A Generalized Correlation Plot Package for the CEBAF Control System*

D. Wu, W. Akers, S. Schaffner, H. Shoaee, W. A. Watson, D. Wetherholt
Continuous Electron Beam Accelerator Facility, Newport News, VA 23606 USA

ABSTRACT

The Correlation Package is a general facility for data acquisition and analysis serving as an online environment for performing a wide variety of machine physics experiments and engineering diagnostics. Typical correlation experiments consist of an initial set of actions followed by stepping one or two accelerator parameters while measuring up to several hundred control system parameters. The package utilizes the CDEV [1] device API to access accelerator systems. A variety of analysis and graphics tools are included through integration with the Matlab math modeling package. A post- acquisition script capability is available to automate the data reduction process. A callable interface allows this facility to serve as the data acquisition and analysis engine for high level applications. A planned interface to archived accelerator data will allow the same analysis and graphics tools to be used for viewing and correlating history data. The object oriented design and C++ implementation details as well as the current status of the Correlation Package will be presented.

INTRODUCTION

The tasks of commissioning and optimizing the performance of an accelerator require performing many machine experiments involving the acquisition and online analysis of a large amount of data. This data is usually obtained from a diverse and disparate array of devices, instruments and processes. Each source of information or data may have its own unique acquisition characteristics or control idiosyncracies, e.g., the required settle time between changing a magnet setpoint and reading a downstream BPM, or the synchronization of the motion of a wirescanner with successive beam samplings. Additionally, most experiments involve a myriad of activities which extend beyond simple data acquisition. This includes complex instrument and experimental setup prior to data taking, monitoring of some control system parameters for conditional data acquisition (e.g. recording BPM data only if sufficient beam current is present), intermediate evaluation of the results and finally a post experiment set of activities including analysis, graphics, and data archiving.

The objective of the development of the Correlations Facility has been to provide accelerator users with an integrated environment for performing a large array of machine physics and engineering diagnostics experiments with uniform access to practically all available machine data. The generalized data acquisition capability of this package provides automatic and transparent access to new data types and sources as they become available within the control system, without the need to rebuild any components of the correlation software. Similarly, within this environment one has the flexibility to automatically vary any system parameter that can be manually adjusted via the control system.

Additionally, by designing the system around a data acquisition engine, we have provided the capability for using this engine in any high level application or automation procedure, such as emittance measurement or cavity gradient calibration, which requires controlled data acquisition and analysis.

The Correlation Package was inspired by SLAC’s Correlation Plot software[2] and provides many of its features. However, its implementation and capabilities reflect the advances in the fields of software engineering, graphics, user interface and math modeling since that pioneering package was developed.

