Particle Accelerators, 1992, Vols. 37–38, pp. 429–437
Reprints available directly from the publisher
Photocopying permitted by license only

HILDA: HEAVY ION LINAC ANALYSIS CODE†

E. CLOSE, C. FONG and E. LEE

Lawrence Berkeley Laboratory, University of California,
1 Cyclotron Road, Berkeley, CA 94720 USA.

(Received 3 December 1990)

HILDA is a program which estimates the cost and finds an optimal design for HIF induction linac drivers. It can model near-term machines as well as full-scale drivers. Code objectives are: (1) A relatively detailed, but easily understood model. (2) Modular, structured code to facilitate making changes in the model, the analysis reports, and the user interface. (3) Documentation that defines, and explains the system model, cost algorithm, program structure, and generated reports.

This report describes the structure and features of HILDA. We illustrate its potential by generating a minimum-cost estimate of an intermediate (30-kJ) facility. The cost estimates are based on near-term technology and prices.

1 INTRODUCTION

A Heavy Ion Fusion Driver (HIFD) is a complex machine consisting of many components. In particular, an HIFD that is a linear induction accelerator could consist of an injector, an induction linac, a drift-compression section, and then a final focus section just prior to the reactor chamber. A cost estimate can be obtained using standard engineering cost estimation techniques and tools such as spreadsheets. However, obtaining results in this way is laborious and time-consuming. Thus it is not practical to compare the costs of many machines based on different design choices when the cost is obtained in this manner. Also, the complexity of performing the calculation makes it difficult to determine what information was used. Ideally we wish to have a tool that permits easy estimation of the costs of many different HIFD designs so that the design which yields an optimal cost can be chosen. HILDA is being developed to provide such a tool.

2 PROGRAM STRUCTURE

HILDA as a program is a closed loop driver. See Figure 1. This very simple but general structure allows expansion and changes to be made in a simple manner. The program asks the user what task to perform, performs that task using the required processes, and then asks the user what to do next. This structure lends itself naturally to a

---

† This work was supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, U.S. Department of Energy under contract No. DE-AC03-76SF00098.

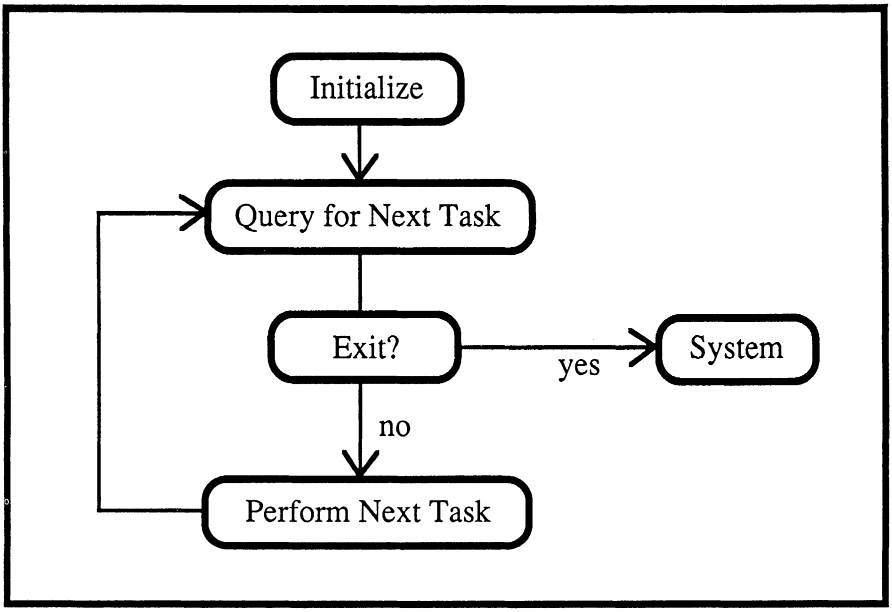

FIGURE 1 HILDA program structure.

menu-driven program that can interact with the user in a friendly, useful manner. Also, processes that perform elementary tasks may be grouped together in order that more complicated tasks can be performed with only one menu selection.

