Calculation of corrected BPM data with polynomials obtained from BpmLab: ESRF and ALBA showcases

A. Nosych, K.B. Scheidt
What is BpmLab?

a FEM-based Matlab tool for BPM response simulation and treatment

2D BPM slices are defined by combinations of simple shapes: circles, ellipses & polygons

Library of many real BPMs (34 and counting...)

Geometry, mesh & map settings

Geometry outlines

Final mesh & boundary nodes precision
Meshing:
Open-source tetrahedral mesh generator based on metric functions (*DistMesh*).

- Uniform or non-uniform meshes
- User-defined min/max mesh size
- Extremely reliable inner mesh
- Boundary mesh QA: by eye. Always check!

FEM Solver:
2D electrostatic Poisson solver with linear basis functions and boundary electric potential excitation (Diriclet boundary conditions):

**Green’s reciprocity theorem**: the surface charge induced on an electrode due to a test charge at \((x,y)\) is proportional to the potential at that same position when the test charge is absent and the electrode is excited by a potential \(V\).
**BpmLab:** Response maps & DOS treatments

ESRF standard SR BPM

\[ \delta/\Sigma \]

\[ 20\log(\delta/\Sigma) \]

\[ \text{diagonal} \]

\[ \text{atanh}(\delta/\Sigma) \text{ (J.Chavanne, 2002)} \]
ALBA BPM family

Booster buttons: DOS in 29x29 (657) map points

Storage ring BPM: DOS in 25x49 (776) map points
Upcoming EBS BPM family

- Large EBS-BPM: DOS in 25x51 (591) map points
  - Large (8 mm button)

- Small EBS-BPM: DOS in 21x51 (333) map points
  - Small (6 mm button)

- EBS Chamber 12: DOS in 29x51 (583) map points
  - Chamber 12 (4b)

- EBS Injection BPM: DOS in 21x31 (541) map points
  - Injection: stored beam

- EBS Injection BPM: DOS in 21x61 (717) map points
  - Injection: bumped beam

- EBS Chamber 12: DOS in 29x51 (583) map points
  - Chamber 12 (2b)
How to correct?
• Linear coefficient (real-time)
• Polynomial families (real-time)
• Specific formulae (real-time)
• Lookup tables (post-process)
• Voltage inversion (iterative optimization, so post-process)

At what level? (by DS/Libera? by end-user?)

How fast? (TbT 100x kHz, FA 10x kHz, SA 10x Hz)
Polynomial correction by DOS

Inverted voltages
Polynomial correction seems to be the straight-forward solution if you want to have large beam drifts, and have fast real-time position nonlinearity suppression.

2 polynomials need to be applied to each data sample:

\[ A, B, C, D \rightarrow \frac{\delta x}{\Sigma}, \frac{\delta y}{\Sigma} \rightarrow P_x(\frac{\delta x}{\Sigma}, \frac{\delta y}{\Sigma}), P_y(\frac{\delta x}{\Sigma}, \frac{\delta y}{\Sigma}) \]

Polynomials are coupled 2D type.

Higher power = more coefficients, e.g.:
Max pwr 7: 36 terms
Max pwr 13: 105 terms etc.
**BpmLab: Coupling**

**Coupling (DOS positions)**

![Graph showing Coupling (DOS positions)]

**Coupling improved (Poly correction)**

![Graph showing Coupling improved (Poly correction)]

**Coupling:**
Ratio of how much the position change in one dimension affect its change in the other dimension.

Using the polynomial corrected positions can minimize the coupling effect.
**BpmLab: Sum, coherence**

**Sum signal**

\[ \Sigma = A + B + C + D \]

**Signal Coherence**

\[ Q = \frac{(A + C - B - D)}{\Sigma} \]

**Sum & Coherence:**

Strongly nonlinear.

Can be linearized with higher-order polynomials.
A polynomial is defined by its effective ROI and max power. How can we pick them?

**Step 1)** Fix max polynomial power (here: $x^7y^7$) and scan the ROI dimensions.

Accuracy starts to deteriorates as the poly needs to fit a more curvy surface:
**BpmLab: Polynomial power fitness**

Step 2) Fix the ROI (here ±20 mm x ± 10 mm) and scan the poly power.

