GLOBAL TRAJECTORY TARGETING VIA

COMPUTER GRAPHICS

F. I. Mann

Report No. 71-39
Contract NAS5-11364
No ember 1971

ANALYTICAL MECHANICS ASSOCIATES, INC.
10210 GREENBELT ROAD
SEABROOK, MARYLAND 20801

ABSTRACT

A technique is described in which the two-point boundary-value-problem (TPBVP) may be solved with the aid of interactive computer graphics. The particular TPBVP considered is the optimal electric propulsion space trajectory problem. An appropriate two-dimensional projection of the TPBVP mapping, or trajectory, is displayed on the computer's television screen, and a man-in-the-loop varies selected trajectory starting conditions in the fashion of a nonlinear walk until the viewed trajectory endpoint lies near a displayed target. Once global targeting is accomplished in this manner, program internal logic can easily handle local targeting to strongly solve the TPBVP.

TABLE OF CONTENTS

Introduction

This work was inspired by a desire to find an easy way of obtaining starting solutions for optimal space trajectories, a problem which has plagued space trajectory analysts for over a decade. The method of attack adopted here is to use an accurate trajectory-generation algorithm in conjunction with the IBM 360 2250 graphic display system. In this approach, the starting conditions of the two-point-boundary-value problem (TPBVP) are guessed, and in general the TPBVP mapping, which is a space trajectory, will miss some specified target, such as the planet Jupiter, by a considerable distance, for instance several astronomical units. Such a trajectory constitutes a bad guess for the solution of the TPBVP, and consequently most numerical TPBVP solution schemes will fail. If, however, a space trajectory can be made to pass near its desired target in space (without regard to other TPBVP end conditions), a sufficiently good TPBVP solution scheme, or iterator, will have a much better chance to converge on a complete solution to the optimal electric propulsion trajectory problem, which has many end-conditions to be satisfied in addition to simple space-targeting.

The accurate trajectory-generation algorithm used is the HILTOP computer program [1]. This program generates trajectories from input starting conditions using numerical integration of the equations of motion for optimal rocket flight. The trajectory endpoint is reached when the specified mission duration for that trajectory is attained, and the endpoint will in general miss the target. The approach adopted here is to guide the trajectory endpoint toward the target by observing the effect which varying a given trajectory starting condition has on the trajectory endpoint on a computer-generated television display. After this crude global targeting is accomplished, program control may be entirely passed back to the HILTOP program for the purpose of converging strongly and accurately on the desired solution. This constitutes one basic method of facilitating TPBVP solutions.

It is interesting to note in passing another entirely different, yet basic, approach to solving a TPBVP, namely, forcing every trajectory to hit the target while sacrificing satisfaction of the required equations of motion. In this instance, each trajectory violates physical reality until the converged solution is obtained. This is the general approach of the Generalized Newton Raphson methods, such as the one described in [2], and also of the Series Expansion Approximation methods, such as the Chebychev polynomial approach described in [3], in which even the converged solution is approximate. The former methods tend to encounter severe convergence difficulties as the dimensionality of the problem increases, and the latter methods sacrifice some accuracy. Yet these methods have the advantage of not requiring human intervention during execution on the computer.

The computer graphics method of guiding the TPBVP toward a solution has the disadvantage of requiring a man-in-the-loop, thereby consuming valuable manhours. On the positive side, besides helping to solve the TPBVP, the computer graphics method affords the experimenter or trajectory analyst a greatly-enhanced amount of "visibility" concerning his particular problem, thereby sharpening his intuition.

Global Trajectory Searching Technique

This section sketches some of the mechanical details concerning the HILTOP graphic display and the performance requirements of the man-in-the-loop. Typically, the computer run is submitted to the computer in the batch mode, and the experimenter waits at a graphics terminal until it is de-queued and commences execution. A basic case (e.g., a Jupiter flyby mission) is submitted as part of the run. At execution commencement, some introductory words appear on the television screen, the entire programmed function keyboard (PFK) lights up, and the execution pauses. The experimenter then releases the program from the pause mode by punching any PFK button, and some trajectory starting conditions are displayed (which he may jot down). All program control is currently accomplished via the PFK, and only applicable PFK buttons are lighted at any given time. The currently-lighted buttons together with the PFK overlay (shown in the Appendix) make the HILTOP graphic system relatively easy to use.

