GRAPHICAL AIDS TO AEROSPACE VEHICLE MISSION ANALYSIS

B. W. Boehm and J. E. Rieber

October 1967

P-3660

GRAPHICAL AIDS TO AEROSPACE VEHICLE MISSION ANALYSIS

B. W. Boehm and J. E. Rieber*

The RAND Corporation, Santa Monica, California

INTRODUCTION

This Paper will describe an interactive computer graphics system, called Graphic ROCKET, currently being developed as a performance analysis aid for the aerospace vehicle systems analysis process (see Fig. 1). It is intended to provide a means for rapidly specifying and evaluating the performance of a wide range of aerospace vehicle designs and flight plans. This capability will enable the aerospace mission analyst either to get his job done much faster or to consider a larger number of alternatives for accomplishing his mission objective.

DESIGN CRITERIA

The first major design criterion is to provide an easy means of specifying a wide range of alternative

---

*Any views expressed in this Paper are those of the authors. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.

This Paper was prepared for presentation at the American Institute of Aeronautics and Astronautics 4th Annual Meeting and Technical Display, Anaheim, California, 23-27 October 1967.

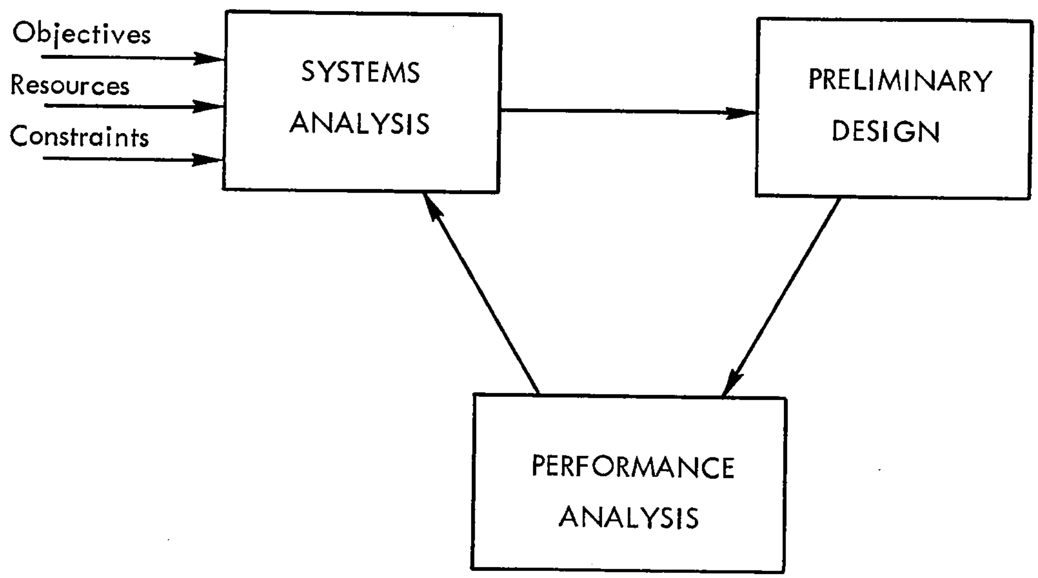

Fig. 1—Systems analysis and preliminary design

designs and flight plans. We don't want to give the mission analyst a case of "technological tunnel vision" by giving him a system which can evaluate only a small range of design alternatives. Implicit in the above is the ability to extend the model easily into areas of the user's choice, and the ability to make comparisons and sensitivity studies.

The second criterion is to make the system communicate in the language of the user. Implicit in this statement is the requirement to make the system as foolproof as possible, providing thorough diagnostics of the user's input and error responses in user language rather than machine language. Also implicit is the ability to extend the model in user language, and to define macros: standard sequences of operations which the analyst can name and refer to at will.

The third major criterion is program efficiency. In an interactive system, the user doesn't like to wait a long time for answers. If five seconds go by with nothing happening, he gets uneasy. Providing efficiency leads to compromises in system design, as one must specialize to increase efficiency, at the cost of some of the system's flexibility and generality.

