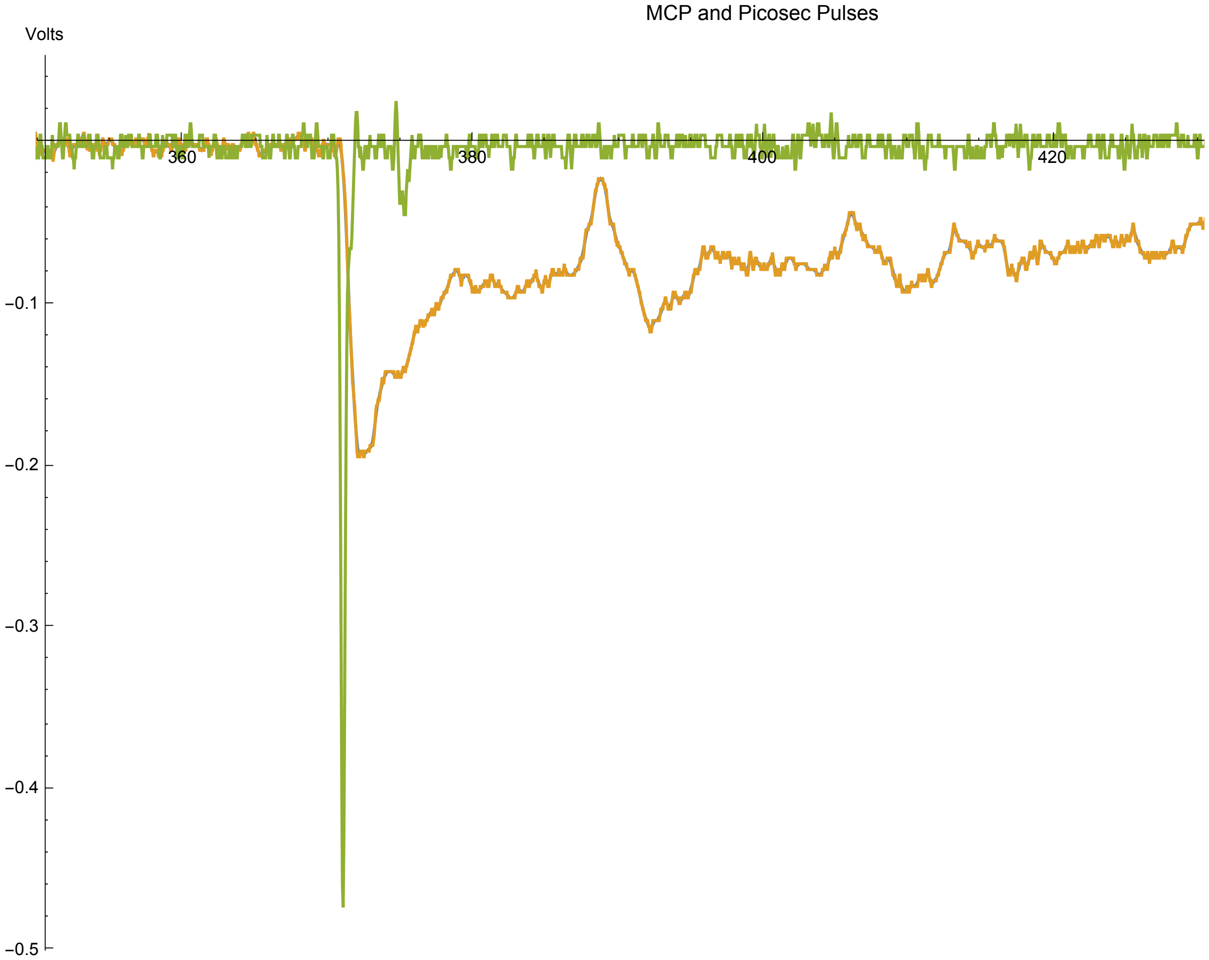
In[126]:=

```
ListPlot[{Transpose[{tt, WienerFilter[picosec1[[1]], 3, .05]}],
  Transpose[{tt, picosec1[[1]]}], Transpose[{tt, mcp[[1]]}]},
 Joined → True, AxesLabel → {"nanosec", "Volts"},
 PlotRange → {{350, 450}, Full}, PlotLabel → "MCP and Picosec Pulses"]
```

