Analysis of CO $^{60}\beta$-source data. Data set consists of 1790 time stamped waveforms recorded at 145 GSa/s.

In[1]:=
```
SetDirectory["~white/Desktop/4mm^2"];
Drop[FileNames[], 1] // Length;
ntrace = %
```

Out[3]= 1790

In[4]:=
```
Namelist = Drop[FileNames[], 1];
time = Transpose[Import[Namelist[[1]], "Data"]][[1]];
nbins = Dimensions[time][[1]]
```

Out[6]= 15 986

In[7]:=
```
trange = time[[nbins]] - time[[1]];
t[bin_] := time[[1]] + trange * (bin) / nbins
bint[tvar_] := (tvar - time[[1]]) * nbins / trange
```

In[10]:=
```
noisemax = 7500;
Print["Do noise analysis for the first  ",
 (t[noisemax] - t[1]) × 10^9, " nanoseconds"]
front = Transpose[Import[Namelist[[1]], "Data"]][[2]];
back = Transpose[Import[Namelist[[1]], "Data"]][[3]];
offset = Mean[Take[front, noisemax]] // EngineeringForm
```

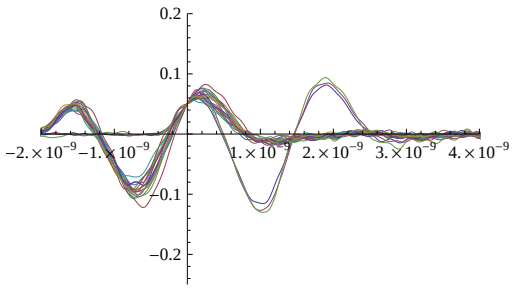
Do noise analysis for the first 46.8658 nanoseconds

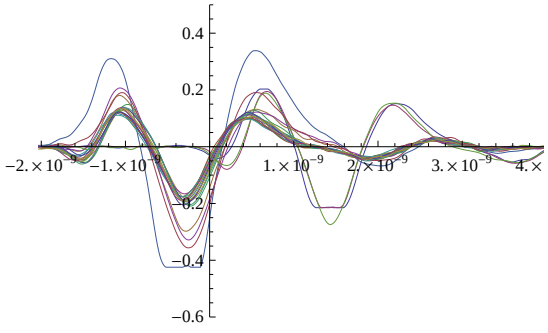
Out[14]//EngineeringForm=

-428.374×10^{-6}

Front and Back are the 2 APDs which face eachother. Front is closest to the CO ^{60} source. Back triggers the scope.
The first 47 nanoseconds are used to extract the baseline correction which is typically ~ -.5 mV.

Inspect some waveforms

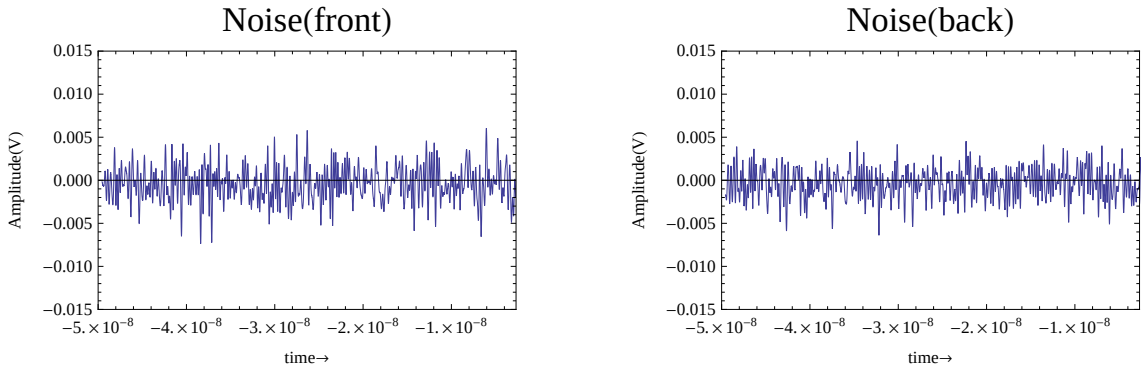




In[16]:=
```
tnoisemax = t[noisemax]
```

Out[16]= -2.68056×10^{-9}

Out[17]=



In[18]:=
```
dt = time[[2]] - time[[1]] // EngineeringForm
f = 1 / (time[[2]] - time[[1]]);
fftrange = f / 2
```

