USE OF SIMULATION PROGRAMS FOR THE MODELLING OF THE NEXT LINEAR COLLIDER*

P. Tenenbaum†, P. Emma, L. Hendrickson, N. Phinney,
T.O. Raubenheimer, M. Woodley
Stanford Linear Accelerator Center, Stanford, CA

Abstract

The Next Linear Collider is an electron-positron accelerator unprecedented in its size, energy, and tight tolerances. We describe the suite of simulation tools which are widely used in designing and modelling the performance of the NLC. In order to achieve a uniform beamline description and permit simulation of all facets of the collider, an extended version of the Standard Input Format (xSIF) has been developed and implemented in MAD and DIMAD. We discuss several enhancements to the MAD and DIMAD calculation engines necessary to properly simulate the most challenging regions of the facility. We also describe enhancements to LIAR which allow it to be used as the tracking engine for a tuning/feedback simulation written in MATLAB. Finally, we discuss the additional software needed to model the beam stabilization and tuning processes.

1 INTRODUCTION

The Next Linear Collider (NLC) is a proposed e^+e^- facility capable of achieving a luminosity of up to $10^{34}\mathrm{cm}^{-2}\mathrm{sec}^{-1}$ at a center-of-mass energy of 1 TeV [1]. The facility includes 2 X-Band (11.424 GHz) linacs, each in excess of 10 km in length; 2 beam delivery systems, each in excess of 5 km in length; and several kilometers of injection linac and damping ring.

In order to reliably achieve such a high luminosity at the NLC, a wide variety of physical phenomena and tuning strategies need to be investigated in considerable detail during the design phase. These include:

- Control of long-range and short-range wakefield effects
- Feedback system architecture for control of instabilities
- Misalignment and error tolerances
- Tuning and capture ranges
- Geometric and chromatic aberrations
- Background sources and transport
- Transport of highly disrupted beams.

In general, we perform modelling and simulation studies of the NLC beamlines with tools which are in wide use throughout the accelerator physics community. However, these tools have been substantially modified to fit our particular needs. The modifications performed are described below. Finally, we should note that all codes are written in ANSI Fortran 77 or Fortran 90. We have adopted this standard for ease of transport and ease of modification. Further documentation on the modeling codes and standards can be found in Ref. [2].

2 EXTENDED STANDARD INPUT FORMAT

The base language for the description of all NLC beamlines is the Standard Input Format [3]. This is the input language used by MAD and DIMAD, as well as other programs. In order to appropriately handle the NLC beamlines, a number of modifications have been made to this language:

- The keywords, parameters, and syntax of the present-day MAD and DIMAD input parsers have been made consistent, generally by adopting the present MAD syntax where conflicts existed
- Apertures have been added to most elements (APERTURE parameter) for use in tracking
- The DIMAD element LCAV, which describes a linear accelerator structure, has been added to MAD; the original LCAV syntax has been extended to allow filenames for wakefield data to be included in the element description
- A number of instrument types (BLMO, SLMO, PROFILE, WIRE, IMONITOR) have been added
- All elements can take a 16 character alphanumeric engineering class (TYPE parameter) and a 24 character alphanumeric database name (LABEL parameter) as parameters
- The DIMAD parameter FINTX (exit-face fringe-field integral) has been added to MAD to improve description of bend magnets which are split into 2 elements in the deck.

The resulting beamline description language is known as the Extended Standard Input Format (XSIF). The NLC versions of MAD and DIMAD will read without errors a beamline description written in XSIF. In addition, a standalone library version of the DIMAD XSIF parser has been

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.
† Email:quarkpt@slac.stanford.edu

created (for AIX, Solaris, NT, and VMS platforms) which can be linked to any new or existing application. The present version of LIAR[4] has been so linked. At this time, all NLC accelerator simulation programs in wide use can accept a common set of beamline descriptions in XSIF.

In addition to the XSIF standard, a fully-instantiated sequential ASCII beamline description was desired. The MAD tape format[5] was enhanced to include new beamline features (beam energy, engineering and database names, new instrument classes, and linac RF structures), resulting in the Extended Tape File Format (XTFF). The SURVEY XTFF provides a compact lattice description that can be used by the engineering teams for component layout while the TWISS XTFF is used to transfer the lattice information into an Oracle database which will become the backbone of the control system and provides a straightforward method of tracking components.