HILDA is being developed on a VAX system using a Macintosh interface. The program and all sub-modules are written in VAX FORTRAN 77. There are exceptions to the language standard; for example, variable names are not limited to 6 characters. However, the exceptions are few and adherence to a standard computer language should allow HILDA to be run easily in many computing environments. FORTRAN was chosen because of its wide use in scientific computing.

All sub-program modules ("subroutines" in FORTRAN) are considered to be, and are referred to here as, processes. Each process is passed a message to perform a task. If the process can perform the requested task, then the values of the process input variables are used to carry out the request. When the process is finished, the output variable values are returned, along with an appropriate exit message. Required data is read from an associated data file and, when requested, an associated output file is written. Thus, the processes in HILDA consist of a body of code, a set of input variables, a set of output variables, associated data and output files. A process is driven by the message it receives when it is invoked. To a large degree, each process in HILDA is a stand-alone module. No use is made of common, or shared, memory blocks ("COMMONs" in FORTRAN). See Figure 2.

This process structure incurs some extra overhead for the programmer. It is necessary to very carefully structure the task or tasks that a process performs. It is

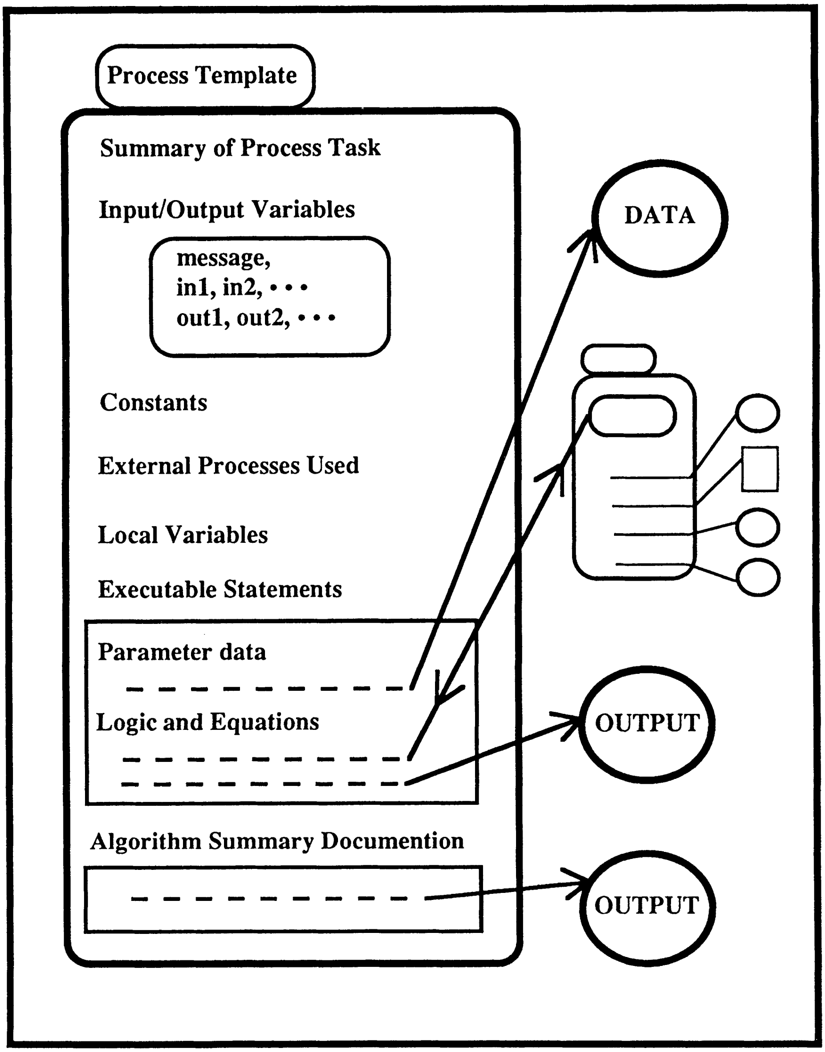

FIGURE 2 HILDA process structure.

also necessary to carefully isolate the input variables, the output variables, the needed data files, and the desired output files. However, once this has been done there are significant advantages. Many of the processes can be run separately and individually pre-tested. This permits us to check not only the correctness of the code logic but also the correctness and applicability of the models used for basic component design. They can also be used separately to explore and define appropriate design parameter regions. If it is desirable or necessary, they can be run as parallel processes. The processes are to a large extent "plug-in" units that can be updated or replaced as the model changes. HILDA processes are built from a template to insure a standard structure, which also helps in producing good documentation.