Out[18]//EngineeringForm=

6.2×10^{-12}

Out[20]= 8.06452×10^{10}

RMS noise and baseline

In[21]:=
```
basef = {}; noisef = {}; baseb = {}; noiseb = {}; front1 = {}; back1 = {};
fbin = {}; bbin = {};
iwave = 1000;
```

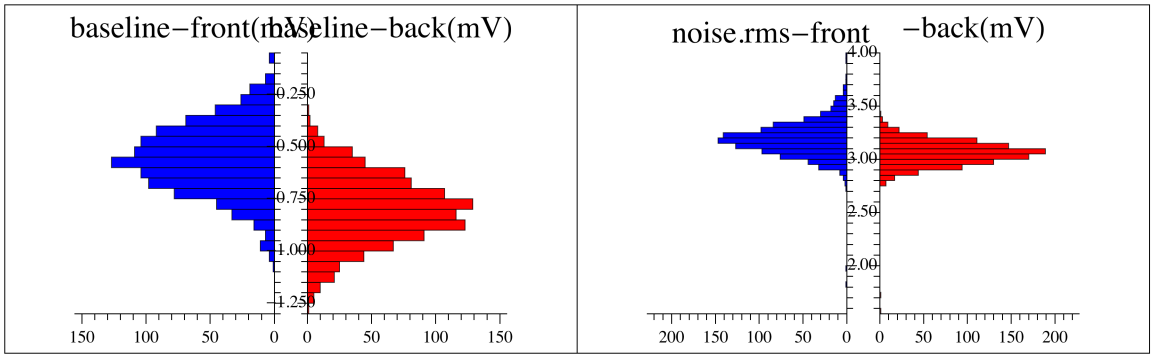
In[24]:=
```
Do[
 front1 = Take[Transpose[Import[Namelist[[itrace]], "Data"]][[2]], noisemax];
 back1 = Take[Transpose[Import[Namelist[[itrace]], "Data"]][[3]], noisemax];
 AppendTo[basef, Mean[front1]];
 AppendTo[baseb, Mean[back1]];
 AppendTo[noisef, RootMeanSquare[front1]]; AppendTo[noiseb, RootMeanSquare[back1]];
 , {itrace, iwave}]
```

In[25]:=
```
avenoisef = Mean[noisef]
avenoiseb = Mean[noiseb]
```

Out[25]= 0.00319179

Out[26]= 0.00305551

Out[28]=



In[29]:=
```
front1 = Take[Transpose[Import[Namelist[[1]], "Data"]][[2]], noisemax];
back1 = Take[Transpose[Import[Namelist[[1]], "Data"]][[3]], noisemax];
time1 = Take[Transpose[Import[Namelist[[1]], "Data"]][[1]], noisemax];
```

