padd}{2} + 1$ $jfs := 0 .. Nfp - 1$

Output frequency resolution, accounting for padding $dfs := \frac{1}{tw \cdot Padd}$ $dfs = 30.517578125\ s^{-1}$

and Output spectrum frequency grid $fs_{jfs} := jfs \cdot dfs$

Read Dorrpler Frequency polynomial $Fcd0 := READPRN(DpcFileName0)$

Polynomial order $Npf := Fcd0_0$ $Npf = 3$ indexing $jpf := 0 .. Npf$

Normalization time $Tnorm := Fcd0_1$ $Tnorm = 16.777216$ $Tnorms := Tnorm \cdot s$ $Tspan = 16.777216\ s$

$Tnorms = 16.777216\ s$

Frequency coefficients $Cf0_{jpf} := Fcd0_{jpf+2}$

$Cf := Cf0$

Make Phase polynomials $Cpp_0 := 0$ $Cpp_{jpf+1} := 2\pi \cdot \frac{Cf_{jpf}}{jpf + 1}$ $Cpp_1 := 0$

$$Cf = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad Cpp = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Though all the polys are zeroes, do it anyway, compute the phase

$$Phdopp_{jt} := Tnorm \cdot \sum_{jjp=2}^{Npf} \left[Cpp_{jjp} \cdot \left(\frac{tt_{jt}}{Tnorms} \right)^{jjp} \right]$$

$max(Phdopp) = 0$

$min(Phdopp) = 0$

Major function to compute a dynamic spectrum
Input : Jspec - averaged spectrum number to compute,
Phcorr - model of the Doppler phase correction.
Parameters defined in the text above:
Nfp - number of output spectral bins
Nps - number of input samples to FFT (not accounting for padding)
Nspav - number of spectra to average to produce a single output spectrum
Bav - input data block length on which the output spectrum is computed
Ovlp - Overlapping parameter
dtn and Tspann - sampling interval and scan length
Padd and dpadd - padding coeff and padding array

```
MakeSpec(Jspec, Phcorr) :=
  for jj ∈ 0 .. Nfp − 1
    spa_jj ← 0
  for jspav ∈ 0 .. Nspav − 1
    skip ← Jspec · Bav + jspav · Nps / Ovlp
    din ← READBIN(DataFileName, "byte", 0, 1, skip, Nps)
    din ← din − 127
    phc ← submatrix(Phcorr, skip, skip + Nps − 1, 0, 0)
    ephc ← exp(i · phc)
    din ← (din · ephc)
    din ← (din · Win)
    dinp ← if(Padd > 1, stack(din, dpadd), din)
    sp ← cfft(dinp)
    for jj ∈ 0 .. Nfp − 1
      spa_jj ← spa_jj + Re(sp_jj)^2 + Im(sp_jj)^2
  return spa
```

Initialize an array for spectrum accumulation

Compute a skip pointer to get a proper data segment

Read data segment to be FFTed

Read the current correction phase for data segment

Compute a complex exponent of phase

Multiply data segment by complex exponent and window

Note the signal becomes complex after phase correction

Padd with zero array for super-resolution, if asked for

Here the two-sided complex-to-complex FFT is used because
after doppler correction the input signal becomes complex

