MONITORING AND ARCHIVING OF NSLS-II BOOSTER SYNCHROTRON PARAMETERS

A. Derbenev[#], P. Cheblakov, R. Kadyrov, S. Karnaev, S. Serednyakov, E. Simonov, BINP SB RAS, Novosibirsk, Russia
M. Davidsaver, BNL, New York, USA

Abstract

When operating a multicomponent system, it is always necessary to observe the state of a whole installation as well as of its components. Tracking data is essential to perform tuning and troubleshooting, so records of a work process generally have to be kept. As any other machine, the NSLS-II booster should have an implementation of monitoring and archiving schemes as a part of the control system. Because of the booster being a facility with a cyclical operation mode, there were additional challenges when designing and developing monitoring and archiving tools. Thorough analysis of available infrastructure and current approaches to monitoring and archiving was conducted to take into account additional needs that come from booster special characteristics. A software extension for values present in the control system allowed to track the state of booster subsystems and to perform an advanced archiving with multiple warning levels. Time stamping and data collecting strategies were developed as a part of monitoring scheme in order to preserve and recover read-backs and settings as consistent data sets. This paper describes relevant solutions incorporated in the booster control system.

INTRODUCTION

NSLS-II synchrotron light source with a 3 GeV electron storage ring is located at Brookhaven National Laboratory [1]. The booster for NSLS-II has to provide full-energy beam acceleration from 200 MeV on injection to 3 GeV on extraction with an average current of 20 mA with single and multi-bunch operation modes [2]. The booster is a complex facility, effective management of it being the key to a successful adjustment and stable operation of the machine. For the interaction of operators and engineers with the installation to be the least time-consuming during both tuning and maintenance processes, the booster control system has to provide a suitable software environment and a set of tools to perform live monitoring and archiving of various system parameters. When introducing solutions, features of the machine should be considered.

GENERAL CONCEPTS

In context of the monitoring and archiving system design, there are two prime specialties of the NSLS-II Booster Synchrotron. First one is a cyclic nature of the machine workflow due to the nature of the acceleration process, and the second one is an abundance of non-scalar data amongst facility parameters (i.e. array data) related to a ramping mode of power supplies operation. The monitoring and archiving system is a part of the booster control system, which is in turn designed around the machine specifics described above.

Thus, no matter what particular monitoring and archiving scheme is to be utilized, it will be a composition of solutions applied on several software levels. These levels are, essentially, a firmware level (e.g. programs running inside front-end controllers), a driver middle-level (generally some hardware-to-PC interface) and a high-level applications (GUIs and scripts). In this model, a lower level should provide an “interface” solely to the adjacent higher level for the sake of scalability, extensibility, and maintainability.

Firmware

Of all software levels mentioned, the firmware level is the least changeable. This is due to the fact that firmware developers are not interested in, nor they should be, in adjustment to specific needs. However, it is mandatory for a firmware to cover basic needs for monitoring of underlying hardware parameters (e.g. power supply on/off status). The inability of a device to report its parameters renders its accounting in monitoring and archiving system as impossible and usually addressed as an issue.

Middle-level Software

The NSLS-II Booster control system is based on the Experimental Physics and Industrial Control System (EPICS). As a consequence, the middle-level software is represented by Input-Output Controllers (IOCs) – EPICS interfaces, either associated with a specific hardware or providing additional processing that may not be related to any real device (Soft IOCs). General solutions, such as alarm flags generation for monitoring or data processing for archiving, are implemented on the IOC level so that all higher-level applications would be able to utilize them without recreating same functionality inside.

High-level Applications

High-level applications are a set of Graphical User Interface (GUI) software and scripts that are intended to be utilized by an end-user. Algorithm behavior control, visualization of alarms, parameters display, live and archived data browsing and comparison, working with machine operation modes and data processing that can not be uniformly carried by the IOC level are typical tasks performed via high-level software. Each application is designed to meet the specific needs while being able to

[#]aderbenev@bnl.gov

ISBN 978-3-95450-139-7

take into account possible system changes, and usually covers a wide range of associated functions.