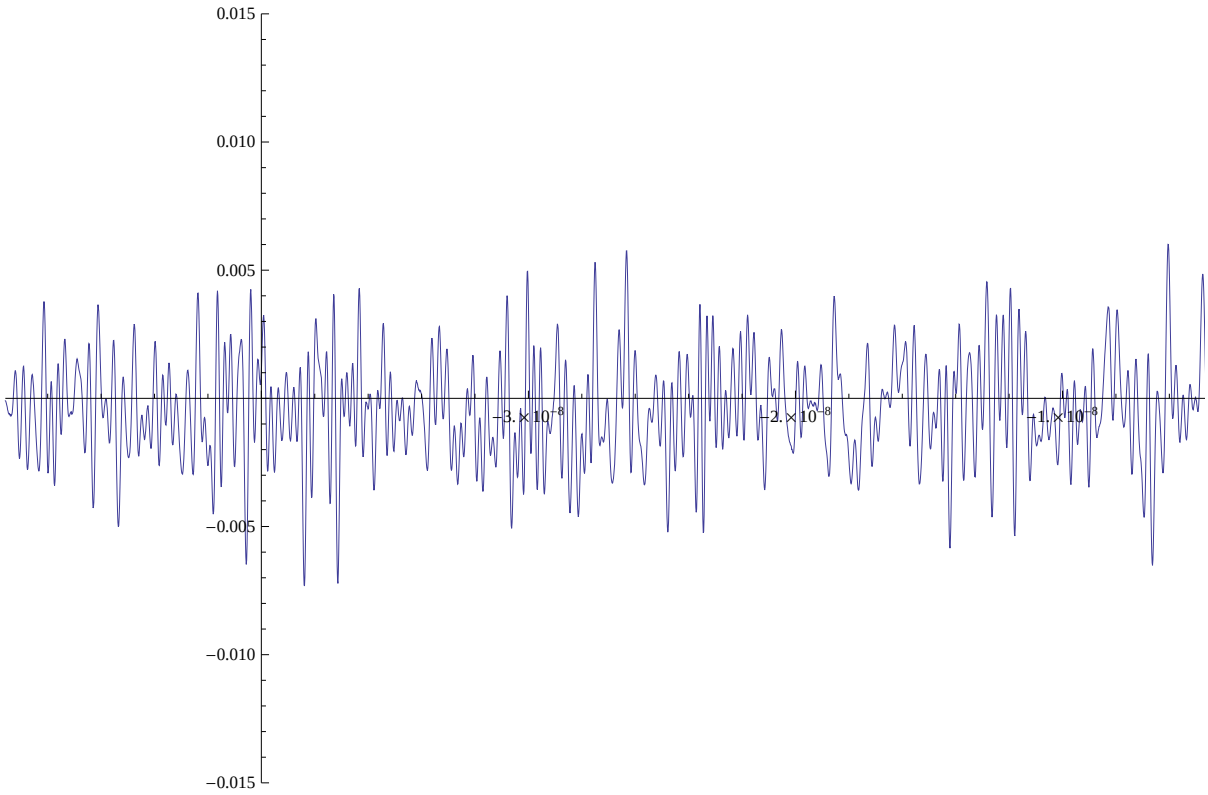
In[32]:=
```
Ff = Interpolation[Transpose[{time1, front1}], Method → "Spline"];
```

In[33]:=
```
Fb = Interpolation[Transpose[{time1, back1}], Method -> "Spline"]
```

Out[33]=
```
InterpolatingFunction[{{-4.95526 × 10^-8, -2.6839 × 10^-9}}, <>]
```

In[34]:=
```
Plot[Ff[x], {x, -4.9586 × 10^-8, -3 × 10^-9},
 PlotRange → {{-4.9586 × 10^-8, -3 × 10^-9}, {-.015, .015}}]
```

Out[34]=



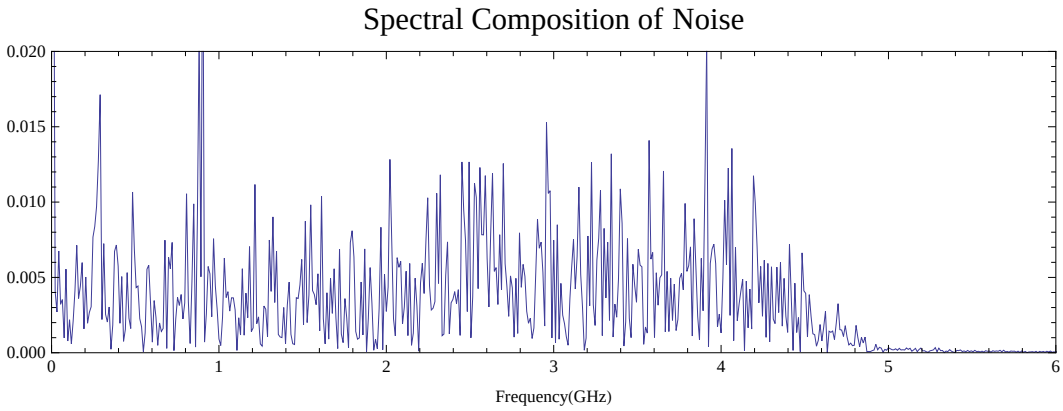
In[35]:=
```
fftout1 = Abs[ FourierDCT[front1]];
df = fftrange / noisemax;
```

