Architectural review of the LHC Orbit and Tune Feedback Systems

Purpose

To analyze the architecture of the LHC orbit, tune, energy and chromaticity feedback systems, with a view to consolidating their implementation for post LS1 running

Reviewers

– Mike Lamont (BE-OP) – Chair
– Javier Serrano (BE-CO)
– Jakub Wozniak (BE-CO)
– Stephane Deghaye (BE-CO)
– Quentin King (TE-EPC)

(Acknowledgements to Ralph Steinhagen for his comments on this document – some of which are included in the final version.)

Charge

The aim of this review is to analyze the architecture of the LHC orbit, tune, energy and chromaticity feedback systems, with a view to consolidating their implementation for post LS1 running. In this context the reviewers have the charge of:
– Identify the limitations in the current implementation
– Identify the areas where improvements must be made
– Propose a revised architecture for post LS1 running

Introduction

The LHC beam-based feedback systems are nested together by design, and were conceived to measure and control tune, coupling, chromaticity, energy, and orbit.

The system can be divided into three main domains:

1. The Orbit Feedback Controller (OFC). This is responsible for real-time data acquisition from instrumentation and power converters, data quality checks, and the implementation of the real-time feedback loops. Actuation is via the real-time channel of the function generation controllers (FGC) of the power converter and RF systems.
2. The Orbit Feedback Service Unit (OFSU). This acts a gateway level agent responsible for passing control and configuration to the OFC, and passing orbit data to the high level.
3. The user level. The RT data acquired by the OFC is highly solicited by users (mainly Java based) who typically use CMW publish/subscribe functionality. Control and configuration input also comes from the high level (for example from the LHC sequencer).

The 3-tier system (OFC, OFSU, Java) has evolved significantly since initial commissioning. This has resulted in a lot of validation code to check the data passed between them.

The 3-tier system was motivated by real-time and reliability constraints, and minimization of IO bottlenecks – mainly CPU and network. CPU considerations are alleviated with the rise of improved CPUs; network limitations remain the same.

The review was called because a significant numbers of unnecessary LHC dumps can be directly attributed to the system in 2012.

OFC

Architecture

The basic architecture is a central controller communicating with many front-ends/gateways over gigabit Ethernet. (now: 68+50; post-LS1: 68+50+ ~80 (DOROS)). UDP is used for data transmission and appears to be the right choice.

Data concentration on the OFC (accumulation period of 10 ms) is intensive and appears susceptible to misbehaving FESA/FECs. There are issues with technical network latency and there are common UDP transmission errors. Better network diagnostics are clearly needed.

Communication with the OFSU is via either Ethernet or TCP. Essentially measurements pushed over Ethernet; control flow over TCP. Heavy data transfer rates are involved.

Actuation of corrector magnets and RF frequency is via RT control functionality of FGCs used by power converters and RF. The values of the actuators are also read back by the OFC at a high frequency.

The design at this level is a nested approach offering tune, chromaticity, coupling, orbit, radial loop and energy feedback. Up to now only tune, orbit, energy feedbacks are operational. The system also offers chromaticity measurement via RF frequency modulation. There is coupling between the feedback loops via the RF frequency (radial loop shifts orbit via dispersion, orbit feedback has to be aware; chromaticity measurement with tune feedback ion).

There is very heavy CPU load. 80% of this is dedicated to error checking with data corruption very much a risk to the behavior of the loops (imagine correcting a bad pick-up reading). Data quality seems a reasonable concern (spike removal, step detection…). The many checks appear to have been added ad hoc over an extended period of operational experience. Rationalization might be possible.

Up to now the OFC has been conceived, implemented and exploited by a single individual. For the most part the system has performed well. Orbit stability of 50

micron has been achieved – happily the machine has more aperture that anticipated.

The bandwidth offered by the system is sufficient, although issues in the squeeze – see below – have pushed the existing implementation to the limit.

The orbit feedback correction strategy is based on SVD. ROOT libraries used for computation.

Some key functionality is still unused/missing – see below.

A peer code review should help to locate the possible problematic areas.

OFC Issues

- Communicating with OFSU – heavy data rates – is all this data transfer really necessary? This is mostly 25 Hz measurement data with some redundancy. *A review of the data transferred and the protocol used must be foreseen, as the presented data rate seems unreasonable.*
- Necessity of the nested approach – conceived before operational use – what are the requirements now?
- Does the loop time match power converter RST periods? *To be checked.* Does the loop time take into account time constant of magnets? *To be checked.*
- Data concentration in same thread? Because all the other tasks cannot commence without the concentration being complete.
- RefOpticsMagic? Ad hoc 'hack' (~100 lines of code) implementing the response matrix recomputation/checks. Added in a time when there was the need for changing matrices but when there weren't resources available to have a proper implementation in the OFSU.
- Core 3 – what's going on? "Safety net" – don't trust the matrix, erroneous BPMs etc – could this functionality be moved out? Can the BPM system do some of the sanity checking? Old BPM system was deployed on poorly performing PowerPCs. Now (MEN-A planned), *some checks should be shifted there.*
- There are in the region of 600 commands (many simple). This propagates to OFSU and causes maintainability problems.

