Jitter correlations and resolution measurement from May '08 data taking run

Glenn Christian

30 June 2008

Goals

- Measure correlation in position between bunches
- Determine how well feedback system corrects the jitter by comparing the jitter of each bunch with the system on to that predicted from the jitters with the system off.
  - Get an estimate of the system resolution based on jitter measurements.

- Most of the analysis done with un-calibrated data as initially more interested in consistent results than absolute scale of position jitter.

Jitter Correlations

Data summary

3.5 sets of 'jitter' data:

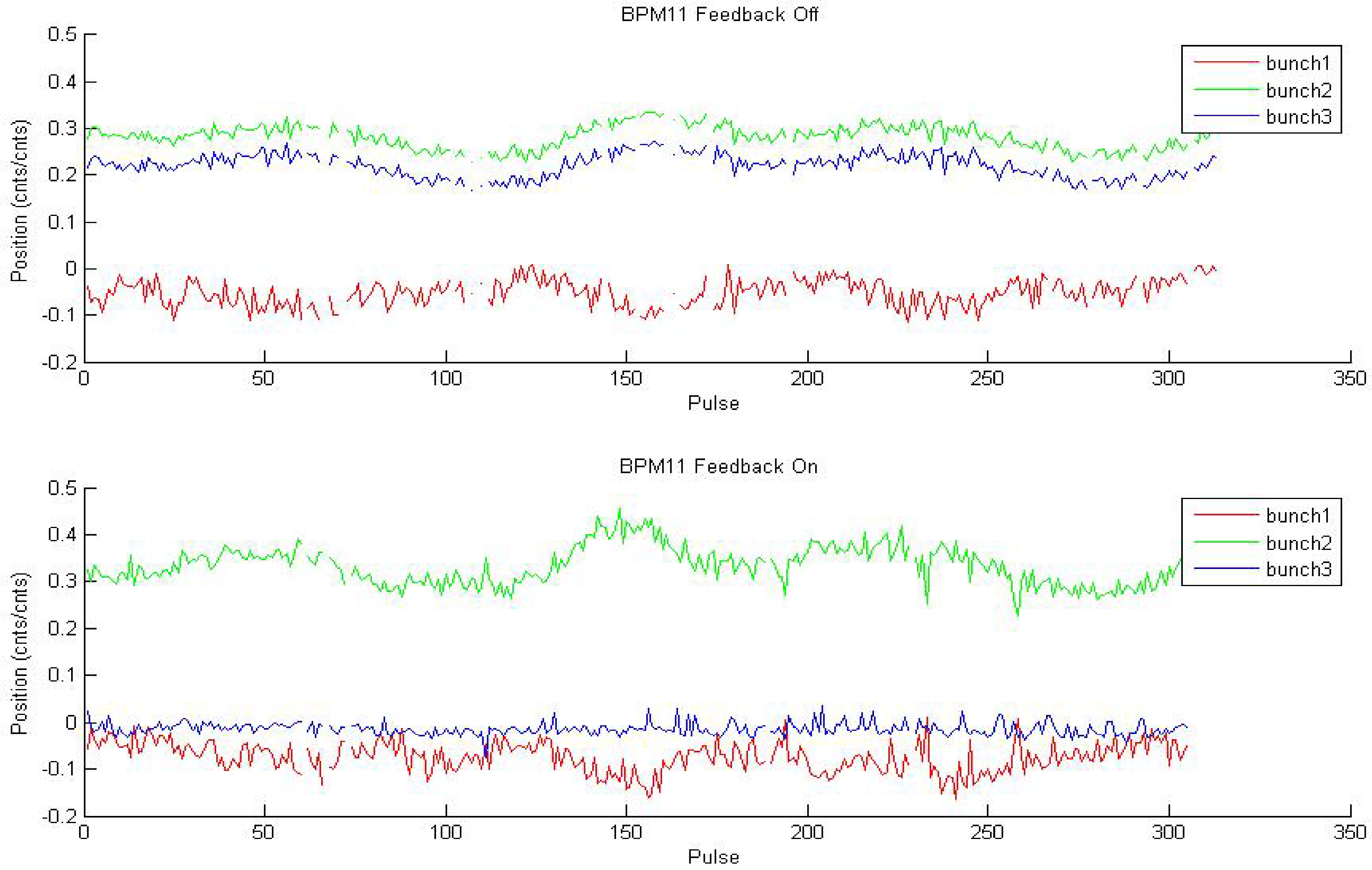
- Run1: 154 ns bunch spacing; feedback on/off; near zero mean position
- Run2: 154 ns bunch spacing; feedback off only; different position setting
- Run3: 148.4 ns bunch spacing; feedback on/off.
- Run4: same as Run3.

Following slides use Feedback off data only, with flyers cut at 3 sigma level, unless stated otherwise.

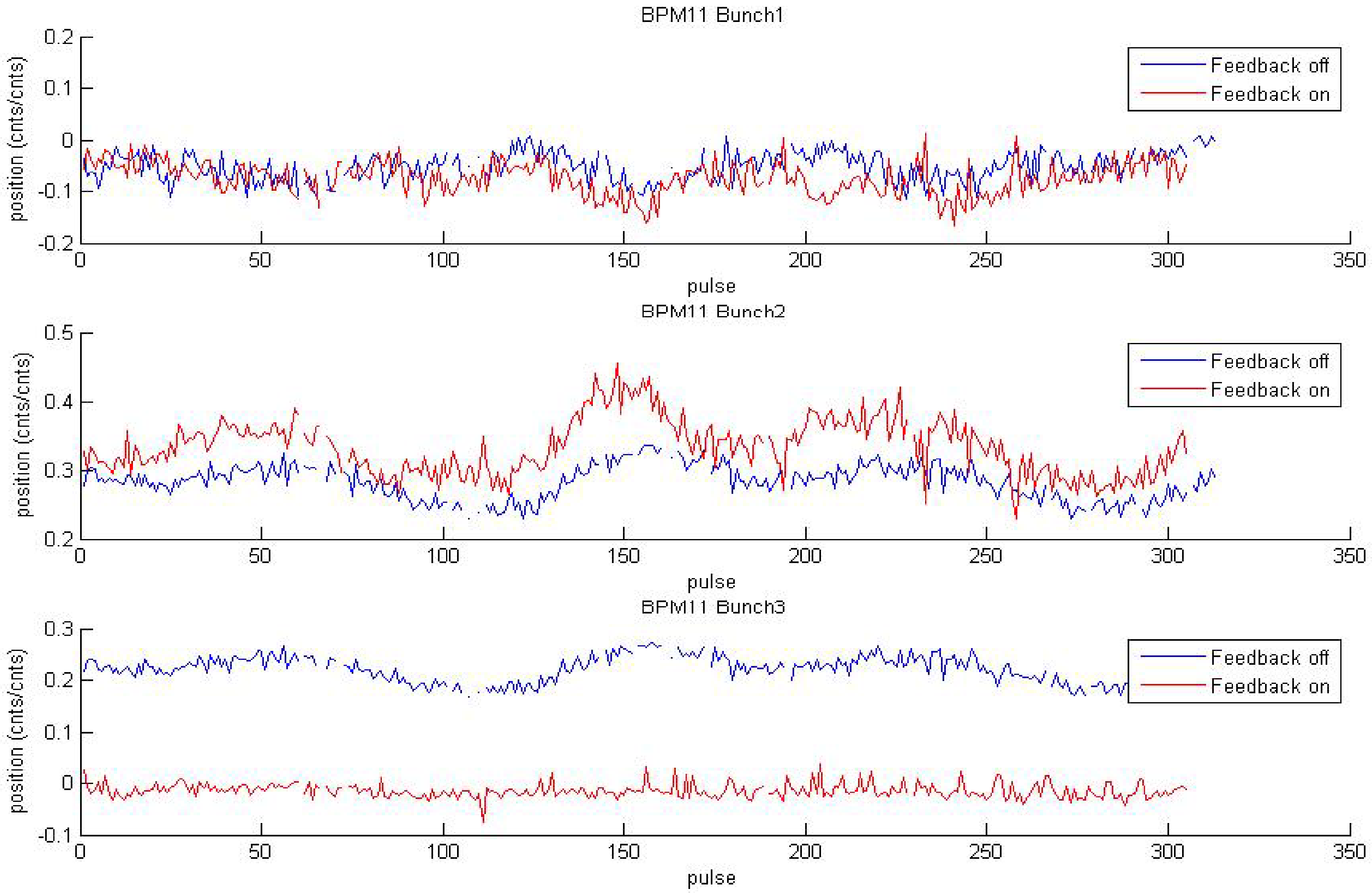
After cuts:

- Run1: off 292(313) / on 300(305) – cut @ 2sigma;
- Run2: off 98(107) - cut@1.5sigma;
- Run3: off 91(94) / on 60 (60);
- Run4: off 50 (50) / 76 (76);

