IBM Report No.
67-M21-11

TECHNICAL PROPOSAL

FLEXIBLE GUIDANCE AND SOFTWARE SYSTEM

(FORMERLY QUICK REACTION GUIDANCE & TARGETING)

SUBMITTED IN RESPONSE TO RFP NO. F04701-68-R0113

DECEMBER 1967

Prepared For

Space Guidance Branch
Space and Missile Systems Organization
Air Force Systems Command
United State Air Force

Each transmittal of this document outside the Department of Defense must have the prior approval of the Deputy for Technology, Space and Missile Systems Organization, Air Force Systems Command, Los Angeles Air Force Station, California.

IBM
FEDERAL SYSTEMS DIVISION
SPACE SYSTEMS CENTER
ENDICOTT NEW YORK

IBM Report No.
67-M21-11

TECHNICAL PROPOSAL

FLEXIBLE GUIDANCE AND SOFTWARE SYSTEM (FGSS) (FORMERLY QUICK REACTION GUIDANCE AND TARGETING)

Submitted in Response to RFP No. F04701-68-R0113

December 1967

Prepared For

Space Guidance Branch
Space and Missile Systems Organization
Air Force Systems Command
United States Air Force

Prepared By

Space Systems Center
Federal Systems Division
International Business Machines Corporation

Notice

This data furnished in response to RFP No. F04701-68-R0113 shall not be disclosed outside the Government or the Aerospace Corporation or be duplicated, used or disclosed in whole or in part for any purpose other than to evaluate the proposal; PROVIDED that if a contract is awarded to this offerer as a result or in connection with the submission of such data, the Government shall have the right to duplicate, use, or disclose this data to the extent provided in the contract. This restriction does not limit the Government's right to use information contained in such data if it is obtained from another source.

TABLE OF CONTENTS

Section 1
INTRODUCTION

This proposal is submitted by the Space Systems Center, Federal Systems Division, IBM, Endicott, New York, in response to RFP No. F04701-68-R0113. IBM will provide the personnel, services, and facilities to perform analyses and provide detailed design specifications for a modularized, highly automated, self-determining, quick-reaction guidance and software system designated Flexible Guidance Software System (FGSS).

IBM proposes to conduct this program for the Space Guidance Branch, SMTAG, (formerly SSTDG) of the Space and Missile Systems Organization, Los Angeles Air Force Station, California. The proposed study is a continuation of work performed under Contract AF 04(695)-1078 entitled Quick Reaction Guidance and Targeting Study. The objective of the new work is to further the system design initiated in Phase I and to further establish the feasibility of the design concepts.

The FGSS program is a part of Air Force Advanced Development Program 681D. The first phase of study was initiated in August 1966 and was completed in June 1967. The Final Technical Report on Phase I was published as Air Force Report No. SSD-TR-67-122.

1.1 BACKGROUND

The overall objective of the FGSS program is to reduce the cost and time to prepare software for space flights.

At the present time, preparation for each space mission involves a sequence of planning, formulation of computer requirements, software program design and development, validation, and pre-launch testing. For a given flight and vehicle configuration, several alternate trajectories are generated, compared, and adjusted until a satisfactory set of flight plans are obtained meeting all criteria and providing for contingencies. The guidance and control equations necessary to implement the mission are then derived (or re-formulated) and simulated. In most cases the equations are closely tailored to the particular mission to conserve limited spacecraft computer capability. The necessary targeting and guidance constants are then generated from trajectory simulations or analytical formulas. Finally, the flight program is written and validated, and the complete system thoroughly simulated. This process, repeated for each flight with very little change, is time consuming and costly. Even minor changes in hardware or flight objectives necessitates repetition of a large part of the procedure. This process is depicted, in simplified form, in Figure 1.

During Phase I, IBM studies mission planning and guidance techniques in depth to determine means for reducing overall cost and time. Software processing techniques and procedures were also studies to establish areas of greatest possible improvement. This work led to the definition of the Flexible Guidance and Software System to reduce overall costs and lead times for space software.

Figure 2 illustrates IBM's concept of FGSS; it embodies the concepts that will reduce software preparation costs and time, namely:

- A multi-mission library of reusable flight program modules.
- Semi-automatic mission planning and targeting capability.
- Explicit, self-optimizing guidance algorithms.
- Modular guidance programs and elements.
- A configurator to form specific vehicle libraries.
- Advanced support software.
- Higher order languages.
- Spacecraft executive control system.

A description of the design and operation of FGSS is given in Section 2 of this proposal. The full development of FGSS will change the software preparation process from that illustrated in Figure 1 to that illustrated in Figure 3. The reaction time for mission operation will be shortened considerably.