The specializations we have incorporated show the limitations of Graphic ROCKET. It is centered around trajectory analysis, and doesn't consider detailed vehicle performance such as bending, sloshing, flutter, hydraulics, or pneumatics. Its optimization facilities are primitive. Also, it doesn't consider other performance factors vital to systems analysis such as basing, operations and maintenance, human factors, and costing. Even with these limitations, however, it provides a useful means of discovering which aerospace vehicle designs are capable of performing a mission and deserving of further analysis.

ARCHITECTURE

The hardware available to the system includes an IBM 360/40 with a 5-microsecond cycle time and 32,768 32-bit words of core storage, with three IBM 2311 disk units for peripheral storage. The interactive graphics terminal includes an IBM 2250 cathode-ray tube display with light pen, and a RAND Tablet [1] or "Grafacon" free-hand input device. A picture of the terminal is given as Fig. 2. Hardcopy is provided by a SC-4060 graphic output device.

Program organization is based on the ROCKET program [2], a general-purpose trajectory-analysis program developed in 1961 and maintained and extended since in a batch-processing mode. We modified much of the organization of ROCKET to make it operate more naturally and efficiently in an interactive environment. For example, ROCKET allowed the user a pass at the FORTRAN compiler once during each run. This provided a lot of flexibility but would take too much time for an on-line user, so Graphic ROCKET provides instead lists of options specifiable via the light pen.

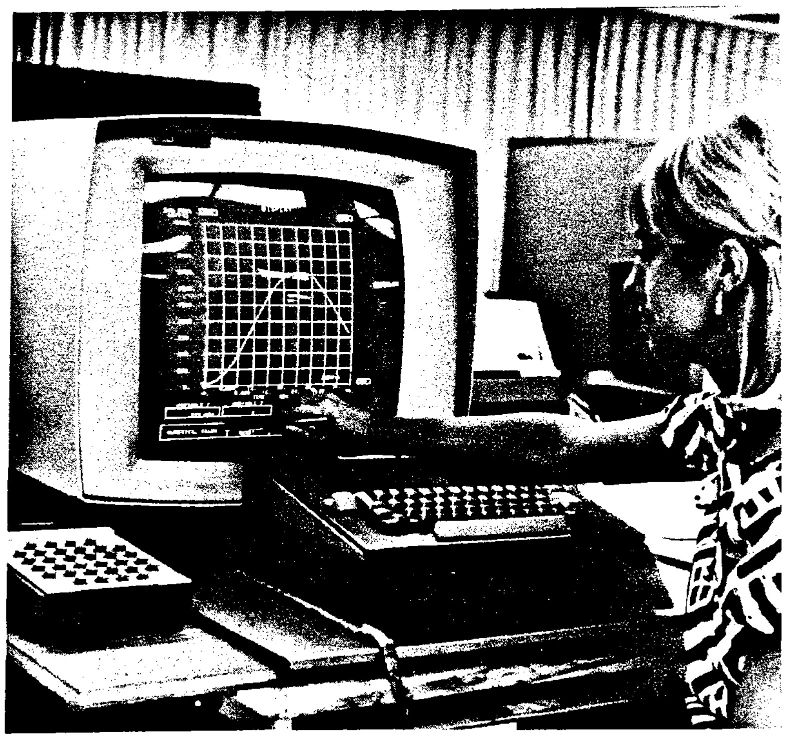

Fig. 2--User Viewing Graphic ROCKET Display

ROCKET also provided for branching, allowing a user to specify up to 36 trajectory variations in his initial run statement. Graphic ROCKET doesn't provide this feature, because a user operating interactively can afford to wait until he has seen the results of his base-case run before he specifies his variations, which he can then do more intelligently.

The language consists of a series of "pages" shown on the cathode-ray tube display. Figure 3 shows the page for specifying initial conditions of the vehicle. One of several options for specifying initial position and velocity is chosen on the upper half of the scope; once an option is chosen, the appropriate input parameter descriptions appear on the lower half of the scope for the user to specify his numerical values. The "page forward" and "page back" boxes allow him to easily skip back and forth as he defines the problem he wants to solve.

Figure 4 shows the page for specifying the performance of the vehicle's propulsion system. It is similar to the above page in allowing several options and in asking the user for values of the parameters appropriate to his option. It is different in that the input represents the propulsion system performance in only one out of up to ten different sections of the trajectory. Thus, the user can handle multistage rockets and multiphase missions in a natural manner.

