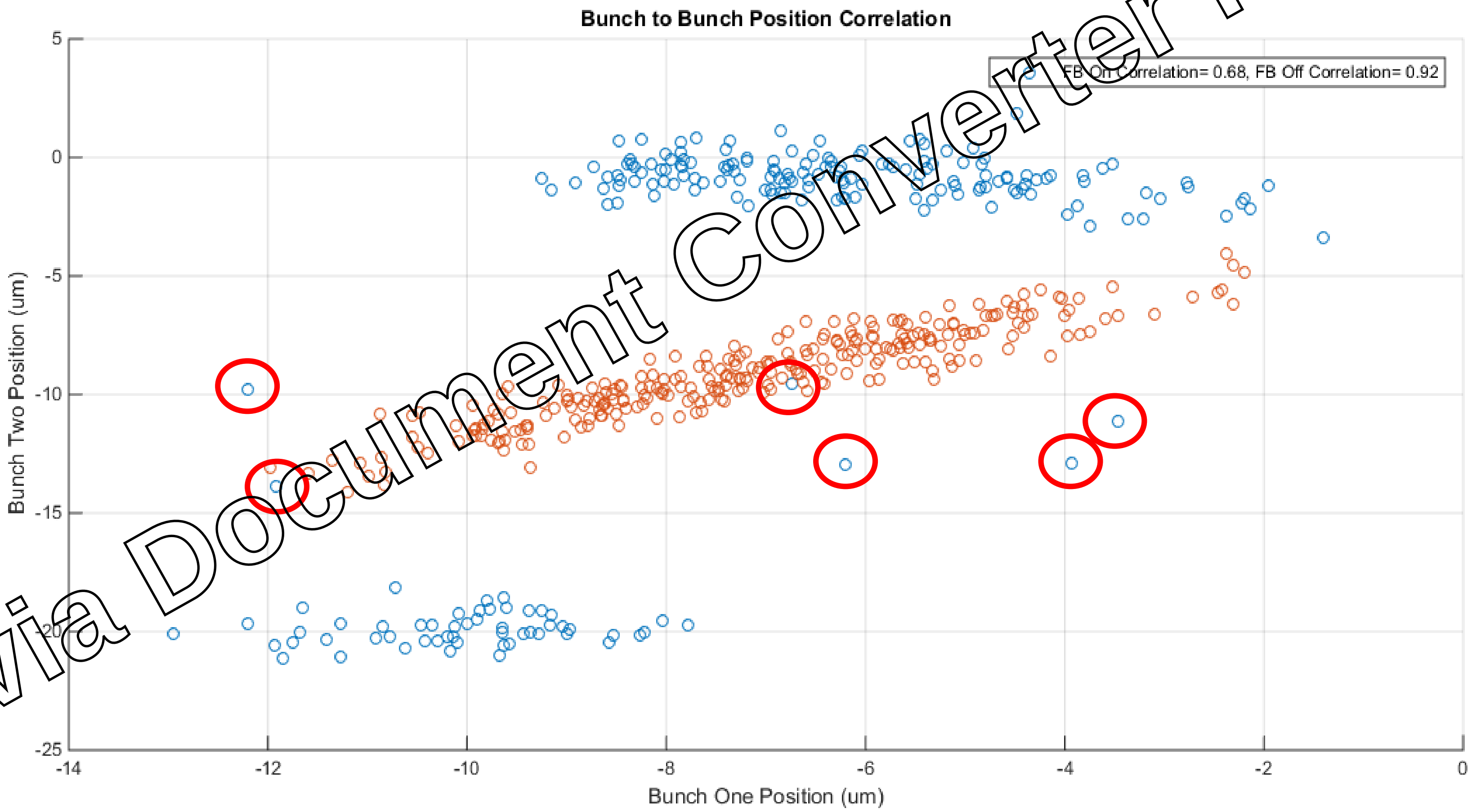
Further Updates on Firmware

Rebecca Ramjiawan

14/07/2017

‘Randomly-Kicked’ Triggers



Timing References

- Board stamp: Time at which the board starts transmitting data.
- Pulse counter: counter that increments on receiving a trigger.
- Timestamp: "**logged by the DAQ software upon receipt of the data**, but as the exact stamp obtained by the computer depends upon factors such as network and processing performance, this software stamp is less accurate" – Ben Constance's thesis

Considerations from last meeting (1)

- From last meeting: (ipfbRun25) 45/1000 of measured **triggers changed phase**: we would expect a 0.66s gap between these two triggers suggesting an intermediate trigger not recorded.
  - This is the case - each pair of triggers that change phase corresponds to a timestamp jump of 0.66s with a minor caveat that I will explain later.
- **Kick values that wrap around** do so on the summed signal - rather than on one BPM only. None of the .dac values in any file saturate in one BPM only because the signal bit width is reduced to 13 bits only at the very end of the FB module after summation of inputs from both BPMs.