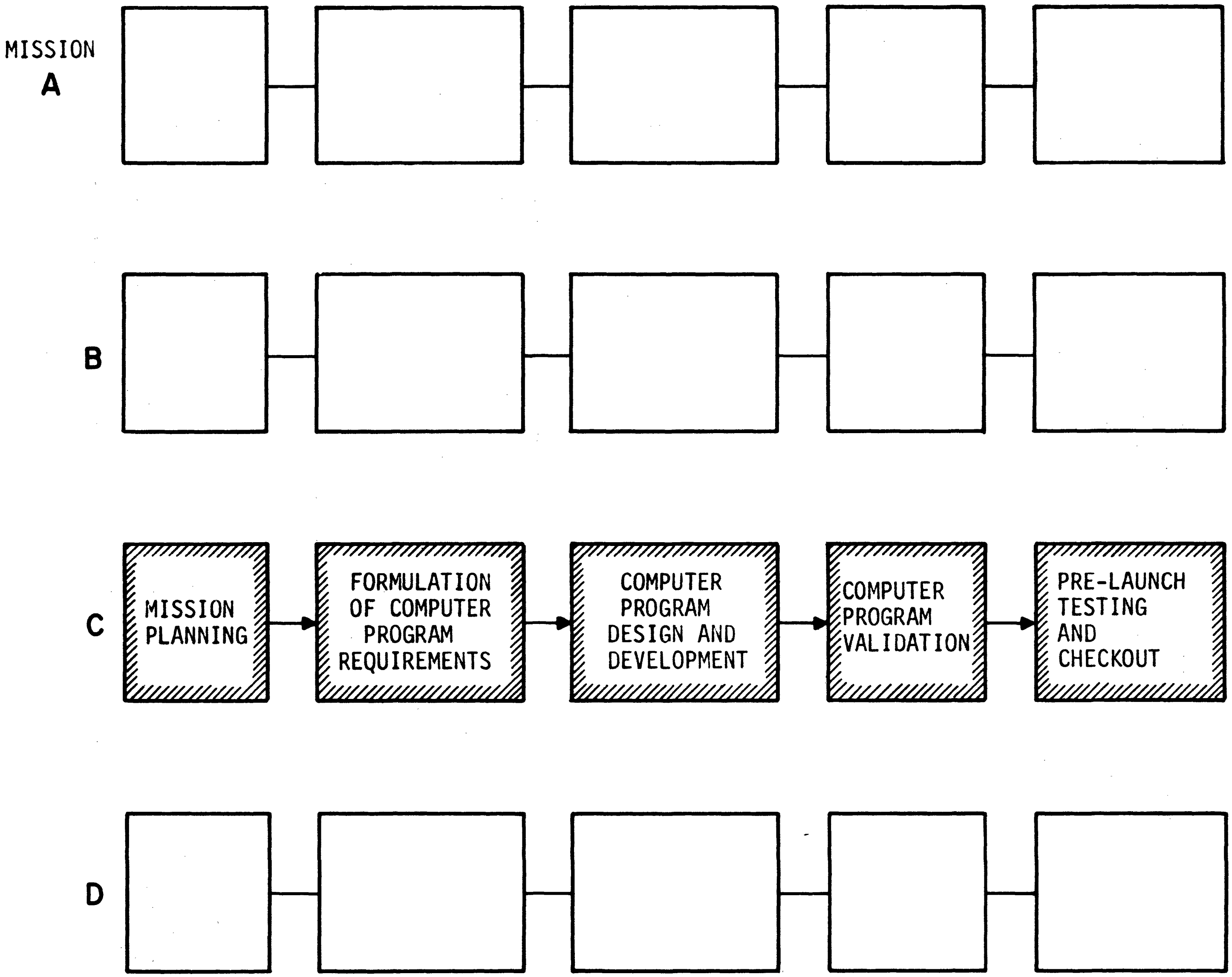

Figure 1 Existing Software Preparation Process

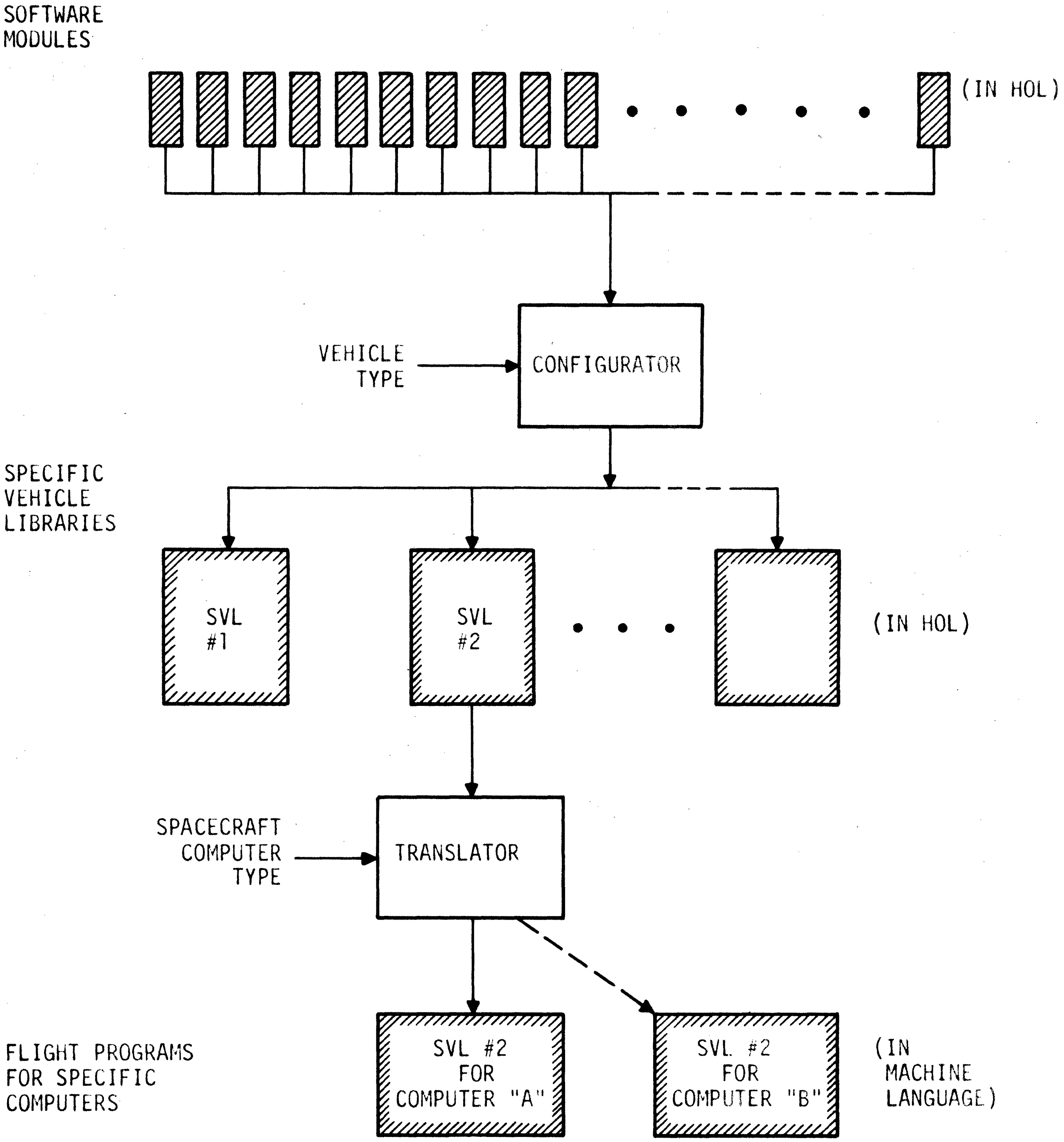

Figure 2 Flexible Guidance and Software System

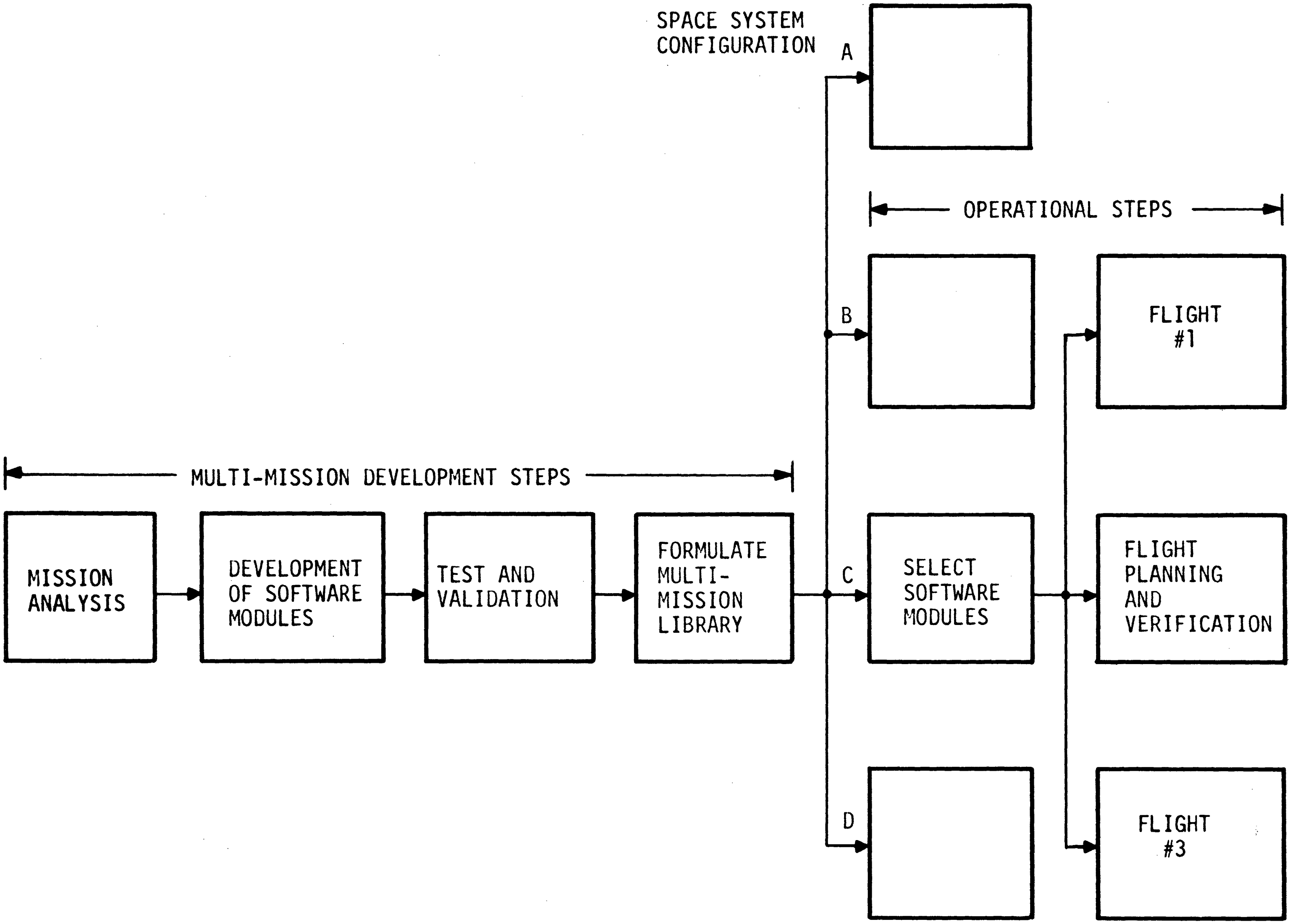

Figure 3 Desired Software Preparation Process

1.2 ACCOMPLISHMENTS OF PHASE I

The Phase I work by IBM indicated that the concept of highly automated, quick-reaction mission planning and execution is feasible, and that considerable reduction in software costs and lead times can be achieved. The major accomplishments of Phase I were:

- Preparation and test of Trajectory Generation Program (TGP) and Trajectory Optimizer Program (TOP). These programs demonstrated the flexibility and adaptability of the Mission Planner.
- Preparation and test of a Direct Search Optimization Program (DSOP) for use in TOP.
- Preparation and test of an optimal-explicit guidance algorithm (OP-EX). OP-EX is versatile and is usable for exo-atmospheric ascent, orbit transfers, rendezvous, intercept and deboost.
- Development and test of a predictive guidance system for use during reentry by lifting-body vehicles. A basic approach was developed for displaying the destination relative to maximum range and cross-range.
- Specification of performance requirements for components in autonomous navigation systems for three (3) classes of missions.
- Estimation of spacecraft computer requirements.
- Functional design of a Ground Operating System (GOS) of advanced support software.
- Functional design of the Configurator in GOS to assemble Specific Vehicle Libraries (SVL) from the Multi-Mission Library (MML).
- Functional design of an On-Board Executive System (OBES) to manage resources and schedule application programs in real-time on the spacecraft computer.
- Determination of basic crew tasks and computer/display requirements for manned missions.

- Analysis of error propagation in the guidance and navigation systems.

