

RECENT DEVELOPMENTS IN THE LIAR SIMULATION CODE

P. Tenenbaum, L. Hendrickson, A. Seryi, G. Stupakov
 SLAC, Stanford, CA, USA *

Abstract

We describe a series of improvements to the LIAR (Linear Accelerator Research) simulation program to permit extended simulations of the low-emittance transport properties of linear colliders. The beam tracking algorithm has been extended to permit proper treatment of high-order multipoles and momentum compaction. A more versatile system for management of RF structure wakefield data permits RF structures with different short-range wakefields to be used in a single beamline, and the management of “design” and “error” long-range wakefields is improved. A comprehensive ground motion model has been added, which includes the correlations in motion between the electron and positron beamlines in a linear collider. A new interface permits users to run LIAR as a sub-program within MATLAB, substantially increasing the flexibility of complex simulation studies.

1 INTRODUCTION

The LIAR (Linear Accelerator Research) simulation program [1, 2] has been used for several years to study the interplay of tuning algorithms, beam dynamics phenomena, and beam instrumentation limitations in the performance of linear accelerators such as the Next Linear Collider (NLC) main linac. More recently, there has been substantial interest in developing a tool that would provide simulation capabilities similar to those in LIAR, but which could be used to model a linear collider from the extraction of its damping rings to the IP. We describe the changes which were made to LIAR in order to permit such broad-based simulations, and present a few examples.

2 TRACKING ENGINE EXTENSION

The original LIAR concept was intended for use in simulating linacs which contain RF accelerating structures, quadrupoles, dipole steering correctors, and diagnostic instrumentation. By constraining the code to elements such as these, it was possible to represent the beam as a small number of first- and second-moment matrices, and to transport these matrices through the beamline. As a further simplification, the matrices representing the beam’s first and second moments were generated at fixed z locations within each bunch and these locations were not allowed to vary.

The beam representation above is not adequate in systems which contain strong sextupole magnets or sources of momentum compaction, such as bunch compressors or linear collider final focus systems. In systems such as these, a

more appropriate representation of the beam is a cluster of dimensionless rays, such as is used in DIMAD [3], MAD [4], or TURTLE [5] tracking codes. The disadvantage of such a representation is that convolution of the beam with the transverse and longitudinal wakefields of an RF structure becomes computationally burdensome.

As a compromise solution, the tracking engine, R - and T -matrix calculation, and beam-generation routines of DIMAD were linked to LIAR, and a set of algorithms were developed that permit the beam representation to be freely converted between the LIAR and DIMAD modes. Additional interfacing routines were created that allowed the beamline information stored in LIAR to be transformed to DIMAD’s format and inserted into DIMAD data structures, and which allowed beam position and size information generated by DIMAD at beam position monitor or profile monitor locations to be sent back to LIAR’s data structures. The interface routines to the DIMAD tracking engine have been designed to be as transparent as possible to the end-user. Beamline and beam information are automatically transferred from LIAR to the DIMAD tracker, and a limited set of beam diagnostic information is automatically transferred from the DIMAD tracker to LIAR. In general, users of LIAR can design algorithms for introducing or correcting errors in an accelerator without consideration for whether the algorithms will ultimately operate on a beamline region which is tracked by LIAR or DIMAD.

3 WAKEFIELD DATA HANDLING

The original implementation of LIAR assumed that the RF structures in a beamline would be nearly identical. Consequently, it was assumed that the short-range longitudinal and transverse wakes of all of the structures could be represented by a single set of wakefield data; only limited capabilities for multiple long-range transverse wakes were included; and management of “error” wakes (wakefields due to construction errors in the structures) was awkward.

These limitations are clearly not acceptable for a program which is to simulate a beamline as complex as a linear collider. As an example, the Next Linear Collider contains RF structures at 1.428 GHz and 2.856 GHz as well as the main linac frequency of 11.424 GHz; several of the structures at 11.424 GHz are used in bunch compression and require wakefield data that extend to larger z values than is required in the main linac; the main linac is likely to use multiple RF structure types, in which each type has the same short-range wakefields as the other types but different long-range transverse wakefields; and a wide variety of error wakefields are anticipated.