As mentioned, one of the design goals is that the model and algorithms used to obtain a cost estimate be accessible for review. Hence, a programming design choice was made that the HILDA program document (the write-up) should in itself define HILDA and its processes. Therefore, as each process is created and inserted into HILDA it is actually placed in the program document. It is then downloaded from the document into the computing environment, compiled, and linked to produce the executable code. That is, the document generates the code. This requires programmer discipline, but a consistent adherence to this convention insures that HILDA is documented as it is created. It insures that the processes in the document are actually executable. It also allows us to use the extensive formatting capabilities of modern word processors. As a result, good documentation of the processes and algorithms is produced.

3 DATA STRUCTURE

For a cost estimate to be understood and believed, it is necessary that the data used in designing the components be readily available. For example, the process that designs an iron-core focusing magnet needs knowledge of the types of materials available, along with their densities and costs. Thus, one of the HILDA design goals is that the design data be on files that are accessible and readable both by people and by HILDA I/O utilities. It is also necessary that we can change this data, and preferably without modifying the actual process that uses the data. Hence, the data files are ASCII text files. The information in these files can easily be updated using a text processor, or by small HILDA I/O utilities. If it should be desirable to do a cost study in which this data is actually the variable of interest, then small modules can be provided to automatically update the files and then invoke the cost optimizing algorithms. These data files are also available for generating reports that define the optimized design. The use of message driven processes permits us to read these files only when the data is needed, e.g., at the beginning of the search for an optimum and not during the actual optimization process.

4 OUTPUT FILES

As noted above, a processor can generate an associated output file. This file contains all the input variable values, all the output variable values, all the associated data information and all appropriate intermediate values. For example, the process that designs the beam acceleration modules returns only the dollar cost of the module. However, the output file contains all the information needed to define that module: the size of the components, their weights and individual material costs, the materials used, etc. When appropriate, each process can also write a summary file of the algorithms it uses. These output files can then later be used to generate a report that presents the total design of an optimum-cost HIFD. These files are also ASCII text files that are readable both by people and by the HILDA utility routines. The use of message driven processes allows us to write the output files only when desired, e.g., after an optimum has been determined and not during the optimization process.

5 BEAM TRANSPORT/ACCELERATION OPTIMIZATION ALGORITHM

An HIFD that is based on a linear induction accelerator is shown schematically in Figure 3. We have, in the example below, asked for a minimum-cost design (MCD) of the focusing quadrupoles and of the acceleration modules in the induction linac section. It is not our intent in this short paper to present in detail the algorithms that are used. The HILDA document is meant to provide that information. We will simply sketch how the program determines such a design.

For the machine station at which we wish to find a MCD, we furnish the ion mass A [amu], ionization state q, total beam charge $Q[C]$, number of beams N, cumulative voltage in V[MV], undepressed tune σ_0[deg], and normalized emittance $\varepsilon[m - r]$. We then specify a parameter grid over which a search will be conducted. A point in this grid defines the local variables: voltage gain $\Delta V[V]$, beam radius a $[m]$, focusing lattice half-period $L[m]$, focusing packing fraction η. Note: in the example the beam radius is fixed to one beam size.

