currently available. Automatic verification of a program requires that the effects of every language construct should be symbolically representable in an exact way. Only in that way can meaningful assertions be constructed to describe the program state. Operations on scalar data items introduce error terms, the magnitude of which is highly data-dependent. The matrix and vector data types, being composed of scalar elements, are also omitted from HAL/S/V.

2) No nonlocal referencing is permitted in HAL/S/V. Nonlocal referencing inhibits program decomposition for proof purposes and complicates program analysis substantially. For parameterized program units, it also permits aliasing since global data may be referenced directly as well as through the parameter list.

3) Replace statements and replace macros, the textual substitution facility of HAL/S, have been omitted from the subset. Each instance of textual modification presents the verifier with two programs to verify—the original and the modified version. In the worst case, the difficulty of verification may be exponential in the number of replace statements appearing. Such a severe penalty seems to outweigh the benefits of the facility.

4) No implicit precision or type conversions are permitted in HAL/S/V. Implicit conversion masks from the programmer and from the verifier the fact that in the conversion process information may be lost or spurious information added. Requiring that the programmer perform all precision conversions explicitly forces him to recognize this possibility and compensate for it.

5) HAL/S/V functions are restricted to conform to the mathematical notion of "function" as much as possible. That is, the entire behavior of the unit should be representable by a functional (in the mathematical sense) input-output relation. The prohibition of nonlocal referencing is an important step in that direction. Additionally, functions may not call, directly or indirectly, time-dependent system routines such as RUNTIME, DATE, or PRIORITY.

6) Input parameters to a routine may not be altered during the execution of the routine. To ensure this, input (value) and assign (reference) parameter lists must be entirely disjoint; input parameters may not be passed as assign parameters to other routines.

7) The HAL/S name data type is not included in HAL/S/V. A restricted pointer facility is a valuable feature of many languages. Verifiable pointer facilities in languages such as Pascal allow pointers only to unnamed, dynamically created objects. However, treating pointers as merely alternate names for data items, as HAL/S does, invites aliasing of the worst kind.

8) Transient event data items are not permitted in the verifiable subset. The semantics of logical operations such as "not E," where E is a transient event, is not well-defined.

9) To reduce the complexity of proofs of concurrent programs, the following restrictions are made. Accesses to shared data should be carried out in mutual exclusion; therefore, all shared data items belong to lock groups. No process may terminate abnormally if it or any dependent accesses shared data. All scheduling should occur at the outermost program level.

Though this list contains the major features of HAL/S omitted from the verifiable subset, there were a large number of more minor deletions, e.g., array processing features and the subbit conversion facility. Some modifications were necessary to make HAL/S/V a consistent language, e.g., requiring that the arguments of schedule and wait statements, which are scalar valued expressions in HAL/S, be integer valued in HAL/S/V.

Conclusions

The attempt to isolate criteria of evaluation for verifiability and apply them to HAL/S has revealed the following:

1) The design of programming languages for aerospace applications must be a process of compromise between the goals of inclusiveness and expressiveness on the one hand and verifiability on the other.

2) A clearer understanding of the kinds of features which are verifiable will aid in future work in designing automatic verification systems for HAL/S and other languages. Such efforts will doubtlessly be applied to many languages as more effective program proving techniques become available.

3) Most users of a language are concerned with writing software which is correct, though not necessarily formally verifiable. For them, verifiability analysis identifies language features which are ill-suited to structured programming and which contribute to the writing of software which is difficult to read, modify, and understand.

Acknowledgments

The authors are grateful to Dr. J. Misra for discussions regarding HAL/S and language evaluation. This work was supported by NASA Grant NSG-1479.

References

[1] Good, D.I., Cohen, R.M., Hoch, C.G., Hunter, L.W., and Hare, D.F., "Report on the Language Gypsy, Version 2.0," The University of Texas at Austin, ICSCA-CMP-10, Sept. 1978.

[2] London, R.L., Guttag, J.V., Horning, J.J., Lampson, B.W., Mitchell, J.G., and Popek, G.J., "Proof Rules for the Programming Language Euclid," *Acta Informatica,* Vol. 10, Jan. 1978, pp. 1-26.