3 DIMAD

The program DIMAD[6] is the primary program for design and simulation of the beam delivery regions of the NLC, and is also used for some studies of the main linac. NLC-DIMAD has been converted to Fortran 90, and all shared data structures have been moved to MODULEs for ease of maintenance. The data space available for beamline and parameter data has been expanded to accomodate beamlines as large as the main linac (15,000 entries in the appropriate LINE).

The NLC main linac contains 4,968 RF structures. It was desired that the main linac deck describe the beamline efficiently, but that the flexibility of individually phasing the structures be preserved. This is accomplished by having a separate array of structure phases, which is set using a new command, PHASE. During calculations the phases in the new array are used rather than the phases in the deck. This permits the user to have a single LCAV in the deck, which can then be used an arbitrary number of times to describe independently-phased structures. NLC-DIMAD also included beam loading in its phasing operations, and can convolve the beam with a longitudinal wakefield description during tracking.

To improve the accuracy of tracking results in the beam delivery region, NLC-DIMAD allows the user to specify a twin-horned energy distribution (similar to that expected at the end of the main linac). The particle analysis code computes a luminosity-weighted RMS beam size, which de-weights the contributions of the beam tails in determining the effective beam size. This size is computed at the end of the beamline, and the luminosity-weighted size at the nearest waist is also estimated; thus if the waists are slightly mistuned the user still gets an estimate of the focused beam size. Several tracking-related variables (including RMS emittances at the end of the line and luminosity-weighted beam sizes) are available as conditions for the least-squares fitter, allowing higly nonlinear problems (such as “Brinkmann-sextupole” tuning) to be automated. For convenience, a syntax for specification of normalized emittances has been added to the BEAM command. In order to accurately track the low-energy tails of the disrupted post-IP beam, an extended chromatic precision option has been added: when this option is selected, quad and sextupole matrices are dynamically calculated for each particle, scaled to match that particle's energy. In the future, we will add to NLC-DIMAD routines for simulation of thermal-photon and residual-gas scattering of the electron beam [7], and use these and the extended chromatic precision option to model backgrounds and track low-energy tails through the system.

4 MAD

The most significant modification to the NLC version of MAD is the addition of linear acceleration and the resulting variation of the beam energy. The additions were made to MAD version 8.23/0 which is written in standard Fortran 77. The code now allows the beam energy to be specified at the entrance to a beamline and this energy will be modified by LCAVITY elements which model traveling wave accelerator cavities; note the LCAVITY elements are treated differently from the standard MAD RFCAVITY elements which do not modify the beam energy. This allows the code to be used for both storage rings and linacs.

The LCAVITY elements are specified in terms of a length, rf frequency, voltage, rf phase, and the loss parameter. All of these parameters are treated like all other MAD element parameters and can be specified in terms of expression, varied during matching, etc. In addition, the beam energy can be used as a matching constraint and is now included in the BETA0 structures. The LCAVITY elements also include approximate representation of the focusing that arises at the entrance and exit of the accelerator structures which can be significant in the low energy ($\sim 100\,\mathrm{MeV}$) regions of the collider. Finally, one can also specify files containing the transverse and longitudinal wakefields for the structures which are used when tracking.

The calculation of the lattice functions has been modified to incorporate the variation in the beam energy. There are a couple of different methods of including the energy variation in the lattice functions. We could have maintained the original definition utilized in MAD where the lattice parameters simply depend on the transfer matrix [8]:

$$\beta_2 = \frac{1}{\beta_1}((R_{11}\beta_1 - R_{12}\alpha_1)^2 + R_{12}^2) \qquad (1)$$

etc., but this leads to lattice functions that, for a given periodic cell structure, decrease with acceleration as $1/E(s)$. For this reason, we normalize the individual transfer elements by the determinant of the transfer matrix. This leads to the more common definition which is consistant with that used in DIMAD.

Finally, three other changes have been implemented. First, different fringing fields can be specified for the upstream and downstream ends of bending magnets using a

FINTX parameter in the element definition. This provides greater flexibility when splitting components into multiple segments and can be used to prevent the error that arises when FINT$\neq$ 0 in split bending magnets. Second, an approximation for the effect of coherent synchrotron radition in the bending magnets can be used when tracking by setting a CSR parameter for bending magnets. Finally, the beamline length has also been added as a matching constraint since it is frequently desirable to minimize to overall length of the system.

