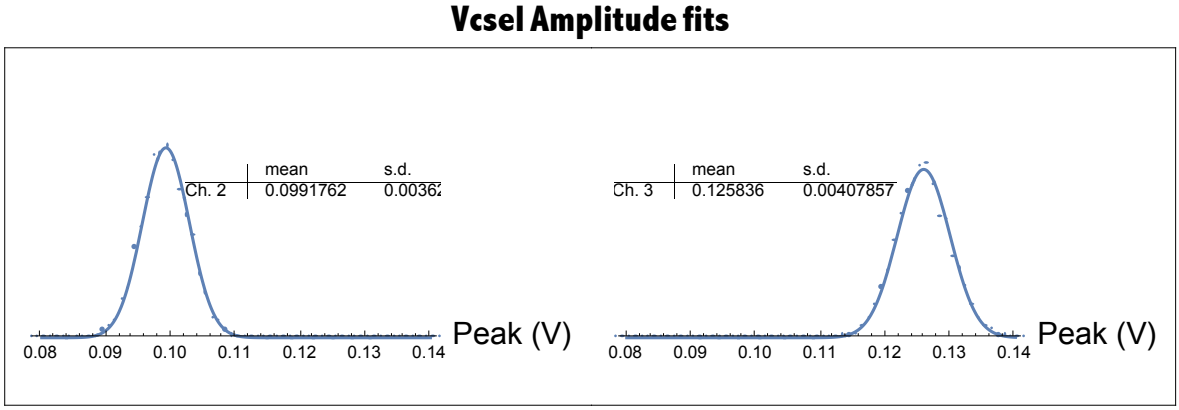
```
GraphicsRow[{Show[slpp, Graphics[Inset[Column[{pt}], {.125, 400}]]],
  Show[slpp3, Graphics[Inset[Column[{pt3}], {.1, 400}]]]},
 ImageSize → Full, Frame → True, FrameLabel → {{None, None}, {None, None}},
 PlotLabel → HoldForm[Vcsel Amplitude fits],
 LabelStyle → {FontFamily → "Abadi MT Condensed Extra Bold", 16, GrayLevel[0]}]
```

