

LINAC ENERGY MANAGEMENT FOR LCLS*

P. Chu[#], G. White, M. Woodley, R. Iverson, P. Krejcik, D. Rogind,
SLAC, Menlo Park, CA 94025, U.S.A.

Abstract

Linac Energy Management (LEM) is a control system program that scales magnet field set-point settings following a change in beam energy. LEM is necessary because changes in the number, phase, and amplitude of the active klystrons change the beam's rigidity, and therefore, to maintain constant optics, one has to change focusing gradients and bend fields accordingly. This paper describes the basic process, the control system application programs we developed for LEM, and some of the implementation lessons learned at the Linac Coherent Light Source (LCLS).

INTRODUCTION

LEM calculates, and optionally implements, magnet field set-point settings following a change in energy. The energy deviation may have been due to operational factors, such as in the number, phase, or amplitude of active klystrons (the so called “klystron complement”), or from gradient calibration errors. LEM is used to scale each magnet relative to its existing field set-point, by a factor encoding the estimated energy change at that magnet:

$$B_{new} = \left(\frac{E_{act}}{E_{des}} \right) B_{old} \quad (1)$$

where B_{old} is the magnet's present B field, sourced from the control system, E_{des} is the “scaling basis” - the coefficient relating this magnet's extant B field setting to the energy of the beam at that magnet when its B field correctly implemented the optics (all other things being equal E_{des} is the E_{act} from the last time this magnet was scaled), E_{act} is the estimated energy at this magnet at the time of the LEM correction (discussed further below) and B_{new} is the resulting corrected B field set-point. Following the calculation, B_{new} should be implemented in the magnet.

Scaling the magnets in this way is necessary particularly in linear accelerators because the klystron complement changes the beam's rigidity, $B\rho$, and therefore, to maintain constant optics (Eq. 2), one has to change focusing gradients and bend fields to maintain proportionality in the basic lattice equation.

$$B = B\rho \cdot Kl \quad (2)$$

where recall $B\rho = f\sqrt{((E^2 - e_0^2)/c)}$ [Eq 3] in which $f = 10^{10}$ KGauss-meters/GeV, c_b is the speed of light in meters, e_0 is electron rest mass, and E is the total energy at the

magnet, Kl is the energy independent focusing strength, and B is the magnet field density set-point (BDES).

Including Operational Magnet Changes

Note that one can scale both sides of the equation to maintain constant optics when a magnet's set-point has been changed without a concomitant change to its E_{des} :

$$\frac{E_{act}}{E_{des}} B = \frac{E_{act}}{E_{des}} [B\rho Kl] \quad (4)$$

where $Kl = B_{old}/B\rho$ and $B\rho = (f/c)E_{act}$. More specifically, for total energy in kG-m, $B\rho = 10^{10} \sqrt{(E_{act}^2 - e_0^2)}/c_b$

where e_0 in GeV, c_b in meters/second.

Kl and $B\rho$ are so expressed using the extant BDES and E , so that we can scale the magnet fields and gradients to match changes made to BDESes since the last LEM. That is, so called, “scaling the tweaks”.

MEASUREMENTS AND CALCULATIONS

In Eq. 1 you see E_{act} is the only part that must be calculated on any iteration of LEM. The other components are either read from the control system or from the last iteration (E_{des}). In the LCLS LEM, the basic process is to sum up the actual contributions to energy of the individual klystrons (taken one “sub-linac” at a time), and finding the coefficient f – called the “fudge factor” – which scales the actual change in energy implied by the actual settings, to the reference energy, wanted for each sub-linac (the so called reference energies or “ E_{ref} s”). A just calculated fudge factor of a sub-linac of the extant machine (as opposed to one recorded in a process variable from a previous energy profile calculation) is sometimes called an “ideal” fudge factor.

The reference energies

In LCLS there are 4 sub-linacs, we call them L0, L1, L2 and L3. Between each, we know the design energy from the model, giving 5 E_{ref} s or reference energies. Given the E_{ref} constants, the main calculation to derive the E_{act} s to be used in Eq. 1, is in two parts. First calculate the ideal “fudge factors”, $f(n)$, $n=1..5$. The basic expression is given below:

$$f(n) \cdot \sum_{acc} K_{amp} \cos K_{phase} - \frac{\left| \sum_{dec} K_{amp} \cos K_{phase} \right|}{f(n)} = E_{ref}(n) - E_{ref}(n-1) \quad (5)$$

*Work supported in part by the DOE Contract DE-AC02-76SF00515.

[#]pchu@slac.stanford.edu

where K_{amp} and K_{phase} are the existing amplitude and phase of the active klystrons. Eq. 5 indicates that the energy gain from all accelerating cavities minus the energy loss from all deceleration cavities equals the net energy change within the given sub-linac.