In[37]:=
```
freq = Range[df, fftrange, df] / 1 000 000 000.;
```

In[38]:=
```
fftnew = Transpose[{freq, fftout1}];
```

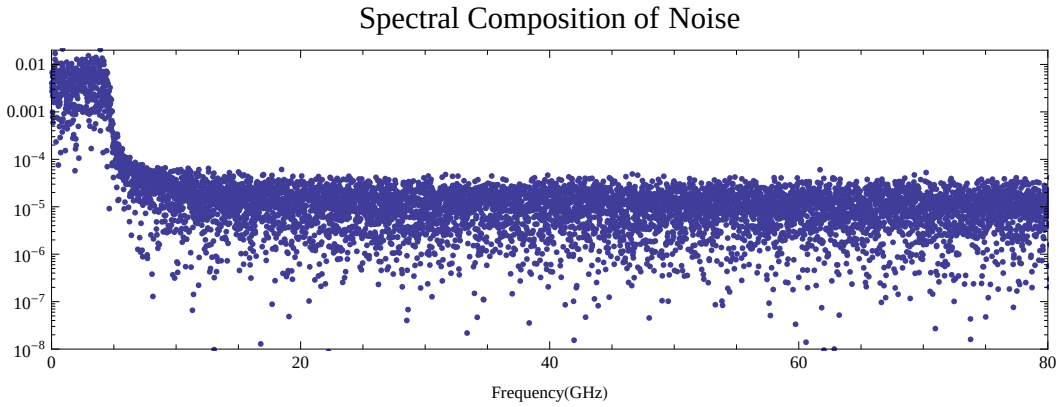
For this scope the Sampling frequency is much higher than the actual bandwidth. Therefore in the plots below there is a white noise distribution out to a few GHz. Above this (the scope bandwidth) there is an apparent reduction in noise. But in this region data are really reflecting the fact that high sampling frequency is really a figment of ' on - chip " interpolation.

Out[39]=



This has zero frequency in element 1. The 7693- th element corresponds to 1/2 the sampling frequency.After that aliassing takes over and the frequency heads back to zero.

Out[40]=



Initialize useful bin locations for region of interest and location of first peak in the waveform

In[219]:=
```
peakbin = IntegerPart[bint[-1.5 × 10^-9]];
startbin = IntegerPart[bint[-2 × 10^-9]];
endbin = IntegerPart[bint[3 × 10^-9]] - 1;
timestrip = Take[time, {startbin, endbin}];
```