Considerations from last meeting (2)

Slow acquisition speed for these files (15/06/2017): consistent with dropping data packets.
Pulse counter from trigger to trigger looks fairly random.
E.g 56, 10, 34, 6, 24, 14, 48, 40, 56, 46, 58, 38, 22 etc.
Timestamps have non constant gaps between successive triggers with values generally larger than 1s.
E.g 01.118 s, 06.060 s, 07.341 s, 08.942 s etc.
Board stamps have non constant gaps between successive triggers. Most gaps are multiples of 8.
E.g 111, 15, 55, 111, 15, 95, 63, 102, 46 etc.

Fixing Firmware Bugs

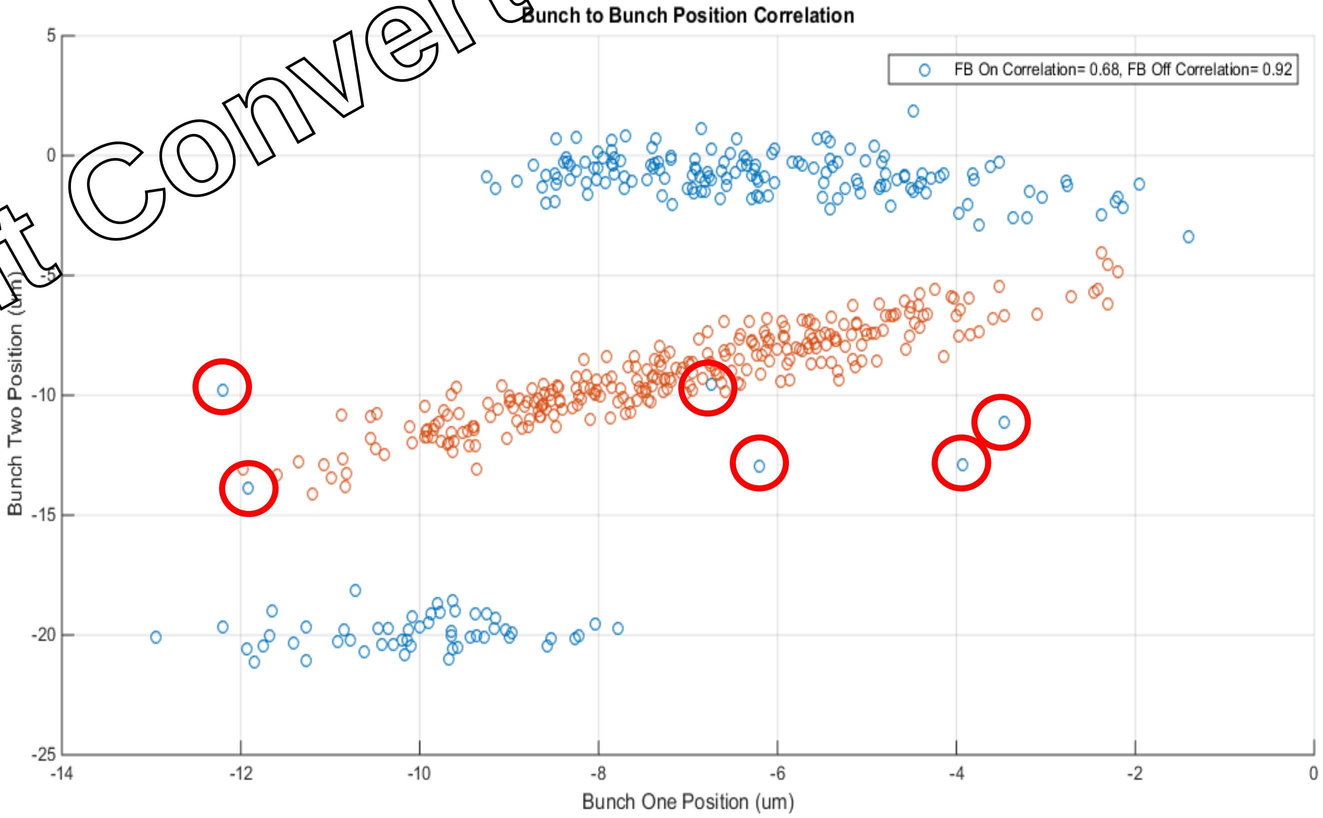
Rebecca Ramjiawan

Rogue triggers

- All bunches (from any data set) which were kicked in random incorrect directions fit into a very specific subset of data:

All incorrectly kicked triggers are part of two consecutively saved triggers each with two bunches, where the first of the two triggers has FB off.