PRINCIPLES OF OPERATION

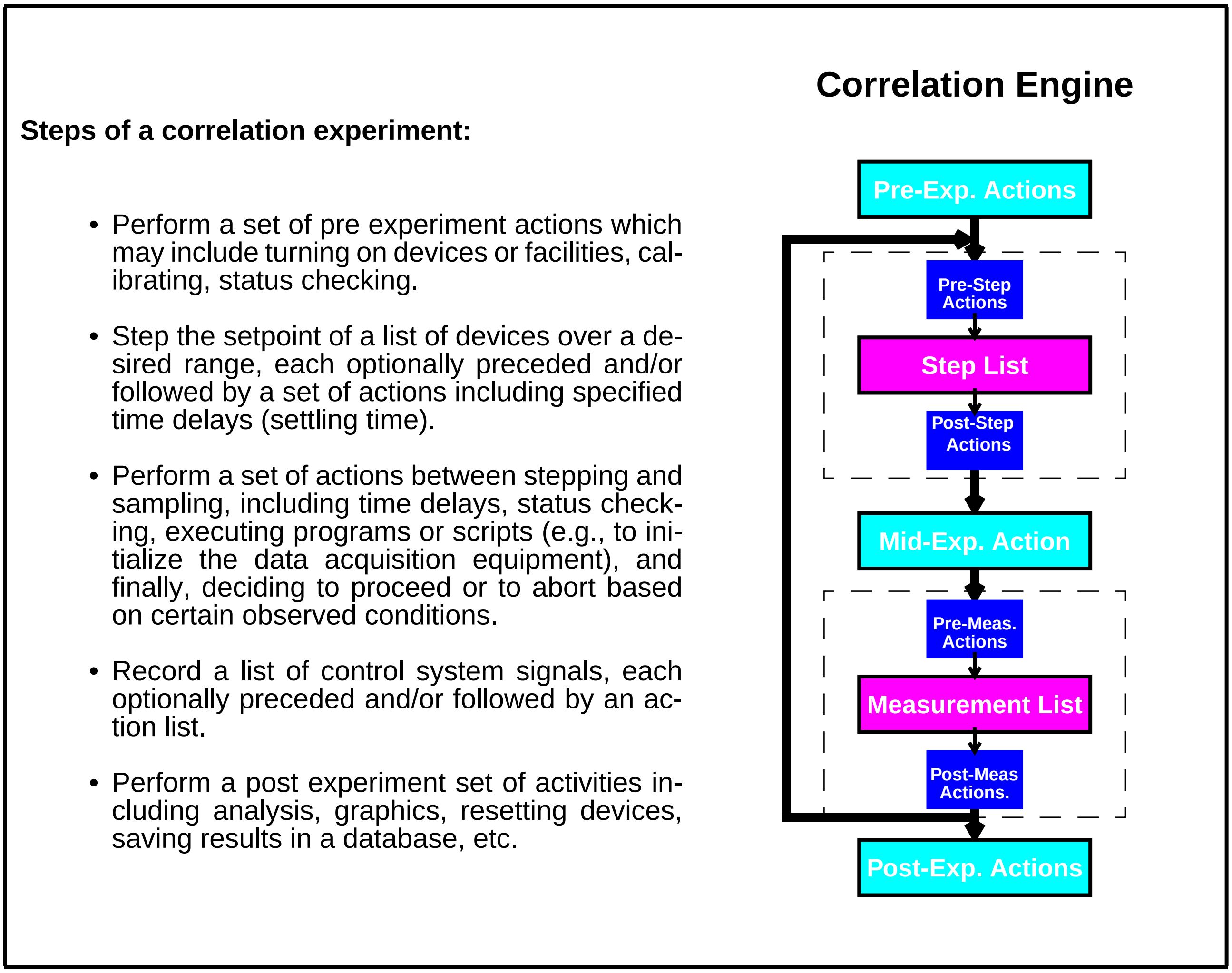
A typical accelerator physics experiment consists of several steps of the following nature:

- A *pre-experiment setup* stage which may include turning on devices or facilities, calibrating, status checking, etc.
- A *control* stage involving varying the setpoint of one or many devices over a desired range.
- An *interim* period between control and data acquisition when the experimenter may perform a set of tasks or actions including waiting for a length of time (e.g., for the transients in the control device to die out), checking the status of any control system signal, executing some stand-alone programs or scripts (e.g., to initialize or prepare the data acquisition equipment), and finally, deciding to proceed or to abort based on certain observed conditions.
- A *data acquisition* step whereby up to several hundred control system signals are sampled and recorded.
- A *post experiment set of activities* including analysis, graphics, resetting devices, saving results in a database, etc.

To accommodate these requirements for an integrated package, the correlation facility has been implemented in terms of several key concepts or module:

- A **control module** capable of stepping the setpoints of one or many control system devices, such as corrector strengths, cavity phases, or feedback setpoints.
- A **data acquisition module** for recording any available control system signal such as beam orbits from BPMs, RF parameters, feedback measurement and control parameters.
- An **action module** capable of performing a variety of tasks including synchronized and time delayed data acquisition, executing stand-alone programs or scripts, evaluating conditions based on machine and beam states for continuing or aborting an experiment, and performing many control system actions such as calibrating BPMs.
- A **graphics module** for displaying any sampled data against any sampled or stepped variable. In the near future, this module will also perform a variety of signal analysis and curve fitting functions on the sampled data.

To facilitate using the correlation package as an embedded engine in high level applications requiring controlled data acquisition, we developed a portable **Correlation Engine** encapsulating the *control*, *data acquisition* and *action* modules. Figure 1 illustrates the functional components of an experiment as performed with the Correlation Engine.



Steps of a correlation experiment:

- Perform a set of pre experiment actions which may include turning on devices or facilities, calibrating, status checking.
- Step the setpoint of a list of devices over a desired range, each optionally preceded and/or followed by a set of actions including specified time delays (settling time).
- Perform a set of actions between stepping and sampling, including time delays, status checking, executing programs or scripts (e.g., to initialize the data acquisition equipment), and finally, deciding to proceed or to abort based on certain observed conditions.
- Record a list of control system signals, each optionally preceded and/or followed by an action list.
- Perform a post experiment set of activities including analysis, graphics, resetting devices, saving results in a database, etc.

