

BPM gain calibration

D. Rubin

(Dated: March 21, 2024)

I. INTRODUCTION

A beam based determination of BPM button gains depends on measurement of the button response for beams at a variety of positions within the active region of the detector. The data can be collected by measuring button response for a large number of distinct closed orbits, or as in the method described in this document, by using turn by turn data for a beam whose trajectory is different on each turn. By driving the beam at the betatron tunes, the trajectory will sample a portion of the active region of the BPM over a range that depends on the shaker drive amplitude. The turn by turn data can be collected much more quickly than a complete set of closed orbit data.

One might then compute the function that maps position to button intensity, $b_i(x, y)$ and invert it to determine x, y for each set of intensities $b_{i=1,4}$. If there is a gain error, that is if the response of one or more of the buttons is different from the response assumed to determine $b_i(x, y)$, then there will be no position x, y that yields the measured b_i . The gain of each button is adjusted to bring the data onto the map.

Alternatively, in the absence of a map, we can exploit the symmetry and redundancy of the four button configuration, to derive a relationship between the button intensities that is good to second order in the displacement of the beam from the center of the BPM. The button gains are then determined by a simple nonlinear least square fit. The analysis for ~ 100 BPMS takes a few minutes.

The simplifications and assumptions that are the basis of the technique for measuring BPM button gain are:

1. The layout of the buttons in the BPM is mirror symmetric about horizontal and vertical axis. A map of button response as a function of beam position is not required.
2. The dependence of button intensity on beam position is given to good approximation to second order in position as long as the displacement of the beam from the center of the BPM is small compared to the distance from center to the electrodes.
3. The beam can be driven to sufficient amplitude by the phase locked shaker so that the beam after many turns will sample a region of at least $\sim \pm 1\text{mm}$ both horizontally and vertically in each BPM.

If the above obtain, then the gains can be measured and analyzed to good precision in minutes, thus allowing for easy repetition, and with luck, convergence.

Sections II-III give a brief overview of the method. See IV for a guide to collecting the turn by turn data and running the code that extracts button gains from that data.

II. MODEL

A brief description of the model is presented. (For details see [Phys. Rev. ST Accel. Beams 9, 011002 \(2006\)](#)). The dependence of the button intensity on position is given by a function

$$B_i = kf(x, y)$$

where (x, y) is the position of the beam and k a constant that depends on bunch current. As long as the beam is near the center of the BPM, x and y are small compared to the distance to the electrodes, and the current independent response function can be expanded to second order

$$B_1 \sim k(c_0 + c_1x + c_y + c_3x^2 + c_4y^2 + c_5xy) \quad (1)$$

Assuming mirror symmetry

$$B_2 = kf(-x, y) \sim k(c_0 - c_1x + c_2y + c_3x^2 + c_4y^2 - c_5xy) \quad (2)$$

$$B_3 = kf(x, -y) \sim k(c_0 + c_1x - c_2y + c_3x^2 + c_4y^2 - c_5xy) \quad (3)$$

$$B_4 = kf(-x, -y) \sim k(c_0 - c_1x - c_2y + c_3x^2 + c_4y^2 + c_5xy) \quad (4)$$

(The button numbering is indicated in Fig. 1.) Taking sums and differences of Eqs. 1-4, gives

$$B_{+---+} \equiv B_1 - B_2 - B_3 + B_4 = 4kc_5xy \quad (5)$$

$$B_{+--+} \equiv B_1 - B_2 + B_3 - B_4 = 4kc_1x \quad (6)$$

$$B_{++--} \equiv B_1 + B_2 - B_3 - B_4 = 4kc_2y \quad (7)$$

Rearranging

$$xy = \frac{1}{4kc_5} B_{+---+}$$

$$x = \frac{1}{4kc_1} B_{+--+}$$

$$y = \frac{1}{4kc_2} B_{++--}$$

Then combining to eliminate x and y gives an expression that simply relates button signals.

$$B_{+---+} = \frac{c_5}{4kc_1c_2} (B_{+--+})(B_{++--}) \quad (8)$$

where c_1 , c_2 and c_5 are constants that characterize the BPM geometry. Each set of button values, regardless of beam position, if combined as defined in Eqs. 5-7, will lie along the line with slope $\frac{c_5}{kc_1c_2}$ and zero offset, according to Eq. 8. As noted above, the relationship derives from the symmetry of the BPM, which in turn requires that all of the buttons have identical gain.