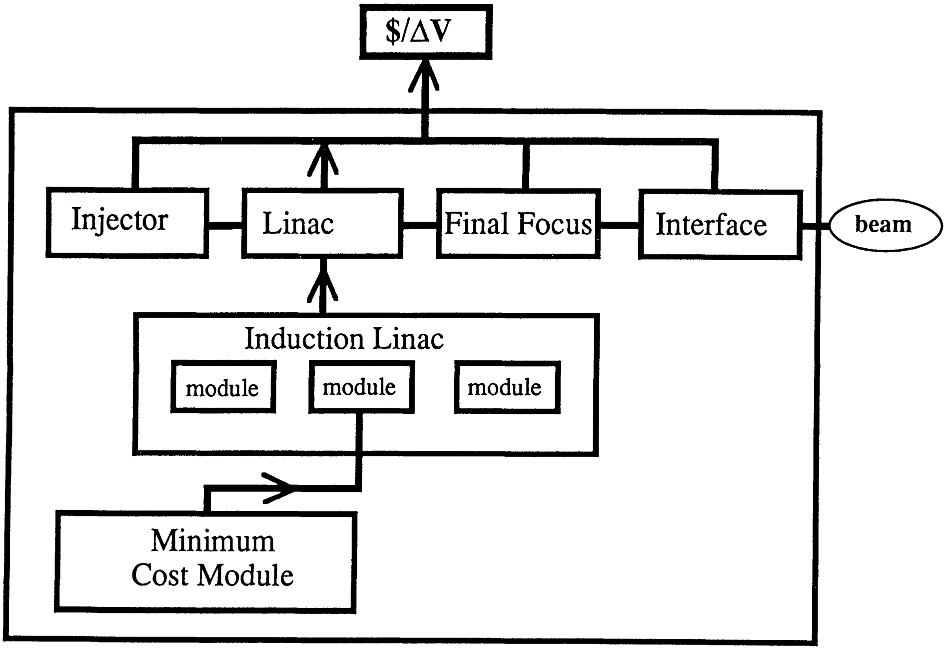

FIGURE 3 HILDA estimates the induction linac module cost.

For the given set of station variables $(A, q, Q, N, V, \sigma_0, \varepsilon)$ and at each point $(\Delta V, a, L, \eta)$ of the parameter grid, HILDA calculates dynamical quantities and determines whether they meet design criteria for beam transport. If it is determined that the beam cannot be transported, then a message to that effect is issued and the next point in the grid is examined. If the beam can be transported, HILDA proceeds to design a focusing quadrupole for the beam. Certain design constraints must be met. For example, the quadrupole must fit in the space available and the focusing fields must be of reasonable strength. If such constraints cannot be met, then a message is issued and HILDA skips to the next point in the grid. If a quadrupole can be designed, the quadrupole beam spacing and size along with its cost are returned and HILDA proceeds to use the spacing and size in the design of an acceleration module. Again, component design limits are observed. For example, the core winding height to width ratio should not be too large, pulser voltage limits should be observed, the quadrupole should fit into the acceleration module, etc. A failure to find an acceptable design causes a message to be issued and a new grid point to be examined. If an acceleration module can be found, then its cost is returned. HILDA records the total cost of the components obtained at the current parameter grid point. It then proceeds to examine the next grid point. The process described above is repeated for all grid points.

Each time a cost is found, it is recorded. At the end of the scan HILDA returns to the grid point producing the MCD and uses those parameters to recalculate the design. For this MCD it records in an appropriate log file: the cost obtained, and the values of all intermediate input, output, design data, and calculated results. Reports can be generated using the data in this file. If desired HILDA can also record in this file summaries of the process algorithms used in the design, i.e. the model equations that produced this design. This file provides an audit trail that allows for independent verification and checking of the obtained design. This audit trail also permits the user to check the data and design criteria actually used; it is all too easy to accidentally furnish an incorrect value for some particular data item.

6. A 30 kJ NEAR-TERM MACHINE

As an example we use HILDA to obtain MCD estimates of a 30-kJ near-term machine. This is an intermediate facility intended for technology development and study of beam physics issues, not a power driver. Hence, it has a low repetition rate (0.1 Hz), moderate-life-span components (10^6 pulses), pulsed iron quadrupole magnets (1.0 ms) for beam focusing, and beam currents that lightly load the pulsers (≤ 20 A/beam). The cost estimates are based on near-term technology and prices.

The transported beam is singly charged potassium (38.9 amu) with an undepressed tune of 72 degrees and a normalized emittance of 2.89×10^{-6} meter-radian. For the length of the machine there are 21 beams and the beam radii are fixed at 8.0 cm. The total beam charge is 0.0003 C. We choose 5 stations starting at 6.5 MV and ending at 100 MV. For each of these stations HILDA finds parameters that give a MCD for the focusing magnets and the acceleration modules.

