Australian Government | ANSTO

FOFB – Soft or Hard Realtime?

Eugene Tan
Accelerator Physics and Operations
Australian Synchrotron (AS-ANSTO)

Science. Ingenuity. Sustainability.

Summary

- History
- Fast Orbit Feedback System
- Soft Realtime Fast orbit Feedback
- Soft vs Hard Realtime Fast Orbit Feedback

History

- 2005: Australian Synchrotron commissioned
- 2006: Users

History

- 2005: Australian Synchrotron commissioned
- 2006: Users

History – Orbit Feedback

- 2005: Australian Synchrotron commissioned
  - Orbit Feedback at 0.25 Hz through Matlab script/application.
  - Power supply: multi-drop serial 9600 baud.
- 2011: Fast Orbit Feedback Project started
  - Platform to develop in-house FPGA expertise
  - Locally built power supplies
  - Libera Grouping vs Communication controller
  - Prototype FPGA
- 2014: Prototype fast power supply
  - ±1 A, 10 kHz update via 10 Mbaud serial Soft vs Hard Realtime Fast Orbit Feedback
  - Engineer developing the FPGA leaves.

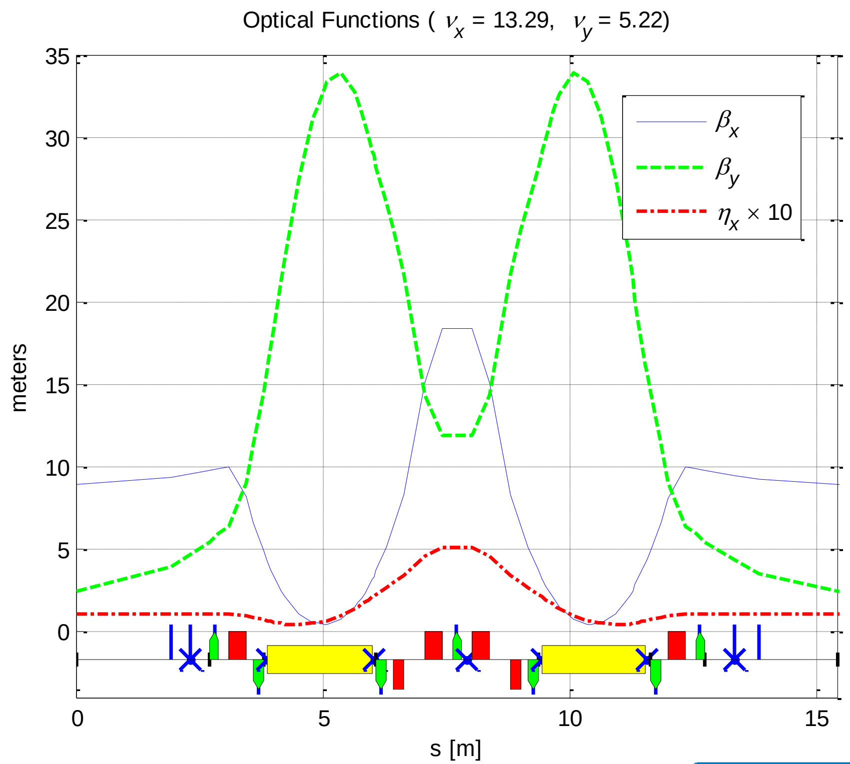
History

- 2015
  - Delivery of Fast Power Supplies
  - RTLinux FOFB project started
  - FPGA housing
  - Arrayware contracted
  - New FPGA engineer joins AS.
- 2016:
  - FPGA based FOFB in operation

AS Lattice

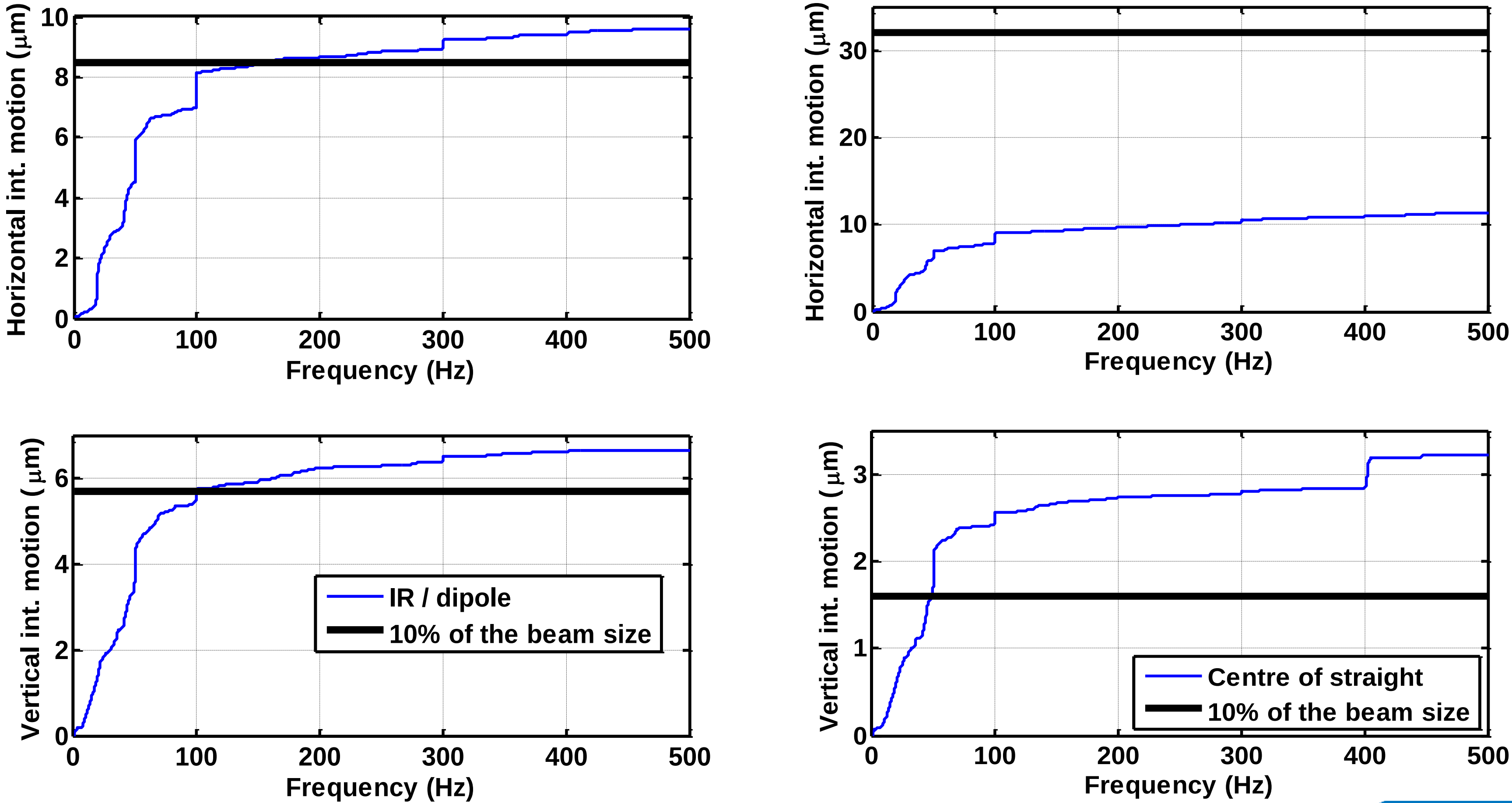
- 14 fold symmetric double bend arcs.
- 98 BPMs
- 42 Horiz and 56 Vert correctors (< 5 Hz)

