FETS BPMs – Calibration and FPGA Processing

Gary Boorman

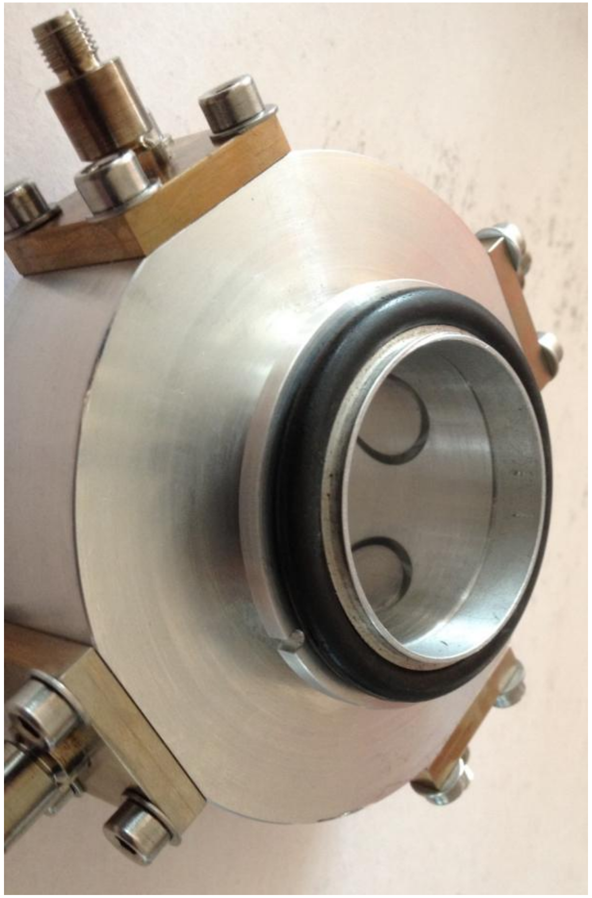
ROYAL
HOLLOWAY
UNIVERSITY
OF LONDON

Contents

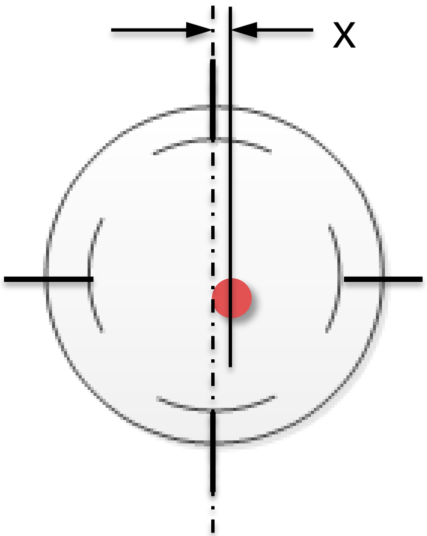
- BPM Calibration
    - Parameters to be determined – sensitivity, offset and linearity
    - Wire rig hardware
    - Results
- BPM Signal Processing
    - FPGA code outline
- Next Steps
    - Full signal chain test
    - Project student to calibrate all BPMs
    - Client viewer and logger
    - Infrastructure

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Sensitivity and Offset



Prototype BPM



$$x = \frac{1}{S_x}\frac{\Delta V}{\Sigma V} + dx$$

x = position (mm)

S_x = sensitivity (mm/V)