EXISTING APPROACHES

Because of monitoring and archiving schemes being the essential part of almost any control system, different approaches to these tasks were developed on other installations. However, it is usually not possible to directly apply these solutions because they may not take into account the booster specifics mentioned above or are designed for infrastructure which differs from that is available. Nevertheless, principles behind these approaches can be successfully adopted when designing custom solutions.

Monitoring

Monitoring process usually suggests that a live value is compared to a reference value with alarm being raised if the difference is too high. Value monitoring is an innate feature present in EPICS IOC database concepts. It is possible to set up an alarm mechanism through the proper configuration of an IOC. However, this does not apply to the case of waveform (array) data that makes up a significant part of the booster parameters. Since EPICS lacks built-in point-to-point waveform comparison as well as direct access to the difference between live and reference values, additional solutions like full-length waveform references should be introduced on the IOC level to achieve the desired functionality.

Essentially, saving and restoring booster parameters are closely related to the monitoring process. This is because reference values are also parameters in the monitoring scheme and should be preserved, though not directly, along with live values. Considering this, an appropriate save/restore solution should be utilized. On parameters save, settings and measurements are both saved. However, on restore only settings have to be directly restored, while both value groups (settings and measurements) should be set indirectly (i.e. value is not restored to the place from which it was saved) as reference values for monitoring purposes.

Several save/restore applications were evaluated:

- SynApps Autosave [3] – an IOC-level tool that is designed to preserve machine settings. Supports only direct restoration, can not save or restore on demand, lacks GUI and browsing capabilities.
- JLab Operations Save/Restore Tool – a high-level application based on BURT [4]. Supports only direct restoration and lacks waveform comparison.
- SNS SCORE [5] – a high-level Java application, supports only direct restoration, requires additional database setup, has site-specific design and lacks extended comparison features.
- BNL MASAR [6] – limited support of direct restoration only, lacks waveform browsing and mode comparison features.

Archiving

A replacement for the EPICS Channel Archiver, the Control System Studio Archive System [7] is an appropriate archiving solution with infrastructure and support being available at BNL. However, it is necessary to take into account specifics of the booster when archiving parameters. Additional value timestamp synchronization should be present on the IOC level to match parameter readings to the machine cycle.

APPLIED SOLUTIONS

It was discovered that the alarm support being available in EPICS is not sufficient to monitor waveform data. Also there are no save-restore tools that are capable of performing the indirect value restoration. Because of this, additional solutions were introduced for extended monitoring capabilities on the IOC level. A custom save-restore high-level application was developed to provide interface to the cycle-synchronized reference comparison scheme. Additionally, a set of high-level applications was developed for monitoring purposes. For archiving, necessary IOC-level support was implemented with high-level interface to archived data.

IOC Support

A monitor-archive algorithm shown in Fig. 1 is implemented on the IOC level to allow specific monitoring and archiving of the booster operation.

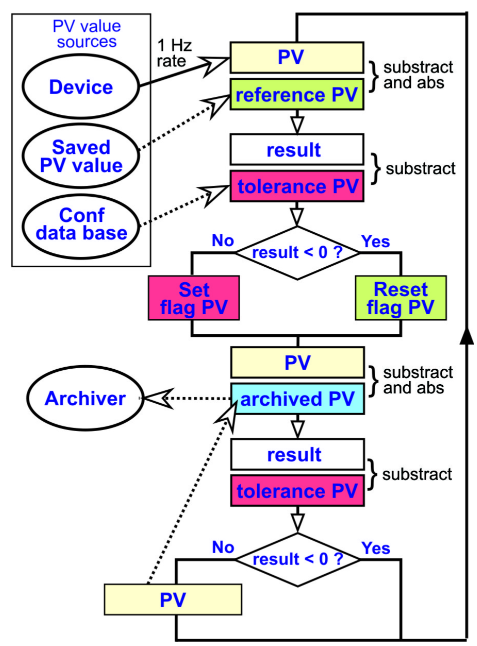

Figure 1: IOC-level monitoring support algorithm.

Presented algorithm traces the operation by comparing current settings and measurements (process variables or PVs in EPICS terms) with corresponding reference values. In a case of waveform (value array) data type comparison operations are performed in a point by point manner.

