EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE

CERN - PS DIVISION

PS/RF/Note 2002-124

ANALYSIS OF BUNCH-TO-BUNCH FLUCTUATIONS IN THE BUNCH TRAIN FOR THE LHC

S. Hancock

Abstract

The role of the PS Complex in the filling of the LHC is to provide a train of 72 bunches within tightly defined beam specifications. Not least among these is the maximum acceptable bunch-to-bunch fluctuation in peak current along the bunch train. This note describes an algorithm to identify which steps in the production process are responsible for the variations observed in the delivered beam. Its aim is to provide a starting point for the introduction of an online diagnostic as part of the bunch shape measurement application.

Geneva, Switzerland
July 2002

1 Introduction

In the PS machine, the beam for the LHC is produced by chopping up the bunches from the Booster in a series of longitudinal splitting steps. Intensity fluctuations along the resultant 72-bunch train must be controlled in order to avoid excursions in luminosity for the individual bunch crossings inside the experiments.

Fourier analysis of the measured beam current sampled over one PS turn is complicated by the spurious issues of bunch shape variation, noise, baseline droop and the gap in the bunch train for the extraction kicker. It is far more informative to analyse the frequency content of the simple list of 72 bunch intensity values.

It is hoped that this note, which describes in detail just such an analysis, will provide a starting point for online monitoring of the bunch-to-bunch reproducibility of the LHC beam.

2 Data Treatment

The algorithm was developed using Mathematica and extracts of the original code appear throughout this note. Readers who are not familiar with Mathematica can largely ignore these as they are intended only to reinforce the explanations provided in the text.

2.1 Baseline Subtraction

If the bunch data are sampled at sufficiently short intervals, the peak of each bunch could simply be taken as the largest data point from each bucket. If this is not the case, then such an approach would lead to an overestimation of the spread in peak values. The alternative is to make a fit to each bunch and a gaussian fit, whilst robust, cannot deal with negative values. Hence the need for baseline subtraction.

The sampled data are assumed to span more than 72 bucket lengths so that the baseline can be estimated from the values acquired in the empty buckets of the kicker gap. The obvious candidate to provide such data in the PS is the bunch shape measurement (BSM) application. This digitizes the signal from a wideband resistive-wall monitor, providing some 15,000 data points at a sampling interval of 0.2 ns. Figure 1 shows an example dumped to file.

```
datalength=14996; dtsample=0.2 10^-9;
filename = "/afs/cern.ch/user/sbh/public/roland/display72.dat";
stream = OpenRead[filename]; Skip[stream, Record, 16];
bsmdata = Read[stream, Table[Number, {datalength}]]; Close[stream];
```

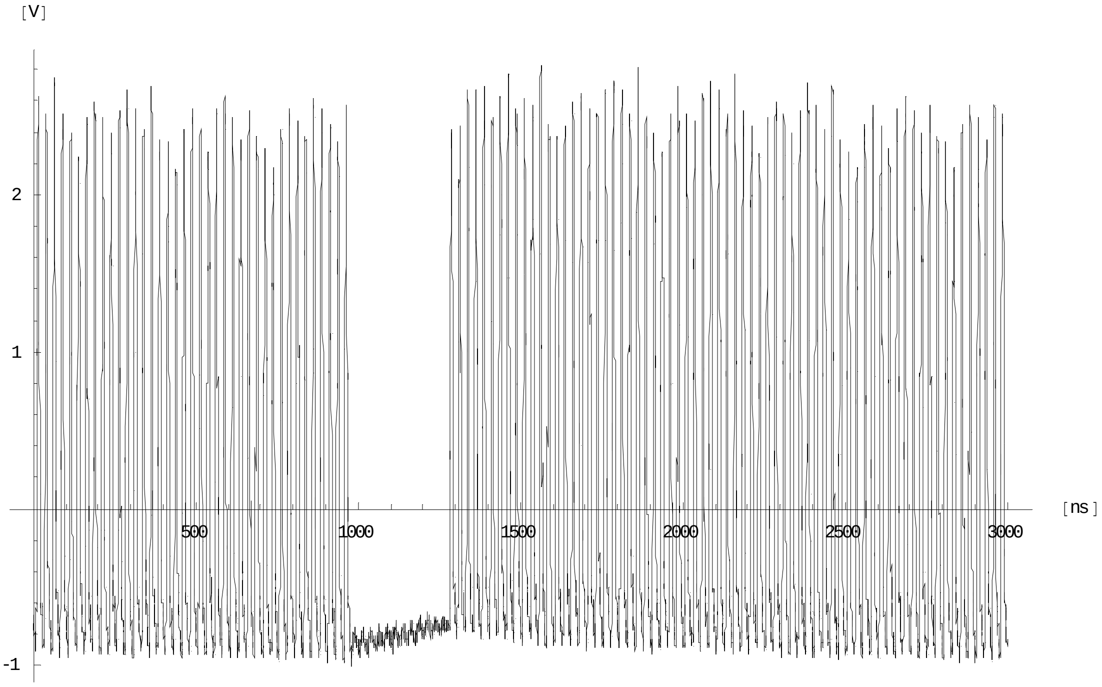

Figure 1: Plot of the contents of the file /afs/cern.ch/user/sbh/public/roland/display72.dat