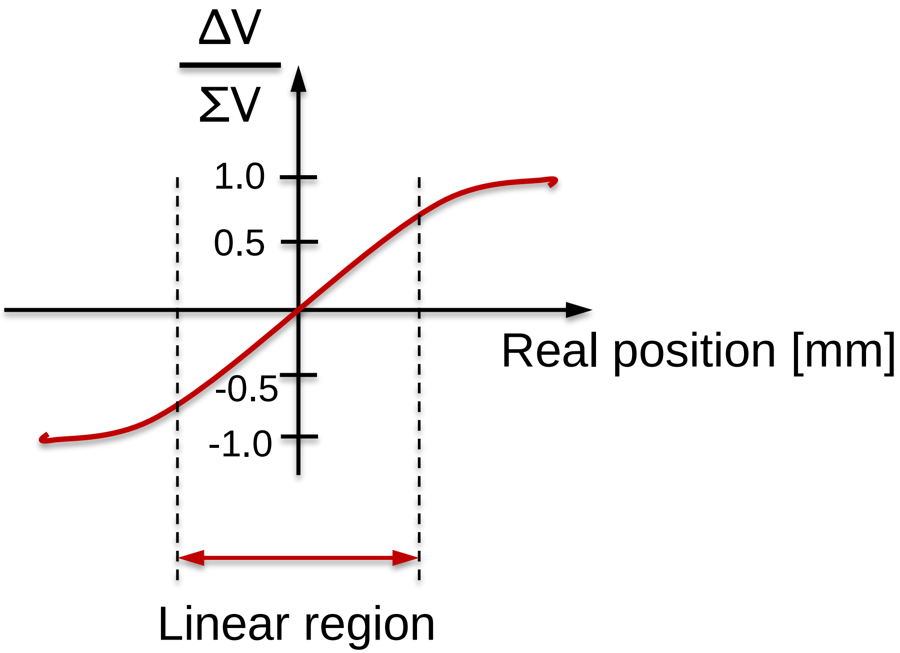
$\Delta V = V_{right} - V_{left}$

$\Sigma V = V_{right} + V_{left}$

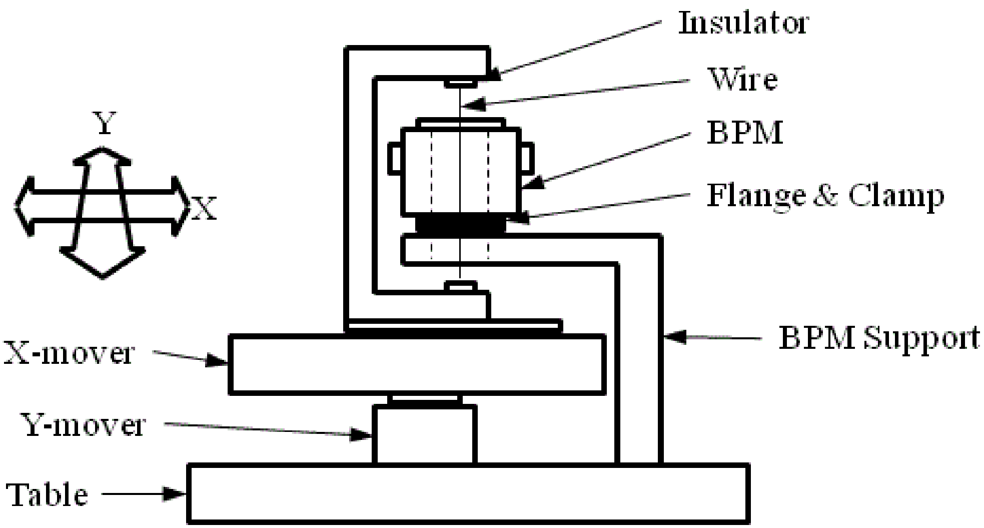
dx = offset

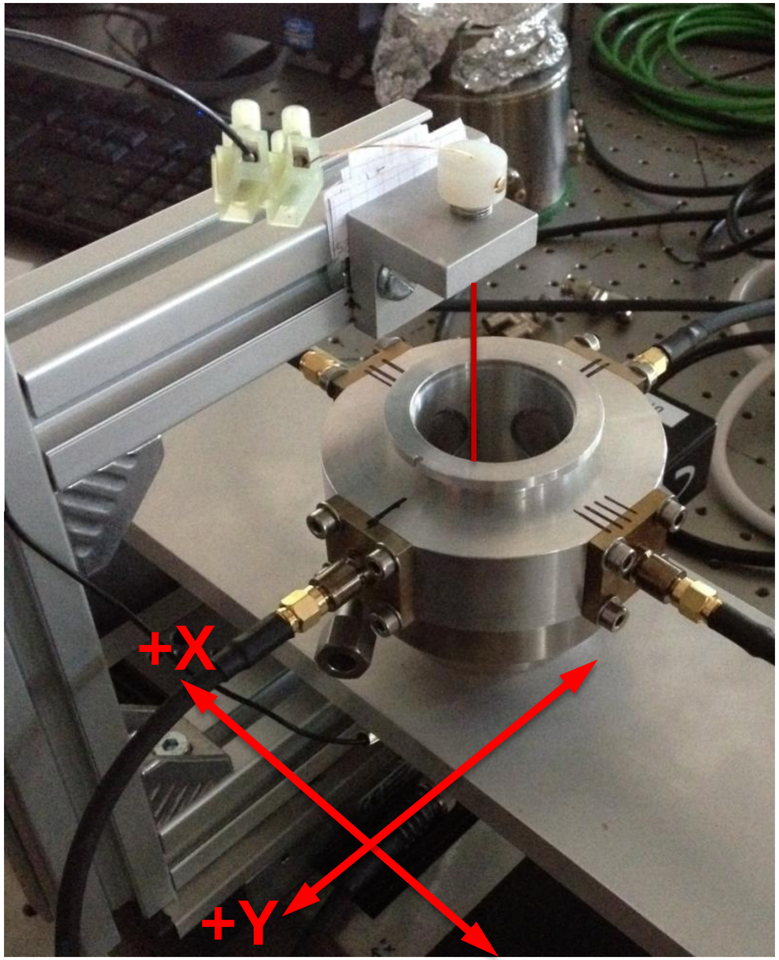
Linearity

- Factors such as electronics dynamic range, saturation etc affect position resolution (not considered here)
- Need to understand linear region of the BPM
  - Intrinsic to beam pick-up



BPM Wire Rig





- Wire has CW RF at 324MHz
- Electrodes read out by 1GHz BW (4GS/s) 4-ch oscilloscope
- VI controls wire movement and scope readout
- Analysis in Python