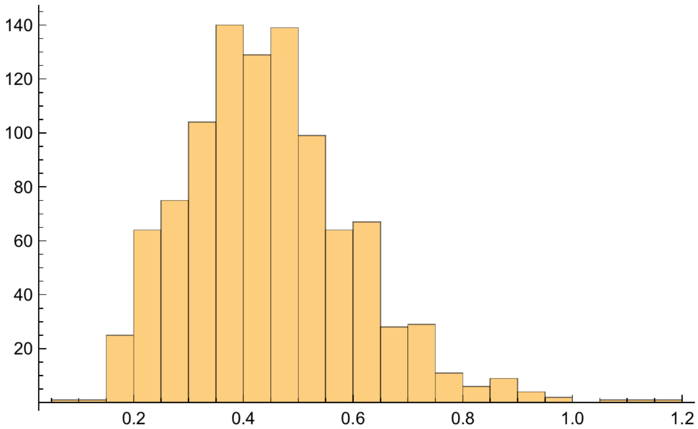
Out[126]=



In[127]:=

```
pkscn = Table[Max[-mcp[[i]]], {i, 1, nevents}];
Histogram[pkscn(*,{0,.4,.05}*)]
```

Out[128]=

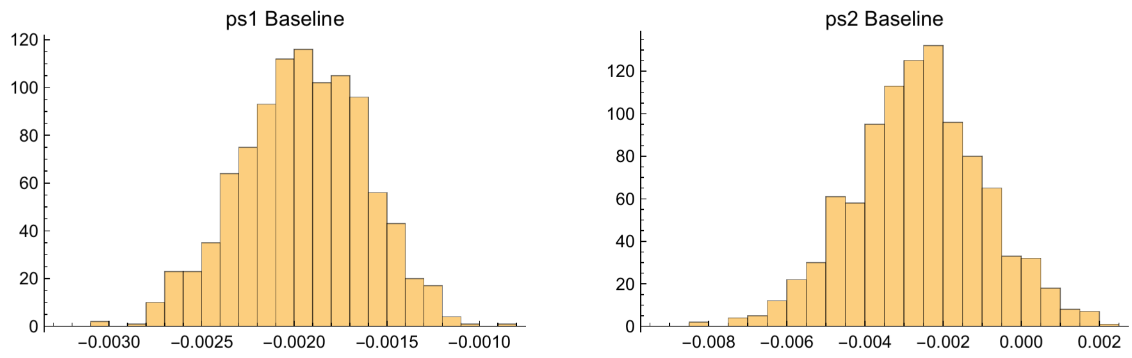


In[129]:=
```
blps1 = Table[Mean[Take[picosec1[[i]], 6000]], {i, nevents}];
blps2 = Table[Mean[Take[picosec2[[i]], 6000]], {i, nevents}];
blmcp = Table[Mean[Take[mcp[[i]], 6000]], {i, nevents}];
noiseps1 =
  Table[RootMeanSquare[Take[picosec1[[i]], 6000] - blps1[[i]]], {i, nevents}];
noiseps2 = Table[RootMeanSquare[Take[picosec1[[i]], 6000] - blps1[[i]]],
   {i, nevents}];
noisemcp = Table[RootMeanSquare[Take[mcp[[i]], 6000] - blmcp[[i]]], {i, nevents}];
```