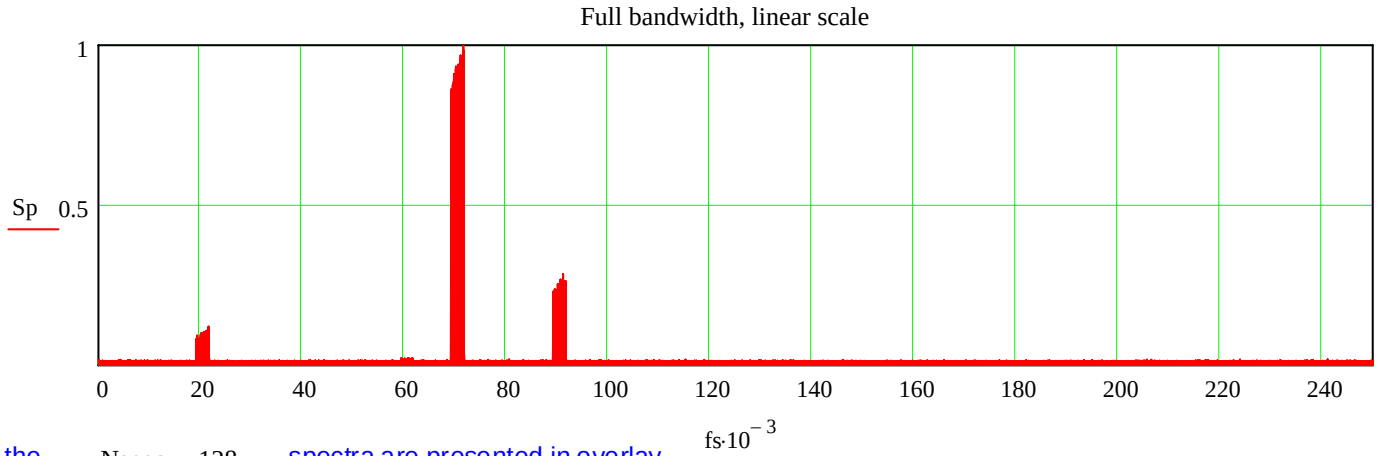
Accumulate the power spectrum, only positive frequencies of FFT

Return the power spectrum for a given averaging interval

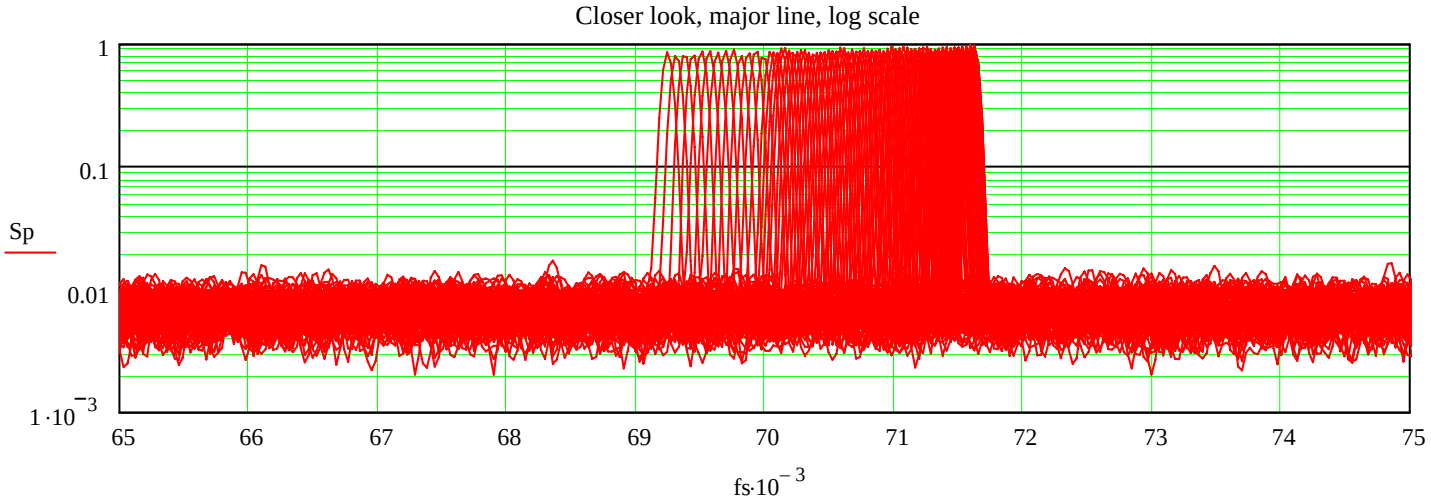
Make a dynamic spectrum $Sp^{\langle jspec \rangle} := MakeSpec(jspec, Phdopp)$

