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

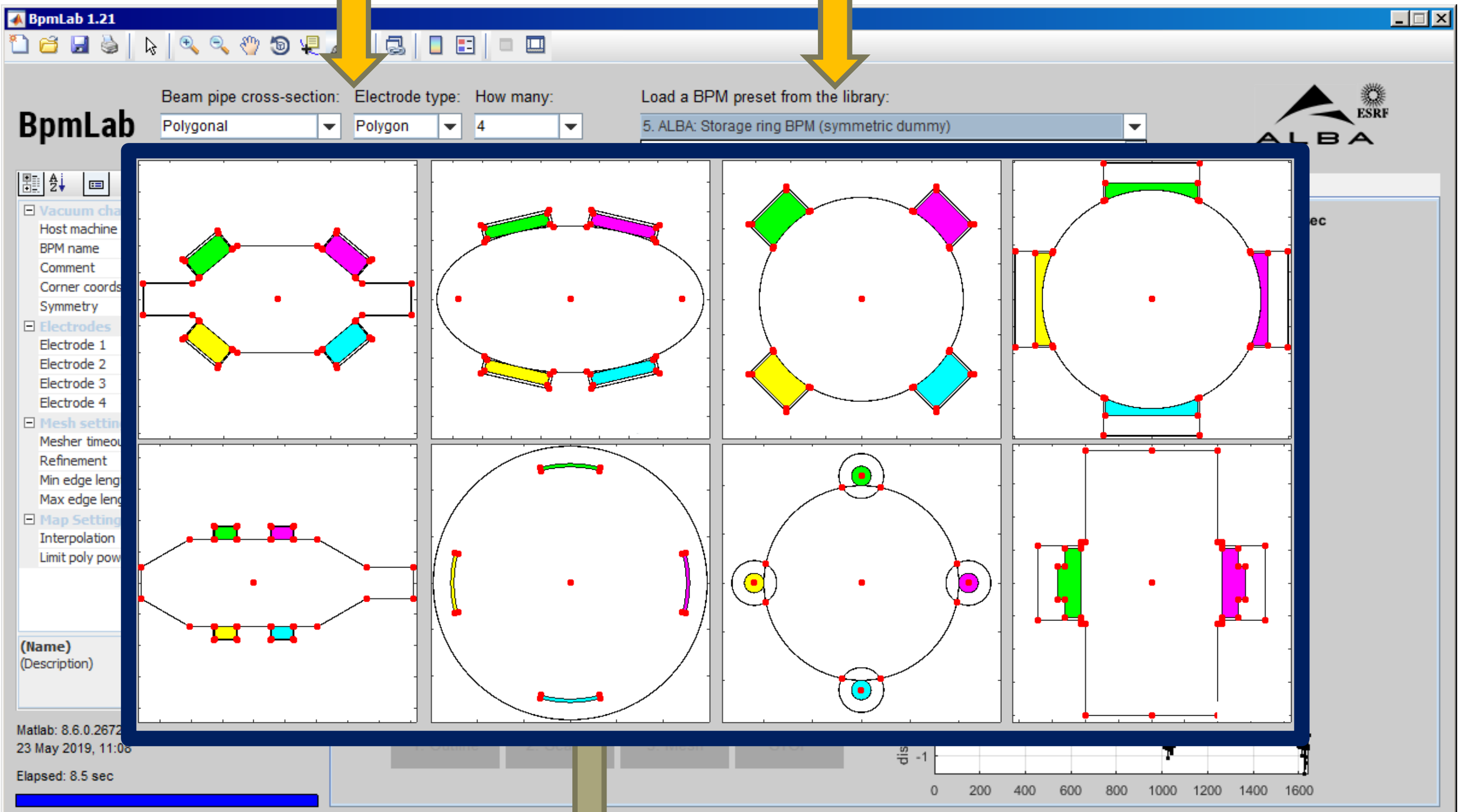
A. Nosych, K.B. Scheidt

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…)



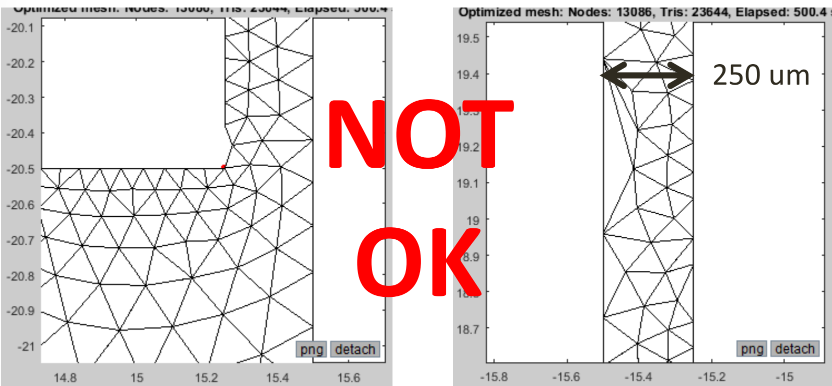
Geometry, mesh & map settings

Geometry outlines

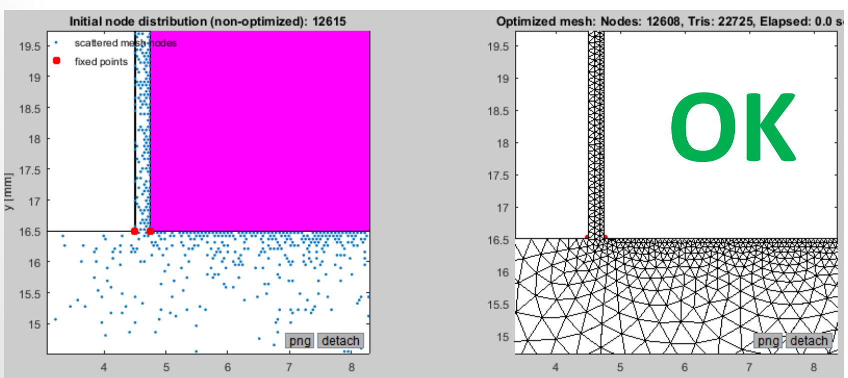
Final mesh & boundary nodes precision

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!

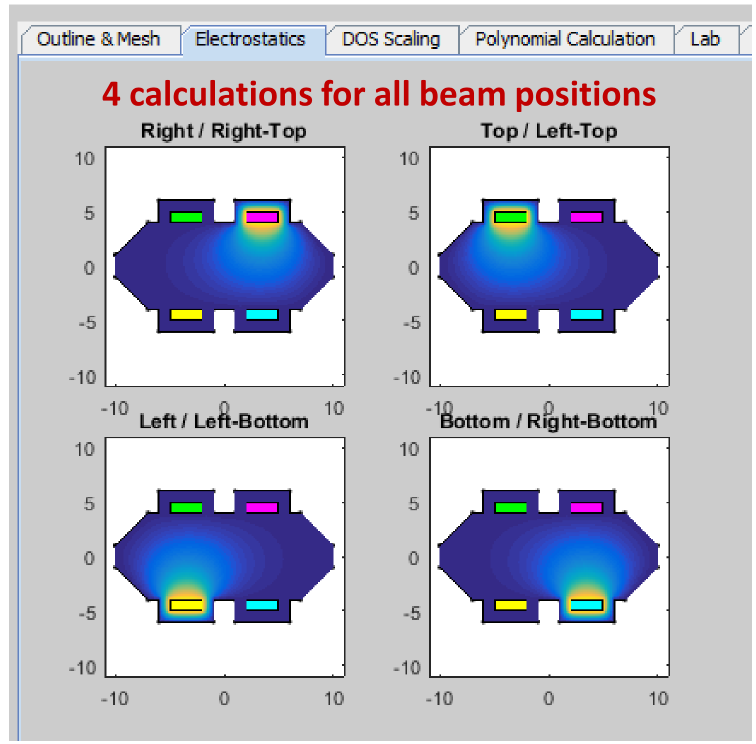


Examples of a badly converged mesh



Properly converged mesh

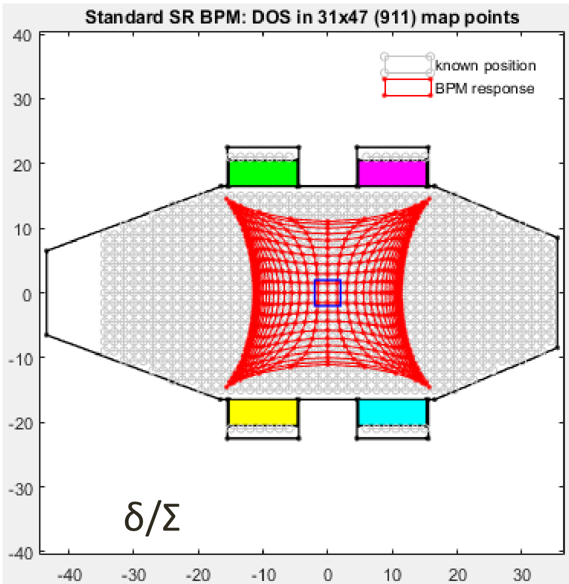
FEM Solver:
2D electrostatic Poisson solver with linear basis functions and boundary electric potential excitation (Diriclet boundary conditions):

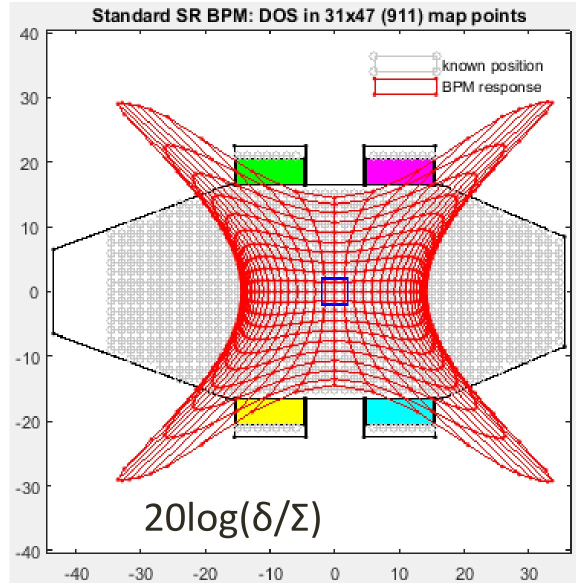


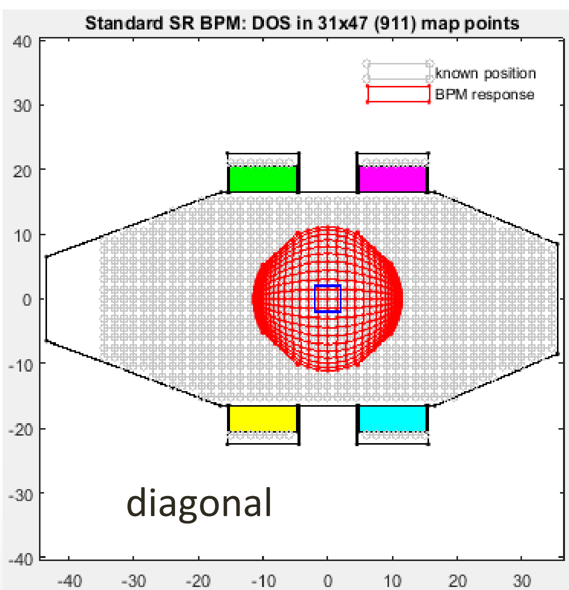
Green's reciprocity theorem: the surface charge induced on an electrode due to a test charge at (x,y) is proportional to the potential at that same position when the test charge is absent and the electrode is excited by a potential V.