The present version of LIAR permits up to 10 differ-

* Work supported by the U.S. Department of Energy, Contract DE-AC03-76SF00515.

ent sets of short- and long-range wakefields to be stored in a central wake data repository. The association between an RF structure and its particular wakefield is through the engineering TYPE attribute, a 16-character alphanumeric string which is specified in the Extended Standard Input Format (XSIF) “deck” file [6]: when a set of wake data is read in, the TYPE string of the structures to be associated with that set of wake data is specified.

Long-range, transverse error wakefields are now parsed by a separate LIAR command; like the design wakefields, error wakes are associated with structures via the XSIF TYPE string. Each RF structure type can be associated with an arbitrarily large number of error wakes, all of which are read from a single data file; the user may apply the error wakes in a given file randomly or sequentially to the appropriate structures in a beamline, and may also rescale the error wakes to make them more or less severe than the file data would indicate.

4 GROUND MOTION MODEL

LIAR permits the user to specify the grouping of beamline components on individual or common supports; in the latter case, the supports are assumed to be in contact with the floor of the accelerator housing only at their endpoints. This arrangement permits realistic simulation of the misalignment of beamline components due to the motion of the ground.

The ground motion is modelled [7] with a 2-dimensional power spectrum, $P(\omega, k)$:

$$P(\omega, k) = \frac{A_d}{\omega^2 k^2} (1 - \cos(kB_d/A_d/\omega^2)) + \sum_i D_i \cdot U_i.$$

Here the first term is diffusive ground motion, which follows the “ATL” relation [8]. The second term represents motion due to isotropic plane waves; the terms are defined as follows:

$$U_i = \begin{cases} \frac{2}{\sqrt{(\omega/v_i)^2 - k^2}}, & |k| \leq \omega/v_i, \\ 0, & |k| > \omega/v_i, \end{cases} \quad (1)$$

$$D_i = \frac{a_i}{1 + [d_i(\omega - \omega_i)/\omega_i]^4}.$$

Eq. 1 describes wavelike ground motion where the power spectrum falls off with the inverse fourth power of frequency from each of a series of peaks; the parameters a_i , ω_i , d_i , v_i correspond to the amplitude, frequency, and width of the peak and the frequency-velocity relation of the waves, respectively. The parameter A_d is the amplitude of the diffusive ground motion, which falls as the inverse square of frequency. Note that, since the diffusive motion falls more slowly than wavelike motion, this model would tend to predict that the relative motion of two separated objects will, for some frequencies, exceed their absolute motion. In order to prevent this, an ATL correction term, B_d , is added.

For evaluation of the collider performance, three models (Fig.1) have been created. Their parameters are given in [2]. Once the parameters of the ground model are specified, simulation of ground motion proceeds by dividing the

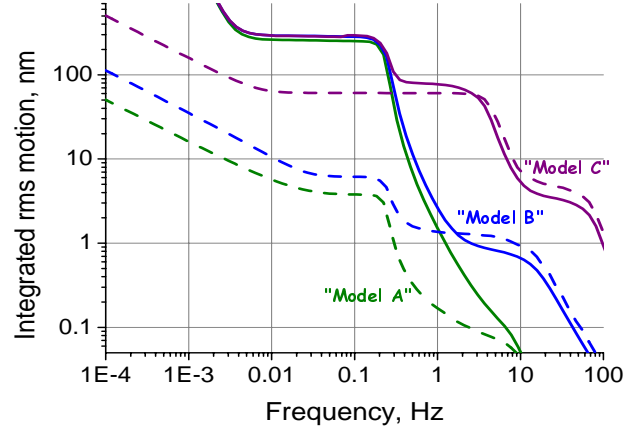


Figure 1: The integrated absolute ground motion spectra (solid lines) and the integrated relative motion of 2 objects separated by 50 m (dashed lines)

(ω, k) plane into bins equidistantly in a logarithmic sense (typically 100 x 100 bins). Within each bin, the algorithm selects a frequency ω_j , wave number k_l , and random phase ϕ_{jl} ; the motion applied to the ground is then a sum of sinusoids, $\delta_y(z, t) = a_{jl} \cos(k_l z - \omega_j t + \phi_{jl})$, where the amplitude a_{jl} is such that the RMS power of the sinusoid is equal to the total power $P(\omega, k)$ in the original bin.