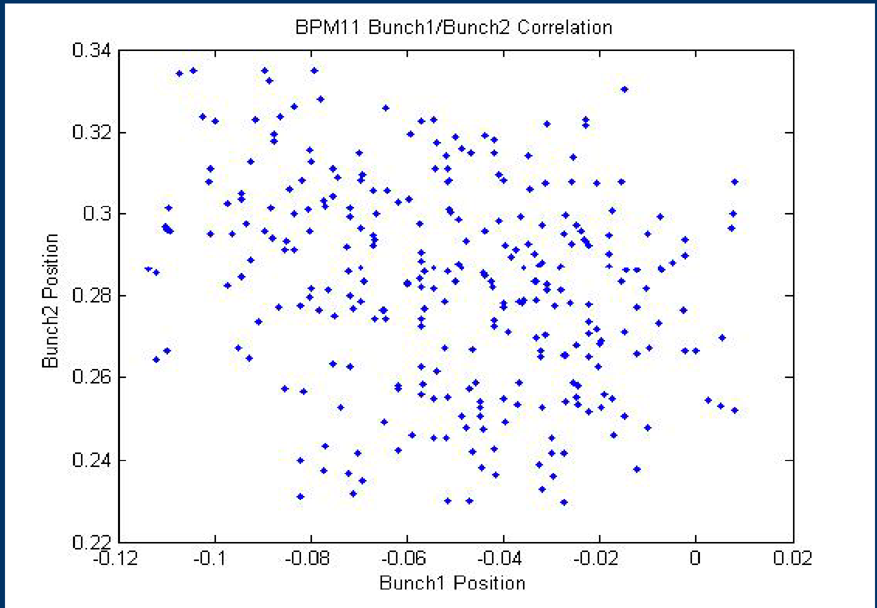
Run1 : bunch 1,2,3 comparison

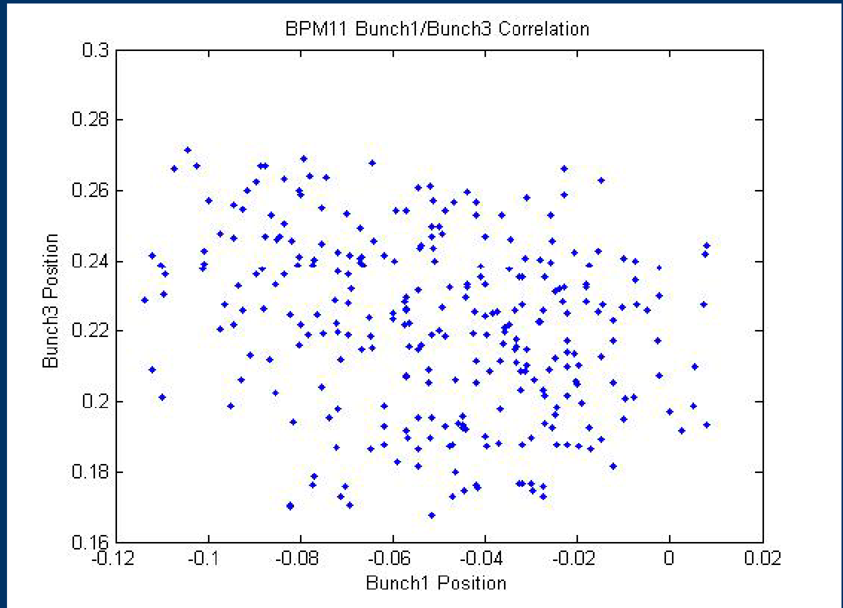


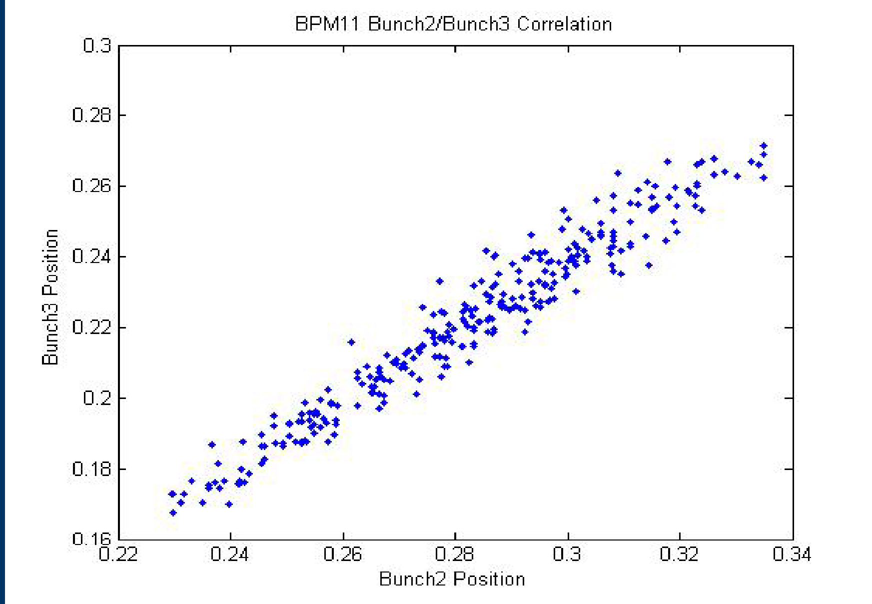
Run1: feedback off/on comparison



Run1: Correlations

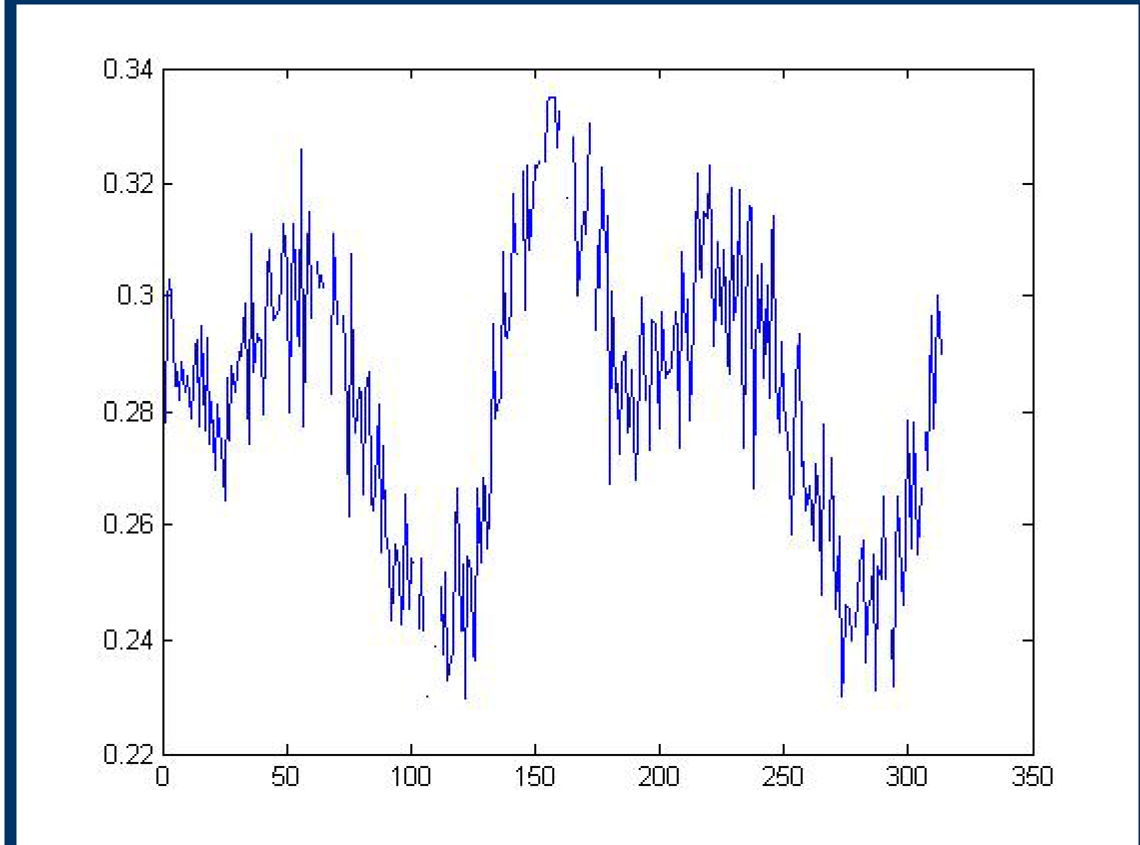






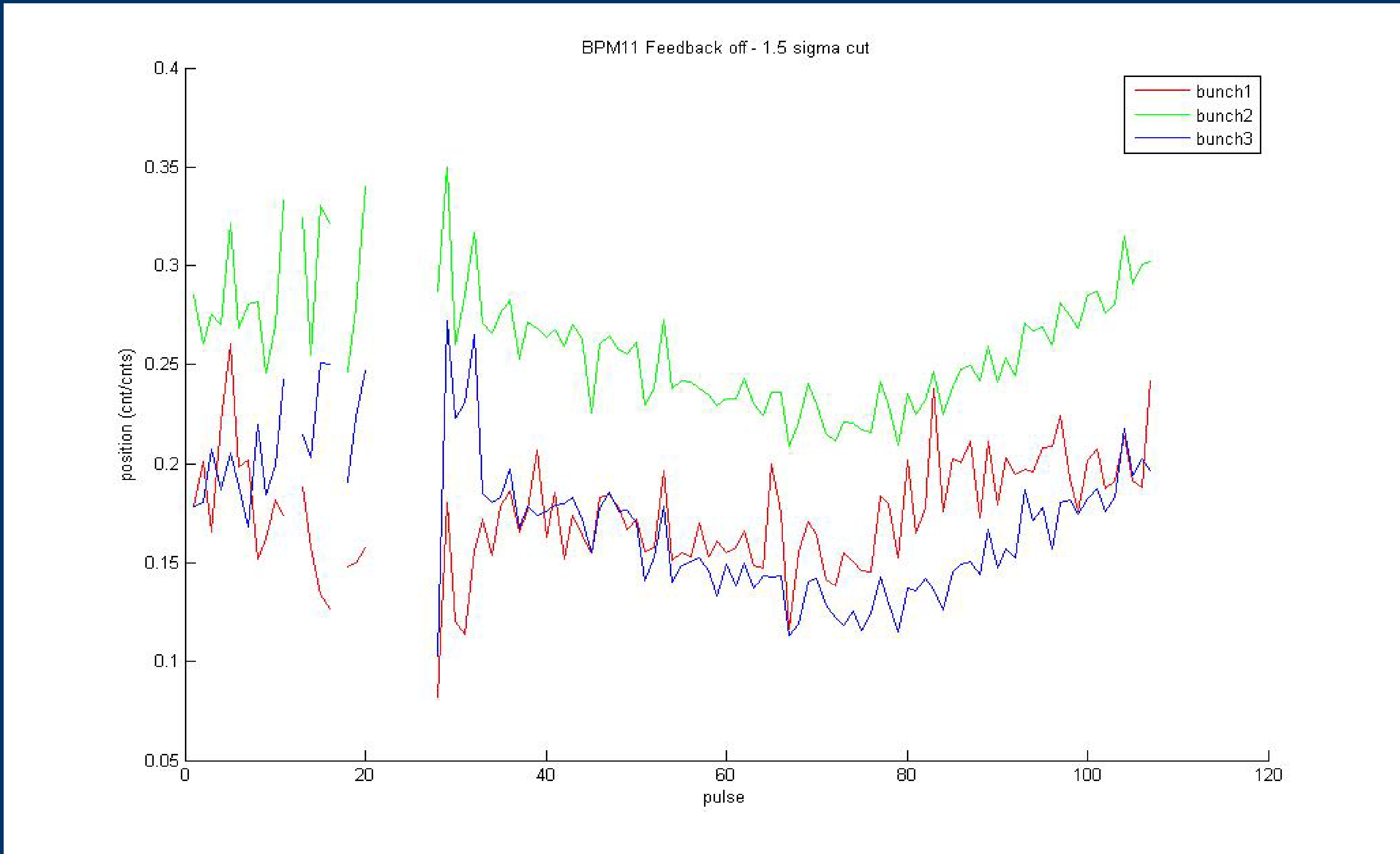
- R(1,2) = -0.2451; P(1,2) = 0.0000
- R(1,3) = -0.2554; P(1,3) = 0.0000
- R(2,3) = 0.9734; P(2,3) = 0.0000