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Sensitivity Calibration

- Method to determine sensitivity
  - Position wire to move along one axis eg move along y-axis in 1mm steps (keep x = 0)
  - Record electrode signals at each point
  - Determine difference/sum of each point
  - Linear fit to Δ/Σ (range -4 mm to + 4mm each axis)
- Results (prototype BPM):

$$S_x = 10.686 \pm 0.180$$

$$S_y = 11.354 \pm 0.240$$

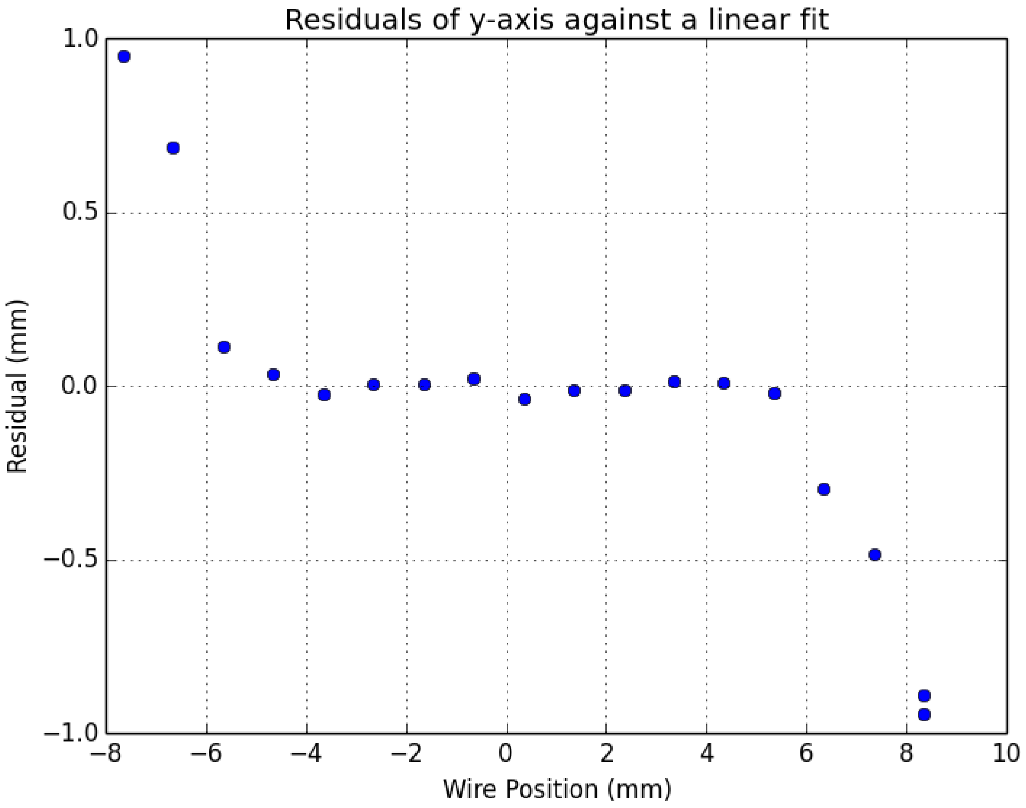
- *PROBLEM*: Difference between Sx and Sy of >6%
- If sensitivity measured along another axis (eg x = 2mm) then difference up to 10%