For a linear collider, it is essential to generate correlated motion in the two beamlines (electron and positron or left and right), since the correlation of the motion over long distances cause cancellations in the relative offset of the beams at the IP. To accomplish this, LIAR automatically sets $z = 0$ to be at the downstream end of the beamline (which is assumed to be the IP), and either increments or decrements z from downstream to upstream (so that, for a beamline of length L , the upstream edge of the first magnet is either at $z = +L$ or $z = -L$). By selecting the same set of frequencies, wave numbers, and phases for one iteration of the former configuration and one iteration of the latter,

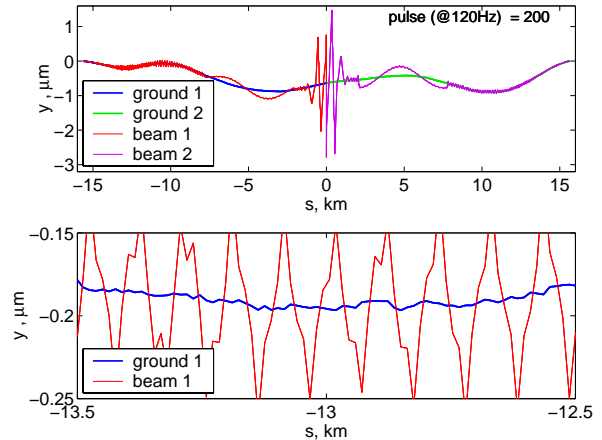


Figure 2: Snapshot of beam orbits and ground distortion in the case of NLC and “moderate” ground motion. The perfect initial alignment has been distorted by 1.67 seconds of simulated ground motion. The bottom plot is an expanded view of part of the linac. IP is at $s = 0$.

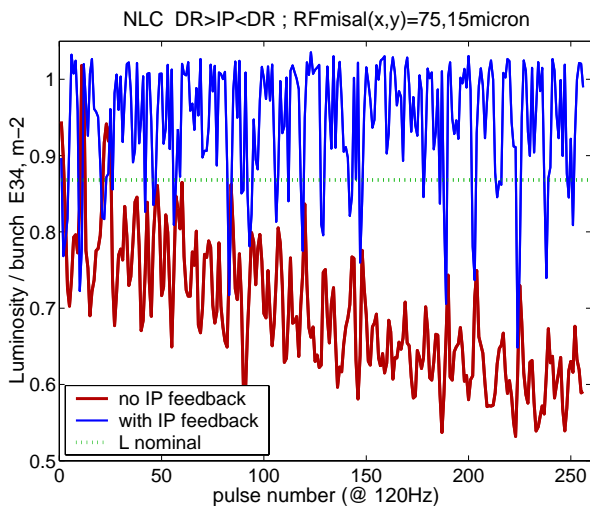


Figure 3: Simulations of NLC with moderate ground motion (model B), with (blue) and without (red) SLC style IP feedback.

it is possible to simulate two linear collider beamlines that meet at the IP and are under the influence of realistically-correlated ground motion, see example in Fig.2.

5 MATLAB-LIAR INTERFACE

In 1998, a set of interface routines between MATLAB [9] and LIAR were introduced. These routines were designed for advanced studies of steering feedback systems in the NLC main linac: LIAR routines would track the beam through the accelerator, introduce perturbations (ground motion or other changes to the beamline), and return diagnostic information to MATLAB; MATLAB would perform calculations to simulate the feedback system and generate new settings for the dipole correctors, which would be passed back to LIAR [10]. This configuration permitted the use of MATLAB's strong scripting, graphics, and mathematical capabilities in LIAR simulations.