Run1: periodic variation

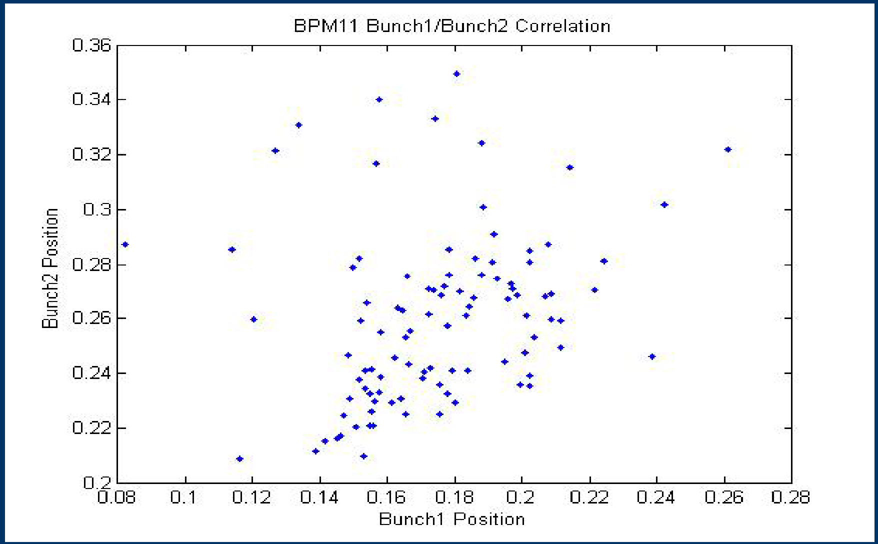


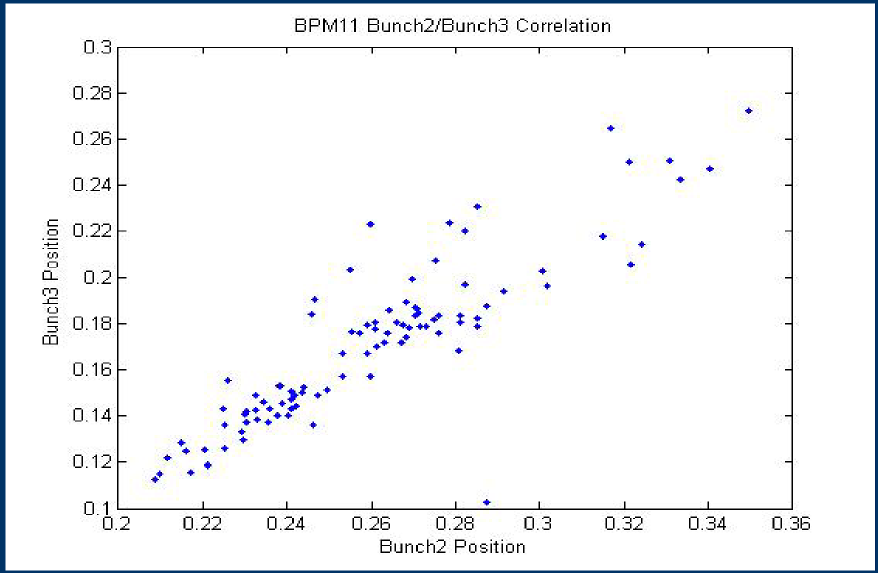
- Period (samples 280-120) = ~ 5 mins
- Bunch2 feedback off data shown
- Initially no attempt made to remove this shape (see later)
- Although other data shorter in duration – shape 'looks' to be consistent across all sets