[3] Young, W.D., Tripathi, A.R., Good, D.I., and Browne, J.C., "Evaluation of Verifiability in HAL/S," *Proceedings of the AIAA Computers in Aerospace Conference II,* Oct. 1979, pp. 359-366.

[4] "HAL/S Programmer's Guide," Intermetrics, Inc., Cambridge, Mass., June 1976.

[5] "HAL/S Language Specification," Intermetrics, Inc., Cambridge, Mass., June 1976.

[6] Cheheyl, M.H., Huff, G.A., Gasser, M., and Millen, J.K., "Secure System Specification and Verification: Survey of Methodologies," MITRE Corp., Bedford, Mass., WP-22473, Sept. 1979.

[7] Wirth, N., "On the Design of Programming Languages," *Proceedings of IFIP Congress 1974,* North-Holland, Amsterdam, 1974, pp. 386-393.

[8] Hoare, C.A.R., "Hints on Programming Language Design," *Proceedings of the Symposium on Principles of Programming Languages,* SIGPLAN/SIGACT, 1973, pp. 1-30.

AIAA 80-0058R

Dumping Momentum Magnetically on GPS Satellites

Jack R. Ferguson Jr.*
University of Texas, Austin, Texas
and
George T. Kroncke†
Martin Marietta Corporation, Denver, Colo.

Introduction

GLOBAL Positioning System (GPS) is a planned worldwide navigation system which will provide users with a three-dimensional location accurate to ± 16 m. The space segment of the system will consist of 18 three-axis stable, Earth-pointing satellites in inclined circular orbits of 12-h periods ($r = 4.17\ R_e$). To date, six satellites have been launched and are being used in the concept validation phase of the program.

Presented as Paper 80-0058 at the AIAA 18th Aerospace Sciences Meeting, Pasadena, Calif., Jan. 14-16, 1980; submitted Jan. 17, 1980; revision received July 28, 1980. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Graduate Student, Aerospace Engineering. Member AIAA.
†Staff Engineer, Systems Engineering Branch, Aerospace Division.

The system requires extremely precise satellite ephemerides to achieve the design navigation accuracies. A major source of ephemeris error is the expulsion of reaction control system (RCS) propellant necessary to reduce (dump) unwanted momentum from the onboard attitude control system reaction wheels. External torques caused by the satellite's magnetic field and by solar pressure are countered by changes in the angular momentum of each wheel. These momenta accumulate and must be discarded before wheel speed becomes excessive. Using the RCS thrusters to dump excess angular momentum produces a net force on the satellite due to thruster misalignment and imbalances and, about the spacecraft roll (x) axis, due to plume impingement on the spacecraft. This net force results in velocity errors and hence, ephemeris errors.

Each satellite contains two electromagnets, one each on the spacecraft roll (x) and pitch (y) axes. A previous study demonstrated the feasibility of dumping satellite angular momentum magnetically using the interaction between the Earth's magnetic field and the onboard electromagnets.[1] This technique was designed to eliminate the thruster impulses and resulting ephemeris errors. A second study proposed an algorithm to compute magnet values at four switch times over one orbit that would dump approximately 0.08 N-m-s of momentum.[2] This study also discussed reasons why GPS poses several new problems in the area of dumping momentum magnetically, the major problems being the lack of on-board magnetometers and the absence of measured magnetic field data in the region between 4 and 5 R_e, the regime in which GPS satellites orbit. The two studies recommended the use of the Mead-Fairfield magnetic field model, which is based on magnetometer data taken primarily beyond 5 R_e (Ref. 3).

The following sections describe the modifications required to make the original algorithm function with the restrictions imposed by actual satellite operations and show the results obtained in testing the method on the GPS satellites in orbit.

Operational Constraints

The original problem was to determine the minimum magnet values which would effect a dump of all unwanted angular momentum using four magnet switch times over one orbit. It was solved using a minimum normalization solution to an underspecified matrix equation. When applied to on-orbit satellites and satellite control techniques not designed for magnetic control, however, extensive modifications had to be made to the algorithm to make it effective in the real-world situation.