Prepare for plotting $xSp := max(Sp)$ $Sp := Sp \cdot xSp^{-1}$

Show the spectra (frequency scale in kHz)

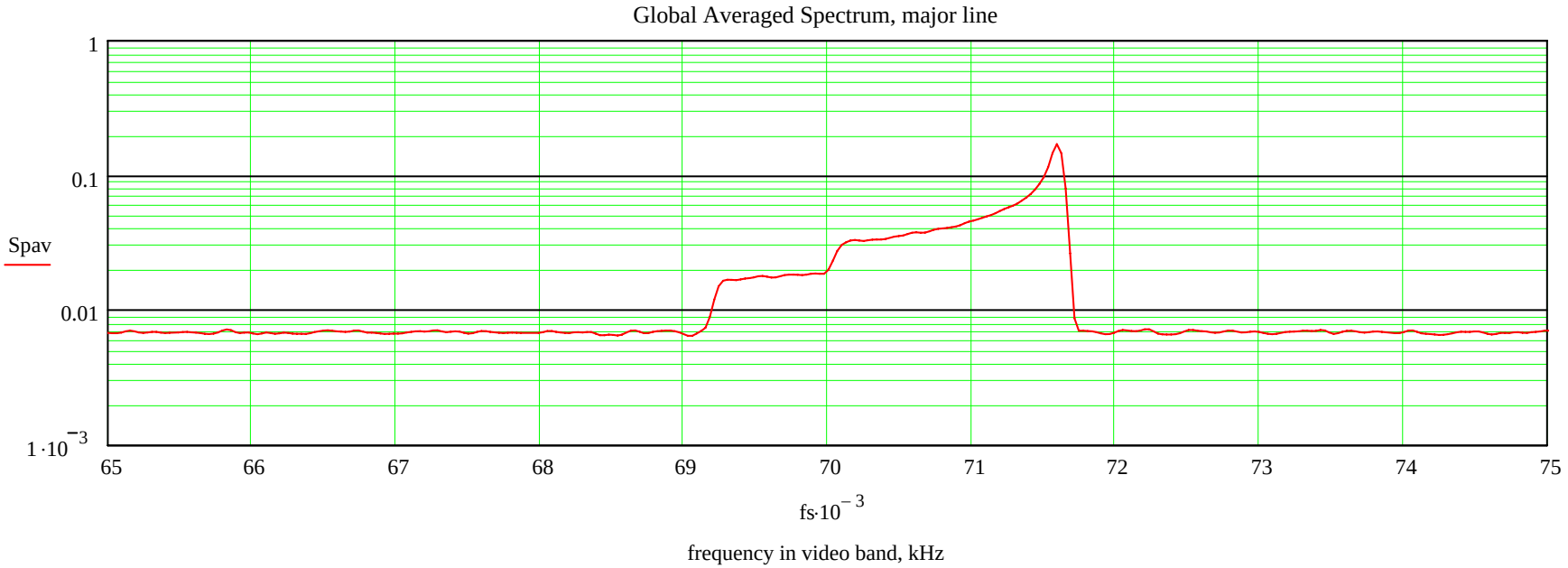


All the $Nspec = 128$ spectra are presented in overlay



Global averaged spectrum

$$Spav_{jfs} := \frac{1}{Nspec} \cdot \left(\sum_{jspec} Sp_{jfs,\, jspec} \right)$$



Width is about 2 kHz, with about 60 Hz resolution

Analyse the results

Input : spectrum, frequency scale,
search window defined by max and min frequency.

