GRAPHICAL USER INTERFACE FOR TRACE 3-D INCORPORATING SOME EXPERT SYSTEM TYPE FEATURES

George H. Gillespie and Barrey W. Hill
G. H. Gillespie Associates, Inc.
P. O. Box 2961
Del Mar, California 92014, U.S.A.

Abstract

A graphical user interface (GUI) has been developed for the beam transport program TRACE 3-D. The interface was developed on the Macintosh personal computer platform and emphasizes ease of initial problem setup and definition. Each of the sixteen types of transport elements (drifts, quadrupoles, etc.) available in TRACE 3-D, as well as the initial and final (for matching) beam emittance (Twiss) parameters, are represented by piece icons on a scrollable palette. The configuration of a beamline is set up visually by selecting and dragging (via mouse) the desired piece icons to a model window. Parameter values (drift lengths, quadrupole strengths, etc.) are entered into piece data windows for each element in the beamline. Several expert system type rules are incorporated into the piece windows. The user can select any of several units for his input, including fixed units or dynamic scaled units. For example, any length parameter can be entered as millimeters, centimeters, meters, or fractions of $\beta\lambda$ where β and λ are determined from the particle mass, initial beam energy and radiofrequency. All input parameters have built-in default values as well as lower and upper limits. The limits are soft (the user can input any value) but are used to alert the user visually when some of his input data may have impractical consequences. Examples of this include specifying a PMQ which requires an extremely high remnant field, or a RFQ cell with a very large Kilpatrick factor. Virtually all other input is set up graphically, including the selection of matching variables and coupling parameters.

I. Introduction

There has been considerable progress in the development of graphical user interfaces (GUIs) for accelerator control systems [1]. Selected accelerator design software has benefited from this, and improved user interfaces have appeared for codes used in conjunction with control systems. Examples include the X-window interfaces for TRACE 3-D [2] and RESOLVE [3]. From the beamline designer's standpoint, however, there have been relatively few efforts aimed at developing GUIs specifically to support standalone design and analysis. This paper describes a software package developed with this objective in mind. The approach is similar to that suggested by Heighway at the 1988 Accelerator Code Conference [4].

II. The Shell for Particle Accelerator Related Codes (SPARC) Environment

To support GUIs for accelerator design and analysis codes a unique interface environment has been developed. This interface is written in C and provides a software shell for each application such as TRACE 3-D. Named the Shell for Particle Accelerator Related Codes (SPARC), it includes the basic elements to support a GUI: specialized windows, palettes, menus, icons, etc. Figure 1 shows the SPARC application screen developed for TRACE 3-D. This example displays the Menu Bar, Palette Bar, Document Window and, on the Model Pane of the Document Window, a beamline model for the radiofrequency quadrupole (RFQ) section of Example A in the TRACE 3-D Documentation [5].

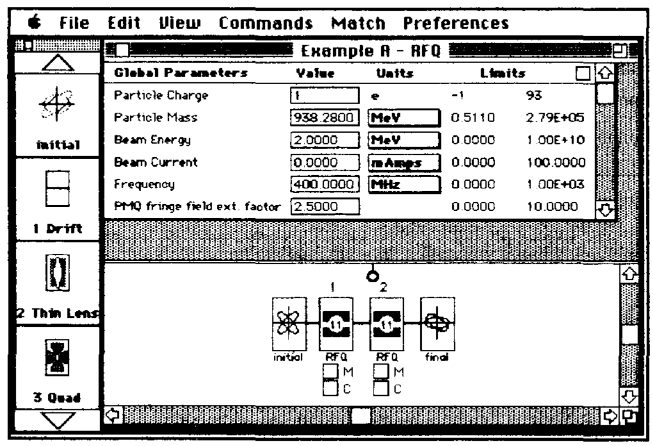

Figure 1. Example of SPARC screen for RFQ section of Example A of reference [5].

The SPARC interface has a number of important features which improve the speed and ease of setting up and defining a TRACE 3-D problem. These are the focus of this paper and several are discussed below. Execution of TRACE 3-D is accomplished directly from SPARC via the Commands menu.

III. Beamline Set Up

Setting up the input file for TRACE 3-D using the SPARC interface is simple. The configuration of the beamline is defined by selecting (with the mouse) the desired elements from the Palette Bar and dragging them to the Model Pane of the Document Window. Groups of elements which will be used more than once may be selected and placed on the Work Space, and then inserted into the beamline. Figure 2 shows a Document Window after setting up the drift tube linac (DTL) section of Example A in the TRACE 3-D Documentation [5]. This example also shows two groups of elements on the Workspace.