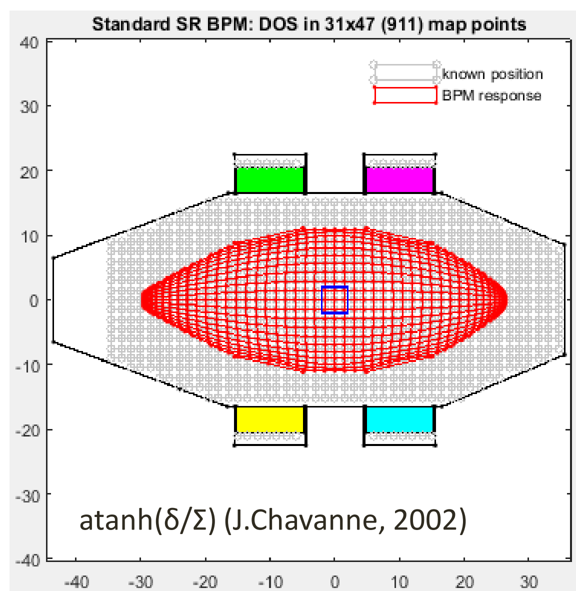
BpmLab: Response maps & DOS treatments

ESRF standard SR BPM

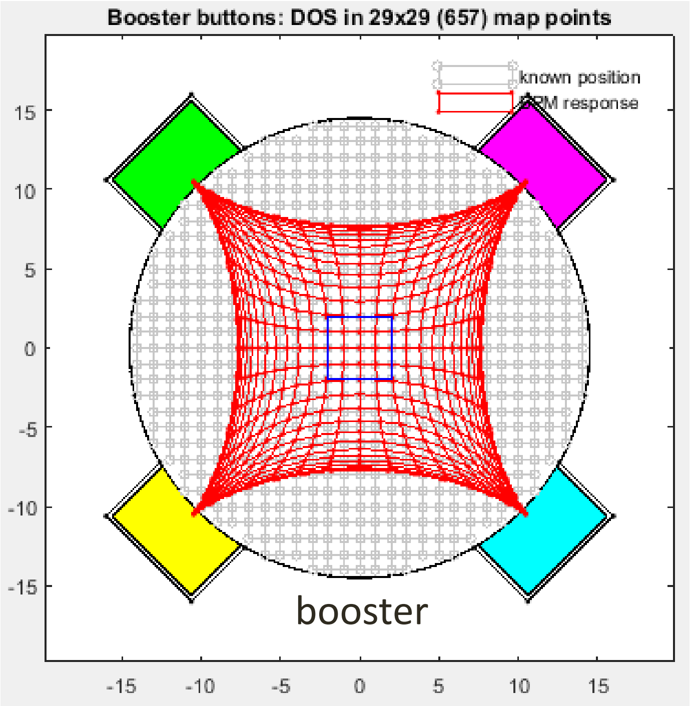


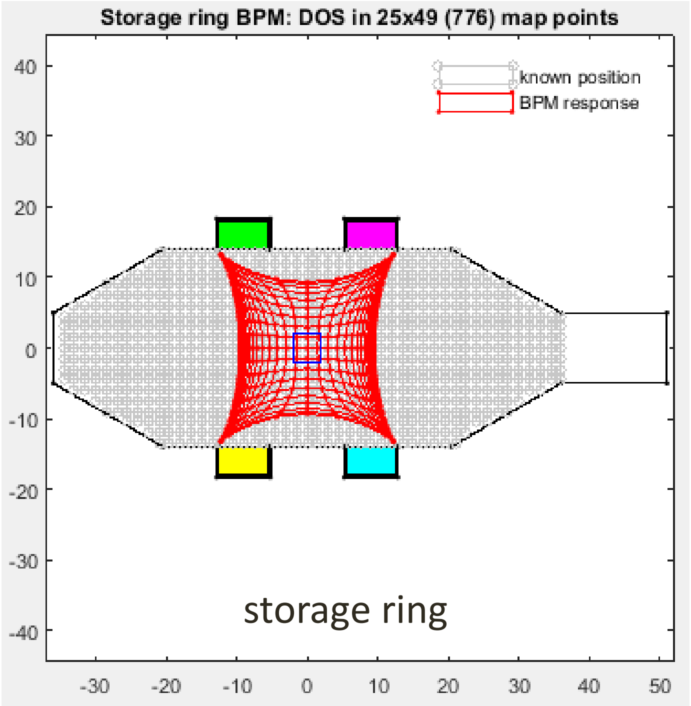




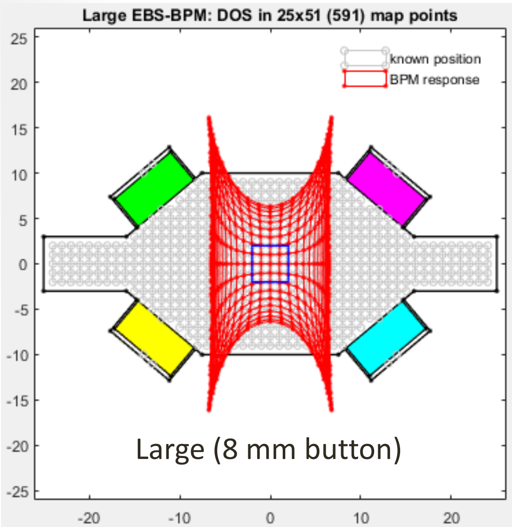


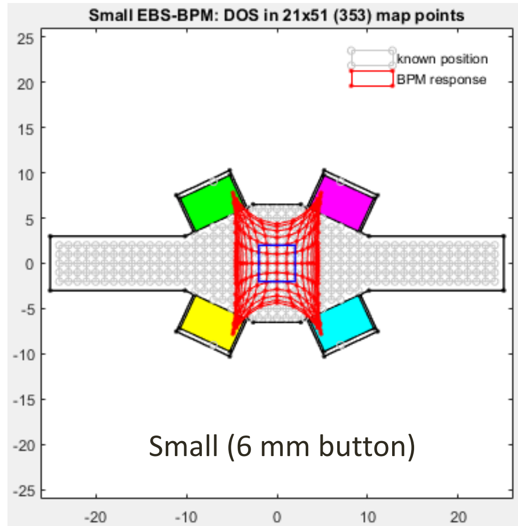
ALBA BPM family

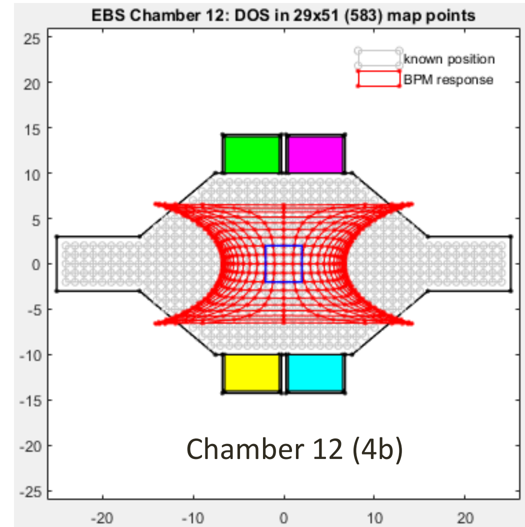


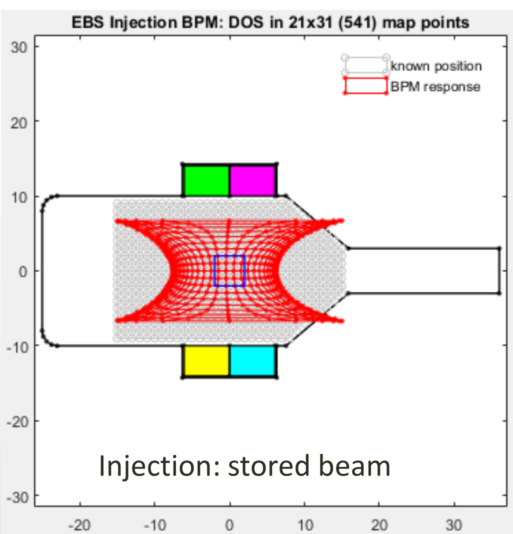


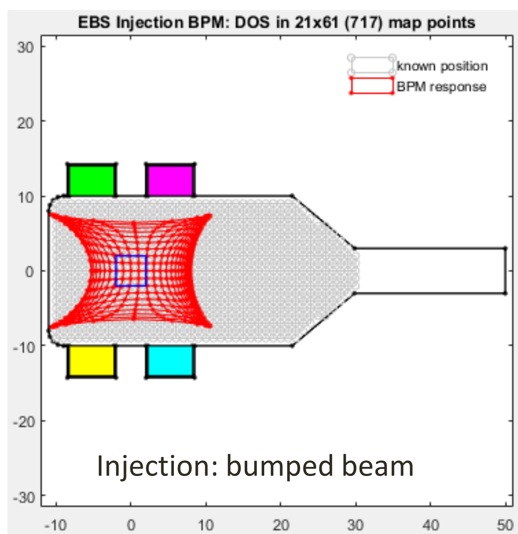
Upcoming EBS BPM family

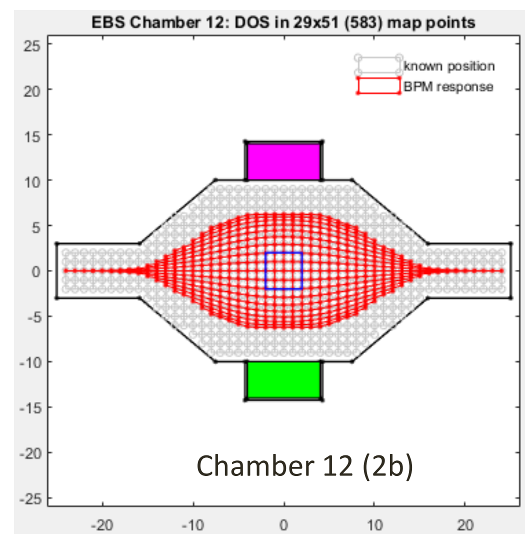


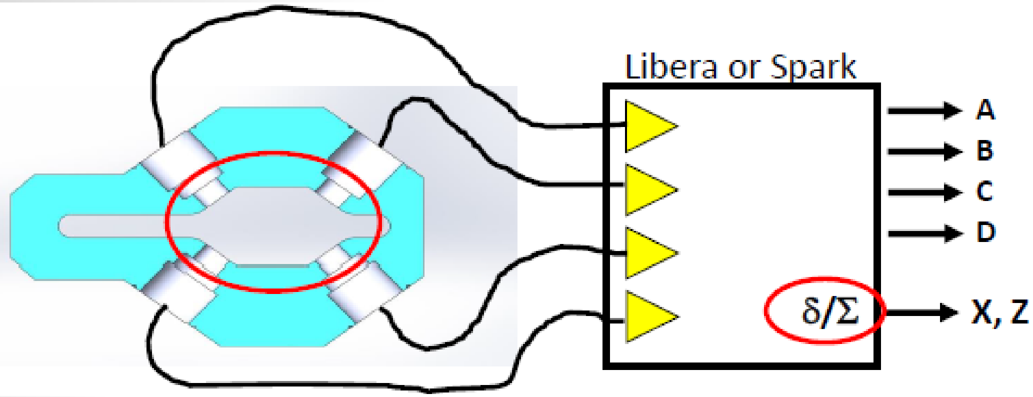






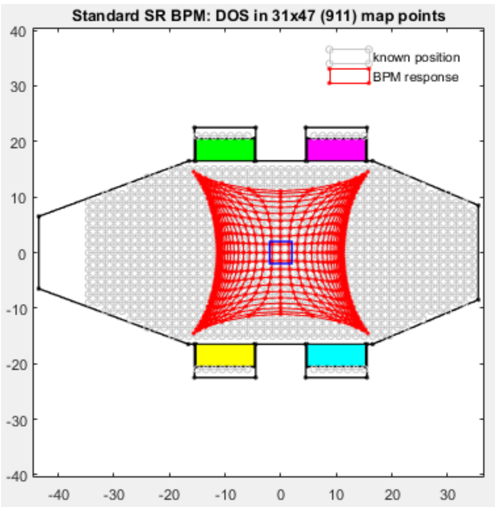


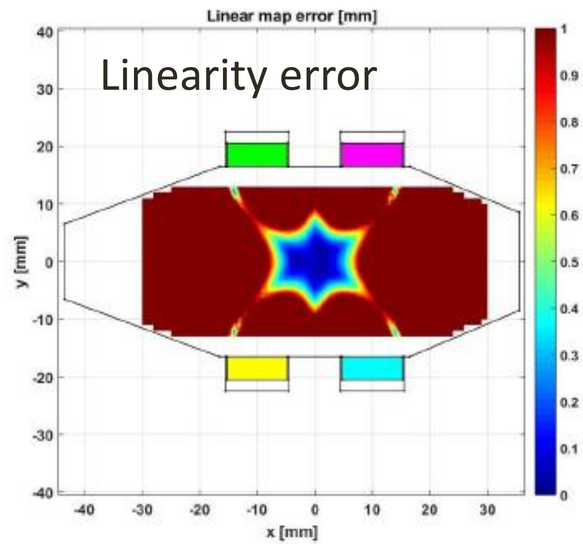




Perfect BPM & low noise stable electronics are still imperfect:

dislinearity !
coupling !





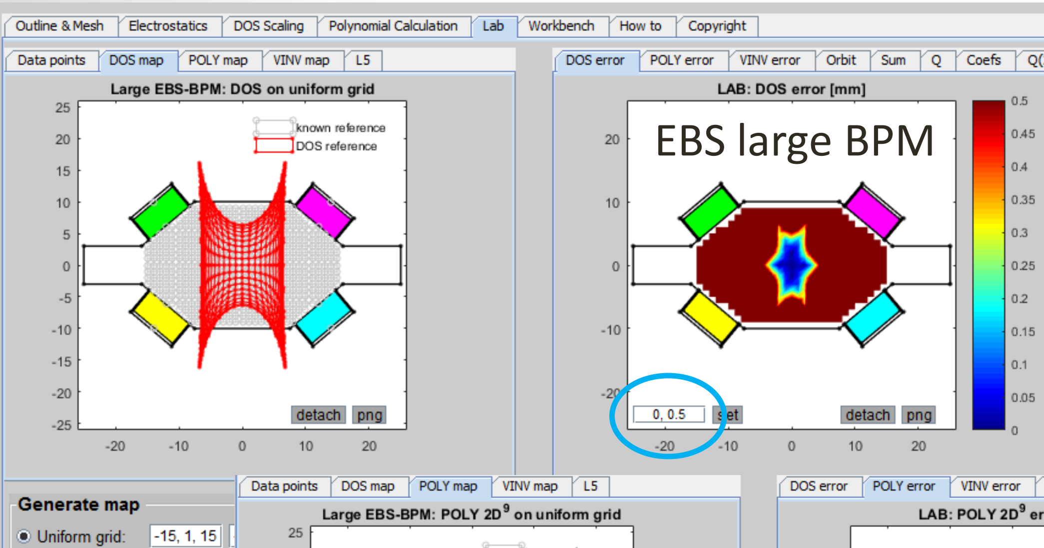
How to correct?

