NLC Specifications

Documenting the existing system is important and will be invaluable later but

- Specs are too detailed for easy evaluation
- Tools and requirements can get entangled in implementation
- NLC may want/need something beyond SLC

Exercise - try writing a few NLC specs

- Determine what they should contain
- Compile a list of tools needed
- Look at implications for overall architecture

Starting list of specs (not simple SCADA-type)

- Some NLC specific examples not in SLC
  - beam-based alignment, linac autosteering
- Facilities which determine networking/bandwidth requirements
  - feedback, MPS

Points to other facilities needed, e.g. contention resolution

- alternative 120 hz datastream concept

Initial Specification List

Beam-based Alignment	NP
Linac Autosteering with Movers	NP
Machine Protection System	NP (per MC/MCR)
Beam-based Feedback	LJH
Beam and Wire Scans	LJH
Correlation Facility	JRB
120 hz Data Acquisition	JRB
Booting and Download Requirements	JJR
Contention Handler	JRB
Linac Energy Management	NP (in progress)
Low-Level RF Control	RC (in progress)
Synchronized Device Control (magnets, movers, klystrons)	
Timing System	
Scripts and Macros	
Save and Restore of Machine Settings	
Optics and Modeling	
Error and Message Facility	
Alarm Handler	

NLC Specifications - another approach

Purpose: very short top-level description of the facility

Use: when, what conditions, how often, machine state

Procedure: coarse, implementation-independent list of major procedural steps

User Interface: minimal description of I/O

Performance requirements: precision, latency, response time, reliability

System impact: possible effects on others

Services Used: ‘toolbox’ - requirements on lower-level utilities

Taxonomy: quick lookup table of usage parameters

How initiated	manually / free-running / both
Frequency	continuous / as needed / every n minutes
Read/write	read / ephemeral write / write
Shareable input	yes / no
Running state	normal running / diagnostic incompatible with normal running
Latency requirement	n pulse / human / none
120 hertz	yes / no (data and/or control)
Sync time	pulse / longer time / doesn’t matter
Missing data	intolerable / can manage / can retry
Geography (input)	local / global / none(offline)
Geography (control)	local / global / none(offline)

Beam Based Alignment of Quads and Sextupoles

Purpose:
Measure the offset between the magnetic center of each quad or sextupole with respect to the the readout center of the BPM captured in it.

Use:
This procedure is done infrequently, on a time scale of weeks/months. It is a setup procedure not run during normal operation. It would use a pilot beam, a 1 bunch train at 120 hz which would vary from low intensity, high emittance to nominal parameters.

Procedure:
For a range of devices, step through each device one at a time, acquiring data at different magnet strengths. Take a reference orbit. Change strength of magnet and allow a settling time. Read ~ 40 BPMs for ~ 200 pulses. Fit the difference orbit for each pulse for incoming orbit, reject errant pulses, fit for the kick at the magnet. After looping over ~ 5 steps, (mini)standardize the magnet and move to the next. Feedbacks downstream of data acq devices would remain on. Procedure should handle feedback within the range automatically.
Analysis: Fit the average kick at each step to find the magnet center/BPM offset Linear for quads, quadratic for sexts. Update the BPM offset in the database. Data should be saved for later examination. For commissioning of the accelerator (and software) one will want to review all data before implementing. Later the analysis and database updating should be completely automated, with QC and errors flagged. Data quality may be better if offsets are implemented as they are determined.

Beam Based Alignment of Quads and Sextupoles (2)

User interface:
User selects range of devices, number of steps, step size, settle time, etc.
User sees a summary table of measured offsets but may examine data for individual devices and display fitted orbits for individual pulses.

Performance requirements:
The large number of devices involved requires a fully automated procedure optimized for speed. Typical resolution required is 1-2 μm in the main linac, 20 μm in the damping rings. Simulation facilities are needed.