5 LIAR

LIAR is a program for the simulation of linear accelerators with high gradients and strong wakefields. Like DIMAD, LIAR has been entirely converted to Fortran 90. This allowed LIAR's data structures to be cast into a standard form, which in turn permitted LIAR to be ported to several platforms with essentially no changes in its code (AIX, Solaris, VMS, NT). LIAR was also linked to the standalone XSIF parser, allowing it to read decks in the standard format and eliminating the need for conversion of the XSIF decks to TRANSPORT format[9].

The beam-based alignment command, QALIGN, has been supplanted by QALIGN_NEW, which allows greater flexibility in specifying the order in which various algorithms are applied to the beamline. A more consise "figure of merit" for multibunch position distortions (non-straight trains) has been added to the program.

LIAR's tracking engine includes longitudinal and transverse short-range wakefields. Because of this it was decided that the LIAR tracking engine would be used for simulations of the NLC trajectory feedback system. The state-space design for the feedback algorithms was written in MATLAB: the MATLAB system, using macros and control toolbox routines, enables easier prototyping than Fortran software, and provides powerful graphics capabilities. Different feedback formulations can be evaluated from this basic platform. LIAR's structure was revised to allow MATLAB to call LIAR routines from within a single MATLAB process, and to pass data between MATLAB and LIAR.

The feedback simulation is driven by MATLAB macros, which call LIAR functions to transport and perturm the beam and return beam position monitor readings. MATLAB routines perform calculations to simulate the feedback system and determine new corrector settings, which are communicated back to LIAR with an interface routine.

6 FUTURE DEVELOPMENTS

The most pressing modelling task facing the Next Linear Collider is simulation of the tuning process, and the related simulation of normal operation (for example, with all feedbacks operating/interacting, including bandwidth limitations). A "cradle-to-grave" simulation of the tuning procedure for the Final Focus Test Beam[10] was performed in SAD[11], and the NLC will require a similarly detailed exploration of its commissioning in order to determine the range and granularity of all tuning devices and the required performance of all instrumentation. Of particular interest is the interaction between the main linac steering feedbacks and the many feedbacks of various kinds in the beam delivery system.

One possible tool for this simulation is the Final Focus Flight Simulator (FFFS), which is an interactive GUI developed at SLAC for Third Order Transport [12]. However, the FFFS has many limitations which make it a non-optimal candidate for this simulation.

A more promising option is to develop a MATLAB interface to MAD or DIMAD, similar to the one presently available for LIAR. This would allow the existing calculational kernels of these two programs to be used without adding large amounts of special-purpose code to the compiled codebase. This is particularly attractive in that it would permit a simulation in which a beam is transported through the linac in LIAR, extracted into MATLAB, translated to a group of particles for MAD/DIMAD, and then transported in that form through beam delivery. This would permit studies of the entire accelerator, despite the fact that different areas are best served by different simulation programs.

7 ACKNOWLEDGEMENTS

The authors wish to acknowledge the contributions of P. Raimondi, G. Roy, I. Reichel, R. Helm, and K. Bane.

8 REFERENCES

[1] NLC ZDR Design Group, "Zeroth-Order Design Report for the Next Linear Collider," SLAC Report 474 (1996).

[2] NLC Accelerator Physics Web pages found at http://www-project.slac.stanford.edu/lc/nlc-tech.html.

[3] D.C. Carey and F.C. Iselin, "A Standard Input Language for Particle Beam and Accelerator Computer Programs," Proceedings *Snowmass 84* (1984).

[4] R. Assmann *et al*, "LIAR: A New Program for the Modeling and Simulation of Linear Accelerators with High Gradients and Small Emittances," Proc. *LINAC 96*, 464 (1996).

[5] H. Grote and F.C. Iselin, "The MAD Program Version 8.19 User's Reference Manual," CERN/SL/90-13(AP)Rev.5, 125 (1996).

[6] R.V. Servranckx *et al*, "Users' Guide to the Program DIMAD," SLAC-Report-285 (1985).

[7] I. Reichel *et al*, "Therman Photon and Residual Gas Scattering in the NLC Beam Delivery," Proc. *ICAP 98* (1998).

[8] K.L. Brown, SLAC Report 75, Rev. 3 (1972).

[9] K.L. Brown *et al*, "TRANSPORT: A Computer Program for Designing Charged Particle Beam Transport Systems," SLAC-Report-91 (1970).

[10] K. Oide, "Design of Optics for the Final Focus Test Beam at SLAC," Proc. *PAC 89*, 1319 (1989).

[11] http://www-acc-theory.kek.jp/SAD/sad.html.

[12] M. Woodley, private communication.