LIAR - A COMPUTER PROGRAM FOR THE SIMULATION AND MODELING OF HIGH PERFORMANCE LINACS*

R. Assmann[†], C. Adolphsen, K. Bane, P. Emma, T.O. Raubenheimer,
R. Siemann, K. Thompson, F. Zimmermann
Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309

Abstract

The computer program LIAR ("LInear Accelerator Research code") is a numerical simulation and tracking program for linear colliders. The LIAR project was started at SLAC in August 1995 in order to provide a computing and simulation tool that specifically adresses the needs of high energy linear colliders. LIAR is designed to be used for a variety of different linear accelerators. It has been applied for and checked against the existing Stanford Linear Collider (SLC) as well as the linacs of the proposed Next Linear Collider (NLC) and the proposed Linac Coherent Light Source (LCLS). The program includes wakefield effects, a 4D coupled beam description, specific optimization algorithms and other advanced features. We describe the most important concepts and highlights of the program. After having presented the LIAR program at the LINAC96 and the PAC97 cnferences, we do now introduce it to the European particle accelerator community.

1 THE LIAR PROJECT

The LIAR ("LInear Accelerator Research code") project was started at SLAC in August 1995 in order to provide a computing and simulation tool that addresses the needs of high performance linear accelerators. Its first objective was to implement advanced simulations for the main linacs of SLC (50 GeV) and NLC (500 GeV) at SLAC. Since then it has been applied to the LCLS project at SLAC (15 GeV), the CLIC project at CERN and to studies of a possible future 2.5 TeV linac. The program can be applied to a broad range of problems that vary widely in energy and beam parameters. Interested scientists are explicitly invited to join the LIAR project and to contribute new features (commands).

2 DISTRIBUTION

The LIAR code is put into the public domain and can be used and distributed freely. However, we expect that publications that contain LIAR results make proper reference to the user's guide [1]. In addition we ask that any extensions and modifications to this program are made available to the scientific community for free usage.

The most recent information on LIAR is available through its home page on the World Wide Web:

http://www.slac.stanford.edu/grp/arb/rwa/liar.htm

Please check for changes in the User's Manual or use the ONLINE manual with the most recent information.

The LIAR source code and executables are available through its AFS site:

/afs/slac.stanford.edu/public/software/liar/release

This directory is world-readable and can be used to download the LIAR files. The present release version of LIAR is 1.9. New versions and updates will be put into this directory. Bug fixes will generally not result in a new version number. The existing version of LIAR will just be updated. Every 1-2 months, however, a new version will be released that contains all the old features plus the new commands that have been added since the last release. If existing commands are enhanced or significantly changed they will be available under a new name. The original commands with their old functionality will be available with their original names. We thus will try to maintain backward compatibility.

3 PORTABILITY

The LIAR code is mainly written in standard Fortran 77. It, however, takes advantage of the STRUCTURE and RECORD extensions that are available in most Fortran compilers. The code is stand-alone, apart from a few system calls (random number generator, time) that need to be adjusted to the actual computer system. No specific libraries are required for the compilation. LIAR version 1.9 is presently running under those operating systems:

- UNIX: (IBM AIX 3.2, etc.).
  Computer: UNIX RISC workstations.

- Windows 95 / Windows NT: (Personal Computers).
  Computer: > Pentium 133 MHz, > 32 MB RAM.
  Compiler: Microsoft Powerstation Fortran 4.0.

The code is easily ported, as long as Fortran compilers are available that support the STRUCTURE and RECORD extensions.

4 ACCELERATOR PHYSICS IN LIAR

The LIAR program models both single and multi-particle beam dynamics in the presence of strong wakefields and beam acceleration. It is essentially a tracking program that
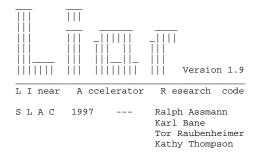
is enhanced by a large number of utility, analysis and optimization routines. Those routines allow adjusting the simulation to a realistic model and estimating the linac performance after the application of optimization routines. The underlying theory and concepts of LIAR are described in the user's manual [1]. Here we give a list of important features:

- Full implementation of quadrupoles, RF structures, dipole correctors, beam position monitors (BPM's), magnet movers and their errors.