Figure 1: Anatomy of A Correlation Experiment

DATA ACQUISITION INTERFACE

XACT (X-based Accelerator Correlation Tool) is an X11/Motif based graphical user interface (Figure 2) for performing correlation experiments. It is the first application software using the Correlation Engine as an embedded facility. From this interface one may specify *step* and *sample* variables, *action* lists and other acquisition and control parameters. It is also a control panel for starting and stopping experiments as well as saving the experimental setup and the acquired data to files for later recall and analysis.

Device Selection

All signals names specified as step or sampled variables are validated against the control system. Validation can be toggled on and off through the pull-down "Option" menu. One may also use wild cards in specifying signal names in which case a list of all matching signals will be presented. You may choose one, many, or all available signals.

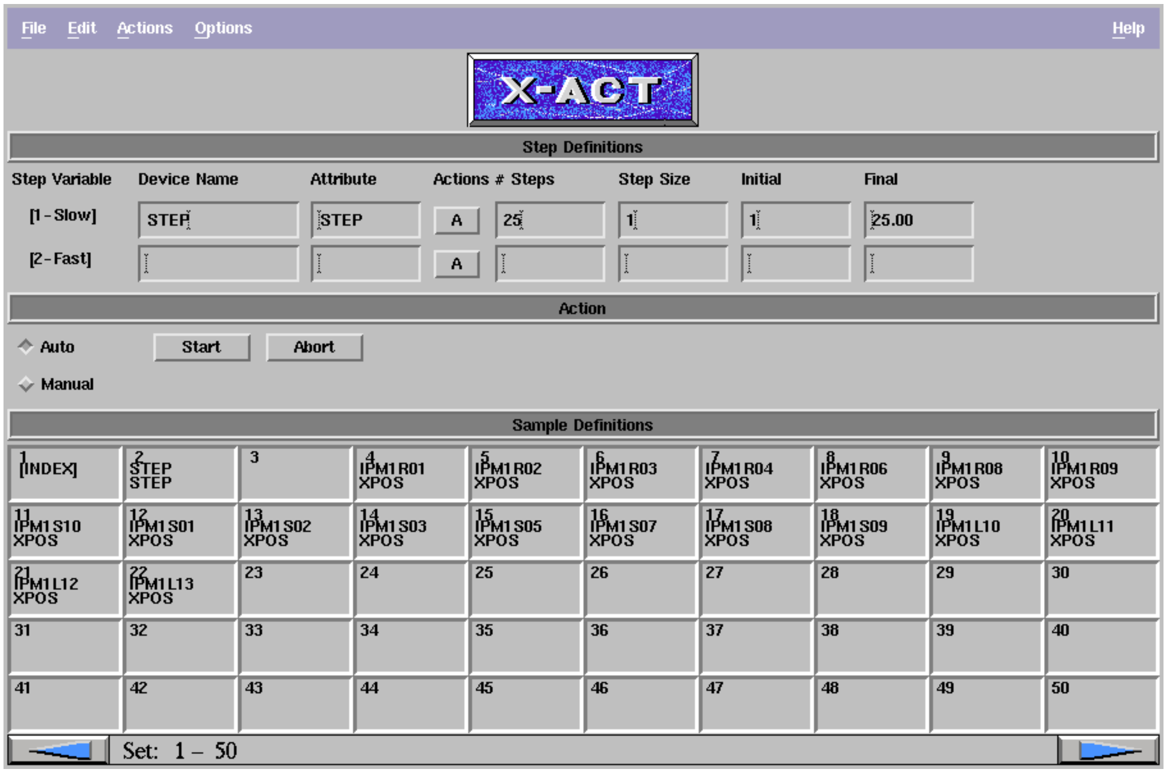
Step and Sample Variables

Users may select up to two step variables although the Correlation Engine allows unlimited number of step variables. Available variables include all control system signals including

- magnet setpoints
- RF setpoints
- Feedback parameters
- Multidevice knobs (composite devices)

A special device keyword, INDEX allows sampling signals without stepping any device. Selecting INDEX as step variable coupled with time delays between samples (entered via action commands), provides a flexible facility for recording the time history of signals. Currently, a stepped device follows a ramp function with the user specifying the number of steps, and the initial and final values of the ramp. Sinusoidal, square and triangular input functions will be introduced in future.

Entering sample variables is similar to the step variables and up to 1000 may be selected. A special sample variable, INPUT, allows manual data entry for each step.