Parameters	Values
E_0	3 GeV
Circumference	216 m
Straight Lengths	4.6 m
RF (Voltage / Freq)	3 MV / 500 MHz
I_0	200 mA
Emittance (η_x = 0.1)	10 nm
ID Beamsize (H/V 1%)	320 / 16 um
Bend Beamsize (H/V 1%)	87 / 58 um



ANSTO

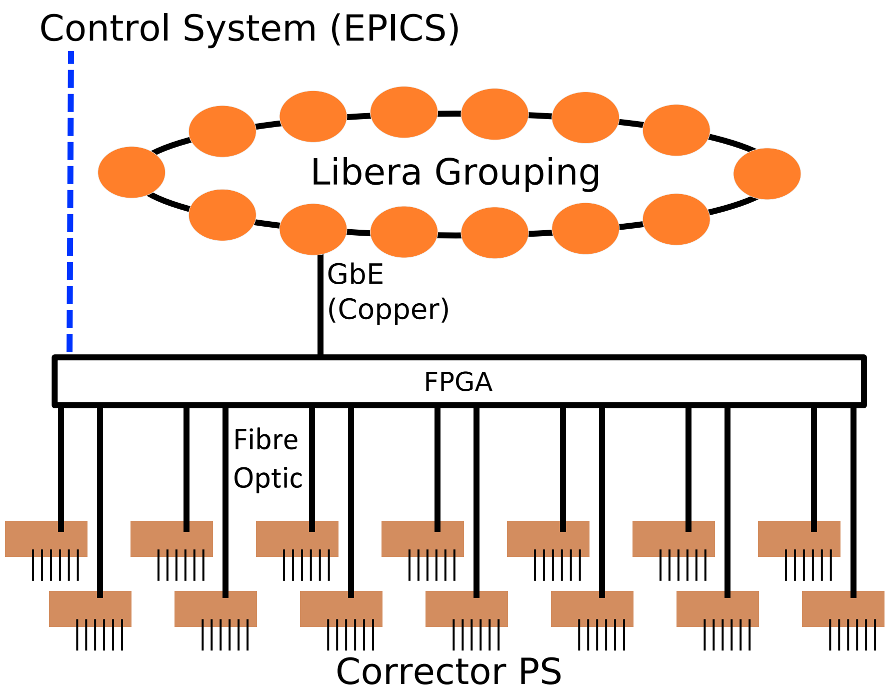
AS Lattice

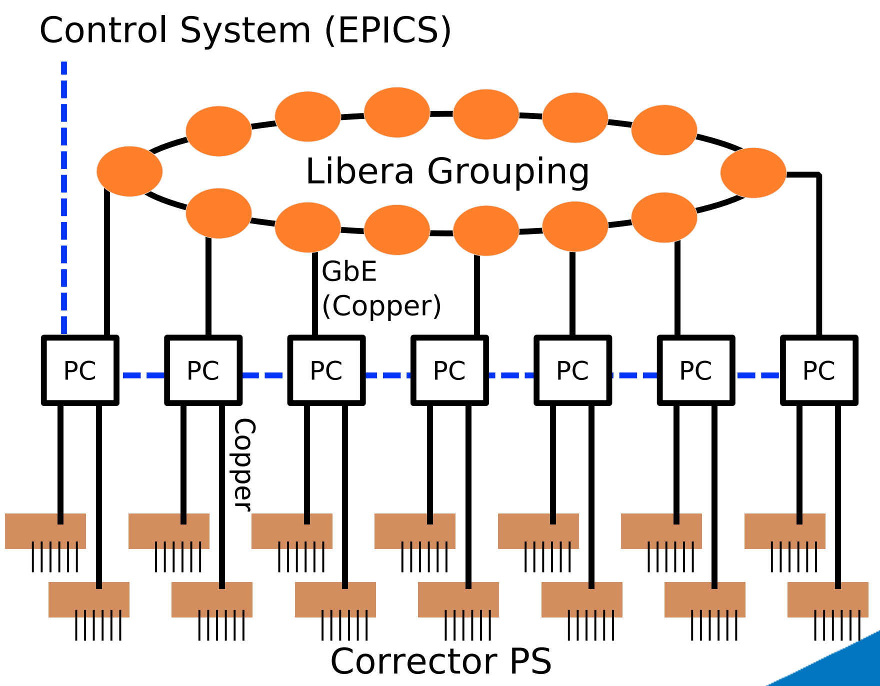


ANSTO

FOFB Topology

- 98 BPMs
- 42 Horiz and 56 Vert correctors (< 5 Hz)
- 42 Horiz and 42 Vert fast correctors (> 2.5 kHz)

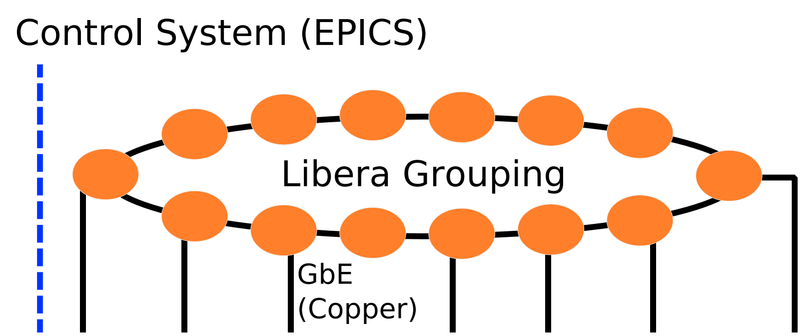


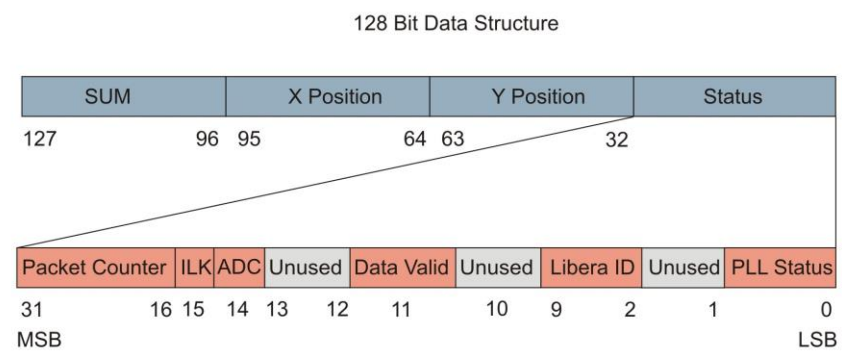


ANSTO

FOFB Topology – Libera Electron

- Libera Grouping
  - I-Tech upgraded to handle 128 BPMs
  - Single bi-directional loop (copper and FO)
  - GbE
  - 10 kHz