Switching Time

The original method assumed that four magnet switches over one orbit could be accomplished each day with each satellite. The method would not eliminate the momentum buildup during the dumping interval, thus momentum built up over the two previous orbits would be dumped every other orbit. With 18 satellites, this would require 72 commands to be transmitted each day, but scheduling requirements dictate approximately 30 min at each tracking station to prepare and send each magnet command. The 36 hours of support required each day would be an impossible task. To operate in the context of satellite control constraints, the problem was modified so that a maximum momentum dump was accomplished over approximately two orbits using nine switch times. This was then repeated whenever the momentum on any spacecraft axis approached the limiting value of 0.34 N-m-s, resulting in dumps being required after several days instead of every day.

To achieve the maximum dump, several changes were made to the algorithm. The first change involved optimizing switch times. In the original technique, switch times were arbitrarily chosen at the ascending node, north point, descending node,

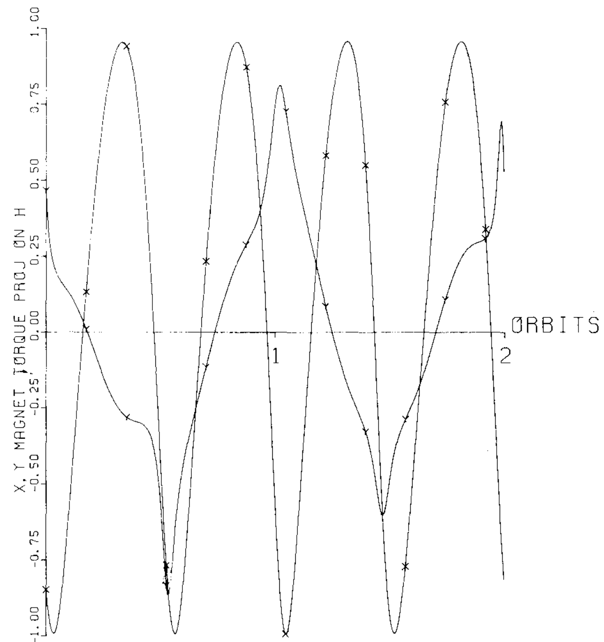

Fig. 1 Normalized x and y magnet torque projection on the angular momentum vector.

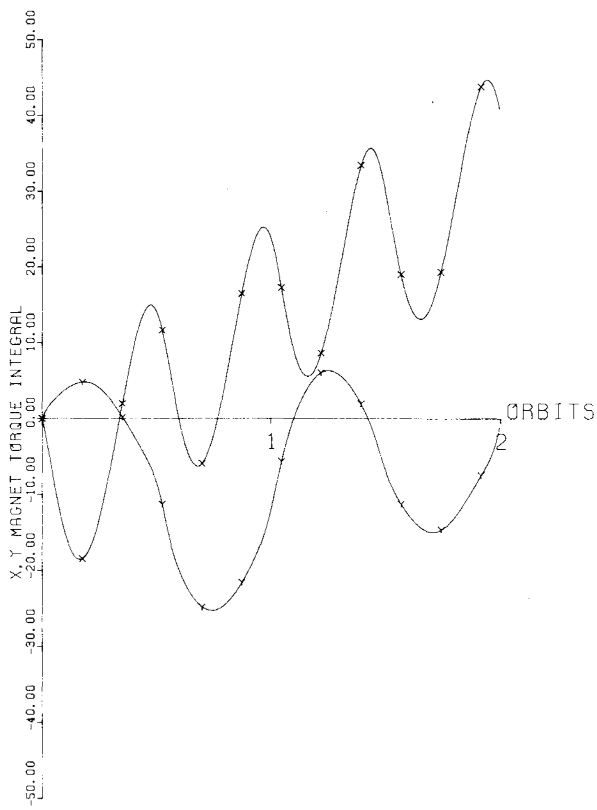

Fig. 2 Integral of normalized x and y magnet torque projection on momentum vector.

and following ascending node. It was found that up to 60% more momentum could be dumped if the times were chosen to optimize the magnetic torque-momentum vector interaction.