1.3 OBJECTIVES OF PHASE II

IBM proposes to further the system design initiated in Phase I towards the development of detailed design specifications for the component parts of FGSS. Efforts will be concentrated on the critical elements of FGSS thus providing an increased measure of feasibility of the system concepts.

In particular during Phase II we will:

Task 1 (Integrated Guidance System)

- Simulate and prepare design specifications for Trajectory Generation.
- Implement vehicle, range safety, and navigation system hardware constraints in mission planning and active guidance.
- Simulate and prepare design specifications for Trajectory Optimization.
- Simulate and prepare design specifications for Atmospheric Ascent Guidance and Exo-Atmospheric Guidance.

Task 2 (Software System)

- Prepare design specifications and test a scheduling algorithm for the On-Board Executive System (OBES).
- Develop and demonstrate interrupt supervision routines for the OBES.
- Develop and demonstrate an experimental Configurator to assemble Specific Vehicle Libraries (SVL) from the Multi-Mission Library (MML) of the Ground Operating System (GOS).
- Examine compatibility of the Space Programming Language (SPL) being developed by SDC and the advanced FGSS concepts.

Task 3 (Man/Computer Interface)

- Determine the mission control tasks during the pre-launch mission planning phase.
- Determine the manner in which these tasks are to be implemented.

Task 4 (Error Analysis)

- Determine the error sources inherent in the functional elements of the guidance system.
- Determine the effect of residual errors on mission planning.
- Provide, if necessary, the capability for including error models in mission planning.
- Establish techniques for assessing the accuracy of system performance as part of the launch-pad verification process.

Figure 4 is a summary of the program activities and shows completed Phase I work and proposed Phase II effort.