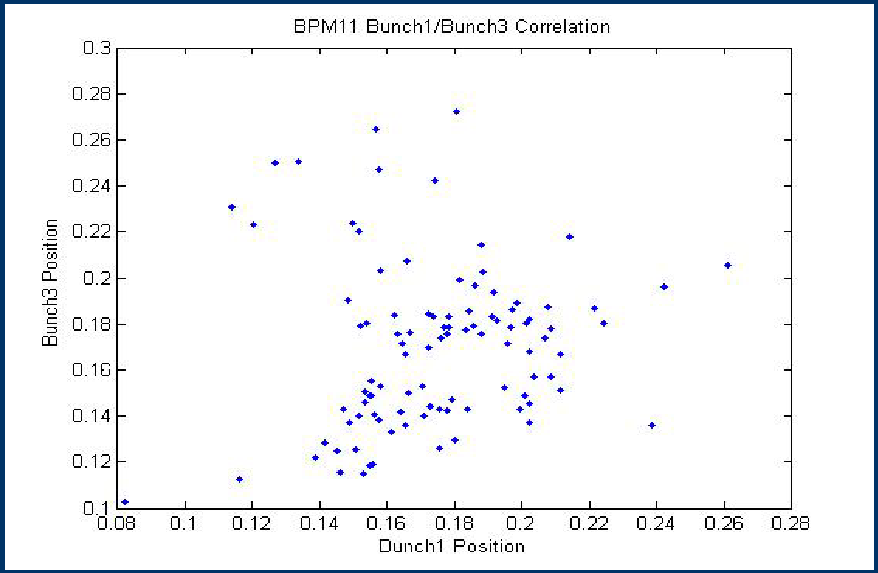
Run2: bunch 1,2,3 comparison



Run2: correlations

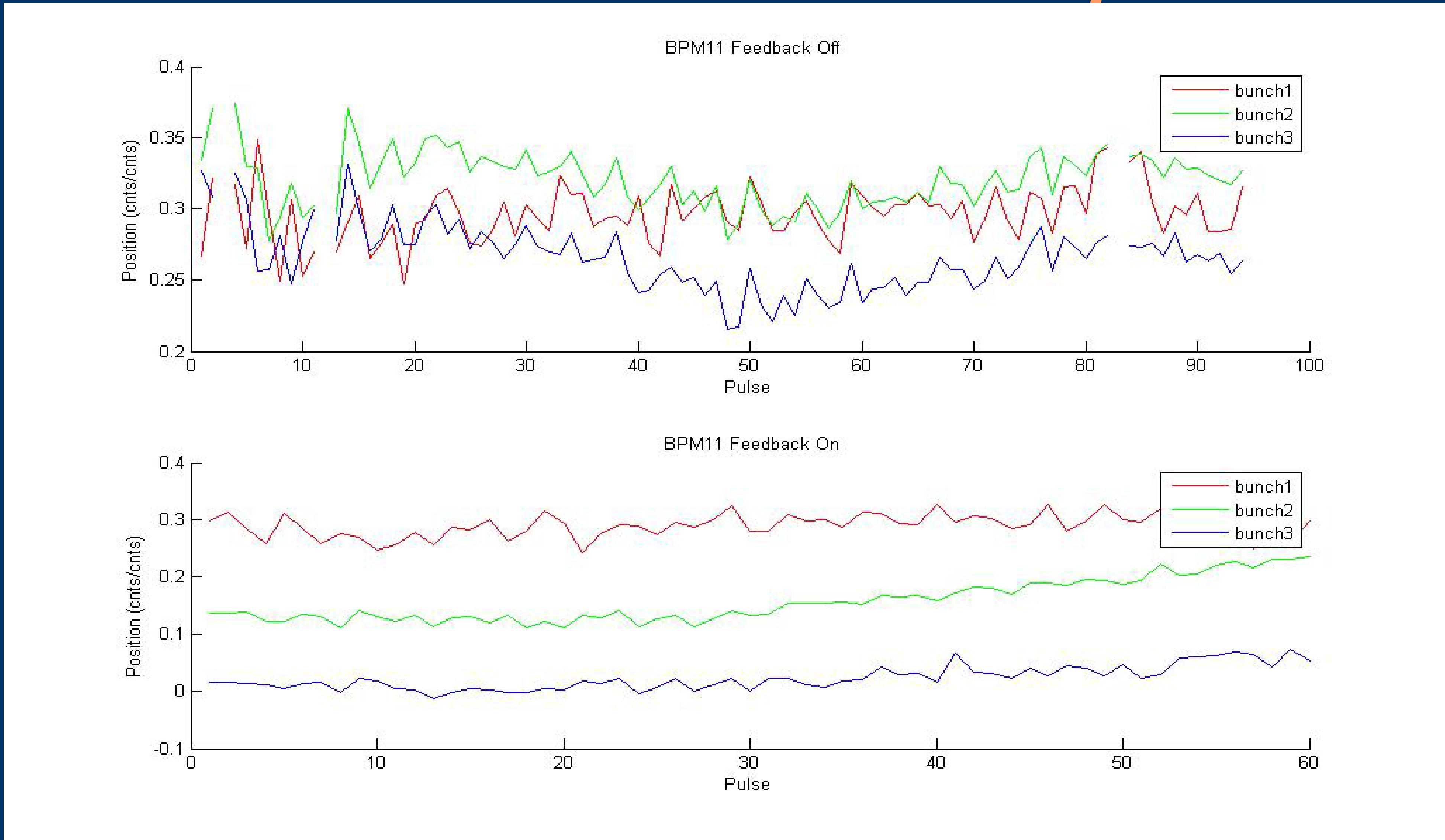




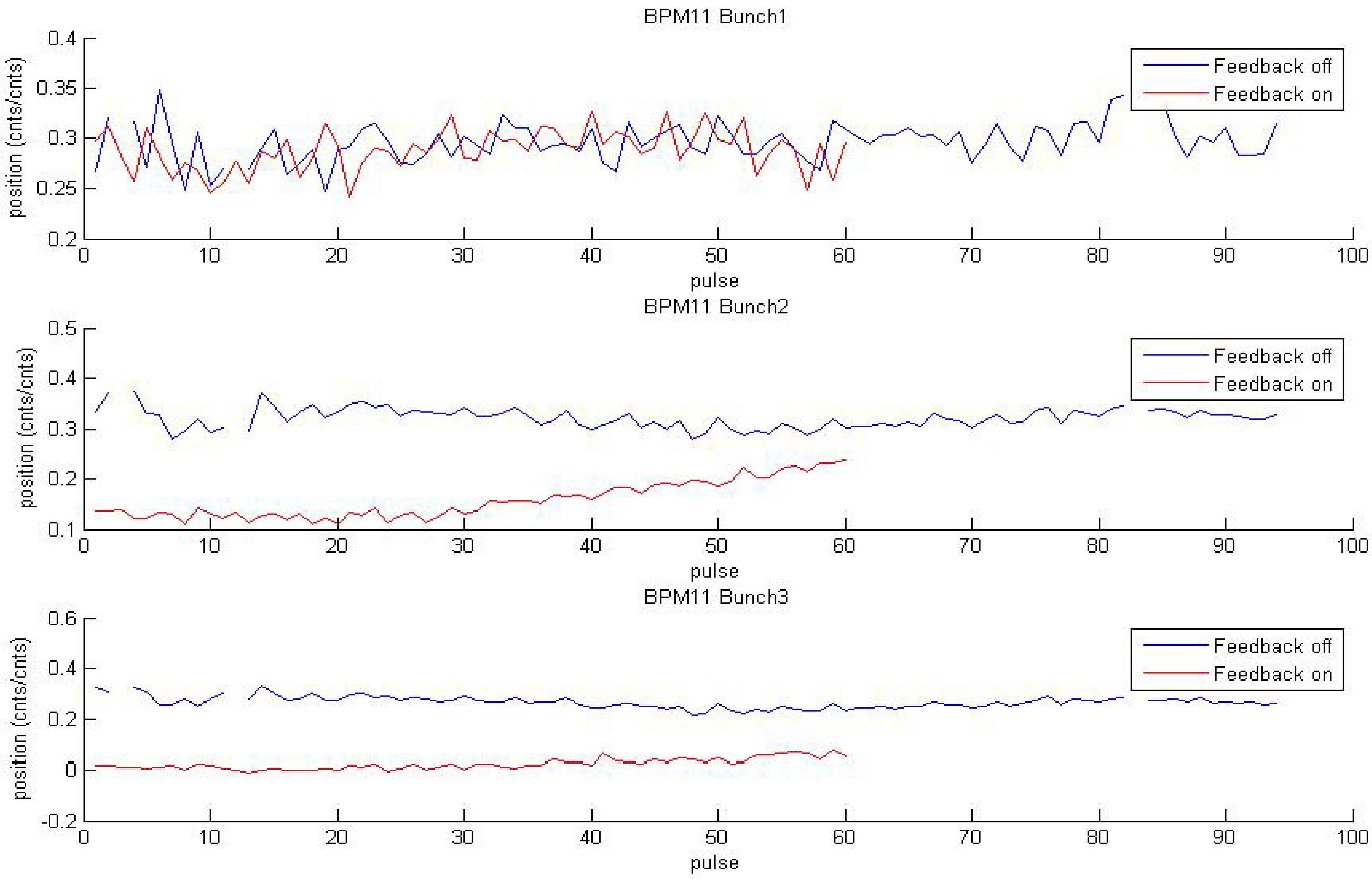


- R(1,2) = 0.2559;
  P(1,2) = 0.0110
- R(1,3) = 0.1226;
  P(1,3) = 0.2290
- R(2,3) = 0.8797;
  P(2,3) = 0.0000

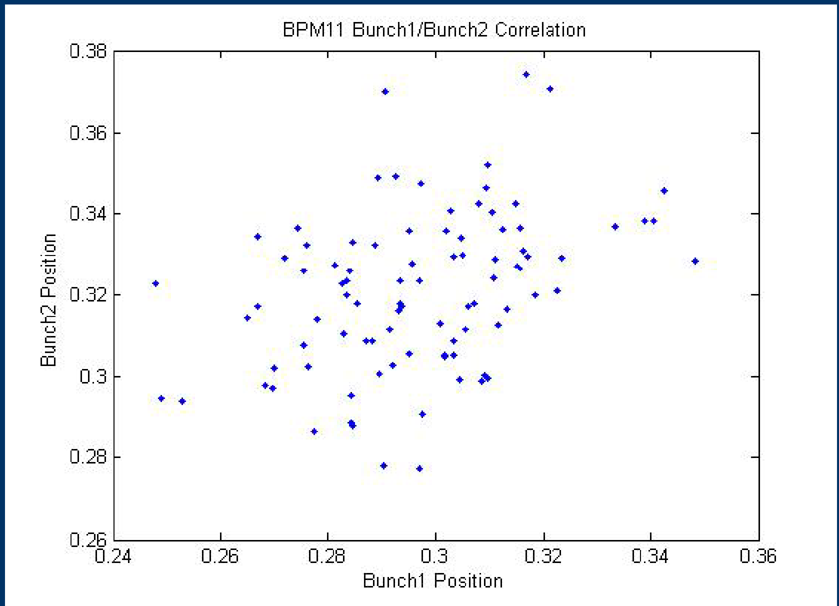
Run3: feedback off/on comparison

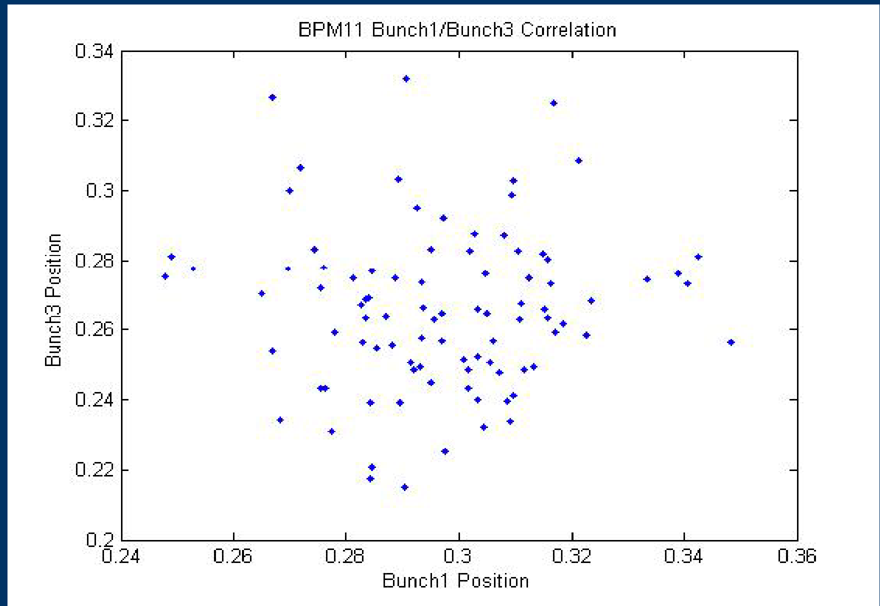


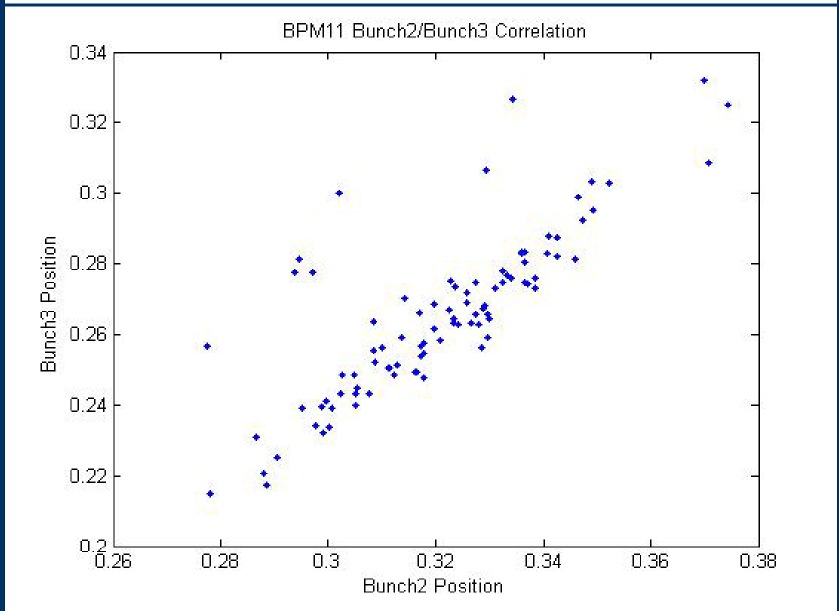
Run3: bunch 1,2,3 comparison



Run3: Correlations

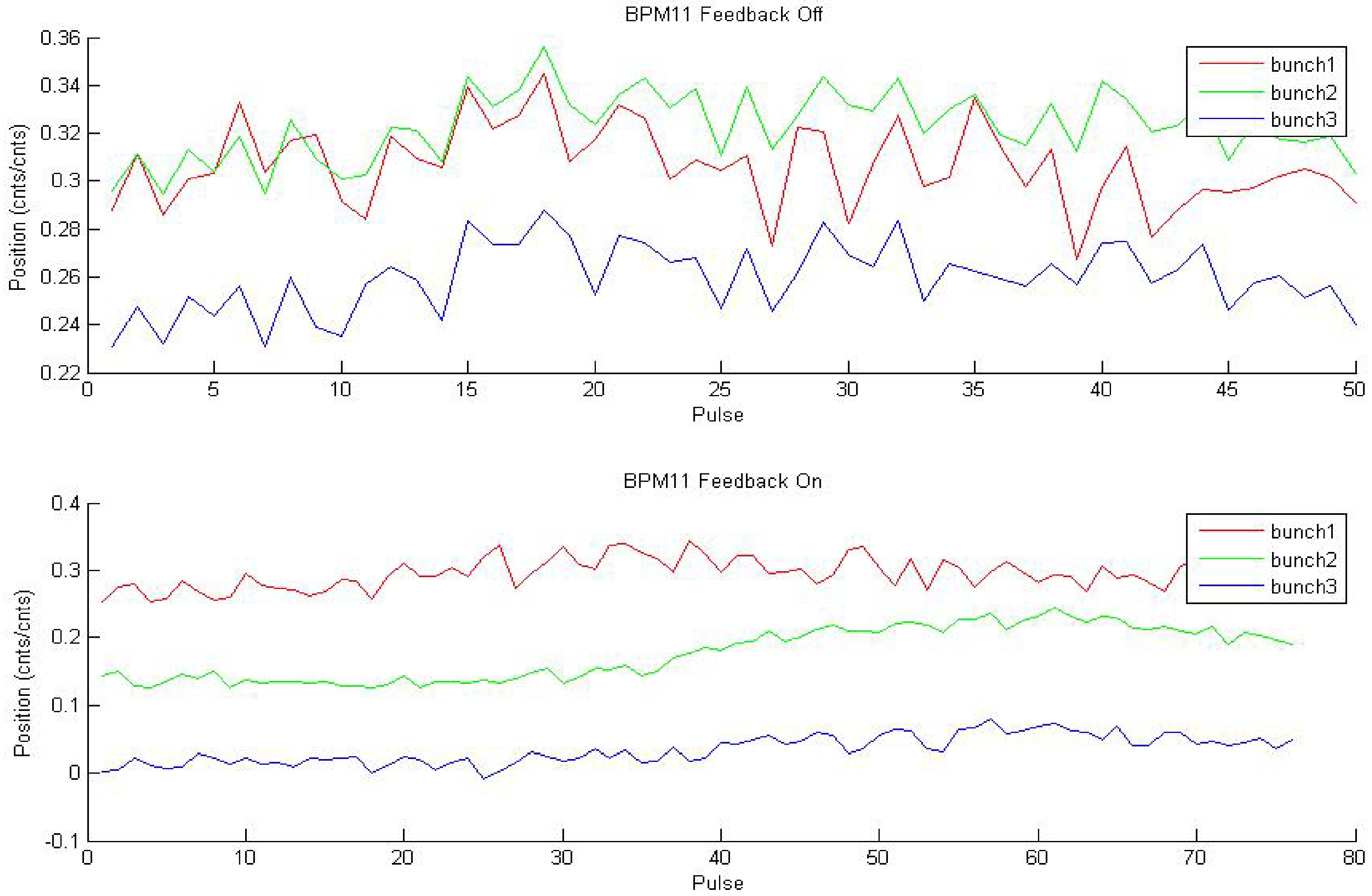




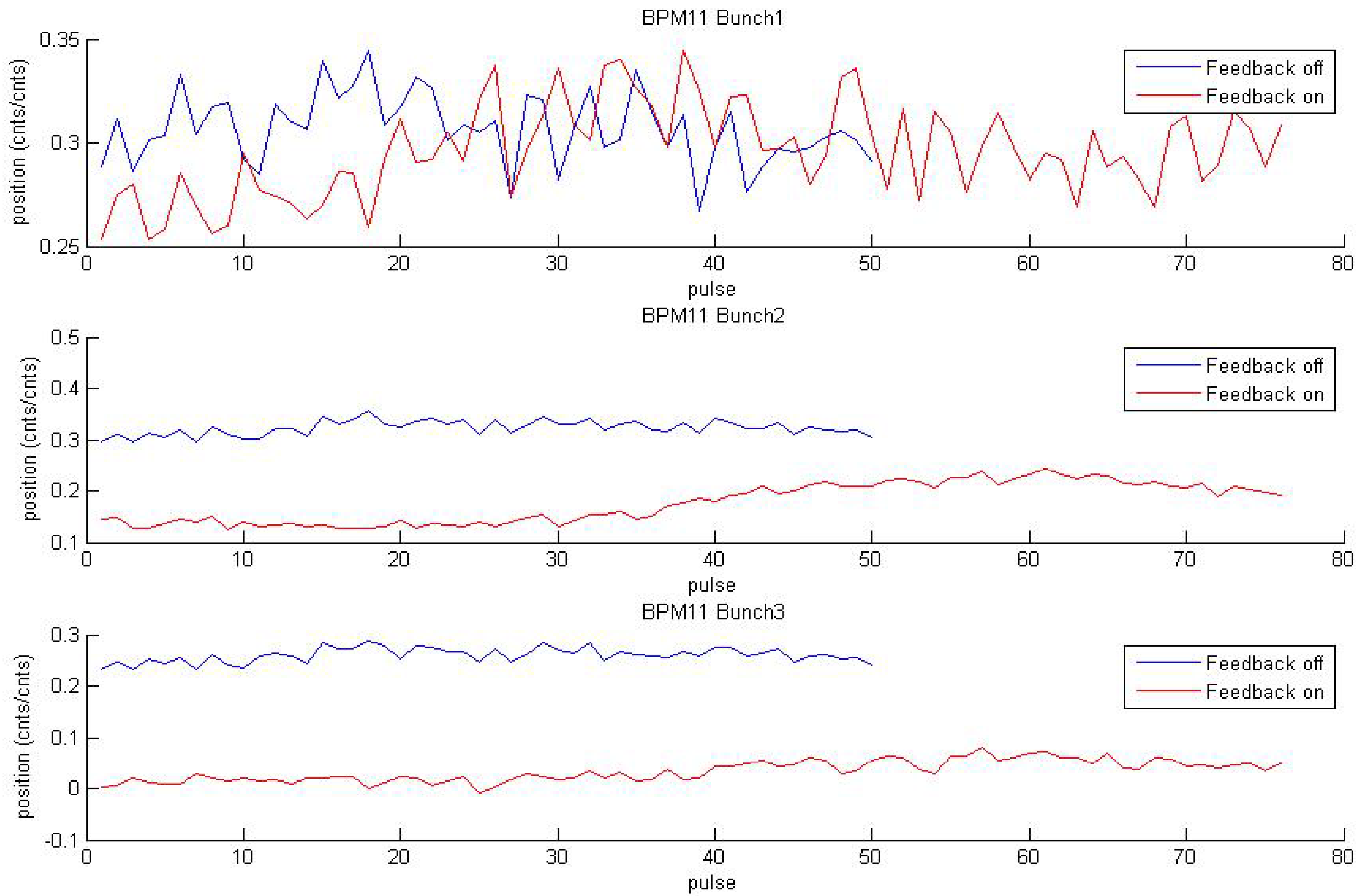


- R(1,2) = 0.3851; P(1,2) = 0.0002
- R(1,3) = 0.0124; P(1,3) = 0.9074
- R(2,3) = 0.8121; P(2,3) = 0.0000

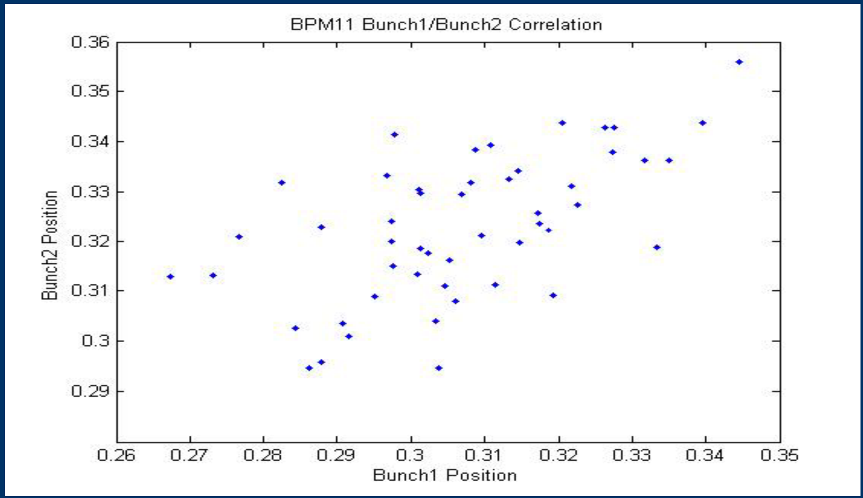
Run4: bunch 1,2,3 comparison

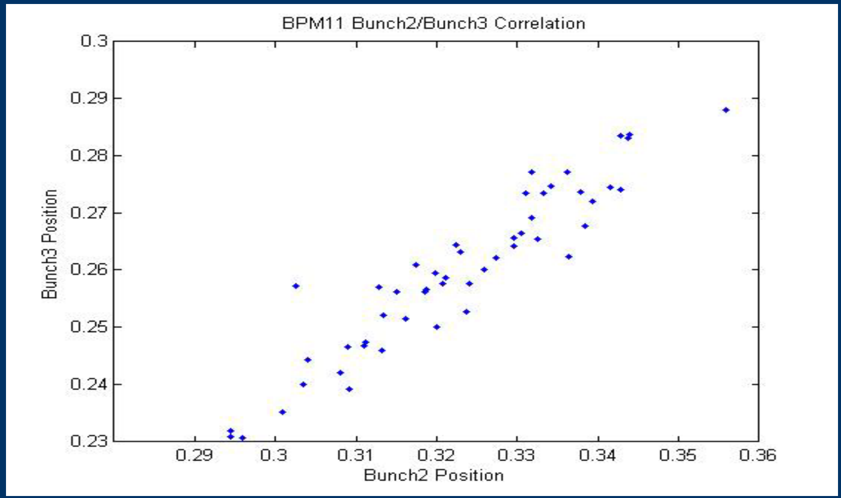


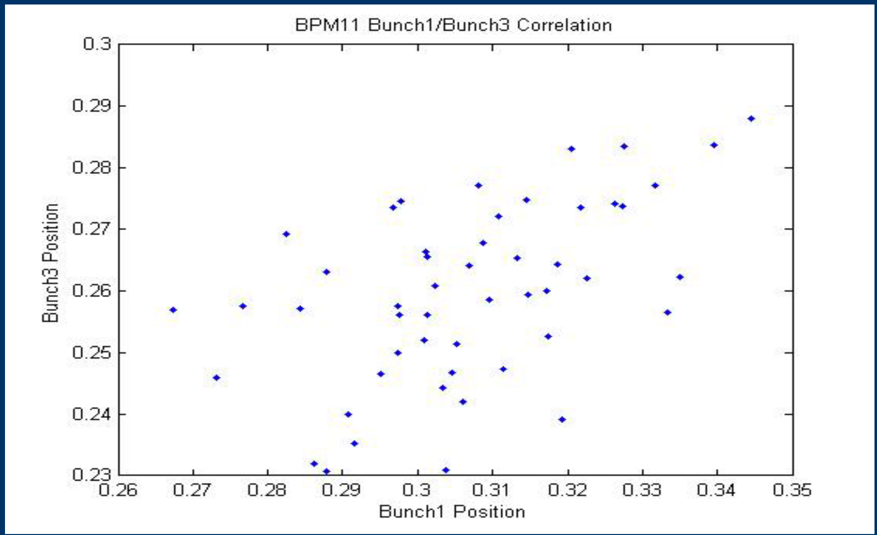
Run4: feedback off/on comparison