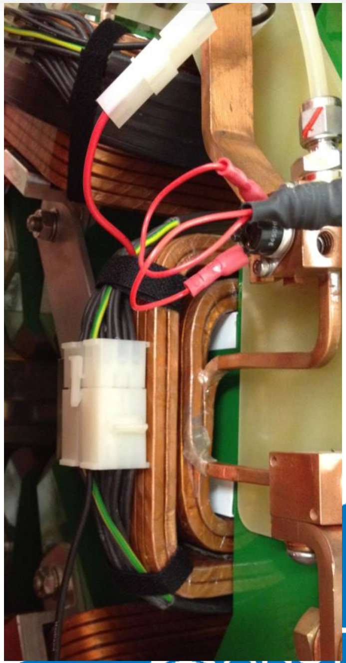


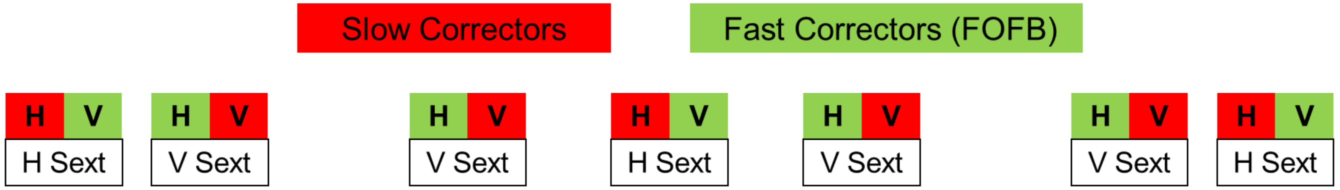


ANSTO

FOFB Topology - Magnets

- 6 Channel ±1A/14V
- Correctors – secondary and tertiary coils on sextupoles
- Sextupole
  - Slow Horizontal corrector
  - Fast Vertical corrector
  - 12 turns each





FOFB Topology – PC Based Controller

- What other choices:
  - VxWorks
  - RTAI
- Why RT Linux?
  - Control system IOCs all CentOS,
  - Patched kernel on CentOS means little or no change to existing software and controls infrastructure.
  - Seemed easier by comparison…

Real-Time Linux PREEMPT - Hardware

- Prototype
  - Dual core Intel Core2 Q8400 2.66 GHz
  - 4GB RAM and recycled HDD
  - CentOS 5 kernel 2.6.29.6-rt24
  - Dual port Intel NIC
  - AXXON 10 Mbaud serial card

Real-Time Linux PREEMPT - Prototype

- RTLinux Kernel patch
- Serial Driver for the AXXON card

- Libera Grouping interface (FA data decoding)
  - FA Archiver (M. Abbott, Diamond)
- Power Supply Interface

- Measure the peak to peak jitter
- Maximum cycle rate

Real-Time Linux PREEMPT - Prototype

- RTLinux Kernel patch
  - https://rt.wiki.kernel.org/index.php/Main_Page
  - Straight forward instructions and worked first time round.
  - Make sure you have EXACTLY the same OS kernel as the patch.

- Serial Driver for the AXXON card
  - For me this was a nightmare…
  - AXXON has instructions to patch the 8250 driver
  - set as module during kernel build, Modify driver, Build and reload driver.

Documentation

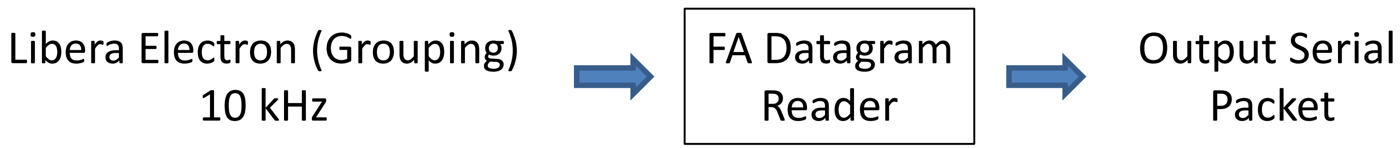
- Frequently Asked Questions
- CONFIG_PREEMPT_RT Patch
- RT Patch tar files
- Actively maintained PREEMPT_RT kernel patces
  - Latest 4.4-rt
  - Latest 4.1-rt
  - Latest 4.0-rt
  - Latest 3.18-rt
  - Latest 3.14-rt
  - Latest 3.12-rt
  - Latest 3.10-rt
  - 3.4-rt Stable README
  - 3.2-rt Stable README
- No longer actively maintained
  - 3.0-rt 3.0 README
  - 2.6.33
  - 2.6.31
  - 2.6.29
  - 2.6.26
  - 2.6.25
  - 2.6.24
  - 2.6.23
  - 2.6.22
- rt-tests(testsuite) git repo
- rt-tests tarballs
- RT PREEMPT HOWTO
- Reporting Bugs
- Building Embedded Linux Systems, 2nd edition
- Earliest Deadline First, EDF papers
- Oleg's QRCU RCU (read/copy/update) LWN article
- Parallel Algorithm Verification of QRCU

More Documentation

Real-Time Linux PREEMPT - Prototype

- RTLinux Kernel patch
- Serial Driver for the AXXON card

- Libera Grouping interface (FA data decoding)
- Power Supply Interface (8-n-1)



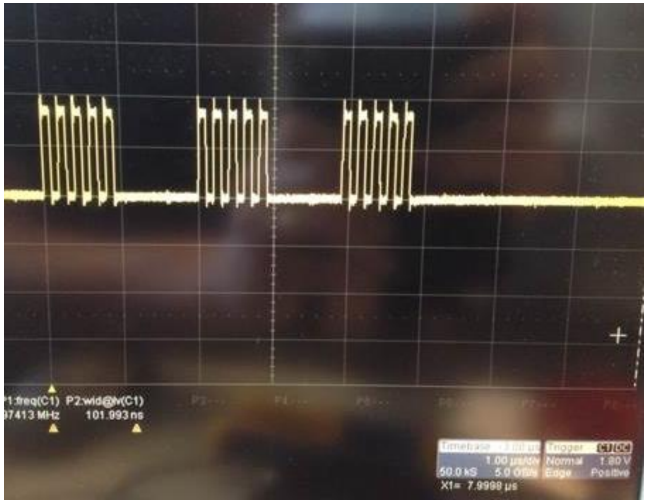
Real-Time Linux PREEMPT - Prototype

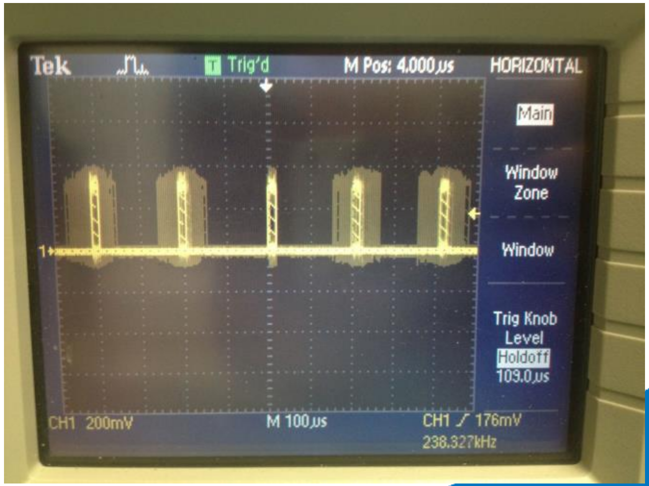
- Used internal OS clock to time the FA datagram period.
- Choice of network card is important.
- Realtek (RTL8111/8168/8411) onboard NIC
  - Period: 1 μs to 400 μs
- Intel (82574L)
  - Period: 70 μs to 130 μs

ANSTO

Real-Time Linux PREEMPT - Prototype

- Serial Output with a gap between bytes
- Each byte 1 us @ 10 Mbaud
- Gap of ~ 1us
- Solution:
  - Disable transmit, Fill buffer, Enable transmit
  - 60 us latency.
- Cause:
  - OS related, and not filling the buffer quickly enough
  - Problem was not present with CentOS 7