A detailed Program Plan for Phase II will be submitted within thirty (30) days of contract authorization. This plan will expand the task descriptions given in Section 3 of this proposal and will incorporate changes resulting from contract negotiations and early technical liaison.

Design Specifications will be submitted as informal engineering documents for Air Force approval no later than 240 days after contract go-ahead.

Progress Reports will be submitted monthly detailing progress-to-date, problems encountered, future plans, and an updated version of a task descriptive network.

A Final Technical Report will be submitted summarizing all significant technical accomplishments. A draft of this report will be submitted for Air Force approval 230 days after contract go-ahead. Final distribution will be made 270 days after contract go-ahead.

1.4 IBM QUALIFICATION

IBM is uniquely qualified to perform the proposed program. IBM obtained a broad and thorough understanding of the guidance and software preparation problem during the Phase I study and has established a sound technical approach for solutions to the problems.

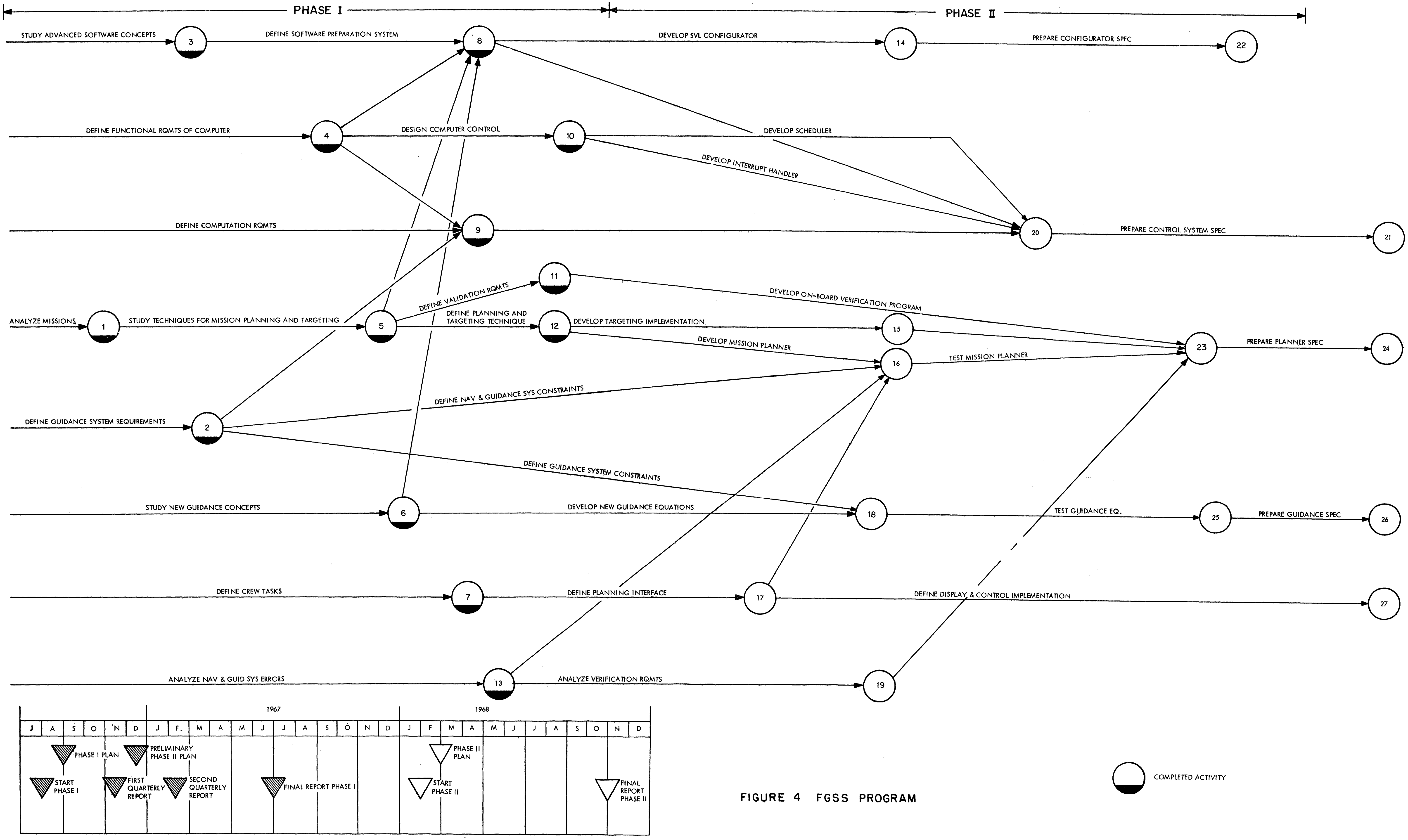

FIGURE 4 FGSS PROGRAM

Key personnel from the Phase I study are available for the Phase II effort. In particular, Dr. Howard M. Robbins will be Technical Manager, Dr. Thomas F. Clancy will lead the Mission Planning and Guidance efforts, and Mr. Joseph J. Constable will lead the Support Software development.

Personnel assigned major roles in the study are familiar with the computer programs developed during Phase I and have a thorough understanding of the basic problems and the capability to carry the developments into the new areas as required by the statement of work.

Management and control of the study program will be performed by personnel in the McLean Building facility of the Space Systems Center. The major portion of the study will also be conducted by personnel in the McLean Building. However, experienced IBM personnel in the other IBM facilities will be utilized as consultants or task assistants.

Several computer programs have been prepared and/or debugged during Phase I in support of the IBM technical approach. These include:

- Trajectory Generator Program (TGP)
- Trajectory Optimizer Program (TOP)
- Optimal Explicit Guidance (OP-EX)
- General Integrated Simulation Model (GISMO)
- Autonomous Navigation Simulation (ANS)
- Inertial Navigation Error Analyst and Diagnostician (INEAD)
- Matrix Manipulator (MM)

More details on these programs are presented in Section 5 of this proposal. These programs provide a unique basis for extending the developments of FGSS at minimum time and cost to the Air Force.

Section 2 of this proposal discusses the problems to be solved in furthering the development of FGSS and presents the selected technical approach. Section 3 presents a detailed plan of action for the study including a schedule and proposed man-power distribution for the tasks.

The management team is described in Section 4 and a summary of applicable IBM experience is presented in Section 5.

Section 2

TECHNICAL DISCUSSION

2.1 DESIGN OF THE SYSTEM

The basic framework for the design of the Flexible Guidance and Software System (FGSS) was established during the Phase I study. This was based on analyses of the planning, guidance, and software preparation problems and a technical approach selected by IBM utilizing advanced planning and guidance algorithms and advanced concepts and procedures for a ground-based software preparation system.

