

ORBIT CONTROL FOR THE CANADIAN LIGHT SOURCE

R. Berg, L. Dallin and J. M. Vogt, Canadian Light Source, University of Saskatchewan, 101 Perimeter Road, Saskatoon, Saskatchewan, S7N 0X4, Canada

Abstract

The orbit control system for the Canadian Light Source storage ring is designed to provide both static global orbit correction and active correction up to 100 Hz. The system is made up of 48 button monitors (X and Y), 24 fast corrector magnets (X and Y), and 24 static correction coils in sextupole magnets (X and Y). The static orbit correction calculations are performed on a Windows Matlab® system. The data acquisition and control for static correction is accomplished via the Experimental Physics and Industrial Control System (EPICS) [1] control system. The calculations for fast correction are moved down to the Input Output Controller (IOC) to achieve greater performance. As the hardware for both the static and fast correction systems are the same with the exception of the number and type of correctors, the differences in software will be covered in this paper. Experience working with the static correction system during commissioning will be discussed first, followed by the proposed software design for the fast correction system.

SYSTEM OVERVIEW

Hardware:

The components involved in orbit correction are outlined in Fig 1. The main hardware component is the PCI/VME (SIS1100/3100) fibre link interface [2] from Struck. The fibre link enables all VME based correction components to be interfaced to a single computer.

The storage ring lattice consists of 12 identical cells [3], the correction hardware services 3 cells per Versa Module Eurocard (VME) crate. The hardware for the static correction system is identical to that of the fast correction with the exception of the number and type of correctors/power supplies used. Static correction uses all 48 of the correctors [4], 24 of which are housed in the sextupole magnets.

Sextupole corrector control is accomplished through EPICS IOC's running RTEMS [5]. The IOC's communicate to the power supplies over an RS-422 serial line which is inherently slow by design. The control interface to the high speed magnet power supplies is via a 32 bit parallel connection from the VMIC-2536 digital I/O board to a board in the bulk power supply that fans out the signals to all power supply channels on a common data bus. For the BPM's, an extensive analysis was performed [6] on the electronics [7] with the intent of quantifying the amount of noise that will be introduced into the correction system.

Additional hardware was added to the design to fan the BPM signals out to more than a single output which

provided input to the machine protection system [8]. The machine protection system is designed to trip the storage ring RF in the case of a serious orbit excursion at beam currents above 10 mA.

Software (Static Correction):

The beam position data is acquired from the ADC's via EPICS process variable device drivers. The current

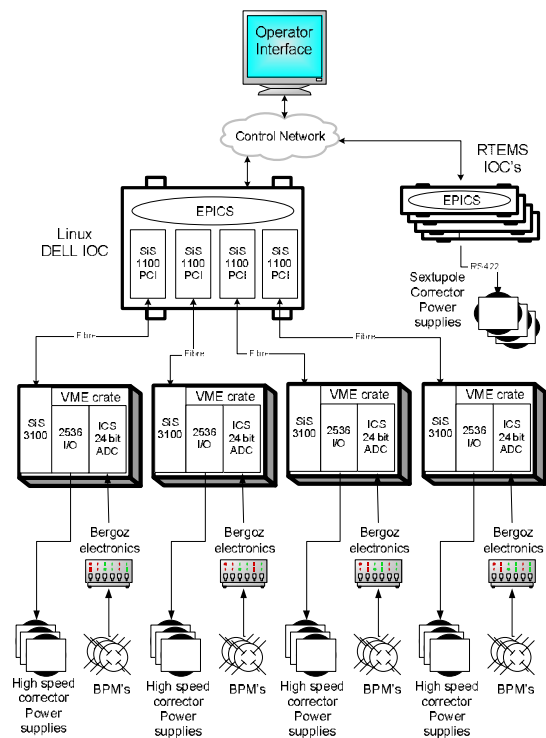


Figure 1: Hardware overview

implementation acquires only enough data to perform minimal averaging of 254 data points per beam position. The current averaging reduces the BPM noise to +/- ~10 microns. Digital filtering is being explored but a solution has not yet been finalized. The averaged position values are then placed into an EPICS record that is an array of double values representing the horizontal and vertical beam positions around the storage ring. Matlab was used to perform the control and feedback interface functions.

STATIC CORRECTION OPERATING EXPERIENCE

Booster Static Correction

A year before static corrections were attempted in the storage ring the software was developed to perform static corrections in the booster ring. The BPM and corrector

feedback and control was provided by EPICS and the operator interface was written in Borland C++Builder®. The type of hardware used was identical to the storage ring design, with the exception of the sensitivity of the Bergoz BPM electronics that were designed for a different vacuum pipe specification. All correction calculations were made on the operator interface side of the system. The ADC's connected to the BPM electronics were hardware triggered to ensure that the orbit measurements were taken after the beam had sufficiently damped following injection.