Step Variable	Device Name	Attribute	Actions	# Steps	Step Size	Initial	Final
[1 - Slow]	STEP	STEP	A	25	1	1	25.00
[2 - Fast]			A				

◇ Auto
◇ Manual

1 [INDEX]	2 STEP STEP	3	4 IPM1R01 XPOS	5 IPM1R02 XPOS	6 IPM1R03 XPOS	7 IPM1R04 XPOS	8 IPM1R06 XPOS	9 IPM1R08 XPOS	10 IPM1R09 XPOS
11 IPM1S10 XPOS	12 IPM1S01 XPOS	13 IPM1S02 XPOS	14 IPM1S03 XPOS	15 IPM1S05 XPOS	16 IPM1S07 XPOS	17 IPM1S08 XPOS	18 IPM1S09 XPOS	19 IPM1L10 XPOS	20 IPM1L11 XPOS
21 IPM1L12 XPOS	22 IPM1L13 XPOS	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Set: 1 – 50

Figure 2: Data Acquisition Interface for the Correlation Package

Actions

For each step and sample variable one may specify a list of pre and/or post actions. Additionally, Initial (pre experiment), Intermediate and Final (post experiment) actions are also available from the Actions menu. Currently available actions include "Time Delay" (simple time delay after stepping or sampling), "Time Delay Sync" (synchronized sampling) and "Script" (executing a shell or a Tcl script at any point during the experiment).

Control

The data acquisition may be performed in auto or manual form. In manual acquisition the program acquires a sample only after the user has so indicated. Users may also abort the acquisition in which case they will be prompted whether the data should kept or discarded.

Once an experiment has been defined, the entire setup including the list of sample variables, step variables and actions may be saved for future recall and execution. This allows for setting up and debugging complex experiments in advance of actual use in the control room. Additionally, after an experiment, one may save the collected data for analysis and plotting with the Matlab analysis package.

ANALYSIS

The analysis portion of the Correlation Package is written using the Matlab [3] modeling package and is called the Matlab Analysis and Graphics Laboratory Environment (MAnGLE). Matlab was chosen as the basis of the analysis portion because it contains a wealth of graphing and analysis functions and because most laboratories have Matlab licenses. MAnGLE presents the user with a graphical interface to the data collected by XACT allowing them to perform basic analysis and graphics functions by pointing and clicking. Communication between MAnGLE and XACT is through disk files. MAnGLE reads the XACT setup file to obtain a list of stepped and sampled signals as well as information about the input functions. XACT stores the data collected in a file specifically formatted for Matlab. In the current implementation, the user must save the XACT setup and data file and then start up MAnGLE as a separate process and select the named data files for analysis. In the future, XACT will spawn MAnGLE and users will be able to analyze and view the data as it is being collected.

The MAnGLE user interface is divided into two sections. The first is a control panel which displays the device/attribute pairs the user selected with XACT along with menus and associated popups which allows the user to select predefined analysis functions and options. Graphics and analysis results are displayed in a separate window. The current implementation of MAnGLE provides 2-D graphics display of any correlation variable against any other. There are currently three plot options. The entire data set may be displayed at once in a scrollable window. A second plot option allows selecting a default X variable and then plotting any other variable as the Y variable. Y variables are selected by clicking on the device/attribute name displayed in the control panel window. The graphics window can be cleared between plots or plots may remain in the window to display a family of curves. A third option is to select a default Y variable and cycle through any number of X variables. Future plans include bar chart displays and 3-D graphics displays. Analysis functions such as curve-fitting, histograms, minima and maxima determination and FFTs will be added later. Future implementations will also allow users to write their own Matlab scripts (m-files) which can be added to MAnGLE.

IMPLEMENTATION

The correlation Plot Package has been developed utilizing Object oriented design and C++ implementation. The core of the package which may be embedded as a data acquisition module in any high level application, consists of the correlation specification, *corrSpec*, and the correlation *engine*. *CorrSpec* maintains all the information that defines an experiment; the correlation engine makes use of this information to run through the experiment.

Correlation specification consists of five major parts: a set of initial actions, a set of step variables, a set of intermediate actions, a set of measurement variables and set of final actions.

From an object point of view, a *corrSpec* object may hold lists of step objects, action objects and measurement objects (Figure 3). Each step or measurement object could optionally contain lists of pre-actions and/or post-actions as needed. The *corrSpec* object also contains all the necessary methods to manipulate these lists of objects.

A step object can vary any accelerator parameter with a given initial value, step size, final value, and a stepping function (ramp, square, triangle, sine). Step specifications may also be time or an index. In the case of an index, the engine collects all sample variables sequentially without stepping any device in between the samples. An *action* object can be any general action, such as a time delay (synchronous or asynchronous), adjusting any control system parameter, conditional action by comparing some parameter values, waiting for some events to take place, or executing a script. A measurement object is just a container which holds the measured data.