TABLE 1

Costs at Minimum Point of the Scan Grid: Station 1–5

Station		1	2	3	4	5
Cost	[$/ΔV]	2.18E + 00	1.25E + 00	7.25E - 01	3.38E - 01	2.69E - 01
Cumulative volt.	[MV]	6.50E + 00	2.41E + 01	5.51E + 01	7.91E + 01	1.00E + 02
Voltage increment ΔV	[V]	2.00E + 05	2.80E + 05	6.00E + 05	2.00E + 06	3.00E + 06
Half period	[m]	9.00E - 01	1.60E + 00	1.80E + 00	1.99E + 00	2.09E + 00
Packing factor	[ ]	5.00E - 01	5.00E - 01	4.00E - 01	5.00E - 01	5.00E - 01

TABLE 2

Associated Calculated Values: Stations 1–5

Station		1	2	3	4	5
Charge per beam	[C]	1.428E - 05	1.428E - 05	1.428E - 05	1.428E - 05	1.428E - 05
Beta * gamma	[ ]	1.892E - 02	3.648E - 02	5.514E - 02	6.607E - 02	7.433E - 02
Gamma mc^2/m_0c^2	[ ]	1.000E + 00	1.000E + 00	1.001E + 00	1.002E + 00	1.002E + 00
Beta v/c	[ ]	1.892E - 02	3.645E - 02	5.506E - 02	6.593E - 02	7.413E - 02
Mag. rigidity	[T-m]	2.291E + 00	4.415E + 00	6.675E + 00	7.998E + 00	8.998E + 00
Depressed tune	[deg]	7.202E + 00	3.983E + 00	3.016E + 00	2.844E + 00	3.107E + 00
Ave. beam size	[m]	6.218E - 02	6.220E - 02	6.167E - 02	6.114E - 02	6.167E - 02
Perveance		1.632E - 03	5.205E - 04	4.048E - 04	3.224E - 04	2.973E - 04
Beam current	[A]	6.700E + 00	1.529E + 01	4.109E + 01	5.630E + 01	7.394E + 01
Pulse width	[s]	2.368E - 06	1.037E - 06	3.862E - 07	2.819E - 07	2.146E - 07
Acc. gradient	[V/m]	2.222E + 05	1.750E + 05	3.333E + 05	1.000E + 06	1.428E + 06
Volt seconds	[V-s]	4.737E - 01	2.905E - 01	2.317E - 01	5.638E - 01	6.439E - 01

TABLE 3

Quadrupole Design Parameters: Stations 1–5

Station		1	2	3	4	5
Cost, quad array	[$]	2.438E + 05	2.429D + 05	3.027D + 05	3.298D + 05	3.497D + 05
Fe, quad array	[kg]	7.390E + 03	7.363D + 03	9.174D + 03	9.996D + 03	1.059D + 04
Beam spacing	[m]	3.463E - 01	3.073D - 01	3.330D - 01	3.169D - 01	3.183D - 01
Vacuum pipe wall	[m]	3.300E - 03	3.300D - 03	3.300D - 03	3.300D - 03	3.300D - 03
Cooling, etc.	[m]	1.100E - 02	1.100D - 02	1.100D - 02	1.100D - 02	1.100D - 02
Wire center	[m]	1.268E - 01	1.258D - 01	1.265D - 01	1.261D - 01	1.261D - 01
Wire thickness	[m]	5.167E - 03	3.125D - 03	4.473D - 03	3.630D - 03	3.706D - 03
Iron thickness	[m]	4.370E - 02	2.622D - 02	3.773D - 02	3.052D - 02	3.177D - 02
Aperture radius	[m]	1.100E - 01	1.100D - 01	1.100D - 01	1.100D - 01	1.100D - 01
Field gradient	[T/m]	8.144E + 00	4.967D + 00	7.070D + 00	5.758D + 00	5.875D + 00
Pole tip field	[T]	1.033E + 00	6.251D - 01	8.947D - 01	7.261D - 01	7.412D - 01
Iron outer radius	[m]	1.731E - 01	1.536D - 01	1.665D - 01	1.584D - 01	1.591D - 01
Overhang length	[m]	9.516E - 02	9.439E - 02	9.490E - 02	9.458E - 02	9.461E - 02
Magnet iron length	[m]	6.403E - 01	9.887E - 01	9.098E - 01	1.189E + 00	1.239E + 00
Fe vol. of 1 quad.	[m**3]	4.309E - 02	4.294E - 02	5.350E - 02	5.829E - 02	6.180E - 02
Iron density	[g/m**3]	8.165E + 03	8.165E + 03	8.165E + 03	8.165E + 03	8.165E + 03
Fe weight, 1 quad.	[kg]	3.519E + 02	3.506E + 02	4.368E + 02	4.760E + 02	5.046E + 02
Cost, 1 quad	[$]	1.161E + 04	1.157E + 04	1.441E + 04	1.570E + 04	1.665E + 04