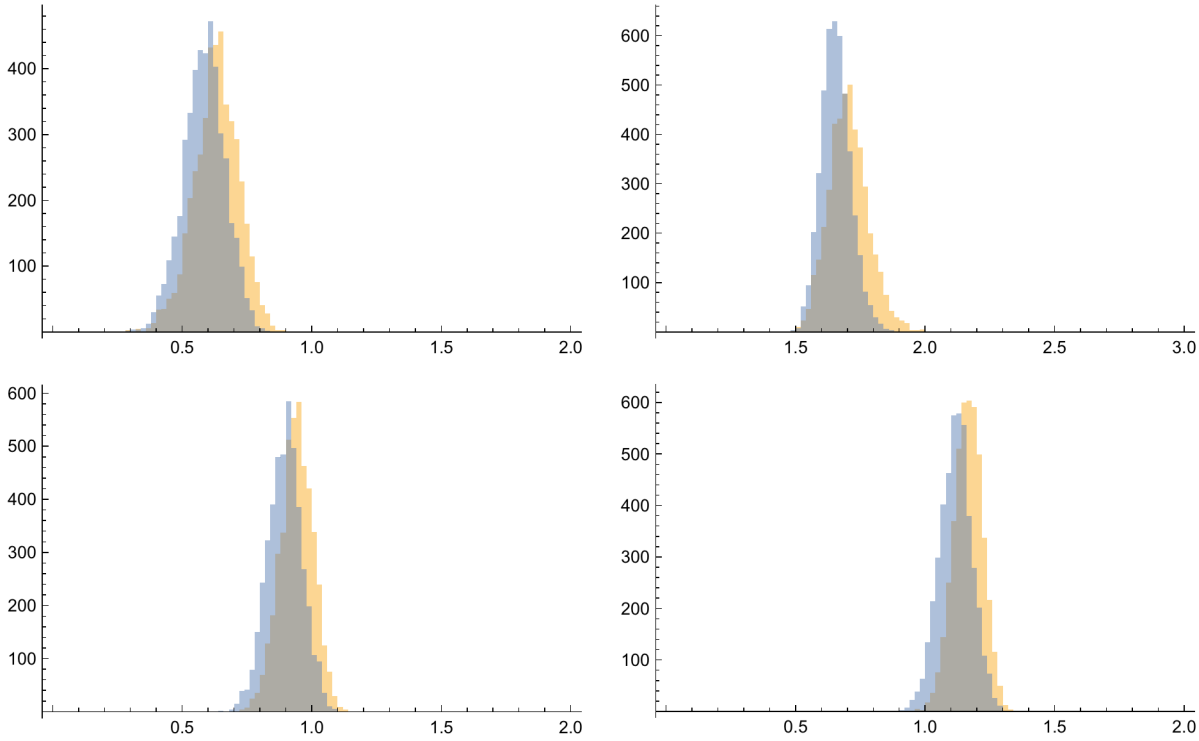


Signal to noise at Landau Peak for all channels.

```
snr = {vp2 / Mean[innoise2], vp3 / Mean[innoise3]}
```

```
{60.2832, 77.3953}
```

```
GraphicsGrid[{{Histogram[{tf21, tf31}, {0, 2, 0.02}],
    Histogram[{tf29 - tf21, tf39 - tf31}, {1, 3, 0.02}]},
   {Histogram[{tf22, tf32}, {0, 2, 0.02}],
    Histogram[{tf23, tf33}, {0, 2, 0.02}]}}, ImageSize → Full]
```




```
risetime2 = Mean[tf29 - tf21];
risetime3 = Mean[tf39 - tf31];
Print[" Average risetime for chs 2, 3= ",
 risetime2, "  ", risetime3, " nanoseconds"]
```

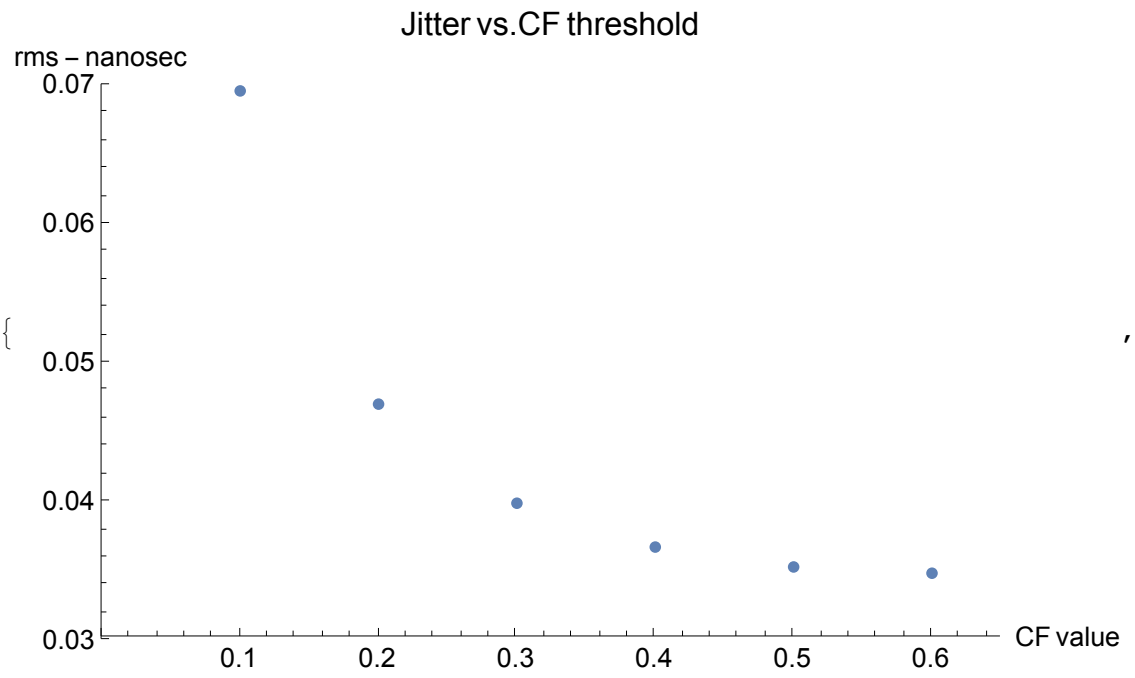
Average risetime for chs 2, 3= 1.70506 1.65667 nanoseconds

```
Print[" Estimated electronic noise term for jitter= ",
 risetime2 / snr[[1]], "  and ", risetime3 / snr[[2]], " nanoseconds "]
```