Output : interger bin number of the max value,
parabolic estimate of max position,
max value itself

$$\begin{aligned}
&\text{FindMax}(\text{Spec}, \text{Fscale}, \text{Fmin}, \text{Fmax}) := \\
&\quad \text{np} \leftarrow \text{length}(\text{Spec}) \\
&\quad \text{mx} \leftarrow 0 \\
&\quad \text{for } \text{jj} \in 0 \,..\, \text{np} - 1 \\
&\quad\quad \text{mx} \leftarrow \text{if}\left(\text{Spec}_{\text{jj}} > \text{mx} \wedge \text{Fmin} < \text{Fscale}_{\text{jj}} < \text{Fmax}, \text{Spec}_{\text{jj}}, \text{mx}\right) \\
&\quad \text{jmax} \leftarrow 0 \\
&\quad \text{for } \text{jj} \in 0 \,..\, \text{np} - 1 \\
&\quad\quad \text{jmax} \leftarrow \text{if}\left(\text{Spec}_{\text{jj}} = \text{mx} \wedge \text{Fmin} < \text{Fscale}_{\text{jj}} < \text{Fmax}, \text{jj}, \text{jmax}\right) \\
&\quad \text{jmax} \leftarrow \text{if}(\text{jmax} = 0, 1, \text{jmax}) \\
&\quad \text{jmax} \leftarrow \text{if}(\text{jmax} = \text{np} - 1, \text{np} - 2, \text{jmax}) \\
&\quad \text{a2} \leftarrow 0.5 \cdot \left(\text{Spec}_{\text{jmax}-1} + \text{Spec}_{\text{jmax}+1} - 2 \cdot \text{Spec}_{\text{jmax}}\right) \\
&\quad \text{a1} \leftarrow 0.5 \cdot \left(\text{Spec}_{\text{jmax}+1} - \text{Spec}_{\text{jmax}-1}\right) \\
&\quad \text{djx} \leftarrow \frac{-\text{a1}}{2 \cdot \text{a2}} \\
&\quad \text{xmax} \leftarrow \text{jmax} + \text{djx} \\
&\quad \text{return } \text{stack}(\text{jmax}, \text{xmax}, \text{mx})
\end{aligned}$$

Input : spectrum, frequency scale, line frequency,
estimation window defined by half width with respect
to the line position and and half width of line avoidance.

Output : mean value in the window and rms in the window

$$\begin{aligned}
&\text{GetRMS}(\text{Spec}, \text{Fscale}, \text{Fline}, \text{Fspan}, \text{Fvoid}) := \\
&\quad \text{np} \leftarrow \text{length}(\text{Spec}) \\
&\quad \text{mw} \leftarrow 0 \\
&\quad \text{ww} \leftarrow 0 \\
&\quad \text{for } \text{jj} \in 0 \,..\, \text{np} - 1 \\
&\quad\quad \text{ww} \leftarrow \text{ww} + \text{if}\left(\left|\text{Fscale}_{\text{jj}} - \text{Fline}\right| < \text{Fspan} \wedge \left|\text{Fscale}_{\text{jj}} - \text{Fline}\right| > \text{Fvoid}, 1, 0\right) \\
&\quad\quad \text{mw} \leftarrow \text{mw} + \text{if}\left(\left|\text{Fscale}_{\text{jj}} - \text{Fline}\right| < \text{Fspan} \wedge \left|\text{Fscale}_{\text{jj}} - \text{Fline}\right| > \text{Fvoid}, \text{Spec}_{\text{jj}}, 0\right) \\
&\quad \text{mm} \leftarrow \frac{\text{mw}}{\text{ww}} \\
&\quad \text{dw} \leftarrow 0 \\
&\quad \text{for } \text{jj} \in 0 \,..\, \text{np} - 1 \\
&\quad\quad \text{dw} \leftarrow \text{dw} + \text{if}\left[\left|\text{Fscale}_{\text{jj}} - \text{Fline}\right| < \text{Fspan} \wedge \left|\text{Fscale}_{\text{jj}} - \text{Fline}\right| > \text{Fvoid}, \left(\text{Spec}_{\text{jj}} - \text{mm}\right)^2, 0\right] \\
&\quad \text{rm} \leftarrow \sqrt{\frac{\text{dw}}{\text{ww}}} \\
&\quad \text{return } \text{stack}(\text{mm}, \text{rm})
\end{aligned}$$

Detect the frequncy of the major tone (in a proper frequency window)