Services Used:
1) 120 hz data acq of ~40BPMs for ~200 pulses synchronized with magnet strength changes
2) BPM difference orbit fit including kick at magnet
3) Magnet control - setting & standardize
4) Model matrices
5) Plotting and fitting of linear and quadratic functions
6) Data filtering
7) Automated handling of feedbacks
8) Database updating and data archiving
9) Contention resolution
10) Simulation facilities

Beam Based Alignment of Quads and Sextupoles (3)

Taxonomy:

Initiated	manually
Frequency	as needed
Running state	diagnostic incompatible with normal operation
Read/write	ephemeral write & write
Latency	none
120 Hz	data and control
Shareable input	maybe
Sync time	few pulses
Missing data	can manage or retry
Geography	regional
# of users	varies, max of 4?
input data	>200K bytes

Linac Autosteering with movers

Purpose:
Establish and maintain the beam orbit well centered in magnets and structures. Each quadrupole and each girder of structures are equipped with movers which are used to align the device on the beam trajectory.

Use:
This procedure is done for initial setup of the main linacs using a pilot beam and then runs continuously during normal operation. This spec describes steering of the main linacs but the same algorithm would be used to steer other beam lines equipped with movers, e.g. the damping rings, bunch compressors, prelinacs, etc. These lines might be steered on demand rather than continuously. A complete pass of steering the main linacs is expected to take about 30 min.

Procedure:
Starting at the beginning of the linac, align the first n (~50) magnets, iterating as needed, then move downstream n/2 magnets and repeat until reaching the end of the beamline.
Read all QBPMs in range plus enough upstream BPMs to establish the incoming trajectory (10-20). May need averaging of orbits for several pulses. Fit incoming trajectory and reject bad pulses.
Calculate mover motions required to realign the quadrupoles and implement.
Read all structure BPMs, calculate girder moves, and implement.
Iterate until converged, then move downstream n/2 magnets.
Feedbacks should in general remain on. Procedure should handle feedback within the range automatically.

Linac Autosteering with movers (2)

User interface:
User selects range of devices, algorithm, etc.
User sees corrected orbit, graphic display of changes.

Performance requirements:
The function must execute as efficiently as possible. It must automatically detect and filter bad data. Mover motions must be synchronized. Simulated data must be available for system testing.

System impact:
Because it runs continuously, this function must gracefully coordinate control with other applications.

Services Used:

1) Continuous operation during normal physics running. Both main linacs and several other regions would typically be being steered at the same time.
2) Intermittent Data acq of ~100 QBPMs and ~ 200 structure BPMs for a few pulses.
3) Model matrices
4) Data filtering
5) Automated handling of feedbacks
6) Contention resolution
7) Simulation facilities

Linac Autosteering with movers (3)

Taxonomy:

Initiated	manually and free running
Frequency	continuous
Running state	normal operation
Read/write	write
Latency	few pulses
120 Hz	data
Shareable input	yes, BPMs must also go to other users
Sync time	data - same pulse, control - few pulses
Missing data	can manage or retry
Geography	local/regional
input data	>?? K bytes

Machine Protection System (MPS)

Purpose:
The MPS system protects the accelerator hardware from damage due to errant beam pulses. A single full intensity beam pulse downstream of the damping rings will destroy any material it intercepts. The MPS protects against single pulse failures as well as integrated damage from thermal stress and radiation. MPS also controls a programmed ramp sequence to return the NLC to full beam intensity after an interruption.

Use:
MPS is active at all times when there is beam and must be fully automated. It acquires data at full 120 hz beam rate from distributed sources and provides diagnostic info to users. Configuration of the MPS system is an expert function done under tight administrative control.

Performance requirements:
MPS is an essential system with the highest priority. It has the tightest time requirements of any software in the NLC and must operate at full beam rate at all times. CPU and network conflicts with other subsystems may not be allowed to adversely impact its performance. The system must be highly reliable and failsafe, but designed to avoid spurious trips. A "heartbeat" from the MPS system in every region of the machine is needed; if the MPS does not respond, the machine must be shut off.

Machine Protection System (MPS) (2) - Procedure:

The MPS has several distinct functions with different time scales and latency requirements. There are effectively two separate MPS systems for the electron and positron halves of the machine.