FGSS (Figure 2) is an integrated set of hardware and software designed to produce software for space flights at minimum cost and with short response times. Depending on the mission and the booster/spacecraft configurations, FGSS will produce software for a spacecraft computer, a ground-based computer -- or a combination of both. FGSS is a ground-based, software producing system utilizing flexible mission planning and guidance techniques.

Cost reduction and reaction-time reduction stem from the basic system concept of reusable software modules.[1] Reusability is exploited across the spectrum of Air Force missions; from one hardware configuration to another, and from one flight to another. The same set of modules may not be used on each flight, but each flight program is assembled, for the most part, from a single, finite collection of software modules.

The collection of software modules is called the Multi-Mission Library (MML). A Configurator (a software program running on a ground-based computer) selects appropriate modules for a pre-specified vehicle (booster/spacecraft/guidance system, etc.). A Translator (a software program) converts the Specific Vehicle

---

[1] Reaction time on the launch pad (and cost) are, of course, reduced by utilizing the generalized mission planner and OP-EX.

Library (SVL) from the Higher Order Language (HOL) of FGSS to the machine language of the spacecraft computer.

The inherent flexibility of FGSS is exercised by the Configurator and the Translator.[1] FGSS can produce software for a multitude of vehicles, spacecraft computer models, and ground-based computer models. This flexibility, coupled with the general purpose modules, will provide the adaptability needed to cope with changing vehicles, changing mission objectives, and hardware advances.

The flexibility of FGSS is further illustrated in Figure 5. This diagram depicts, without regard for implementation, the combination of software modules into flight programs. It should be noted that a final flight program will, in general, be assembled from software modules of varying complexity. The MML will contain the lowest level elements as well as combinations (functions, modes, phases) known to be of general or multiple purpose utility.

Some typical reusable software modules are:

- Mission Planning Routines
- Guidance Laws
- Gravity Models
- Trigonometric Routines
- Numerical Methods
- Vector and Matrix Manipulators
- Schedules
- I/O Supervisions
- Main Storage Supervisions
- Error Recover Routines
- Self Test Routines
- Diagnostic Routines

2.1.1 MISSION PLANNING AND GUIDANCE

The key to success of FGSS lies in the development of general purpose mission planning algorithms and general purpose guidance laws. In Phase I, IBM developed mission planning algorithms utilizing self-optimizing techniques to provide a general capability for trajectory selection within restraints of system hardware, booster performance, and launch conditions. Highly explicit guidance algorithms were developed to minimize the number of guidance modules needed

---

[1] An additional aspect of flexibility of major importance is provided by the generalized planner and OP-EX guidance.

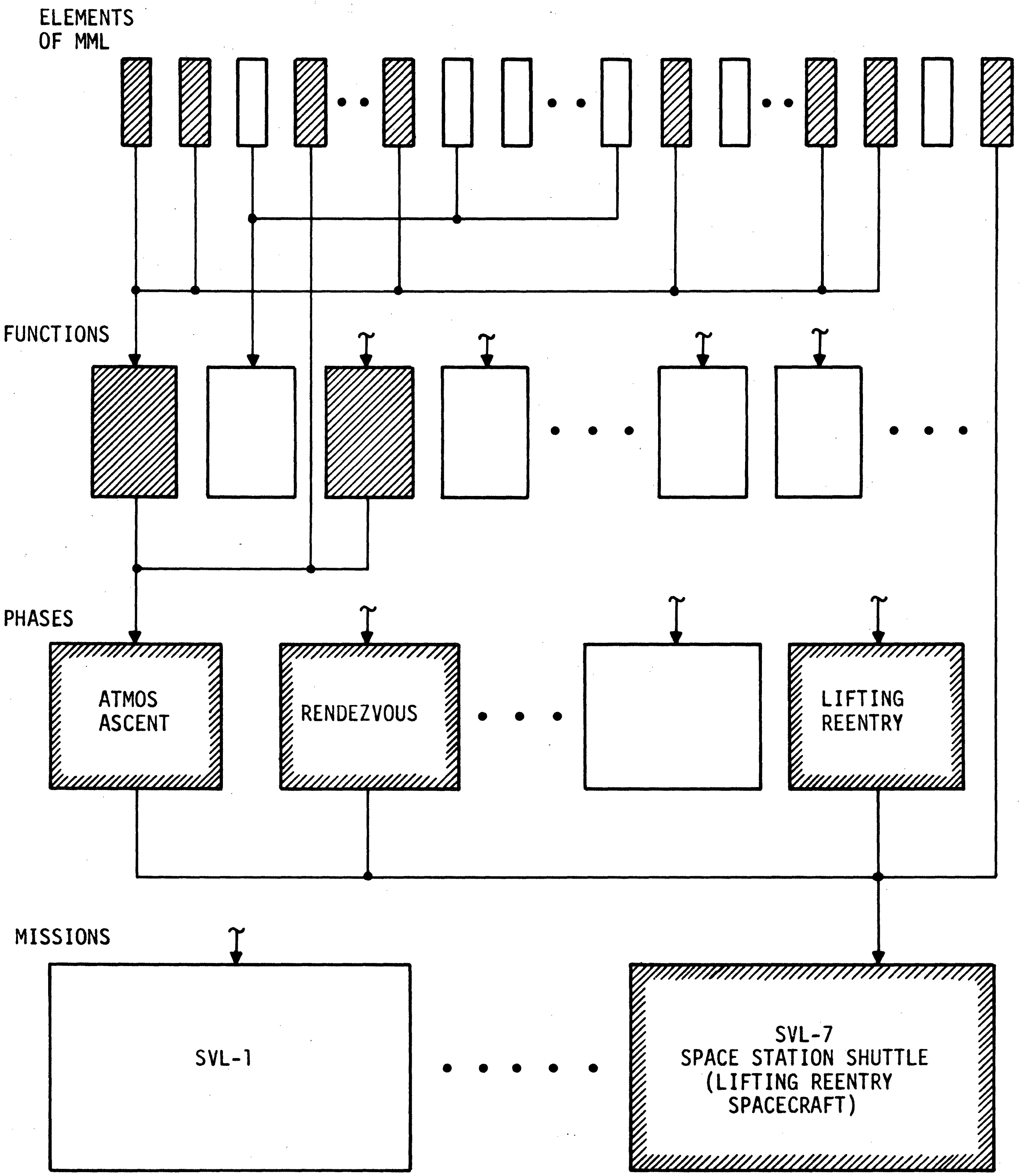

Figure 5 Reusable Software Modules

to cover the spectrum of missions and minimize the need for precomputations. The guidance algorithms were designed for maximum compatibility with the mission planning algorithms to minimize computer storage requirements.