Run4: Correlations







- R(1,2) = 0.5879;
  P(1,2) = 0.0000
- R(1,3) = 0.5157;
  P(1,3) = 0.0001
- R(2,3) = 0.9430;
  P(2,3) = 0.0000

	Pulses	bunch	mean	std	corr	R	P
Run1	292	1	-0.0506	0.0284	(1,2)	-0.2451	0.0000
		2	0.2826	0.0248	(1,3)	-0.2554	0.0000
		3	0.2214	0.0248	(2,3)	0.9734	0.0000
Run2	98	1	0.1744	0.0283	(1,2)	0.2559	0.0110
		2	0.2599	0.0310	(1,3)	0.1226	0.2290
		3	0.1701	0.0355	(2,3)	0.8797	0.0000
Run3	91	1	0.2968	0.0198	(1,2)	0.3851	0.0002
		2	0.3209	0.0196	(1,3)	0.0124	0.9074
		3	0.2656	0.0231	(2,3)	0.8121	0.0000
Run4	50	1	0.3068	0.0171	(1,2)	0.5879	0.0000
		2	0.3228	0.0145	(1,3)	0.5157	0.0001
		3	0.2596	0.0144	(2,3)	0.9430	0.0000

Jitter correlations - conclusions

- Bunches 2 and 3 always have a good correlation
- Where correlations between bunches 1 and 2 (and hence 1 and 3)exists, it is statically significant but is always too low for feedback and changes randomly!