$FsearchMin := 60 \cdot kHz$

$FsearchMax := 80 \cdot kHz$

$xf^{\langle jspec \rangle} := FindMax\left(Sp^{\langle jspec \rangle}, fs, FsearchMin, FsearchMax\right)$ $\qquad Fdet_{jspec} := dfs \cdot xf_{1,\,jspec}$

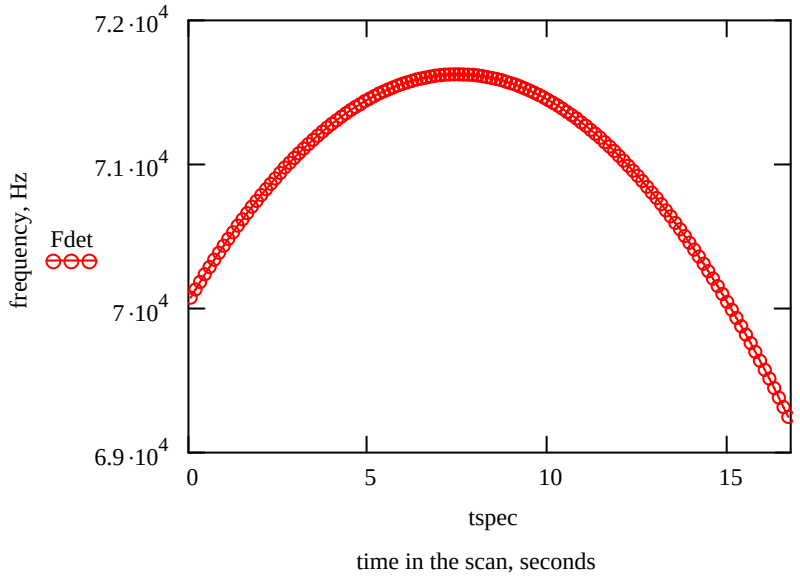
Get SNR

$HalfWindow := 8 \cdot kHz$ $\qquad LineAvoidance := 0.5 \cdot kHz$

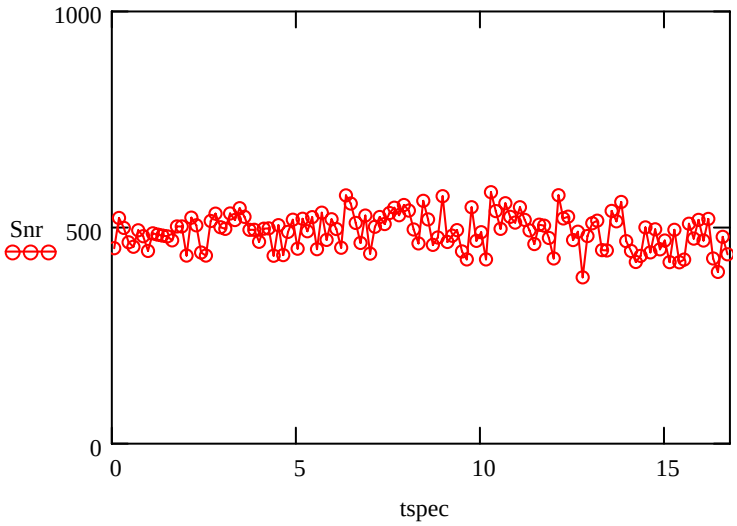
$rmsd^{\langle jspec \rangle} := GetRMS\left(Sp^{\langle jspec \rangle}, fs, Fdet_{jspec}, HalfWindow, LineAvoidance\right)$

$$Snr_{jspec} := \frac{xf_{2,\,jspec} - rmsd_{0,\,jspec}}{rmsd_{1,\,jspec}}$$ $\qquad mean(Snr) = 490.6276972409357$