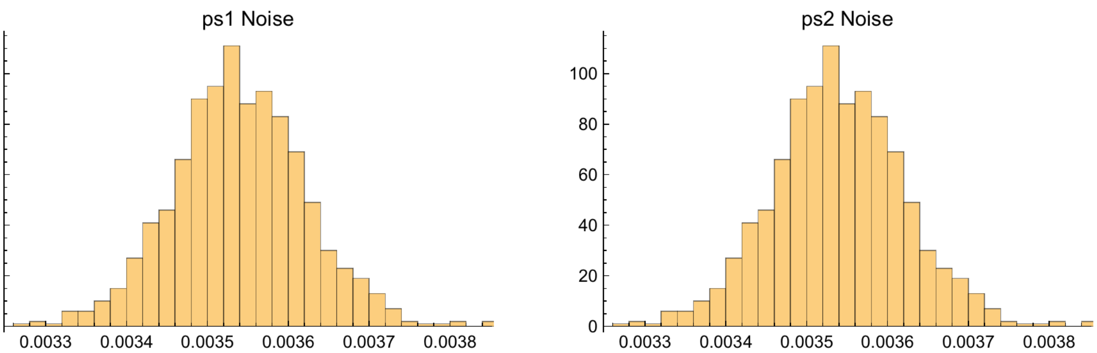
Find Baseline Offset and Noise

In[135]:=
```
GraphicsRow[{Histogram[blps1, PlotLabel → "ps1 Baseline"],
  Histogram[blps2, PlotLabel → "ps2 Baseline"],
  Histogram[blmcp, PlotLabel → "MCP Baseline"]}]
GraphicsRow[{Histogram[noiseps1, PlotLabel → "ps1 Noise"],
  Histogram[noiseps2, PlotLabel → "ps2 Noise"],
  Histogram[noisemcp, PlotLabel → "MCP Noise"]}]
```

Out[135]=



Out[136]=



Baseline Subtract and Apply mild filtering (assumes~5 % noise level).

In[137]:=
```
subps1 = Table[-(picosec1[[i]] - blps1[[i]]), {i, nevents}];
subps2 = Table[-(picosec2[[i]] - blps2[[i]]), {i, nevents}];
submcp = Table[-(mcp[[i]] - blmcp[[i]]), {i, nevents}];

noiseraw = RootMeanSquare[Take[subsi[[1]], 750]];
noisefilt = RootMeanSquare[WienerFilter[Take[subsi[[1]], 750], 2, .05]];
noisefilt2 = RootMeanSquare[WienerFilter[Take[subsi[[1]], 750], 3, .05]];
Print[" raw noise=", noiseraw,
  " filter 2 bin= ", noisefilt, " filter 3 bin= ", noisefilt2];
```

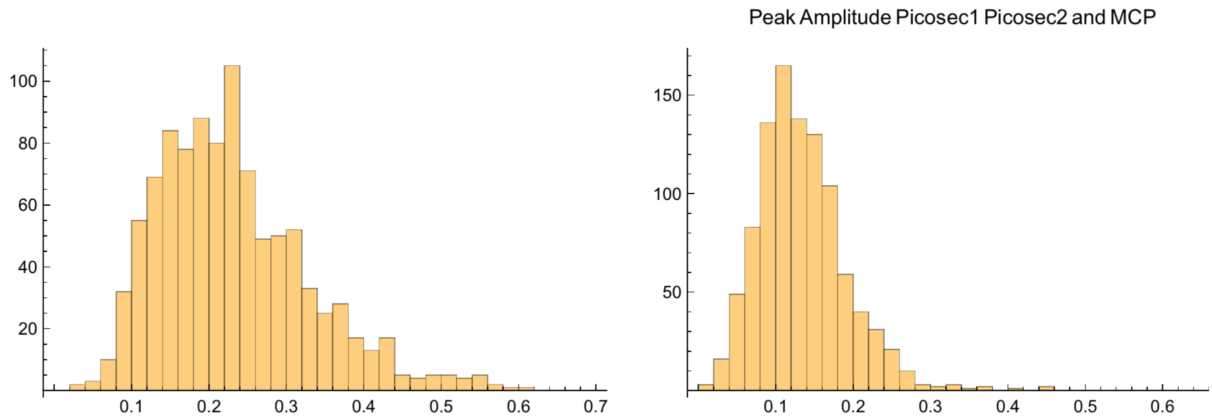
```
raw noise=0.00427136 filter 2 bin= 0.0040679 filter 3 bin= 0.00392634
```

In[140]:=
```
peakps1 = Table[Max[subps1[[i]]], {i, nevents}];
peakmcp = Table[Max[submcp[[i]]], {i, nevents}];
peakps2 = Table[Max[subps2[[i]]], {i, nevents}];
meanmcp = Mean[peakmcp];
rmsmcp = RootMeanSquare[peakmcp - meanmcp] / meanmcp;
Print[" mcp mean= ", meanmcp, "  mcp rms/mean= ", rmsmcp];
```

```
mcp mean= 0.442216  mcp rms/mean= 0.346456
```