- Jitter of bunches 1, 2, 3 is roughly the same magnitude and so it is not likely that a real correlation between 1 and 2 exists, hidden by a source of random jitter affecting bunch 1 only.
- If goal is to ‘stabilise bunch3’ then only require a correlation between bunches 2 and 3 – current beam should be good enough!

Jitter measurements and predictions

Jitter Predictions

$\sigma_n'^2 = \sigma_{n-1}^2 + \sigma_n^2 - 2\mathrm{cov}(n-1,n)$

Where σ_n' is the jitter of bunch n with the feedback on.

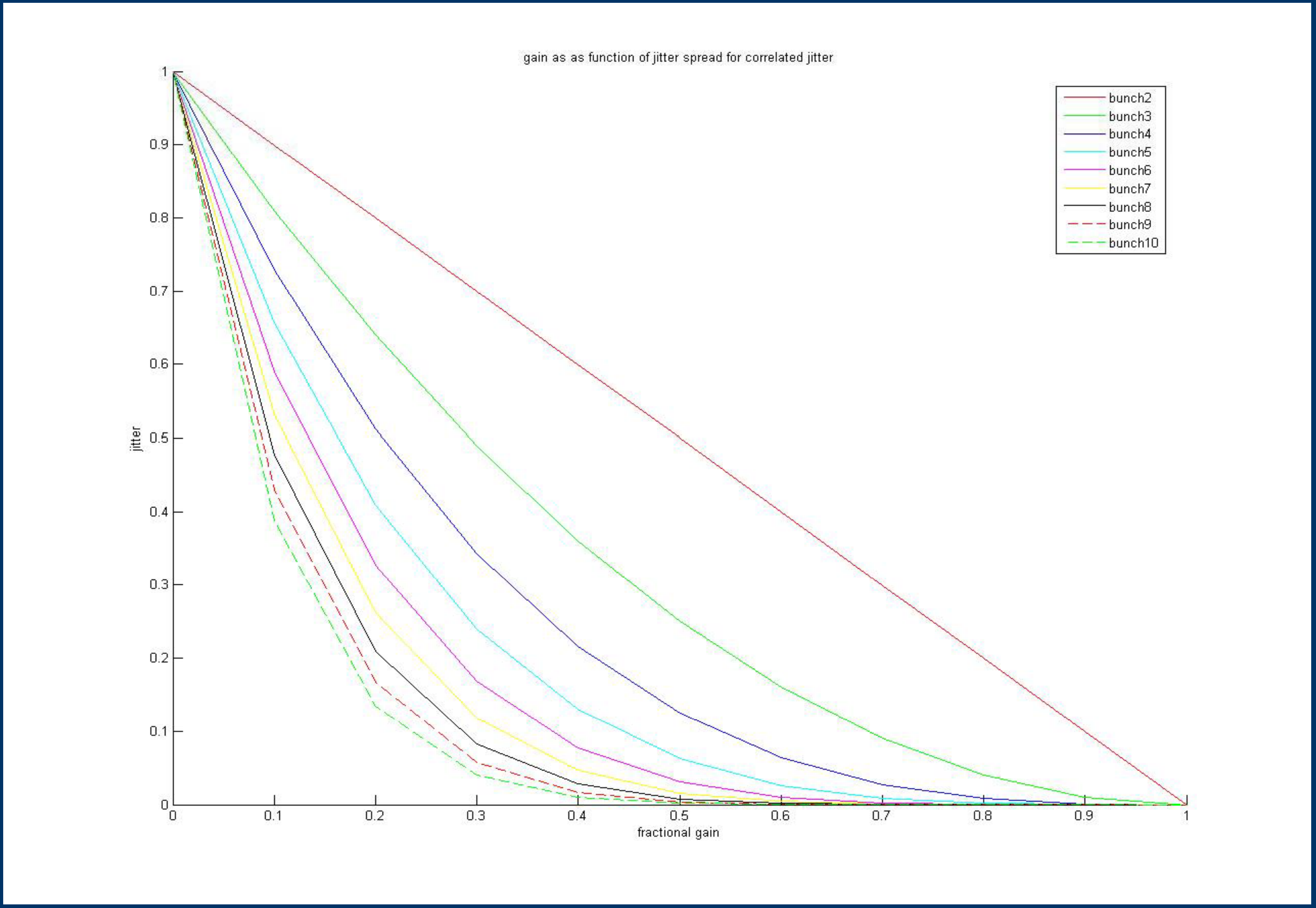
Also have simple simulation (originally used to verify calculation).

```
bunch1' = bunch1;
delay = 0;
bunch2'= bunch2 - (bunch1'*gain + delay); [= bunch2 - bunch1]
delay = delay + bunch1'*gain;
bunch3'= bunch3 - (bunch2'*gain + delay) [= bunch3 - bunch2]
delay = delay + bunch2'*gain;
…
```