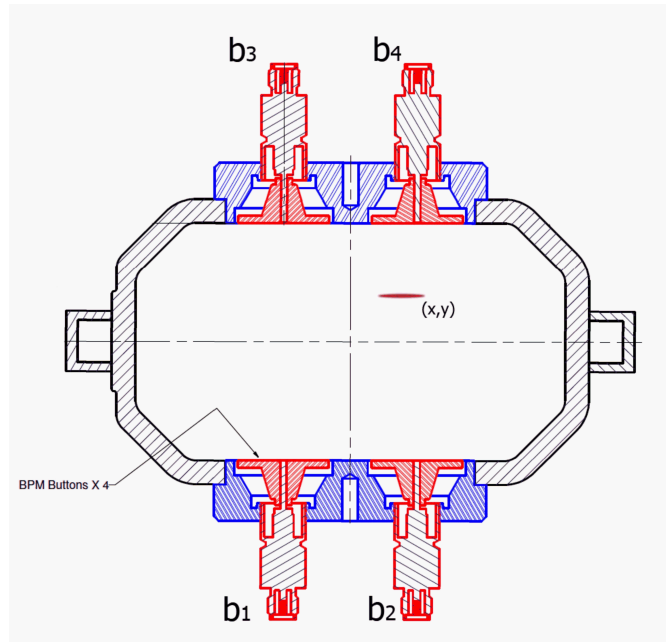


FIG. 1: CESR Beam Position Monitor. Note labeling of buttons $b_1 \dots b_4$

Deviations from the form of Eq. 8 of the $B_{+--+}B_{++--}$ versus B_{+---+} curve indicate gain variations between the electrodes. A nonlinear least squares fit can be used to compute the gains such that the data will best reproduce the Eq. 8 relationship. For notational convenience define the vectors

$$\mathbf{g} \equiv (g_1, g_2, g_3, g_4)$$

$$\mathbf{B}_0 \equiv (B_1, B_2, B_3, B_4)$$

$$\mathbf{B}_{+--+} \equiv (B_1, -B_2, B_3, -B_4)$$

$$\mathbf{B}_{++--} \equiv (B_1, +B_2, -B_3, -B_4)$$

$$\mathbf{B}_{+---+} \equiv (B_1, -B_2, -B_3, B_4)$$

Then to determine the best fit \mathbf{g} , minimize

$$\chi^2 = \sum_i^{orbits} \left((\mathbf{g} \cdot \mathbf{B}_{+---}^i) - \frac{c}{I} (\mathbf{g} \cdot \mathbf{B}_{-+++}^i) (\mathbf{g} \cdot \mathbf{B}_{++--}^i) \right)^2 \quad (9)$$

with respect to the components of \mathbf{g} and c , where c is a geometrical coefficient.

III. NORMALIZATION

χ^2 has a best fit minimum for finite \mathbf{g} but also a trivial minimum if $\mathbf{g} = 0$. The null result is precluded by setting one of the g_i to unity. In practice, in order to limit any systematic effect from our arbitrary selection of a particular choice of reference button, we fit the same data four times, using for reference each of the four buttons respectively, and then normalize and average. That is:

$$\mathbf{g}^j = 4 \frac{\mathbf{g}^j}{\sum_i g_i^j}$$

$$\langle \mathbf{g} \rangle = \sum_{j=1}^4 \frac{\mathbf{g}^j}{4}$$

where for g_i^j the lower (i) is the button index and the upper (j) the index of the button set to unity gain.

Finally note that the quantities $\langle g_i \rangle$ are technically the inverse of the button gains. A particular fitted g_i is greater than unity, if the intensity of B_i is smaller than the others. That is, a fitted g_i greater than 1 corresponds to a button with gain less than 1. Therefore the real BPM button gains are given by

$$G_i = 1/\langle g_i \rangle \quad (10)$$

IV. INSTRUCTIONS

A. Collecting Turn by Turn Data

Store a single bunch with bunch current that gives best BPM resolution. Flatten the orbit. Lock the tune trackers at the betatron tunes. Increase the shaker drive amplitudes until the bunch gets knocked out and then back off a bit. Take turn by turn data for some large number of turns. Since the fitting is quick, you can experiment with the bunch current, shaker amplitudes, tunes, number of turns etc. In general, the fit quality tends to improve with the number of turns and amplitude. The width and height of the region sampled in each BPM will be proportional to $\sqrt{\beta_h}$ and $\sqrt{\beta_v}$ respectively. If width and height are very small, (typically the case where β 's are small), there is insufficient redundancy in the data to get a good fit. Collect turn by turn data with the `CESRV` command `TAKE TBT` and note the `TBT file number`. An example of such turn by turn data files was collected June 13, 2023 with the digital tune tracker locked and both horizontal and vertical gain set to maximum achievable with 0.75 mA in a single bunch. A summary of the log entry [2312](#) follows

```
Tune tracker gain: H: 0.0 db, V: -28db, 0.75mA
8192 turns 103802/03
16384 turns 103804/05
```

```
H: 0.0 dB, V: -25.9dB 0.62mA (vertical gain is the limit, increase more will lose the beam)
16834 turns 103806/07
16384 turns 103808/09
```