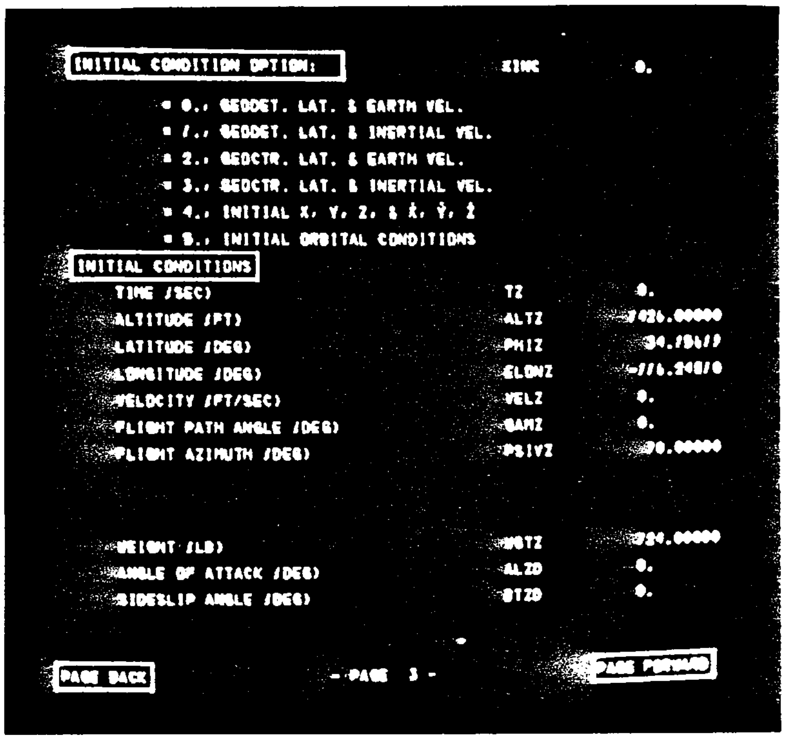

Fig. 3--Graphic ROCKET Display, Showing Initial Position and Velocity

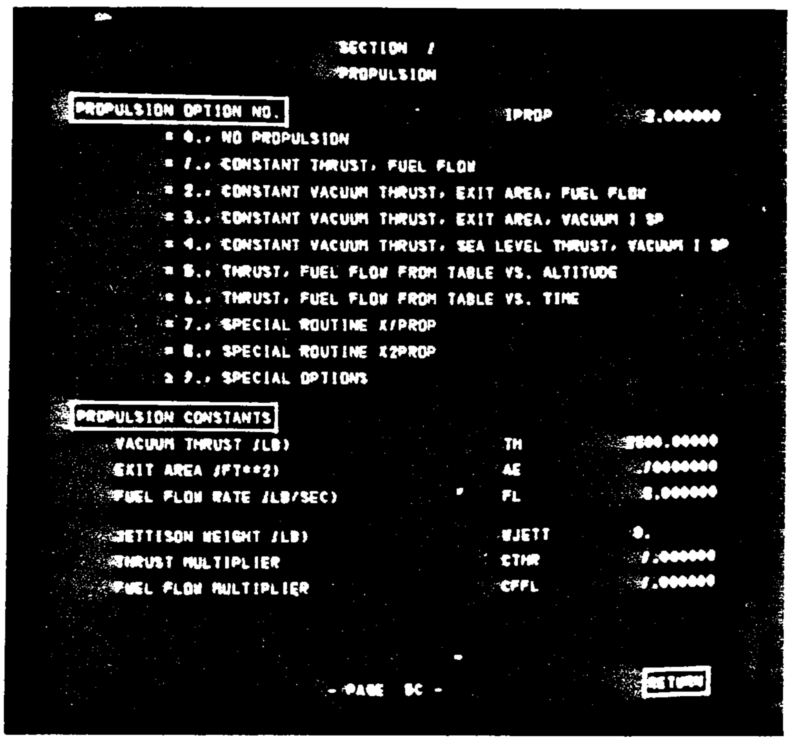

Fig. 4--Graphic ROCKET Display, Showing Section 1 Propulsion System

Figure 5 gives a sample of the graphical output of the system, a curve showing dynamic pressure as a function of time during a rocket vehicle's trajectory. Two display control pages allow the user to specify variables for graphic display and to change displays during the course of the run.

