INTEGRATION OF ORBIT CONTROL WITH REAL-TIME FEEDBACK*

F. Lenkszus, L. Emery, R. Soliday, H. Shang, O. Singh, M. Borland[†]
Argonne National Laboratory, Argonne, Illinois 60439 USA

Abstract

The Advanced Photon Source uses two distinct control programs for orbit control and stability—a full-featured workstation-based program for orbit control (sddscontrollaw) and an EPICS-based system, the Real-Time Fast Feedback System (RTFS), to reduce orbit motion. The sddscontrollaw program has been ported to EPICS and moved from the UNIX environment to an EPICS IOC attached to the RTFS. This EPICS-based program uses the RTFS's reflective memory to gather beam position information and write corrector power supplies, thus avoiding variable network latencies. This allows the orbit control to run at a correction rate 50 times that of the workstation implementation, which virtually eliminates orbit motion caused by insertion device gap changes. Issues raised by the integration of orbit control into the real-time feedback system and performance improvements are discussed.

INTRODUCTION

The Advanced Photon Source uses two feedback systems operating in parallel to correct the transverse orbit. The "slow," DC orbit correction (OC) system performs orbit control and local steering [1,2], while the real-time orbit feedback system [3] reduces faster orbit motion. The real-time feedback (RTFS) system is AC coupled and operates at an iteration rate of 1534 Hz, while the DC system when operated on a UNIX workstation has operated at iteration intervals of 0.5 to 4.0 seconds.

The workstation-based control-law software has been ported to the real-time operating system, VxWorks, used by the APS control system EPICS IOCs. A new IOC, the datapool, has been added to the RTFS reflective memory network. The datapool IOC has access to all beam position monitor (BPM) position data and all corrector set points through the reflective memory connection. Thus the datapool provides a dedicated platform on which to run orbit correction without local area network involvement except to set up and monitor operation. The result is that DC orbit correction now runs 50 times faster with a very deterministic iteration interval.

EVOLUTION OF THE DATAPOOL

Prior to the addition of the datapool IOC, DC orbit correction ran at iteration intervals of 2.5 to 4 seconds. The sddscontrollaw program was required to fetch BPM position data from 43 BPM and RTFS IOCs and write corrector values to 20 power supply IOCs. The time required to fetch data, compute corrections, and write correctors for a single plane was measured to be 800 milliseconds with a variability of ±500 milliseconds [4]. Variable network latencies due to the large number of IOCs involved in the process and the uncertain process scheduling in the UNIX workstation environment were probably large contributors to the long iteration time with such a large variability.

Initially, while the port of sddscontrollaw to VxWorks was in process, the datapool IOC was used as a single point of focus for position data for the workstation-based OC; i.e, all BPM position data were transferred as vectors between the workstation and the datapool. Corrector set points were still sent as scalars to power supply IOCs. This permitted the OC iteration rate to be increased to 2 Hz, but a large variability in time to complete each iteration was still observed.

The VxWorks port of the control-law software allows both planes to be run on the datapool IOC simultaneously at 20 Hz. This is a 50-fold increase over the iteration rate in use prior to the datapool IOC.

THE HARDWARE

The datapool is an EPICS IOC based on a Motorola 366-MHz, MVME2700 processor. This IOC also contains a Pentek 50-MHz model 4284 DSP card [5]. A VMIC 5588 reflective memory card [6] provides the interface to the fast-feedback system's network and thus provides access to all BPM position vectors and corrector set-point vectors. Reflective memories are connected to a 29.5-MB/s fiber-optic ring. An in-house card, the feedback system interface card, receives the RTFS sample clock, which is currently set to 1534 Hz. A block diagram is shown in Fig. 1.

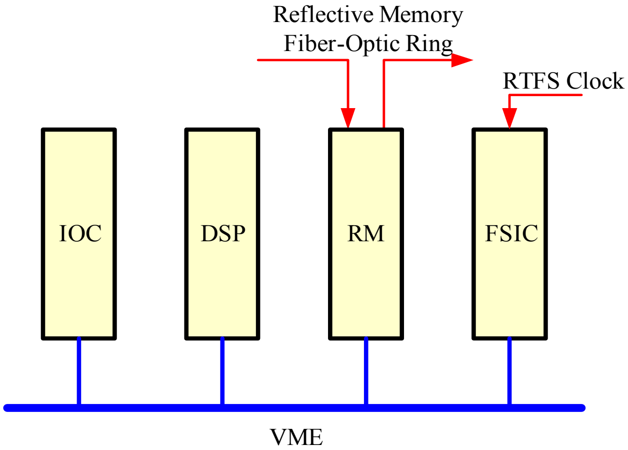

Figure 1: Datapool IOC block diagram showing the IOC and DSP processors, reflective memory (RM), and feedback system interface (FSIC).

