ACCELERATOR SYSTEMS OPTIMIZING CODE[†]

C. C. Paulson, A. M. M. Todd, M. A. Peacock, M. F. Reusch, D. Bruhwiler,
S. L. Mendelsohn, D. Berwald, C. Piaszczyk, and T. Meyers;
Northrop Grumman Advanced Technology and Development Center,
4 Independence Way, Princeton, New Jersey 08540-6620, USA.

G. H. Gillespie, and B. W. Hill; G.H. Gillespie Associates, Inc. PO Box 2961, Del Mar Ca. 92014

R. A. Jameson; Los Alamos National Laboratory; PO Box 1663, Los Alamos, N.M. 87545

Abstract. Many accelerator applications need a flexible, rapid, and accurate systems analysis tool capable of performing beamline and subsystem plant optimizations. This paper describes a systems code that has been developed to model ion accelerator systems and perform design evaluation and system level trade studies. The code uses a series of FORTRAN modules incorporated into a graphical user interface (GUI) that constructs the accelerator in a window on a computer screen. The studies provide overall systems level performance and subsystem requirements within the context of a fully integrated device. The code allows for system optimization based on global system parameters, individual element parameters, cost, footprint, or other design constraints. The code has been applied to possible Accelerator Production of Tritium (APT) and Fusion Materials Irradiation Facility (FMIF) design concepts. Algorithms used, flow charts showing the code logic, and examples of code applications will be given.

I. INTRODUCTION

System complexity, coupled with the large capital and operating costs of the Accelerator Driven Tranmutation Technology (ADTT) accelerator facilities, makes optimization with respect to life cycle, reliability and maitainablility, and capital cost essential. This paper describes a systems code the authors have been developing to model ion accelerator systems for design evaluation and for system level trade studies. The code uses a series of FORTRAN modules incorporated into a graphical user interface (GUI) to provide for the "construction" of the accelerator in a window on the computer screen. Three sets of input information, global parameters, element parameters, and user preferences are utilized by the code. Relevant accelerator, beam transport models, and engineering models from earlier systems codes and other work have been adapted and developed within this framework. Separate accelerator and engineering routines are provided for each accelerator element. Each accelerating module has been provided with the capability of generating complete tank and cell designs based on a set of pre-selected geometrical and / or engineering parameters. In addition to tracking the beam and generating / evaluating the physics design, the code provides a complete set of engineering parameters and preliminary costing for the construction of the accelerator. The paper presents the algorithms used and the results obtained for the Los Alamos APT design.

Due to its elegance and ease of use, the Accelerator Systems Model (ASM)[1] initially developed by G. H. Gillespie Associates (GHGA) was chosen to provide the framework for the code. Section 2 discusses the overall structure of the ASM code, and briefly describes a few Grumman component models presently available within the framework. Section 3 illustrates the application of these models to the Los Alamos APT design. Finally, in section 4, we present our conclusions and future plans.

II. ACCELERATOR MODEL SUMMARY

The goals set for the code were: to provide a comprehensive tool for design and analysis of accelerator systems; to treat the engineering requirements of accelerator design on a basis comparable to the beam physics; to develop a basis for the bridging of different levels of accelerator modeling (CAD, analytic, envelope, particles); to reduce the time required for component / system trade studies; shorten training time for new engineers and scientists; and be accessible to non-specialists. The code has capitalized extensively on existing software. Prior ASM focus has been on accelerator and beam transport physics models, and the GUI I/O (Input/Output) structure. Simultaneously, a Grumman systems code, ABSOC[2] (Accelerator Based Systems Optimization Code), had evolved to feature fairly complex engineering models of the beamline systems with less emphasis on beamline physics models. The first step was therefore to incorporate the best of the ABSOC engineering and certain useful physics models within ASM. Thereafter, we began the development of those new models required to com-plete the beamline systems evaluation. A modular approach has been used for the code with each major component described by a group of FORTRAN subroutines. The subroutines are called iteratively, based on the sequence of components set up by the user on, and controlled by, the GUI side of the code. A common I/O structure is maintained for the modules allowing complete interchang-ability. Critical physics and engineering parameters are assigned to vectors which are tracked and

[†] This work is supported by the Northrop Grumman Corp., Los Alamos National Laboratory (DOE contract W-7405-Eng-36), and G.H. Gillespie Associates (LANL contract 9119K0014-9Q).