Plot the detected frequency



And SNR



Weighted Polyfit, returns fit

$$\text{PolyfitW1}(\text{tin},\text{din},\text{win},\text{np}) :=$$

$nx \leftarrow \text{length}(\text{tin})$

$xx \leftarrow \max(\text{tin})$

$dtin \leftarrow \text{tin}_1 - \text{tin}_2$

$xn \leftarrow \text{tin} \cdot \text{Tspan}^{-1}$

for $jp \in 0 .. np$

$$Vp_{jp} \leftarrow \sum_{jj=0}^{nx-1} \left[\left(xn_{jj}\right)^{jp} \cdot din_{jj} \cdot win_{jj} \right]$$

for $ip \in 0 .. np$

$$Mp_{jp,ip} \leftarrow \sum_{jj=0}^{nx-1} \left[win_{jj} \cdot \left(xn_{jj}\right)^{jp+ip} \right]$$

$Mr \leftarrow Mp^{-1}$

$Cp \leftarrow Mr \cdot Vp$

for $jj \in 0 .. nx - 1$

$$yf_{jj} \leftarrow \sum_{jp=0}^{np} \left[Cp_{jp} \cdot \left(xn_{jj}\right)^{jp} \right]$$

return yf

Weighted Polyfit, returns fit coefficients

$$\text{PolyfitW1C}(\text{tin},\text{din},\text{win},\text{np}) :=$$

$nx \leftarrow \text{length}(\text{tin})$

$xx \leftarrow \max(\text{tin})$

$dtin \leftarrow \text{tin}_1 - \text{tin}_2$

$xn \leftarrow \text{tin} \cdot \text{Tspan}^{-1}$

for $jp \in 0 .. np$

$$Vp_{jp} \leftarrow \sum_{jj=0}^{nx-1} \left[\left(xn_{jj}\right)^{jp} \cdot din_{jj} \cdot win_{jj} \right]$$

for $ip \in 0 .. np$

$$Mp_{jp,ip} \leftarrow \sum_{jj=0}^{nx-1} \left[win_{jj} \cdot \left(xn_{jj}\right)^{jp+ip} \right]$$

$Mr \leftarrow Mp^{-1}$

$Cp \leftarrow Mr \cdot Vp$

$dout \leftarrow Cp$

return $dout$

Read the input signal model

ModelData := READPRN("Data3.NoSpin.NoPhNoise.PhaseModel.txt")

ttd := $ModelData^{\langle 0 \rangle}$

Fmodel := $ModelData^{\langle 1 \rangle}$ Finp := Spline(ttd, Fmodel, tspec · s^{-1})