B. Gain fitting

The code that fits the turn by turn data and computes gains resides in the directory

`/home/dlr/development9_linux/BPM_tbt_gain/gain_tbt` and `/home/dlr/development9_linux/BPM_tbt_gain/code`.

Consecutive turn by turn data files for west and east BPMs respectively are read with the help of

```
subroutine tbt_read_data_file
```

found in

```
File: /nfs/acc/libs/Linux_x86_64_intel/current/mpm_utils/tbt/tbt_mod.f90
```

C. Example

Before running the program to process the data, it is convenient to create a new working directory, for example (`gain_test_dir`). From within that directory run `/nfs/acc/user/dlr/development9_linux/production/bin/gain_tbt`. (That way the considerable output will be easy to delete when no longer of interest.)

The program output to the terminal with appropriate responses to the prompts is reproduced below.

```
[dlr10@lnx6186 gain_test_dir]$ /nfs/acc/user/dlr/development9_linux/production/bin/gain_tbt
Expert mode ? n
```

For the time being we will work in the standard non-expert mode. Expert mode allows among other things, the possibility of reading simulated turn by turn data.

```
file_name = 106680
Turn by turn data file ? (def = 106680) 103805
```

The default file name is the most recently written tbt data file. Sometimes that is convenient. But we prefer to specify the tbt data file 103805. The previous file, that contains the data for the BPMs in the west (103504), will be read automatically.

```
file_name = 103805
number =      103805
in_file = RD-103805.dat true number =      103805
tbt_file_type =      3
Correct tbt data for gain and pedestal ?
(1=no_correct, 2=pedestal correct, 3=gain and pedestal correct ) 1
```

If pedestal data was collected and/or a gain correction has been previously determined, those corrections can be applied. We assume for the moment that there is no pedestal or gain information.

```
gain correct =      1
CORRECTING OFF-BY-ONE PROBLEM...
Note: Hard wired bpm names for south arc bperms.
SHAKER PHASE MISMATCH!   66 :   16337   8916   506 :   16362   8102   507
SHAKER PHASE MISMATCH!   67 :   16357   4603   511 :   16353   2317   511
SHAKER PHASE MISMATCH!   68 :   16336    539   506 :   16330   9072   510
SHAKER PHASE MISMATCH!   69 :   16312  15672   510 :   16365  15547   506
SHAKER PHASE MISMATCH!   70 :   16317   7340   508 :   16350    299   507
SHAKER PHASE MISMATCH!   71 :   16352   4244   511 :   16352   2317   511
SHAKER PHASE MISMATCH!   72 :   16296   3411   510 :   16362   3613   509
SHAKER PHASE MISMATCH!   74 :   16306   4469   508 :   16329   8721   509
SHAKER PHASE MISMATCH!   75 :   16332   4469   507 :   16351   7925   508
SHAKER PHASE MISMATCH!   76 :   16367   3529   511 :   16343   2317   511
SHAKER PHASE MISMATCH!   80 :   16326   4847   504 :   16356    828   510
SHAKER PHASE MISMATCH!   81 :   16306   5626   508 :   16371  15796   509
SHAKER PHASE MISMATCH!   82 :   16354   1061   511 :   16345   1912   511
SHAKER PHASE MISMATCH!   84 :   16358   2653   510 :   16354    119   510
SHAKER PHASE MISMATCH!   86 :   16353  14520   497 :   16358   9541   503
SHAKER PHASE MISMATCH!   97 :   16331   3720   508 :   16367  11517   509
Success reading file 103805
CORRECTING OFF-BY-ONE PROBLEM...
Success reading file 103804
Bunch number? 1
```

The data was collected for bunch 1.

```
bunch = 1
Pedestal bunch number ? (0 if none)0
```

There is no pedestal data.

```
pedestal bunch = 0
  There is no pedestal bunch
Norbits = 16383
Number of orbits so far 16383
Another RD file ? (y/n) n
```

Here it is possible to include another tbt file. We will stick with the single pair that has already been read.

```
which bpm ? (-1 for all)-1
```

We want to compute gains for all BPMs.

```
Data written to file RD-103805.dat_out
```

Then follows lots of information that is useful for debugging, and a plot of fitted gains at each of the beam position monitors. (Fig. ??). Note that for many (most?) of the BPMs the gain corrections are greater than 20% and offscale in Fig. ??, most likely indicating a poor fit. There are a number of reasons for a poor fit:

- The region sampled by the orbits in the BPM is insufficient in width or height
- The sampled region is displaced from the center of the BPM
- There are an insufficient number of measurements
- The BPM buttons are mis-wired
- ?