The fully developed FGSS will have the flexibility to product mission planning and guidance software for space systems with varying degrees of sophistication. For unmanned vehicles with little more that an autopilot and command receiver, FGSS will produce a mission planner and active guidance algorithm suitable for use on a ground-based computer. For manned vehicles with autonomous operating ability, FGSS will produce a mission planner for use in the spacecraft computer that will perform mission planning on the launch pad and allow mission plan refinements to be made in space.

The work in Phase I gave a strong indication that the approach chosen for FGSS mission planning and guidance is technically feasible and that it will lead to the desired reductions in cost and reaction time.

A major portion of the Phase II Program will be devoted to further development of mission planning and guidance alforithms. The algorithms developed in Phase I are described in Section 2.2 together with the proposed extensions in capability.

2.1.2 THE GROUND OPERATING SYSTEM (GOS)

The Ground Operating System (GOS) plays three fundamental roles in FGSS.

- Analysis and development of software modules
- Formation and maintenance of the MML
- Configuration and test of SVL's

Where a ground computer is required for final mission planning or for support of active guidance, these functions may also be performed by GOS.

The interrelationship of these roles in software preparation and utilization is illustrated in Figure 6. This is an extension of the software cycle shown in Figure 3.

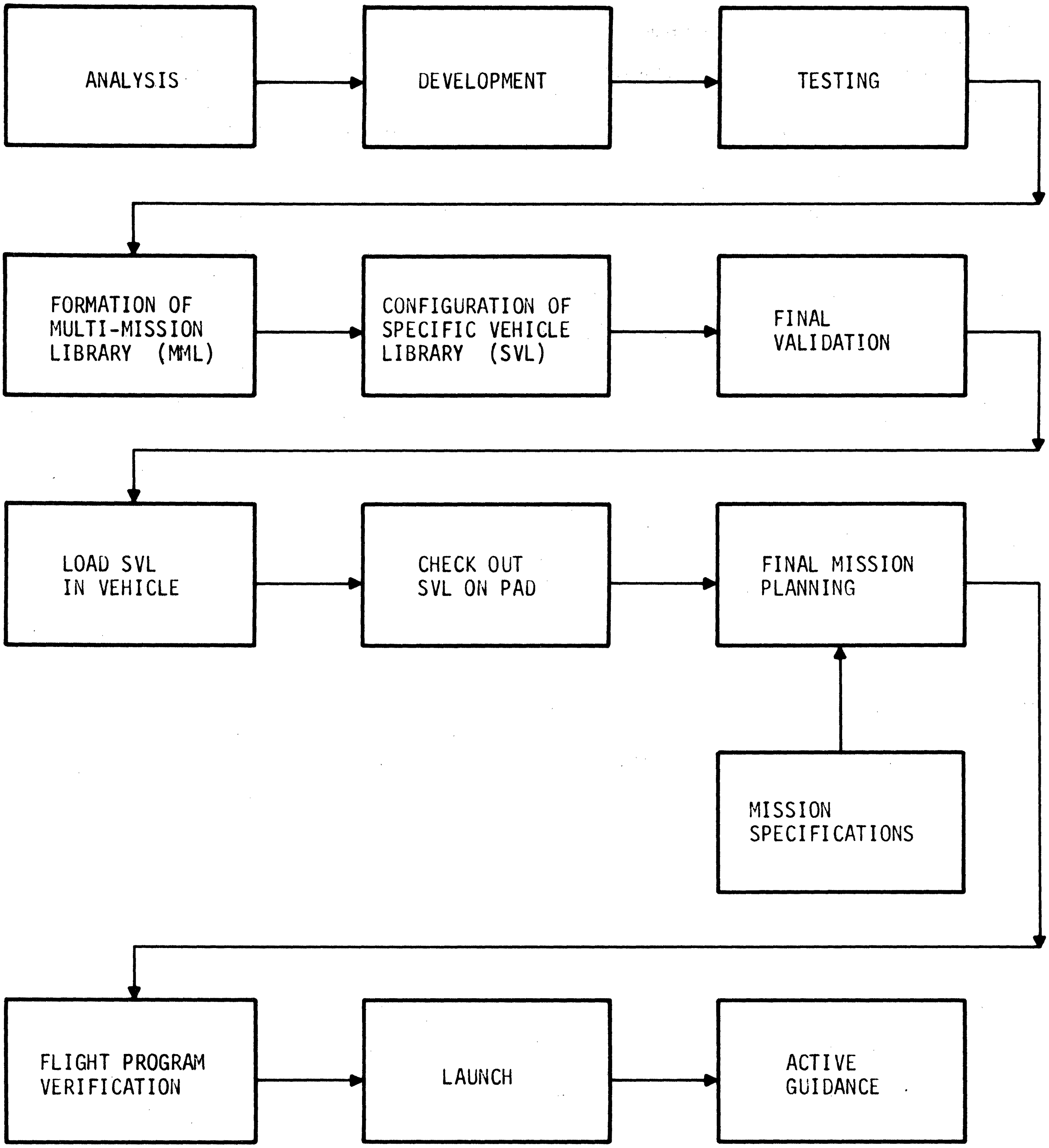

Figure 6 Advanced Software Implementation Cycle

GOS is an integrated set of hardware and software; it is the operating portion of FGSS. The hardware will consist of a general purpose computer (with associated input/output devices) and facilities for real-time simulation of space operations.

The major software components of GOS are:

- Control Program
- SVL Configurator
- Language Translator(s)
- Simulation Processors
- Linkage Editor

A block diagram of GOS is shown in Figure 7.

The control program schedules work, allocates computer and software resources, and performs I/O. The control may be sequential, but ideally permits concurrent execution of a number of tasks based on priority.

A job control language is used by programmers to communicate with all components of the system through the control program. The job control language enables the programmer to specify jobs, define the job steps, specify the source and organization of input data, and specify the organization and destination of output data.

Software modules on the MML are generally applicable to many mission classes, vehicle configurations, and spaceborne computers. In most cases modules will be stored in a Higher Order Language. The language translators (assemblers and compilers) convert source modules to object (machine language) modules.

The linkage editor combines modules from the language translators into higher-level modules ready for loading and execution.