OFC Missing functionality

- *OFC should be timing event aware and accept functions.*
- FESA on OFC is seen as a possible improvement. This would enable the OFSU to be by-passed for some control flow. The RT feedback part should still be done outside FESA.
- *Programmed optics changes* are definitely required and the passage through optics changes, in say, the squeeze should be driven by functions.
- Need to tidy up overlay/reference handling top and bottom.
- *The OFC needs to expose more properties and provide better diagnostics.*

OFC Communications

- **UDP transmission errors** – where are the data going? Bursts – where are they coming from? Switch is dropping frames – more diagnostics needed.
- All the 'deadline missed'/'communication lost'/'data missing'/'data corrupted' events should be logged and accessible in the high level GUIs.
- **QOS agreement with IT** is important – the technical network is causing the problems.
- **BPM data arrival** – what's going on?
- **Energy error reception** – hack to fix – majority voting on telegram information – should be fixed with new lib TimDT. (Suggested to keep this because it adds an important level of redundancy/reliability).
- Greedy or **bad CMW clients** on BPM front-ends. Proxies & RBAC rules to be put in place to shield the FECs. New RDA should help.

A tighter collaboration will be needed with external actors such as IT, to make sure we can have enough diagnostics information to unequivocally determine the source of a problem. Once this information is available, it should drive development activities aimed at increasing the robustness of the system.

OFC hardware architecture

Controls input should be sought. The situation is stable but has evolved under operational constraints. It should be possible to improve the situation.

- Was originally non-standard RT kernel with no support. OK now
- Proliant/RT – sledgehammer – is this the most appropriate solution? Proliants have both PCI (timing card) and PCIe slots.
- **Timing receiver on the OFC** is a favored solution. At present the OFC time ignorant and relies on OFSU to trigger changes in references & optics. This not good. Initial rationalization: OFC only periodic/constant load (↔ very good for RT behaviour), OFSU handling the 'rare' timing and settings changes and re-publication. If we cannot make a reliable OFSU/FESA server, this could be shifted to OFC. Some caveats apply...
- Are Proliant OS and FESA compatible? Yes for data handling, but not for RT.
- New multi-core environment? Speeding up matrix inversion… don't want to overly complicate things – what are the real requirements?
- Can **out of action closed orbit dipoles (COD)** be handled dynamically? **Handling bad BPMs** – always going to be an issue – again this would be good to handle dynamically. Could one imagine on-the-fly matrix re-conditioning in event of COD/BPM loss? Apparently yes, e.g. using GPUs (initial test x10 speed-up). *A nice to have feature.*

- Accumulation pile-up, operating system threads pinned? Parallelization of SVD? Memory i/o limit? Faster RAM? Improvements possible with some thought – CO should support this in standard way.
- Migration to 64-bit Linux would definitely be desirable.

OFSU

This is implemented under FESA. It is large: > 200 files, 30,000 lines of code, 27 real time routines, UDP workers, optics related workers, others… Its maintainability and operability are clearly under question – while recognizing that it has done an important job thus far. It is evolved code including monkey code to deal with passing data in and out of shared memory.

The general feeling was that this has evolved into a monster. *Serious rationalization is required. The publishing responsibility should be split out to a separate agent. Control functionality should be deployed in its own container.*

Up to date requirements should be established and this part of the system rewritten.

OFSU issues

- **FESA class stability issues**: memory leaks/corruption, synchronization problems. We note that tools are available to make automatic checks for memory leaks. Move to FESA 3 with its future RT diagnostics should help but propagation of this information higher up is nevertheless required.
- 3rd party libraries are in use.
- **25 Hz UDP feed from OFC** plus TCP at 10 Mbits/s with the potential to block
- Feeding out **to 30/40 clients over CMW** – put in place a proxy? OFSU was supposed to be the 'proxy' in a time when there weren't any CMW proxies.
- Is the **number of fields** manageable? All data structure flattened, poor encapsulation. Again can we split critical services and publishing?
- **Thread safety**? Known limitation of FESA 2.10 where server part and RT part step on each other. Classic approach based on RT priorities is insufficient in a multi-core environment.
- FESA SHM – custom data objects.
- What's going on in the TCP stream? Claimed to be necessary for restart capabilities in case of OFC crash – to be checked. A move to FESA 3 where settings persistence is provided will help here.
- OFSU is responsible for data synchronization and logging.
- Getting optics from server via polling? Really?
- Need to move **optics handling functionality** up to the java server level. *Details to be established.* Should move the preparation of the optics data needed for the matrix calculation into the Java layer and before each run, send to the OFC all the optics data it will need. The BPM/COD power

converter availability mask can be applied by the OFC before computing a new matrix on receipt of the relevant timing event.
- Making the OFSU less critical by giving timing event reception capability directly to OFC should improve the situation further.