Figure 1 is a typical plot of the projection of the normalized x and y magnet torque on the angular momentum vector. To optimize the effect of these torques, new magnet values should be computed for each zero crossing. Looking at the effect of both magnets on the momentum vector, it is seen that more than 12 switches would be required over a two-orbit interval. Discussion with the mission control team (MCT) led to the decision to use nine times (eight switches and one turnoff) as a compromise. The procedure which determined the optimum switch times used the torque-momentum vector projection and the integral of the projection (Fig. 2). This integral corresponds to the change in momentum caused by a magnet value of 1 pole-cm. The choice between two switch times was then made by choosing the magnet that caused the largest change in momentum from one time to the next.

Optimum Dump

Since dumps were to be scheduled as far apart as possible, the probability of reducing the momentum vector to zero was low, and the goal of the procedure was changed from finding the minimum magnet values which dump all the momentum to finding the magnet values which minimize the remaining momentum. This is a quadratic programming problem whose objective is:

$$\min\|H-AM\| = \min(H-AM)^T(H-AM)$$
$$= \min(H^TH - H^TAM - M^TA^TH + M^TA^TAM) \qquad (1)$$

subject to the constraints

$$|m_i| \leq 7500 \text{ pole-cm}$$

where H is the 3×1 momentum vector, M is a 16×1 magnet solution vector, and A is a 3×16 array of influence coefficients relating change in momentum to magnet setting. Since H^TH is simply a constant and H^TAM and M^TA^TH are scalars, the objective function reduces to

$$\min(-H^TAM + \tfrac{1}{2}M^TA^TAM) \qquad (2)$$

and the constraints may be expressed as

$$m_i + 7500 \leq 15{,}000 \qquad m_i + 7500 \geq 0 \qquad (3)$$

The problem is now in a form that can be handled by the quadratic programming technique derived by Lemke[4] to compute magnet values that minimize the objective function.‡ The use of this technique, along with optimum switch times, produced theoretical dumps of up to 0.35 N-m-s depending on satellite-sun-momentum geometry. The actual dump also depends on the actual magnet switch times scheduled since nonvisibility and conflict with other satellites desiring support from the tracking stations may preclude some optimum switches.

Magnet Bias

In addition to optimizing switch times and momentum dumped, the algorithm was required to compute magnet values for biased magnets. In an effort to reduce the momentum growth rate, the mission control team calculated and implemented constant magnet values. These values represent biases to the nominal zero points and result in changes to the limiting magnet strengths in both the plus and minus direction. A negative bias value reduces the negative limit available to the algorithm and increases the positive limit, while a positive bias does the converse. To handle this situation, the quadratic programming problem constraints were changed to

$$|m_i + b_i| \leq 7500 \text{ pole-cm} \qquad (4)$$

and the solution reformulated to take the biases into account. The result was a drop in dump capability in some cases and an increase in capability in others, depending on the signs of the magnet settings and the biases.

‡This technique was brought to our attention by Joseph C.H. Smith, Associate Professor of Mathematics, USAF Academy, Colo.

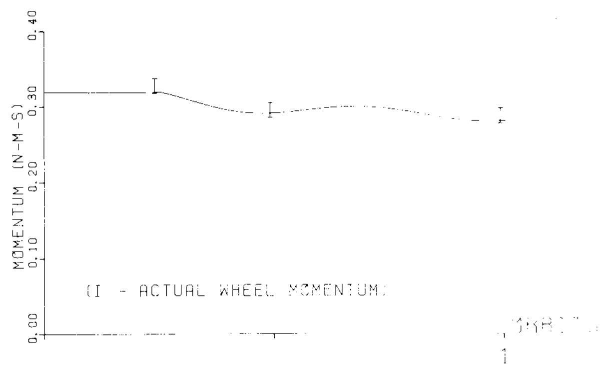

Fig. 3 Actual and predicted spacecraft total momentum—July 12, 1979 test.