*Work supported by U.S. Department of Energy, Office of Basic Energy Sciences, under Contract No. W-31-109-ENG-38.

[†]also at Lyncean Technologies, Palo Alto, CA 94300 USA

DATA ACQUISITION

Figure 2 shows the data flow and processing in the datapool IOC. The RTFS deposits all BPM data into its reflective memory network at its iteration rate (1534 kHz). This data consists of x and y positions from 320 monopulse rf BPMs, 70 narrowband rf BPMs, and 70 insertion device x-ray BPMs; and y positions from 70 bending magnet (BM) x-ray BPMs. Not all values have underlying physical devices yet. These 990 position values are available to any IOC attached to the RTFS reflective memory network including the datapool IOC. The datapool IOC digital signal processor (DSP) reads these position vectors at one-half of the feedback system sample rate and applies a 2-pole digital filter to each value. All computations are vectorized. The processing is done at one-half of the feedback system sample rate because the DSP card currently in use cannot read the 990 values and compute the required 990 digital filters at the full sample rate. The digital filter cutoff frequency is specified by an EPICS process variable and thus is easily changed. The DSP generates an interrupt to the IOC processor at a decimated rate determined by another EPICS process variable.

Upon each interrupt from the DSP, the IOC reads the filtered position vectors from the DSP's dual-ported RAM and deposits them into EPICS waveform records. Polynomial linearization appropriate to chamber types is applied to each vector. Offset, set-point, and gain vectors are applied using EPICS subroutine records to implement the vector processing.

At this time the sddscontrollaw does not process synchronously at the decimated data rate, rather its iteration rate is derived from the real-time operating system's 60-Hz system clock.

The control-laws compute corrector delta vectors and deposit them into corrector delta waveform records. These vector deltas are accumulated in another EPICS waveform record. The resultant 317 element horizontal and vertical corrector vectors are deposited into the RTFS's reflective memory network along with an update flag. Each RTFS DSP (20 horizontal plane and 20 vertical plane) reads the portion of the corrector vector corresponding to its span of control and writes the values to the individual corrector power supply regulators.

The datapool IOC records the last 128 values written to each of the 634 correctors. Separate time stamps are recorded for each plane. Upon beam loss, the control-laws suspend iterations and thus, the corrector history buffers contain the last 128 values for each corrector. These temporal records are available as EPICS waveform records for postmortem analysis of beam-loss events.

The datapool IOC also provides BPM set-point vectors for the RTFS. These vectors are processed by each RTFS DSP in a manner similar to the corrector set-point vectors. This feature is a key component of the orbit control/RTFS overlap compensation discussed below.