1) The Fast Beam Abort operates within 200 μsec before beam time. It checks the status of all critical fast devices including RF, kickers, etc. and dumps the beam before the main linac if they are not within tolerance. This system is almost hardware, but requires a software interface to configure it.

2) The Beam Abort system operates at 120 Hz and must be capable of acting on the next beam pulse. It interlocks directly against a class of device failures. A watchdog in each sector issues a beam permissive only if the timing and control system are active. This system also handles vacuum valves, stoppers, water and other slower devices (why not average MPS?).

3) The Maximum Allowable Interpulse Difference (MAID) operates at 120 Hz and must be capable of acting on the next beam pulse. It interlocks indirectly against a class of device failures by using a distributed set of BPMs to determine that the beam orbit is within spec and issue a permissive. NOTES: this prohibits accelerator operation with full current below the nominal beam rate of 120 Hz. This also implies that all devices which affect orbits are too slow to cause a problem in <120 hz.

4) The Average MPS system operates on slow (~1 second) time scales. It reads loss monitors, thermal sensors, etc. distributed over many kilometers, with many inputs and determines whether full rate operation is permitted. In the injector regions, it may allow reduced rate operation to the damping rings. Downstream it allows only a pilot beam until the condition is corrected.

Machine Protection System (MPS) (3) - Procedure (continued):

5) The Beam Ramp system operates at 120 Hz. After an MPS trip, it generates a low charge, high emittance 1 bunch beam to verify system integrity. The beam is restored to nominal rate, intensity, emittance and # of bunches through a programmed sequence of steps using the permissives from MAID and other MPS to determine that the beam quality is adequate to proceed.

User interface:

Users must be able to monitor all aspects of MPS functions, including display of BPM data used by MAID on pulses preceding a trip. User must be able to bypass malfunctioning devices with password control. Users and invasive application software must be able to request any of the ramp up beam configurations with less than nominal parameters.

Services Used:

1) 120 hz BPM data.
2) Pre-beam 120 Hz klystron and kicker status.
3) Orbit plotting facility for archived BPM data
4) Contention resolution
5) Simulation facilities

Machine Protection System (MPS) (4)

Taxonomy:

Initiated	free running
Frequency	continuous
Running state	required for all operation
Read/write	‘writes’ permissive
Latency	<1 pulse
120 Hz	data and control
Shareable input	yes, BPMs must also go to other users
Sync time	same pulse
Missing data	intolerable
Geography	global

Misc comments:

A global control system watchdog in each sector trips the beam if the controls are not alive. This has an implication on accelerator down-time, since rebooting any microcomputer would take away the beam.

Where beam position monitor data is used (MAID), each region will have specified criteria indicating what fraction of data may be bad or missing. In the case of unusual expert BPM functions which may not be compatible with MPS, the contention resolution system must insure that the allowed fraction of missing measurements is not exceeded.

NLC Requirements: Correlation Facility

Purpose

Flexible acquisition, analysis and display of (finite-length) sequences of correlated data from different sources. Must also allow for correlated control of devices and pseudo-devices, such as feedback loops (in what follows "device" is usually shorthand for "device or pseudo-device").

Use

The Facility will be used frequently, both in an interactive mode and in canned procedures. Requests to the Correlation Facility may be disruptive of normal running, depending on details of the control component (if there is one) of the request; however many typical uses will be compatible with normal running.

Procedure

Acquire a sequence of data points for a collection of specified devices approximately concurrently, meanwhile optionally stepping one or more control devices. Time may also be used as a step variable.

If there are control devices, for each acquisition point the Facility will wait until the new value is reached (or settle time has elapsed?) before acquiring data from read-only devices. Control devices are returned to their initial states at the end of the sequence

NLC Requirements: Correlation Facility (2)

User Interface

User (human being, script, other code) specifies data to be acquired (i.e., sources), time structure of request, and devices to be controlled along with a sequence of setpoints and, if applicable, settle times for each controlled device.