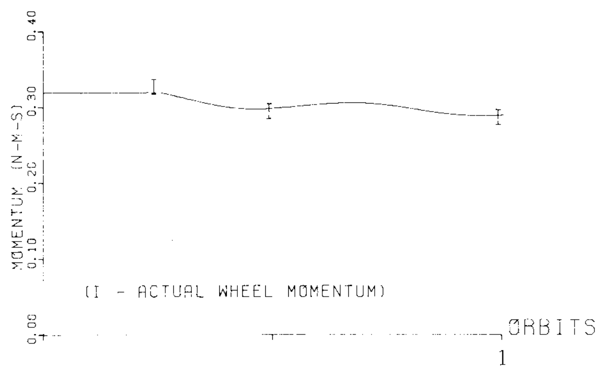

Fig. 4 Actual and predicted spacecraft total momentum—July 12, 1979 test, *y* magnet reversed.

Results

The algorithm was initially tested on July 12, 1979 with the GPS satellite NAVSTAR 1. Since we were concerned about the possibility of a sign error, a conservative one-orbit test was conducted in which a dump of only 0.042 N-m-s was attempted.

Figure 3 is a plot of the predicted and actual momentum magnitude seen during the test. The vertical bars (I) represent actual data with the height representing the width of the least significant bit in the telemetry bit stream. Sensor error would be in addition to the telemetry error. The major result obtained at the time was that the telemetry system noise level was close to the desired momentum dump value.

Shortly after the test was concluded, we were notified that one of the electromagnets on the satellite was reversed. A period of testing was initiated and a simulation with the *y* (pitch) magnet reversed (Fig. 4) showed a slightly closer agreement with the telemetry values, but the noise level still clouded the issue, making a decision as to which magnet was reversed difficult. The spacecraft contractor then conducted a test with one of the assembly line vehicles and verified that, according to our sign convention, the *y* magnet was indeed reversed.

The first full-scale test was conducted August 8, 1979 in which a maximum dump was attempted. The momentum vector magnitude at the start of the test was 0.267 N-m-s. The algorithm then computed that 0.158 N-m-s could be dumped during the two-orbit dump interval. Figure 5 shows the predicted and actual momentum magnitude and the predicted momentum vector in a spacecraft-fixed frame. The use of this frame for momentum predictions provides the MCT with values with which real-time analysis of the dump can be made.

It is seen that the predicted and actual total momentum values compare fairly well in this test. Differences between predicted and actual values may be caused by momentum growth between the computer run and the dump start time, momentum growth during the dump period, and noise in the starting momentum value. This test did not achieve a

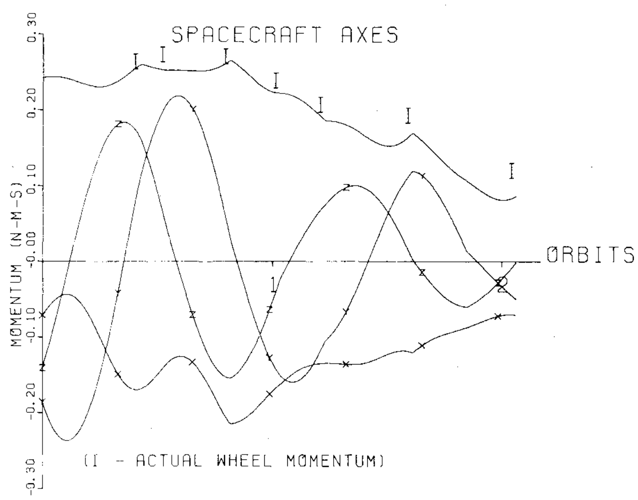

Fig. 5 Actual and predicted momentum vector components and magnitude—Aug. 8, 1979 test.

maximum dump because not all optimum switch times were schedulable. In fact, the predicted total momentum actually increased at four times, with these increases being caused by later than optimum switch times.

Conclusions

The main unknowns before testing were the usability of the Mead-Fairfield model at radii lower than its design regime and how much momentum could actually be dumped by on-orbit satellites using the magnetic control technique. It was found that the magnetic momentum dump algorithm works well with on-orbit GPS satellites using the Mead-Fairfield magnetic field model. As of Nov. 1, 1979, a total of 15 tests had been run with the four satellites in orbit at that time. The method has succeeded in dumping up to 0.33 N-m-s excess angular momentum in two orbits. For the specific satellite/orbit geometry combination tested, this corresponded to a buildup occuring over about 12 days.