- All incorrectly kicked triggers had this pattern:
  - 02**_0_22**02
  - Where green is FB on and red is FB off.



Example double two bunch triggers

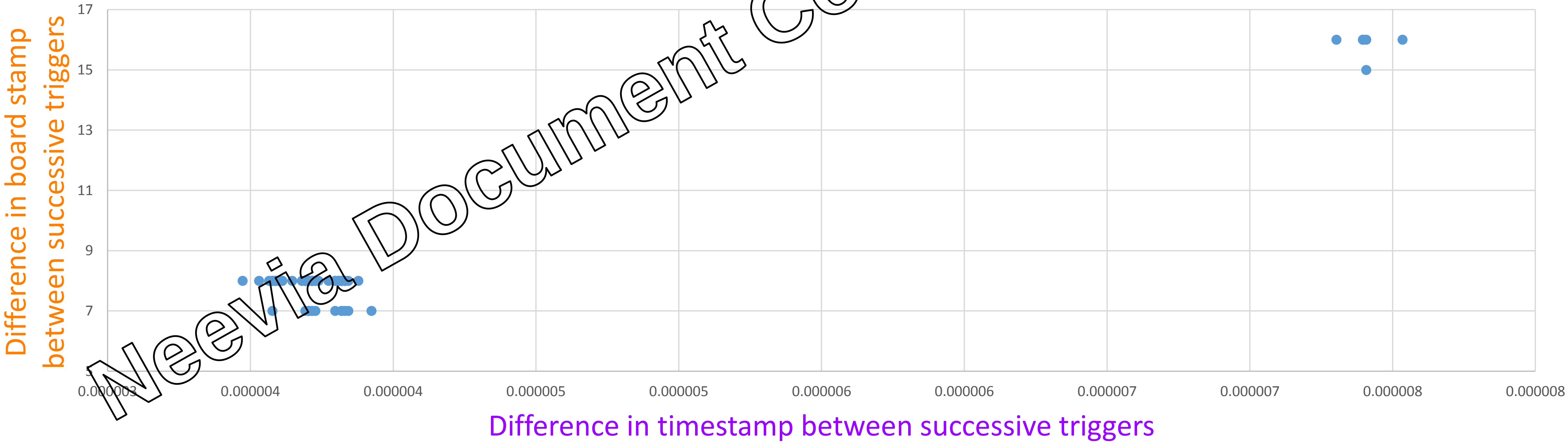
- Timestamps between double two bunch triggers are always only ~0.33s indicating no missing data packets, which we know cannot be the case.
- However, every double two- bunch trigger is followed by a 0.66s gap suggesting that the timestamps and board stamps are offset by one trigger.

Trigger Number	Number of bunches	Timestamp (s)		Pulse counter		Board stamp	
381	Empty	26.989		45		108	
382	Empty	27.310	0.321 s	51	+6	124	+16
383	**2 bunches**	**27.949**	0.639 s	**57**	+6	**4**	+8
384	**2 bunches**	**28.270**	0.321 s	**63**	+6	**20**	+16
385	2 bunches	28.910	0.640 s	5	+6	36	+16