It is not easy to establish a single representative baseline for such data. A rough first approximation is obtained from the mean of those points that lie below a certain value.

```
a1 = Max[bsmdata]; a0 = Min[bsmdata];
mean[list_] := Apply[Plus,list] / Length[list];
bsmdata -= mean[ Select[bsmdata, (# < a0 + 0.05 (a1-a0)&)] ];
```

5% is an arbitrary threshold. The next step is to define a generic gaussian fit over one bucket length. This requires a knowledge of the relevant rf frequency, which is just above 40 MHz.

```
frf = 40.053 10^6; xrf = 1/(frf dtsample); halfrf = Floor[xrf/2];
bucketdata[xc_] := Take[bsmdata, {xc-halfrf,xc+halfrf}];
<<Statistics`NonlinearFit`;
gaussianparameters[m1_] := BestFitParameters /. NonlinearRegress[
  bucketdata[m1],
  amplitude Exp[-(x - (mu-m1+halfrf+1))^2 / (2 sigma^2)],
  x, {{amplitude,a1},{mu,m1},{sigma,s1}}];
```

Here, a1, m1, s1 are the first-guess amplitude, mean and standard deviation and the fit is made over halfrf samples on either side of m1. The value of m1 is passed to the fitting routine, but the other parameters are fixed. Taking the expected standard deviation to be a compromise between the 1 ns at PS extraction and the 3 ns before bunch rotation (in the hope that this data treatment can be applied at any time after the final bunch splitting), the position, x1, of the strongest peak is located.

```
a1 = Max[bsmdata]; s1 = 2. 10^-9 / dtsample;
x1 = mu /. gaussianparameters[First[Flatten[Position[bsmdata, a1]]]];
```

Since the bucket length, xrf, is known, x1 allows the nearest data point to the centre of each bucket to be located.

```
xcentres = Floor[0.5 +
  Join[ Reverse[Range[x1,1,-xrf]], Range[x1+xrf,datalength,xrf] ]];
If[ First[xcentres]-halfrf < 1, xcentres=Drop[xcentres,1] ];
If[ Last[xcentres]+halfrf > datalength, xcentres=Drop[xcentres,-1] ];
```

Care is taken that rounding to the nearest sample does not introduce points outside the range [1,datalength] to this (integer) list. Finally, the baseline adjustment can be refined by subtracting the mean contents of all empty buckets.

```
empty[xc_] := Max[ bucketdata[xc] ] <  0.2 a1;
bsmdata -= mean[Flatten[
  Map[ bucketdata[#]&, Select[xcentres, empty] ]]];
```

A more sophisticated (variable) baseline may also be considered. An empty bucket is defined as one where all data points – from halfrf before the central one to halfrf after it – lie below 20% of the strongest peak. Again, this threshold is somewhat arbitrary.

2.2 Analysis

If the beam is inside specification, the mean bunch amplitude should not fall more than 10% below that of the strongest peak. The gaussian fit is made to the first 72 buckets that are not empty.

```
a1 = 0.9 Max[bsmdata];
fitvalues =
  Map[gaussianparameters, Take[Select[xcentres, !empty[#]&], 72]];
```

A simple histogram of the resultant 72 fitted amplitudes (with respect to their mean) immediately reveals whether or not the beam is inside specification.

```
peaks = amplitude /. fitvalues;
<<Graphics`Graphics`;
Histogram[peaks/mean[peaks] - 1,
  HistogramRange->{-0.25,0.25}, HistogramCategories->20];
```

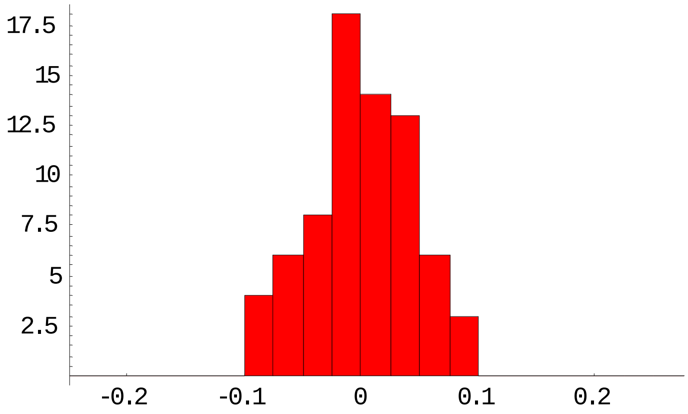

Figure 2: Amplitude spread of the first 72 bunches of Figure 1.