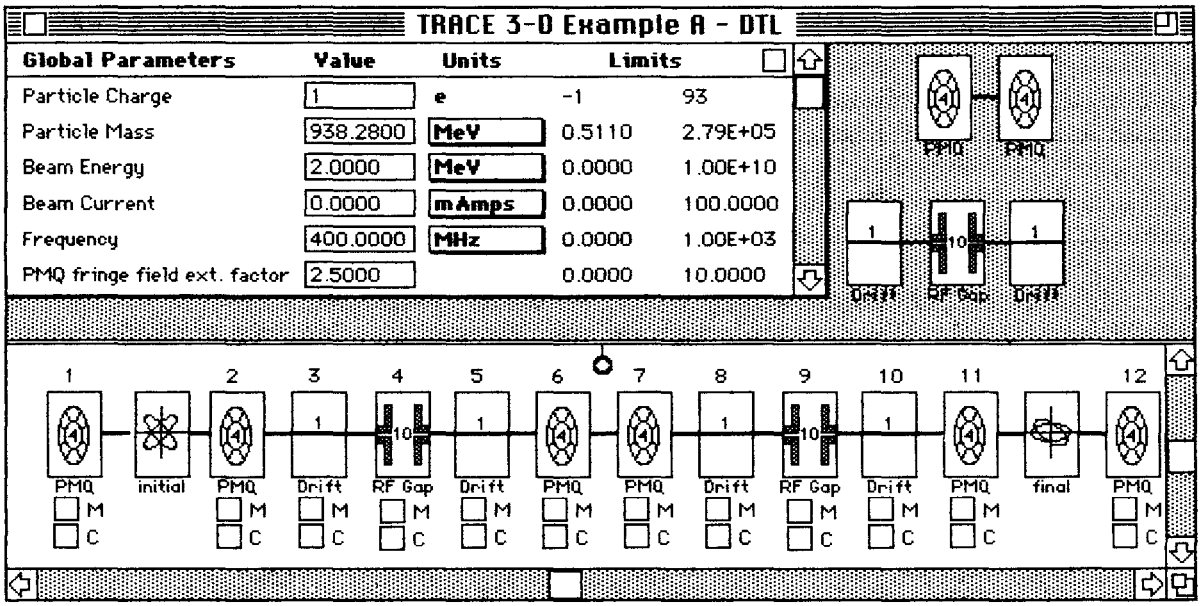

Figure 2. SPARC Document Window for DTL section of Example A of reference [5].

For inputting numerical data the TRACE 3-D input has been divided into four categories:

- Global Parameters,
- Piece Parameters,
- Matching Parameters, and
- User Preferences.

There are ten Global Parameters which include the particle charge, particle mass, initial beam energy, etc. The corresponding TRACE 3-D parameters are Q, ER, W, XI, FREQ, PQEXT, ICHROM, SMAX, PQSMAX, and IBS. These parameters are input through the Global Parameter Pane in the Document Window (Figure 2). Many parameters have pop-up menus for selecting different units. For example the user may input the particle mass in MeV, or in atomic mass units (amu-real), or to the nearest atomic mass integer value (amu-int). The SPARC interface stores all inputs in the units used by TRACE 3-D, these are the default units which appear on start-up, but it has the needed conversion factors built-in to accommodate individual user preferences. One of the expert system type features incorporated into SPARC is that some units selections depend upon the values of the Global Parameters. An example is discussed in the abstract: the ability to input length parameters, such as SMAX and PQSMAX, in units of $\beta\lambda$. These are referred to as "smart units" and are used in many other input windows.

The Piece Parameters include the transport parameters used by TRACE 3-D (arrays NT and A), the initial beam characteristics (arrays BEAMI, EMITI and SIGI), and the final beam characteristics used for matching (BEAMF array). These parameters are accessed by "double clicking" on the icons appearing in the Document Window. Figure 3 shows the input window for setting up the initial beam parameters. This Piece Window includes another example of the smart units: the longitudinal emittance can be input in either π-degree-keV or π-mm-milliradian units, and either as rms or the equivalent uniform values. Also note the "Ellipse Display" pop-up menu. This provides an option, "Update," which automatically displays the results of changes to any of the parameters. This is used in SPARC to set initial values for the graphics display parameters (XMI, XPMI, etc.) used by TRACE 3-D.

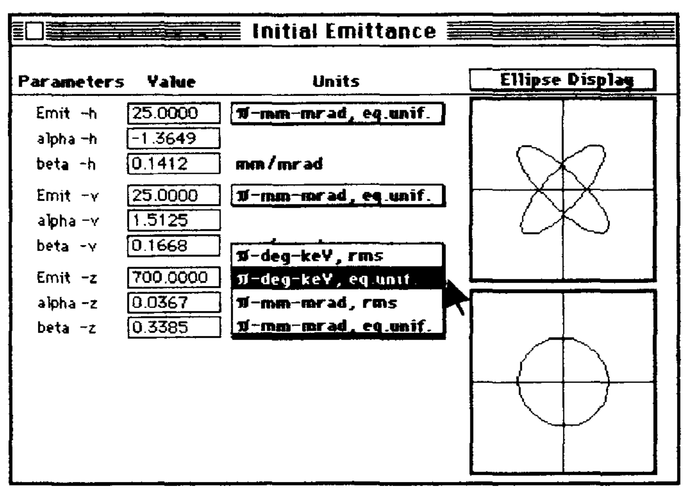

Figure 3. Piece Window for initial beam set up, accessed by double-clicking the "initial" icon.

IV. Setting Up Matching Problems

The TRACE 3-D matching options may be divided into two groups:

- Finding Matched Twiss Parameters, or
- Finding Variables to Achieve a Match.