Sensitivity Axis Disagreement

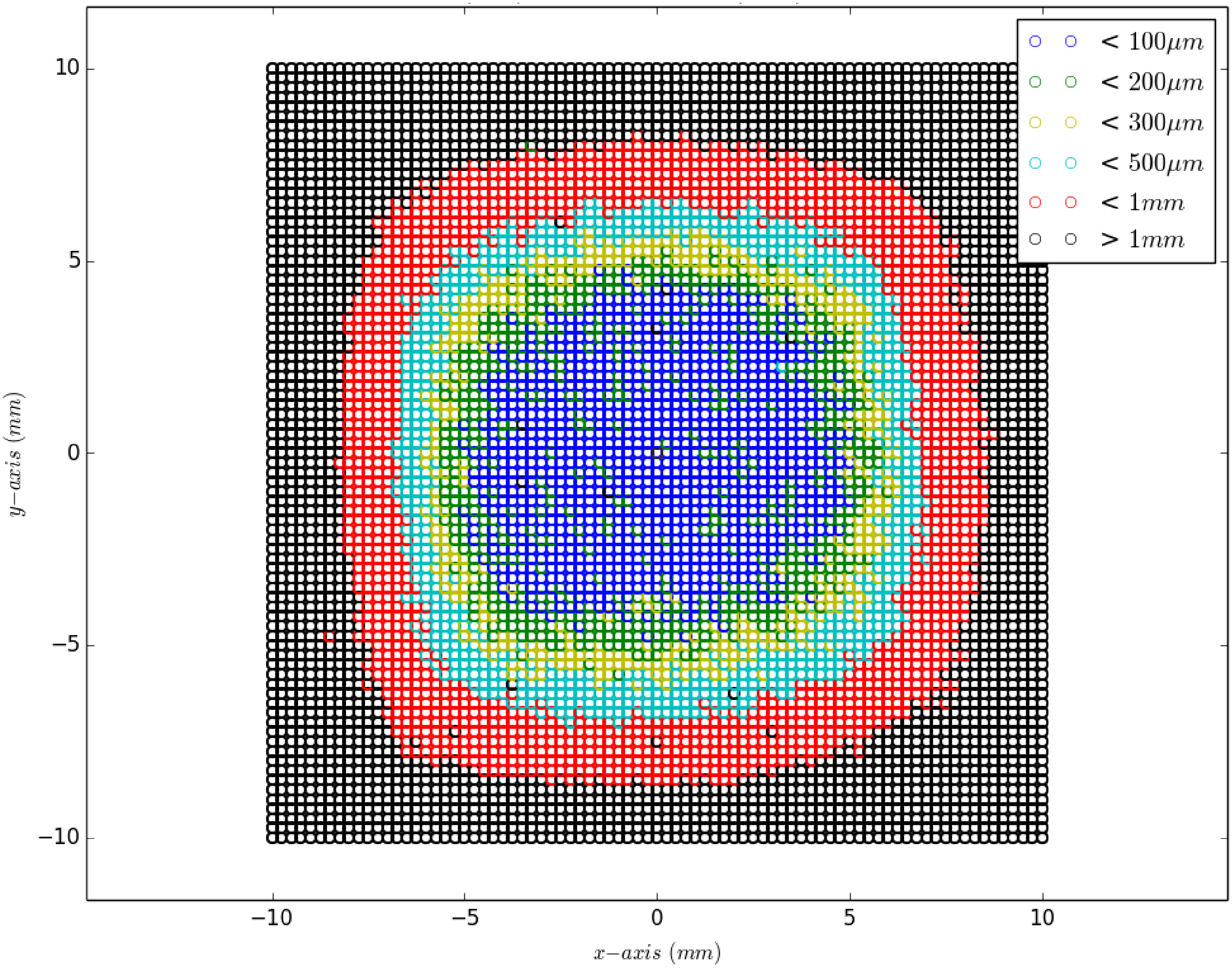
- What causes the disagreement between the sensitivity coefficients for the x and y axes?
  - All cables swapped, scope inputs changed, BPM rotated 90°, electrodes re-seated, wire verticality re-checked…
  - Rotating the BPM by 45 degrees could make the difference decrease - difference now about 3%!
- Concluded that the aluminium frame holding the wire is causing a disturbance in the RF field seen by the BPM electrodes
- A new perspex frame to hold the wire is being fabricated and will be tested this week.

