GeneratorWarmup

January 9, 2016

1 Does it matter Part 7: Generator warm-up

Many of my mentors have insisted that the X-ray generator needs some time to stabilize after setting the generator voltage and current. The given waiting times vary from half an hour to about an hour, during which one cannot expect a stable measurement (wasting energy in the process). However, is this really an issue?

To find out, I started a measurement right after setting our old-ish Philips PW1830 to the operating conditions (40 kV, 40 mA). This measurement consisted of 180 repetitions of 10 seconds. The final Anton Paar .pdh file is pretty useless, as it has averaged the data over the 180 repetitions, but fortunately they supply a .tif image file.

Upon closer inspection, the Anton Paar Tif file contains the raw data from the measurement (in counts). The image is a multi-frame .tif image, with one frame per repetition. Fortunately, reading this image appears to be problem-free, although since the image contains only 16-bit values, and the Mythen detector can store up to 24 bits in its registers, I wonder what will be stored in the .tif file if the amount of detected counts in one pixel exceeds 65k...

Anyway, back to the problem at hand: do we see any evidence in the patterns that would support the practice of "warming up" the generator? Let's start up Python and see what we get...

1.1 Start

We start by setting up the Python3 notebook environment the way I like it:

```
In [36]: # initialization:
         import h5py, os, pandas, time, scipy, glob
         import numpy as np

         %matplotlib inline
         import matplotlib
         import matplotlib.pyplot as plt
         from PIL import Image
         from scipy import stats
```

1.2 Data Read-in

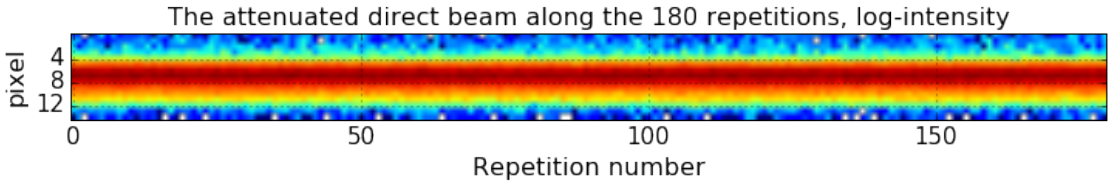
Now, we can read in the .tif image that is produced when we ask our Anton Paar SAXSess to measure 180 repetitions of 10s frames. For this, we use the PIL library, which is an image library that can read a variety of images. We extract the individual 1-by-1280 pixel frames and put them alongside:

```
In [40]: pillowImage = Image.open("S4581 warmup.tif")
         imDat = list()
         for framei in range(pillowImage.n_frames):
             pillowImage.seek(framei)
             imDat.append(np.array(pillowImage.getdata()))
         imDat = np.array(imDat, dtype = float)
```

1.3 Initial data evaluation

Let's see what the data looks like, focusing in on the 15 pixels around the direct beam signal:

```
In [41]: plt.figure(figsize = [12,4])
         plt.imshow(np.log10(imDat[:,140:155]).transpose())
         plt.title("The attenuated direct beam along the 180 repetitions, log-intensity", fontsize = 16)
         plt.xlabel("Repetition number", fontsize = 16)
         plt.ylabel("pixel", fontsize = 16)
         plt.xticks(fontsize = 15)
         plt.grid("on")
         plt.yticks([4, 8, 12], fontsize = 15)
         plt.savefig("directbeam.png")
         plt.savefig("directbeam.pdf")
```