- Linear coefficient (real-time)
- Polynomial families (real-time)
- Specific formulae (real-time)
- Lookup tables (post-process)
- Voltage inversion (iterative optimization, so post-process)

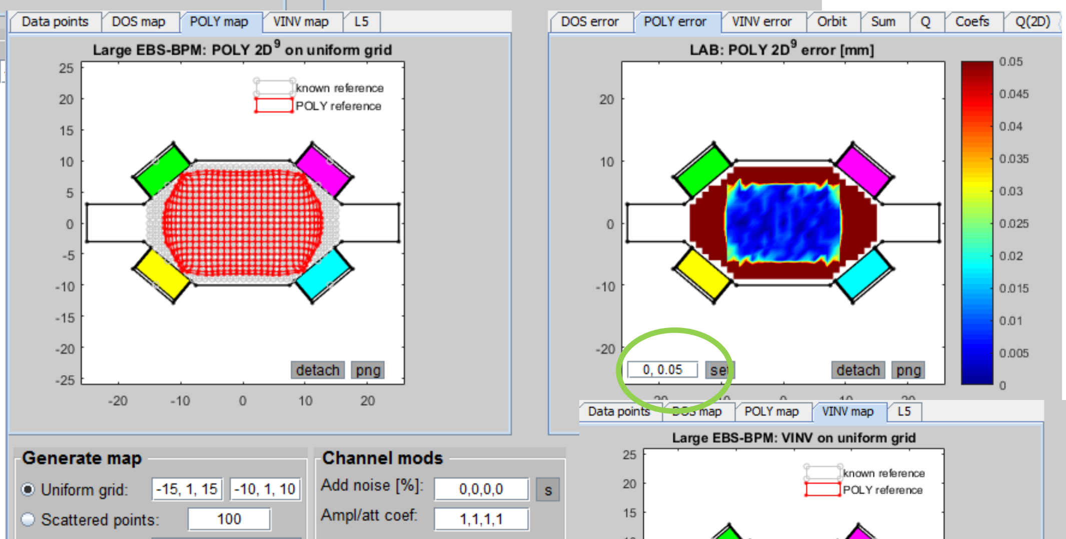
At what level? (by DS/Libera? by end-user?)

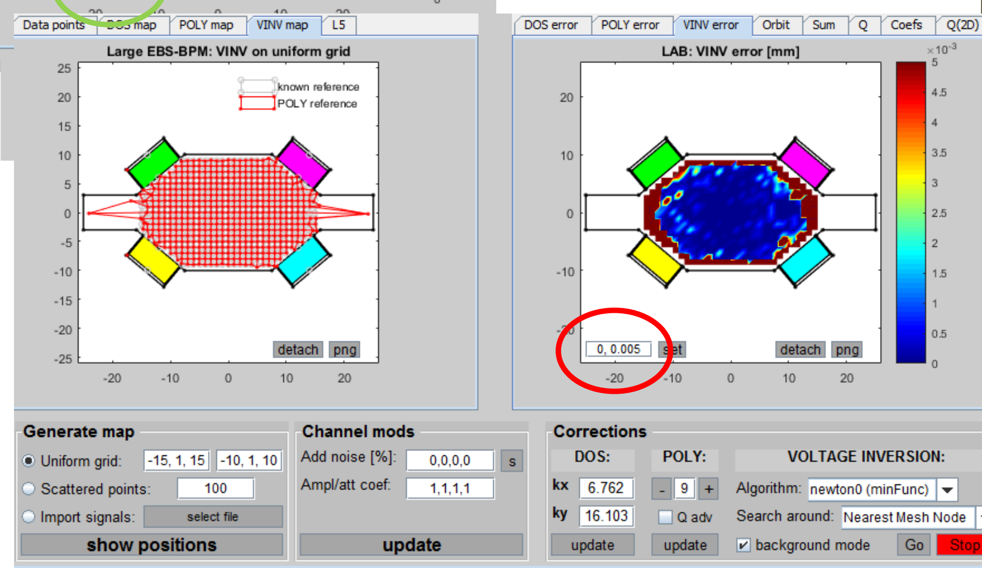
How fast? (TbT 100x kHz, FA 10x kHz, SA 10x Hz)

Linear correction by DOS



Polynomial correction



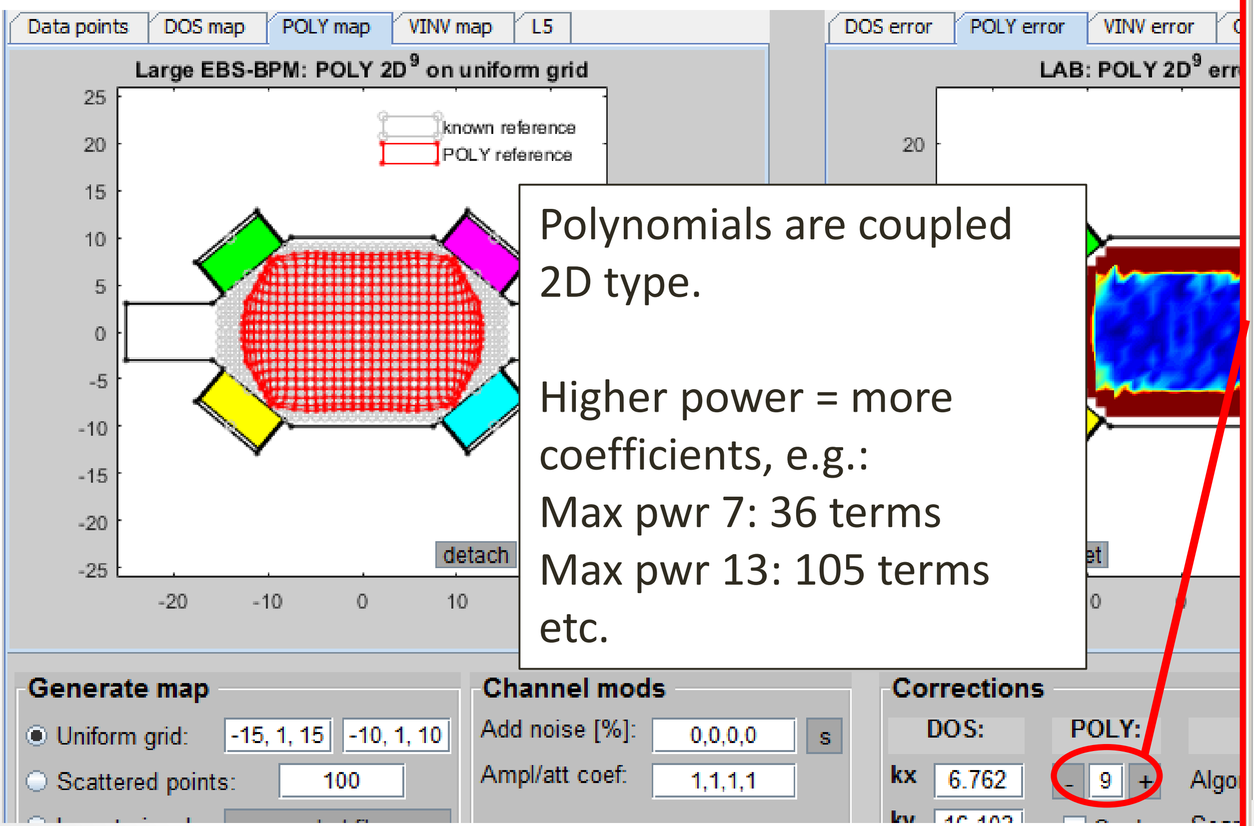


Inverted voltages

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.

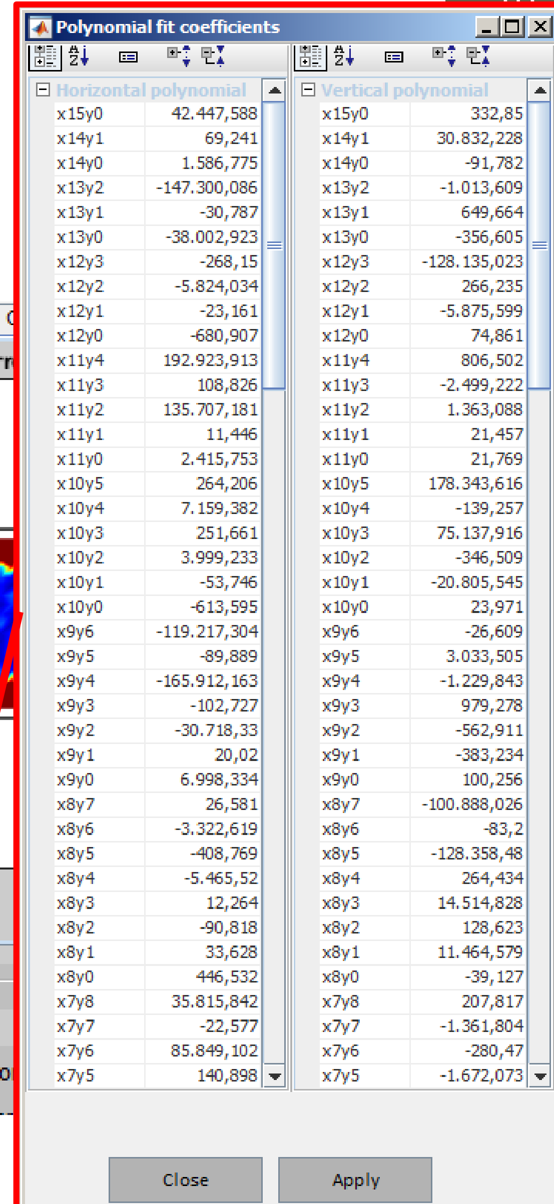
2 polynomials need to be applied to each data sample:

A, B, C, D → $\delta x/\Sigma$, $\delta y/\Sigma$ → $Px(\delta x/\Sigma, \delta y/\Sigma)$, $Py(\delta x/\Sigma, \delta y/\Sigma)$



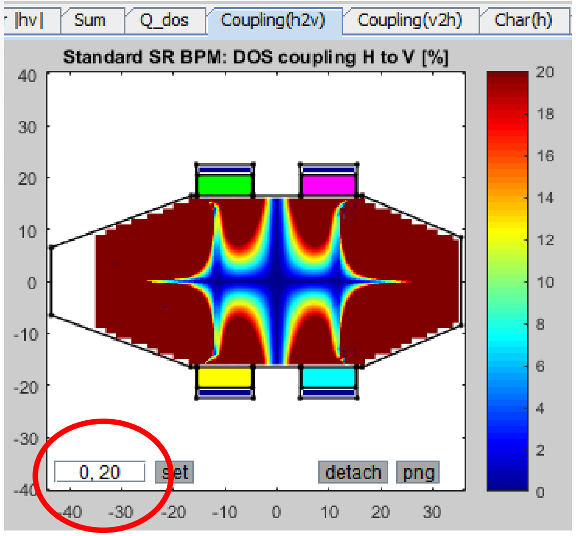
Polynomials are coupled 2D type.

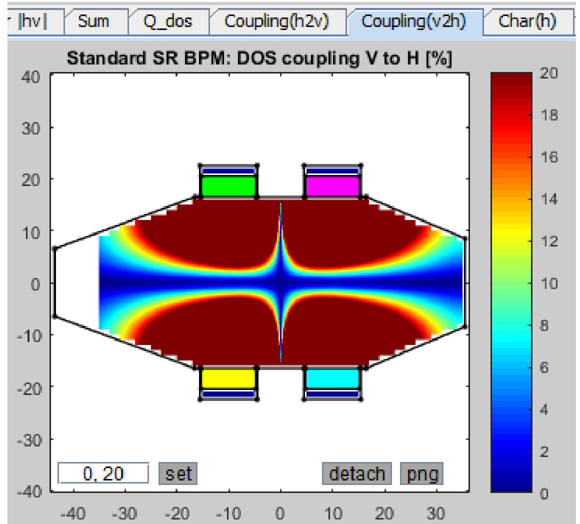
Higher power = more coefficients, e.g.:
Max pwr 7: 36 terms
Max pwr 13: 105 terms
etc.



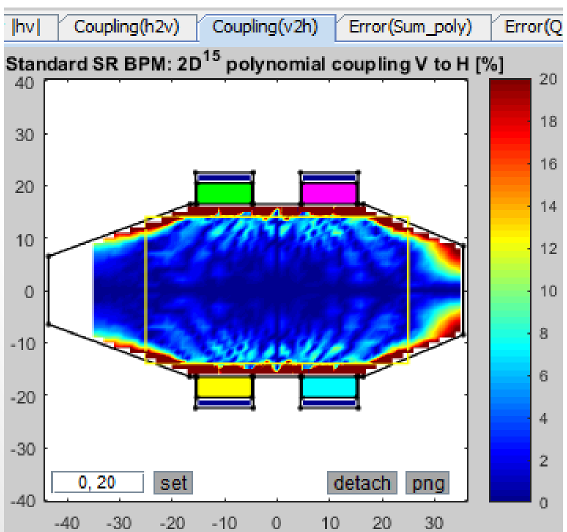
BpmLab: Coupling

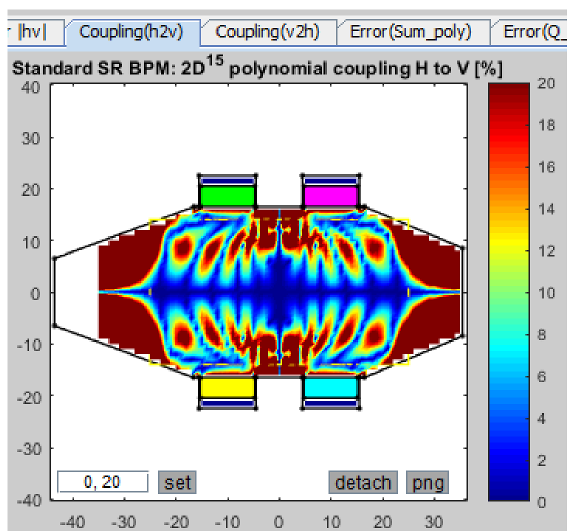
Coupling (DOS positions)





Coupling improved (Poly correction)



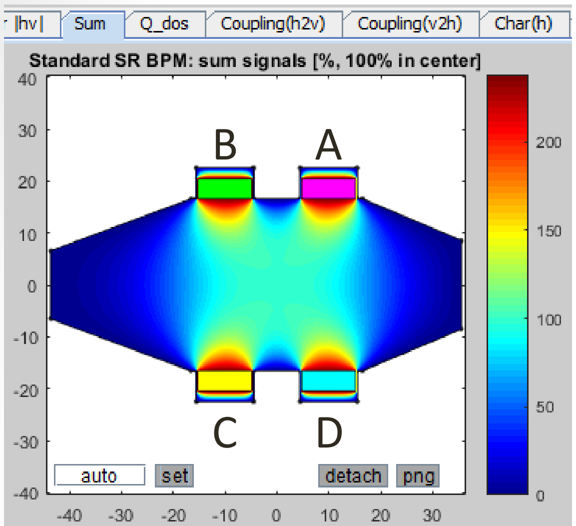


Coupling:
Ratio of how much the position change in one dimension affect its change in the other dimension.

Using the polynomial corrected positions can minimize the coupling effect.

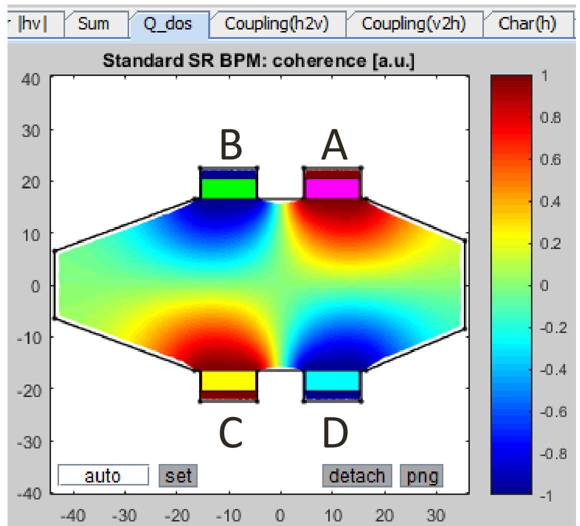
BpmLab: Sum, coherence

Sum signal

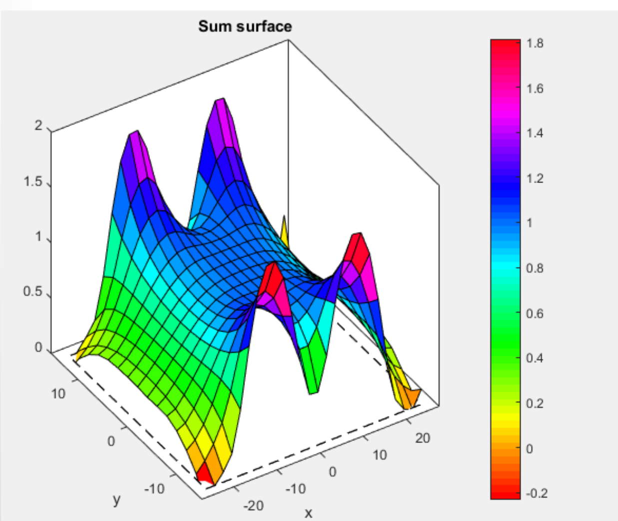


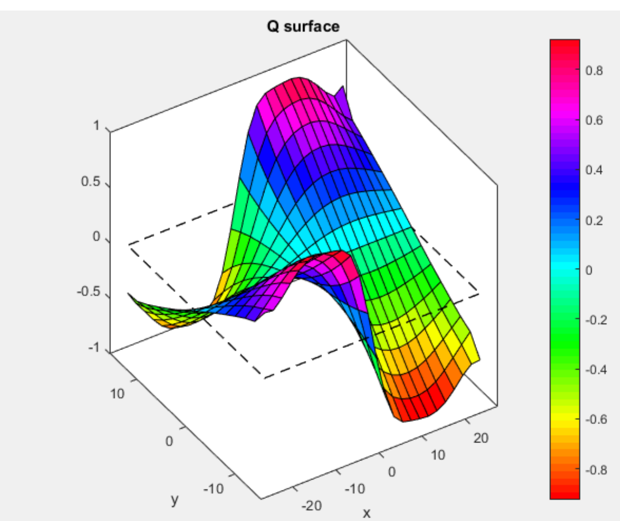
Σ = A+B+C+D

Signal Coherence



Q = (A+C-B-D) / Σ





Sum & Coherence:
Strongly nonlinear.

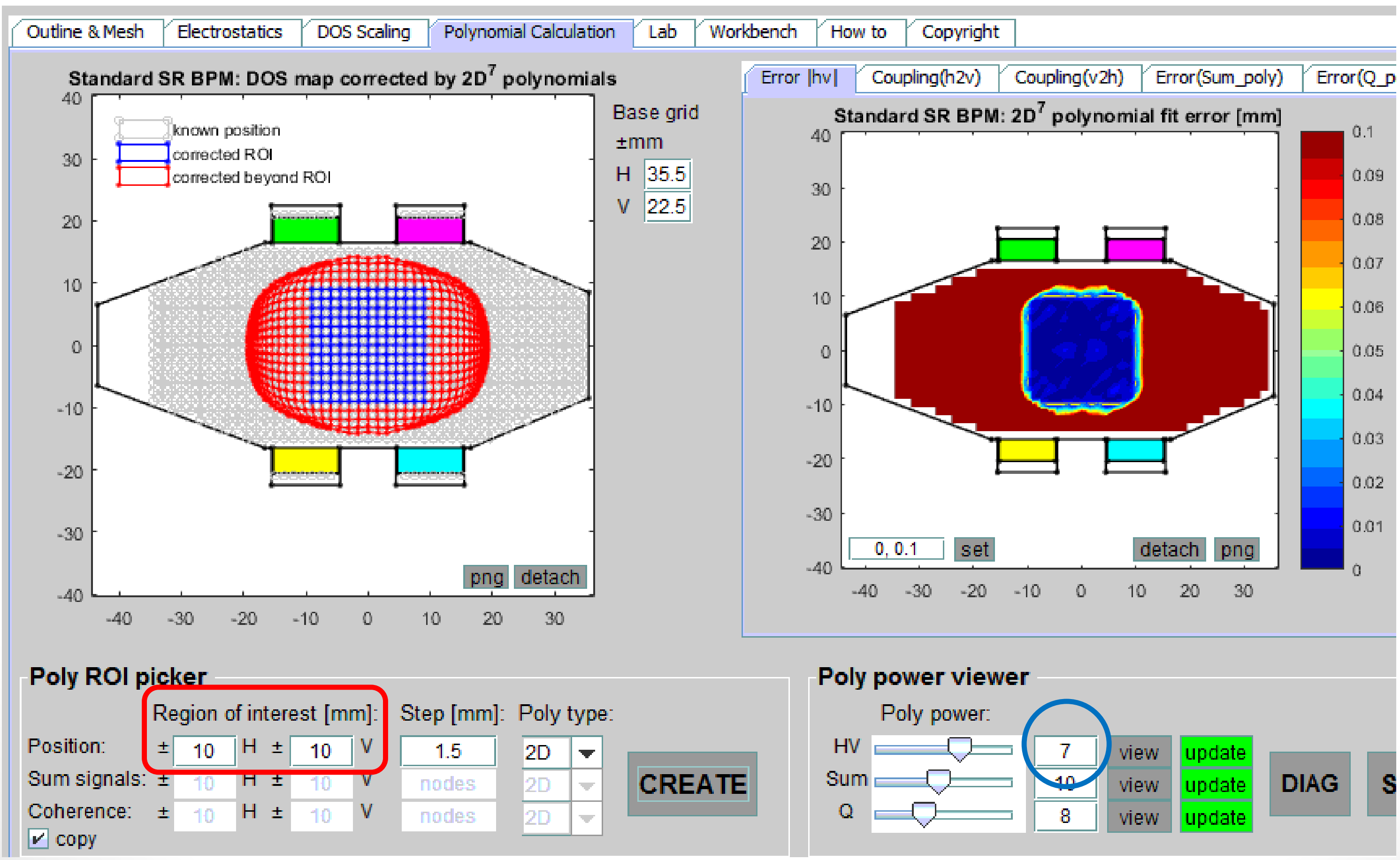
Can be linearized with higher-order polynomials.

BpmLab: Polynomial ROI fitness

A polynomial is defined by its effective ROI and max power. Ho 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:

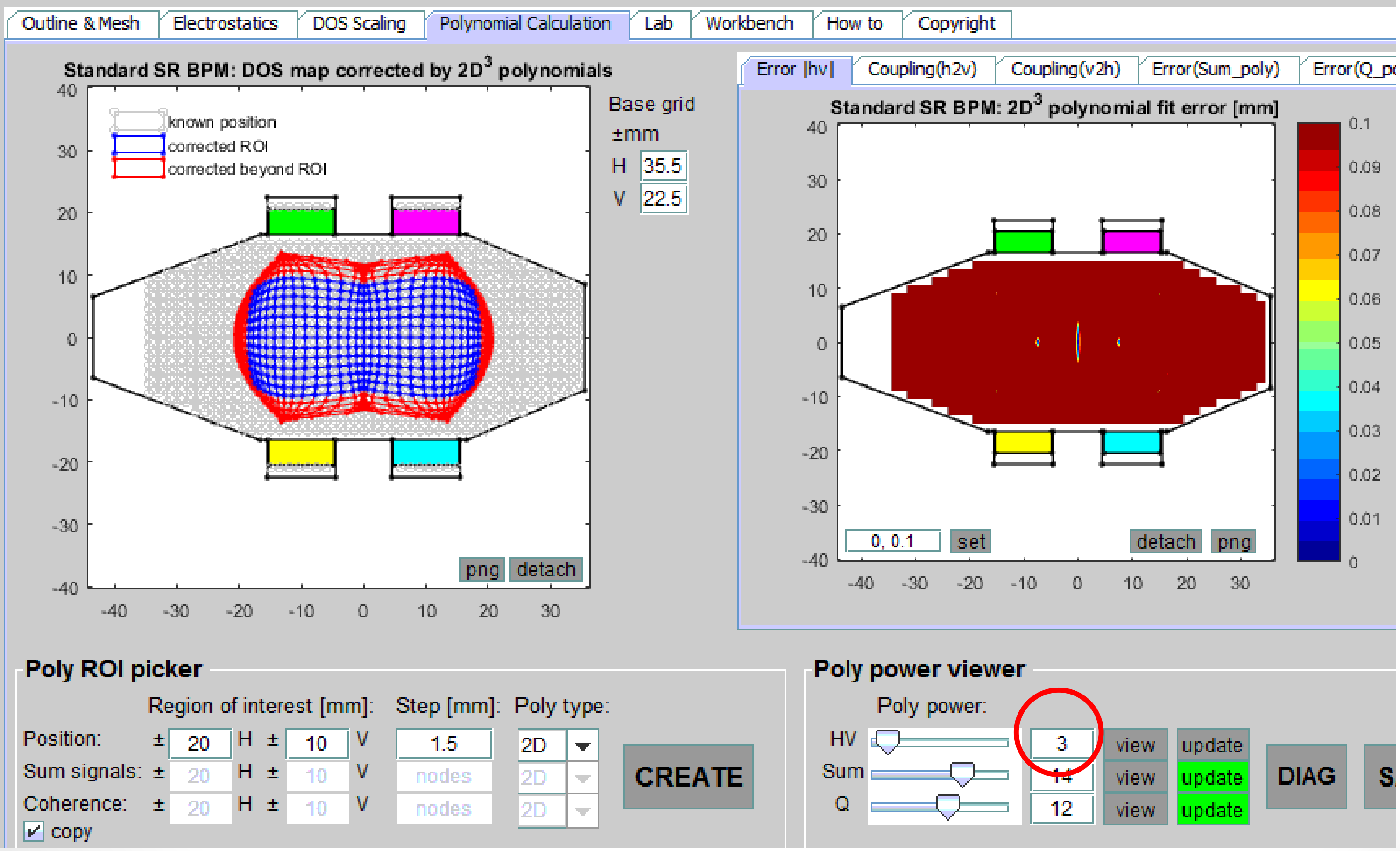


BpmLab: Polynomial power fitness

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

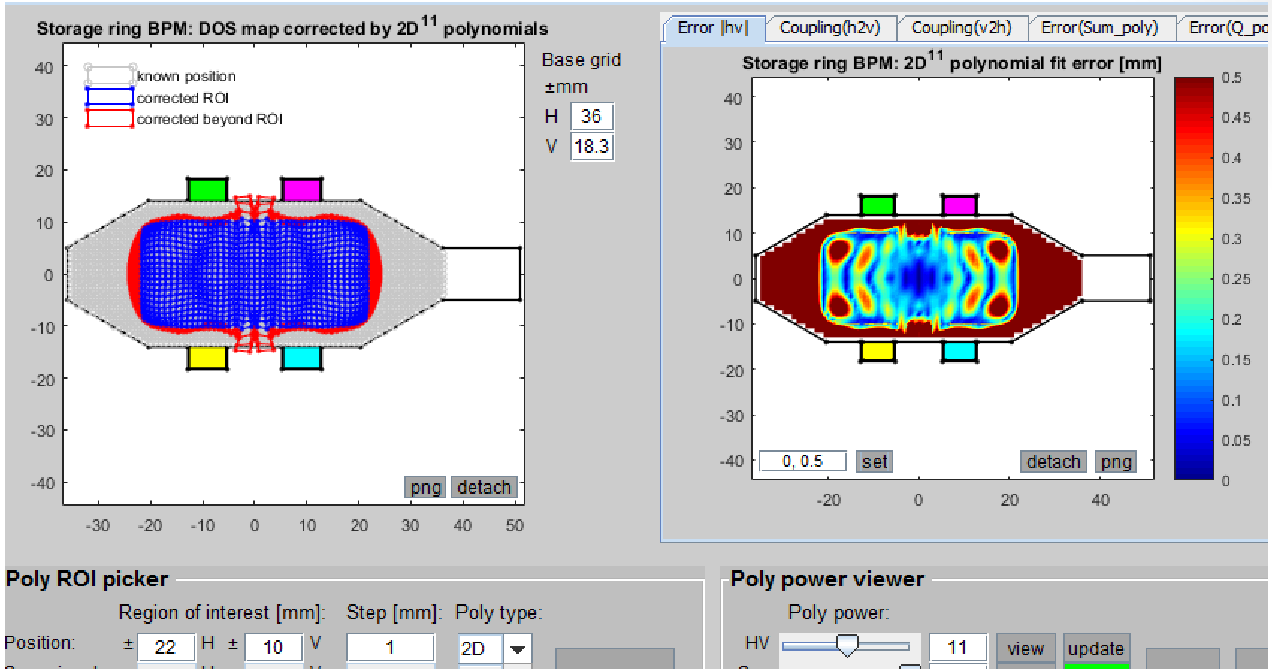
Beware: speed of calculation can increase for higher power polynomials.



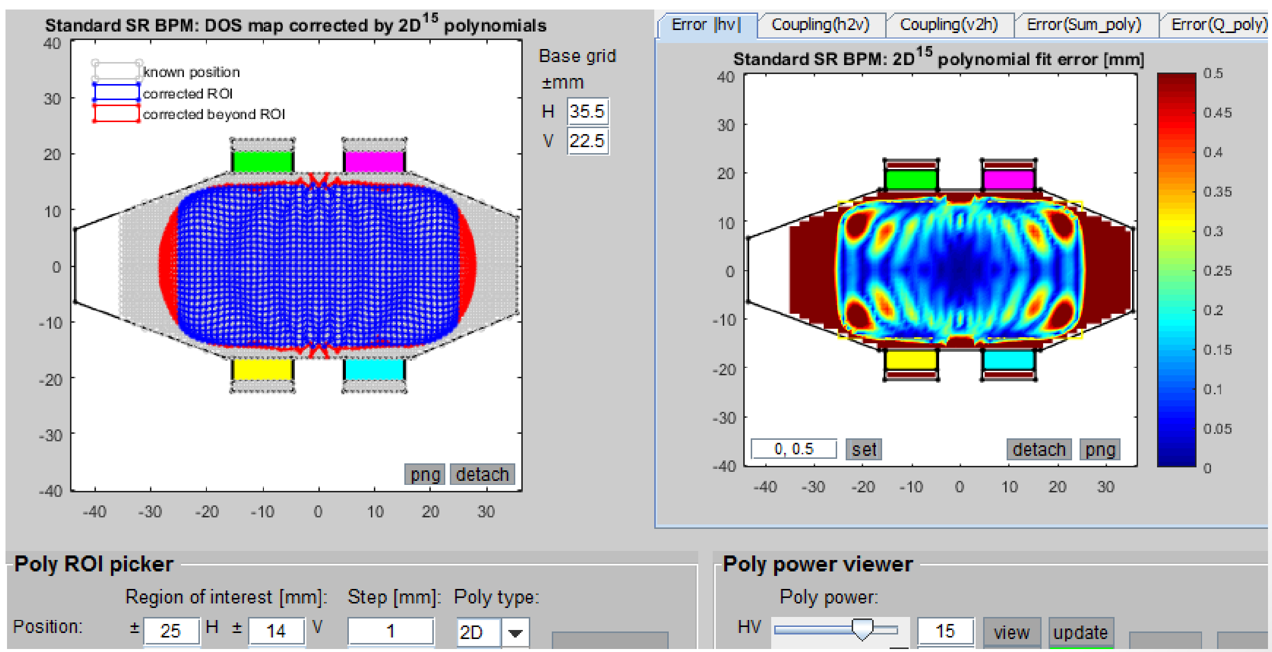
BpmLab: Polynomial fitness

Finally we have to stop somewhere, e.g.:

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



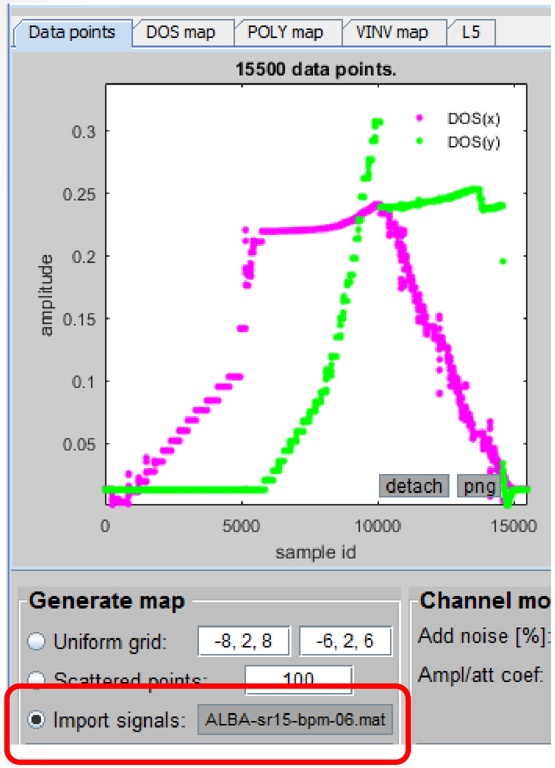
ESRF SR BPM
ROI = ±[25, 14] mm
max pwr = 15

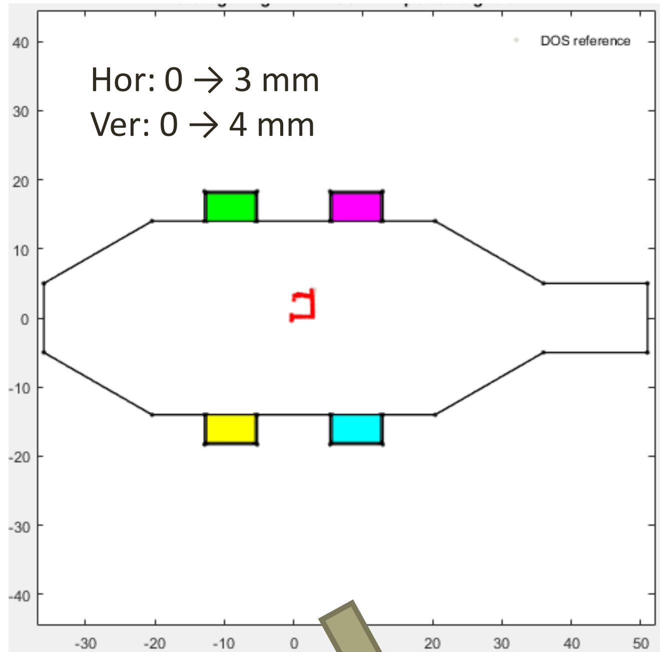


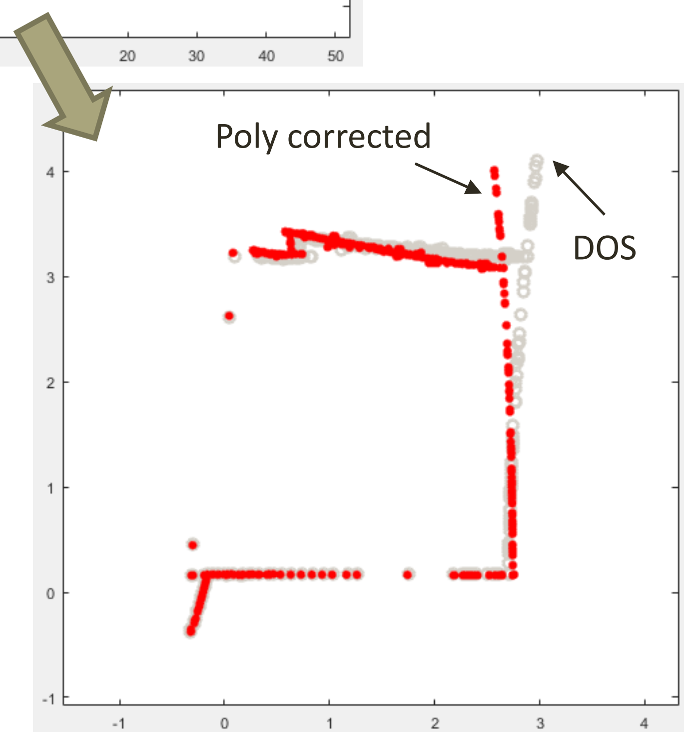
EBS BPM
Several Poly families selectable in Libera DS.
max pwr = ??

BpmLab: Post-processing real signals

ALBA SR
Local bump, 4 orbit-correctors

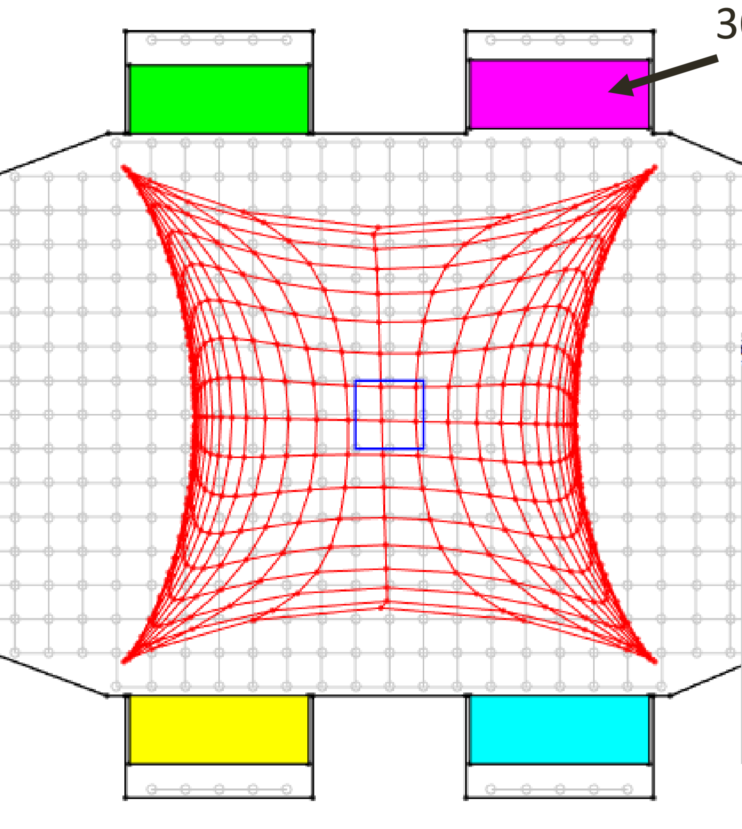






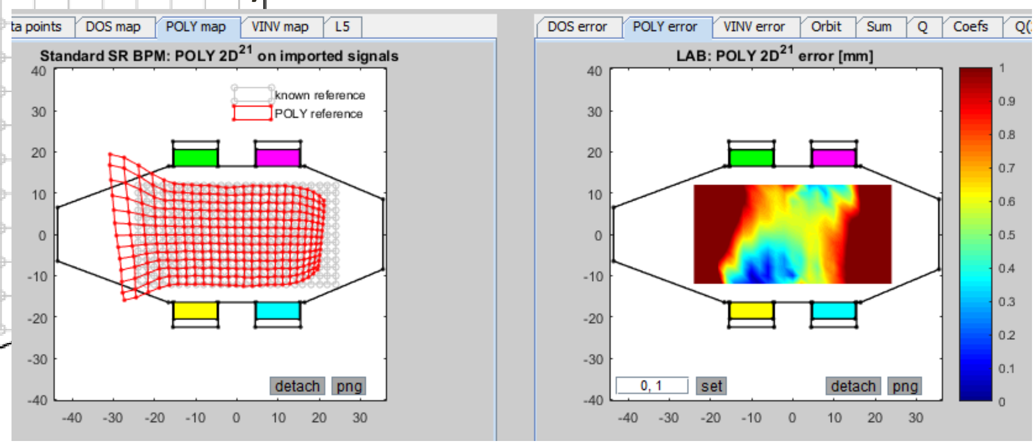
BpmLab: Button imperfections (retraction/protrusion)

300 um retracted

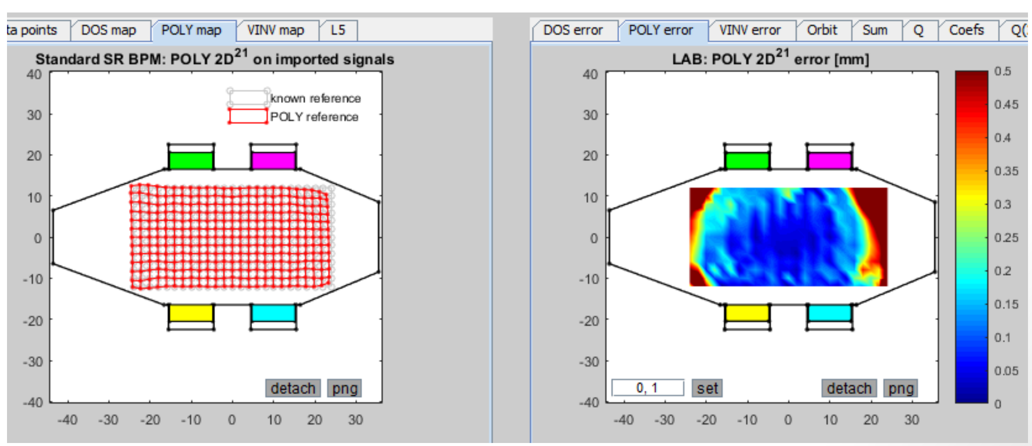


Use “perfect Poly” to correct this distortion

Poly correction: channel A as is



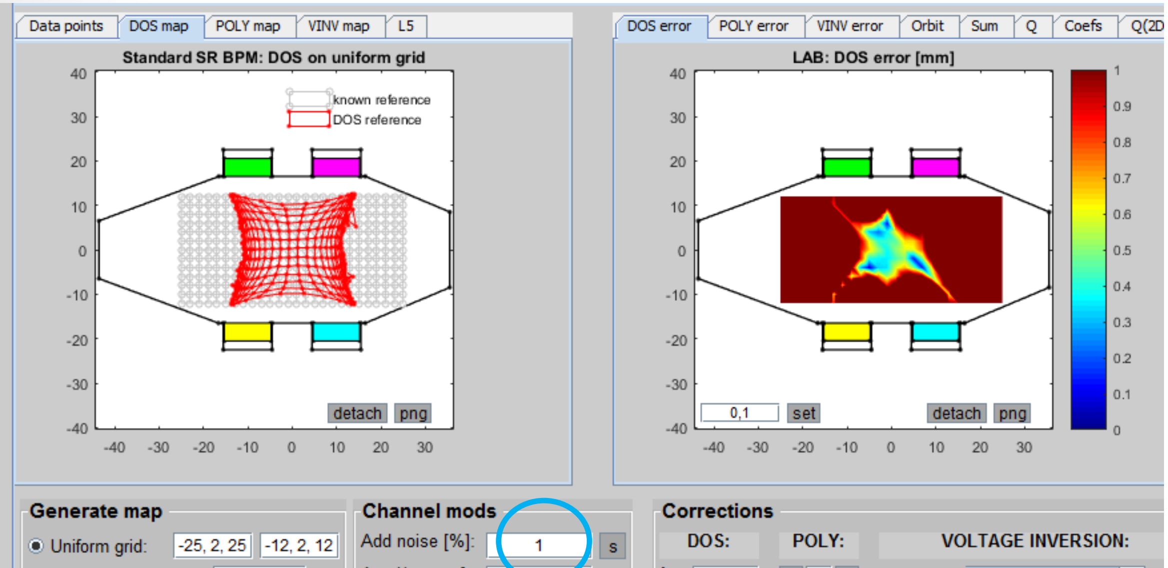
Poly correction: channel A compensated



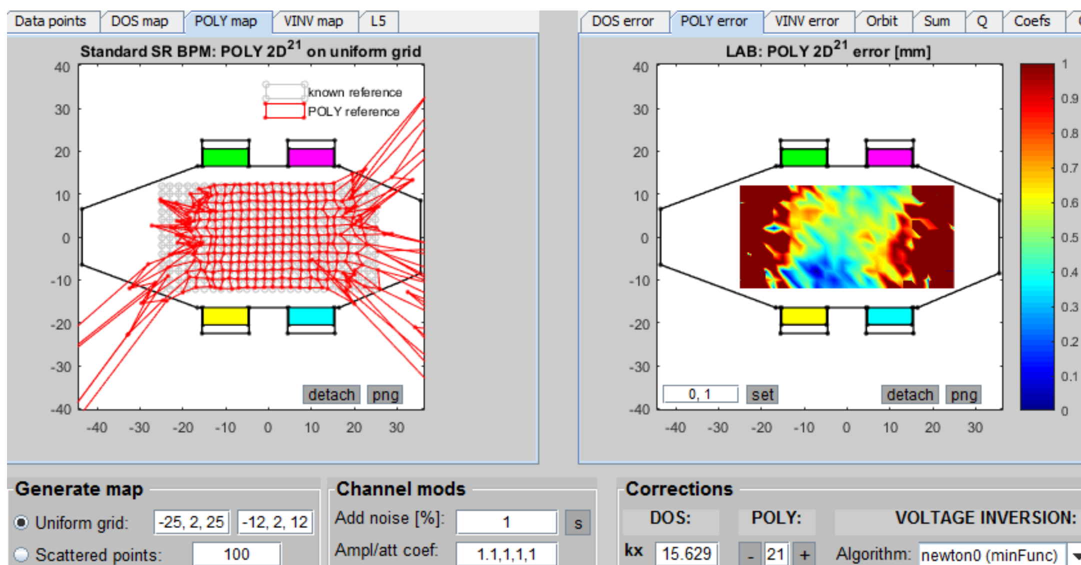
BpmLab: Signal manipulation

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

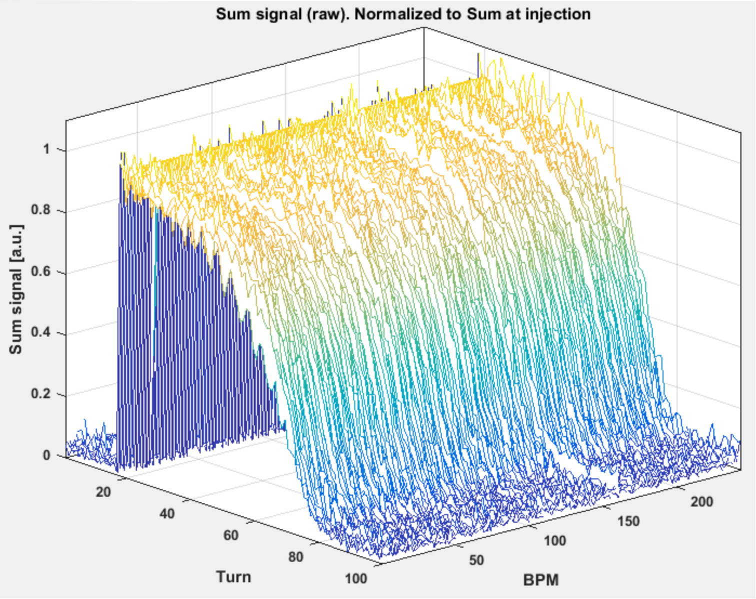
Download from here:
https://intranet.cells.es/Divisions/Accelerators/RF_Diagnostics/Diagnostics/OrbitPosition/Tools/bpmlab

Now the interesting stuff…

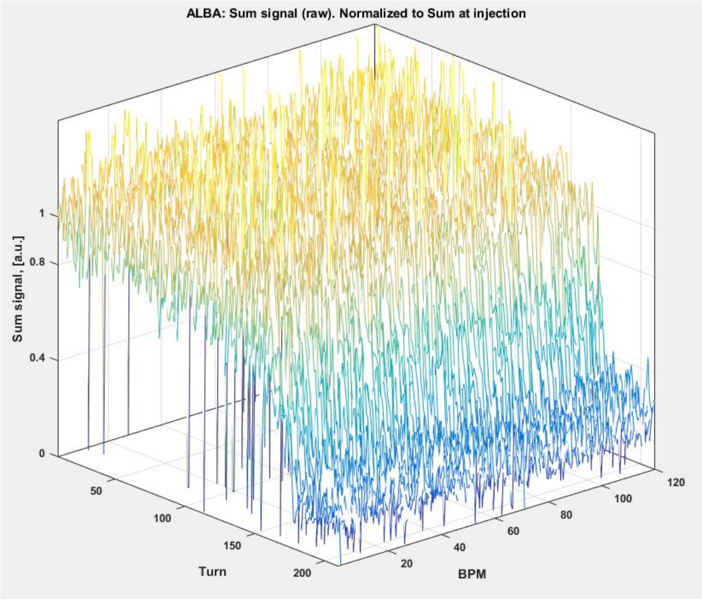
Experiments!

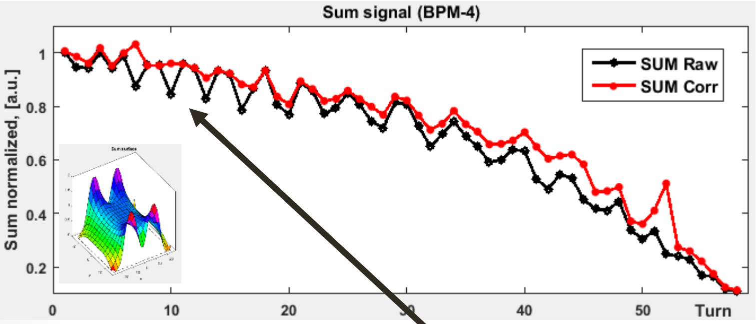
Injecting with RF off: Signal SUM & COHERENCE

ESRF (Nov 26, 2018)



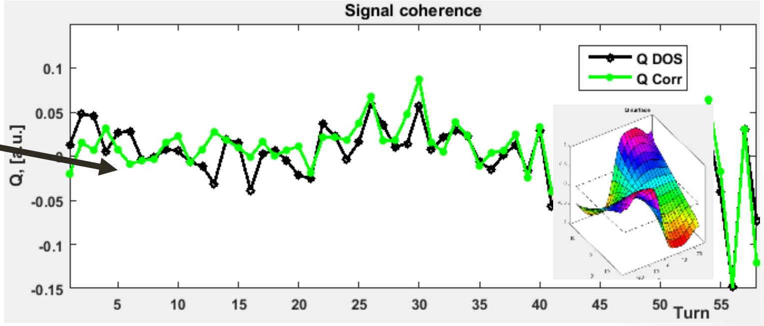
ALBA (May 2019)





Corrected Sum =
SUM_{RAW} / (norm_coef * SUM_{model}) → 1

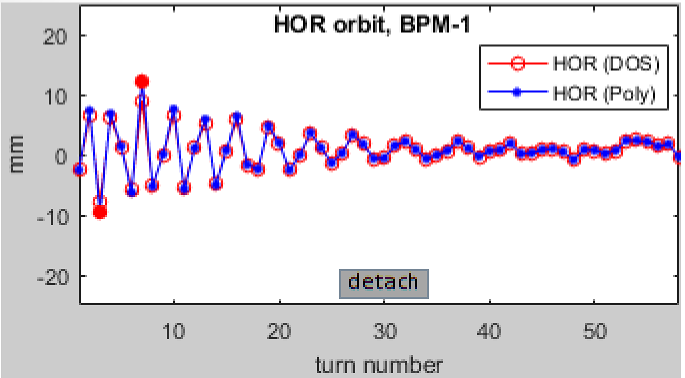
Q Corrected = $Q_{DOS} - Q_{model}$ → 0

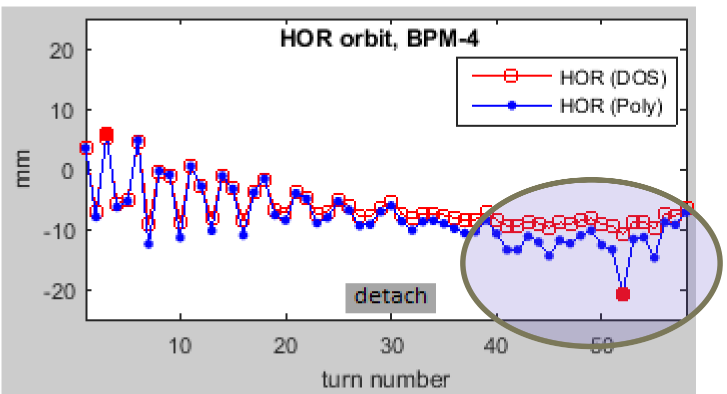


Compensation in the first turns.
Then becomes very noisy, why? Sum and Q are sensitive to beam size. It deteriorates without RF (blows up, bunch charge decrease, path & energy change).

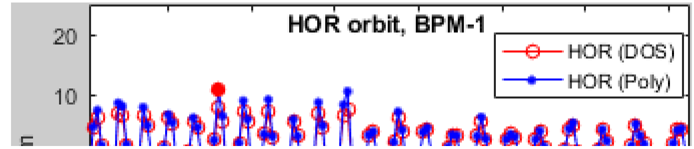
Injecting with RF off: Polynomials vs. DOS

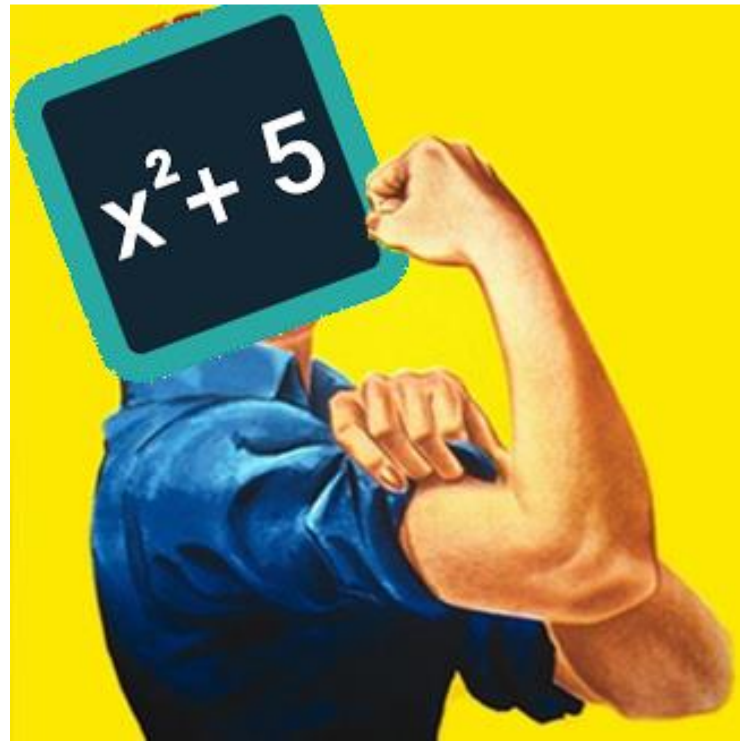
ESRF





ALBA





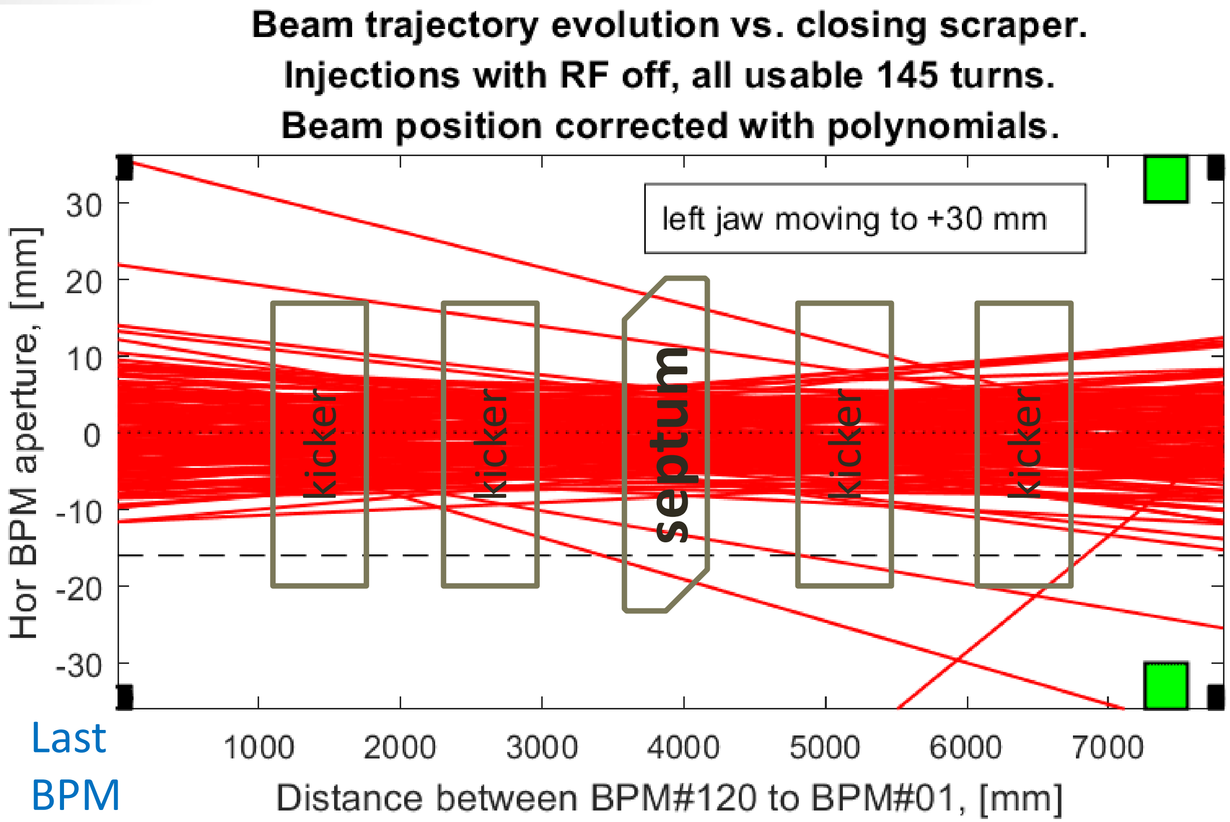
RF off → beam spirals inwar
Linear formula underestima
Polynomial can see farther (

Polynomial power!!!

So finally, polynomials or DOS?

Injecting with RF off: Scraper test

ALBA SR injection straight, PSA May2019



first BPM

Last BPM

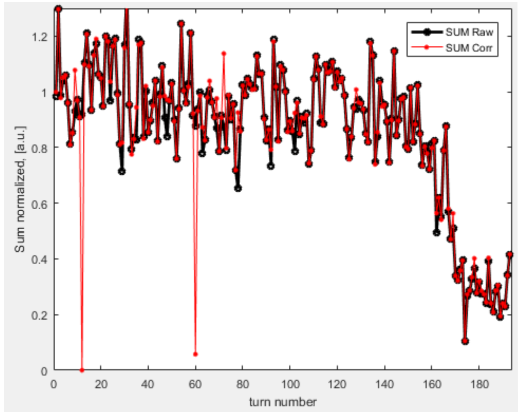
Each jaw kills beam at ±7 mm

DOS error ≤ 2.5 mm

Poly error ≤ 0.16 mm

← Septum blade

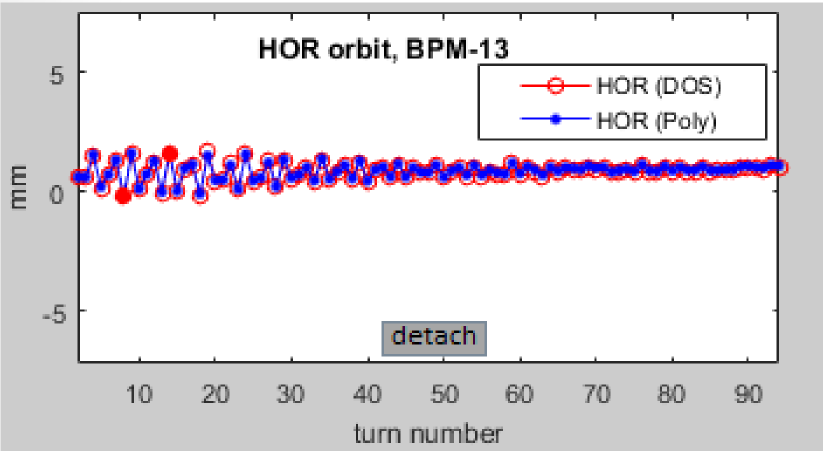
← ~8 meters →

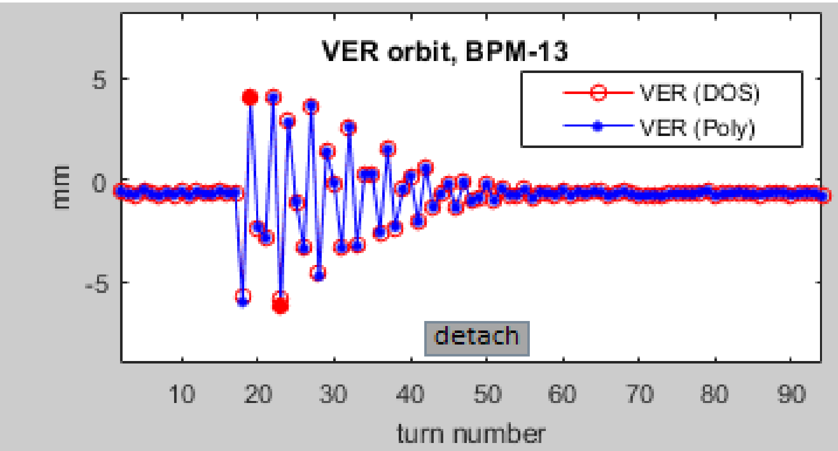


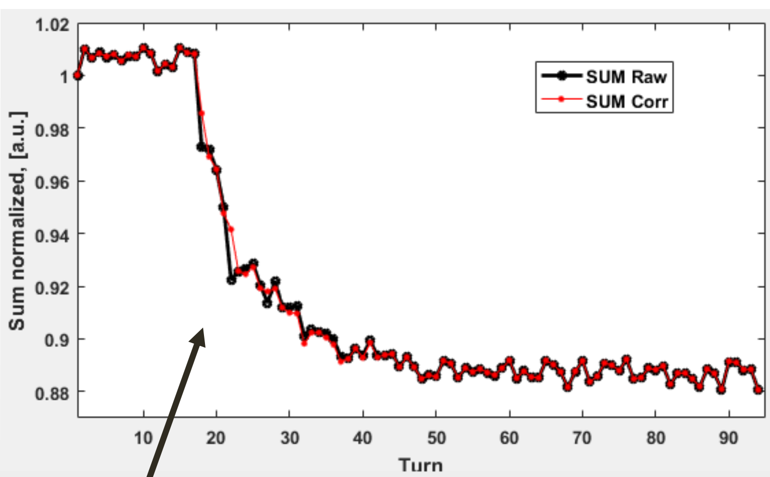
Beam kicks: vertical plane kick

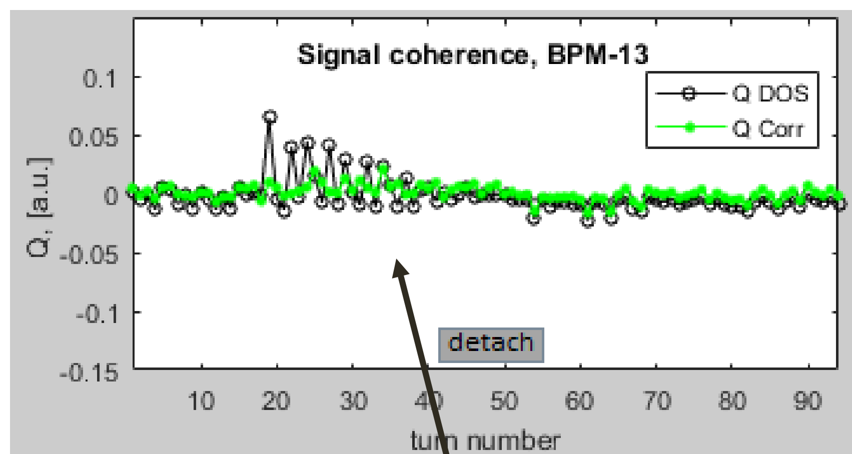
ESRF MDT Nov27: big V-kick, partial loss

MAF off
Anti-smearing on







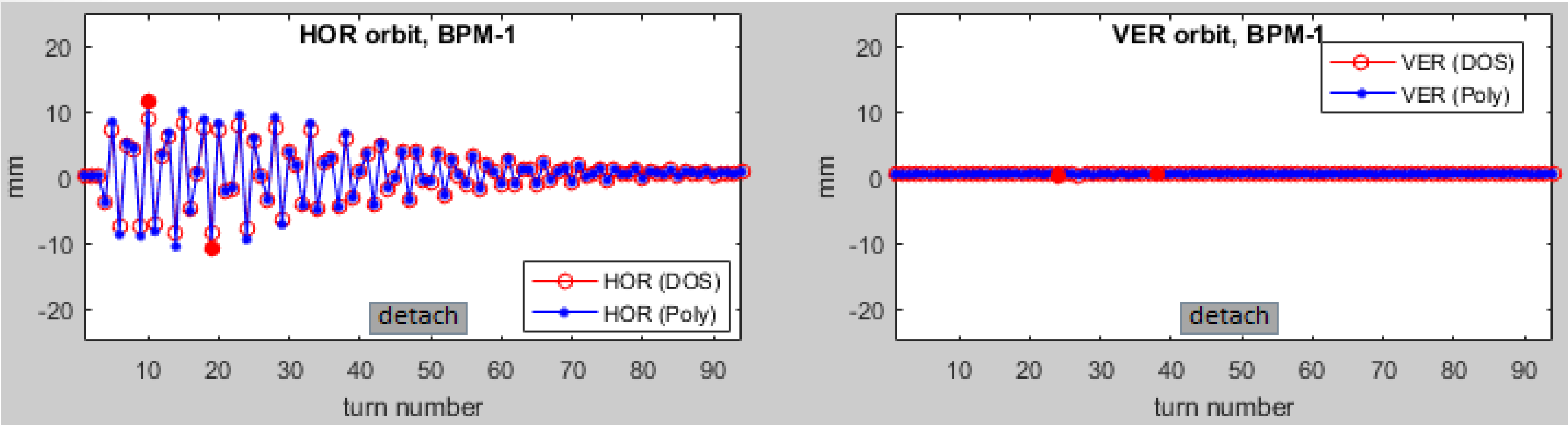


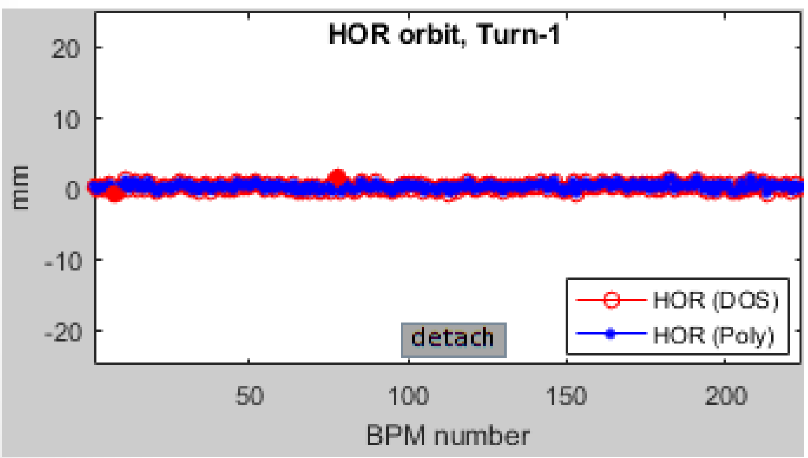
Note the behavior of the Sum during the V-kick (bunch length doesn't change?)

Very nice Q suppression!

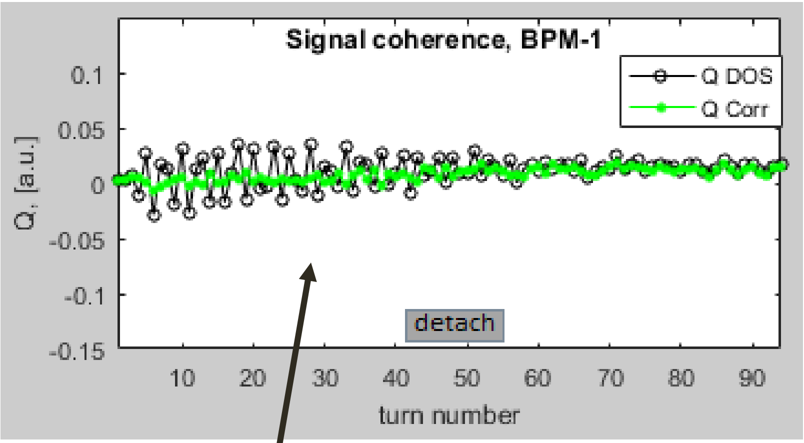
Beam kicks: horizontal plane kick

ESRF MDT Nov27: moderate H-kick, small loss





View of all BPMs

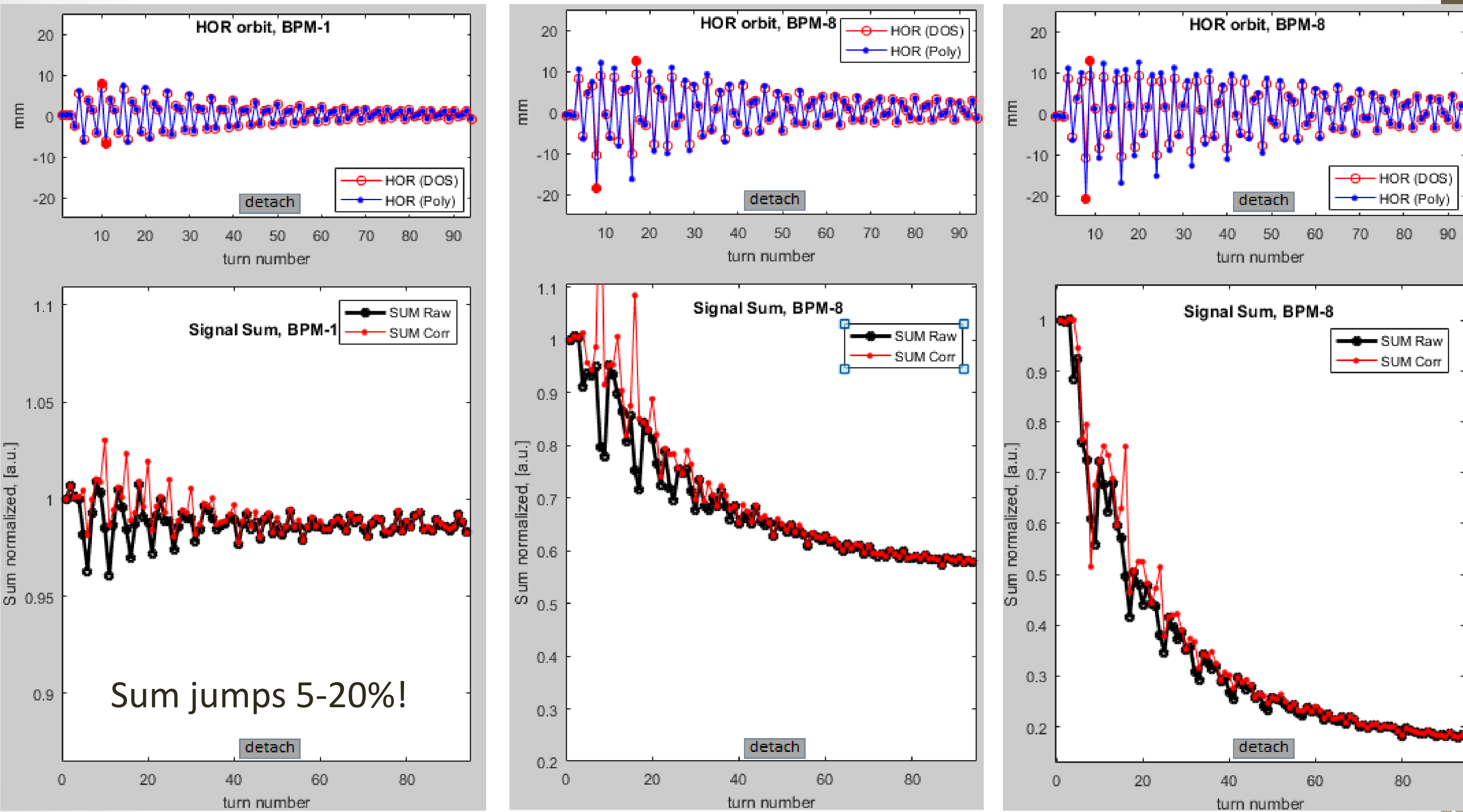


Again a very nice Q suppression

Beam kicks: horizontal plane kicks

ESRF MDT Nov27: H-kicks, loss variation

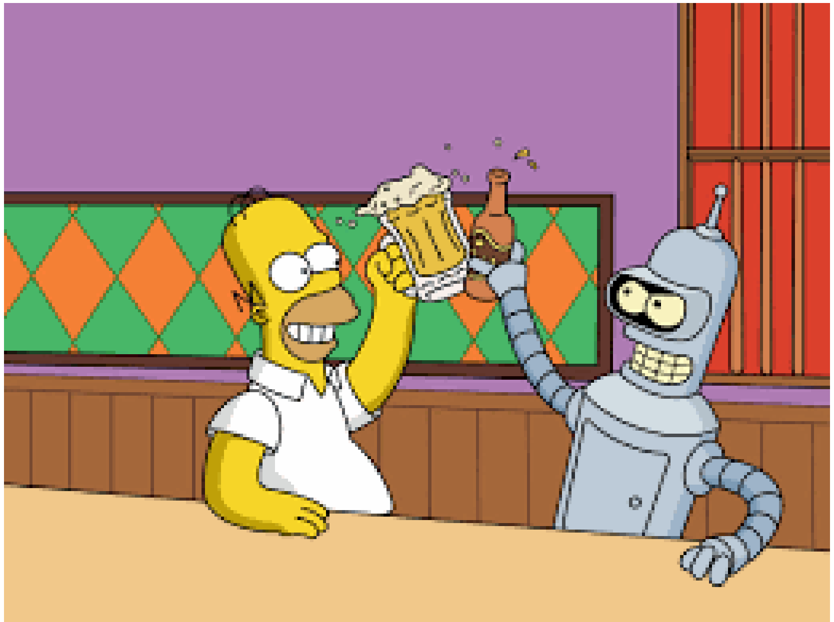
3 mA → ~3 mA | 3 mA → 1.5 mA | 3 mA → 0.5 mA



Sum oscillates only during H-kicks! Why?

Gets under/over compensated by SumPoly correction.

Thank you for your attention!

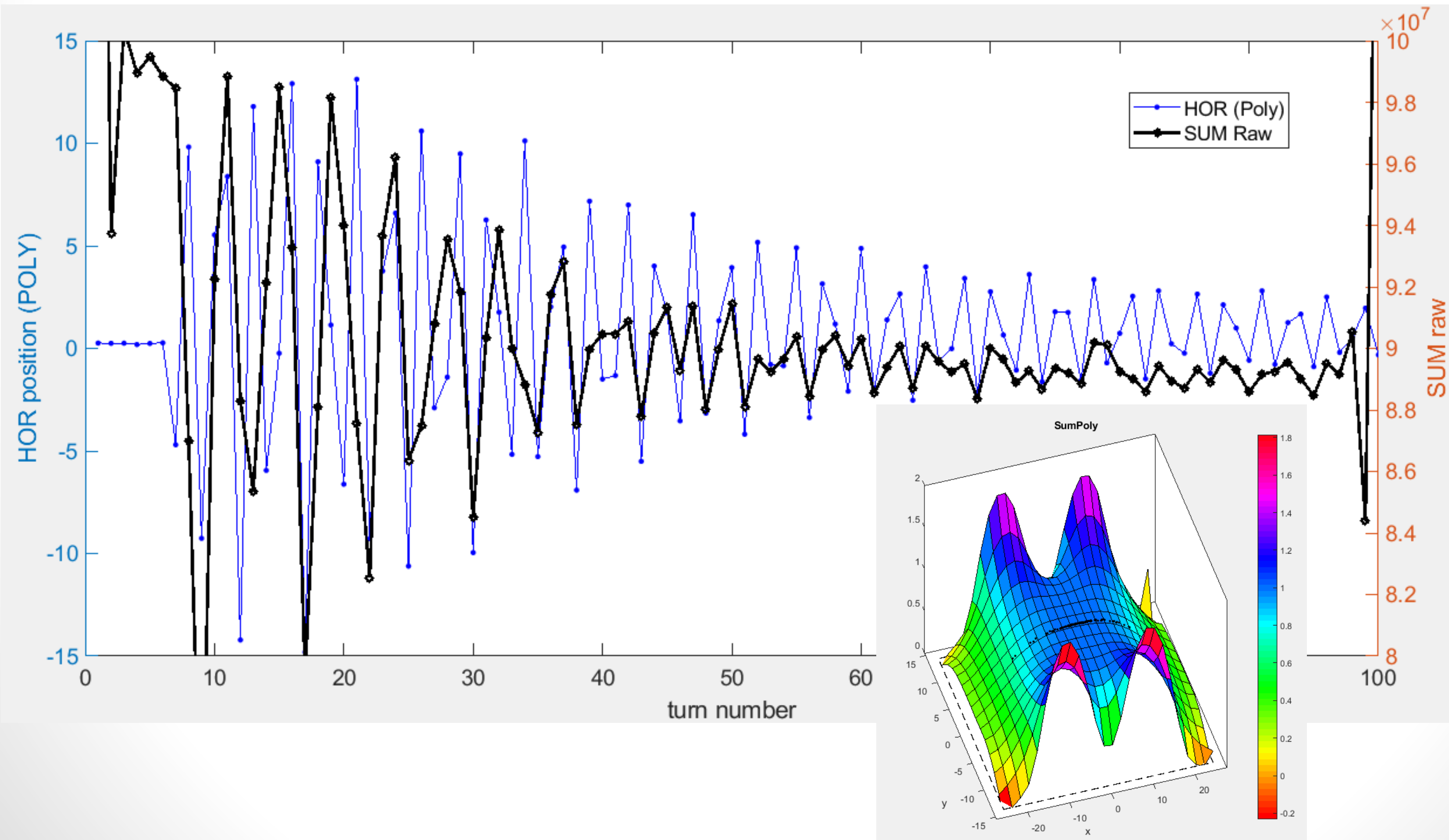


extras

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