The MATLAB-LIAR interface has been extended to support a wider variety of tuning and operations simulations. The present interface allows any valid individual command or command file to be passed to the LIAR command parser; a full set of beam diagnostic information (BPM data, beam profiles at profile monitors, etc.) can be transferred from LIAR to MATLAB, and the complete state of the lattice (magnet strengths, positions, etc.) can be read into MATLAB, modified, and written back into LIAR. Since all LIAR commands are supported, heterogeneous simulations – in which some corrections are performed in MATLAB and others use built-in LIAR algorithms – are possible.

6 INTERFACE TO GUINEA-PIG

MATLAB permits multiple images of LIAR to be loaded simultaneously, and provides each image with its own memory space. This capability allows a simulation in which two completely independent beamlines are simu-

lated, and the two colliding beams are transported to a common interaction point. This capability is used, for example, in simulations of ground motion: the two beamlines are separate, but MATLAB permits a single ground motion model to be applied to each beamline in the appropriate manner.

In order to fully include collision dynamics in the simulations, IP particle coordinates in each beam can be saved for beam-beam simulations by the program GUINEA-PIG [11]. GUINEA-PIG calculates luminosity as well as beam-beam deflections. The deflections are used as the input signal for an IP beam-beam feedback loop implemented in the SLC style [12, 13], see Fig.3 which shows example of simulations with and without feedback.

7 CONCLUSIONS

The LIAR simulation program has been extended to permit full simulations of a linear collider with two complex and independent beamlines. The complete set of tools required for such a simulation are the LIAR program with DIMAD tracking algorithms and multiple short-range wakefields, the MATLAB-LIAR and LIAR-GUINEAPIG interfaces (including the use of 2 LIAR images in MATLAB), the program GUINEA-PIG, and the parameters of the appropriate ground motion model. With a modern Pentium computer and assuming only 1 bunch per linac pulse, 2-beam simulation speeds of 50 linac pulses per hour have been achieved; this is sufficiently fast to permit realistic simulations of beamlines with pulse-to-pulse variation and feedback.

8 REFERENCES

- [1] R. Assmann *et al.*, "LIAR, A Computer Program for the Modeling and Simulation of High Performance Linacs," SLAC/AP-103 (1997).
- [2] <http://www.slac.stanford.edu/accel/nlc/local/AccelPhysics/codes/liar/web/liar.htm>
- [3] R. Servranckx *et al.*, "Users' Guide to the Program DIMAD," SLAC-Report-285 (1990).
- [4] H. Grote, F.C. Iselin, "The MAD Program (Methodical Accelerator Design) Version 8.19 User's Reference Manual," CERN/SI/90-13 (AP) (Rev. 5) (1996).
- [5] D.C. Carey, "TURTLE (Trace Unlimited Rays Through Lumped Elements)," FERMILAB-NAL-064 (1971).
- [6] P. Tenenbaum, <http://www-project.slac.stanford.edu/lc/ilc/TechNotes/LCCNotes/PDF/lcc0060.pdf>.
- [7] A.Sery, O.Napoly, PRE **53** 5323 (1996).
- [8] V. Shiltsev, "Space-Time Ground Diffusion: The ATL Law for Accelerators", in Proc. IWAA95, KEK-95-12,(1995).
- [9] <http://www.themathworks.com/products/matlab>.
- [10] L. Hendrickson *et al.*, "Feedback Systems for Linear Colliders," Proceedings PAC-99 (1999).
- [11] D. Schulte, "Beam-Beam Simulations with GUINEA-PIG," CERN-PS-99-014 (1999).
- [12] L. Hendrickson *et al.*, "Fast Feedback for Linear Colliders," Proceedings PAC-95 (1995).
- [13] <http://www.slac.stanford.edu/~ljh/fdesign.htm>