Step 3) Choose the power/ROI combination that fits your needs

Beware: speed of calculation can increase for higher power polynomials.
Finally we have to stop somewhere, e.g.:

**ALBA SR BPM**
ROI = ±[22, 10] mm  
max pwr = 11

**ESRF SR BPM**
ROI = ±[25, 14] mm  
max pwr = 15

**EBS BPM**
Several Poly families selectable in Libera DS.  
max pwr = ??
**BpmLab: Post-processing real signals**

**ALBA SR**
Local bump, 4 orbit-correctors

- **Hor:** 0 → 3 mm
- **Ver:** 0 → 4 mm

Graph showing data points and graphs for DOS and POLY map.

- **Generate map:**
  - Uniform grid: -0.2, 0.2
  - Scattered points: 100

- **Channel model:**
  - Add noise [%]:
  - Ampl/att coef:

- **Import signals:** ALBA-krlS-bpm-06.mat
**BpmLab**: Button imperfections (retraction/protrusion)

- 300 um retracted
- Poly correction: channel A as is
- Poly correction: channel A compensated

Use “perfect Poly” to correct this distortion.
BpmLab: Signal manipulation

Add noise (1%) to all channels: DOS treatment

Poly correction
**BpmLab summary**

Requires Matlab toolbox-free installation (versions range 2014b-2017)

**Instruments:**
- meshing (DistMesh)
- short 50-line 2D electrostatic FEM solver
- optimization (minFunc)

**Features:**
- Easy arbitrary geometry input & many presets
- BPM response map calculation in 2D
- Correction polynomial calculation (within ROI)
- Direct voltage-to-position inversion (post-processing)
- Introduce geometry imperfections
- Signal manipulation (channel distortion, attenuation, amplification)
- Import real BPM signals for studies and post-processing

Now the interesting stuff...

Experiments!
Injecting with RF off: Signal SUM & COHERENCE

ESRF (Nov 26, 2018)

Corrected Sum = \( \frac{\text{SUM}_{\text{RAW}}}{(\text{norm\_coef} \times \text{SUM}_{\text{model}})} \rightarrow 1 \)

Q Corrected = \( Q_{\text{DOS}} - Q_{\text{model}} \rightarrow 0 \)

Compensation in the first turns. Then becomes very noisy, why? Sum and Q are sensitive to beam size. It deteriorates without RF (blows up, bunch charge decrease, path & energy change).
Injecting with RF off: Polynomials vs. DOS

RF off → beam spirals inwards.
Linear formula underestimates beam offset.
Polynomial can see farther (depends on ROI).

So finally, polynomials or DOS?

Polynomial power!!!
Injecting with RF off: Scraper test

ALBA SR injection straight, PSA May 2019

Beam trajectory evolution vs. closing scraper.
Injections with RF off, all usable 145 turns.
Beam position corrected with polynomials.

left jaw moving to +30 mm

Each jaw kills beam at ±7 mm
DOS error ≤ 2.5 mm
Poly error ≤ 0.16 mm

First BPM

Septum blade

~8 meters
Beam kicks: vertical plane kick

ESRF MDT Nov27: big V-kick, partial loss

MAF off
Anti-smearing on

Note the behavior of the Sum during the V-kick (bunch length doesn’t change?)

Very nice Q suppression!
Beam kicks: horizontal plane kick

ESRF MDT Nov27: moderate H-kick, small loss

View of all BPMs

Again a very nice Q suppression
Beam kicks: horizontal plane kicks

ESRF MDT Nov27: H-kicks, loss variation

3 mA $\rightarrow$ $\sim$3 mA

3 mA $\rightarrow$ 1.5 mA

3 mA $\rightarrow$ 0.5 mA

Sum jumps 5-20%!

Sum oscillates only during H-kicks! Why?
Gets under/over compensated by SumPoly correction.
Thank you for your attention!
extras
Overlaying the measured Sum with the horizontal position -> no expected correlation:

1) **black dots** = simulated sum, so the sum does not get higher than 1 from beam moving left and right from the center.
2) beam goes right -> sum goes up, beam goes left -> sum goes down. **How is this possible. The sum is supposed to behave symmetrically, going down in both cases after crossing zero?**
Experimental application: ESRF (Nov 2018) kicks

MDT Nov27: H-kicks, loss variation