ANSTO

Real-Time Linux PREEMPT - Prototype

- Peak to peak jitter measurements
  - Internal clock Serial 1 port : 80 us
  - Internal clock Serial 2 ports: 110 us
  - FA datagram triggered (every 2nd packet) and serial 1 port: 120 us
  - FA datagram triggered (every 2nd packet) and serial 2 port: 172 us
  - [eth reader thread and serial writer thread]
  - FA datagram triggered (every 2nd packet) and serial 2 port: 188 us

```
root     -50   0 22920  22m 3816 R 99.8  0.6   0:06.11 fofb_pc
root     -86  -5     0    0    0 S 90.8  0.0  35:56.39 IRQ-18
root     -86  -5     0    0    0 S 47.9  0.0  20:08.81 IRQ-17
root      20   0  237m 196m 4256 R 15.9  4.9 101:11.75 Xorg
root     -50   0 22920  22m 3816 S  8.6  0.6   0:00.51 fofb_pc
root     -86  -5     0    0    0 S  4.3  0.0  21:54.32 IRQ-32
root     -76  -5     0    0    0 S  3.3  0.0  11:07.57 sirq-net-rx/3
root     -76  -5     0    0    0 S  2.6  0.0  19:36.08 sirq-net-rx/1
root     -76  -5     0    0    0 S  2.3  0.0  18:18.93 sirq-net-rx/0
```

- IRQ scheduling works
  - eth (IRQ-32) cpu2+cpu3
  - Serial (IRQ-17 + 18) cpu0+cpu1
  - Move USB interrupts away from cpu0 and cpu1.
  - Jitter reduced from 188 us → 136 us

Real-Time Linux PREEMPT - Prototype

- RTLinux Kernel patch
- Serial Driver for the AXXON card

- Libera Grouping interface (FA data decoding)
  - FA Archiver (M. Abbott, Diamond)
- Power Supply Interface

- Measure the peak to peak jitter
- **Maximum cycle rate → 3.33 kHz**

Real-Time Linux PREEMPT - Prototype

- What did I learn?
  - Good network card
  - Threading
  - IRQ scheduling
  - Main bottleneck is the serial output. Nothing could be done in the near term.

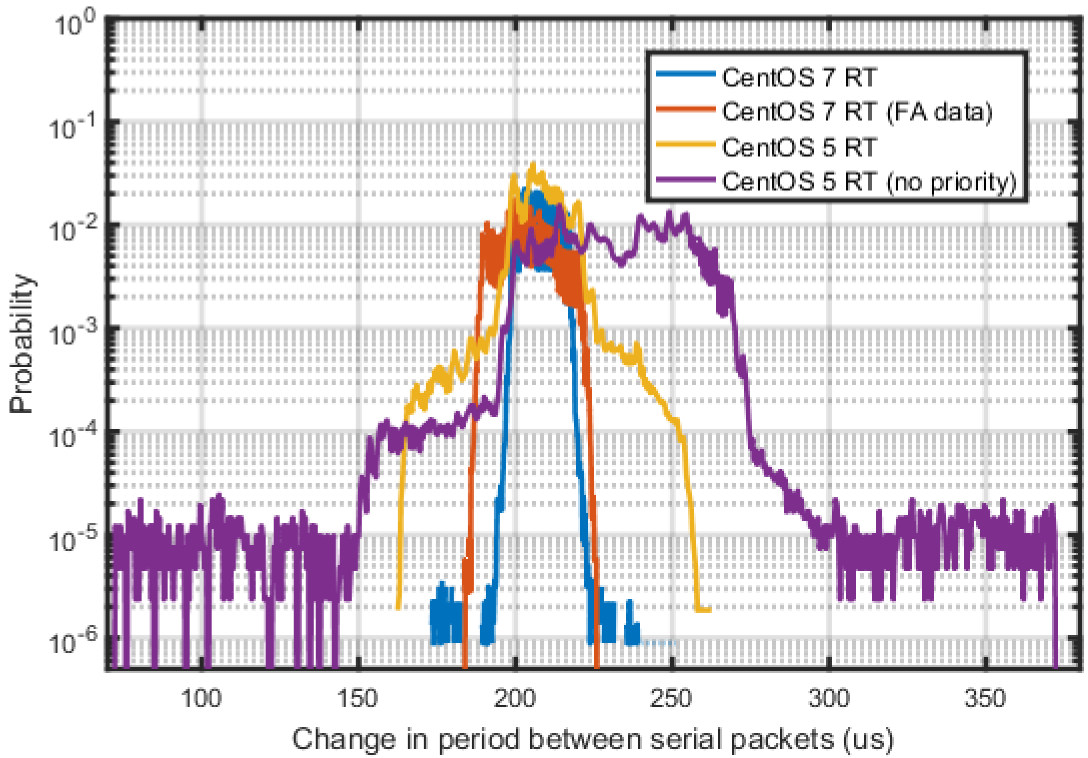
Real-Time Linux PREEMPT - Production

- 7 PCs ($330/each → $2310):
  - Single core Intel Celeron G1840 2.8 GHz (IRQ scheduling does not improve jitter by much)
  - 2GB RAM and Recycled HDD
  - CentOS 7 (kernel: 3.10.75-rt80)
  - Single port NIC: Intel (82574L) GbE

- 7 Serial Cards ($280/each → $1960):
  - AXXON LF686KB PCIe 2 Port RS422/RS485

Real-Time Linux PREEMPT - Production

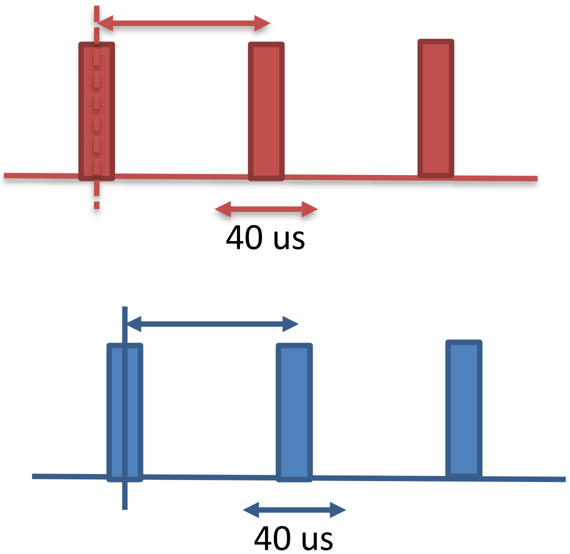
- Tested with CentOS 5 (kernel 2.6.29.6-rt24) and CentOS 7 (kernel 3.10.75-rt80) on the production PCs.
- Initially seeing gaps between bytes on CentOS 5. **Not seen in CentOS 7.**
- Using "clock_nanosleep" an application was developed to send serial data at 5 kHz and FA datagram triggered.

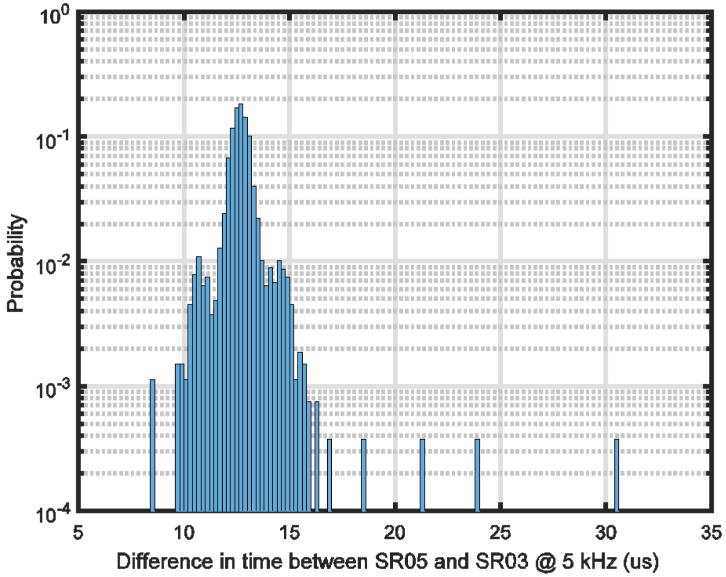


ANSTO

Real-Time Linux PREEMPT - Production

- Synchronicity between different PCs.
- Spread period between packets ~40 us
- Spread between PCs ~7 us!
- Is the jitter in the FA data correlated?