Currently, we are working on a file management scheme which would allow the analyst to store and recall inputs and outputs from his previous runs, or store and recall reference data such as tabular representations of atmospheric, propulsion, and aerodynamic properties.

Further extensions will include the ability to compute and use performance curves from physical designs specified by the user via the Tablet: determining aerodynamic, center of gravity, and moment of inertia characteristics from a vehicle design; and determining propulsion performance from a solid propellant grain design. Another extension will provide some curve-manipulating operations for "post-flight" analysis.

EXAMPLE OF USE

The oral presentation of the Paper will contain an example of the use of the Graphic ROCKET system in evaluating the orbital payload performance of an airlifted SCRAMjet-rocket vehicle for a possible orbital rescue mission. The parameter values used in the example are fictitious, but within a reasonable order of magnitude.

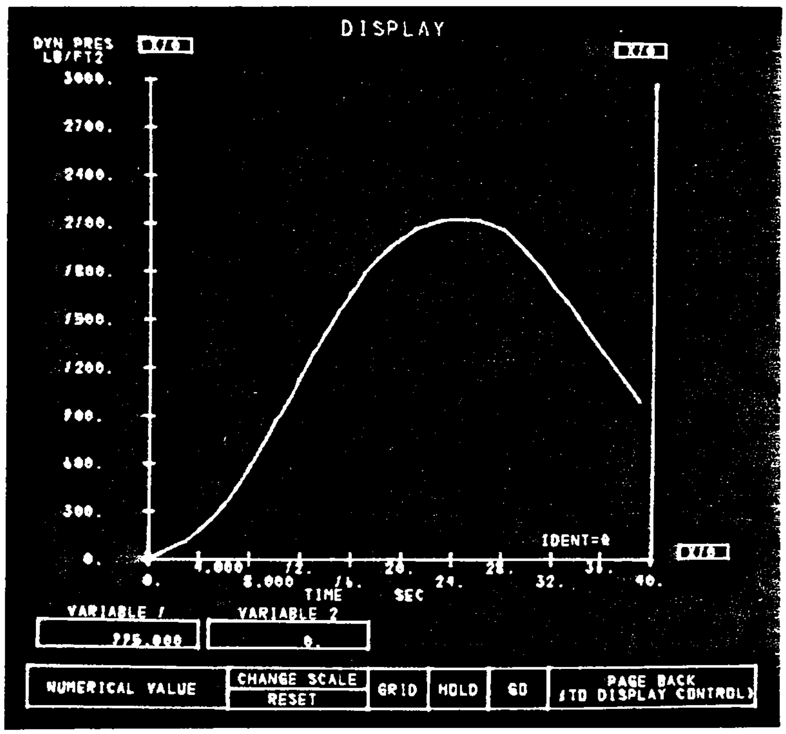

Fig. 5--Graphic ROCKET Output, Dynamic Pressure vs. Time

Emphasis is on the methods of using the system rather than on the utility of the specific results.

CONCLUSIONS

1. The major benefit of interactive operation is the reduction of calendar time required to analyze a mission or the increased number of alternatives which can be investigated in a given time.

2. Even with its higher overhead rate, interactive operation can provide more efficient machine usage, since human judgment can reduce the number of runs required to establish a result.

3. Even a good batch-mode program needs considerable redesign to reorient it toward interactive processing.

4. User enthusiasm for interactive operation depends mainly on two factors: the degree of user-orientation of the language, and the degree of the user's involvement with his problem.

5. Man-machine interfacing for problems involving design creativity is still a little-understood subject. In designing such systems, one shouldn't try to build a closely-optimized system around anticipated usage patterns. Instead, one should build a flexible system, wait and see how the analyst uses it, then modify it to serve him better.

REFERENCES

1. Davis, M. R., and T. O. Ellis, The RAND Tablet: A Man-Machine Graphical Communication Device, The RAND Corporation, RM-4122-ARPA, August 1964.

2. Boehm, B. W., ROCKET: RAND's Omnibus Calculator of Kinematics of Earth Trajectories, Prentice-Hall, 1964.