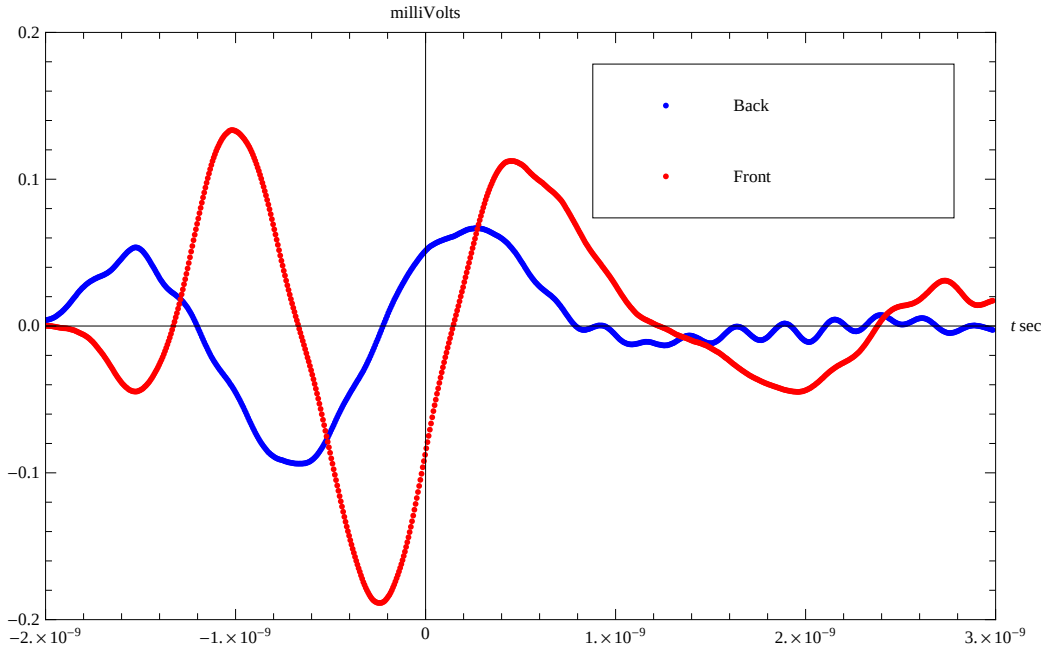
In[223]:=
```
Clear[backstrip]; Clear[frontstrip];
Clear[backs]; Clear[fronts];
frontstrip = {}; backstrip = {}; isaved = 0; backs {}; fronts {};
```

In[226]:=
```
Do[
 frontraw = Transpose[Import[Namelist[[i]], "Data"]][[2]] - basef[[i]];
 (*baseline restoration on waveforms using average level of 1st 46.8 nsec*)
 backraw = Transpose[Import[Namelist[[i]], "Data"]][[3]] - baseb[[i]];
 If[backraw[[peakbin]] < 0.02, Goto["skipwave"], isaved++];
 (*first analyze events which trigger on the 2nd peak*)
 backs = Take[backraw, {startbin, endbin}];
 fronts = Take[frontraw, {startbin, endbin}];
 AppendTo[backstrip, backs];
 AppendTo[frontstrip, fronts];
 Label["skipwave"];
 , {i, iwave}]
```

In[227]:=
```
Dimensions[backstrip]
Dimensions[frontstrip]
```

Out[227]= {653, 800}

Out[228]= {653, 800}

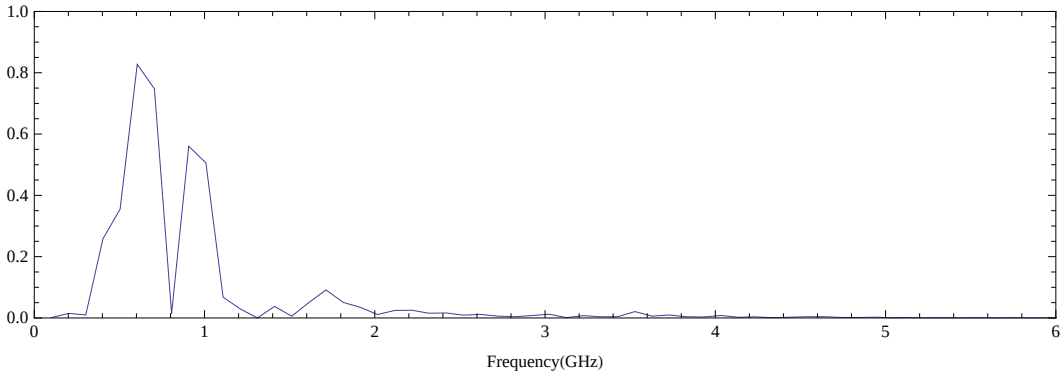


.

Now examine the frequency content of the signal.

