Advanced Photon Source
High-Level Orbit Correction Software

L. Emery
December 5th, 2002
Development Philosophy

• Divide the work in several single-function GUI applications and IOC software.
  – Each part of system can be debugged more easily.

• Usually each application started out simply, then increased in complexity, as needed.

• Applications share information (data and configuration) through files in SDDS format.
Development Philosophy

- GUI written in tcl/tk using widget libraries with consistent look and feel.
- Need to make software structure general to prevent extensive rewrites when a complication arises.
- This talk emphasizes the software components needed for orbit control rather than the particular implementations of the software.
Application Types

- Management of data related to orbit.
- Configuration management, i.e. to easily handle frequent changes in device availability.

- Actions:
  - Execution of orbit correction
  - Testing and control of other control system quantities
Orbit Data Processing in IOC

- Offset: beam-based measurement.
- Setpoint: where Users' want the beam.
- Offset and setpoint interchangeable for error, but good to maintain accurate database to keep correct physical interpretation.

Raw readback

Adjusted = Raw - Offset

Error = Adjusted - Setpoint = Raw - Setpoint - Offset
Orbit Data Processing

- Available averaging of bpm data in bpm iocs:
  - None (i.e. output of 7 ms averaging)
  - Software boxcar averaging of above (0.1 sec to 12.8 sec)
  - Low-pass filter
- 9 readback quantities available for each bpm.
- Boxcar used in orbit correction.
- Others used in data logging.
Orbit Data Processing

Low-level display illustrating many of the readback data related to one bpm.
Other Orbit Data Processing

• Polynomials for raw data
  - Allow several types of bi-variate polynomials.
  - Application to update setpoint and offset when converting.
  - Use one polynomial type all the time, really.

• Gains
  - Same application to update setpoint and offset.
  - Values measured with lattice model.
Save/Compare/Restore

• General application, which we use for saving setpoint, offsets (with other SR control names).

• Required for different lattices (setpoints) and for different bunche pattern (offsets)

• Ability to restore part of save-set

• Use named links as "Preferred" files for current operations.
Data Logger

- Log low-passed bpm readbacks, offsets, setpoints at specified intervals (1 minute).
- Log average bpms at faster rate for a short time when orbit glitch occurs.
- Log turn-by-turn bpm history data when a beam loss occurs.
Beam-Based Measurements

- Measure offsets with orbit and quadrupole scan. Others are measured by "straight line" interpolation method.
- Measure Xray bpm's gains with mechanical mover.
- BPM gains calculated by lattice optics fitting.
- Archive all measurements.
Orbit Display

- General application for display array of similar control names (Array Display Tool - ADT, part of Epics distribution).
- Configuration for any of the 11 bpm data possible.
- Tcl/tk wrapper for general substitution and wildcard matching.
Run Control Pvs

- General method of registering with the control system a workstation process to ensure that only one instance of that process is running at a time across the network.
- A run control PV must be pinged continuously by the associated application to ensure application is still running.
- If pinging stops, then an alarm condition on run control PV is created. Useful for detecting problems.
Offset Intensity Dependence Compensation

- Applies offset changes continuously when beam decays according to preset table.
- Separate application to make measurement of intensity dependence. Also used to find bad bpm's.
Device Status Management

- Each device (bpm or corrector) have various qualities, which affect their use. Permitted uses is saved as data.

- BPM fields: "non-existing", "ok for logging", "ok for DC OC", "ok for RTFS", "OK for steering", "Type", "Electronics type", etc

- Corrector fields: "non-existing", "ok for DC OC", "ok for RTFS".

- Data accessed by many applications for configuration of widgets and actions.
Orbit Correction Configuration

- Selection of bpms and correctors.
- Automatically hides which bpms and correctors that are bad.
- Creates configuration and correction matrix written to a named or dated directory, i.e. 2002-1205.00
- Selection of reference response matrix.
Orbit Correction Configuration

- Selection of system of bpms and correctors, i.e. regular channel access or reflective memory.
- Frequency band overlap FF matrix calculated.
- Same application for both RTFS and DC OC.
Lattice Optics Selection

- Application to re-link several database directories that depend on lattice optics.
- Orbit correction and local steering files are (obviously) optics dependent.
- Other links are for knob files and other optics correction files.
Orbit Correction

- Selection of configuration from a list.
- Selection of parameters for test files and command line arguments, i.e. interval, gain, overlap compensation, despiking parameters.
- Start and abort orbit correction
- Sets up other bpm and corrector PVs as indicators useful for operations and alarm configuration.
Orbit Correction (cont'd)

- Run PV tests as separate process to reduce work by orbit correction process.
- Bpm, corrector and stored current limits for correction validity.
- Corrector range error, i.e. current changes relative to a reference file. Alarm used to prevent large orbit drifts.
Local Steering

- Procedure Execution Manager (PEM), general tcl/tk application that allows automation of complex sequence of actions.
- Local bump steering uses a PEM to setup the orbit readback for steering (i.e. sets the error everywhere to zero), do the local steering with special orbit correction, then restore the orbit readback.
- Steering setpoints changes can be entered as delta positions and angles.
Orbit Correction in Epics

- Orbit correction software ported to VxWorks (Epics).
- Waveforms created for bpms and correctors in RTFS ioc to rapidly connect to bpms and correctors through reflective memory and RTFS DSPs.
- Though setup is completely different from workstation based OC and extremely complicated, the GUIs for orbit correction are essentially unchanged.
Orbit Recovery Application

- Restore orbit at light source points for a new focusing optics or for a return to an old one.
- Series of alternating orbit correction with multiple local bumps (for dipole sources and ID sources in h and v planes) and partial recovery of savesets of bpm setpoints.
Summary of Databases

- BPM status
- Corrector status
- BPM septoints, DC offsets (i.e. general save/compare/restore application)
- History of BPM setpoint/offset restore actions
- DC Offset measurements
- Offset intensity dependence
- Gain measurements (xray bpms, rf bpms)
Summary of Databases

• Orbit correction configurations for both DC OC and RTFS.

• History of local steering actions

• Logged data:
  – Continual readbacks of relevant BPMs and correctors on long time scale (e.g. 60 seconds),
  – same as above but with much shorter time scale (1 seconds) at an orbit glitch or beam dump, and
  – Turn-by-turn bpm history before beam dump.
Summary of Databases

- Lattice optics and dependent files.
- Feedforward waveforms for circularly polarized undulator.
Summary

• This represents an important effort to organize the data, to make the configurations easy to modify when necessary, and to make improvements to beam stability.