Linearity

- Linear region extends to almost 5 mm from the BPM centre, but more work required with new wire frame



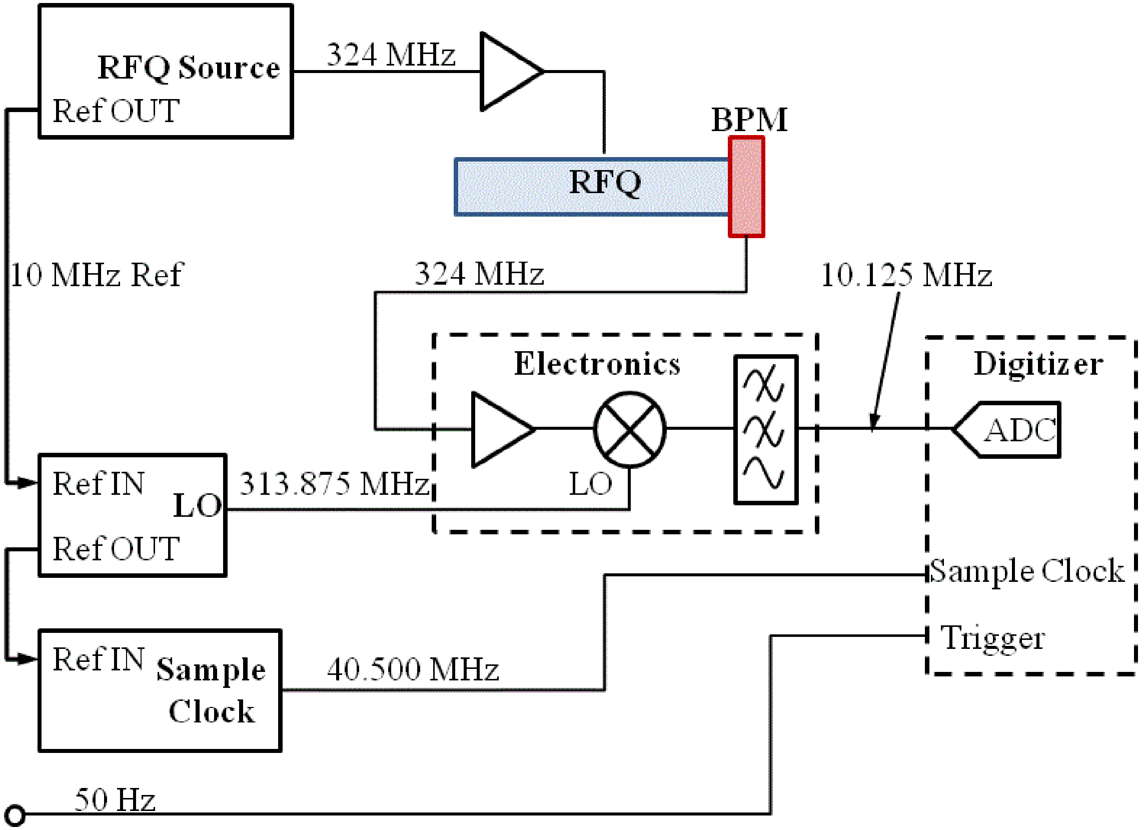
Position Resolution



ROYAL HOLLOWAY UNIVERSITY OF LONDON

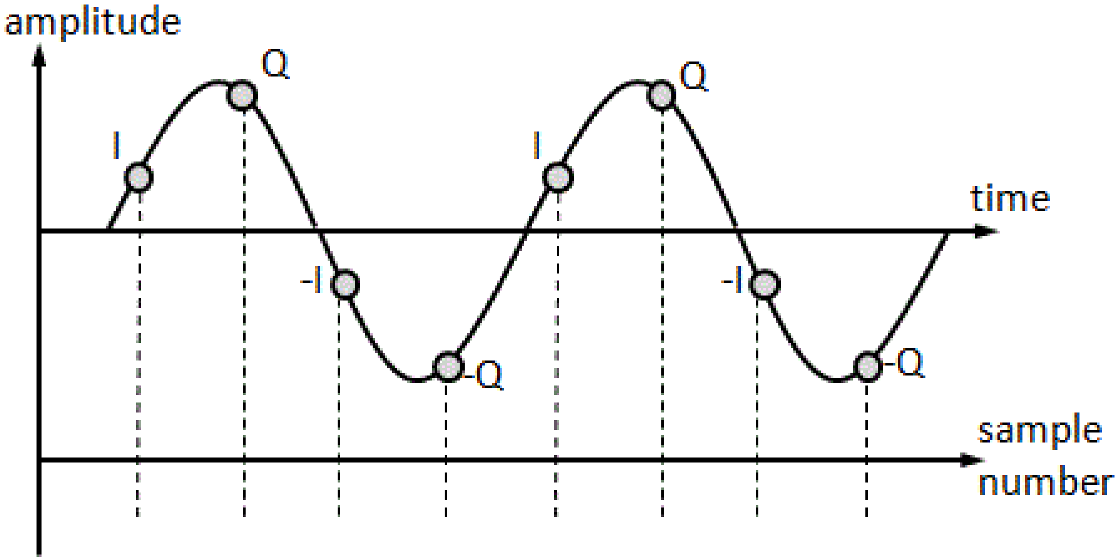
Signal Acquisition and Synchronisation