D. Output

The `gain_tbt` program writes copious output files.

- `RD-103805.dat_out`: The region of the BPM sampled by the complete set of turn by turn orbits is bounded horizontally by $h_{min} : h_{max}$ and vertically by $v_{min} : v_{max}$. The bounds, their difference, the fitted gains, the constant $k \frac{c_5}{c_1 c_2}$, and the fit χ^2 for each BPM are listed.
- `BPM.n.dat` The file includes intensity on each button, position, the scalar products $(\mathbf{g} \cdot \mathbf{G}_{+---}), (\mathbf{g} \cdot \mathbf{B}_{+--+}), (\mathbf{g} \cdot \mathbf{B}_{++--})$ and $(\mathbf{g} \cdot \mathbf{B}_{+--+})(\mathbf{g} \cdot \mathbf{B}_{++--})$, for $\mathbf{g} = \mathbf{1}$ for the the n^{th} BPM for each measurement. The product $(\mathbf{g} \cdot \mathbf{B}_{+--+})(\mathbf{g} \cdot \mathbf{B}_{++--})$ versus $\mathbf{g} \cdot \mathbf{B}_{+---}$ for BPM 90 (Eq. 8) is plotted in Fig. ??(left) in purple. The position is shown in Fig. ??(right).
- `BPM.n.dat_fit` contains $(\mathbf{g} \cdot \mathbf{G}_{+---}), (\mathbf{g} \cdot \mathbf{B}_{+--+}), (\mathbf{g} \cdot \mathbf{B}_{++--})$ and the product $(\mathbf{g} \cdot \mathbf{B}_{+--+})(\mathbf{g} \cdot \mathbf{B}_{++--})$, for the fitted gains \mathbf{g} for the n^{th} BPM. The product $(\mathbf{g} \cdot \mathbf{B}_{+--+})(\mathbf{g} \cdot \mathbf{B}_{++--})$ versus $\mathbf{g} \cdot \mathbf{B}_{+---}$ for BPM 90 is plotted in Fig. ??(left) in green.

V. COMMENTS

The range of measurable beam positions in each BPM is determined by the shaker drive amplitude. The shaker drive amplitude is limited by the smallest effective aperture in the storage ring. Ever since the first narrow gap undulator was installed in CESR, it has been difficult to get sufficient range on all (most) of the BPMs to get good fits. Indeed if you plot data for other BPMs as in Fig. ??, you will find that BPM_90 is not typical. This limited range might be remedied by collecting more turns, but there may be fluctuations in the timing due to the large shaking amplitude that compromise the single pass resolution. It may well be that the only way to map a reasonably sized grid in many of the BPMs will require closed bumps.