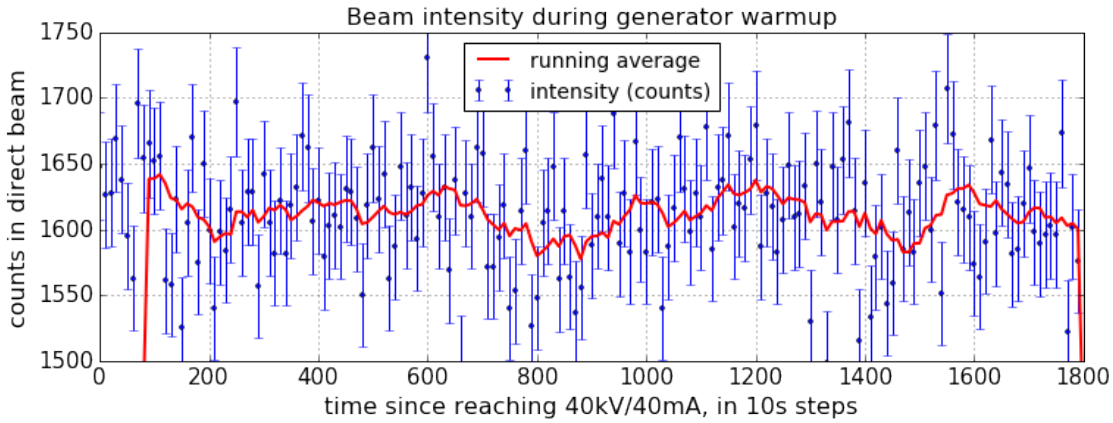

```
In [46]: intensity = imDat[:,140:155].sum(axis =1)
         smoothed = np.convolve(intensity, np.ones(10)/10)
```

This looks fairly uniform over time, with no obvious time-dependent beam shifts or drastic intensity variations. Let's take a closer look at the intensity.

1.4 The beam intensity

We now look at the summed intensity in a given image over these 15 pixels. At the same time, we plot a ten-sample running average to get a better idea of possible trends.

```
In [43]: plt.figure(figsize = [12,4])
         plt.errorbar(np.arange(0, 1800, 10), intensity, label = "intensity (counts)", yerr = np.sqrt(i
         plt.plot(np.arange(0, 1890, 10), smoothed, "r-", label = "running average", lw = 2)
         plt.ylim(1500,1750)
         plt.xlim(0,1800)
         plt.xticks(fontsize = 15)
         plt.yticks(fontsize = 15)
         plt.grid("on")
         plt.xlabel("time since reaching 40kV/40mA, in 10s steps", fontsize = 16)
         plt.ylabel("counts in direct beam", fontsize = 16)
         plt.title("Beam intensity during generator warmup", fontsize = 16)
         plt.legend(loc=0, fontsize = 14)
         plt.savefig("intensityplot.png")
         plt.savefig("intensityplot.pdf")
```