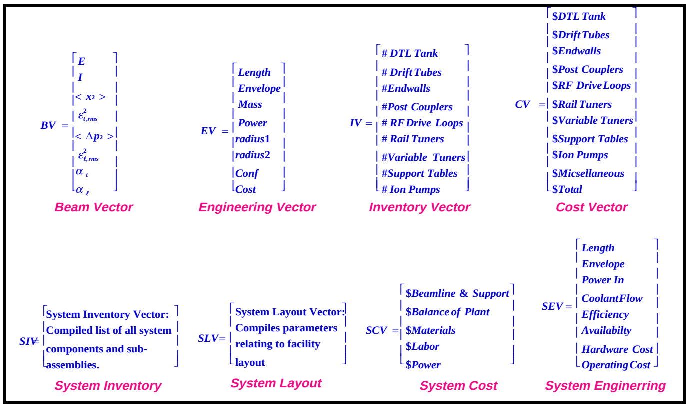
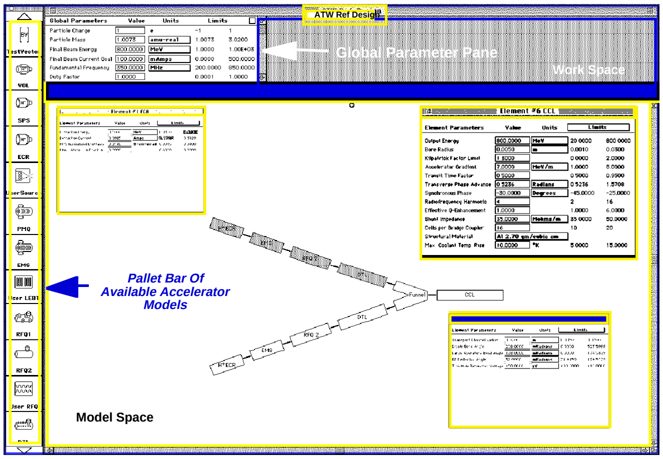


Figure 1: ASM Vectors

updated from element to element. Figure 1 shows a sampling of these vectors. Within each element, the beam is represented by two vectors, the Beam Vector {BV} and the Engineering Vector {EV} which represent the state of the beam and accelerator at that point. Each module generates a BV and an EV as its primary output. In the Beam Vector: E is the beam energy; I, the current; $<x^2>$, the rms transverse size; $<\Delta p^2>$, the square of the momentum dispersion; and α and ε, the Twiss alpha and beam emittance. In the Engineering Vector: L is the device length; V, the device volume; M, the device mass; P, the device power; x and y, the device maximum horizontal and vertical dimension; Conf, a confidence factor for the device as modeled; and Cost, an estimate for the device cost. Internally within the accelerator component models, the design is based on a cell by cell determination of the device. The cell and tank parameters thus calculated are read to off-line arrays for archival purposes and for use in the engineering portions of the model. Secondary vectors which count and cost elements within the device are calculated from the arrays.

In addition to the individual element parameters, there are ten user global parameters available in ASM. These are: Charge; Mass, Final Energy, Final Current goal; Fundamental Frequency; duty factor; operating time; temperature regime; Magnetic Quadrupole material; and Structural material. These parameters are available in the parameter window associated with the accelerator building. The first seven are numerical inputs and the final three have a finite set of choices. The code provides flexibility in the design of each of the elements by allowing different definitions. One such example is the provision for a beam {σ_{Lo}, σ_{To}} or a field description {Eacc, BPT or Bgrad}of each of the accelerator components. The phase advances of the beam definition may be input in the device parameter window or derived from the previous element with a Matching Section element. Similarly, the lattice structure of each device {DTL, BC-DTL, CC-DTL, CCL, and SCL} may be input or derived from the phase advance per unit length of the previous element. When derived within the matching section element, the lattice structure is obtained by: using the maximum pole tip field allowed and calculating the minimum length magnet that will fit within the size constraints of the following element to determine the minimum intertank spacing; and requesting the maximum number of accelerating cells while maintaining beam stability. Tanking within each device, may be specified as specific energy values, or calculated by requiring equal beam energy gains, power dissipation, cell number, or tank lengths. Inter tanks spacings may be derived from the matching section, specified lengths, or specified multiples of the local cell length.