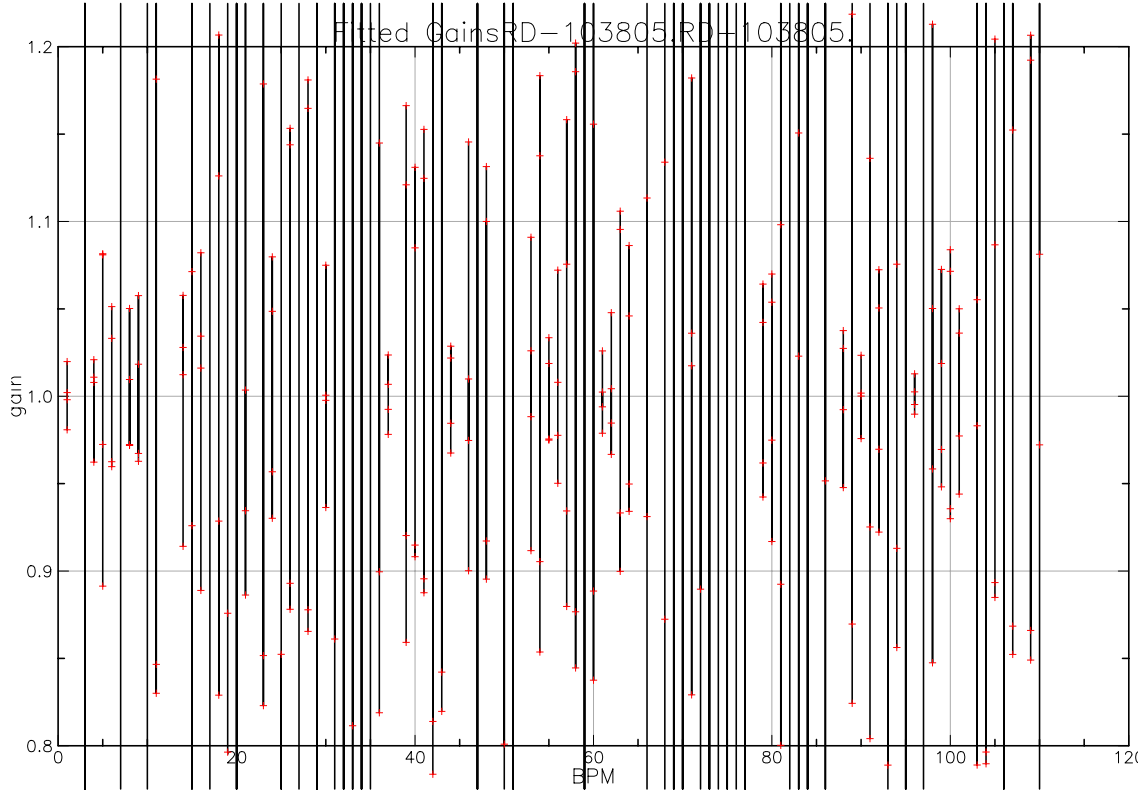


FIG. 2: The fitted gains of each of the four buttons are indicated by the red crosses. For example, at BPM 90, the gains are 1.023, 1.0, 0.976, and 1.002. For many of the BPMs the fitted gains are off scale, indicating an unreliable fit.

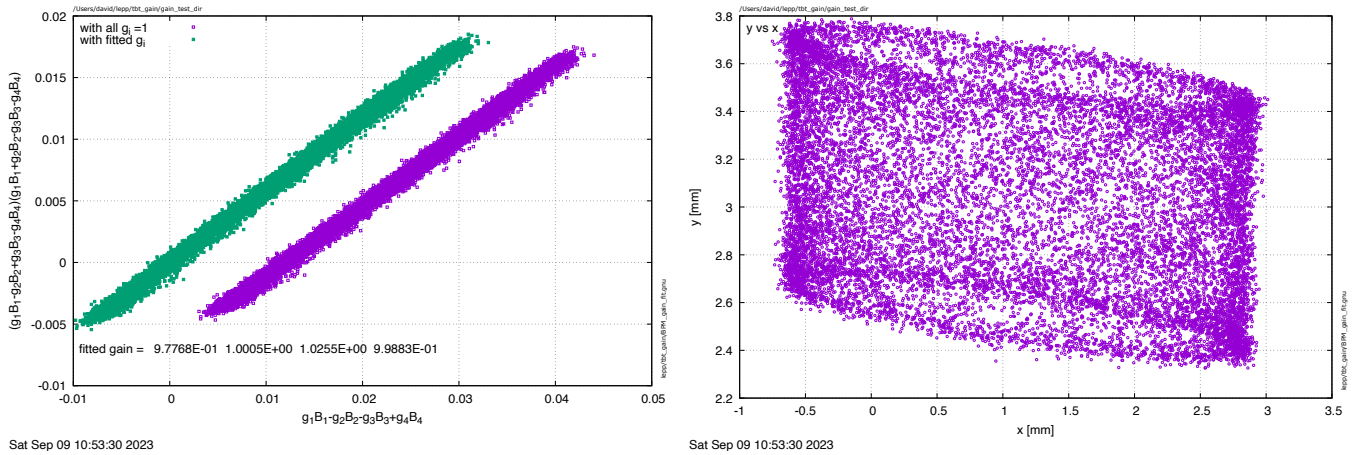


FIG. 3: (left) The right side of Eq. 8 is plotted on the vertical axis and the left side on the horizontal axis for each orbit measurement in BPM 90. The purple points are assuming all gains are equal. For the green points the measured intensities are corrected using the fitted gains. Note that the green line includes (0,0) and the purple line does not. (right) Measured positions of all of the orbits in BPM 90.