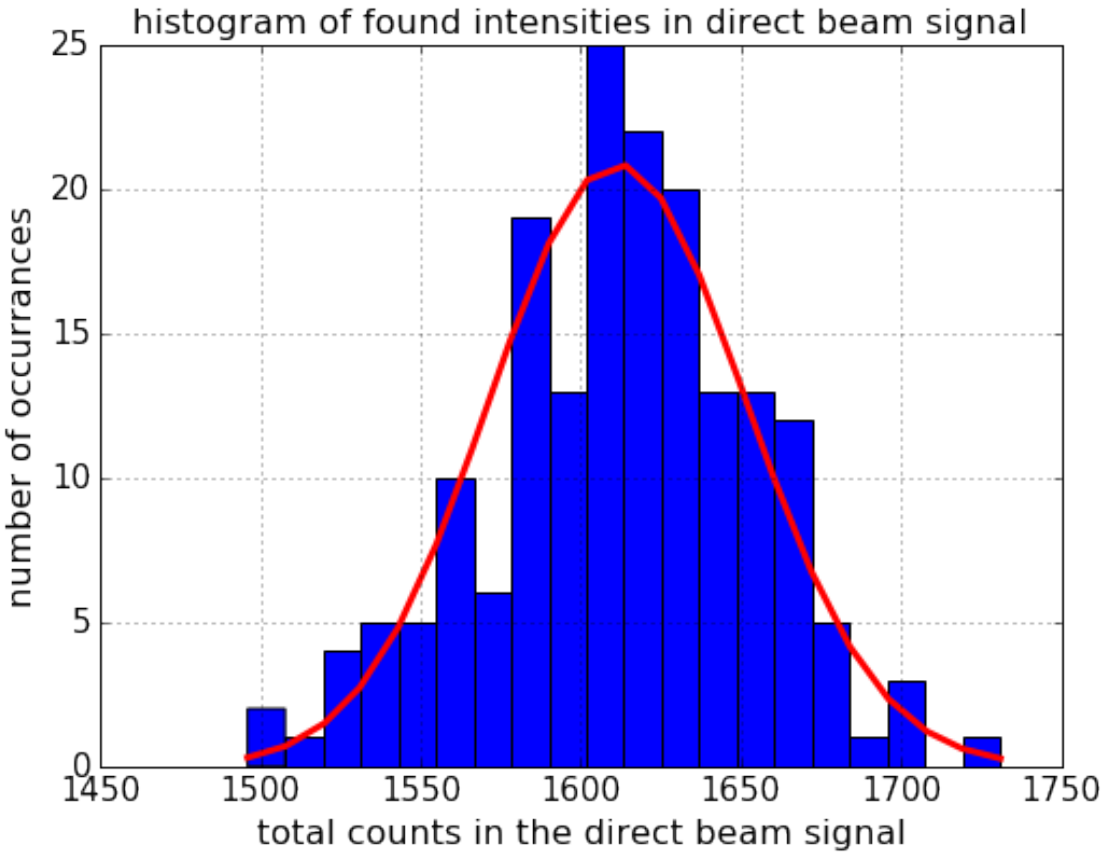


1.5 Intermediate conclusion:

From the looks of it, there appears to be little to no trends in the beam intensity over time. The running average does show some fluctuation, but whether it is significant or not is hard to tell.

One way of determining whether the deviations are significant is to use our good friend: the Poisson distribution. If the fluctuations we are seeing are just due to the counting uncertainty of random events, all the counts should fall within a Poisson distribution. If, however, there is an additional contribution to the fluctuation, for example an effect from warm-up or other generator-instabilities, then the distribution of counts should exceed that of a Poisson distribution.

```
In [47]: plt.figure(figsize = [8, 6])
         y, x, dummy = plt.hist(intensity, 20)
         plt.plot(x, stats.poisson.pmf(x, intensity.mean(), loc = intensity.mean()))
         pStat = stats.poisson(intensity.mean())
         # plt.plot(x, pStat.cdf(x))
         plt.plot(np.round(x), pStat.pmf(np.round(x)) * 2100, lw = 3) # pmf for discrete functions seem
         plt.title("histogram of found intensities in direct beam signal", fontsize = 16)
         plt.xlabel("total counts in the direct beam signal", fontsize = 16)
         plt.ylabel("number of occurrances", fontsize = 16)
         plt.grid("on")
         plt.xticks(fontsize = 15)
         plt.yticks(fontsize = 15)
         plt.savefig("distributionplot.png")
         plt.savefig("distributionplot.pdf")
```



1.6 The Poisson distribution:

Histogramming how many cumulative counts we have in the fifteen pixels of each of the direct beam images gives us the data shown in Figure 3. On average, we find 1612 cumulative counts in the fifteen pixels. We can use this value to plot the Poisson probability mass function for a this mean value. This probability mass function gives us the likelihood of finding a particular number of counts in a single sample for that mean.

As we see, the Poisson probability mass function very nicely describes our data, and there is no indication that any of our findings are due to any other effect. That means that with these experiments, we have not been able to demonstrate the existence of a warm-up effect.

So for now, until we have evidence to the contrary, we may as well save ourselves some Joules and time, and start measuring right after getting our generator up to power. The same may not hold for other generators, but this is an easy method for checking it for yourselves.

In [ ]: