Beamline Modeling Tools

Stephen Molloy*

(with much help from Glen White)

SLAC

16th December, 2008

*smolloy@slac.stanford.edu

Overview

- Development of a beamline modeling “toolkit”
  - A suite of matlab routines to help with tuning and trouble-shooting of ATF2 lattice
  - Developed within Flight Simulator
    - Part of the base FS installation
- Will discuss each of these in turn
  - Begin with “simple” tools, and progress to more complex
  - Usage instructions for these tools
  - Show some examples of simulated results

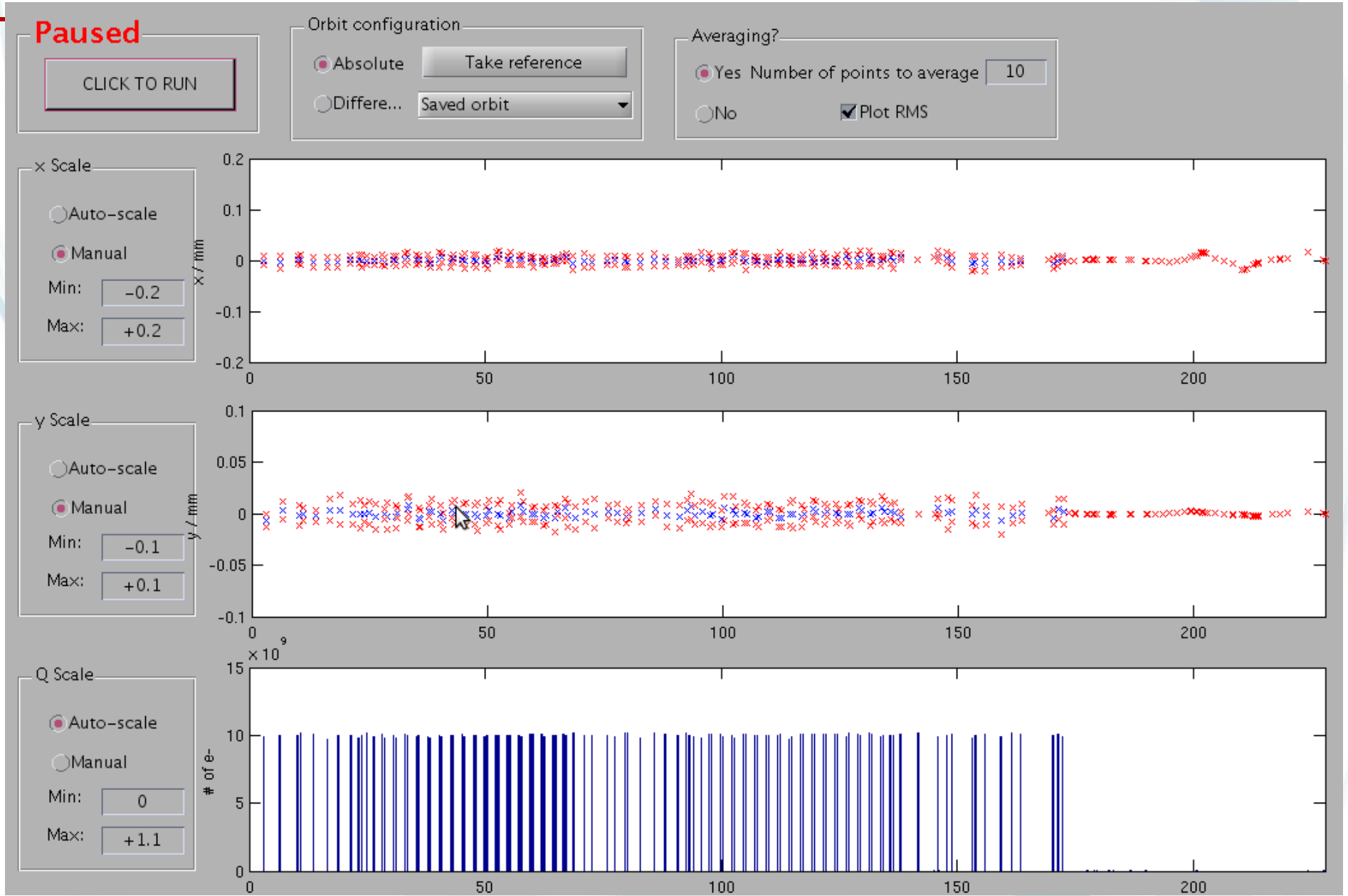
Tools under development

- BPM display
- Twiss calculation
  - Theoretical & measured
- Bump generation
- Beam based alignment
  - Quads (Ext & FF) and sexts (FF)
  - Movers or beam bumps
- Lattice verification
  - $R_{(1/3,2/4)}$ measurement and magnet strength fitting

A few details

- Generic set of tools for use with the FS
  - Integrated with FS **right from the start**.
  - Developed entirely within the FS environment to ease transition to ATF2 control room
- Tracking or modeling calls use Lucretia libraries
- Most tools are already uploaded to Lucretia CVS server
  - Various stages of development, and testing

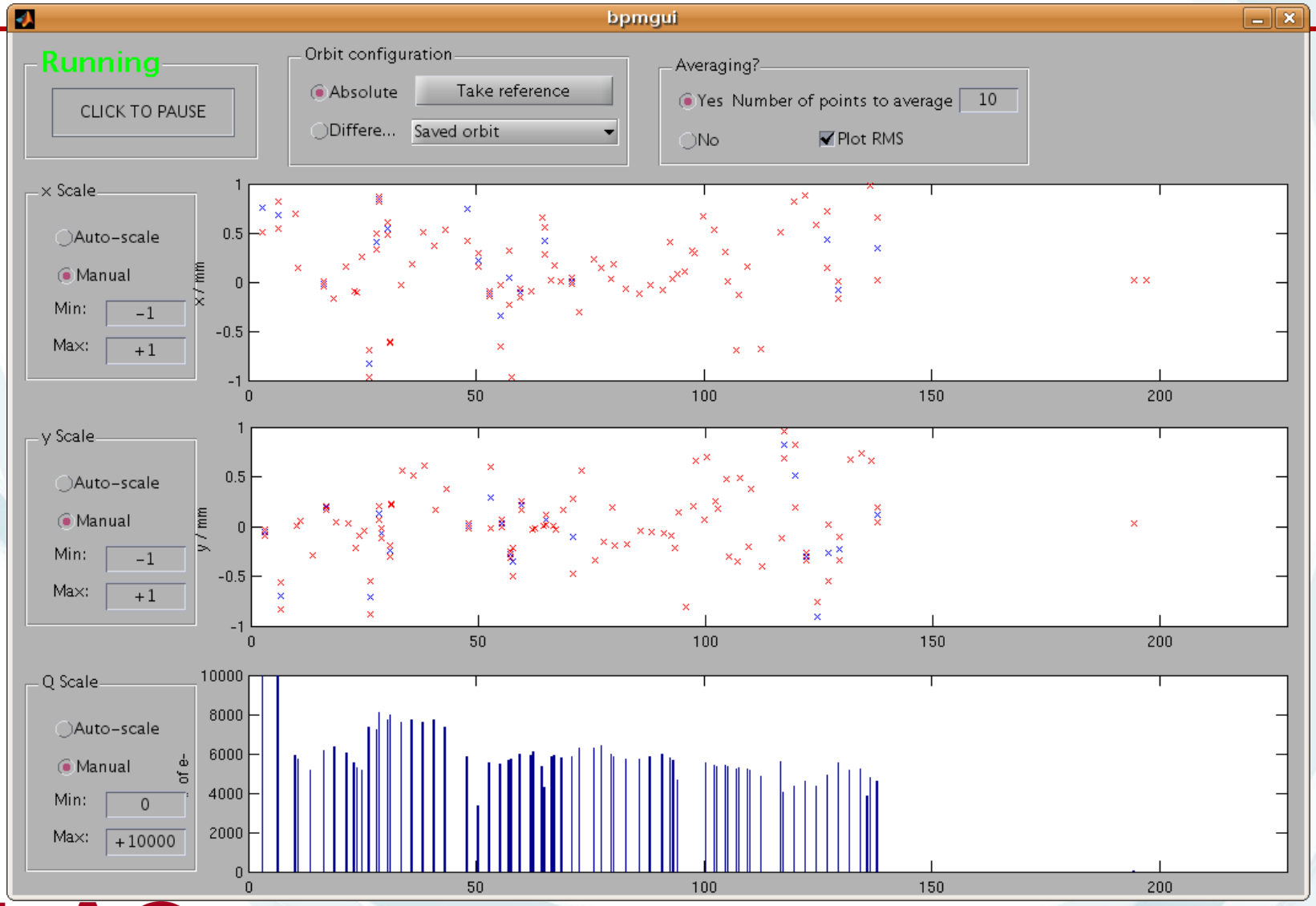
BPM Display


Paused
CLICK TO RUN
Orbit configuration
Absolute
Take reference
Differe...
Saved orbit
Averaging?
Yes
Number of points to average
10
No
Plot RMS
x Scale
Auto-scale
Manual
Min:
-0.2
Max:
+0.2
x / mm
y Scale
Auto-scale
Manual
Min:
-0.1
Max:
+0.1
y / mm
Q Scale
Auto-scale
Manual
Min:
0
Max:
+1.1
x 10^9
# of e-

BPM Display

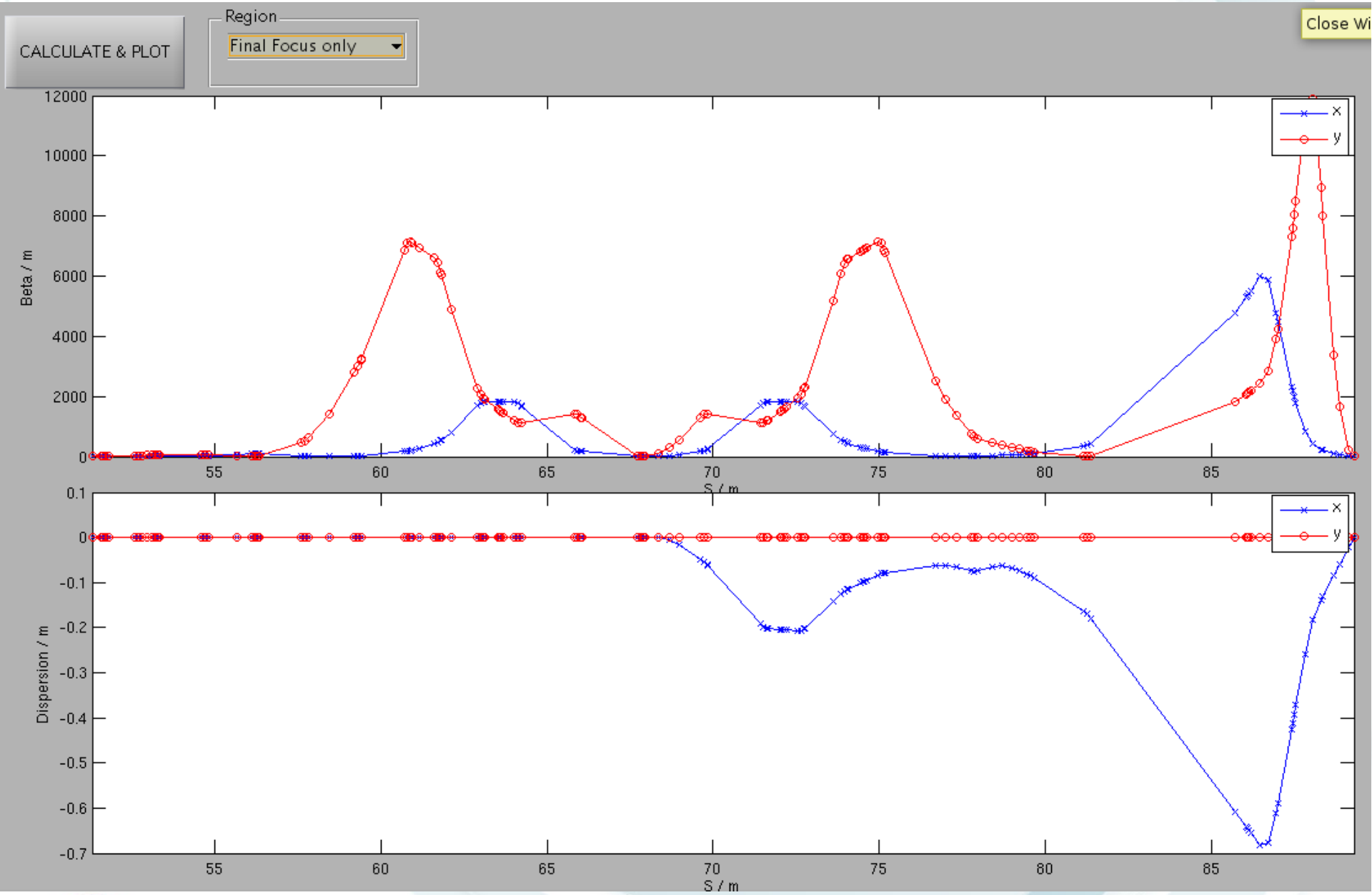
- Current functionality
  - Display all BPMS
  - Single-shot, or averaging
    - Mean and rms
  - Absolute or difference to a reference
  - Auto/manual scaling
- Still to do
  - Only plot one region
    - e.g. only FF BPMs
  - Save reference to file
  - Orbit fitting

BPM Display – real beam!



Multibunch beam & uncalibrated BPMs

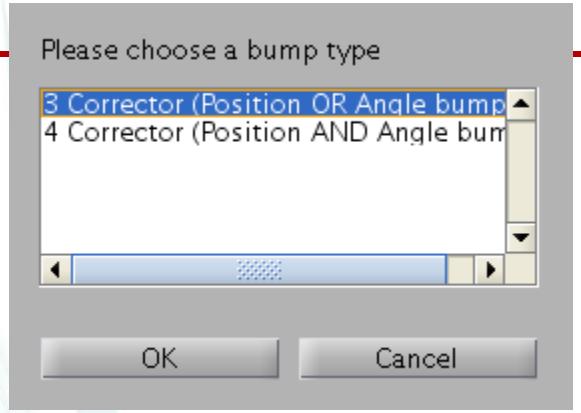
Twiss Calculation (very basic)

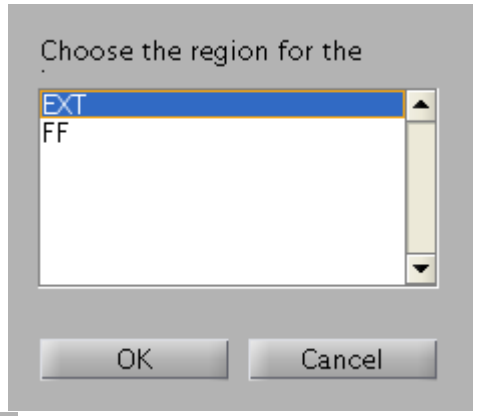


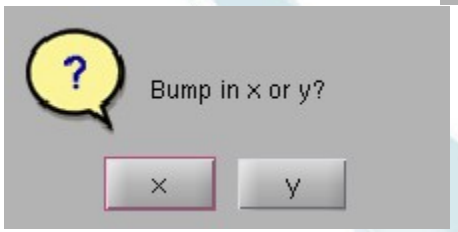
Twiss Calculation

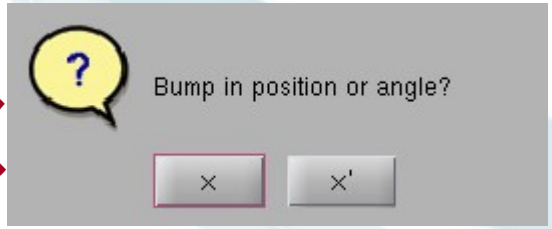
- Current functionality
  - Calculate and display model Twiss
    - Beta & dispersion (x/y)
  - Choose region
    - EXT, FF, both
- Still to do
  - Measurement
    - Extrapolate from model
  - Try to fit machine errors
    - e.g. quad errors causing mismatch
  - Display alternative twiss
    - e.g. alpha, phase advance.

Bump Generation



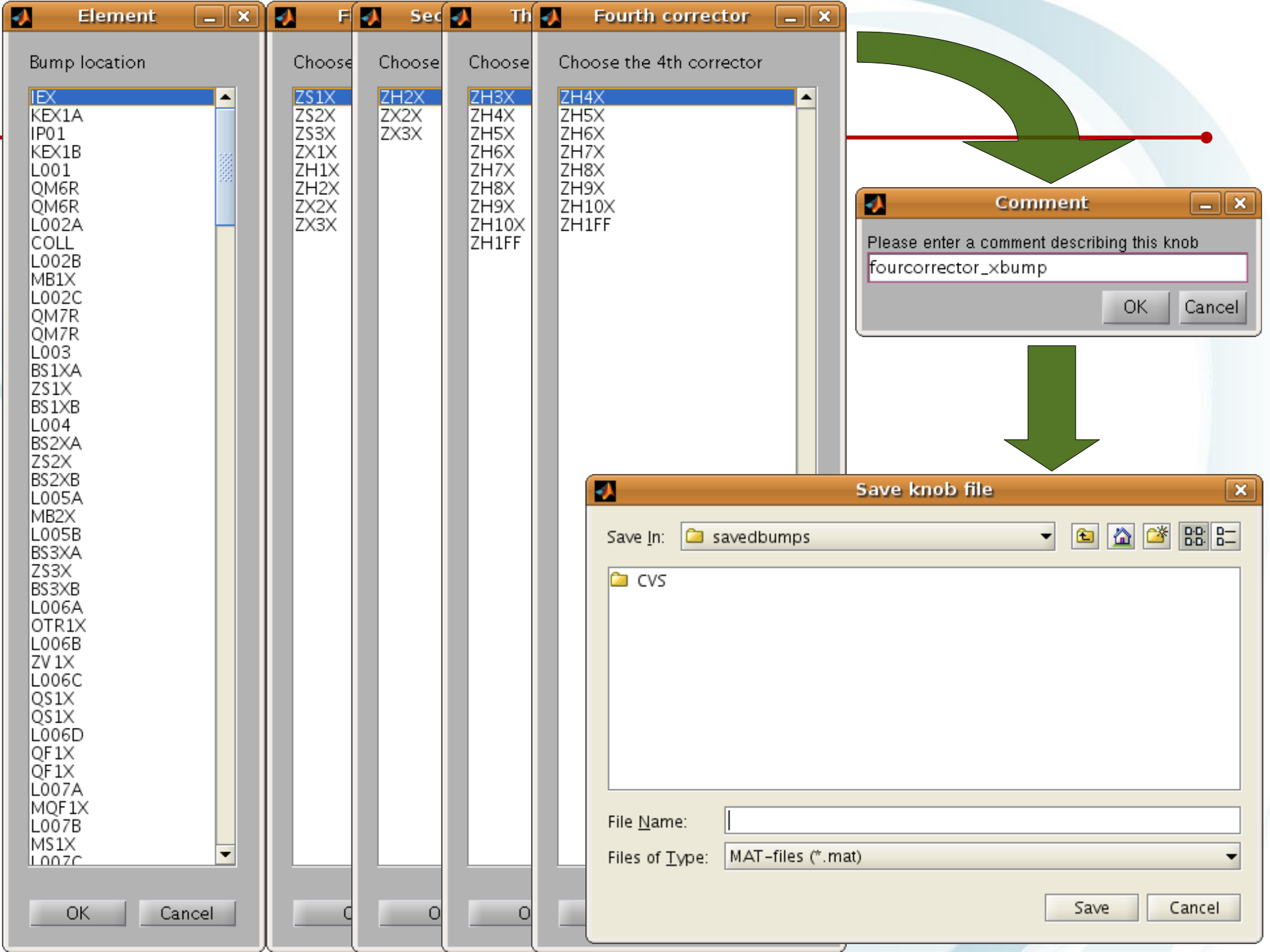






- Call directly from the command line
- Or, step through a series of dialogue boxes
  - Bump type, location, plane, correctors, etc.
- Makes a Lucretia knob
  - Use directly in FS

SLAC


Element
Bump location
IEX
KEX1A
IP01
KEX1B
L001
QM6R
QM6R
L002A
COLL
L002B
MB1X
L002C
QM7R
QM7R
L003
BS1XA
ZS1X
BS1XB
L004
BS2XA
ZS2X
BS2XB
L005A
MB2X
L005B
BS3XA
ZS3X
BS3XB
L006A
OTR1X
L006B
ZV1X
L006C
QS1X
QS1X
L006D
QF1X
QF1X
L007A
MQF1X
L007B
MS1X
OK
Cancel
Choose
ZS1X
ZS2X
ZS3X
ZX1X
ZH1X
ZH2X
ZX2X
ZX3X
Choose
ZH2X
ZX2X
ZX3X
Choose
ZH3X
ZH4X
ZH5X
ZH6X
ZH7X
ZH8X
ZH9X
ZH10X
ZH1FF
Fourth corrector
Choose the 4th corrector
ZH4X
ZH5X
ZH6X
ZH7X
ZH8X
ZH9X
ZH10X
ZH1FF
Comment
Please enter a comment describing this knob
fourcorrector_xbump
OK
Cancel
Save knob file
Save In:
savedbumps
CVS
File Name:
Files of Type:
MAT-files (*.mat)
Save
Cancel

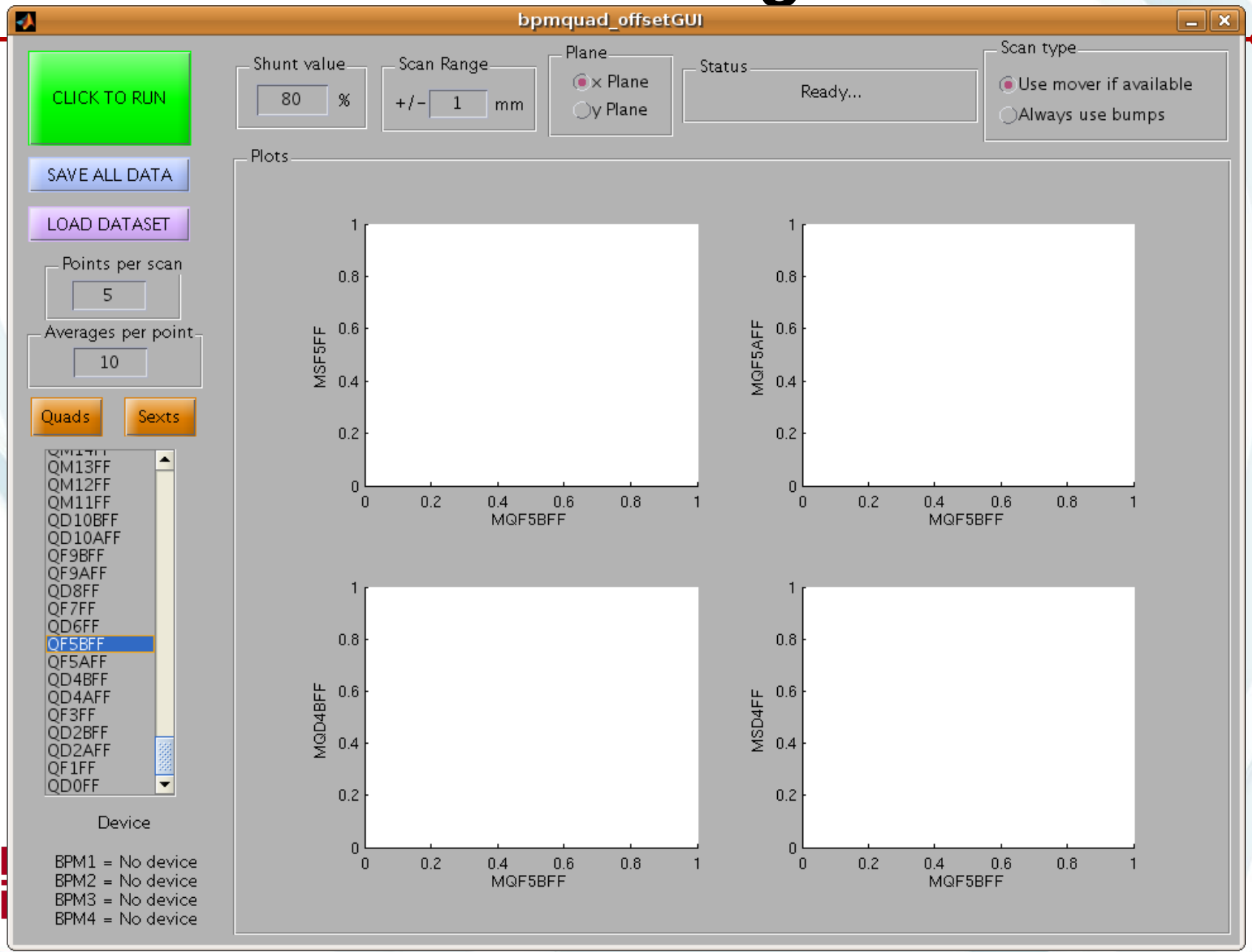
Bump Generation

- Calculate bump from command-line options, or GUIs
  - Code makes checks on inputs
    - e.g. bump location must be between correctors!
- Generates a Lucretia “knob”
  - A matlab structure containing corrector names, ratios, etc.
  - Can be used with Lucretia functions
    - “*SetMultiKnob*”, “*IncrementMultiKnob*”, & “*RestoreMultiKnob*”

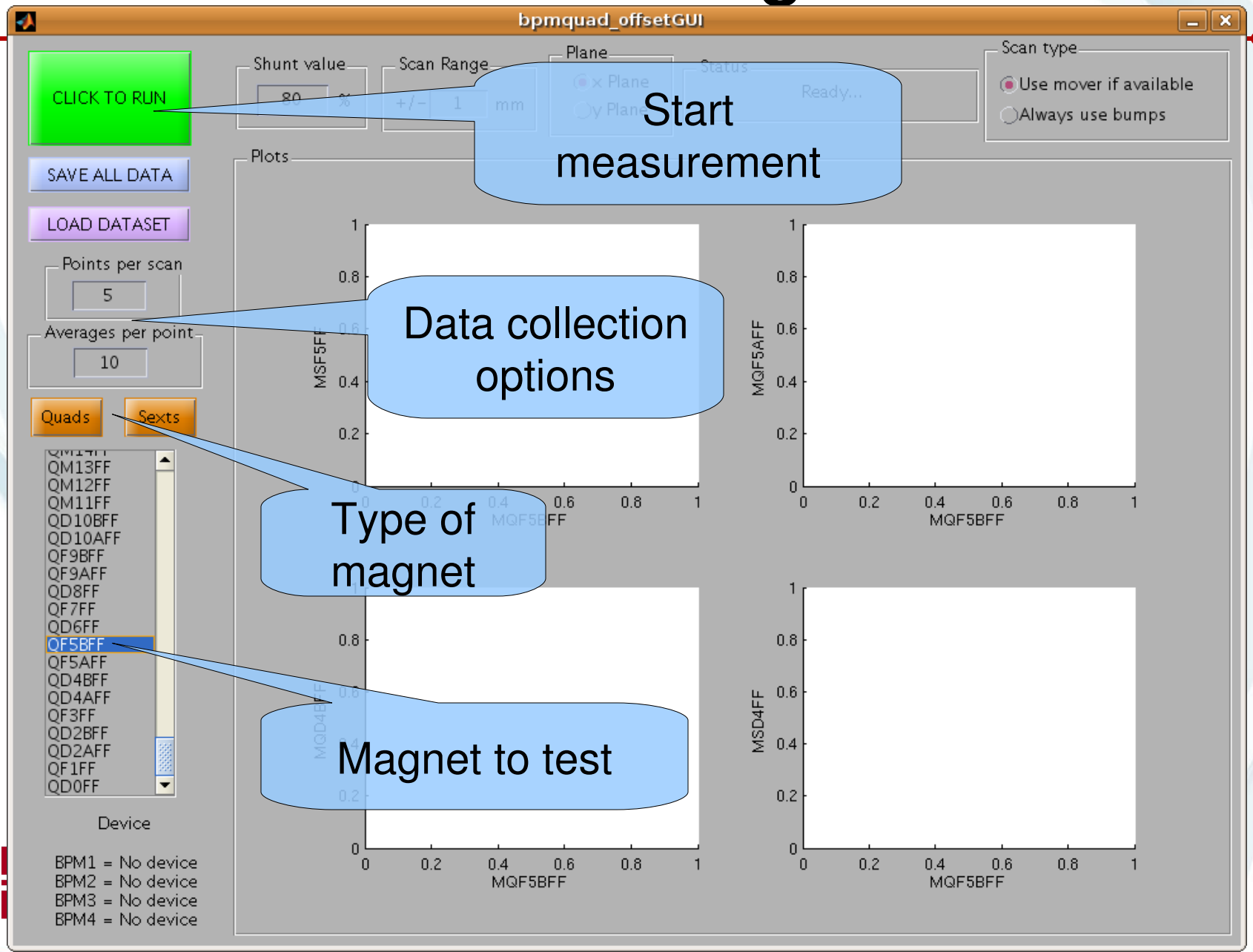
Bump Generation

- Current functionality
  - 3 or 4 correctors
  - Position or angle (x/y)
  - Save to file
  - Command-line function
    - For integration into other apps
    - Correct input verification
- Still to do
  - Assumes non-coupled lattice
    - Useless in strongly coupled regions
  - *Assumes linear transfer matrices*
    - *Sexts are problematic*

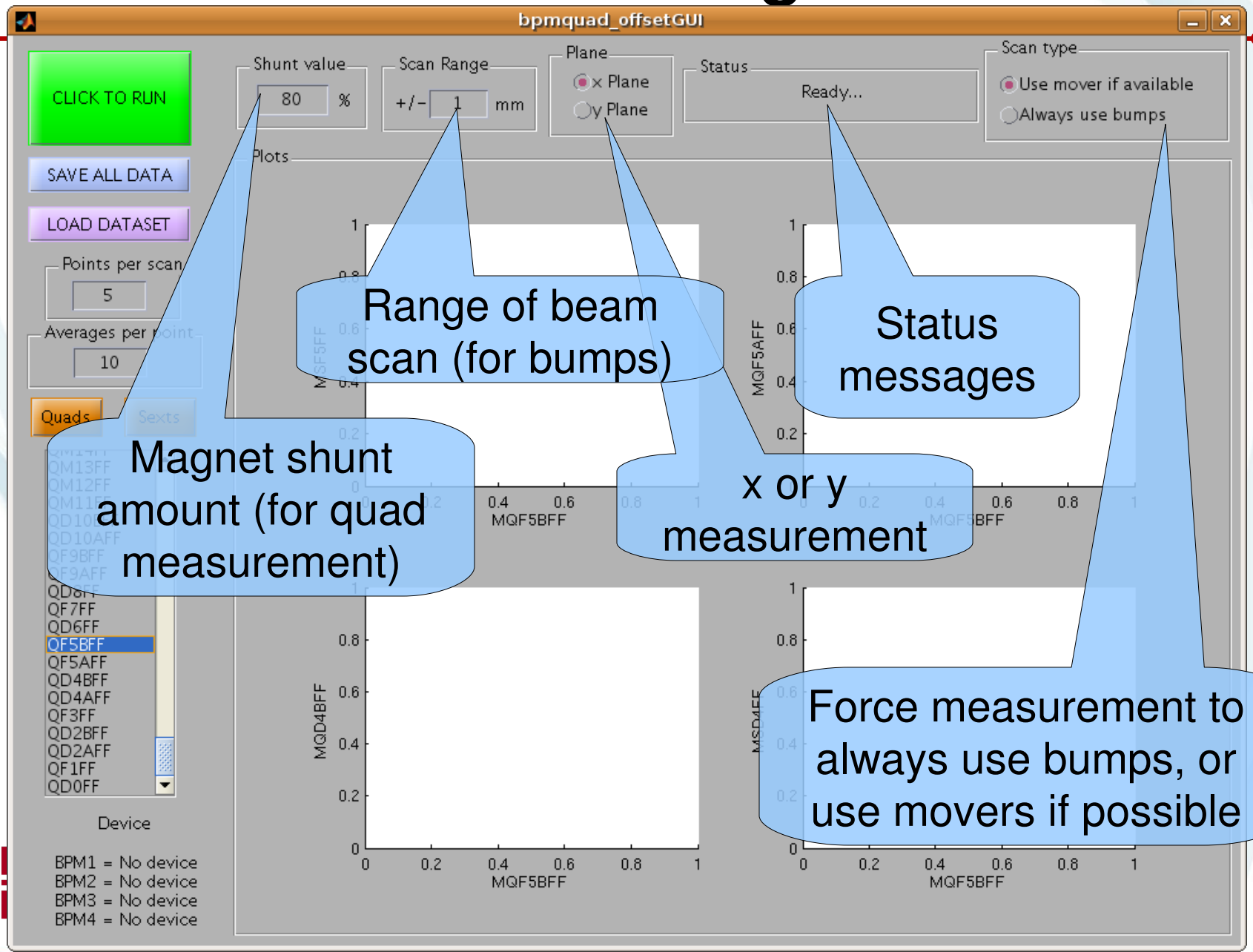
Beam-Based Alignment



Beam-Based Alignment



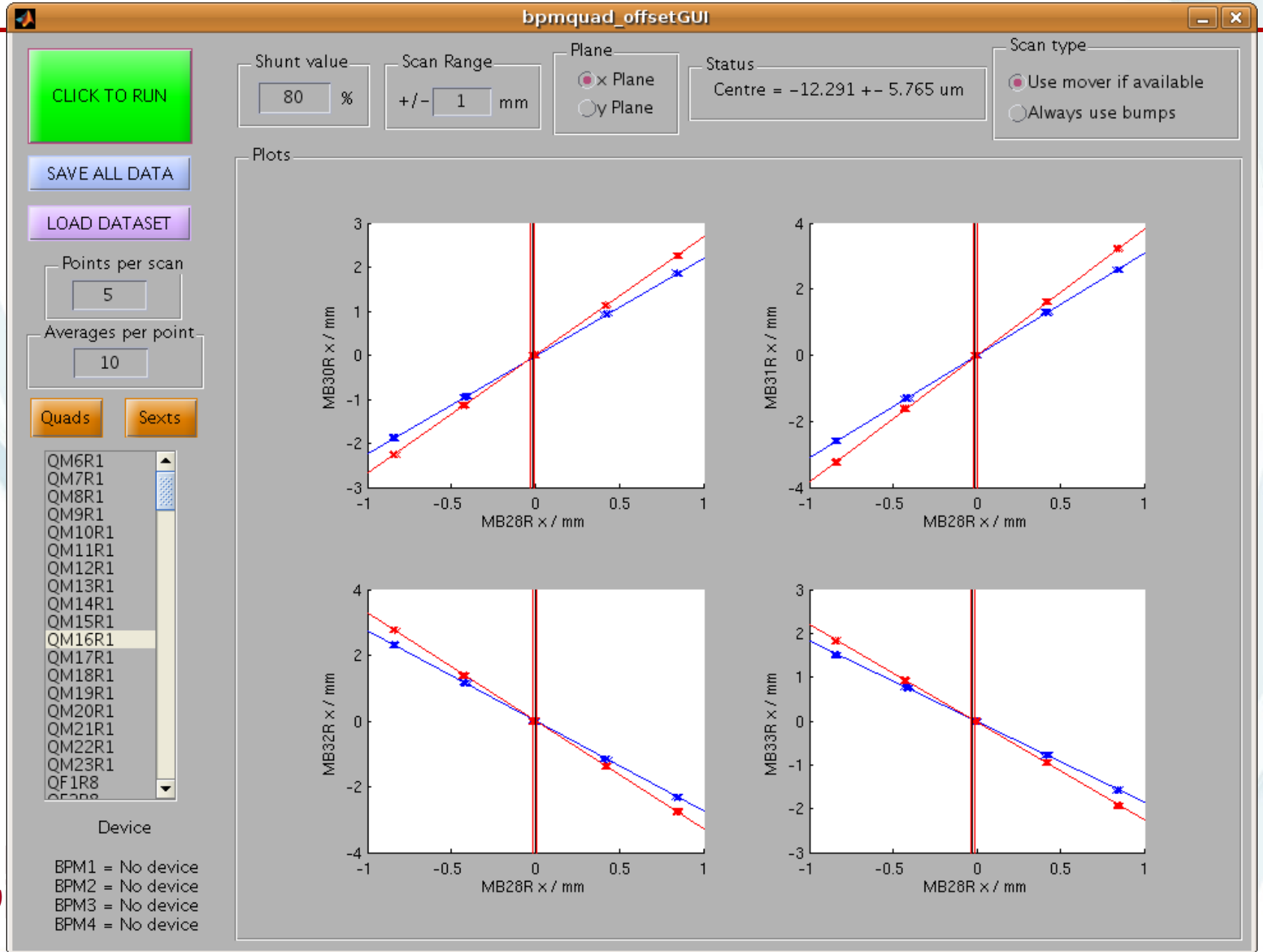
Beam-Based Alignment



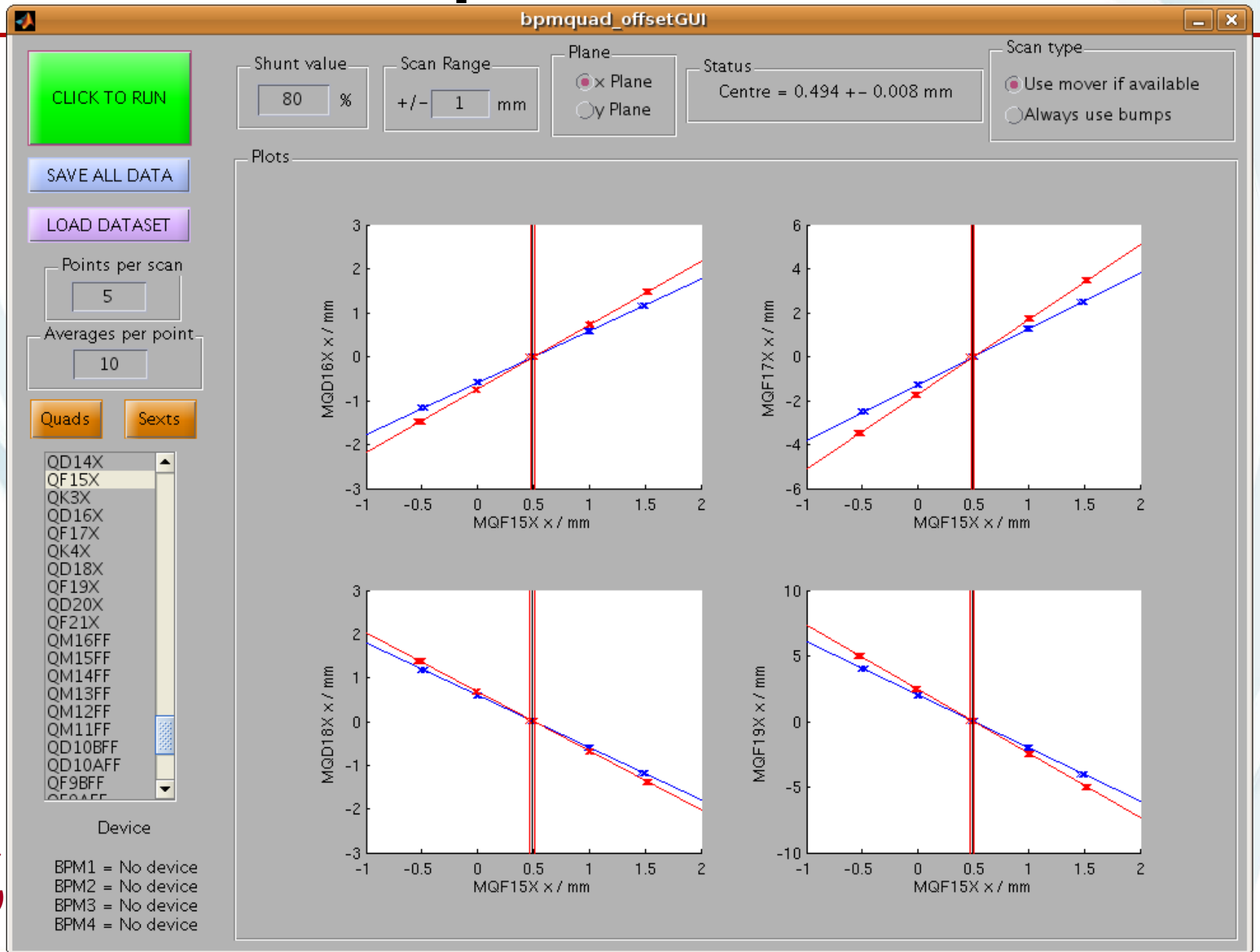
BBA – Quads without movers

- Calculate a 4-corrector bump around the quad
  - Makes use of the bump GUI tool
- Record BPM data at several bump settings
    - At associated BPM, & four closest downstream BPMs
  - Fit straight line through this data
- Shunt the quad, and repeat bump measurement
- Quad centre is at the crossing point of the fits

BBA – Ring quadrupole



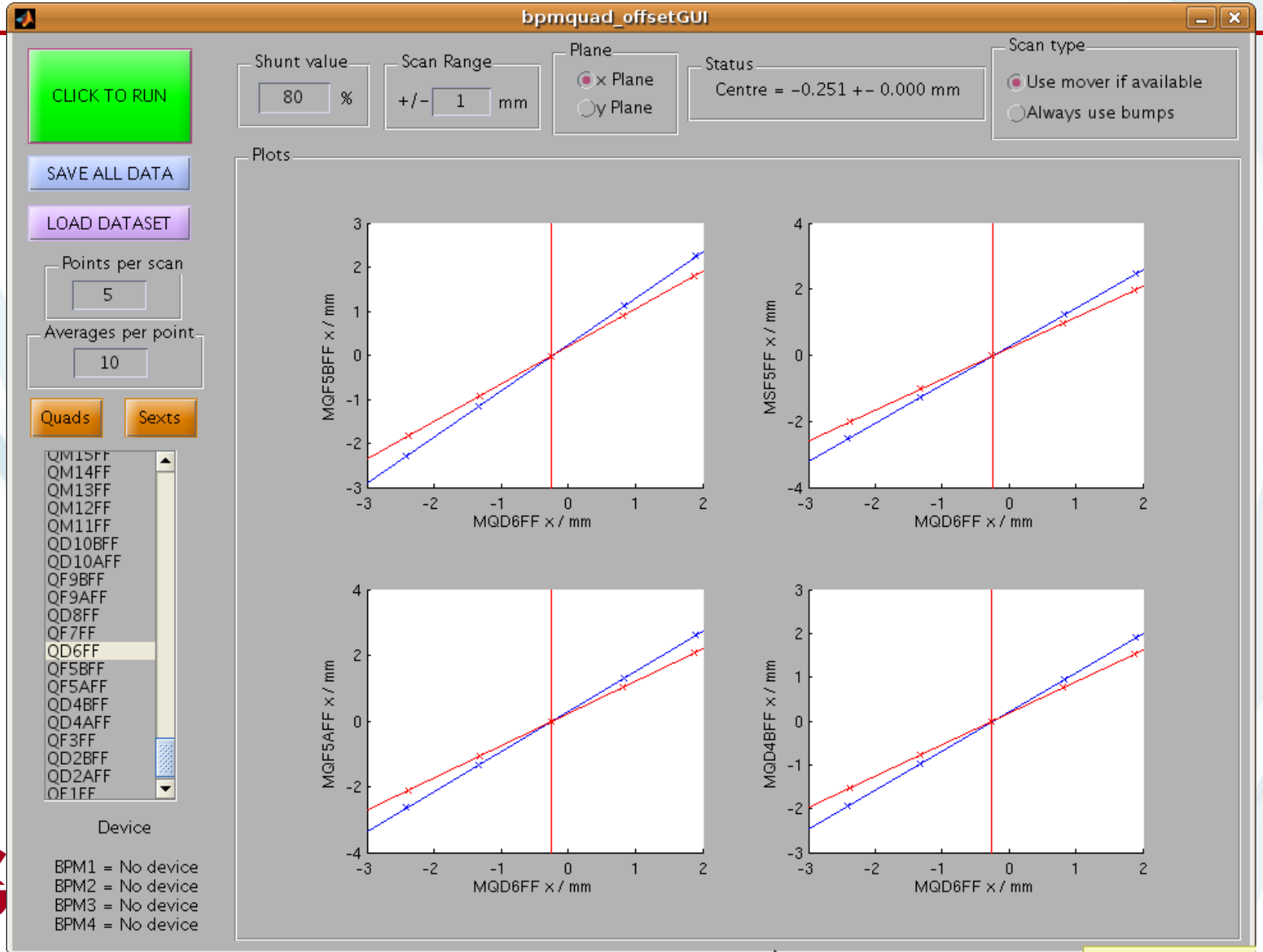
BBA – EXT quad with 0.5 mm error



BBA – Quads with movers

- Quads on movers may be aligned without bumping the beam
  - Step the mover through its max range of motion
    - Calculate the range from the current cam positions
    - Ignore the front panel “Scan Range” field
  - Record BPM data
    - Associated BPM, and next four
  - Shunt the quad, and repeat
- Centre is at crossing point

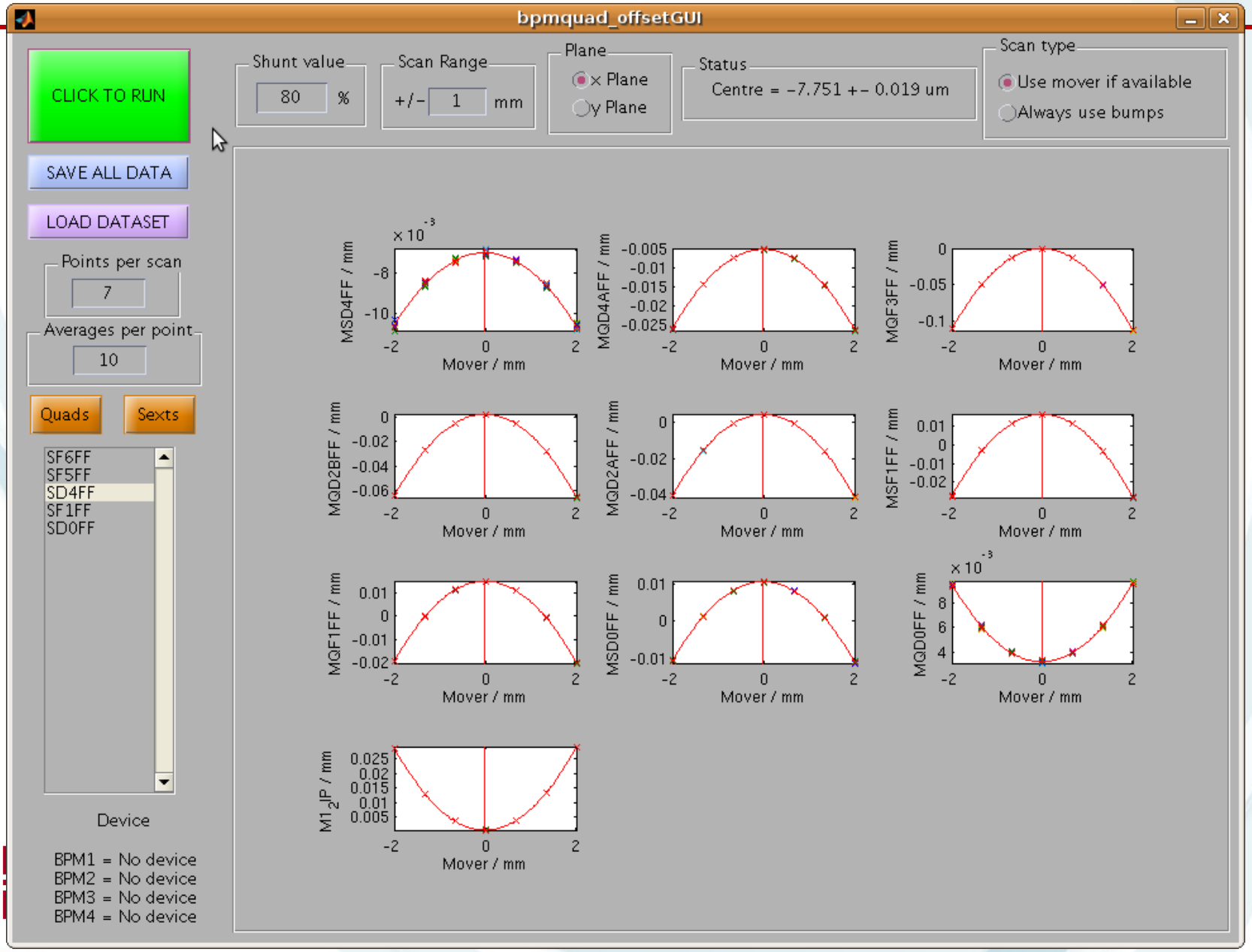
BBA – FF Quad, -0.25 mm error



BBA – sextupoles

- Only working on FF sexts
  - If necessary I can include ring sexts
  - All on movers, so no need to bump beam
- Step mover through a range of positions
  - Based on number of steps, and max range possible
- Fit parabola to **all** downstream BPMs
  - Find the min (or max) of each
  - Centre is error-weighted mean of these results

BBA – sextupoles



Lattice Verification

- Two steps
  - Measurement and fitting

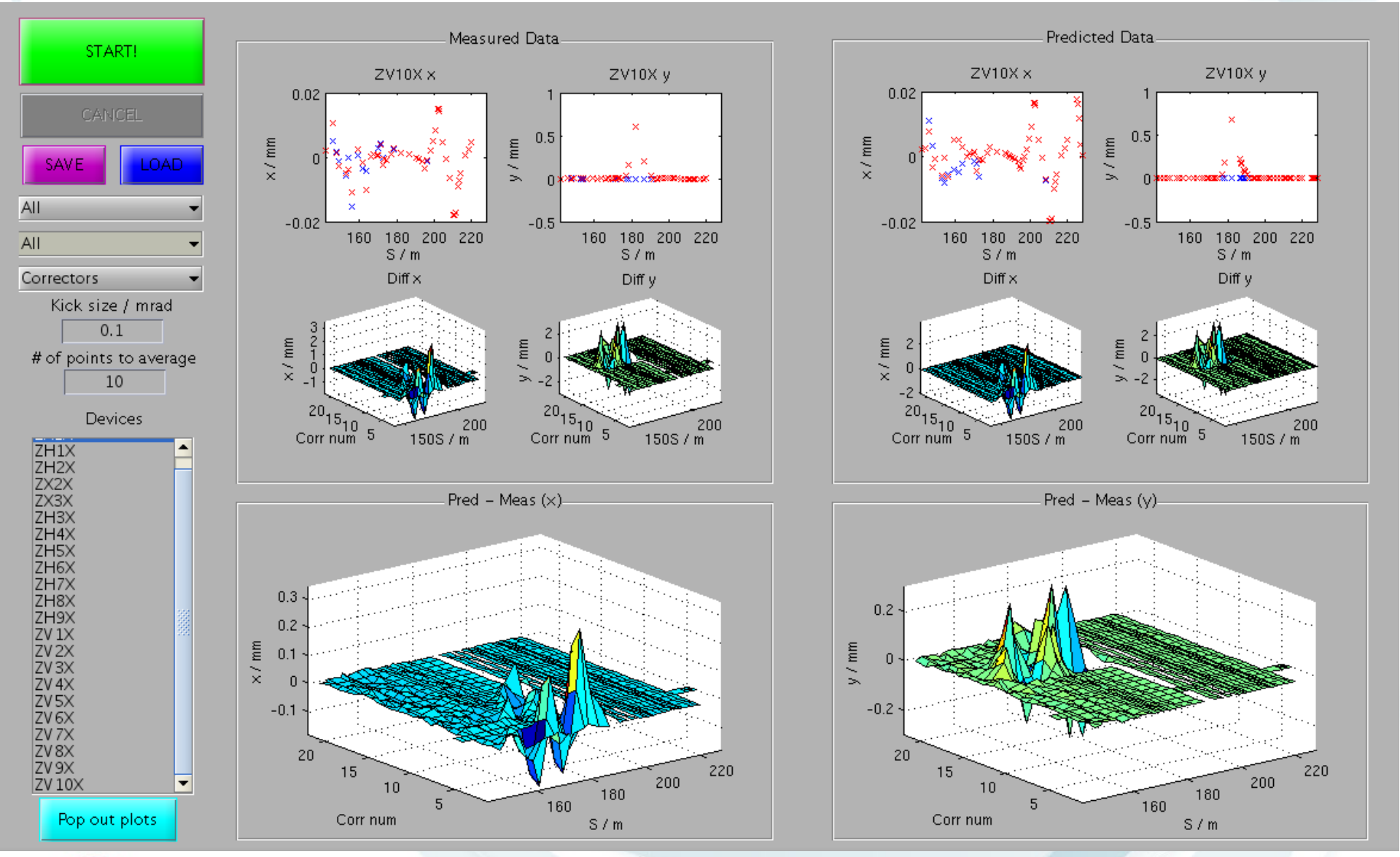
1 Measurement

  - Tweak each corrector in turn, recording BPMs
    - Close bump if necessary
    - Tweak quad movers to measure FF

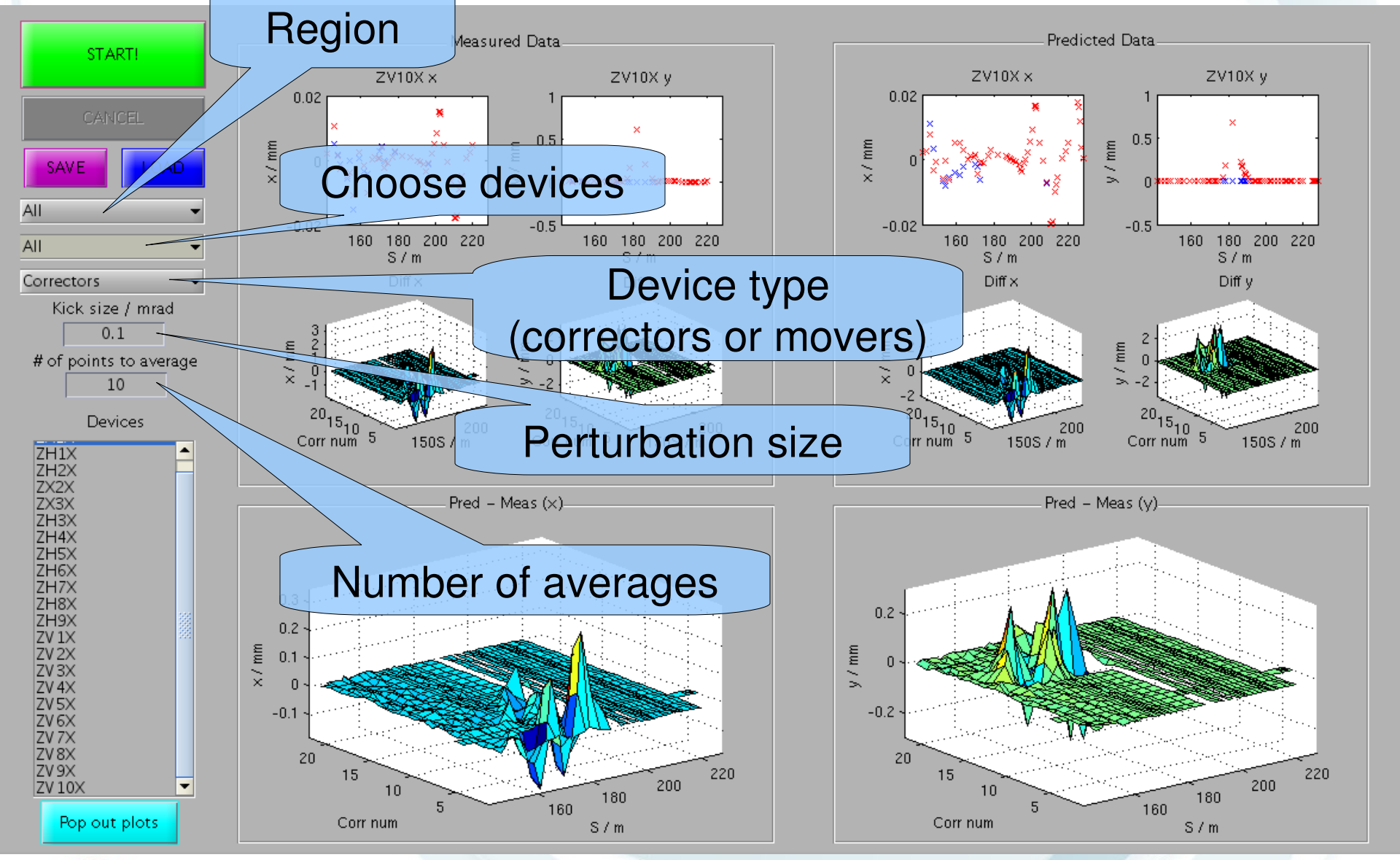
2 Fitting

  - Calculate difference between predicted and measured response
  - Fit quad errors to minimise this difference

Lattice Verification – Measurement



Lattice Verification – Measurement

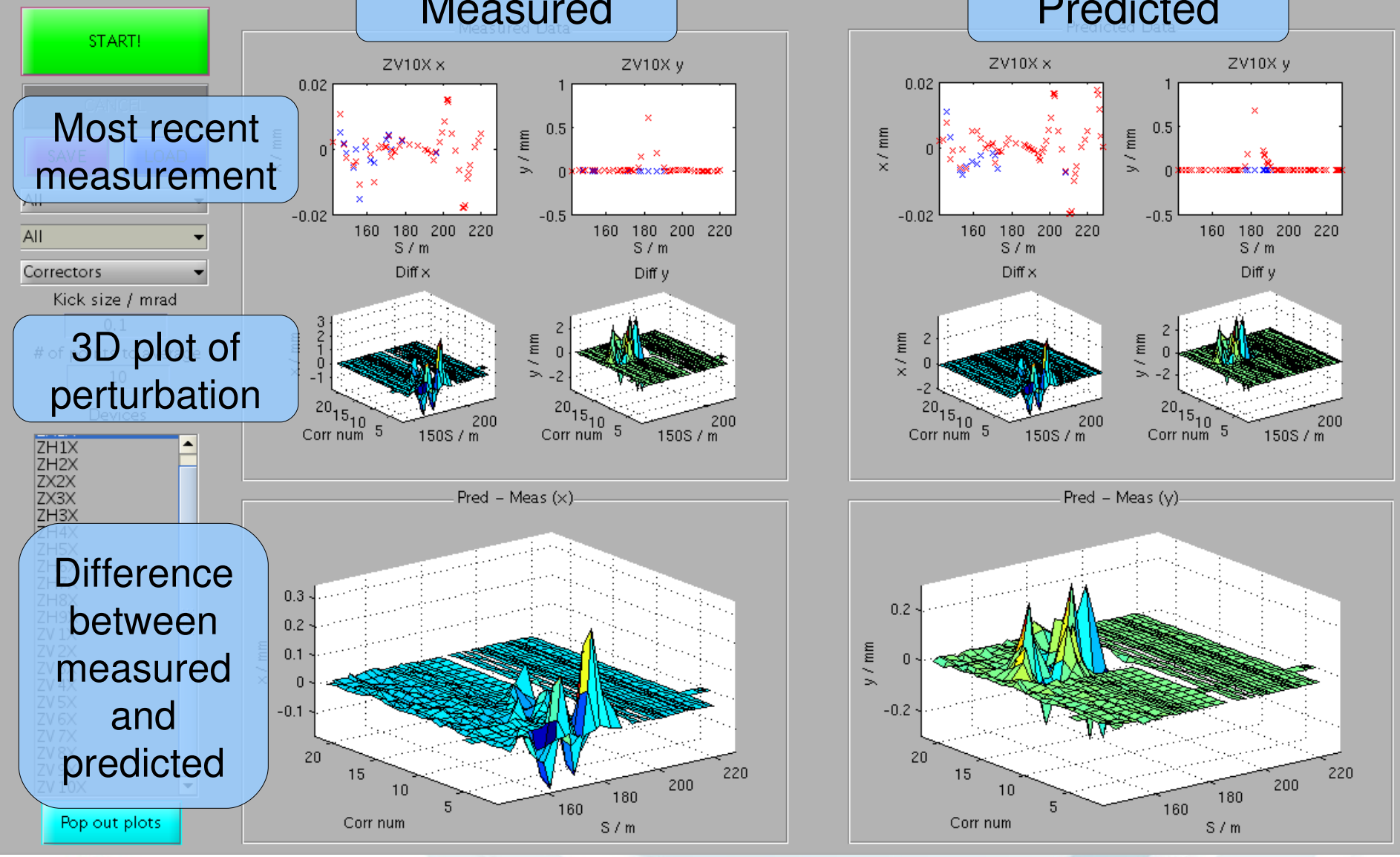


Lattice Verification – Measurement

Most recent measurement

3D plot of perturbation

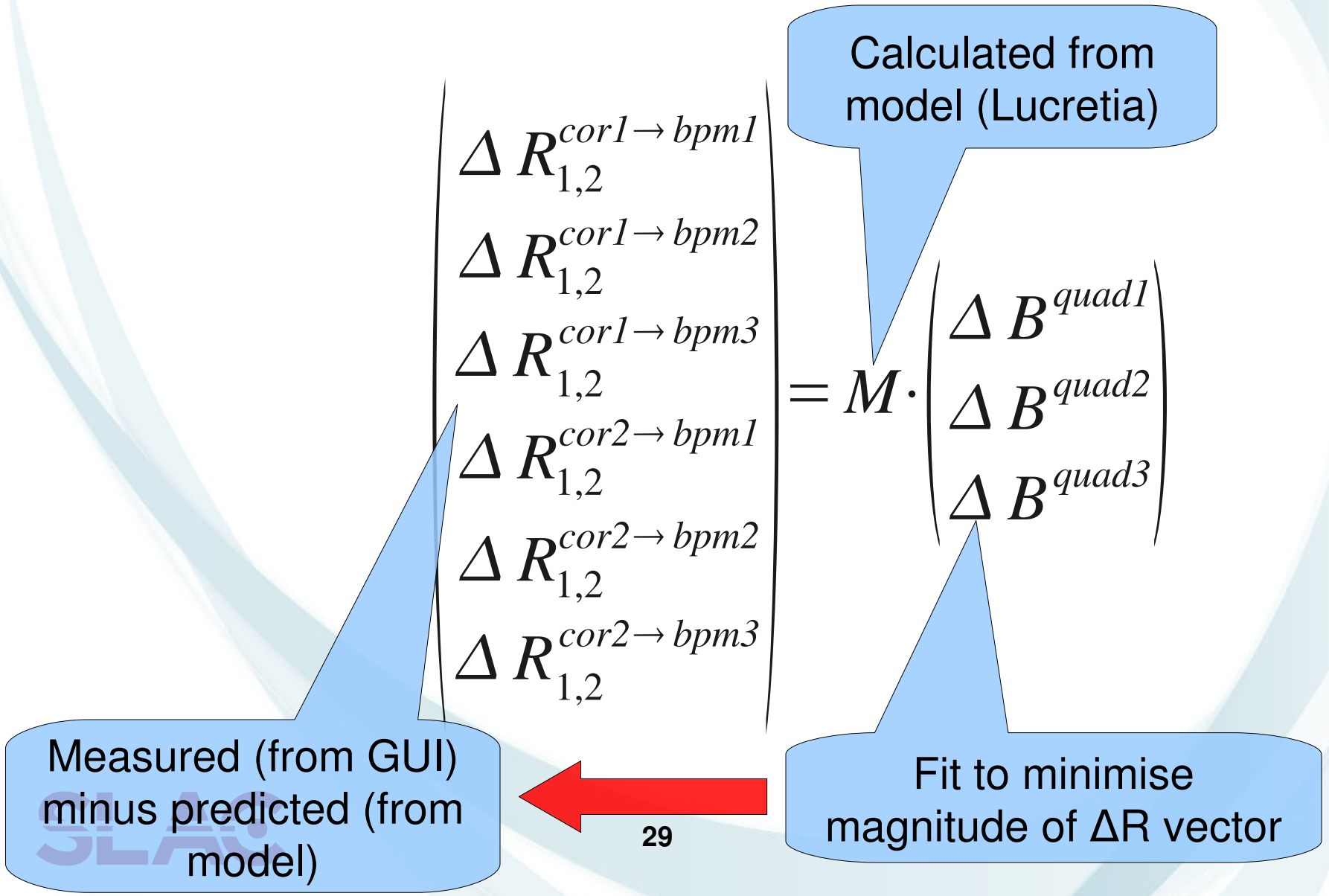
Difference between measured and predicted



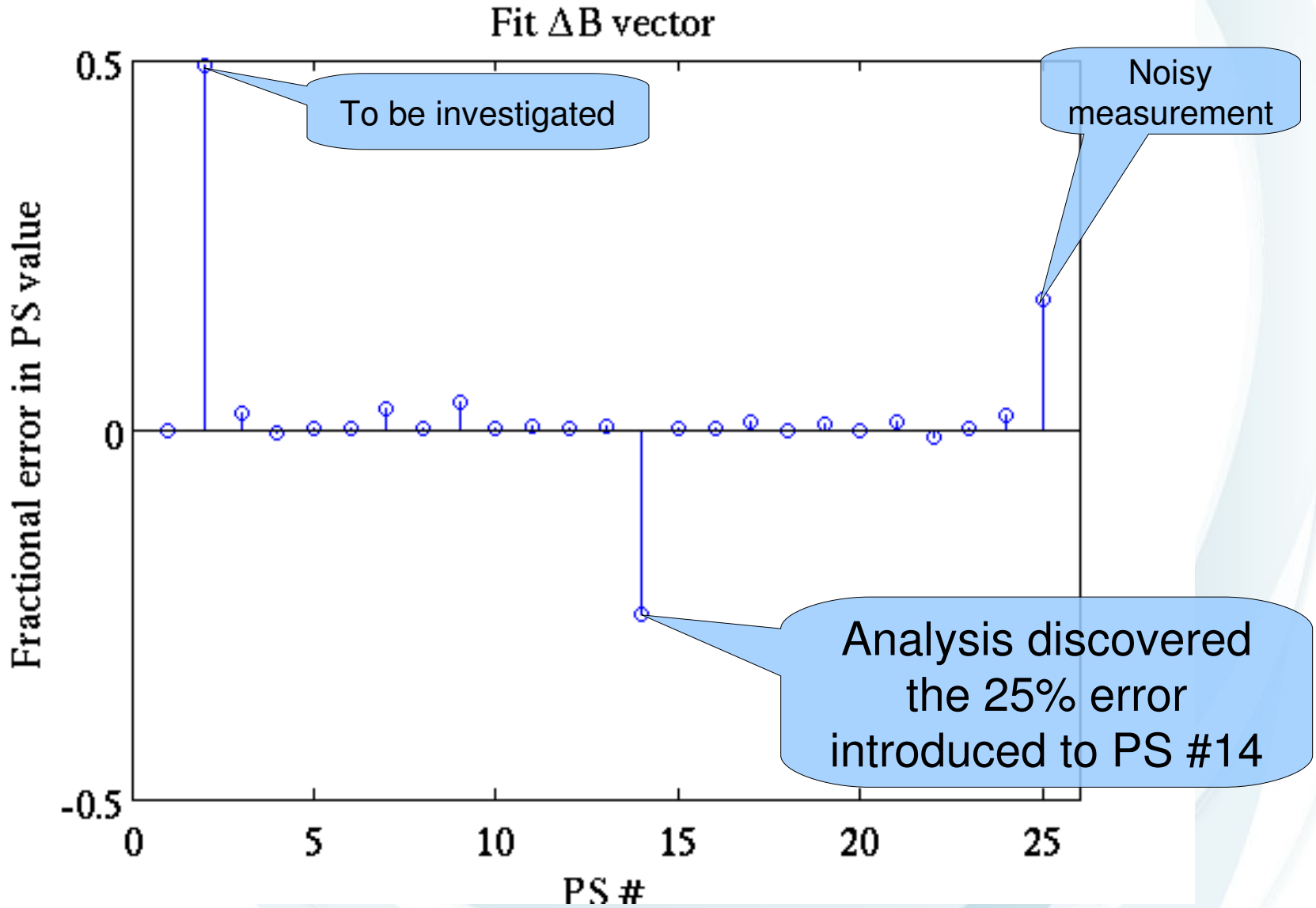
Lattice Verification – Analysis (1st attempt)

$$\begin{pmatrix} \Delta R_{1,2}^{cor1\rightarrow bpm1} \\ \Delta R_{1,2}^{cor1\rightarrow bpm2} \\ \Delta R_{1,2}^{cor1\rightarrow bpm3} \\ \Delta R_{1,2}^{cor2\rightarrow bpm1} \\ \Delta R_{1,2}^{cor2\rightarrow bpm2} \\ \Delta R_{1,2}^{cor2\rightarrow bpm3} \end{pmatrix} = M \cdot \begin{pmatrix} \Delta B^{quad1} \\ \Delta B^{quad2} \\ \Delta B^{quad3} \end{pmatrix}$$

Lattice Verification – Analysis (1st attempt)



Simulated results (preliminary)



Lattice Verification

- Current functionality
  - Tweak correctors or movers to measure orbit distortion
    - Close bumps if necessary
  - Fit to fractional quad PS error
- Still to do
  - Extra corrector/mover steps to investigate higher-order fields
    - Sextupoles, multipole distortions
  - Additional fit parameters
    - Quad tilts, bends, sexts, etc.

Summary (1/3)

- Several tools developed
  - Included in base installation of Flight Simulator
  - Tested in simulation
    - Beam tests tomorrow!

1 BPM display

- Absolute/difference, single-shot/averaging, etc.

2 Twiss display

- Predicted, comparison with measurement

3 Bump calculation

- Linear, non-coupled, bump generation.
- GUI or command-line operation

SLAC

Summary (2/3)

4 Beam-based alignment

- With corrector bumps
  - Bump around quad, shunt quad, then bump again
  - Centre is crossing point of these lines
- With movers
  - For quads, same method as above, but with quad moves instead of bumps
  - For sexts, scan with mover, and fit to parabola. (Shunting is unnecessary.)

5 Lattice verification

- Measure orbit distortion due to corrector tweaks or quad moves
- Fit to quad strengths to determine any errors
- Additional fit parameters, constraints, can be added

Summary (3/3)

- All tools made available to FS as soon as possible
  - Even if functionality is limited
    - I try to avoid uploading broken code!
  - I hope others can make use of these tools
- Interested in hearing ideas:
  - Other tools?
  - Extension of these tools?
  - ...
  - Email me (smolloy@slac.stanford.edu) with ideas!