The experimenter then allows the program to continue execution. The ecliptic-projection of an interplanetary trajectory begins to appear on the screen, and the experimenter watches the progress of the numerical integration. A symbol for the sun appears at the screen's center, which corresponds to the center of the solar system, and an X appears at the target's position. The experimenter notes the trajectory endpoint and target miss distance. He may interrupt program execution at any time, transfer control back to the starting point, change any subset of the program's input quantities by means of the PFK, and then begin again to allow the program to integrate a single trajectory or several trajectories. The global targeting is accomplished by running a sequential set of single trajectories while altering some selected trajectory starting parameter between each trajectory. The experimenter may also allow the program's iterator to operate without interruption. The iterator generates major (or nominal) trajectories, minor (or neighboring) trajectories for constructing a partial derivative matrix for the TPBVP, and also sub-major trajectories, all of which are described in [1]. The experimenter views these trajectories on the screen, and may watch the iterator attempt to move the TPBVP toward a solution. He may also see a telescopic view of the nominal trajectory endpoint and the relation of all the neighboring trajectory endpoints to the nominal endpoint in space. This allows him to detect possible erratic neighboring trajectories and subsequently to decrease the size of the corresponding independent variable perturbation step size via the PFK.

An inherent limitation of global targeting of a trajectory using computer graphics is that the television screen constitutes a two-dimensional surface. This means that only two dependent variables (such as the x-coordinate and y-coordinate target miss components) out of the total N dependent variables of the TPBVP may be conveniently satisfied by performing a nonlinear walk in selected trajectory starting conditions, and the remaining N-2 dependent variables are ignored. While it may be theoretically possible to monitor all N dependent variables somehow on the television screen, it would be virtually impossible for a man-in-the-loop to perform a meaningful nonlinear walk in a given independent variable such as to force N dependent variables toward a solution. Fortunately in the optimal rocket trajectory problem, the x and y coordinate target miss distances are of relatively great importance compared to most of the other dependent variables.

Software Development

A special graphics-version of the HILTOP computer program has been created and is referred to as HILTOP/GRAPHIC. A considerable amount of programming work was done to the HILTOP program in order to produce a graphics-version with reasonably comprehensive execution supervision features. This constituted most of the effort of this work. In addition to modifying many HILTOP subroutines, eighteen (18) new subroutines were created, including such major subroutines as a general-purpose PFK input editor routine, some floating-point and integer to BCD conversion routines, and a trajectory-plotter routine.

Example Case

The following specific HILTOP/GRAPHIC run is given as an example of how global trajectory targeting can aid the TPBVP solution.

A typical Mode A Jupiter flyby solar electric propulsion mission using a Titan III B (core)/Centaur launch vehicle was simulated. The trajectory starting conditions (TPBVP independent variables) are displayed in Table I along with the initially-guessed values and the finally converged values. (Symbol definitions and units are given in [1].) The initial guesses resulted in a target miss distance of 2.6 astronomical units (AU). (Jupiter is 5.2 AU from the sun.) Without computer graphics, the HILTOP program generated 669 trajectories (total) before arriving at a converged solution. Indeed, it is easy to conceive of instances where the HILTOP program would never converge without graphics. With computer graphics, the man-in-the-loop first tried $\lambda_1 = 3.00$, which resulted in a trajectory closer to Jupiter but missing on the other side compared to the original trajectory. The man-in-the-loop therefore backed off to $\lambda_1 = 2.80$ and obtained a trajectory satisfyingly close to Jupiter, having a miss distance of 0.14 AU. Control was turned back to HILTOP's iterator, which produced convergence in 166 trajectories (total), representing a 75% machine time savings relative to the original guess. Thus a simple nonlinear walk in just one of the trajectory starting conditions $\lambda_1 = (2.0,\ 3.0,\ 2.8)$ effected rapid global targeting of the trajectory.

Concluding Remarks

This work constitutes a preliminary assessment of global trajectory targeting via computer graphics. This concept appears very promising as a valuable tool for finding new optimal space trajectory solutions, and could conceivably be employed to aid in the solution of other two-point boundary-value problems.

TABLE I

INDEPENDENT VARIABLES FOR 1000 DAY JUPITER MISSION

Variable	Initial Guess	Converged Value
λ_1	2.00	2.415
λ_2	-2.00	-2.624
λ_3	0.01	0.021
λ_4	4.00	3.128
λ_5	0.00	0.157
λ_6	-0.01	-0.008
a_o	5×10^{-4}	4.8×10^{-4}
c	3×10^{4}	2.7×10^{4}
$V_{\infty D}$	2000	2183
t_1	-50	-53
t_2	950	947
θ_{thrust}	75°	$75^{\circ}.7$