The vertical axis of the histogram shown in Figure 2 is autoscaled, but, arguably, this could be fixed or even omitted. The total count is, of course, 72.

A fast fourier transform reveals the frequencies (in inverse bucket lengths) inherent in the bunch-to-bunch fluctuations. Not only bunch amplitude but also bunch length variations are of interest. So the relevant parameter is the product of the amplitude and standard deviation returned by the fit, since this is a measure of bunch intensity. In order to increase the sharpness of the lines, the list of values is repeated ten times. This is analogous to collecting data over more turns, but without the complication of a kicker gap.

```
fft =
  Abs[Fourier[Flatten[ Table[amplitude sigma /. fitvalues, {10}] ]]];
ListPlot[Transpose[{Range[1,719]/720, Drop[fft,1]}],
  PlotRange->{{0,0.51},{0,0.05 First[fft]}},
  PlotJoined->True, Axes->{True,False}];
```

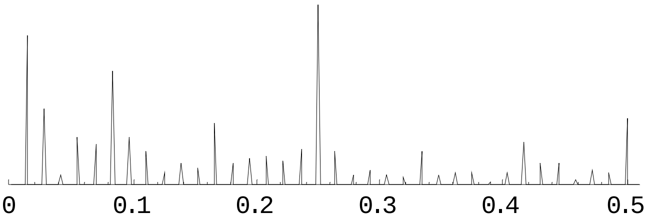

Figure 3: Intensity fft of the first 72 bunches of Figure 1.

The vertical scale of the fft magnitude plotted in Figure 3 is fixed at 5% of the zero-frequency component, while this line itself is suppressed. The fixed scale makes the interpretation of the plot independent of the total beam intensity. Any asymmetry in the triple splitting produces lines principally at 1/12 and 1/6 inverse bucket lengths (but also at 1/3 and 5/12). Problems with the successive double splittings show up purely at 1/4 and 1/2, respectively. All other frequencies can be traced back to the Booster. Depending on which rings supply more beam and which rings are repeated for the second injected batch, a line at 1/72 and its first four harmonics (plus many others) will appear.

A complete data treatment based on these ideas is given in the Appendix.

Acknowledgements

I am grateful to Jose-Luis Sanchez Alvarez for carefully reading and providing comments on the first draft of this document.

Appendix

```
BeginPackage["Display72`"];
Needs["Statistics`NonlinearFit`"]; Needs["Graphics`Graphics`"];
display72::usage = "Please consult Steve Hancock.";
Begin["`Private`"];

display72[filename_String] := Module[
  {datalength=14996,dtsample=0.2 10^-9,frf=40.053 10^6},

  mean[list_] := Apply[Plus,list] / Length[list];
  gaussianparameters[m1_] := BestFitParameters /. NonlinearRegress[
    bucketdata[m1],
    amplitude Exp[-(x - (mu-m1+halfrf+1))^2 / (2 sigma^2)],
    x, {{amplitude,a1},{mu,m1},{sigma,s1}}];
  bucketdata[xc_] := Take[bsmdata, {xc-halfrf,xc+halfrf}];
  empty[xc_] := Max[ bucketdata[xc] ] <  0.2 a1;
  Off[NonlinearRegress::infail];

  stream = OpenRead[filename]; Skip[stream, Record, 16];
  bsmdata = Read[stream, Table[Number, {datalength}]]; Close[stream];
  a1 = Max[bsmdata]; a0 = Min[bsmdata];
  bsmdata -= mean[ Select[bsmdata, (# < a0 + 0.05 (a1-a0)&)] ];
  a1 = Max[bsmdata]; s1 = 3. 10^-9 / dtsample;
  xrf = 1/(frf dtsample); halfrf = Floor[xrf/2];
  x1 = mu /. gaussianparameters[ First[Flatten[ Position[bsmdata, a1] ]] ];
  xcentres = Floor[0.5 +
    Join[ Reverse[Range[x1,1,-xrf]], Range[x1+xrf,datalength,xrf] ]];
  If[ First[xcentres]-halfrf < 1, xcentres=Drop[xcentres,1] ];
  If[ Last[xcentres]+halfrf > datalength, xcentres=Drop[xcentres,-1] ];
  bsmdata -= mean[Flatten[
    Map[ bucketdata[#]&, Select[xcentres, empty] ]]];
  a1 = 0.9 Max[bsmdata];
  fitvalues = Map[gaussianparameters, Take[Select[xcentres, !empty[#]&], 72]];
  peaks = amplitude /. fitvalues;
  fft = Abs[Fourier[Flatten[ Table[amplitude sigma /. fitvalues, {10}] ]]];
  y1 = 0.05 First[fft];
  Show[GraphicsArray[{
    { Histogram[peaks/mean[peaks] - 1,
        PlotLabel->StyleForm["Amplitude Spread", FontSize->16],
        HistogramRange->{-0.25,0.25}, HistogramCategories->20,
        ApproximateIntervals->True, DisplayFunction->Identity
      ]
    },
    { ListPlot[Transpose[{Range[1,719]/720, Drop[fft,1]}],
        PlotRange->{{0,0.51},{0,y1}},
        PlotLabel->StyleForm["Intensity FFT", FontSize->16],
        PlotJoined->True, Axes->{True,False}, DisplayFunction->Identity,
        Epilog->{Hue[0],
          Text["PSB", {3/72, 0.9 y1}, {0,-1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {1/72, 0.9 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {1/36, 0.9 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {3/72, 0.9 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {1/18, 0.9 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {5/72, 0.9 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["3-Split", {3/24, 0.8 y1}, {0,-1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {1/12, 0.8 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {1/6, 0.8 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["1st 2-Split", {1/4, 0.9 y1}, {0,-1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {1/4, 0.9 y1}, {0,1}, TextStyle->{FontSize->14}],
          Text["2nd 2-Split", {1/2, 0.8 y1}, {0,-1}, TextStyle->{FontSize->14}],
          Text["\[DownArrow]", {1/2, 0.8 y1}, {0,1}, TextStyle->{FontSize->14}]
        }
      ]
    }}],
    AspectRatio->1, ImageSize->{600,650}
  ];
  Return[]];

End[];
EndPackage[]
```