ANSTO

Real-Time Linux PREEMPT - Production

```
EPICS IOC FOFB PVs :
   open/close loop
   shutdown process
   inverse response mat.
   reference orbit
   Kp and Ki coeff.
   Harmonic supp. coeff.
   Decimation
   diagnostic data
```

```
FOFB service () {

Realtime setup: sched_priority
Open serial and eth ports;
Initialise EPICS;
Open EPICS channels;

Read all config. param.;
Create thread with RT priority;
Start thread for FOFB_main

Loop (5 Hz){
   EPICS poll
   mutex lock;
   read loop state;
   read DCCT;
   read param;
   write diagnostic data;
   read shutdown command;
   mutex unlock;
}

Thread join;
Clean up;
}
```

```
Global
param
```

```
FOFB_main () {
   local param;
…
}
```

Real-Time Linux PREEMPT - Production

```
FOFB_main () {

Clear eth buffer;

Loop:
   read datagram;
   sort data based on ID;
   if frame_count % 2^14 (2,4,8, .. Hz)
      mutex lock;
      set local shutdown status;
      set local loop state;
      set local param;
      mutex unlock;

   if frame_count % decimation
      continue;

   Integrity checks
      frame timing < 20%;
      position < 250 um;
      change in sum < 0.1%;
```

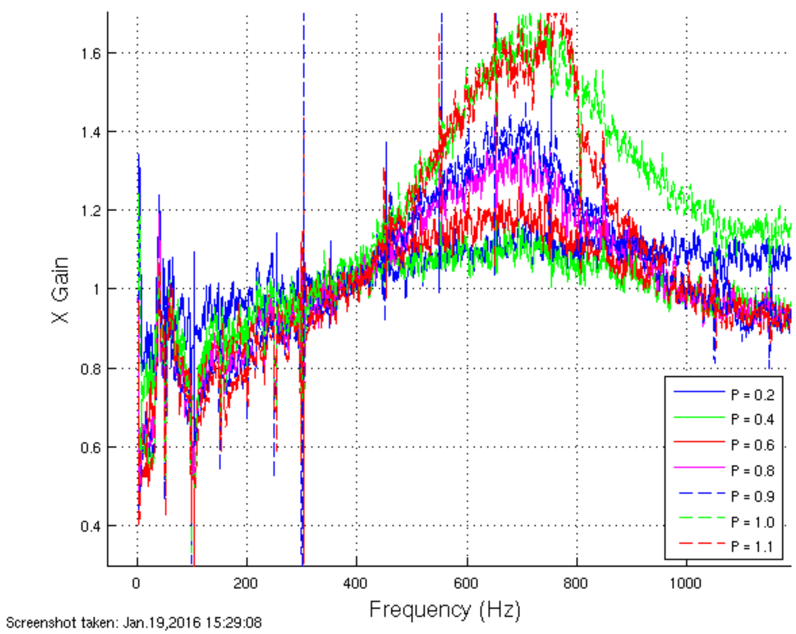
```
   if closed loop
      calc PI corrections;
      calc harm sup. Corrections;
      calc corrector average (MAF)
      mutex lock;
      write diag. data to global param;
      mutex unlock;
      output corrections;
   else
      output average


Clean up;
}
```