The functions of GOS may be categorized as program preparation, program integration, and simulation. Preparation consists of analysis, coding, debug, and component-level test. Integration consists of the generation and maintenance of a multi-mission program library, and the extraction and combination

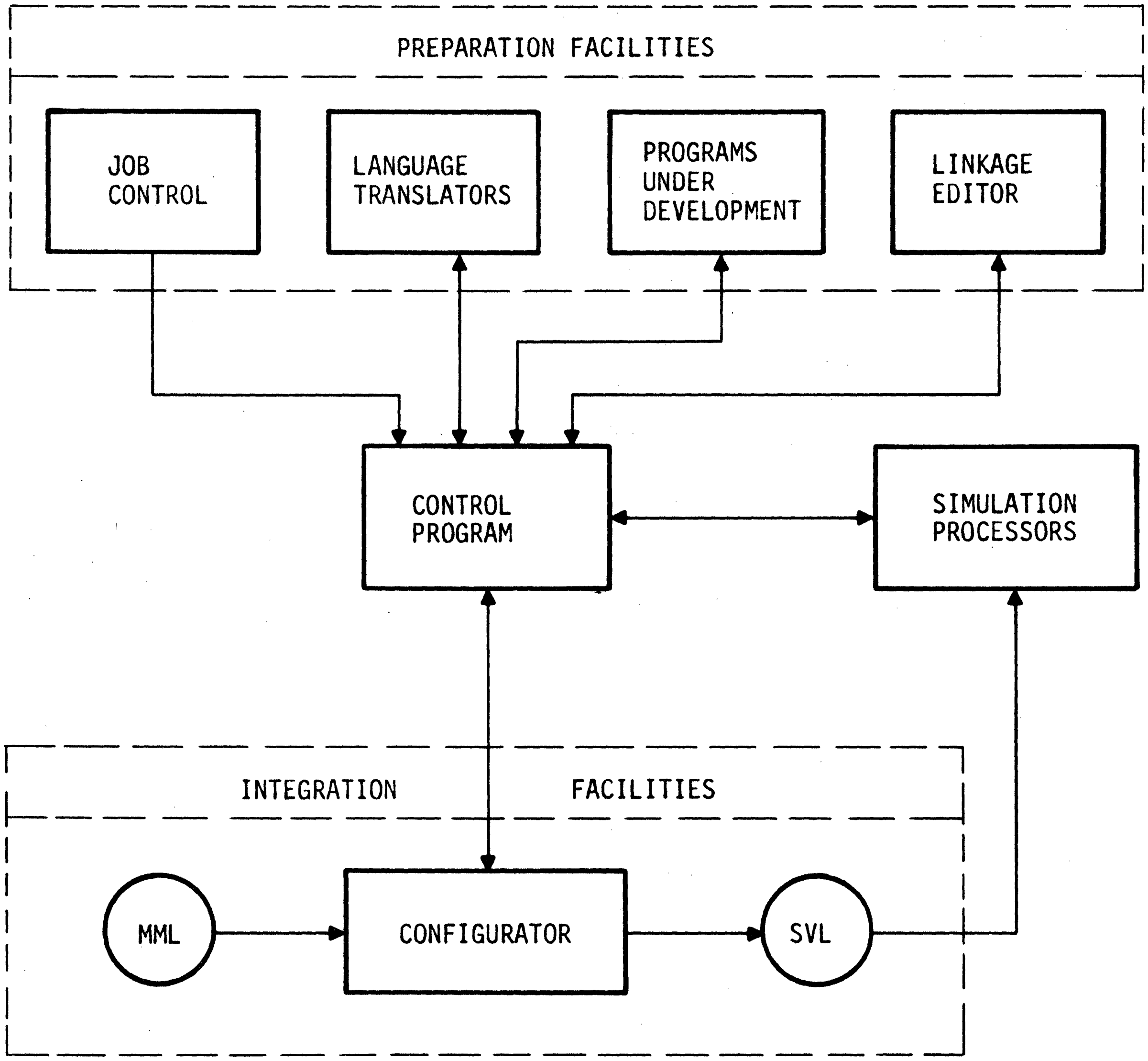

Figure 7 Ground Operating System

of MML modules to form an MML subset referred to as the specific vehicle library.

Interpretive and/or real-time simulations are made to insure that the flight program functions as designed. During validation, software is tested using a wide range of typical mission parameters. In FGSS applications, specific parameters are determined by the onboard mission planner at prelaunch time or re-determined while the mission is in progress. The final verification of the software is accomplished using the onboard computer or a launch support computer. Verification is performed using specific mission parameters generated by the mission planning routines and the onboard software.

The key design requirements for GOS are:

1) Maximum sharing of software elements (and of the effort needed to generate, validate, and maintain these elements) among different vehicles and missions, so as to minimize duplication of effort.

2) A system design which permits the generation and validation of new software elements and their integration into validated flight programs to be performed independently of the precise, quantitative details of mission specifications, and prior to the final definition of such specifications.

The Phase I study produced a preliminary design for a Ground Operating System (GOS). An objective of the Phase II study is to develop one of its key elements: the Configurator. The details of the proposed development are discussed in Section 2.2.4.

2.2 DESIGN OF THE KEY ELEMENTS

The success of FGSS depends on the satisfactory developments of four items:

- A mission planner with multiple mission capability
- A set of explicit guidance laws with multiple mission capability
- A configurator to form SVL's from the modules of MML
- An executive control system for the spacecraft computer.

Basic design requirements and technical approaches for these elements were established in Phase I (Reference 1). Most of the Phase II program will be devoted to their development.

2.2.1 MISSION PLANNING AND GUIDANCE WITHOUT PRECOMPUTATION

To be independent of precomputation, planning and guidance algorithms should be designed to accept explicitly stated mission objectives and data on vehicle stage masses, mass flow rates, etc. From this information, a flight plan and guidance policy should be established to meet the given objectives and satisfy all constraints. The only general way to accomplish this is to choose a tentative guidance policy, predict the trajectory that will result from it, compare this trajectory with the objectives and constraints, and iteratively adjust the tentative policy until the objectives and constraints are predicted to be satisfied. During the mission, real-time guidance commands would be based on present-state information and the guidance policy established during the mission planning phase.

The above approach can be used to find a feasible trajectory and guidance policy, i.e., one which satisfies all objectives and constraints. However, all feasible policies and trajectories are not equally desirable. To maximize the vehicle's performance capability, maintain maximum freedom of choice in planning, and increase constraint margins, the optimum plan and guidance policy must be found.