Storage Ring Static Correction

For the storage ring the operator interface was created in Matlab. To connect to the control system, the Matlab Channel Access (MCA) [9] package was used. Once connected, it was fairly straight forward to automate orbit response function measurements, scan injection kicker and steering magnets to improve the injection efficiency and perform many other control functions. Matlab was a key tool in being able to improve our abilities to diagnose and solve machine problems.

The orbit was displayed along with the set point and feedback values for given power supplies. Early on, Matlab scripts were written to: 1) acquire the current orbit, 2) retrieve the current corrector power supply set points, 3) use Singular Value Decomposition (SVD) to calculate corrections and 4) output those corrections to corrector power supplies.

As commissioning progressed, orbit corrections were performed using an interactive orbit correction application [10] written in Matlab by Jeff Corbett (SSRL). Called CORBIT (Corrected ORBIT), the application contained many advanced features that had not yet been developed at the CLS. Use of this program was helpful in achieving the goal of reduced closed orbit deviations.

FAST ORBIT CORRECTION SOFTWARE DESIGN

The design of the software for fast orbit correction was faced with the following constraints:

- 1) Complete Beam Position Monitor (BPM) and feed forward data acquisition then complete application of new calculated (SVD) corrector settings in less than 10 ms.
- 2) Retrieve data required for feed forward capability used in orbit correction calculation.
- 3) Interactively controlled from the EPICS control network and also provide feedback to that same control system at the lowest performance cost possible.

With these constraints in mind, a prototype was developed consisting of 4 main components working together (see Fig. 2) to offer:

- 1) An operator interface to the fast correction system.
- 2) Interactive control of the orbit correction parameters such as response functions, singular values, amount

of correction to apply and the rate at which corrections should be performed. This same application will provide feedback to the operator of the current state of the core correction manager as well as other feedback such as current horizontal and vertical orbit.

- 3) A Core Correction Manager (CCM) using optimised correction code to perform all data acquisition and correction calculations.
- 4) A Linux kernel ADC driver to provide interrupt based data acquisition from the ADC's for the BPM data.

The hardware for the fast correction remains the same as in the static correction with the exception of a reduced number of correctors and BPM's. The total number of corrector magnets used in the fast correction algorithm is 24 horizontal and 24 vertical. The 24 BPM's in the 12 straights will be used for fast correction. Additional hardware will be introduced to the design to achieve a feed forward capability from the insertion devices into the fast correction system. The hard and software requirements of the feed forward component of the system will be determined during testing.

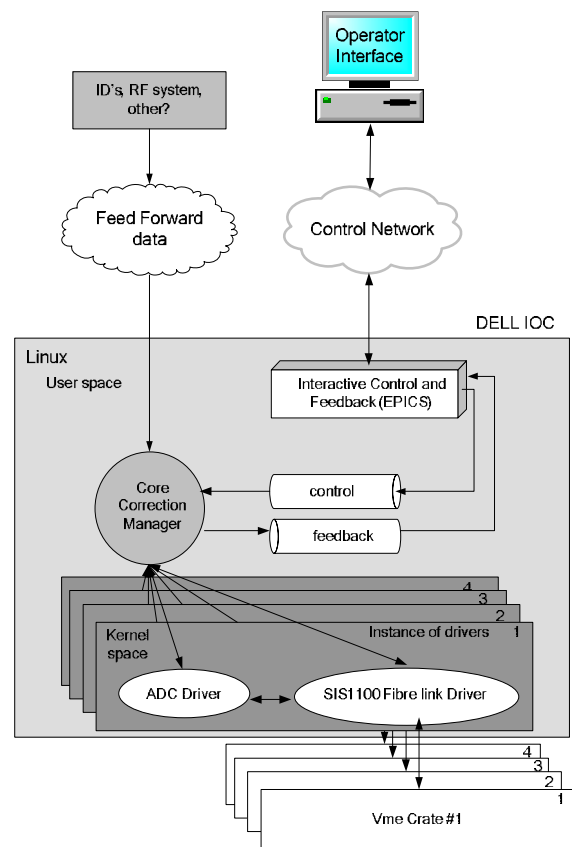


Figure 2: Fast correction software overview

Operator Interface to Fast Correction System

Initially, Matlab will be the commissioning tool used as the operator interface. Matlab was chosen for its ease of use when working with matrices with respect to the

EPICS control system. To maintain continuity with the majority of the control software at the CLS, the final implementation of the operator interface will be a compiled application or an Extensible Display Manager (EDM) screen.