Secondly, we estimate the extant energy at each individual magnet. At LCLS we use a particle tracking code, XAL [1] to accomplish this. To this tracking code we give the extant B field values of all magnets, and for each klystron the extant K_{amp} times the beginning fudge factor of its linac, and its phase.

The result is everything we need to calculate Eq. 1 for each magnet, to scale it to a new "energy" as implied by the klystron complement and extant magnet energizations.

THE LEM APPLICATION PROGRAM

The algorithm described above was coded into a Graphical User Interface (GUI) program. First, a prototype was written in Matlab, then for performance and maintenance purposes, the Matlab program was converted to Java.

Fig. 1 shows the data flow between the various LEM application modules. At the bottom of the diagram, a module called LEM "Lite" carries out the computation in Eq. 5. Then the fitted RF data get passed to the XAL [2] online model code to calculate the E_{act} for each magnet. A LEM computation (or "collect data") module then takes these data and follows Eq. 1 to derive the desired B field (BDES) for affected magnets. Both B_{old} and E_{des} are saved as control system process variables. Optionally, the newly calculated E_{act} values are saved as process variables too.

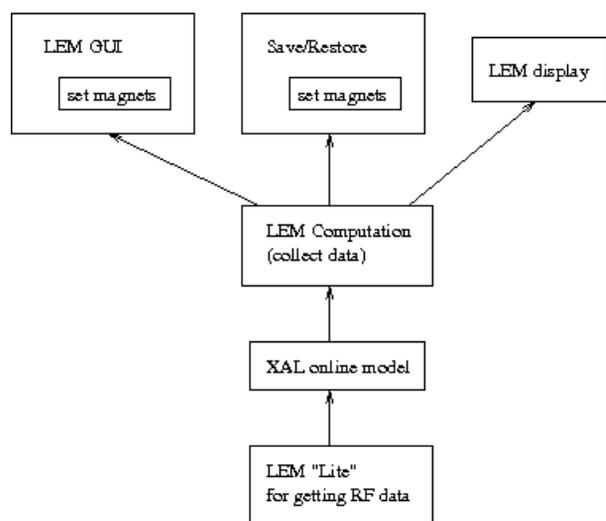


Figure 1: LEM program data flow diagram.

The computation modules are self-contained callable packages. A number of applications use the computation modules. Examples are shown at the top of Fig.1. It is up to each application to implement the resulting magnet settings and E_{des} updates.

LEM GUI

One Java based LEM GUI program was created to facilitate the primary LEM operation. A screen snapshot of this LEM GUI is shown in Fig. 2. In the middle of the application display panel, there are five tabs for various groups of data; shown here is the energy profile display. The "Optics" tab shows how far the present lattice deviated from the desired one by displaying $E_{act} - E_{des}$. The "Phases" tab shows klystron and local compensation (or sub-booster) phase plots. The " B_{mag} " tab displays a table of how far each magnet setting is from the design lattice. It is formulated as:

$$B_{mag} = 1 + \frac{\beta_{design}^2 \cdot (Kl - Kl_{design})^2}{2} \quad (6)$$

where β_{design} is the design beta-function for the magnet. The "Magnets" tab lists the detailed numbers such as B_{new} , E_{des} , E_{act} , B_{mag} and allowed magnet field range. On the right-hand side of the LEM GUI panel, the reference energies and klystron fudge factors are listed. The *Extant* fudge factors are the ones calculated by the previous LEM operation, and the *Ideal* ones are from the present calculation. The "Collect Data" button performs the LEM computation mentioned above and the "Scale Magnets" button applies the newly calculated BDESes to the field setpoints. An "Undo" button is provided for reverting the magnet changes to their previous, pre-LEM settings. Below the main data display panel, a message window shows any information, warning or errors that occurred during the LEM operation. The LEM GUI application also turns off all the transverse feedback controls for steering correctors during the magnet field changing operation and then turns the feedbacks back on afterward. Some beta matching quadrupoles are excluded from the LEM scaling because the matching solution is usually not scalable with respect to energy change anyway.

A typical use case for the LEM GUI is to restore a given lattice when the klystron complement has been changed but the final energy remains the same. There are two meanings for the lattice restoration. Originally the program was set to restore a "tweaked" lattice, which might include some hand tuning. Lately, the program has been used to restore a design lattice with beta matching tweaks preserved. For restoring to the design lattice, the B_{old} and E_{des} in Eq. 1 are taken from the design lattice.

Save/Restore with LEM

The LEM process can also be part of a magnet set-point save and restore operation. In order to restore a previously saved machine snapshot, the K value for a magnet should be maintained. Given the saved snapshot B and E values, LEM can figure out the right BDES to deploy. This, of course, requires that at the time that BDES control system settings are saved, the E_{act} is also calculated and saved. Usually this save/restore with LEM operation takes place during an energy change or when

trying to restore a previously saved lattice with the present energy profile. The LCLS Save/Restore program was originally adapted from the XAL SCORE program.