There are four different options of the first type and seven of the second. The match option is determined by the TRACE 3-D match type parameter MT. In the SPARC interface this parameter is set by selecting one option from a pull down menu (the Set Match Type submenu under the Match menu).

For MT = 1 to 4 (finding matched Twiss parameters) this is the only action necessary to set up the matching problem after the beamline is defined. The interface sets the number of matching conditions (NC parameter of TRACE 3-D) and has built in defaults for the number of iterations and convergence criteria (NIT and DELTA). These may be changed by the user, but it is not usually necessary.

For MT = 5 to 11 the user must specify the beamline element parameters which are to be varied by TRACE 3-D. These matching variables are selected using the Piece Windows. Each Piece Window, for any element which contains a parameter that can be varied during matching, has a "Match/Couple" option on the Limits pop-up menu. Selecting this option displays check boxes for each parameter which can legitimately be varied during matching. Checking a box (via mouse) assigns that parameter to be a matching variable. This is illustrated in Figure 4 for a PMQ. SPARC automatically sets up the two-dimensional MP array (for MT = 5 to 9), or the VAL and two-dimensional IJM arrays (for MT = 10 and 11), needed by TRACE 3-D to perform the match. Coupling parameters are set up similarly and SPARC sets the MVC array.

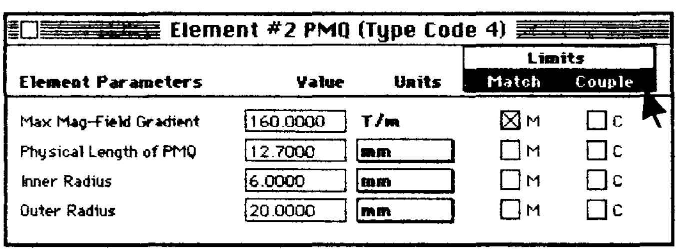

Figure 4. Selecting matching variables in PMQ Piece Window.

V. Expert System Type Rules for Limits

For each parameter of the sixteen transport elements available in TRACE 3-D an upper and lower "limit" are provided in the Piece Windows. A knowledge rule base has been developed for calculating the limits [6]. These rules are of two origins:

- TRACE 3-D driven, and
- Practical hardware constraints.

The first type includes some mundane TRACE 3-D constraints, such as requiring the input parameter for any "identical element" (type 16) to lie between the first and last element numbers of the beamline, and others based on step sizes (SMAX and PQSMAX) used in the beam dynamics calculations. Practical hardware constraints are derived from specific accelerator technology. These limits are intended to provide the novice user with some guidance when setting up a beamline and to reduce the number of off-line calculations by advanced users. For example, practical limits on PMQ gradients are directly related to the inner and outer radii of PMQ magnets. This is used to construct rules for the PMQ limits. A baseline set of rules has been developed for specific technologies (e.g. samarium cobalt PMQ). It is anticipated that these rules will be expanded to incorporate other technology options.

VI. Summary

A sophisticated GUI for the beam optics program TRACE 3-D has been developed. The GUI has been integrated with TRACE 3-D to form a seamless application. Problem setup and definition are accomplished with a minimal amount of alpha-numeric (keyboard) input. The GUI takes care of setting up arrays and similar bookkeeping. Expert rules are built in to the data input windows to assist users in assigning parameter values. These include options for parameter units, both fixed and scaled smart units, and limit guidelines for each parameter. The limits are based on a knowledge rule base which incorporates constraints imposed by TRACE 3-D and practical constraints derived from hardware experience.

Acknowledgements

The authors are indebted to James Gillespie for writing the prototype of the SPARC software and for continued assistance during its development. The assistance of the Los Alamos Accelerator Code Group, without which the seamless integration of SPARC and TRACE 3-D would not have been accomplished, is greatly appreciated. The authors also thank Peter Van Staagen for his review of this manuscript.

References

[1] See, for example, S. Kleban and M. Lee, "GENI: A Graphical Environment for Model-Based Control," Nuc. Instr. Meth. Phys. Res. A 293 (1990) 486-490.

[2] U. Raich, "Beam Modeling with a Window-Oriented User Interface," Nuc. Instr. Meth. Phys. Res. A 293 (1990) 450-455.

[3] D. Dobrott, D. Keeley, G. Kolte, Z Mikic, M. Lee, J. Corbett, S. Howry and A. King, "Control System User Interface for Accelerator Commissioning and Operation," Conf. Record of the 1991 IEEE Particle Accelerator Conference, Vol. 2 (1991) 1305-1307.

[4] E. Heighway, "Magnetic Optics Design (A Designer's View of the Beam Transport Environment)," AIP Conference Proceedings No. 177 (1988) 181-203.

[5] K. Crandall and D. Rusthoi, "TRACE 3-D Documentation," Los Alamos National Laboratory Report LA-UR-90-4146 (1990) 92 pages.

[6] G. H. Gillespie and P. K. Van Staagen, "A Knowledge Rule Base for Problem Set Up with the Beam Optics Program TRACE 3-D," to be presented at the 12th Inter. Conf. on Application of Accelerators in Res. and Ind.