Acquired data are available to calling code and to the interactive user for analysis (fitting, filtering, transforms,...) and display. Standard fits (e.g., polynomial, trigonometric) and plots (histograms, scatter plots) will be readily available to the interactive user. The programming interface will include the ability to supply custom analysis and plot routines.

Interface Issues

- Contention control will be necessary for some uses of the Facility, depending on devices involved, and impact on beam.
- Require some degree of uniformity of interface of devices to be read and controlled so that the Facility may handle them in a generic fashion.
- Controlled devices should provide some form of readback or other acknowledgement that they have reached the requested setpoint (or are not going to).

NLC Requirements: Correlation Facility (3)

Performance Requirements

- The Facility should be extremely broad and flexible, encompassing essentially all forms of read-only data available to the Control System (raw device readback, database quantities, etc.) and many control operations.
- A wide variety of standard plots and analysis should be available to the interactive user and callable by code. There should also be a means for calling applications to plug in custom analysis.
- Response time should be commensurate with the kind of data and time sequence requested. That is, the Facility should not add significant (on a human time scale) overhead to the acquisition
- Guarantee of appropriate form/degree of synchronization: handling of settling time; maybe user-specified tolerance on total time for acquisition of all data belonging to a single point. This is the core functionality of the Facility; other pieces of it – code to access specific devices and post-acquisition analysis - are probably farmed out to lower-level utilities.
- Facility is multi-user.
- Must handle a variety of returned-data formats (floating point, integer, bitmasks, ...).
- Provide graceful handling of acquisition/control errors and missing data.
- Support a standard file format for acquired data so that data can be imported to and exported from the package.
- Maintainability: should be possible to introduce new devices into the Correlation Facility with only minimal, localized changes to Correlation Facility code. (This depends heavily on 2. in **Interface Issues**.)

NLC Requirements: Correlation Facility (4)

Services Used

- Plotting (scatter plots, histograms, ...), fitting (polynomials, trig functions, ...) filtering and analysis. That is, these are probably not built into the Correlation Facility itself, at least not in a tightly-coupled way, but are available as separate packages.
- Access to all devices and pseudo-devices for which one might wish to gather data or issue control operations in a correlated manner.

Taxonomy

How initiated	manually; perhaps also automatic-cyclic for some appl.
Frequency	as needed (probably often)
Read/write	read and often ephemeral write
Shareable input	often yes
Running state	often but not always compatible with normal running; depends on ap
Latency requirement	human interactive time
120 hertz	may read 120 hertz data
Sync time	typically fraction of a second
Missing data	can tolerate (but caller may have to take some action, such as retry)
Geography (input)	may be global
Geography (control)	may be global

<u>120 hertz Correlation of Acquisition/Control</u>

Most of the preceding applies to pulsed data/control virtually unchanged. In what follows remarks specific to pulsed data and control will be in *italics*.

<u>Purpose</u>
Flexible acquisition, analysis and display of (finite-length) sequences of correlated data from different sources. Must also allow for correlated control of devices and pseudo-devices. *All devices read must be able to provide readback labeled with a pulse id and may have to do the readback on pre-selected pulses indicated by some synchronizing signal (and similarly for writing to control devices).*

<u>Use</u>
Same as above. *There is a potential for producing large amounts of network traffic and poor response due to serialization if underlying facilities don't provide efficient support.*

<u>Procedure</u>
Acquire a sequence of data points for a collection of specified *pulse-aware* devices concurrently *(on the same pulse)*, meanwhile optionally stepping one or more control devices. *May also want to acquire data for non-pulse devices at approximately the same time.*

<u>User Interface</u>
Setup is as above *except user must be able to specify conditions under which a pulse may be used for acquisition/control. Also any specifications having to do with time -- e.g. settle time – would have to be handled differently.* All other remarks above apply as well to correlation of 120 hertz data and control.