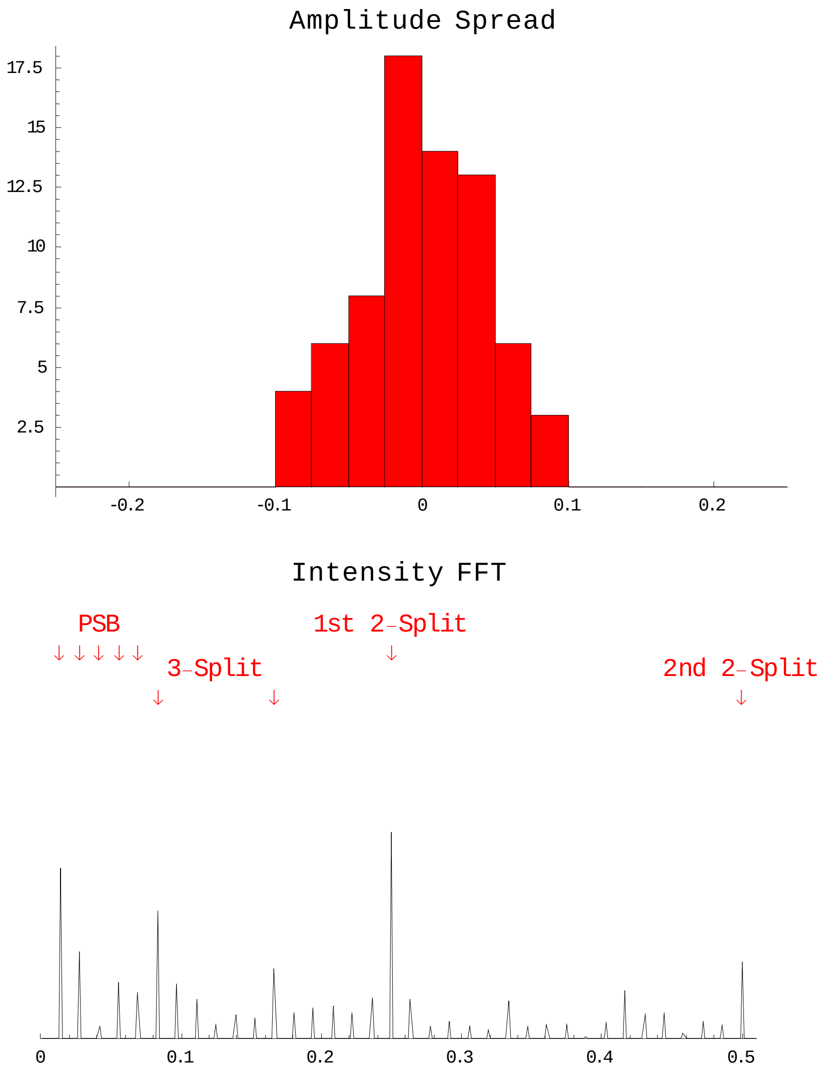

Figure 4: Complete treatment of the data of Figure 1.