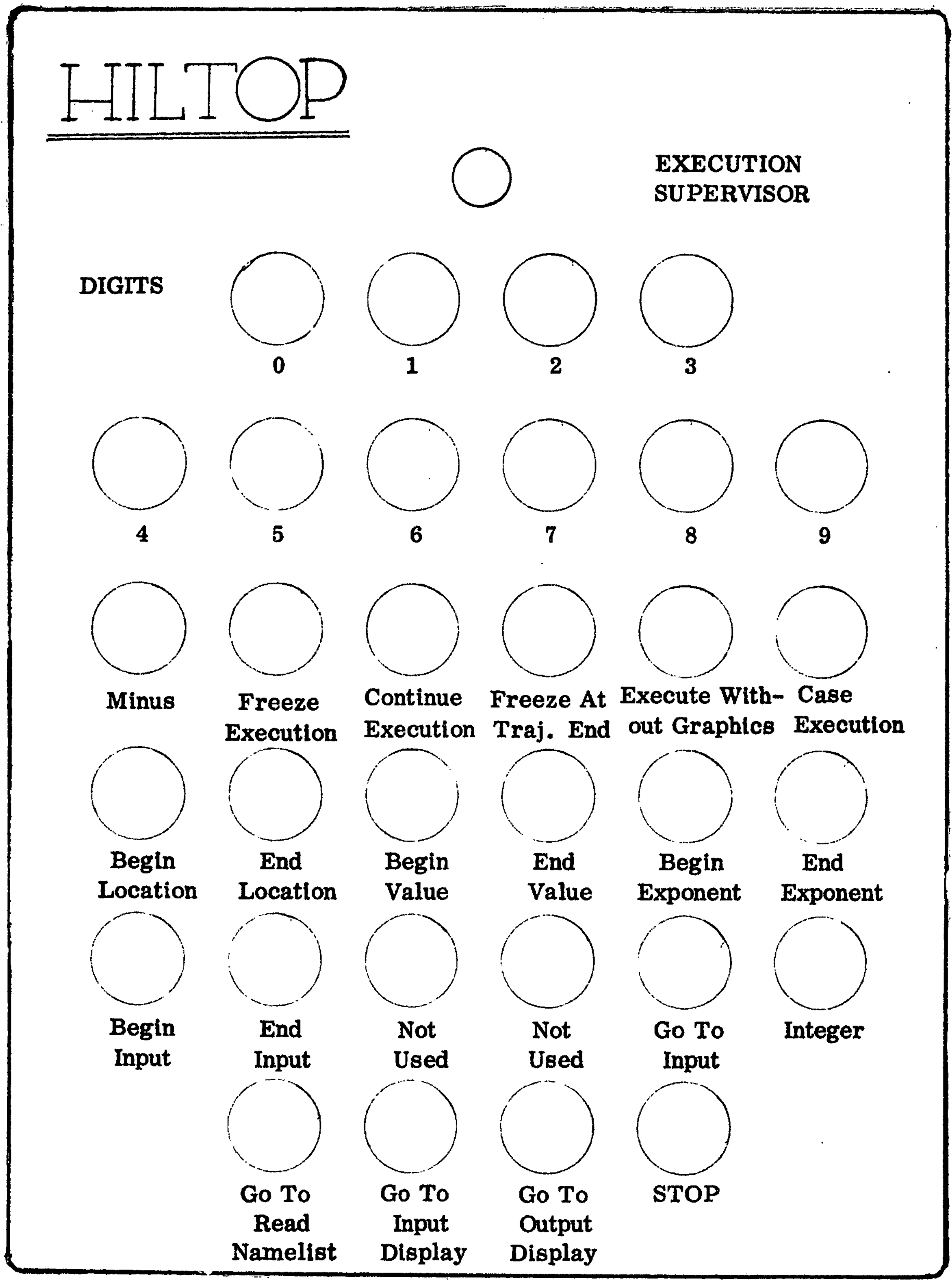
REFERENCES

[1] J. L. Horsewood, P. F. Flanagan, and F. I. Mann, "HILTOP: Heliocentric Interplanetary Low Thrust Trajectory Optimization Program," Analytical Mechanics Associates, Inc., Report No. 71-38, November, 1971.

[2] C. P. VanDine, W. R. Fimple, and T. N. Edelbaum, "Application of a Finite-Difference Newton-Raphson Algorithm to Problems of Low-Thrust Trajectory Optimization," AIAA Paper No. 65-698, Astrodynamics Specialist Conference, Monterey, California, September 16-17, 1965.

[3] D. W. Hahn and F. T. Johnson, "Chebychev Trajectory Optimization Program (CHEBYTOP II)," The BOEING Company Report D180-12916-1, June, 1971.

APPENDIX



HILTOP/GRAPHIC PFK OVERLAY