<u>120 hertz Correlation of Acquisition/Control</u> (2)

<u>Performance Requirements</u>

- The Facility must be extremely broad and flexible, encompassing essentially all forms of read-only *pulsed* data available to the Control System (including detector data) and *pulsed* control operations.
- Plots and analysis: remarks as above.
- Response time: remarks as above.
- Guarantee of appropriate form/degree of synchronization. *For pulsed data, this means all returned data must be labeled with a pulse id and be synchronized to a pulse. It must be possible to specify the kind of pulse on which a control operation or readout is to occur.*

Items 5 to 9 as above.

<u>Interface Issues</u>

As above. Also add

- *Because of the potential for generating requests for large volumes of read-only data, and particularly since it is likely that this data will be in demand by other applications, there must be a way to share it.*

<u>120 hertz Correlation of Acquisition/Control</u> (3)

<u>Services Used</u>

As above. Also,

- *The primary mission of this Facility -- to provide for synchronized acquisition and control at 120 hertz -- cannot be accomplished without help from lower layers of the Control System. It depends on the existence of a utility to provide broadcast signals or in some other way to synchronize read or write operations on an arbitrary collection of far-flung pulsed devices.*

<u>Taxonomy</u>

How initiated	manually; perhaps also automatic-cyclic for some applications
Frequency	as needed (probably often)
Read/write	read and often ephemeral write
Shareable input	often yes
Running state	often but not always compatible with normal running; depends on appl.
Latency requirement	can tolerate human interactive time wait before start
120 hertz	*reads 120 hertz data; may write to devices at 120 hertz*
Sync time	*one pulse*
Missing data	can tolerate (but caller may have to take some action, such as retry)
Geography (input)	may be global
Geography (control)	may be global

Feedback Systems

Purpose

Feedback systems provide automated control and monitoring of beam-based machine parameters. This includes BPM-based feedbacks throughout the machine, except for the ultra-high rate feedbacks in the damping rings and IP.

Use

Feedback is a fully automated system which runs continuously.

Procedure

A variety of feedback systems will be used with differing algorithms and specifications. For the large number of systems, the software must be generalized and database-driven.

Here is a proposed procedure for the linac steering feedback system:

Each of the two linacs will contain 5 feedback loops, each with around 30 beam position monitors (BPMs) and 8 correctors. (For a single loop, these devices are distributed along the accelerator and not clumped in a single location.) The feedback algorithm will be designed in advance using the accelerator model, and matrices will be periodically calculated offline using a math package and stored in the online database.

After the beam passes, the BPM system calculates position measurements for a single selected bunch, or for the average of a number of bunches. Each feedback calculates intermediate beam states for its region and sends information to all downstream loops. The downstream loops combine the beam information from all upstream loops and their own measurements to determine the amount of a disturbance which must be corrected by this loop. Actuator settings are calculated and implemented.

Feedback Systems (2)

User Interface

The user can enable/disable feedback control, modify gain and setpoints, save and restore machine configurations. Real-time and archived data from feedback systems can be displayed and correlated with other 120 hz data.

Performance Requirements

Typical feedback response time should be 2 120-Hz machine pulses. This includes time for measurement digitization, any system latencies, network traffic, calculation time, and time for corrector power supplies to change and affect the beam. Since corrector changes are likely to require a full 120-Hz interval, only a single 120 Hz period can be allocated for the other functions. Timing variability should be limited to 1/360 second.

Because of the large number of feedback loops and their importance to operations, the system must remain functional without requiring frequent human intervention. For instance, a power supply driver which frequently freezes up and requires user reset would be unacceptable for feedback use. System designers should plan to accommodate periodic, automated control even for slow devices for which feedback control is not initially anticipated.

Interface Issues

Feedback must coordinate with other applications using devices in the same region. This includes pausing feedback for another activity, sharing of data, and synchronization of control requests. All feedback data must be shareable with other users.