- Uses NI PXI-7954R and NI-5752
- 8 BPMs, 4 electrodes each -> 32 channels required



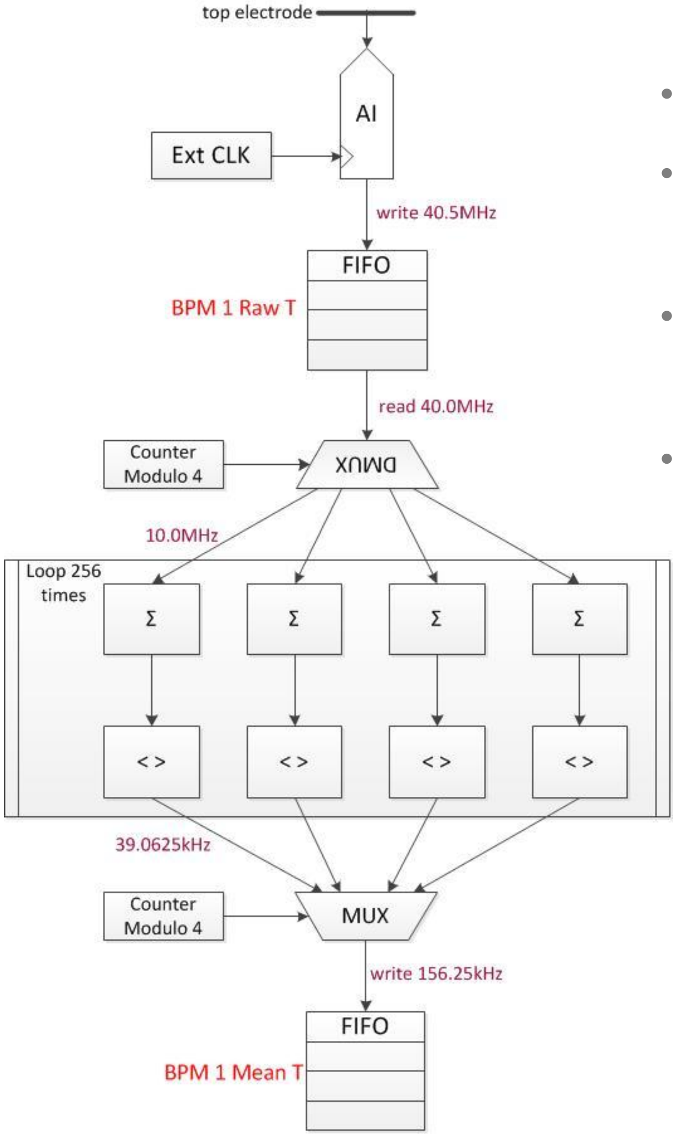
I/Q Sampling

- Each electrode sampled simultaneously after trigger
- The IF signal of 10.125 MHz is sampled at exactly four times this frequency – 40.500 MHz