References

[1] Fuchs, R.P. and Eller, T.J., "GPS Magnetic Momentum Dumping Feasibility Study," USAF Academy, TR-76-14, 1976.

[2] Kroncke, G.T. and Fuchs, R.P., "An Algorithm for Magnetically Dumping GPS Satellite Angular Momentum," *Journal of Guidance and Control,* Vol. 1, July-Aug. 1978, pp. 269-272.

[3] Mead, G.D. and Fairfield, D.H., "A Quantitative Magnetospheric Model Derived From Spacecraft Magnetometer Data," *Journal of Geophysical Research,* Vol. 80, Feb. 1975, p. 523.

[4] Lemke, C.E., "A Method of Solution for Quadratic Programs," *Management Science,* Vol. 8, 1962, pp. 442-453.

AIAA 81-4004

Special Cubic Solution Function

Dunstan Graham*
Princeton University, Princeton, N.J.

Introduction

CONSIDER the equation with real coefficients

$$ax^3+bx^2+cx+d=0 \qquad (a>0) \tag{1}$$

Received April 15, 1980; revision received Aug. 27, 1980. Copyright © 1980 by D. Graham. Published by the American Institute of Aeronautics and Astronautics with permission.

*Consultant. Associate Fellow AIAA.

Girolamo Cardan[1] published a solution, but the general procedure is so tedious that it is seldom attempted. Instead, usually, recourse is had to iterative calculations. These may be greatly facilitated by either graphs or tables, and there have been several attempts to provide such aids in a convenient form (see, e.g., Refs. 2-6).

Although today there are very powerful methods for the numerical solution of polynomial equations (including cubic equations) and, typically, several such root solving methods are a part of the "libraries" associated with stored-program computers, an exclusive reliance on such methods may not always be advised. It is primarily to the development of a simple, general, and reliable method for the numerical solution of cubic equations on a hand-held calculator that this paper is addressed. Of course, if the cubic equation can be solved, the quartic equation is also amenable to solution by Ferrari's, Lagrange's, or a similar technique.[6]

Combining the methods of the reduced[1,2,6] and normalized equations[3,4] leads to a new and more convenient equivalent equation. A solution of this equation (corresponding to an isolated real root) has overlapping asymptotic properties, and easily learned asymptotic starting estimates can be used in effective procedures for very rapidly refining an approximate solution. The remaining two roots may then be found readily as the solutions to a depressed equation.[7] Charts and tables are superfluous.

The Normalized, Reduced Equation

In Eq. (1), set $x=y-b/3a$ This yields the reduced equation[7]

$$y^3+\left(\frac{c}{a}-\frac{b^2}{3a^2}\right)y+\left(\frac{d}{a}+\frac{2b^3}{27a^3}-\frac{bc}{3a^2}\right)=0 \tag{2}$$

Then set[4]

$$y=s\,|m|^{1/3}$$

where

$$m=\left(\frac{d}{a}+\frac{2b^3}{27a^3}-\frac{bc}{3a^2}\right)$$

This yields the normalized reduced equation

$$s^3+\frac{(c/a-b^2/3a^2)}{|m|^{2/3}}s+(1)\,\mathrm{sgn}(m)=0 \tag{3}$$

For the moment, *assume* $\mathrm{sgn}\,m>0$, so that

$$s^3+ks+1=0 \tag{4}$$

where

$$k=\frac{-3(b^2-3ac)}{|27a^2d+2b^3-9abc|^{2/3}}$$

and (since $a>0$), in fact, the sign of the trailing term in Eq. (4) will be the sign of $(27a^2d+2b^3-9abc)$.

Equation (4) will always have at least one real root s_1. This root, of course, is a function G of the parameter k and is opposite in sign to the quantity m. Thus,

$$s_1=-\mathrm{sgn}(27a^2d+2b^3-9abc)\,G(k) \tag{5}$$

Using the previous definitions, together with the understanding that $(\cdot)^{1/3}$ means the real cube root, the relationship between the root s_1 and a corresponding real root x_1 of Eq. (1) shows that

$$x_1=\frac{-b}{3a}+y_1=\frac{-b}{3a}-\frac{(27a^2d+2b^3-9abc)^{1/3}}{3a}G(k) \tag{6}$$