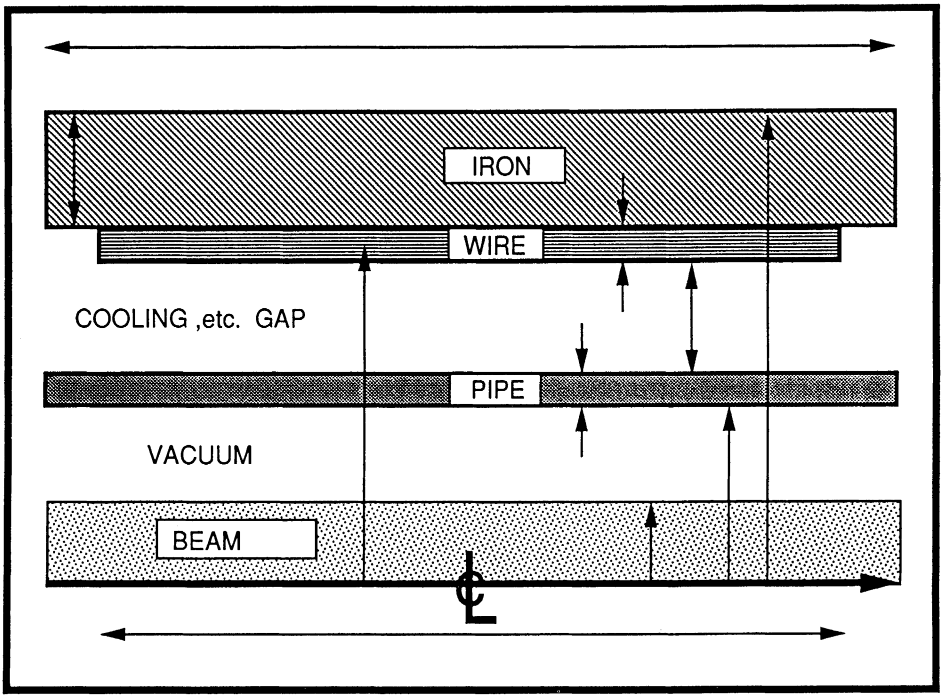

FIGURE 4 HILDA quadrupole.

Results of these calculations are given in Tables 1–3. In Table 1 we show at each station the dollar cost per voltage increment, the corresponding cumulative voltage of the beam, and the values of the search grid parameters that produced this MCD. In Table 2 associated calculated values are shown for each station. In Table 3 we show a table of the quadrupole design parameters for each station. In Figure 4 we show a schematic drawing to which the quadrupole parameters in Table 3 can be related. A design parameter table and associated figure also exist for the acceleration modules, but they are not shown here due to space limitations.

At each station the MCD values were found by calculating the design cost for the parameters (ΔV, a, L, η) taken from a specified grid of allowable values. At some points limits were exceeded and those points were excluded. For example, the magnetic field at the pole tip exceeded 1.5 T. At others a cost was obtained, but it exceeded the cost in the tables. A log file of the run was kept and it would be easy to investigate solutions near the minimum. For example, the tables show that it is reasonable to require the quadrupole packing fraction η to be 0.5 throughout the machine. The MCD value of 0.4 obtained at station 3 is really not necessary and HILDA could easily find the MCD for η fixed at 0.5. The cost would be close to the 0.4 case.

7 CONCLUSION

HILDA is presently a useful tool for cost estimation. As experience is gained and as more details are included in the cost model it should be easy to update the program. A consistent adherence to the design structure of the processes and their associated data and output files along with a persistent effort to keep the HILDA document up to date will produce a program that has a good quality assurance audit trail. This should provide a means for understanding and a basis for believing the results that it produces.