Board Stamp vs Timestamp

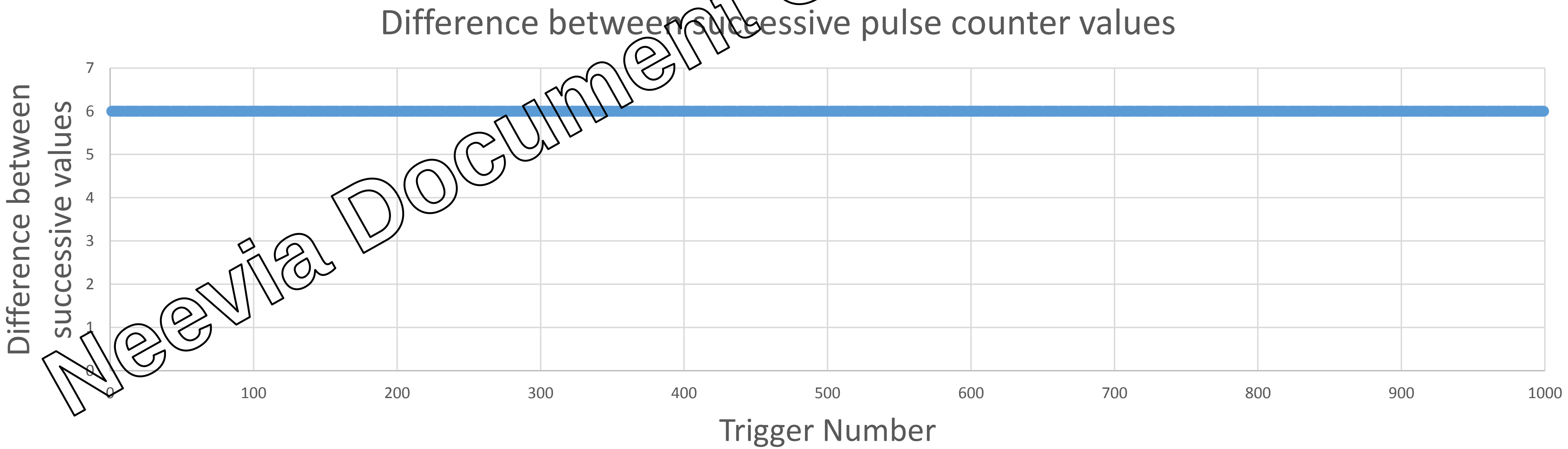
- It appears that board stamp and timestamp are offset by one trigger in data file.
- E.g: difference in board stamp between triggers 1 and 2 corresponds to difference in time stamp between triggers 2 and 3 etc.

Board stamp difference between triggers n, n+1 vs. timestamp difference between triggers n+1, n+2



Pulse counter

- Pulse counter increments by 6 in ATF but in the lab it increments in steps of 1.
- If it is picking up reflections/noise on the trigger would you expect some variation on this value of 6 due to the randomness of its nature?
- Also increments by six in ALL files taken on the 20/06/2017, but not on those files with slower acquisition speed 15/06/2017 which increments randomly and by large amounts.



Timing References Summary

- Board stamp: Time at which the board starts transmitting data.
  - Between two double triggers this increments by 16 rather than the usual 8 showing unrecorded data in between pulses.
- Pulse counter: counter that increments on receiving a trigger.
  - Pulse counter always increments by 6 – whether interleaved/empty trig blk is on or off.
  - N.B this increments in steps of one when measured in the lab with a clean trigger.
- Timestamp: Time that DAQ receives data.
  - Each double two-bunch pulse comes associated with a jump in timestamp of 0.66s, but the triggers that the 0.66 s gap occurs between are a pair one trigger in the future from the pair we would expect.

Rogue triggers

Each line has three recorded triggers.

- Red lines indicate the final trigger in the sequence has a value very far from the correct output.
- Green lines indicate the final trigger has an apparently correctly calculated value + previously mentioned offset.

	Trigger 1	**Trigger 2**	**Trigger3**	**Trigger4**
Double two-bunch pulse	(FB On – Noise)	(FB Off – Data)	**lost data**	(FB On – Data)
	(FB On – Noise)	**lost data**	(FB Off – Noise)	(FB On – Data)
	(FB On – Noise)	(FB Off – Data)	(FB On – Noise)	

- It didn't make sense to me that losing a packet of data should have an impact on the feedback performance for future triggers, as the feedback system shouldn't know that it has missed a trigger.
- I thought it was the combination of running FB on noise (Trigger 1) and then running FB on data (Trigger 4) that somehow broke the FB module. But the pattern displayed in line 2 worked but line 3 didn't!