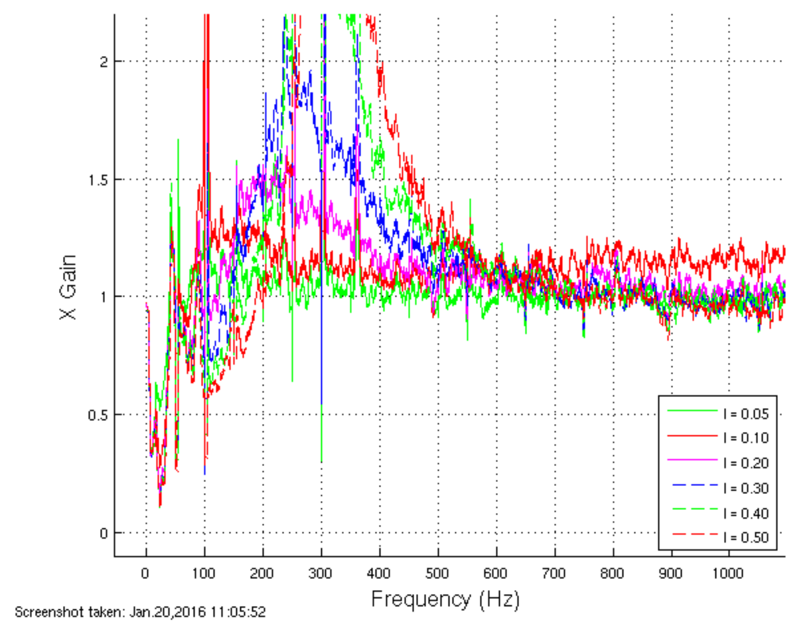

ANSTO

Real-Time Linux PREEMPT - Production

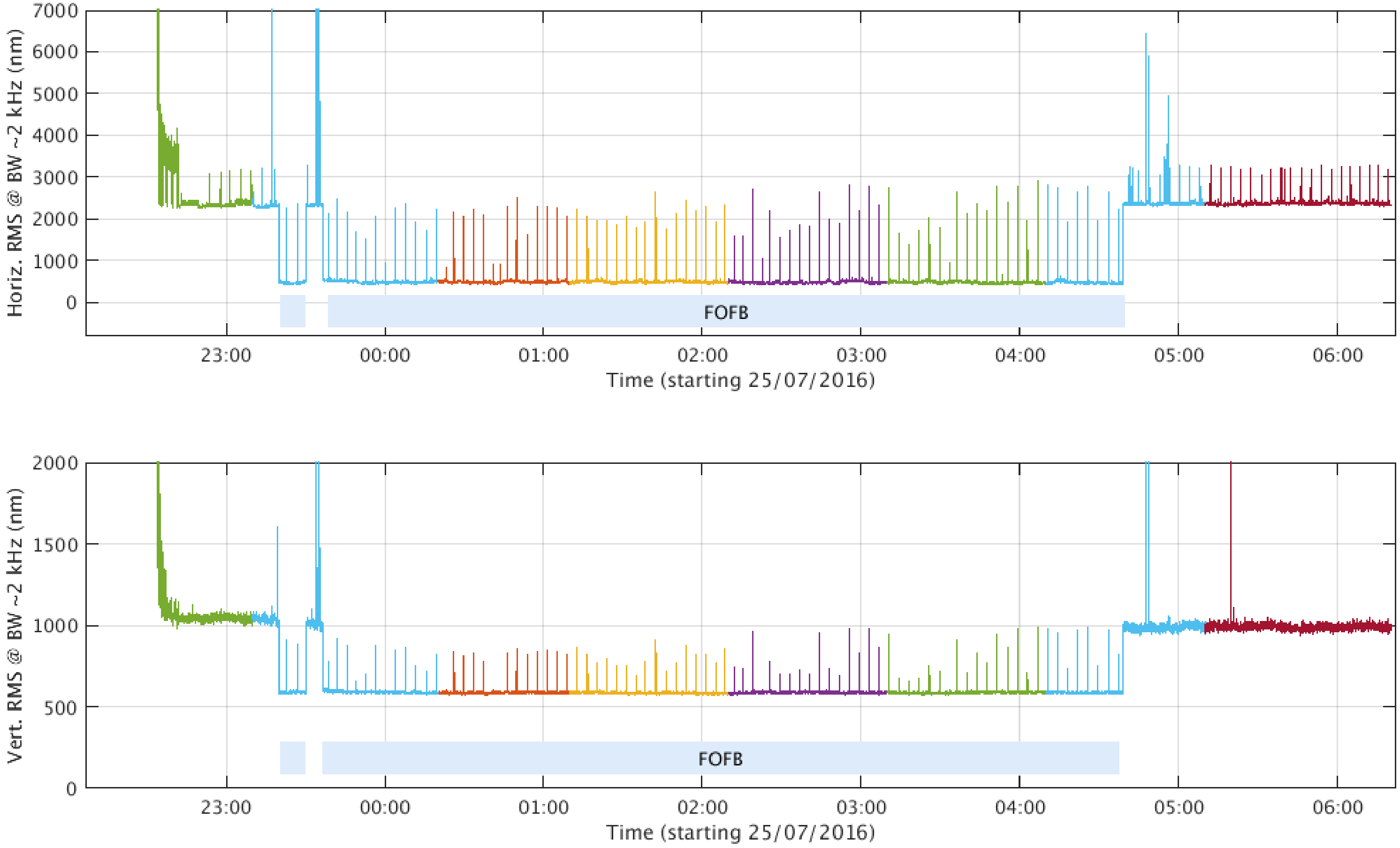
- At the time it was not possible to estimate the PC processing latency.
- Could we go to 10 kHz?
- No.
- At 10 kHz the system would fail after a few seconds. Implying the processing delay was just under 100 us.

RTL Based FOFB (5 kHz) – Early Results

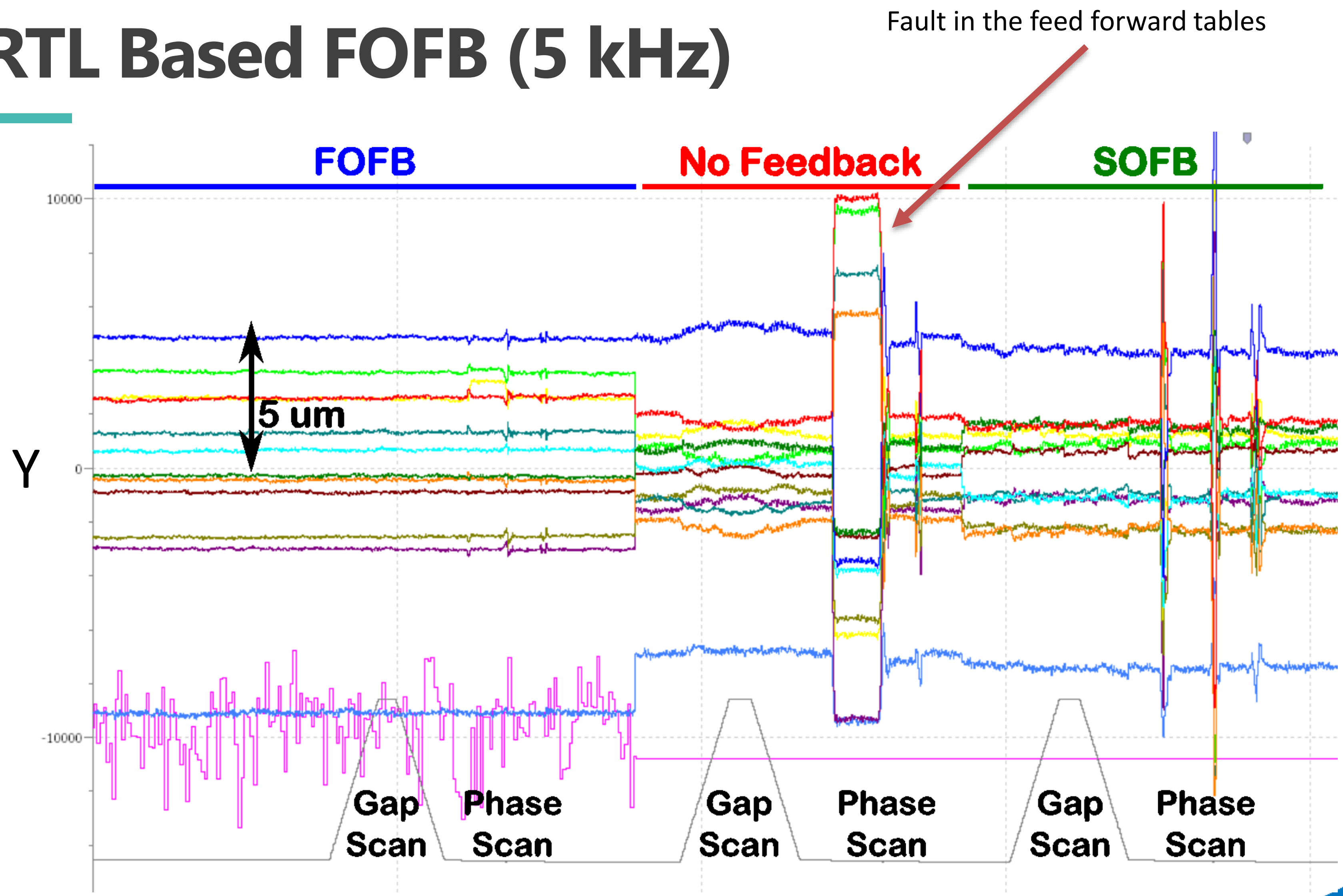




RTL Based FOFB (5 kHz)