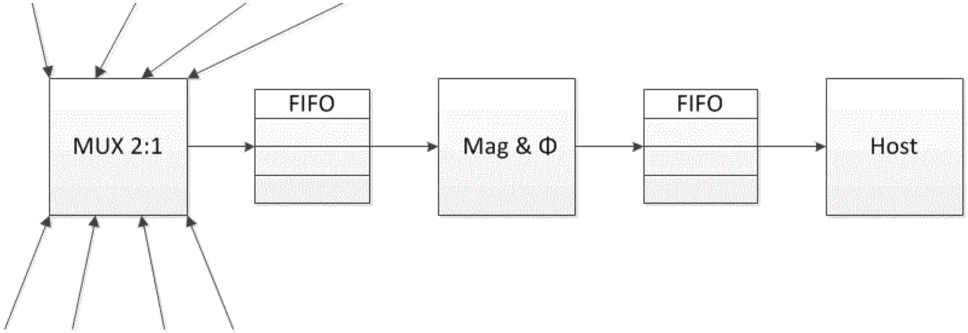


$$A = \sqrt{I^2 + Q^2} \qquad \Phi = \tan^{-1}\left(\frac{Q}{I}\right)$$

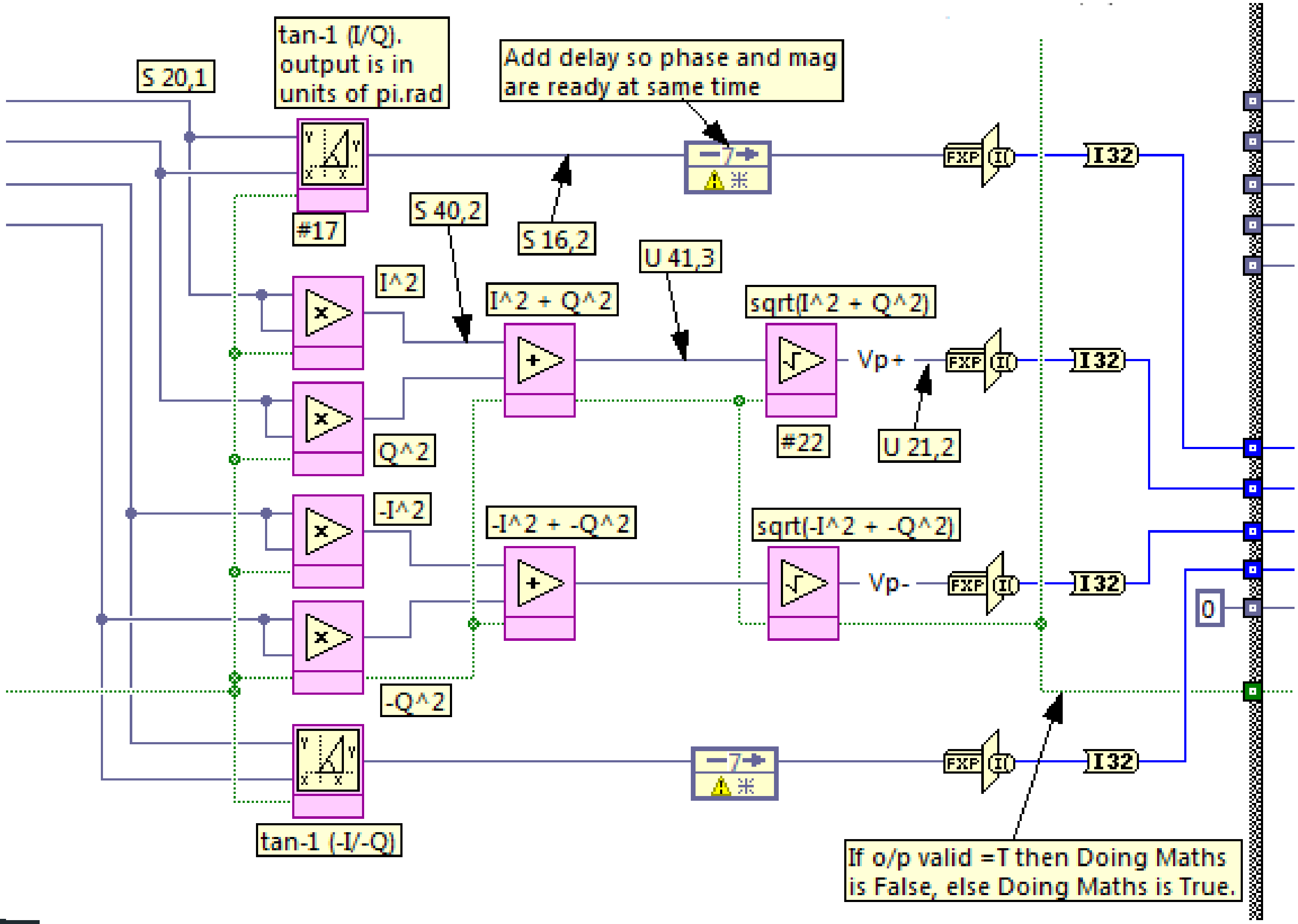
BPM Processing



- *ALL* code written in LabVIEW
- For each electrode the I, Q, -I, -Q are accumulated and then averaged
- Averages are FIFOed and MUXd to maths unit to get magnitude and phase
- Maths unit does delta/sum * sensitivity then DMAs result to RT host



FPGA Code Snippet



tan-1 (I/Q). output is in units of pi.rad

Add delay so phase and mag are ready at same time

If o/p valid =T then Doing Maths is False, else Doing Maths is True.

Processing Time

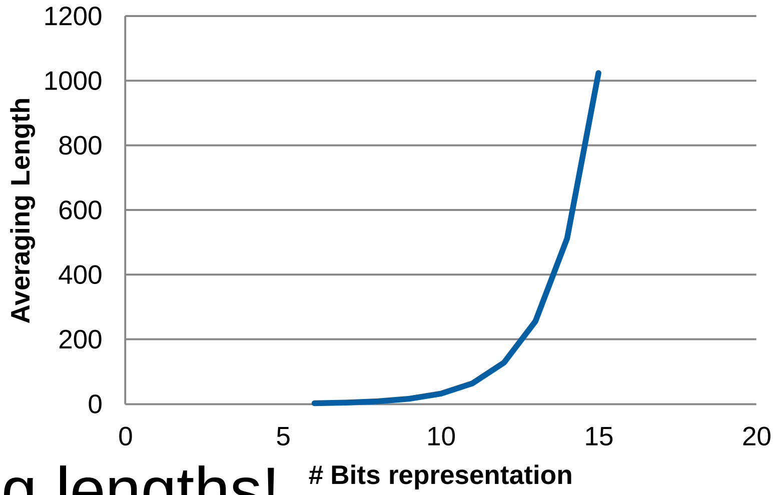
- Currently all processing uses the FPGA 40 MHz clock
- 128 accumulator channels averaged every 25.6 µs
- Uses four maths units to process pairs of BPMs in parallel
- Uses 32 DSP blocks in FPGA for square root, multiplication and division
- Magnitude (sqrt) and phase ($\tan^{-1}$) takes 26 clock cycles
- Difference/sum (div) takes 20 clock cycles
- Processing eight BPMs takes about 10.4 µs

How can processing time be reduced….?

(Update – total processing time is now 1.3 µs)

Processing Optimisation

- Three optimisation approaches:
  - Increase clock speed (up to 200 MHz clock available)
  - Shorten representation of fixed-point numbers
  - Pipeline maths functions



- Increasing clock speed means more difficult to fit the logic!
- Maths function: #cycles ∝ #bits
  - Also allows shorter averaging lengths!
- Pipelining allows start of next calculation before previous one has finished

Distributing the BPM Position

- The BPM number and vertical/horizontal axis encoded into a 32 bit number that is DMA’d to RT host
- RT host has EPICS server to allow client access
    - EPICS client for data-logging
    - EPICS client for control room viewing
- EPICS allows platform/OS/language agnosticism

- Testing and verifying processing time was, and remains, a challenge.

Conclusion

- BPM Wire Rig works reasonably well, but a non-metallic frame will improve calibration
- FPGA code is working, and total processing time is 1.3µs
- Production BPMs delivered to RAL
  - Calibration of all BPMs will occur this autumn with project student
  - Full signal chain test happening as soon as filter components on mixer electronics changed
- BPM Client viewer – requirements….