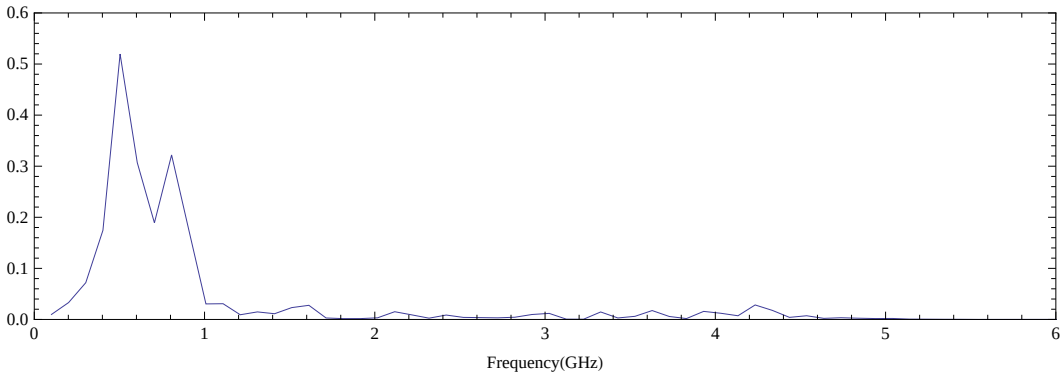
In[279]:=
```
fftout2 = Abs[ FourierDCT[frontstrip[[1]]]];
df2 = fftrange / 800;

freq2 = Range[df2, fftrange, df2] / 1 000 000 000.;
fftnew2 = Transpose[{freq2, fftout2}];
```

Out[290]=



In[291]:=
```
fftout3 = Abs[ FourierDCT[backstrip[[1]]]];
fftnew3 = Transpose[{freq2, fftout3}];
```

Out[295]=



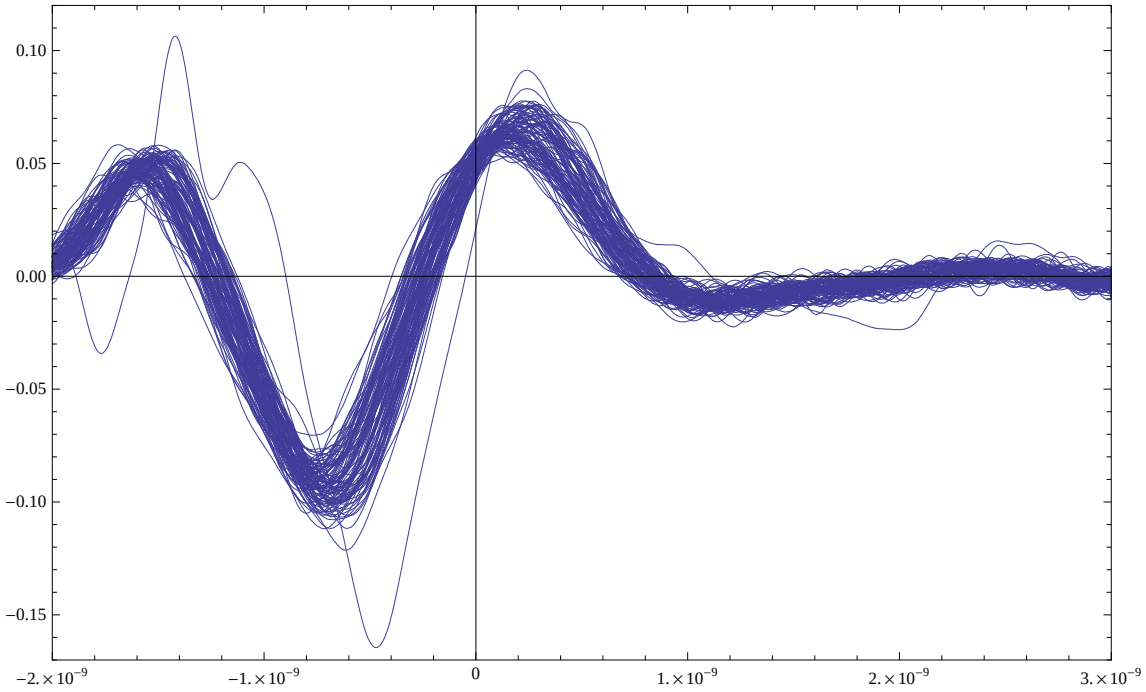
In[229]:=
```
Wf = ConstantArray[0, 1000]; Wb = ConstantArray[0, 1000];
```