Interactive Control and Feedback (EPICS)

The desire to have interactive control of the fast correction will provide the best ability to resolve unforeseen problems with the system during commissioning. The system encompasses many different hardware devices as well as many different levels of software all interacting together to accomplish a single goal. Without interactive control, it can be difficult to diagnose miscalculations or hardware errors if the hooks into the system are not left available when difficulties are encountered. The ability to introduce "known" data into the system can provide a significant amount of confidence in the system.

The control and feedback will be provided by an EPICS application that offers all correction parameters as well as all correction feedback as EPICS process variables. The EPICS application will communicate with the CCM sending control parameters and receiving feedback. The feedback from the CCM is then placed into EPICS process variables and offered to the control network.

Core Correction Manager (CCM)

The CCM will periodically monitor the EPICS interactive control application. Structures will be passed to the core application carrying instructions as well as data parameters. To perform the matrix calculations required for correction, the Math Kernel Libraries (MKL) [11] will be linked in to the application. These libraries are optimized ports of the Linear Algebra Package (LAPACK) [12] and Basic Linear Algebra Subprograms (BLAS) [13] software libraries that are offered by INTEL. The libraries are optimized for the particular Pentium processor in use. The use of the MKL libraries not only reduces in house code but also makes the execution faster by using the latest CPU enhancements.

Linux Kernel ADC Driver

The BPM data needs to be acquired at as high a rate as possible, and the amount of data acquired must be enough to satisfy the needs of any filtering algorithm. In order to process interrupts on the VME crate, a driver for the ADC will register its interrupt service routine (ISR) with the 1100 PCI driver. An interrupt will be generated by the ADC when the data buffer reaches half full. The ISR for the ADC accomplishes several tasks:

- 1) Disable interrupts on the ADC,
- 2) Send Interrupt Acknowledge cycle (*IACK) on the VME bus signalling the acknowledgment of the current interrupt.
- 3) Request a block transfer of half the FIFO size from the ADC,
- 4) Copy the buffer to the user

- 5) Re-enable interrupts on the ADC after which the driver returns to sleep until the next interrupt.

The core application opens this driver and performs a blocking read to acquire the data at the highest rate possible.

SUMMARY AND THE FUTURE

The approach to orbit control system design at the CLS has evolved along with the increasing experience. The booster ring was used as a starting point in developing a correction system that used a single CPU connected to multiple ADC's and VME crates. An operator interface was developed to control and monitor the correction progress and was tested successfully for performing static corrections in the booster. The same type of hardware was used to perform static corrections in the storage ring with the software being written in Matlab.

A prototype for fast correction has started to be developed with the current focus on filtering the BPM data to reduce noise. Further testing with digital filtering will continue in August 2004 with the goal of improving the resolution to the sub micron level.

Within the next year the CCM prototype will integrate aspects of a feed forward system from insertion devices (ID's) and RF system. The use of a real time operating system will also be investigated and a comprehensive timing analysis performed.

REFERENCES

- [1] L.R. Dalesio (LANL), M. R. Kraimer (ANL), A. J. Kozubal (LANL) "EPICS Architecture" (9/93)
- [2] R. Berg, CLS Design Note 7.2.39.19 Rev. 0 "EPICS VME Crate Monitor Software", Jan 2003
- [3] L. O. Dallin , CLS Design Note 5.2.31.4 Rev. A - "XY Orbit Correctors", May 2000
- [4] L. O. Dallin, D. Lowe, J. Swirsky, "Canadian Light Source Magnets", PAC 2003, Portland, p.2195
- [5] Real-Time Executive for Multiprocessor Systems <http://www.rtems.com>
- [6] D. Bertwistle , CLS Design Note 7.2.38.1 Rev. 0 " CLS Button Position Monitor Sensitivity Analysis", Nov 2001
- [7] I. Stavness, CLS Design Note 7.2.38.5 Rev. 0 "BPM Test Report", Sept 2003
- [8] J. M. Vogt, CLS Design Note 7.2.39.20 Rev. 0 "Fast Machine Protection for Storage Ring", April 2003
- [9] A. Terebilo, "Channel Access Client Toolbox for Matlab", ICALEPCS 2001, San Jose.
- [10] J. Corbett and A. Terebilo, "Interactive Orbit Control Program in MATLAB", PAC2001, p. 813.
- [11] Intel® Math Kernel Library, <http://developer.intel.com/software/products/mkl/>
- [12] Anderson, E. et al. Lapack Users Guide 3rd Edition, 1999, ISBN 0-89871-447-8
- [13] C. L. Lawson, et al. "Basic Linear Algebra Subprograms for FORTRAN usage", ACM Trans. Math. Soft., 5 (1979), pp. 308--323.