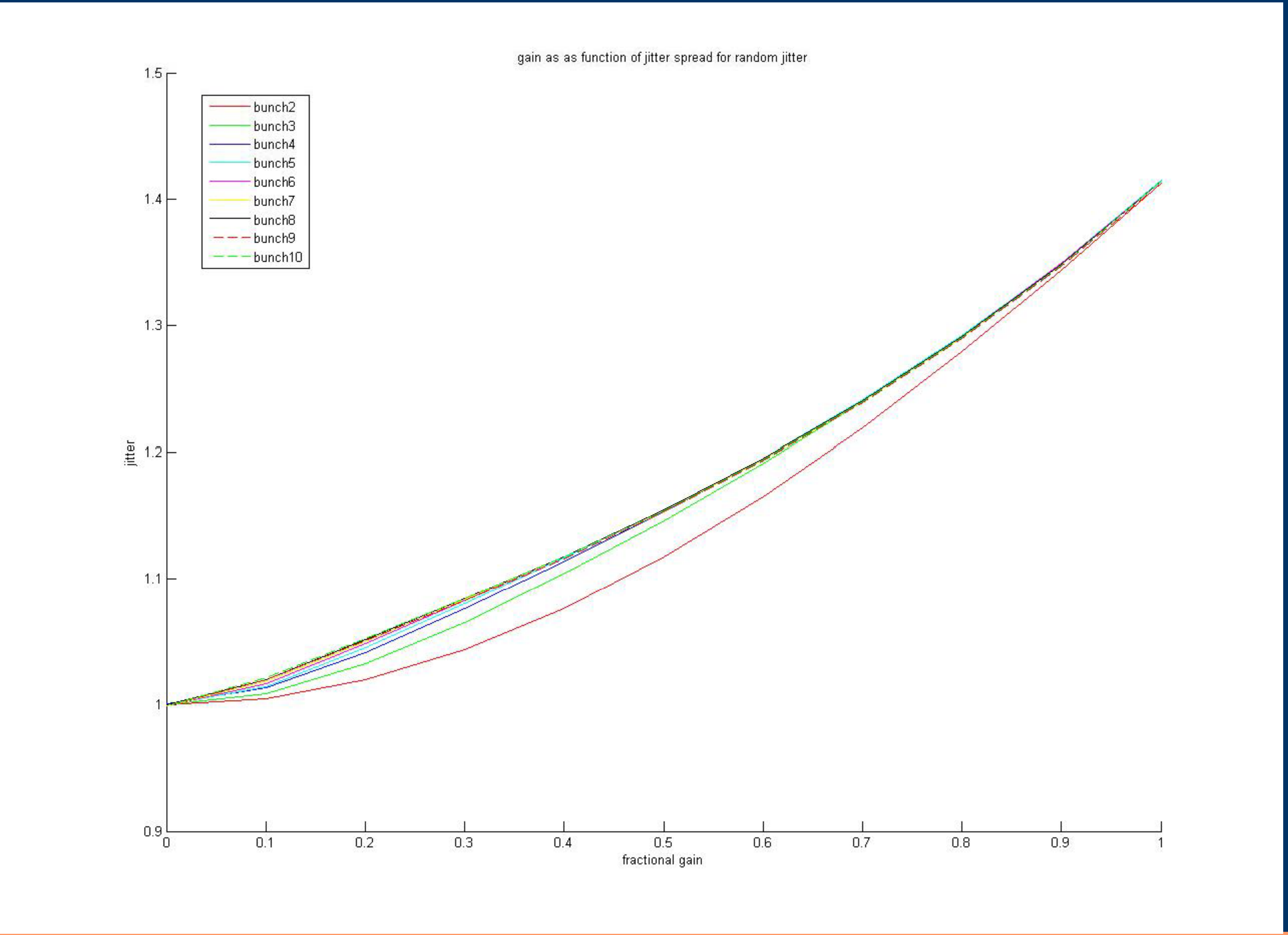
Assumes that unkicked jitter is not changing between feedback off and feedback on runs

Simulation can also be used to take account of the mis-optimsation of the gain.

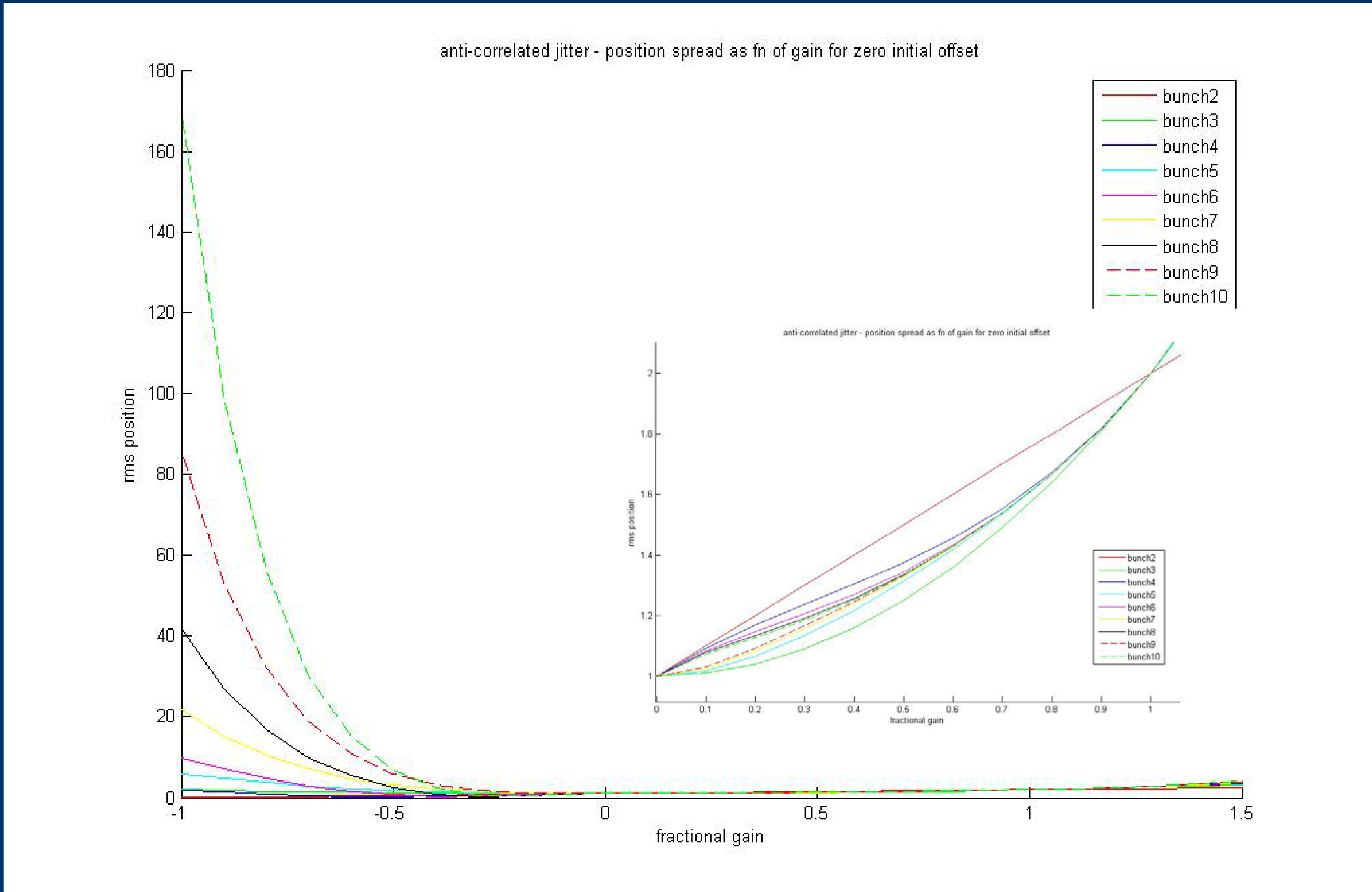
Gain mis-optimisation – correlated jitter



Gain mis-optimisation – random jitter



Gain – anti-correlated jitter (zoom inset)



First results

- Define $\sigma_{FB} = \mathrm{sqrt}(\sigma'_{meas}{}^2 - \sigma'_{pred}{}^2)$
- First pass results:

	σ_2' meas	σ_3' meas	σ_2' pred	σ_3' pred	σ_{FB2} calc	σ_{FB3} calc
Run1	0.0405	0.0138	0.0440	0.0057	0.0173i	0.0126
Run3	0.0323	0.0180	0.0219	0.0135	0.0237	0.0119
Run4	0.0386	0.0197	0.0145	0.0049	0.0358	0.0191

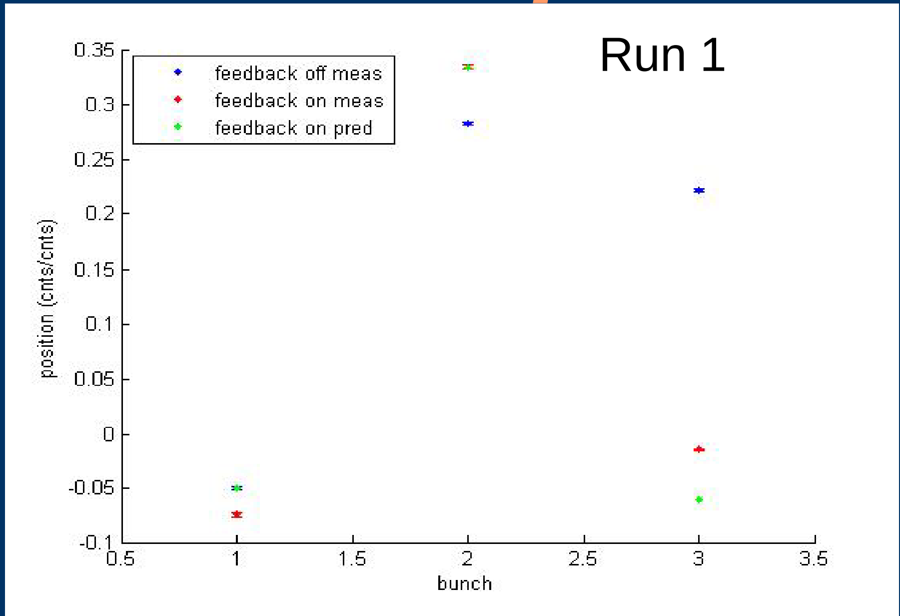
Gain optimisation

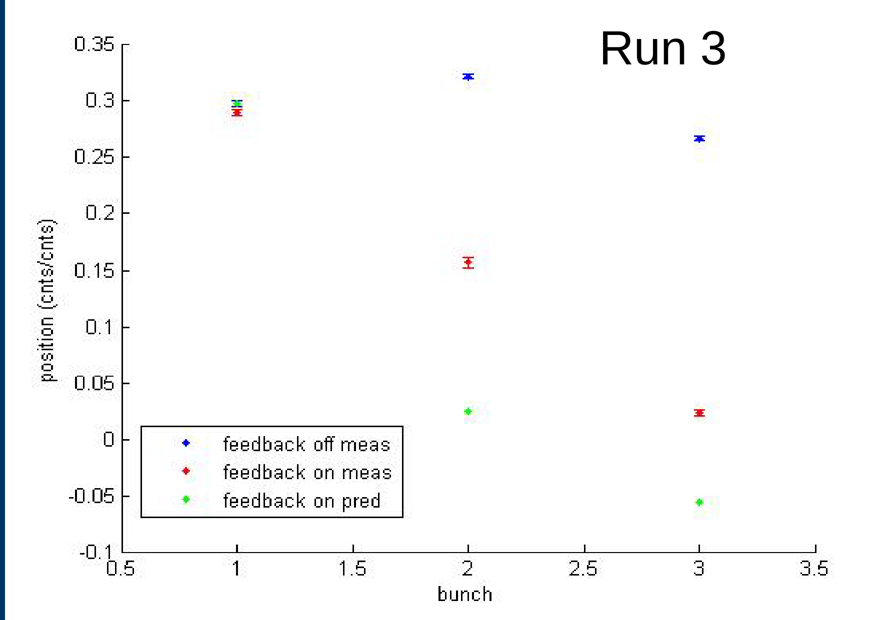
- Originally thought that I could take into account mis-optimisation of the gain:

  Defining, effective gain = (<b2> - <b2'>)/<b1'>

  - Using the actual data for 'feedback off' positions as an input to the simulation with the appropriate effective gain.
  - Could not get consistent results, always saw that for Run3 and Run4 effective gain appeared to be around 0.5!
  - Method assumes that the mean positions of the bunches do not move between feedback on and feedback off data sets (not necessarily true!)
- Have to assume that the gain is correctly set (for now anyway!)

