

Nonlinear BPM Positions using Poisson

A. Bartnik

(Dated: September 28, 2018)

Our goal is to find which beam position (x, y) has predicted signals that are as close to the measured button signals V_k as possible. To simplify the problem, exploiting the geometry of our buttons, the first thing we do is calculate the difference-over-sum of the horizontal and vertical buttons, and define these as the new target quantities.

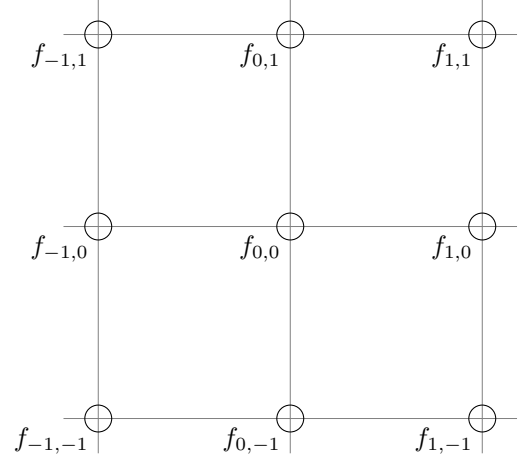
$$q_x = \frac{V_{\text{right}} - V_{\text{left}}}{V_{\text{right}} + V_{\text{left}}}, \quad q_y = \frac{V_{\text{top}} - V_{\text{bottom}}}{V_{\text{top}} + V_{\text{bottom}}}$$

At the same time, the predicted button signals have been computed in advance using Poisson [1] or some other field solver. The method to transform the time-dependent 3D problem of the bunch interacting with the buttons into a much simpler static 2D problem, allowing solution in Poisson, is detailed in [2]. Regardless of their source, these predicted signals are assumed to have been interpolated onto a uniform grid in the region of interest. We form the same difference-over-sum with these values, and call them $Q_x(x_i, y_j), Q_y(x_i, y_j)$, where the indices on x and y are to indicate that it's only evaluated on a 2D grid. We will define the beam position (x, y) as the point that minimizes the following function f .

$$f = (Q_x(x, y) - q_x)^2 + (Q_y(x, y) - q_y)^2$$

where it's understood that (x, y) off the fieldmap grid would be 2D interpolated, as described below. We can start by evaluating f on the fieldmap grid, and just checking every point to see which grid point has the smallest value. This is the simplest possible search, and smarter initial guesses, along with searches over a smaller area of the grid, might reduce the computational time. But, for the purposes here we will just exhaustively search all the points.

As a result, we now know the closest grid point to the desired best fit (x, y) , which we will call x_0 and y_0 . To do better, one needs to minimize a 2D interpolation of f in between grid points. Depending on the order of the interpolation, this has varying levels of complexity. For a bicubic or higher order interpolation, one would probably have to use some standard minimization algorithm on the interpolation function. If a "biparabolic" interpolation is sufficient, then we can find the root analytically, and write it as a simple function of the grid points directly surrounding the (already found) closest grid point. If we call the value of f on this closest grid point $f_{0,0}$, then we have the following surrounding points.



We approximate f in the following form around that point, where δx and δy are now relative to that central point:

$$\chi^2 \approx f_{0,0} + a \delta x + b \delta y + \frac{1}{2} c \delta x^2 + \frac{1}{2} d \delta y^2 + e \delta x \delta y$$

The coefficients can be written:

$$\begin{aligned} a &= \frac{f_{1,0} - f_{-1,0}}{2\Delta_x} \\ b &= \frac{f_{0,1} - f_{0,-1}}{2\Delta_y} \\ c &= \frac{f_{1,0} - 2f_{0,0} + f_{-1,0}}{\Delta_x^2} \\ d &= \frac{f_{1,0} - 2f_{0,0} + f_{-1,0}}{\Delta_y^2} \\ e &= \frac{f_{1,1} + f_{-1,-1} - f_{-1,1} - f_{1,-1}}{4\Delta_x \Delta_y} \end{aligned}$$

where $\Delta_{x,y}$ is the grid spacing in either x or y . The final best fit beam positions (x, y) are as follows.

$$\begin{aligned} x &= x_0 + \frac{ad - be}{e^2 - cd} \\ y &= y_0 + \frac{bc - ae}{e^2 - cd} \end{aligned}$$

Edge cases have to be handled with some care. If any of the points surrounding (x_0, y_0) are outside of the pipe in the fieldmap file, then $f = 0/0$ is ill-defined on those points. The simplest solution in that case is to just choose $x = x_0, y = y_0$, that is, to do nearest-neighbor interpolation. This is a pretty rough approximation, but since

the beam position is apparently at the edge of the pipe, the operator probably doesn't need to know its position better than that.

So, the complexity of this part of the algorithm is tiny—just a few floating point operations to fine tune the best

fit value. All of the real computational burden is in the initial calculation of f on each of the fieldmap grid points, which has to be repeated for every new beam position. Still, if there are only 10s of thousands of grid points, this shouldn't be a problem.

[1] http://laacg1.lanl.gov/laacg/services/download_sf.phtml (2012).

[2] R. W. Helms and G. H. Hoffstaetter, *Physical Review Accelerators and Beams* **8**, 062802 (2005), physics/0406145.