LEM Display Module or Continuous Update Display

Another application related to LEM is a *read-only* display for the control room. The display takes the first

three tabs of the LEM GUI, reformatted as one single panel. This application is running continuously and updates its display periodically. The same display panel is also packaged as a callable module so that other applications can use it. Examples of users are the Save/Restore system, and a script used to implement changes in beam energy.

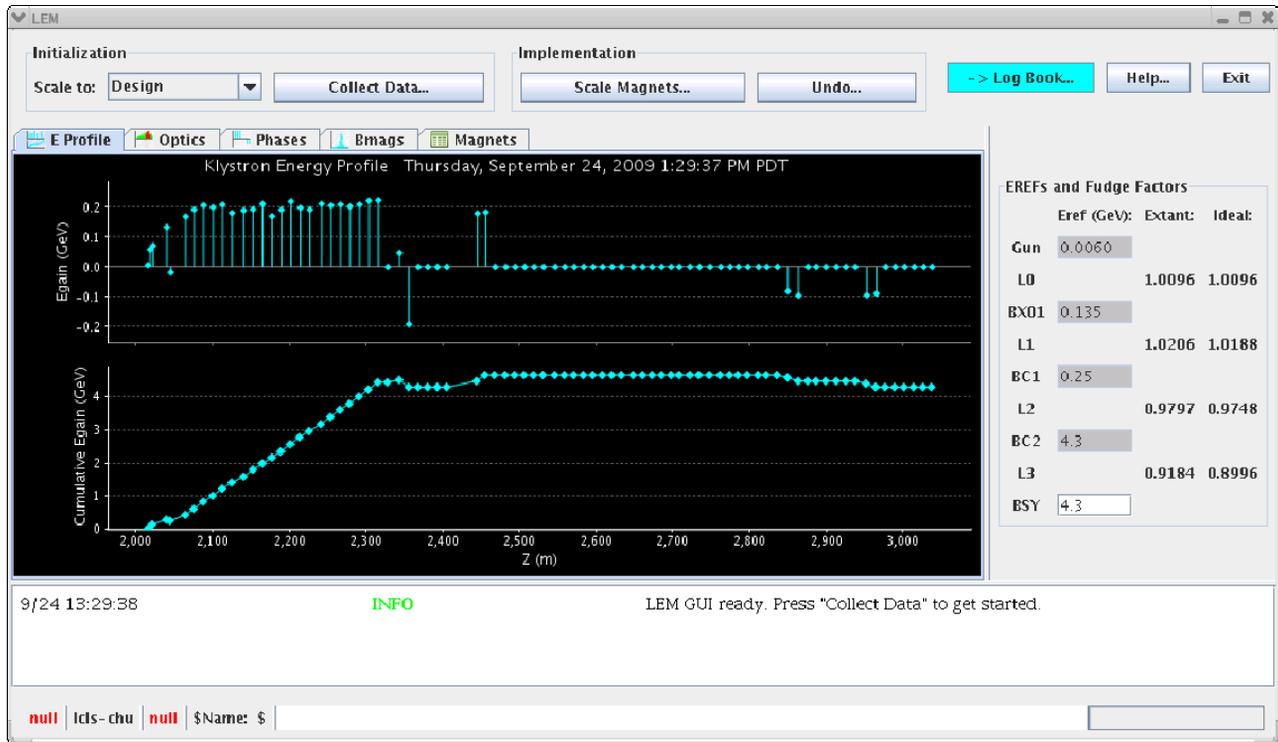


Figure 2: LEM Application screen snapshot. The plots shown are klystron energy gain (top) and energy profile along the LCLS linac.

CONCLUSION

LEM is frequently used by LCLS operations to maintain the lattice at different energies. The basic idea for LEM is quite straightforward. However, many operational considerations can make the program logic rather complicated. The LEM modules and applications have matured, though there are still some improvement plans. Ideas include saving extant machine snapshots in a Save/Restore database, which can be used for LEM undo operations, and to improve LEM computation performance by continuously running the process on a server.

ACKNOWLEDGEMENT

The authors would like to thank the LCLS accelerator physicists and operators for their help to commissioning the LEM related applications. Controls engineers who helped to diagnose magnet and RF data access problems are also appreciated.

REFERENCES

- [1] C. K. Allen, "Theory and Technique of Beam Envelope Simulation", Los Alamos National Laboratory Internal Report LA-UR-02-4979 (2002).
- [2] J. Galambos, *et al*, "XAL Application Programming Structure," p. 79, Proceedings of 2005 Particle Accelerator Conference.