- Fully coupled beam transport.

- Single-particle Twiss parameters.

- Consideration of all chromatic and dispersive effects.

- Transverse and longitudinal dipole wakefields.

- Transverse quadrupole wakefields.

- Calculation of wakefield effects within a single bunch (short-range wakefields) and within a bunch train (long-range wakefields).

- Charge sensitive beam transport (e.g. bunch train with electron and positron bunches).

- Calculation of beam offsets, emittances, β-mismatch, effective luminosity reduction, etc. at all BPM's.

- Calculation of full phase space at freely definable Marker points.

- Steering trajectory correction.

- Automatic generation of freely definable position feedbacks.

- General multi-device knobs.

- Automatic emittance bump optimization.

- Advanced beam-based alignment algorithms (dispersion-free steering, alignment using magnet movers, ...).

- Multi-particle beam optics diagnostics to determine beam phase advance, wakefield growth of betatron oscillations, etc.

- Output and plotting routines for all lattice and optics parameters and all observables.

Machine specific parameters are defined interactively by the user (beam current, bunch length, ...) or are defined in external input files (lattice, wakefield functions, ...). Though the required input format is fixed, the names of input files are chosen by the user. Lattices can be specified in the TRANSPORT or MAD format.

5 EXAMPLE

If LIAR is called without command line parameters it enters into interactive mode:

```
  |||        |||
  |||        |||
  |||        |||   _____     ___
  |||        |||  _||||||    _|||
  |||        |||  |||_||_     |||
  |||___     |||  |||_||_     |||
  |||||||||  |||  |||||||||   |||      Version 1.9
  _____
  L I near    A ccelerator   R esearch  code

  S L A C    1997    ---    Ralph Assmann
                            Karl Bane
                            Tor Raubenheimer
                            Kathy Thompson


LIAR>
```

and commands can be entered. Alternatively LIAR can be run in batch mode, reading its input from a file. Let's consider the example of a simple SLC simulation. We go through in steps. Many of the commands shown here have more parameters than are specified. If parameters are not explicitly given, they are set to their default values. Command lines are automatically continued if an input line is ended with a comma.

In the first two commands we initialize all internal data structures to zero, we switch off the debug level and we set the output unit to the standard output:

```
RESET, all
SET_CONTROL, debug   = 0,
             outlun  = 6
```

Then we read the lattice from a transport input deck and define the injection energy:

```
READ_TRNS, infile  = 'input/slc.trns',
           energy  = 1.19
```

In the next step the RF-phases for BNS damping are defined and the acceleration and lattice is rescaled accordingly. Note that two different RF-phases are specified along the linac with the switch at 800 m:

```
SET_RF, energy   = 46.6,
        scale    = .t.,
        phase1   = 22.,
        lswitch1 = 800.,
        phase2   = -16.5
```

Next we calculate the Twiss parameters, specifying the initial values:

```
CALC_TWISS, betax   = 3.40,
            betay   = 3.08,
            alphax  = 0.156,
            alphay  = 0.066
```

After that we need to define the initial conditions for the beam setup. We define an initial beam offset y, the injection energy and energy spread, the normalized emittances and the Twiss values:

```
SET_INITIAL, y        = 200.d-6,
             energy   = 1.19,
             espread  = 0.014,
             nemitx   = 3.0d-5,
             nemity   = 0.3d-5,
             betax    = 3.40,
             betay    = 3.08,
             alphax   = 0.156d0,
             alphay   = 0.066d0
```

Everything is prepared now to set-up the beam. We specify the bunch population, the bunch length, the number of bunches, the number of slices per bunch and the number of mono-energetic beam ellipses per slice:

```
SET_BEAM, current = 4.d10,
          blength = 1100.d-6,
          nb      = 1,
          ns      = 20,
          nm      = 3
```