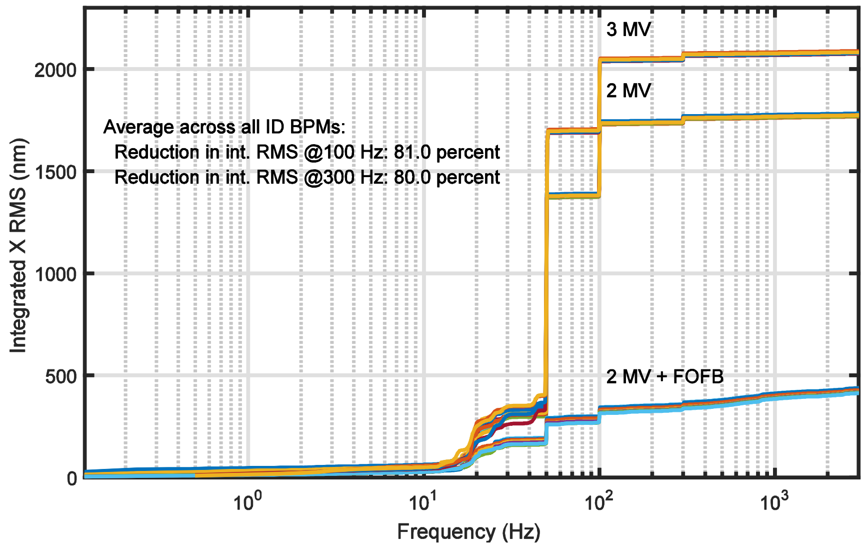


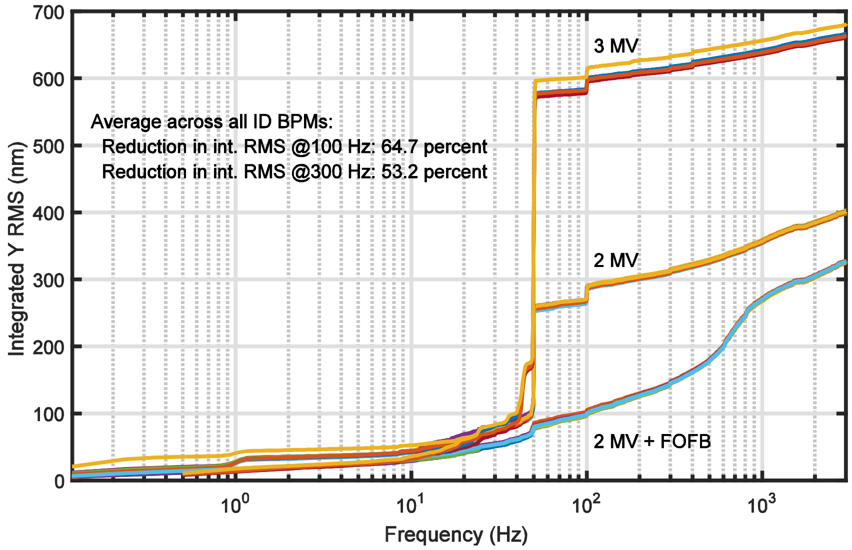
RTL Based FOFB (5 kHz)



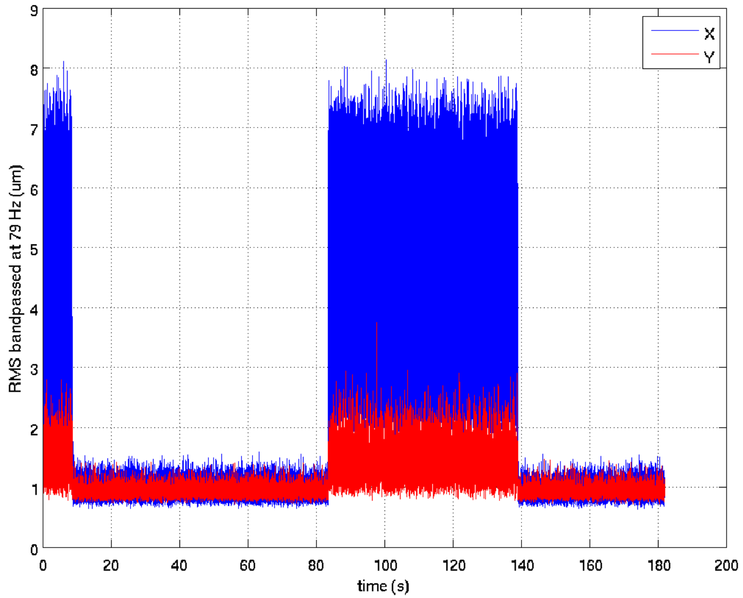
RTL Based FOFB (5 kHz)

Running with 3 out of 4 cavities for normal operation showed to improve beam stability. One particular RF cavity has a significant contribution to the 50 Hz perturbation

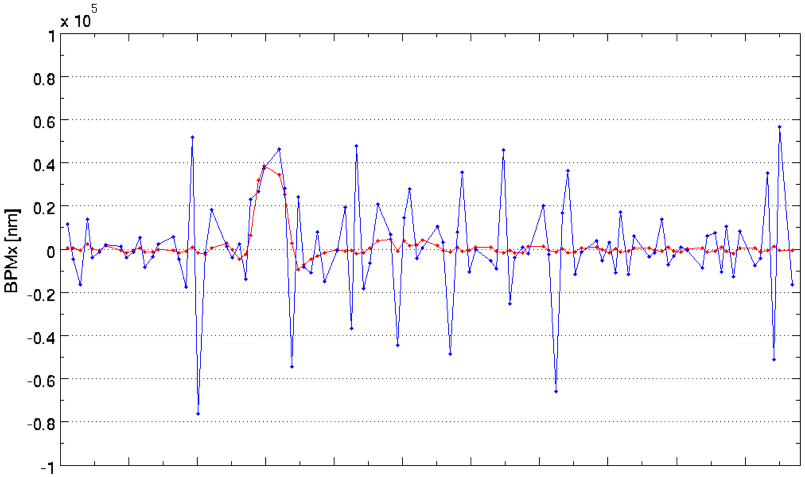




RTL Based FOFB (5 kHz)



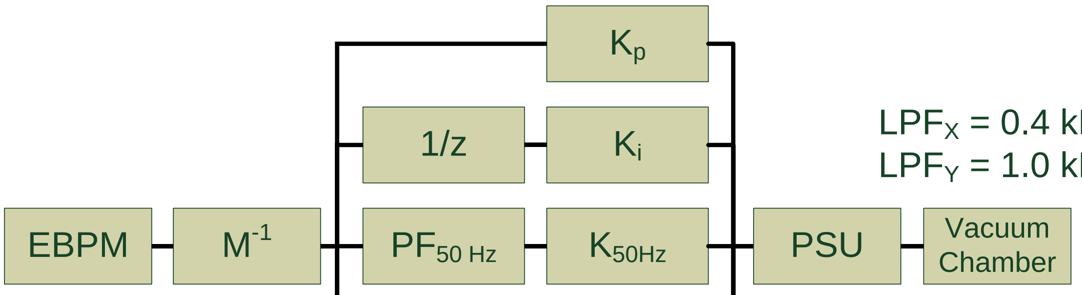
Turning on and off the loops are "clean" and synchronised between all the realtime **PC**s as its application is synchronised to the packet counter or packet sum when there is a beam dump.



Orbit bumps by changing the reference orbit.

ANSTO

Feedback Controller – PI + Harm. Supp.