In[230]:=
```
Do[
Wf[[itrace]] =
  Interpolation[Transpose[{timestrip, frontstrip[[itrace]]}], Method → "Spline"];
 Wb[[itrace]] = Interpolation[Transpose[{timestrip, backstrip[[itrace]]}],
   Method → "Spline"];
 , {itrace, isaved}]
```

In[244]:=
```
Clear[Zeroesb]; Clear[Zeroesf];
Zeroesb = x /. FindRoot[Wb[[#]][x], {x, -1.2 × 10^-9}] & /@ Range[isaved];
```

In[246]:=
```
Zeroesf = x /. FindRoot[Wf[[#]][x], {x, -0.7 × 10^-9}] & /@ Range[isaved];
```

Out[256]=



In[258]:=
```
oa = OpenAppend["InterpolatedWaveForms.dat", PageWidth → Infinity];
Do[
 Write[oa, {i, Wb[[i]], Wf[[i]]}];,
 {i, 1, iwave}]
Close[oa];
```

In[296]:=
```
Dimensions[backstrip]
```

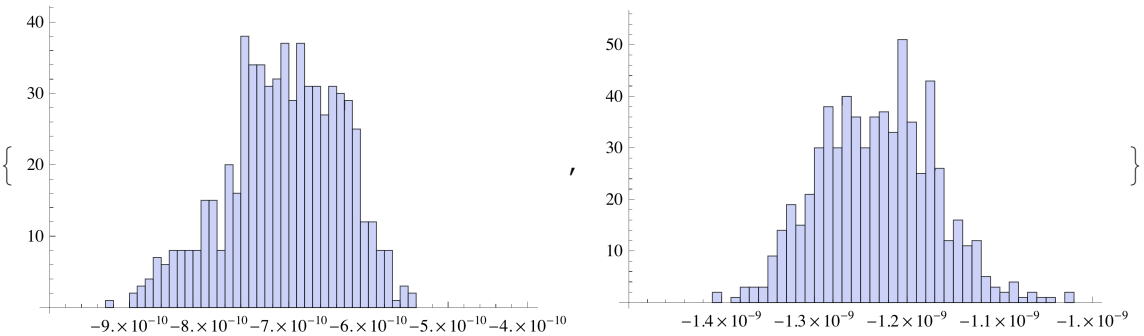
Out[296]= {653, 800}

In[297]:=
```
ob = OpenAppend["StrippedWaveForms.dat", PageWidth → Infinity];
Write[ob, timestrip];
Do[
 Write[ob, {backstrip[[i]], frontstrip[[i]]}];,
 {i, 1, isaved}]
Close[ob];
```

In[247]:=
```
Clear[Differ]
Differ = Table[(Zeroesf[[i]] - Zeroesb[[i]]), {i, 1, isaved}];
```

In[276]:=
```
{Histogram[Zeroesf, {-1 × 10^-9, -.4 × 10^-9, 0.01 × 10^-9}],
 Histogram[Zeroesb, {-1.5 × 10^-9, -1 × 10^-9, 0.01 × 10^-9}]}
```

Out[276]=



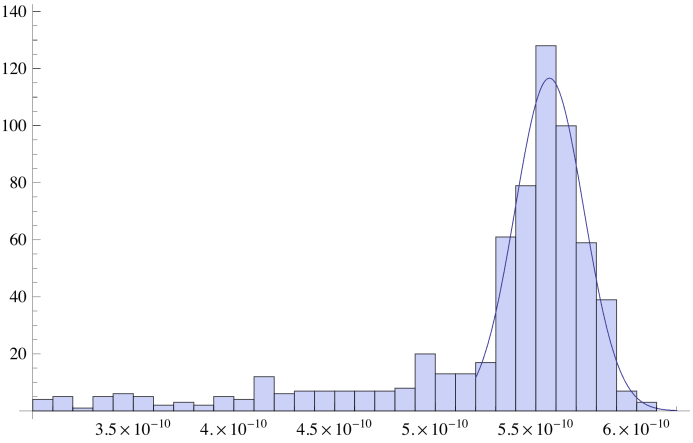
In[317]:=
```
histdiffer = Histogram [Differ , {0.3 × 10^-9, 0.62 × 10^-9, 0.01 × 10^-9}];
```

```
ff1 = FindFit[tbc1, amp * PDF[NormalDistribution[mean, sd], x],
  {{amp, amp1}, {mean, mean1}, {sd, sd1}}, x]
```