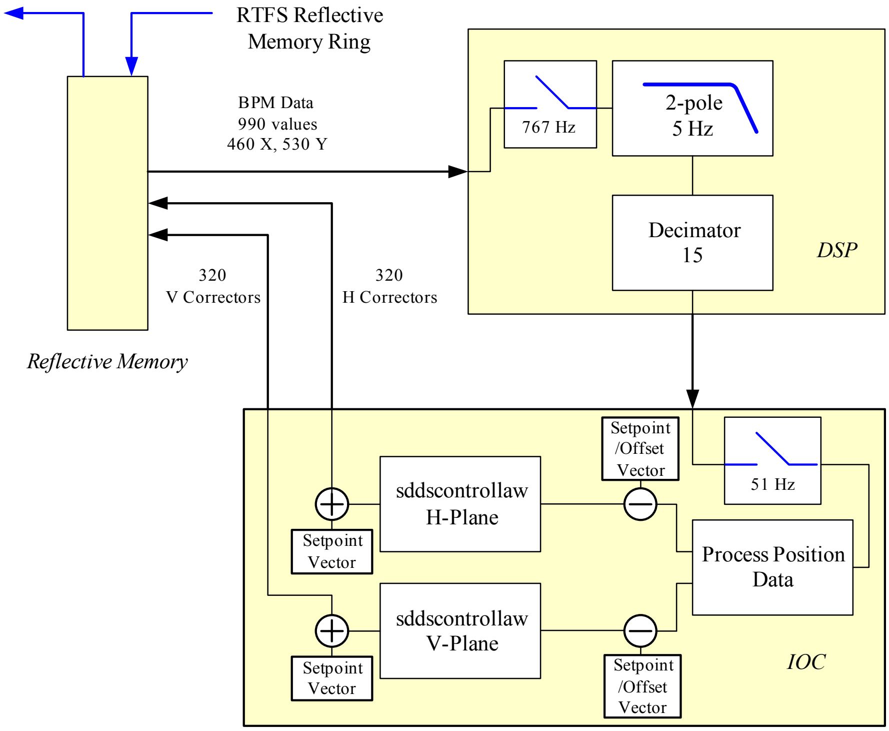

Figure 2: Data Acquisition and Processing Flow.

ORBIT CONTROL PROGRAM

Sddscontrollaw is part of the SDDS suite of tools [7]. The workstation version is used extensively at APS to do feedback with EPICS process variables. It offers configuration flexibility to accommodate various focusing lattices. It also provides "despiking" to automatically remove bad BPMs without operator intervention, limits on the maximum change per iteration, and testing of quantities for valid conditions [2]. Compensation for intensity-dependent BPM offsets is handled by a separate workstation program (sddsfeedforward). An input file containing measurement and control process variables defines a gain matrix for a simple control-law equation. By default, the sddscontrollaw tries to regulate the measurement values to zero. An additional file containing process variables to check for out-of-range conditions may also be defined.

The VxWorks version of sddscontrollaw provides most of the features of the workstation version. Testing for out-of-range conditions is handled by a separate workstation process (sddspvtest) that communicates out-of-range conditions via summary EPICS PVs. The use of an external process to test for out-of-range conditions significantly reduces the IOC load when the number of test PVs is large. The VxWorks version is controlled by a custom EPICS record type named sddsLaunch. This record includes an array field that is loaded with a text string containing sddscontrollaw configuration file names and options. One sddsLaunch record is required for each instance of sddscontrollaw. For the datapool orbit control case, two such records are required – one for each plane.

Porting sddscontrollaw to VxWorks required modifying not only the sddscontrollaw code, but also all the SDDS libraries. Several factors conspired to increase the difficulty of this port. The original sddscontrollaw made extensive use of the EPICS EZCA library, which is not available under VxWorks. Thus all SDDS channel access routines had to be rewritten in native EPICS channel access (CA) calls. Another major change to the code involved aggressively going after memory leaks, no matter how small, because VxWorks, unlike other operation systems, does not release memory after a task exits. The difficulty of this task was increased because the original code was written to exit without memory deallocation upon encountering an error. The code also contained many global variables that would confound running multiple instances of control-law. This problem was solved by placing the global variables in a large structure, which is accessible through a VxWorks task variable pointer, thus making the structure private to the task.

Additional problems arose because the original SDDS libraries were not written to be reentrant. Thus, if there was more than one instance of sddscontrollaw running on an IOC, odd problems would occasionally occur. This problem was overcome by loading a separate but identical instance of SDDS libraries for each sddscontrollaw.

Local steering is accomplished by aborting sddscontrollaw for the plane to be steered, suspending the other plane, loading a new configuration that combines both planes, and restarting sddscontrollaw. Upon completion of the steering, sddscontrollaw is again aborted and restarted with the original orbit control configuration. The suspended plane is also restarted.

We encountered random and severe problems with the initial port of sddscontrollaw to VxWorks following an abort. The IOC was left in a state that required reboot. The exact failures were not repeatable and occurred about 10% of the time. The exact cause was not identified, but we suspect it was associated with ancillary tasks spawned by channel access (CA). Since all the EPICS process variables are local to the datapool IOC, a second version of sddscontrollaw was created that used EPICS database access in place of CA. This version ran significantly faster and eliminated the failures we experienced when aborting sddscontrollaw. The datapool IOC now uses the database access version of sddscontrollaw.