In both the DTL and BCDTL routines, the beam is characterized by the synchronous particle and the cell lengths are calculated from the local beta. These calculations are relatively straightforward. CCL tanks are characterized by the parameters of the particle at the beam center in the tank center (cell lengths calculated from the tank average beta). Magnetic and cell lattice values for the BCDTL and CCL may by entered as input parameters or derived beam matching requirements.



Global Parameters	Value	Units	Limits	
Particle Charge	1	e	-1	1
Particle Mass	1.0073	amu-real	1.0073	3.0200
Final Beam Energy	800.0000	MeV	1.0000	1.00E+03
Final Beam Current Goal	100.0000	mAmps	0.0000	500.0000
Fundamental Frequency	350.0000	MHz	200.0000	850.0000
Duty Factor	1.0000		0.0001	1.0000

Element Parameters	Value	Units	Limits	
Output Energy	800.0000	MeV	20 0000	800 0000
Bore Radius	0.0050	m	0.0010	0.0300
Kilpatrick Factor Limit	1 3000		0 0000	2.0000
Accelerator Gradient	7.0000	MeV/m	1.0000	8.0000
Transit Time Factor	0 5000		0 5000	0.9500
Transverse Phase Advance	0 5236	Radians	0 5236	1.5708
Synchronous Phase	-30.0000	Degrees	-45.0000	-25.0000
Radiofrequency Harmonic	4		2	16
Effective Q-Enhancement	1.0000		1.0000	6.0000
Shunt Impedance	35.0000	Mohms/m	35 0000	50.0000
Cells per Bridge Coupler	16		10	20
Structural Material	Al 2.70 gm/cubic cm			
Max Coolant Temp Rise	10.0000	°K	5 0000	15.0000

Figure 2: Graphical User Interface

The GUI for the ATW reference design is shown in Figure 2. The window is divided into two sections, a work space in which the global parameters are entered and a model space in which the accelerator is built. The accelerator elements are obtained by dragging them from the pallet bar on the left. Element parameters may be entered into individual windows for each of the elements. Three element parameter windows are shown superimposed upon the model space. These windows are opened by pointing and double clicking on the element. The accelerator is designed in the GUI model space. The element parameters along with the analysis, lattice, and tanking methods are entered using the element windows. The program is then started through a pull down menu. The code initializes all parameter arrays and steps through the modeling and optimizing of each element. Calculated parameters for each cell of each accelerating cavity are stored in arrays such as: DTLValues(Itank, Icell, Array). Itank refers to the DTL tank number. Icell refers to the cell number within the tank. Array refers to the value being saved. The current APT configuration calls for 342 tanks and thus could impose severe memory problems if maintained internally. To control the sizes of these arrays, we presently store the results in binary files each time the maximum for any array index is reached. The code architecture is shown in figure 3.

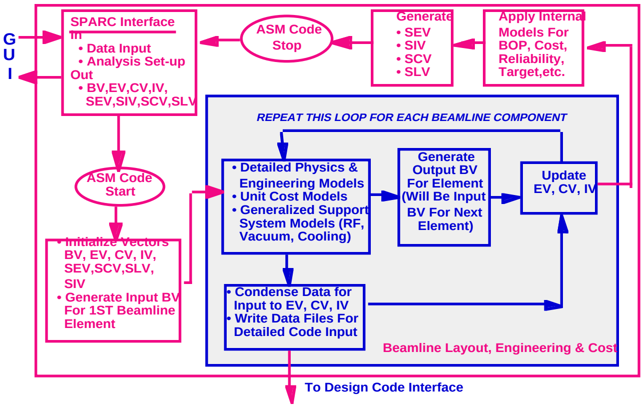
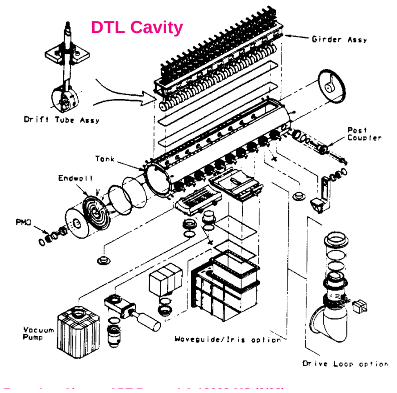