Out[316]= $\{\text{amp} \to 4.99124 \times 10^{-9}, \text{mean} \to 5.56713 \times 10^{-10}, \text{sd} \to 1.70724 \times 10^{-11}\}$

```
In[311]:= Show[histdiffer, Plot[amp * PDF[NormalDistribution[mean, sd], x] /. ff1,
  {x, .52 × 10^-9, 0.62 × 10^-9}, PlotRange → All] ]
```

Out[311]=



Evaluate the slope of the wave function over 300 psec near the zero - crossing. This can be used to estimate the contribution to time jitter from scope input noise.

```
In[266]:= bslopes = {}; fslopes = {};
Do[
 AppendTo[bslopes, Wb[[i]][-.1 × 10^-9] - Wb[[i]][-.4 × 10^-9]];
 AppendTo[fslopes, Wf[[i]][.2 × 10^-9] - Wf[[i]][-.1 × 10^-9]];,
 {i, 1, isaved}]
```

```
In[268]:= meanbslope = Mean[bslopes] / (.3 × 10^-9)
          meanfslope = Mean[fslopes] / (.3 × 10^-9)
```

Out[268]= 2.51672×10^{8}

Out[269]= 6.56288×10^{8}

We then derive the expected noise jitter on the time difference measured from the zero - cross method and get 13 picoseconds which is not far from what we are finding!

```
In[270]:= resexpb = avenoiseb / meanbslope
          resexpf = avenoisef / meanfslope
          dtwidthexp = Sqrt[resexpb^2 + resexpf^2]
```

Out[270]= 1.21408×10^{-11}

Out[271]= 4.8634×10^{-12}

Out[272]= 1.30787×10^{-11}

Signal - to - Noise we see is roughly consistent with noise floor of Agilent scope we used for source tests at Princeton.

RMS Noise Floor ($V_{RMS\ AC}$)

	9064A		9104A		9254A		9404A	
Volts/div	full BW	500 MHz filter	full BW	1 GHz filter	full BW	2 GHz filter	full BW	4 GHz filter
10 mV	213 uV	138 uV	240 uV	120 uV	273 uV	210 uV	402 uV	263 uV
20 mV	470 uV	175 uV	481 uV	154 uV	445 uV	330 uV	627 uV	424 uV
50 mV	1.15 mV	.464 mV	1.24 mV	.415 mV	1.22 mV	.780 mV	1.67 mV	1.12 mV
100 mV	2.37 mV	.895 mV	2.43 mV	.786 mV	2.54 mV	1.50 mV	3.17 mV	2.16 mV
200 mV	4.65 mV	1.75 mV	4.85 mV	1.50 mV	5.06 mV	2.86 mV	6.18 mV	4.15 mV
500 mV	11.8 mV	4.60 mV	12.3 mV	4.15 mV	12.2 mV	7.61 mV	15.8 mV	11.26 mV
1 V	23.9 mV	8.91 mV	24.3 mV	7.85 mV	25.2 mV	14.9 mV	31.5 mV	21.9 mV

We are still far from noise limit due to detector leakage current. Nevertheless, dealing with this example of white noise is of interest for a heavily irradiated detector. So we are studying filtering techniques. A nice example is the "Wiener Filter".

In[322]:=
```
data4 = Table[ Sin[x], {x, 0, 2 Pi, 0.05}];
noisy = data4 + 0.1 * RandomReal[NormalDistribution[], Dimensions[data4]];
filtered = WienerFilter[noisy, 6, 0.1];
ListPlot[{noisy, filtered}, Joined → True]
```

Out[325]=