ISBN 978-3-95450-139-7

In order to understand whether or not the parameter deviation from a desired value is too big, a reference value is subtracted from a live value. If the "Diff", an absolute value of the subtraction result (or at least one point when processing array data), exceeds the corresponding preset tolerance level "Tol" then the "flag PV" is set (see Table 1 for possible flag values). "Flag PV" can be used by a high-level application or mapped to an alarm server to indicate that alarm is raised.

Table 1: Flag PV Values and Corresponding Alarm Levels

Condition	Flag Value	Alarm Level
Diff <= Tol	0	-
Tol < Diff <= Coeff*Tol	1	HILO
Diff > Coeff*Tol	2	HIHI

"Coeff", or alarm severity coefficient, is set to "10" by default and may be changed.

It is not appropriate to perform parameter archiving on each value update. Because of this, value archiving "dead-bands" are usually introduced to specify the deviation limit, below which value changes will be ignored. In the monitor part of the algorithm, this dead-band is represented by tolerance. But the live value should be dead-band-tested not only against the reference, but also against the previously archived value. This double-check comparison allows to avoid the excessive archive space consumption in case if the difference between the live value and the reference value is too high for an extended period of time. Otherwise the abnormal measurement would be archived on each booster cycle.

In order to understand whether or not the parameter deviation from a desired value should be archived, the archiving part of the algorithm is executed afterwards. The archiving procedure subtracts the "archived PV", which is the previously archived parameter value, and the live PV value, with the result being compared with the corresponding tolerance value once again. If the result is bigger than the tolerance value then the live value is written to the "archived PV" and EPICS archiver can store this value.

Additionally, two modes of monitoring are supported in the booster PS IOCs:

- "Mode 2" – reference value is a value restored from the previously saved booster operation parameters values. This monitoring mode is to be used for regular monitoring of the facility operation.
- "Mode 1" – reference value is a set value. This mode is to be used for machine performance monitoring during the parameters adjustment process. It allows to see how well settings are executed by the booster devices.
- "Mode 0" – monitoring is disabled.

An auto-adjustment flag adapter scheme is designed for solutions on the IOC level as a mediator between the monitor-archive algorithm flags and high-level applications. Sometimes, even if there is a difference between a live and a reference values and a corresponding flag is raised, it should not be treated as an alarm event by a high-level application. An example would be the automatic modification of machine settings by some correction program – in this case live values indeed differ from references, but those differences are "legitimate" and should be considered. It is possible, but is not convenient to account this logic in the monitor-archive algorithm itself. This is because it would bring specific dependencies from higher level (e.g. from the high-level correction program) that would affect all high-level applications that use the monitor-archive algorithm flags. Thus, it was decided to create a separate IOC-level solution to handle the advanced alarm flag logic. An "adapted" set of flags is generated, allowing the correction program to introduce parameter offsets without interfering with booster status monitoring.

A timestamp consistency scheme is introduced on the IOC level to consider the cyclic nature of the booster operation process. As this scheme is deployed, all settings written and all measurements read during any cycle will have the same EPICS timestamp equal to the time when this cycle is started, derived from the global timing system. This solves synchronization problems with devices processing data at different rates and network latency. Consequently, all data in the archiver database can be easily decomposed into cycles for further analysis and visualization.

High-level Applications

Parameters Auto-adjuster EPICS Channel Access based Python application is designed to negate the influence of long-term destabilizing factors like instability of the device characteristics depending on the temperature. Its purpose is to provide a stable machine behavior by modifying device settings using a feedback algorithm. The algorithm to be used for a particular device is customizable due to the diversity of the booster structure. The Auto-adjuster application is intended to be run on background and to interact with the auto-adjustment flag adapter IOC-level application described above.