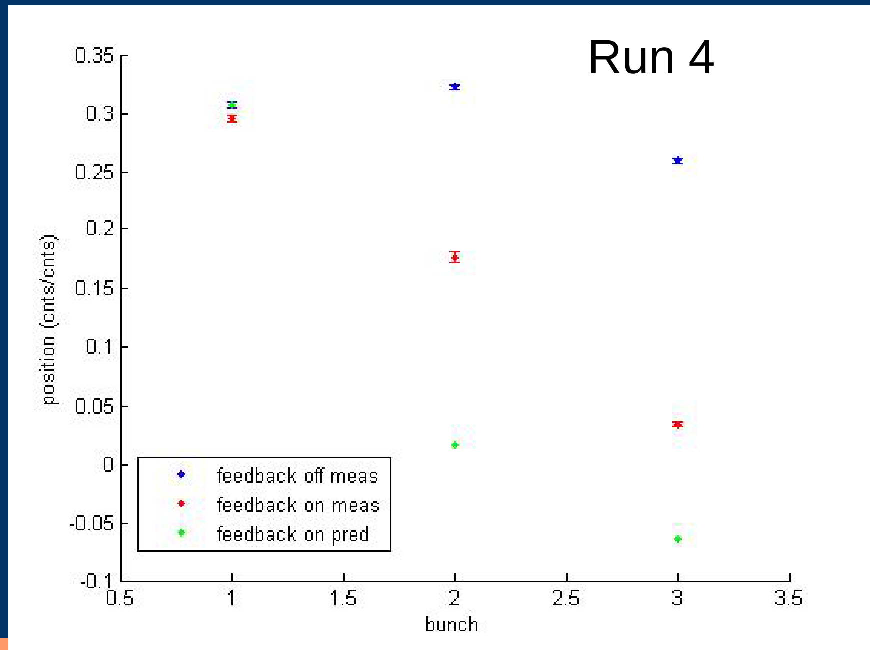
Position plots for three runs



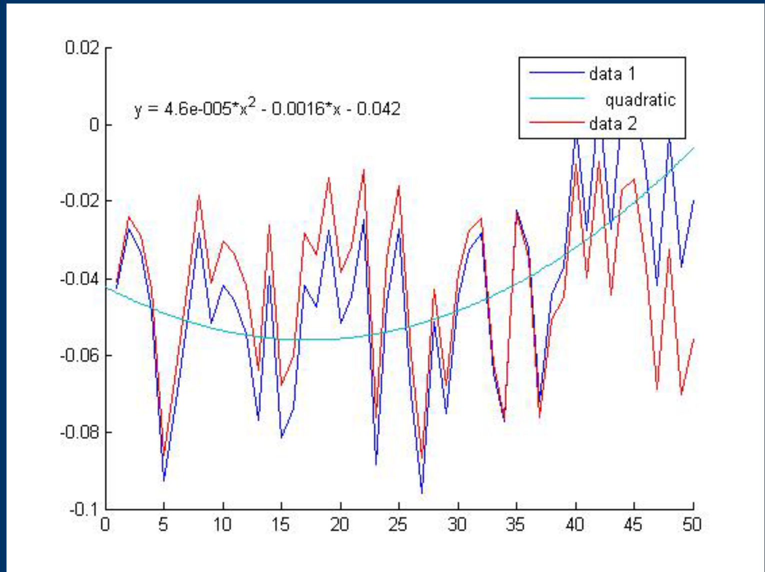


Run1 – bunch3 position seems to have changed relative to bunch1 and bunch2. Gain appears to be correct

Runs 3 and 4 – gain appears to be ~50% too low! Initial positions not changed.



Removal of the periodic variation



- From smaller data sets (i.e. Run4) could more easily remove variation with Quadratic fit.
- Chose to reduce all three data sets down to 50 pulses, and use consecutive data points
  - Run1off_roi=[80:129] Run1on_roi=[1:50];
  - Run3off_roi=[20:69] Run3on_roi=[1:50];
  - Run4off_roi=[1:50] Run4on_roi=[1:50];

Second pass results – with calibration constant of 299 microns

	σ_2' meas	σ_3' meas	σ_2' pred	σ_3' pred	σ_{FB2} calc	σ_{FB3} calc
Run1	0.0151 4.5 um	0.0114 3.4 um	0.0184 5.5 um	0.0041 1.2 um	0.0105i 3.1 um i	0.0106 3.2 um
Run3	0.0090 2.7 um	0.0103 3.1 um	0.0127 3.8 um	0.0050 1.5 um	0.0089i 2.7 um i	0.0090 2.7 um
Run4	0.0082 2.5 um	0.0100 3.0 um	0.0117 3.5 um	0.0048 1.4 um	0.0083i 2.5 um i	0.0087 2.6 um

Interpretation of results (1)

- Had originally (last year) believed that σ_{FB} encompassed all sources which would degrade the resolution. Had convinced myself that it contained the resolution, as all measured jitters are really jitter*=sqrt(jitter^2 + resolution^2), and had thought that as the predicted value contained the resolution twice (i.e. two measured parameters) and the measured jitter once, then the difference between them would be an upper limit on the resolution. In this case imaginary values do not make sense.

- In reality, the predicted jitter contains an extra resolution term which is not present in the real system, therefore it is directly predicting jitter*, and the expected value for the residual would be zero. In this case, the imaginary values are fine as the measured value can be smaller than the prediction.

- Added resolution (uncorrelated jitter) into simulation and verified this.

Interpretation of results (2)

- Not completely sure what σ_{FB} actually is a measure of!
  - Not resolution effects !
  - How well we can actually predict the jitter, due to drift in measured jitter?
  - Gain mis-optimisation (makes sense in terms of the ‘sign’ of the residual)
  - DAC/amplifier effects?
- In any case, from applying calibration constants to results, see that jitter* is small and resolution must be less than this. Can use the fact that bunches 2 and 3 have a good correlation to make an estimate of the resolution (see later).

Summary of jitter results (microns)

		Bunch 1	Bunch2	Bunch3
Run1	FB OFF	6.3	3.1	3.2
	FB ON	6.0	4.5	3.4
Run3	FB OFF	4.1	3.2	3.6
	FB ON	5.3	2.7	3.1
Run4	FB OFF	4.7	3.2	3.4
	FB ON	5.2	2.5	3.0

Estimation of resolution from jitter

- As measured jitter is convolution of actual jitter and resolution, can get an upper-limit estimate of the contribution of the resolution to the measured jitter using the correlation. Assume resolution always looks like uncorrelated jitter and so a strong correlation can only arise from real beam jitter. (For example, if real jitter and resolution were the same magnitude, would measure correlation of 0.5).
- Using Run4 data set: mean jitter (2,3) = 3.3 um; corr(2,3)= 0.91;
  - Jitter = sqrt(0.91)*3.3um = 3.1 um
  - Resolution = sqrt(0.09)*3.3 = 1.0 um
- Similar results from Run1 and Run3:
  - Resolution = 0.9 um (Run1) 1.0 um (Run3)

Some thoughts about resolution measurement (1)

- FONT3 results: From Steve's thesis, single bunch resolutions were 24, 12, 34 um (I've taken the liberty of correcting the geometric factors). For multibunch: 2.8, 5.0, 6.0.
- Steve put the difference down to:
  - Greater power at 714 into the mixer due to bunching frequency
  - Better tuning of LO
  - 20 dB amplification at scope
- Could the difference between classic resolution measurement and jitter measurement just be due to sampling with 14-bit ADCs as opposed to 8-bit scope?

Some thoughts about resolution measurement (2)

- Simulation of resolution calculation
  - Last year simulated both steve's (fitting/mldivide) method and alexander's (direct calculation) method with purely random data and showed that fitting reproduced the initial resolutions better.
  - Looked again at this, but added correlated jitter to the simulation. Found that when jitter >= resolution Alexanders calculation gives a better measurement of the average resolution in the 3- bpm system and the fitting method overestimates
  - For example, putting in corrected jitter with 3.1 um rms, and resolution of 1.0 um, calculation gives back 1.0 micron, compared to 1.2 with the fitting method (incidentally this corresponds to sqrt(2/3)!)
- Not a large effect but may indicate a weakness with the fitting method. Probably deserves further investigation.

Conclusions – jitter and resolution

- Unfortunately the calculation of predicted versus measured jitter does not yield an estimate of the resolution. Theoretically they should be equal and the residual is probably just a measure of jitter drifts in between measurements and the effect of mis-optimisation of the gain. Nevertheless, this ‘error’ appears to be at the level of 2.5-3 microns.

- However, measured jitters are small, 2-3 micron level, so the resolution must be smaller than this. Using the correlation coefficient can estimate the magnitude of the resolution to be around 1 micron.

- A resolution of 1 micron, and correlation of >90% should be enough to reduce the actual beam jitter from 3 microns down to 1 microns.