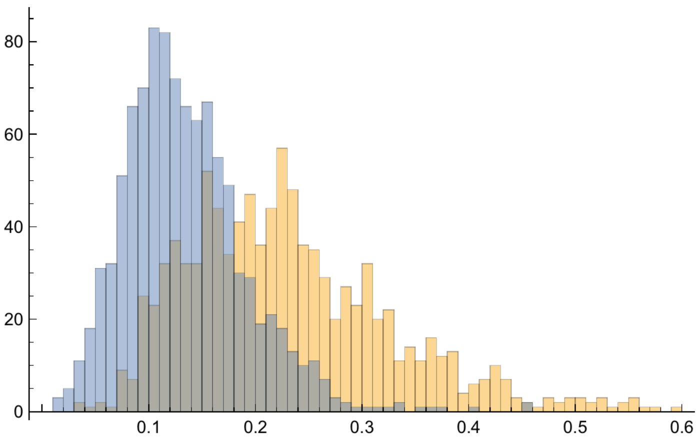
In[229]:=
```
GraphicsRow[
 {Histogram[peakps1, {0.00, 0.7, .02}], Histogram[peakps2, {0.00, 0.7, .02}],
  Histogram[peakmcp, {0, 0.9, .05}, AxesOrigin → {0, 0}]},
 PlotLabel → HoldForm[Peak Amplitude Picosec1 Picosec2 and MCP],
 LabelStyle → {GrayLevel[0]}, ImageSize → Large]
```

Out[229]=



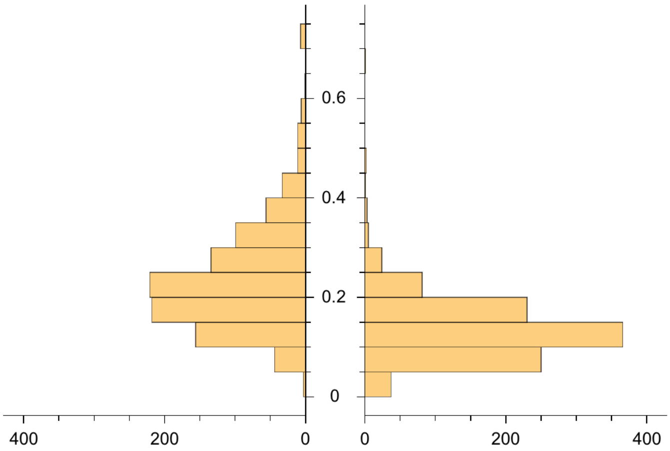
In[232]:=
```
Histogram[{peakps1, peakps2}, {0, 0.6, 0.01}]
```