Operational considerations/requirements

Requirements have clearly evolved since the first implementation. This is really not surprising; the LHC has come a long way operationally since initial commissioning.

- **BPM management** (adding/removing good/dodgy BPMs) is less than perfect and tools/procedures should be tightened up.
- **Automatic optics changes** were never fully commissioned/used. This is a must for post LS1 (solution used in run 1 was a compromise and not ideal).
- **Reference orbit management** is critical (change of x-angle, separation bumps, and effects of optics changes). Reference orbit overlays were used in run 1.
- **Overlays:** Why are these not functions in OFC? Apparently they are. Question is how to manage and update >2200 x_ref(t) coherently (many more BPM channels than power converters).
- Overlays used in run 1 were computed from the model. Perhaps the overlays used should be measured otherwise we're missing the BPM response.
- Possibility of lengthening squeeze to give OFB more of a chance. Approaching the stability limit with higher CL BW in present scheme. Too many optics were taken out of the squeeze during generation– speed is not everything, sometimes quality matters.
- **Linear interpolation between overlays in squeeze** – this should be adapted to following the parabolic-linear-parabolic scheme used by LSA. Squeeze spikes were reproducible and correctible with high enough bandwidth; but try and avoid the spikes in the first place.
- Gain/bandwidth/SVD configuration management – needs to be slicker
- There is a general lack of diagnostics, particular in case of errors.

Orbit correction strategy

- SVD used with configurable number of eigenvalues (400/440) established by experience – constant through fill.
- BW drops off at higher eigenvalues – time/controller philosophy
- Orbit feedback doesn't use MCBX and redundant LSS correctors at present.
- Deltap/p subtraction – this was a real mess at one point. Are we sure that this problem is mastered?

- More BPMS – better BPMs!!! Plan to add redundant BPM read-out, initially only for the BPMSWs directly around the IPs and later possibly for all LSS BPMs (~7L ↔ 7R) for priority IRs. → Marek's new DOROS acquisition system

Feed-forward in squeeze was useful in reducing 10-12 micron peak to 2 micron peak – valid for a few weeks (ground motion perhaps…). Note issue with spikey orbit in squeeze – may be addressed – see above.

Tune correction/measurement

The tune feedback system has generally worked well. The tune has proved to be a non-trivial observable due to coexistence with the ADT.

TFB/QFB issue was resolved via masked bunches. 2015 – might worry about signals with colliding beams if we need to squeeze in this configuration.

The implementation of the sanity checks split with BBQ makes sense.

The BBQ front-end is overloaded. It should be upgraded.

Testing

The lack of a test bench for offline development and debugging seems to be a big limitation and may have allowed many detectable bugs to enter operation. This has led to a lot of paranoia from everyone – operators and developers – and some quite ugly workarounds. Progress towards a better architecture will be very hard without an effective test bench - this ranks very high on the list of recommended activities.

- No real test bench exists so the software could not be very thoroughly debugged offline. As a result, many of the bugs seen in operation were tackled quickly with workarounds, rather than real solutions. This has added to the complexity of the overall system.
- Version control and release management are also issues.

We recommend the development of a test facility. This is important. Although recognizing this is a big job, it should be possible. It is an essential pre-requisite before any major changes.

2015

The system has to work at 7 TeV (at least 6.5) after LS1 in the face of risks associated with migration to 64-bit Linux, FESA3, new CMW and timing.

- 6.5 TeV: higher PC current for same kick – reduced BW – must change something or we won't get the performance we've enjoyed so far. Have to sort optics changes out to claim this back.
- 100 micron@20mHz 35micron@1Hz@4 TeV – not much margin to push at 6.5 TeV – explore alternative feedback strategy such as: Schmitt trigger; anti-windup etc.
- *Check the implications for QPS/RT interplay at 7 TeV.*
- Possible BW demands from colliding squeeze and beta* leveling.

Controls

A number of improvements/upgrades are incoming, clearly these should be exploited where appropriate.

- MEN A20: No longer network priority boost as in LynxOS; VME DMA (to be tested)
- FESA 3: Threading; Finer control of RT priority; RDA3 with better scalability & client priority; Other developer goodies (structures, …)
- OFSU: Configuration fetching (file transfer); Settings flow: foresee a redesign
- Serving external clients a CMW proxy should be envisaged as the simplest and cheapest solution. Ideally the situation should improve with the introduction of the prioritized network packages & improvements in the CMW layer.