G
U
I
SPARC Interface
In
• Data Input
• Analysis Set-up
Out
• BV,EV,CV,IV, SEV,SIV,SCV,SLV
ASM Code Stop
Generate
• SEV
• SIV
• SCV
• SLV
Apply Internal Models For BOP, Cost, Reliability, Target,etc.
REPEAT THIS LOOP FOR EACH BEAMLINE COMPONENT
ASM Code Start
• Detailed Physics & Engineering Models
• Unit Cost Models
• Generalized Support System Models (RF, Vacuum, Cooling)
Generate Output BV For Element (Will Be Input BV For Next Element)
Update EV, CV, IV
• Initialize Vectors BV, EV, CV, IV, SEV,SCV,SLV, SIV
• Generate Input BV For 1ST Beamline Element
• Condense Data for Input to EV, CV, IV
• Write Data Files For Detailed Code Input
Beamline Layout, Engineering & Cost
To Design Code Interface

Figure 3. Beamline Code Analysis Flow

in addition to generating the beamline design, the code evaluates it based on physics and engineering desires. This evaluation has been included on both the beam and engineering vector level and on the device level. Automatic iteration has been included for the beam and engineering parameters and trade studies based on these elements are presently underway. On the device level, the code determines the availability of required components for each design. One example of this is for RF power sources. An extensive data base of tube characteristics has been provided in the code to determine the most efficient manner to provide power to each accelerating tank. Figure 4 illustrates the engineering modeling objectives of the code.



Modeling Objectives

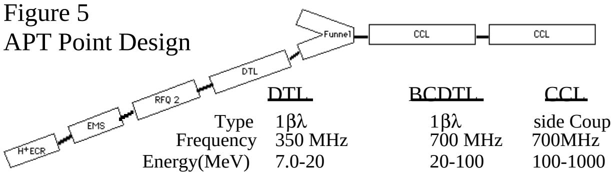
- Model all major components of each beamline element
- Model all major support system interfaces
- Provide data useful to detailed design codes
- Provide inventory & unit costs for major components
- Provide reliability & performance data

Figure 4. Beamline Modeling Objectives

3. APT DESIGN

Many of these models are already yielding good agreement with the baseline APT design (Figure 5). For instance, using the equal length option set at 250 cm and using a ramp of 500 cm for phase and electric fields, the code yielded a DTL consisting of 3 tanks of 23, 20, and 16 cells. This is within 1 cell of the actual design. The DTL total length was 8.1 m long (≈ 1% longer than the design; had a total power dissipation of 2.3 MW (≈6% low) and an output phase advance of 72.7°. These results are in excellent agreement with the published results. The differences are easily attributable to the ramp length and cylinder size. The present BCDTL module yields an estimate for the length within 2% and the total code yields an estimate for the entire system within 1.5%

Figure 5
APT Point Design



RFQ
0.075 to
7.0 MeV
8.1 m
433 cells

	DTL	BCDTL	CCL
Type	1βλ	1βλ	side Coupled
Frequency	350 MHz	700 MHz	700MHz
Energy(MeV)	7.0-20	20-100	100-1000
Current(mA)	100	200	200
Gradient(MV/m)	0.83-2.24	1.7, 1.5	1.5, 1.3, 1.38
Phase(deg)	-35 - -25	-40 - -30	-30
Length(m)	8.0	93.6	1039
Cell/Tank	22-15	7	14
Aperture(cm)	1.0	2 - 2.25	2.5
Focusing	FOFODODO	FDO	FDO
Phase Adv(deg)	70 - 80	80	70

4. CONCLUSIONS

The immediate goal is to provide a code capable of realistic trade off studies. To completely provide this, we still need to add some transport models. Additionally, we need to totally integrate the models for systems engineering, reliability and maintainability; improve the internal engineering, physics and costing models and complete the interfaces with the external existing facility optimization model and the new target / blanket model. New component models, such as for the HILBILAC accelerator, will be added in the future.

Although not complete, the code does provide a useful and powerful tool for the design and evaluation of future beamlines. Physics and engineering models for most components of interest have been implemented and bench-marked against the existing concepts and the code is presently being used to provide trade studies for the IFMIF project. Continued collaboration with GHGA and Los Alamos based on this work is planned

6. REFERENCES

[1] G. H. Gillespie et al.;"A New Approach to Modeling Linear Accelerator Systems"; to appear in the Proceedings of the 1994 International Conference on Accelerator Driven Transmutation Technology and Applications, Las Vegas, Nevada; July 1994

[2] M. A. Peacock et al.;"Overview and Performance of an NPB Accelerator Modeling Code"; Fourth Neutral Particle Beam Technical Symposium; ANL, Chicago, Ill.