Feedback Systems (3)

Services Used

1) 120 hz data from BPMs, charge monitors, and other pulse-synchronized devices
2) 120 hz control of magnets, RF, timing devices
3) 120 hz distribution of feedback state data to other systems
4) Synchronized control of slow devices
5) Orbit plotting facility for BPM data
6) Database access for matrices, parameters
7) Archiving of data from ~few K pulses + long term archiving
8) Error logging and metering
9) Contention resolution
10) Simulation facilities

Taxonomy

Initiated	manually or automatically
Frequency	continuous
Running state	compatible with normal operation
Read/write	write
Latency	2 pulses
120 Hz	data and control
Shareable input	yes
Sync time	1 pulse
Missing data	can tolerate some broken BPMs
Geography	regional/global
# of users	varies, single feedback instances by region
input data	lots

Services used summary

	Feedback	MPS	Beam/wire scans	BBA	Autosteering	Correlations	120 hz data acq	LEM	RF control	Offline analysis
120 hz data acquisition	X	X	X	X	X		X		X	X
120 hz control	X		X				X		X	
Sync. device control	X		?	X	X	X		X	X	
BPM difference orbit	X	X	X	X	?	X	X			X
Model matrices			X	X	X			X		X
Data filtering			X	X	X	X	?	X	?	X
Data archiving	X	X	X	X			X		X	X
Plotting and fitting	X		X	X		X	X		X	X
Database	X	X	X	X	X	X	X	X	X	X
Contention resolution	X	X	X	X	X	X	X	X	X	
Simulations	X	X	X	X	X	X	X	X	X	X

Contention Handler

Purpose

Handle competing demands on Control System devices (magnets, klystrons, etc.) and pseudo-devices, such as machine parameters, as efficiently as possible while attending to priorities. NLC is bigger than SLC and Control System functions will be significantly more automated. The ad hoc mechanisms used to avoid conflict which were barely adequate for SLC will certainly not do for NLC.

Users

Procedures initiated interactively (from Correlation Facility, individual device control, etc.)
Regularly-scheduled automated procedures
Continuous (e.g. feedback)

Conflict Situations

1. Multiple interactive users writing to a device (or pseudo-device, e.g. machine parameter)
2. Continuous users and one-shot interactive users
3. Writing to a device (pseudo-device) may affect downstream information
4. Multiple readers of by-request data

Implications for Functionality

1. Need write locks, perhaps with associated queues, for devices and pseudo-devices. (At what granularity?)
2. Need a low-overhead way to tell continuous users to hold off temporarily
3. Need read-share for significant quantities of 120 hertz data

120 hz data model

Many users require shared access to Mbytes of 120 hz data/control

- MPS, Feedback – principal users, require low latency
- BBA, steering, beam/wire scans, 120 hz correlations – human time-scales
- offline analysis (e.g. MIA, luminosity)

A possible model is to ‘stream’ all of the data all of the time & archive it all (if possible)

- including BPMs etc., feedback states/actuators, RF parameters, and more
- data load is constant and latencies more predictable
- users ‘subscribe’ to a subset of the data stream
- avoids complex scheduling software – just select from existing data

Examples of simplification:

- Feedback ring buffers, incl. multi-loop correlations
- MPS post-trip orbit display
- BBA & steering – select n pulses
- Beam/wire scans, 120 hz correlations – select data sync’d to action
- MIA has more data than it can imagine
- Allows powerful offline luminosity analysis

<u>120 hz data model</u> (2)

Challenges

1) High bandwidth (aggregate > 10 Gbytes/second)
2) Low latency (~50 μsecs overhead)
3) Reliability (distances ~10 km, 50 nsec noise bursts)
4) Scalable

Almost possible with today's hardware, should be OK in a few years

Implications

- topology – multiple links rather than multiplexed single cable for scalability
  - => dedicated point-to-point links between front-end CPUs and central farm
- front-end CPUs merely ship the data => improves reliability
- transport protocol must be optimized – limit demands on memory, etc.
- data distributed via publish/subscribe model using multicast protocol
  - farm CPUs publish neatly packaged data on known multicast address
  - CPUs wishing data sign up on the appropriate multicast address
- large (beneficial) impact on applications – isolated from network details
  - => spec definition should not presume a particular implementation