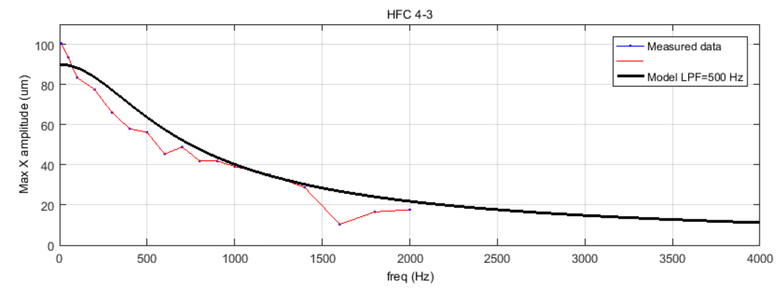


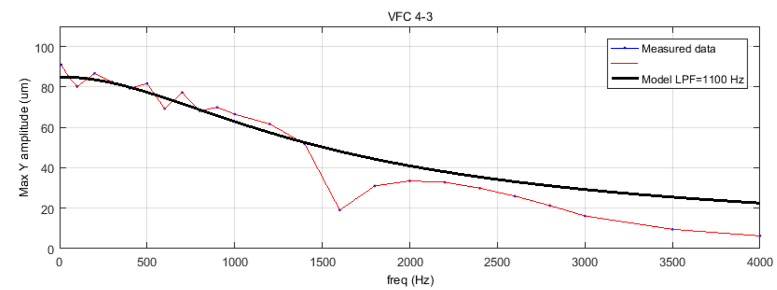
$LPF_X = 0.4$ kHz
$LPF_Y = 1.0$ kHz

$\Delta t_{EBPM} = 70\ \mu s$
$LPF_{EBPM} = 2$ kHz

$\Delta t_{PSU} = 50\ \mu S$
$LPF_{PSU} = 6.0$ kHz

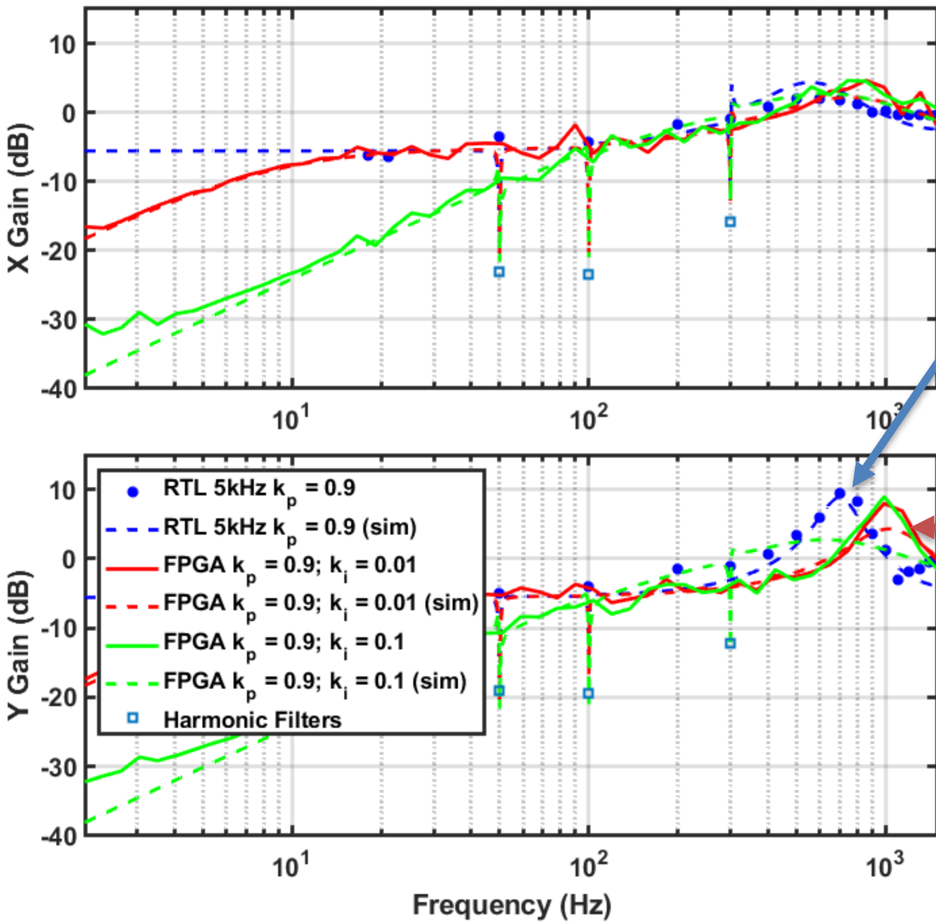
$\Delta t_{proc} \approx$?



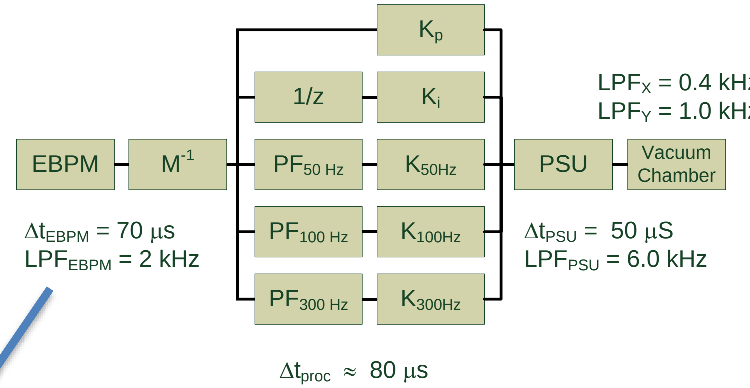


ANSTO

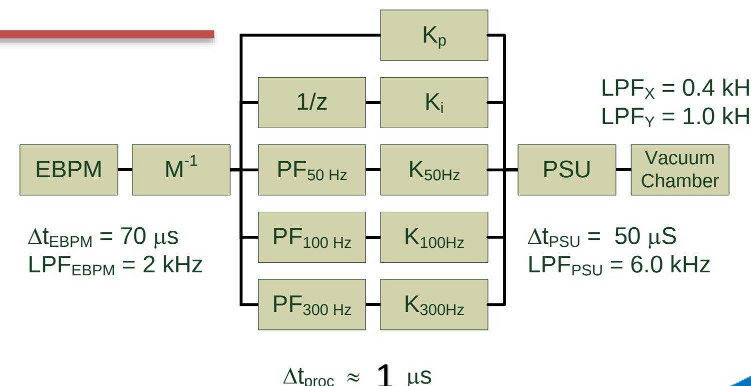
RTL Based FOFB (5 kHz)



RTL @ 5 kHz



FOFB @ 10 kHz



Soft vs. Hard Realtime

RT Linux PREEMPT	FPGA
BW = 170 Hz (H) / 150 Hz (V)	BW = 250 Hz (H) / 450 Hz (V)
Processing delay 80 us	Processing delay 1 us
In-expensive hardware	Expensive hardware and software
Rapid development cycle	Longer development cycle
Engineering resources more common	Specialist engineering resource
High reliability	High reliability

- Difference is usually characterised by jitter in the calculation times.
- For feedback the minimum delay is more important.

Conclusion

- Fast Orbit Feedback processing platform using COTS PCs is a viable option.
- Synchronisation and clocking comes from the system that aggregates all BPM positions.
- Designed properly 20 kHz cycle rates is possible.
  - 2 or more cores
  - IRQ management
  - Power supply interface (Ethernet?)
  - Real software engineer….

- Is hard real-time necessary? …No

THANK YOU

- Controls Group
  - Andrew Starritt, Terry Cornall, Emmanuel Vettoor, Adam Michalczyk, Simin Chen,
- Opreators
  - Joel Trewhella, Rod King, Peter Jones, Jonathan DeBooy, Louise Hearder, Nicolas Rae, Cam Rodda, Simon Cunningham, Madeline Chalmers
- Arrayware
  - Brett Dickson
- LNLS
  - Daniel Tavares, Sergio Marques
- Diamond
  - Michael Abbott (FA Archiver)

The 10th International Particle Accelerator Conference invites you to

Melbourne, Australia
May 19 -24, 2019

IPAC19

With over 1000 delegates and 70 industry exhibits this is a unique opportunity to network with, learn from and meet a wide range of decision makers, opinion leaders, buyers and new kids on the block.

At IPAC'19, you will have the opportunity to meet and interact with accelerator scientists, engineers, students, and vendors while experiencing the world's most liveable city.

https://ipac19.org

ANSTO

HOW LONG DOES IT TAKE TO GET TO MELBOURNE, AUSTRALIA?

WEST COAST USA

FLIGHT DURATION EQUIVALENT TO 6 ENTHRALLING MOVIES

ASIA

FLIGHT DURATION EQUIVALENT TO 3 CAPTIVATING MOVIES

EAST COAST USA

FLIGHT DURATION EQUIVALENT TO 7 RIVETING MOVIES

EUROPE

FLIGHT DURATION EQUIVALENT TO 7 INTRIGUING MOVIES