Booster Status Monitor (BSM) is an application based on Control System Studio and is designed to provide the operator with tools to track the state of the entire booster system, its elements (e.g. magnets), and elements' parameters and to allow, when necessary, a quick access to diagnostics data available. The application displays visualized status data on a screen so that the operator can receive a feedback during the facility tuning or operation process and quickly respond to any abnormal condition. By itself, the status of a single element parameter is a value that is either Good (green color, operating

ISBN 978-3-95450-139-7

normally), Warning (yellow color, decreased beam performance) or Fault (red color, device does not work properly, impeding beam operation). These states correspond to monitoring flag value of 0, 1 and 2 respectively, as listed in the monitor-archive algorithm description above. In order to comprehend the state of the booster machine as a whole at any given point in time, statuses of each channel in all elements have to be put together and checked for being in normal operating state. To do so, BSM utilizes the parallel update-on-subscription scheme to simultaneously track the state of each element, this state being represented as the aggregation of its channels' states. Thus, if any of element channels appear to be outside of their normal operation, the condition of the corresponding element is flagged as inconsistent and an applicable alarm is triggered.

Application for Save and Restore is an EPICS Channel Access based application written in Python and is designed to provide the operator tools necessary to fully or partially save a consistent (i.e. relevant to a single cycle) condition of the booster system elements and to allow, when necessary, a quick restore to a previously saved state. During the machine state save process, the application monitors the value of every tracked PV in real-time and composes a change history, which is afterwards collapsed into a consistent value set and saved as a state. A consistent state is the state where all values have the same timestamp and therefore are relevant to a single booster cycle, which is supported by a timestamp consistency scheme described above. Naturally, it is required that the process of PVs monitoring would cover several booster cycles so that all PVs have enough time (given network and processing delays) to update their values and a consistent state can be composed.

By monitoring system design, state save and state restore processes are not symmetric, which was the main issue when trying to adapt existing save/restore solutions. In the booster Application for Save and Restore, the indirect restoration, i.e. when value is not restored to the place from which it was saved, is natively supported. Thus the application fits perfectly with monitoring and archiving system design.

Additionally, saved states can be browsed and compared with each other or with a snapshot of live parameters values. Plotting and comparison tools are available for 2D array to allow easy visual perception of waveform parameters.

LiveCompare is an EPICS Channel Access based Python application and is designed to perform visualization and extended comparison of live scalar and array data parameters. Not infrequently during the machine tuning routine it is required to perform a "manual" monitoring of system parameters for in-depth analysis of the device behavior. LiveCompare allows to dynamically switch between and display any live parameter present in the control system, either of scalar or array type, and to quickly compare any two parameters of compatible type. Difference, ratio and normalized ratio may be displayed for compared channels. With LiveCompare, setting may be operatively compared with corresponding measurement for the same element, or consistency between measurements from different elements may be observed.

COCo View is an EPICS Channel Access and Channel Archiver based Python application designed to provide a user interface for the closed orbit correction algorithm that is present in the booster control system. Over the process of machine configuration it is essential to monitor the performance of this algorithm in order to have information about the details of the correction process, so that relevant problems may be discovered. COCo View is able to track and visualize relevant live or archived data such as beam orbit measurements, correctors settings, and beam-related parameters like energy and current.

CONCLUSION

A solid monitoring and archiving scheme was designed and implemented by introducing an assortment of high-level applications based on IOC level solutions. It was proven during booster control system tests that machine specifics are considered and that developed tools are sufficient to perform monitoring and archiving during the installation tuning process [8]. Presented tools, once refined and improved to meet secondary needs, will be used during the NSLS-II Booster Synchrotron commissioning.

REFERENCES

[1] F. Willeke, "Status of the NSLS-II project," IPAC'11.
[2] S. Gurov et al., "Status of NSLS-II booster," IPAC'13
[3] T.M. Mooney, "synApps: EPICS Application Software for Synchrotron Beamlines and Laboratories," PcaPAC2010.
[4] BURT: Back Up and Restore Tool, http://www.aps.anl.gov/epics/extensions/burt/
[5] J. Galambos et al., "SCORE – A Save, Compare, and Restore Application for Snapshotting Machine Settings," ICALEPCS2007.
[6] L. Yang et al., "Development Progress of NSLS-II Accelerator Physics High Level Applications," IPAC2012.
[7] Control System Studio Archive System Description, http://cs-studio.sourceforge.net/docbook/ch11.html
[8] S.E. Karnaev et al., "Status of the NSLS-II Booster Control System," these proceedings.

ISBN 978-3-95450-139-7