Out[232]=



In[233]:= `PairedHistogram[peakps1, peakps2]`

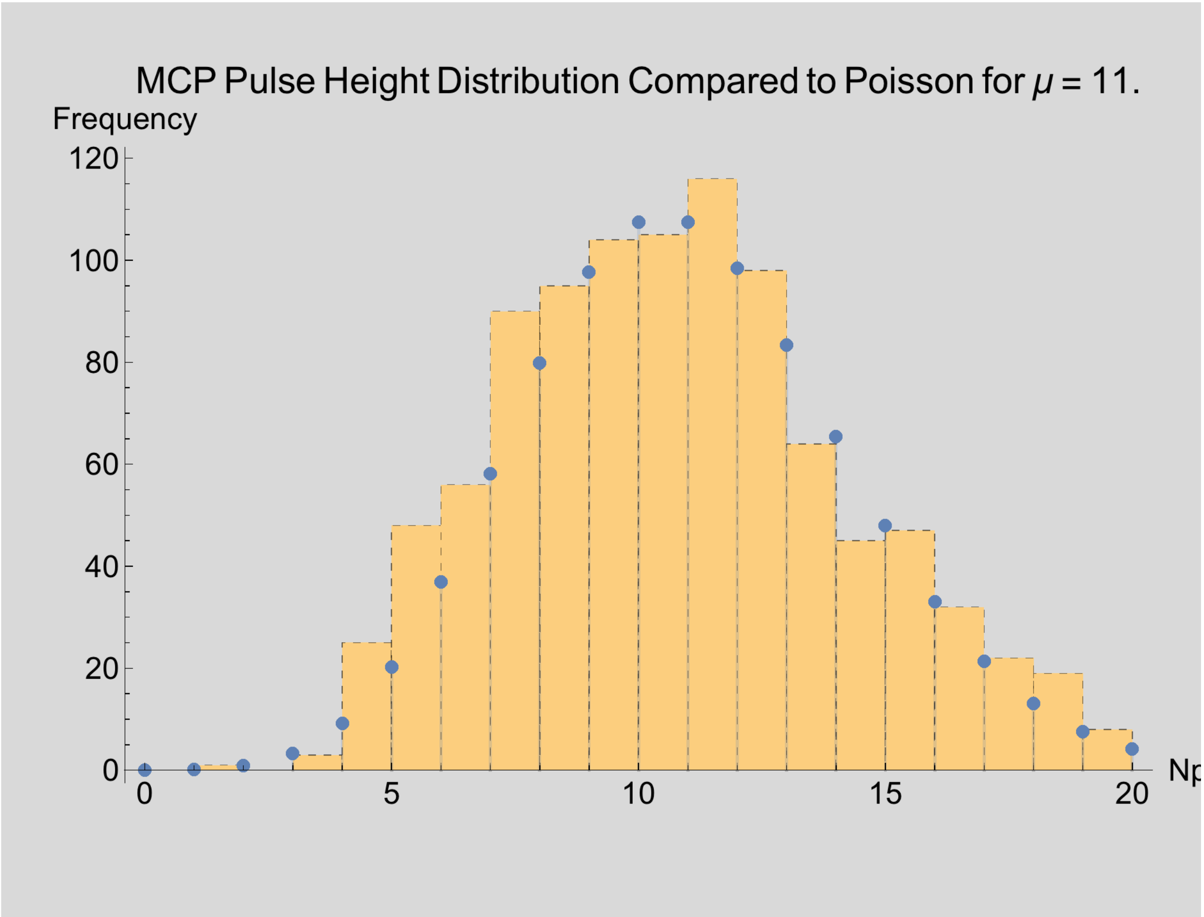
Out[233]=




```
(*edist=EstimatedDistribution[peakmcp/.04, PoissonDistribution[μ]];*)
```

In[182]:=
```
Show[Histogram[peakmcp / .04, {0, 20, 1}, ChartBaseStyle → EdgeForm[Dashed]],
 DiscretePlot[900 * PDF[PoissonDistribution[11], x], {x, 0, 20},
  PlotStyle → PointSize[Large]], ImageSize → Large, AxesLabel →
  {HoldForm[Npe (rescaled MCP Pulse Height)], HoldForm[Frequency]}, PlotLabel →
  HoldForm[MCP Pulse Height Distribution Compared to Poisson for μ = 11.0`],
 LabelStyle → {16, GrayLevel[0]}, AxesStyle → Black, Background → LightGray]
```

Out[182]=



Now Align mcp and Si data relative to the ~30 nsec jitter of the beam scintillator trigger. We get approximate offset using the time bin (50 nsec wide) with the maximum MCP digitized amplitude.

In[146]:=
```
t0 = Table[
    tt[[Flatten[Position[submcp[[i]], Max[submcp[[i]]]]]][[1]], {i, nevents}];
nt0 = Table[Flatten[Position[submcp[[i]], Max[submcp[[i]]]]][[1]],
    {i, nevents}];
t1 = Table[tt[[Flatten[Position[subps1[[i]], Max[subps1[[i]]]]]][[1]],
    {i, nevents}];
nt1 = Table[Flatten[Position[subps1[[i]], Max[subps1[[i]]]]][[1]],
    {i, nevents}];
```

In[150]:=
```
ListPlot[Transpose[{(t0 - t1), t1}]]
```

Out[150]=