Next we must define the wakefields. They are read in from external input files. We only define short-range wakefields:

```
SET_SR_WF, file = 'input/srwf_slc.dat'
```

Before we track the beam through the lattice, we misalign all quadrupoles vertically by 200 μm RMS:

```
ERROR_GAUSS_QUAD, name     = '*',
                  y_sigma  = 200.e-6
```

Now we can finally track the beam through the lattice:

```
TRACK
```

and measure the normalized beam offsets at the BPM's:

```
MEAS_BPM, file = 'output.data',
          norm = .t.
```

The output is saved into a file. As already mentioned, many more parameters and commands are available in LIAR. For example the command TRACKC would have tracked the beam through the lattice while applying a 1-to-1 trajectory correction. The complete reference information is available in the LIAR manual [1].

During the execution of the commands LIAR provides extensive information. For example the progress of the tracking is indicated by a "trackometer":

```
T R A C K O M E T E R :
~~~~~~~~~~~~~~~~~~~~~~~~~
0   10  20  30  40  50  60  70  80  90  100 %
|____|____|____|____|____|____|____|____|____|____|
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

At the end of the tracking, summary information is printed to the standard output. We just show the first few lines:

```
End of tracking   A N A L Y S I S :
-----------------------------------

Beam energy
  - acceleration:   E_0  =    1.190 GeV  -->  E_f   =   45.998 GeV
  - spread:         E_sig =    .016 GeV  -->  E_sig =     .126 GeV
  - rel. spread     SIGE/E =   1.322 %   -->  SIGE/E =    .274 %
Beam blow-up
  - Emittance (b1): g_x  =    0.000 %          g_y   =  205.158 %
```

The observables are defined in [1]. We just mention a few of the summary results: emittance growth in the linac, lattice mismatch, RMS trajectory offsets, beam energy, ... In addition most of the results are saved at every BPM location. They can be printed out into files, for example in order to study the emittance growth along the linac.

6 SUMMARY

The computer program LIAR ("LInear Accelerator Research code") is a numerical modeling and simulation tool for high performance linacs. Amongst others, it addresses the needs of state-of-the-art linear colliders where low emittance, high-intensity beams must be accelerated to energies in the 0.05-1 TeV range. LIAR is designed to be used for a variety of different projects. It has been applied to and checked against the existing Stanford Linear Collider (SLC), the linacs of the proposed Next Linear Collider (NLC) and the proposed Linac Coherent Light Source (LCLS) at SLAC.

LIAR allows the study of single- and multi-particle beam dynamics in linear accelerators. It calculates emittance dilutions due to wakefield deflections, linear and non-linear dispersion and chromatic effects in the presence of multiple accelerator imperfections. Both single-bunch and multi-bunch beams can be simulated. It is possible to simultaneously study the acceleration of positive and negative charges in a linac. Diagnostic and correction devices include beam position monitors, RF pickups, dipole correctors, magnet movers, beam-based feedbacks, multi-device knobs and emittance bumps. Several basic and advanced optimization schemes are implemented. Present limitations arise from the incomplete treatment of bending magnets and sextupoles.

A major objective of the LIAR project is to provide an open programming platform for the accelerator physics community. We invite interested scientists to join this project. Due to its design, LIAR allows straight-forward access to its internal FORTRAN data structures. The program can easily be extended and its interactive command language ensures maximum ease of use. Presently, versions of LIAR are compiled for UNIX and MS Windows operating systems. An interface for the graphical visualization of results is provided. Scientific graphs can be saved in the PS and EPS file formats. In addition a Mathematica interface has been developed. LIAR now contains more than 40,000 lines of source code in more than 130 subroutines.

In the future we will add the ability to include bunch compression into the simulation.

7 REFERENCES

[1] R. Assmann et al., "LIAR - A Computer Program for the Modeling and Simulation of High Performance Linacs", SLAC/AP-103 (1997).