The horizontal and vertical control-laws operate independently each using an instance of sddscontrollaw. Each plane uses 80 correctors (two per sector). This allows compensation for perturbations from individual insertion device gap changes. The vertical plane uses 80 rf BPMs, including all the narrowband BPMs, and all the available BM x-ray BPMs. Thus, those sectors with BM sources use four BPMs. Excluded are rf BPMs affected by a vacuum chamber higher-order mode [8] that gives a strong dependence on small bunch pattern changes.

All available rf and BM BPMs are used by the horizontal-plane sddscontrollaw. The vacuum chamber higher-order mode mentioned above doesn't affect the horizontal position readbacks.

We are in the process of including ID x-ray BPMs. These BPMs require offset feedforwarding to compensate for effects caused by ID gap changes. At the time of this writing, one pair of ID x-ray BPMs is included.

OPERATOR INTERFACE

The graphical user interfaces (GUIs) developed for orbit control used two fundamental principles that have governed all GUIs developed at APS, which are, sadly, not widely followed elsewhere but are worth repeating here: 1) Split up the work into smaller GUI applications, which may be usable for other applications. 2) Allow the GUI application to be configurable for different modes.

The operator interface for orbit control has been split into three GUIs: Modification of a database of "good" BPMs and correctors for OC, creation of corrector and BPM configuration files, and orbit control.

The GUI for maintaining a database of "good" BPMs and correctors is configurable for either "corrector" or "BPM" mode. This database also includes fields for whether or not a BPM or corrector exists, is valid for logging, is available for DC OC, is available for RTFB, etc. The database contains about 20 fields for BPMs and six fields for correctors, many of which are useful for applications other than OC.

The files related to corrector and BPM configurations for the DC OC and the RTFB are handled by a GUI configured by a base directory unique to the DC and RTFB correction systems. The datapool and workstation configurations saved for DC OC are distinguished by parameter data saved at the time of the creation of the configuration.

To simplify operations, the same GUI is used to operate DC OC in either datapool or workstation mode. A "START" button executes all the necessary setup commands and launches an sddscontrollaw process in either a workstation or datapool as determined from the parameter data read from the entered configuration name.

The setup commands that are common for both modes of OC consist of aborting any previous instance of sddscontrollaw, asserting the correct operation mode of the correctors, setting up BPM averaging, and writing status to several "In Use" corrector and BPM EPICS process variables.

In datapool mode, there are additional steps to initialize vector PVs for corrector set points, BPM offsets and set points, and RTFB BPM set points. Differences between the vector PVs and the scalar-equivalent PVs outside of the datapool IOC are calculated and significant differences are reported.

INTEGRATION WITH REAL-TIME FEEDBACK

Increasing the OC iteration rate from 0.4 Hz (2.5-second iteration interval) to 20 Hz increases its correction bandwidth from 0.025 Hz to 1.3 Hz. Since the AC coupled RTFS's correction bandwidth extends down to approximately 0.02 Hz, the range where the two systems overlap is significantly increased. If no precautions are taken, when OC and RTFS are run simultaneously they will tend to "fight" each other within their shared overlap band. The lower cutoff of RTFS could be increased to reduce the overlap. Thus one is faced with trading off overlap and the resulting "fighting" with no overlap and the resultant dead band where no correction occurs. This issue is of increased importance with the higher bandwidth OC, since insertion device gap changes near minimum gap contribute to significant beam perturbations below 1 Hz.

An overlap compensation scheme has been developed [9] and implemented that allows the two systems to overlap and operate with reduced fighting in the overlap band. The scheme is based on DC OC applying feed-forward to subtract its anticipated orbit perturbations from RTFS BPM set points. This effectively "blinds" RTFS to DC OC orbit changes, thus allowing the two systems to overlap without fighting and avoid a dead band where no orbit correction occurs.

RESULTS

The measured total IOC processor load running both orbit control planes was measured as 13%, 25%, 39%, 52%, and 91% at iteration rates of 0, 10, 20, 30, and 60 Hz. Currently both planes run at 20 Hz.

X-ray-beamline insertion device (ID) gap changes near minimum gap cause significant beam perturbations below 1 Hz. Increasing the bandwidth of the DC orbit correction combined with OC/RTFS overlap compensation reduces the perturbations from these moving ID gaps. Figure 3 shows the effect of ID gap changes for OC running on the datapool IOC at iteration intervals of 2.2, 0.8, and 0.1 seconds. It is expected that OC running at 20 Hz will reduce these perturbations further.