Estimated electronic noise term for jitter=
0.0282841 and 0.0214053 nanoseconds

```
mean1 = Mean[tf31 - tf21];
jitter1 = RootMeanSquare[(tf31 - tf21) - mean1] / Sqrt[2];
mean2 = Mean[tf32 - tf22];
jitter2 = RootMeanSquare[(tf32 - tf22) - mean2] / Sqrt[2];
mean3 = Mean[tf33 - tf23];
jitter3 = RootMeanSquare[(tf33 - tf23) - mean3] / Sqrt[2];
mean4 = Mean[tf34 - tf24];
jitter4 = RootMeanSquare[(tf34 - tf24) - mean4] / Sqrt[2];
mean5 = Mean[tf35 - tf25];
jitter5 = RootMeanSquare[(tf35 - tf25) - mean5] / Sqrt[2];
mean6 = Mean[tf36 - tf26];
jitter6 = RootMeanSquare[(tf36 - tf26) - mean6] / Sqrt[2];
jitter = {jitter1, jitter2, jitter3, jitter4, jitter5, jitter6};
cfth = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6};
```

```
{ListPlot[Transpose[{cfth, jitter}], PlotRange → {{0, 0.65}, {0.03, .07}},
  AxesLabel → {HoldForm[CF value], HoldForm[rms - nanosec]},
  PlotLabel → HoldForm[Jitter vs.CF threshold], LabelStyle → {14, GrayLevel[0]},
  ImageSize → Large], TableForm[Transpose[{cfth, jitter}]]}
```




```
{                                                                    ,
```

```
0.1    0.0695532
0.2    0.047014
0.3    0.0398832
0.4    0.0367287
0.5    0.0352968
0.6    0.0348468
```

There isn' t any significant improvement from taking average of several constant fractions (Delagens' method).

```
cfav2 = (tf23 + tf24 + tf25 + tf26) / 4;
cfav3 = (tf33 + tf34 + tf35 + tf36) / 4;
mcfave = Mean[cfav2 - cfav3]
jittercfave = RootMeanSquare[(cfav2 - cfav3) - mcfave] / Sqrt[2]
```