Controls: recommended activities once the mandatory activities are covered:

- For possible consideration: merging of OFC and OFSU onto one platform – probably a new generation dual Xeon CPU Proliant. The OFSU part would remain under FESA3 and the OFC part would be a separate real-time process. Communication would be via shared memory using double buffering to solve synchronization issues. N.B. Not without issues: present solution has reasonably low overhead and advantages w.r.t. RT performance, maintainability, dependability and flexibility w.r.t. technology choices.
- Set up three new OFC/OFSU merged systems: 1. Operational, 2. Test for application developers, 3. Test for OFC developers. Set up two test bench systems so that each test OFC/OFSU can be paired with a test bench so that it can operate in simulation.
- Move the calculation of the matrix into the OFC layer only so that a floating-point accelerator could be used in the future.

Controls: desirable longer term activities:

- Investigate moving the matrix calculation to a PCIe accelerator card. Cards with over 1 TFlops (double precision) are available for about $3K (e.g. Nvidia Tesla K20X, AMD FirePro SM10000 or Intel SC5110P Xeon Phi). These offer more than 10x the performance of a Proliant Xeon core and should allow the matrix computation time to be around 1s (hopefully less).

This would open the possibility to survive the loss of a non-essential BPM or COD power converter, but at the cost of having to understand and maintain a PCIe accelerator card and its associated drivers. BE-BDI may have an interest in using accelerators for other projects so there may be some synergy possible.

The system seeks to exploit hard real-time behavior in the FECs and deterministic data transmission through the technical network. We favor the suggestion of exploring hardware acceleration schemes if it is seen as a piece of kit CO (or someone else) is willing to support in the long term. Otherwise a risk exists that the current maintainability and dependency problems are made even worse.

Consider FPGA technology as a potential alternative to GPUs for hardware acceleration. They might be less capable in terms of TFLOPS (although that depends to a large extent on the algorithm, which we're not qualified to comment on) but it's a well-known technology in BE-BI, TE-EPC, BE-CO and many other groups, so maintenance could be easier.

Manpower

BI, CO, OP provided a solution for a critical system. Ralph contributed with his expertise in RT programming; control theory; and the necessary beam physics. Importantly he has also enthusiastically supported the exploitation of the system. He's done a great job considering the demands and resources provided.

The technical choices were justified at the time they were taken, and the panel is impressed by what has been accomplished by (mostly) two people.

This is a critical system with a staffing and exploitation model based on essentially two people. There is too strong a dependency on the pair. *The consensus of the panel is that this non-technical issue is a major concern.*

Care should be put here to try to build a larger group of experienced people involved in the core activities. It would improve the understanding of the current system, its weak points & ease the knowledge transfer. This should be done definitely before any major technical decision is taken (i.e. rewrite/architecture change). Solution providers (e.g. CO) should be requested to provide the basic building blocks needed and a commitment from the project made to use them in the future.

This has to be managed properly, quickly.

- Ralph has a long list of planned improvements
- Ditto Maxim
- Both are coming to the end of fixed-term contracts.
- Where's the manpower coming from?
- Where's the succession planning?

The mid-term goal should be to build a team around the system. An effort should be made to shadow Ralph and start a concerted effort of knowledge transfer as soon as possible

Conclusions & recommendations

Summary:

- OFC appears to be essentially robust and good for purpose. Additional functionality is required. A number of issues have been identified above and should be addressed. A code review should be performed.
- The OFSU is unmaintainable in its present state and factorization is required.
- The staffing of the system is a serious issue and must be addressed urgently.

Three tiers of requirements/solutions concerning the future evolution and upgrade of the system:

- Mandatory
- Recommended
- Desirable

The review was largely technical and threw up a number of recommendations related to the hardware and system architecture. Recommended and desirable actions are shown in italics in the text above.

Key recommendations:

- Mandatory: OFC needs to provide (or expose) additional functionality (optics changes, functions, timing event aware).
- Mandatory: OFSU needs to be re-factored: publish orbit data; controls and configuration flow; optics handling.
- Mandatory: Full test environment to be established.
- Mandatory: small team to be put in place, resources and responsibilities to be clearly defined. Support in the exploitation phase must be anticipated. The OFC needs a good real-time C++ programmer at the 50%

level (at least initially). The OFSU, following factorization, could be taken by a combination of BI, OP and CO resources. Resources are also required to establish the test environment.

- Mandatory: a small team that includes Ralph should perform an OFC code review. This will aid the process of knowledge transfer.
- Mandatory: a small team should step back and perform an analysis of the requirements of the OFSU (and its interface to the OFC and the high level software). This is a prerequisite to factorization.