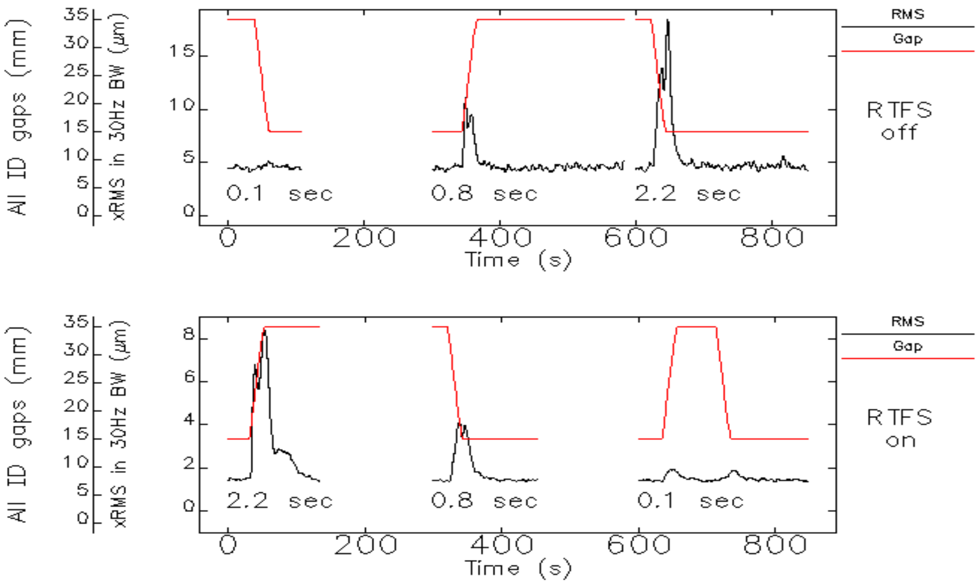

Figure 3: Reduction in orbit motion due to ID gap changes for orbit control iteration periods of 2.2, 0.8, and 0.1 seconds.

PLANNED IMPROVEMENTS

Presently the orbit control programs sample the BPM vectors asynchronously at a rate derived from the VxWorks 60-Hz system clock. We plan to add a mechanism to facilitate synchronizing sddscontrollaw iterations with the decimated BPM sample rate.

As mentioned previously, the IOC processor presently used is a 366-MHz Motorola MVME2700. We have in house a 450-MHz Motorola MVME5100 that we plan to install, and anticipate this will allow OC to run at even higher iteration rates.

CONCLUSION

The orbit control code used at APS on workstations has been ported to VxWorks and now runs on an IOC processor. This has allowed orbit control to run at 20 Hz, which is 50 times the rate used prior to addition of the datapool IOC. A primary benefit has been a significant reduction in orbit perturbations due to insertion device gap changes near minimum gap.

REFERENCES

[1] G. Decker, "Orbit Control at the Advanced Photon Source," DIPAC 2001 Proceedings, ESRF, pp. 177-179 (2001).

[2] L. Emery, M. Borland, "Advances in Orbit Drift Correction in the Advanced Photon Source Storage Ring," Proc. of the 1997 Particle Accelerator Conference, Vancouver, BC, pp. 742-744 (1998).

[3] J. Carwardine, F. Lenkszus, "Real-Time Orbit Feedback at the APS," AIP Conference Proceedings 451, pp. 125-144 (1998).

[4] Louis Emery, APS, private communication.

[5] Pentek, Inc., Upper Saddle River, NJ 07458, www.pentek.com.

[6] VMIC, Inc., Huntsville, AL 35803, www.vmic.com.

[7] M. Borland, "Applications Toolkit for Accelerator Control and Analysis," Proc. of the 1997 Particle Accelerator Conference, Vancouver, BC, pp. 2487-2489 (1998).

[8] X. Sun, G. Decker, "Eigenmodes in Two Simplified Chamber Structures Studied for Spurious Microwaves in the APS Storage Ring Beam Chamber," APS Light Source Note LS-299, April 30, 2003.

[9] C. Schwartz, L. Emery, "Compensating the Frequency Deadband of the APS Real-Time and DC Transverse Orbit Correction Systems," Proceedings of the 2001 Particle Accelerator Conference, Chicago, IL, pp. 1234-1236 (2001).