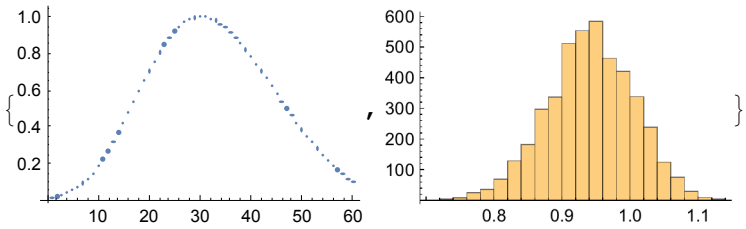
```
0.0548705
```

```
0.0360587
```

Now we try the template method

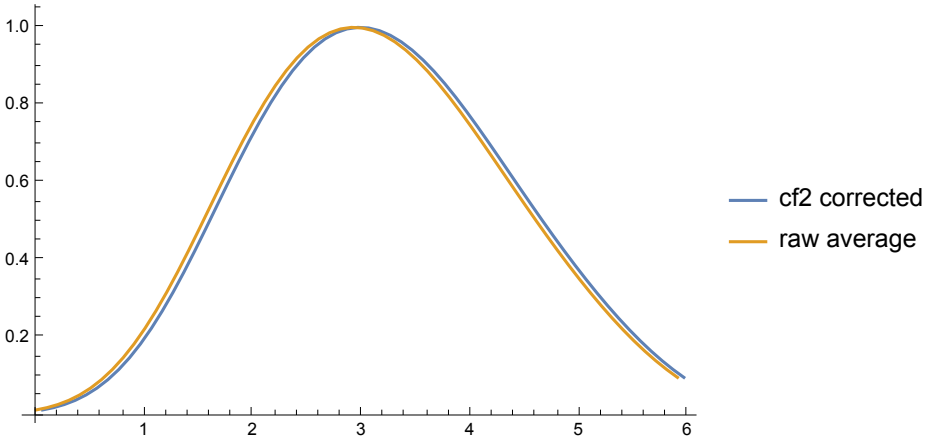
```
av2 = Sum[norm[[1, i]], {i, nevents}] / nevents;
```

```
{ListPlot[av2], Histogram[tf22]}
```



```
avcorr =
  Table[Transpose[{(timerest - tf22[[i]] + 1`), norm[[1, i]]}], {i, nevents}];
av2c = Sum[avcorr[[i]], {i, nevents}] / nevents;
```

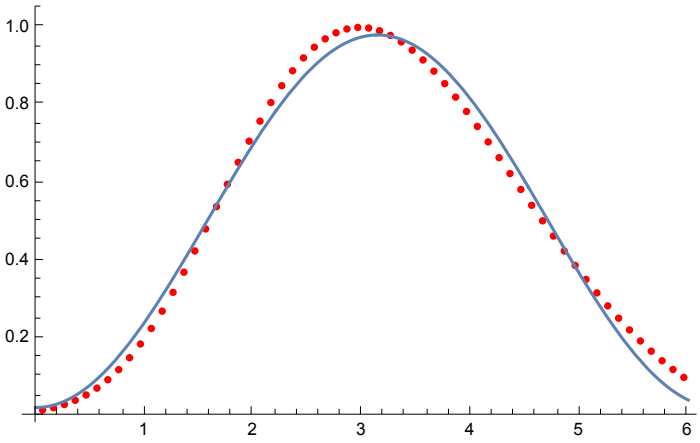
```
ListPlot[{av2c, Transpose[{timerest, av2}]},
 PlotLegends → {"cf2 corrected", "raw average"}, Joined → True]
```




```
fit = FindFormula[av2c, x]
```

```
0.499765 - 0.479884 Cos[x]
```

```
Show[ListPlot[av2c, PlotStyle → Red], Plot[fit, {x, 0, 6}]]
```



```
data = av2c;
line = Fit[data, {1, x}, x]
parabola = Fit[data, {1, x, x^2}, x];
cubic = Fit[data, {1, x, x^2, x^3}, x]
quartic = Fit[data, {1, x, x^2, x^3, x^4}, x]
quintic = Fit[data, {1, x, x^2, x^3, x^4, x^5}, x]
six = Fit[data, {1, x, x^2, x^3, x^4, x^5, x^6}, x]
```

$$0.437347 + 0.0281854\,x$$

$$-0.250487 + 0.69129\,x - 0.0998744\,x^2 - 0.00190802\,x^3$$

$$0.0453358 - 0.276884\,x + 0.621211\,x^2 - 0.188096\,x^3 + 0.0154707\,x^4$$

$$0.0968233 - 0.527018\,x + 0.909974\,x^2 - 0.315694\,x^3 + 0.0393027\,x^4 - 0.00158419\,x^5$$

$$0.0302206 - 0.0804178\,x + 0.177108\,x^2 + 0.168863\,x^3 - 0.111324\,x^4 + 0.0204245\,x^5 - 0.00121916\,x^6$$