All classes including step, action and measurement, are organized in a hierarchical structure so that a simple interface can be presented. A major benefit of this organization is the ability to augment the action class without the need to modify the access interface.

The correlation engine's interface to the accelerator environment is solely through the CDEV device layer and is thus independent of the underlying control system. This allows any correlation application to have uniform access to any machine data regardless of its source, such as online data from low level microprocessors, static data from a central database, modeling information, etc. Similarly, once a new device (or device type) has been added to a service under CDEV, such as EPICS [4], it becomes automatically available to correlation applications for measurement and/or control.

The correlation engine is also capable of periodically calling user callback functions to allow applications to perform other tasks such as temporarily suspending data acquisition or aborting the experiment.

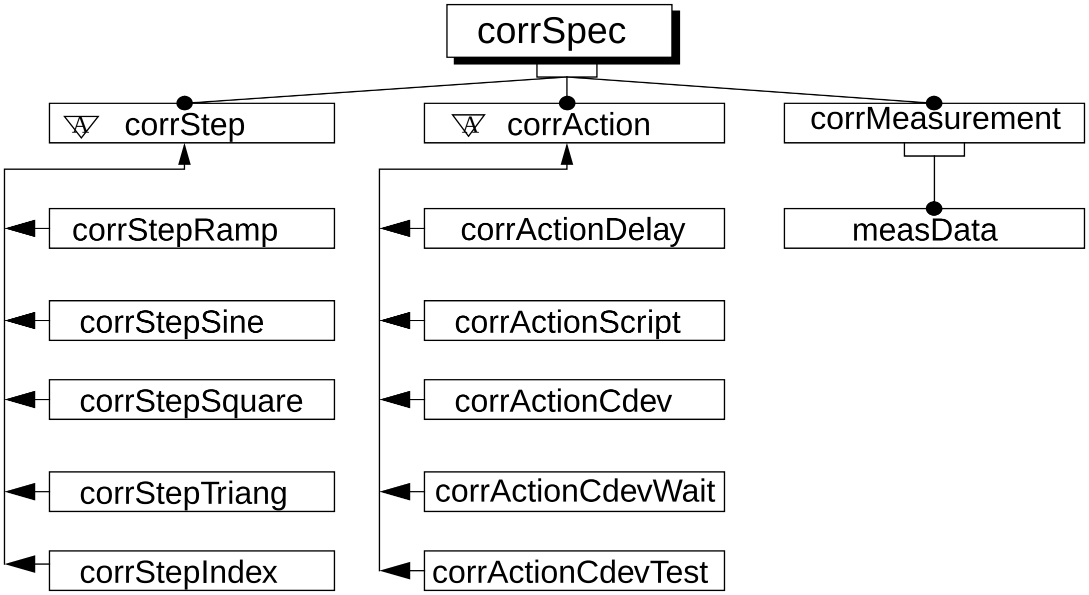


Figure 3: Correlation Plot Package Object Classes Scheme

FUTURE PLANS

The correlation package which is currently undergoing beta testing contains the basic features that are necessary for online data acquisition and control. In order to provide a comprehensive environment for performing online accelerator physics studies, the following developments are planned:

- Uniform access to all control system data including correlation, archived history, and high speed buffered data;
- Complete online data analysis capability including curve fitting, signal processing, and statistical analysis;
- Availability of graphics and analysis directly from XACT, and transparent data transfer from XACT to MAnGLE;
- Availability of a variety of input functions such as triangular, square, and sinusoidal;
- Implementation of correlation engine as a service under CDEV. This would allow its use as an embedded engine by high level applications such as emittance calculation and lattice diagnostics.
- Implementation of a scripting language to automate all phases of an experiment including data analysis.

REFERENCES

[1] J. Chen, G. Heyes, W. Akers, D. Wu, W. A. Watson III, *CDEV: An Object-Oriented Class Library for Developing Device Control applications,* International Conference on Accelerators and Large Experimental Physics Control Systems, October 1995.

[2] L. Hendrickson, N. Phinney, L. Sanchez-Chopitea, *Correlation Plot Facility in the SLC Control System*, Proceedings of Particle Accelerator Conference, May, 1991.

[3] Matlab reference Manual, Copyright The MathWorks, Inc., 1984-1993

[4] Leo R. Dalesio, et. al. *The Experimental Physics and Industrial Control System Architecture: Past, Present, and Future,* International Conference on Accelerators and Large Experimental Physics Control Systems, October 1993.