Make a polynomial fit to detected frequency

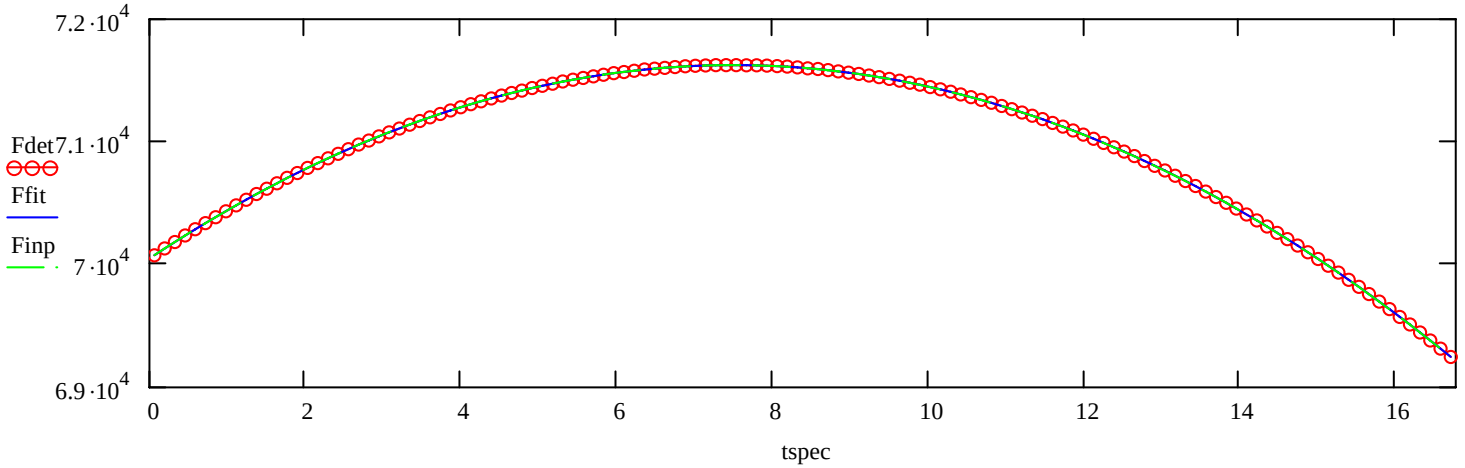
$Weight_{jspec} := (Snr_{jspec})^2$

Npoly := 3

Ffit := PolyfitW1(tspec, Fdet, Weight, Npoly) rFdet := Fdet − Ffit stdev(rFdet) = 0.684990066781327 s^{-1} meanSNR := mean(Snr) meanSNR = 490.6276972409357

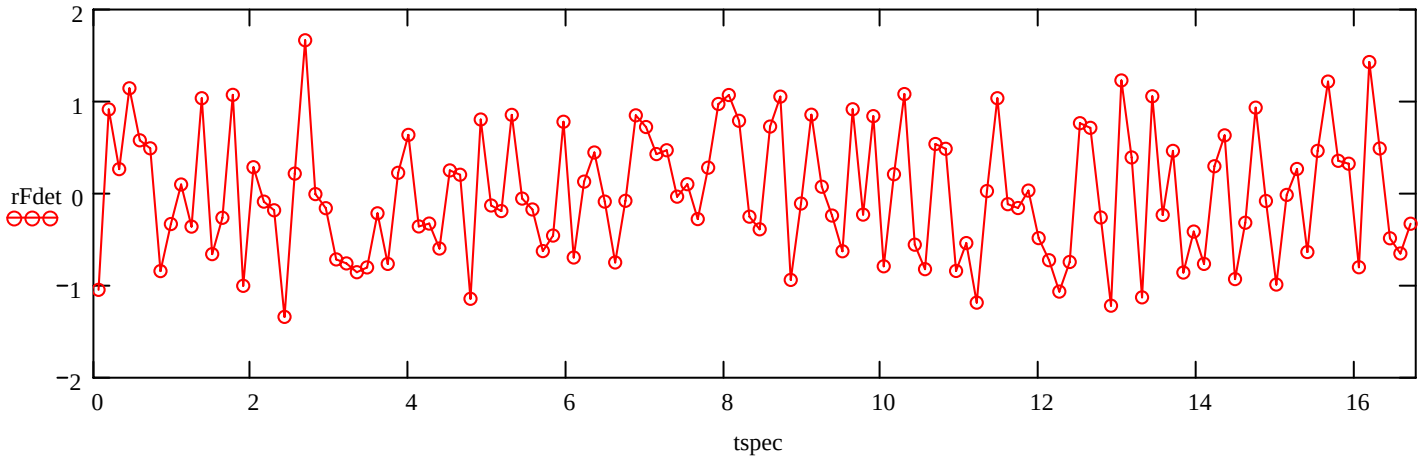
FitCoeffs := PolyfitW1C(tspec, Fdet · Hz^{-1}, Weight, Npoly) $\frac{df}{meanSNR}$ = 0.124402182333435 s^{-1}

Detected frequency and fit



FitCoeffs = $\begin{pmatrix} 70048.91028919304 \\ 7052.868226490915 \\ -7888.8134511858225 \\ 1.104737497866154 \end{pmatrix}$

Post fit residual



Write it down

Dout := stack(Npoly, Tspann, FitCoeffs)

WxRITEPRN("Data3.Det.Pass1.tone0.fitcoeffs.txt") := Dout

$Ffit_0 = 70076.34018202822\ s^{-1}$

$Finp_0 = 70077.13464662689$

Compare with input frequency model used to generate the input signal