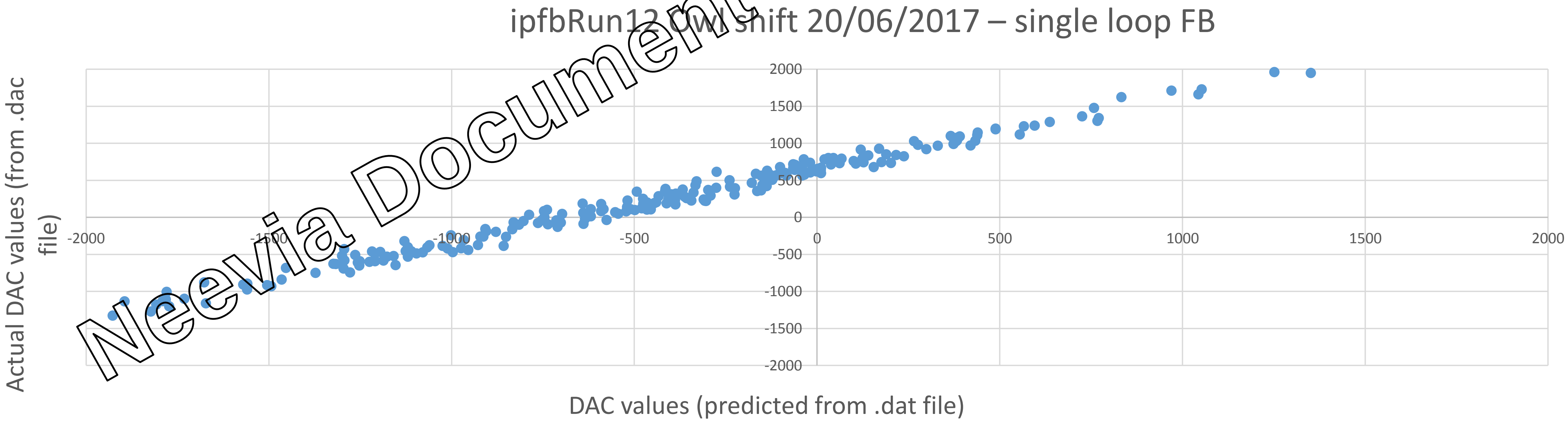
Finding a Solution

- It took me the entire week (and a lot of trips to Glenn's office) to figure out that the only thing the missing trigger had to do with the randomly kicked bunches was that the slip in phase allowed us to see a mistake the FB module was ALWAYS making.
- The key to whether a bunch was randomly kicked lay with what the FB module was doing when feedback was OFF. Whenever you had a trigger with FB off and two bunches the subsequent pulse would be kicked with a value that *apparently* bore no resemblance to the correct value.

Trigger 1	**Trigger 2**	**Trigger3**	**Trigger4**
(FB On – Noise)	**(FB Off – Data)**	lost data	(FB On – Data)
(FB On – Noise)	lost data	(FB Off – Noise)	(FB On – Data)
(FB On – Noise)	**(FB Off – Data)**	(FB On – Noise)	

Two Birds - One Stone

- The feedback system has no knowledge of whether there are bunches or not for FB off so it is unreasonable to believe that it was only going wrong for [FB OFF & 2 bunches] and not also for [FB OFF & 0 bunches].
- The mistake I made when considering the green row from the previous slide was to assume it had correctly calculated the kick and that the **offset (seen last week) was a separate problem** and not the same problem manifesting itself in different ways.



Two Birds – One Stone cont.

- Both problems lay with the FB module itself and were in the mechanism to clear the ‘sample integrator’ which was cleared only when FB was ON.
- **Therefore, if FB was OFF the ‘sample integrator’ would roll the data from the FB off pulse forward.**
  - If FB off on noise it would roll forwards a small value – **slight offset from correct value.**
  - If FB off on data if would roll forwards a large value – **after wrapping around the kick appeared random.**
- Usually if FB OFF was for 2 bunches, FB ON was on noise and I mistakenly excluded this data from my analysis. So I never picked up on this issue until slipped phase shifted bunches into this pulse.
- Most data had FB OFF on noise and FB ON for 2 bunches and this created the offset for bunch one.
- After bunch one was finished, as FB was ON the sample integrator was cleared finally and **bunch two was then calculated correctly.**
- In the first shift we had **no triggers with FB OFF** so all kicks for bunch one and bunch two were calculated correctly.

Mysteries remaining

1. Why isn’t ‘empty trigger blk’ blocking any of the triggers? Is this an effect of trigger reflections?
2. Why is interleaving 0101010101 not 0011001100110011? Empty trigger block?
3. Why is pulse count progressing 6 times too quickly? What is it registering a signal on? Noise? Reflections? If so why is it so constant at 6?
4. Is the board stamp and data indeed offset from the time stamp by a trigger and if so why?

LabVIEW

- We actually ran with the same LV DAQ on both shifts, any changes to the DAQ occurred on the 14/06/2017 before our first shift.
- The increase in acquisition speed isn't from alterations in the DAQ.
- In future, run without 1 in N mode to reduce problems with acquisition speed?