With the exception of generally-impractical methods of exhaustive search, all optimization techniques presently known find only local optima rather than the global optimum. To find a "good" local optimum, the only recourse is to use theory, insight, and experience to choose likely starting points for optimization, optimize from several starting points, and compare the results.

To keep computer requirements within reasonable bounds, and permit consideration of as many alternatives as possible, every possible means of reducing and simplifying the optimization problem should be employed. One very important way of reducing the computational requirements is to divide the optimization procedure into two phases; a preliminary phase of rough optimization which is speeded by using every practicable approximation, and a subsequent refinement phase that produces a more nearly optimal solution, if required. If the rough optimization is sufficiently accurate to permit valid comparison of different alternatives (i.e., different local optima) the refinement phase need only be applied to one selected alternative rather than to all.

IBM's mission planning and guidance techniques are based on the above concepts.

Mission planning is performed before launch, or during coast phases, and is concerned with the overall mission from present state to completion. Its primary product is phase targeting, i.e., a specification of end conditions for the first upcoming powered phase. These end conditions may be expressed as a prescribed aim point, or as orbit parameters for a coast arc.

Active guidance is an on-line, rapidly repeated replanning of the present powered phase only. It is based on the latest information from the navigation system and the end conditions provided by the mission planner.

The mission planning and guidance algorithms developed during Phase I are described in Sections 2.2.2 and 2.2.3 respectively. Before proceeding to these details, there are other technical considerations that affect the design of both items.

Fortunately, for the space systems being considered, the quality of optimization is not very sensitive to approximations. The use of approximations causes the computed "optimal" trajectory to deviate from the true optimal trajectory, but the performance penalty incurred by this deviation is of second order in the deviation, and will therefore be small unless the trajectory deviations are large. This is the reason for the success of explicit ascent guidance schemes which use quite crude approximations for trajectory prediction (e.g., use of an average value for gravitational acceleration) but give nearly optimal performance.

Because of this insensitivity, it is possible to ignore small effects in the refinement phase of optimization. These affects are accounted for by targeting-corrections so guidance accuracy is not degraded.

For rough optimization, all exo-atmospheric mission phases can be treated with sufficient accuracy by using closed (Keplerian) formulae for all coast phases and approximating the powered phased by impulses. The excellent accuracy of the impulsive approximation for in-space powered phases has long been known to mission analysts. It has been confirmed analytically by Robbins (Reference 2) and numerically by McCue (Reference 3) and by many others.

Kepler arcs and impulsive maneuvers are widely used in mission analysis, and have been successfully used in the Variable Point Guidance scheme (Reference 4). They reduce the computation time for a trial trajectory by several orders of magnitude, as compared to numerical integration. Also, they reduce the optimization problem from an infinite-dimensional one (i.e., a problem in the calculus of variations) to a finite-dimensional one of minimizing a function of n variables subject to a set of constraint relations.

For the refinement phase, performance can be improved at small cost by introducing closed-form approximate corrections for the secular effects of atmospheric drag and gravity harmonics during coase phases, and for the finite durations of in-space powered phases. Expressions for the latter correction have been given by Robbins (Reference 2). The nonsecular effects of drag and of gravity harmonics need not be considered even in a refinement phase of optimization so long as appropriate corrections are made in the targeting for a powered phase.

The optimization techniques used in refinement phases need not have broad convergence properties since it always starts from a good first approximation. However, it should have rapid convergence to reduce the number of trial trajectories that need be generated in the refinement phase, and to make possible an accurate re-optimization in one iteration after a small change in present state or other parameters.

The desired rapid convergence in the refinement phase may be obtained by use of a second-order method, or a first-order method accelerated by use of information generated as by-products of the rough optimization phase (sensitivity coefficients, etc.).

In the approach taken by IBM, the active guidance algorithms used for atmospheric ascent and atmospheric reentry will be used by the Mission Planner.

This will save storage capacity and provide hightly accurate trajectory plans. OP-EX, the guidance algorithm used for active guidance outside the atmosphere, is also available for use in mission planning. This algorithm will provide highly accurate predictions of Δ V required, end state, and burning time.

Although precomputations must be minimized for efficient mission planning and guidance, changes in the vehicle configuration and characteristics that take considerable time to execute and check out do not require an immediate response in the corresponding software changes. Therefore, it is important to distinguish between mission-dependent parameters and vehicle-dependent parameters.

To maintain the quick-reaction capability, all mission-dependent parameters must be either explicitly given, or generated on board with little or no delay. For parameters which are mission independent but dependent on vehicle characterictics not subject to change on short notice, precomputations are acceptable. Vehicle data such as stage masses, expected burning rates, and thrust levels, etc. should be treated as mission data.

The results of vehicle-dependent precomputations can be expressed as constraints, or as restrictions on the allowable control policies and trajectories. A principal example is the atmospheric phase of ascent, where there is essentially only a two-parameter family (lauch azimuth and kick angle) of allowable trajectories.

2.2.2 MISSION PLANNING ALGORITHMS

The primary requirements for the design of the Mission Planner for FGSS are:

- Plan any probable mission within the performance and constraint capability of the vehicle.
- The algorithms must perform on-pad mission planning rapidly enough so this function is not the pacing item for reaction time.
- In addition to the primary mission plan, the system must be capable of providing alternate-mission and abort planning for various degrees of flight hardware failure.

- For autonomous space systems, the planner must provide the capability for planning and replanning missions after launch.
- The system must be capable of accepting updated navigation data, equation-parameter data, and decisions from external sources (e.g., crew members or ground stations).
- The system must be capable of communicating information to crew members and/or ground controllers (via adequate displays) to enable them to make mission decisions such as selection of final plan.
- For autonomous systems, the spaceborne software system must have self-contained capability for verifying mission plans generated or modified in orbit.
- For all vehicles, the spaceborne software system must interface with on-site ground support equipment for pre-launch hardware calibration and flight program verification.

These requirements will be met by the Mission Planner being developed on the FGSS program. The work in Phase I demonstrated the feasibility of the selected approach.

Figure 8 illustrates the operation of the proposed mission planner in a prelaunch planning mode. Alternate preliminary trajectories are generated and optimized. The parameters for the guidance equations are then generated for the selected trajectory. These parameters and other data are then combined with selected software modules to form a flight program which is verified to insure operational readiness.

IBM's approach to mission planning is characterized by:

- Initial trajectories chosen by empirical rules.
- Explicit computation of trial trjectories.
- Optimization to improve performance and satisfy constraints.

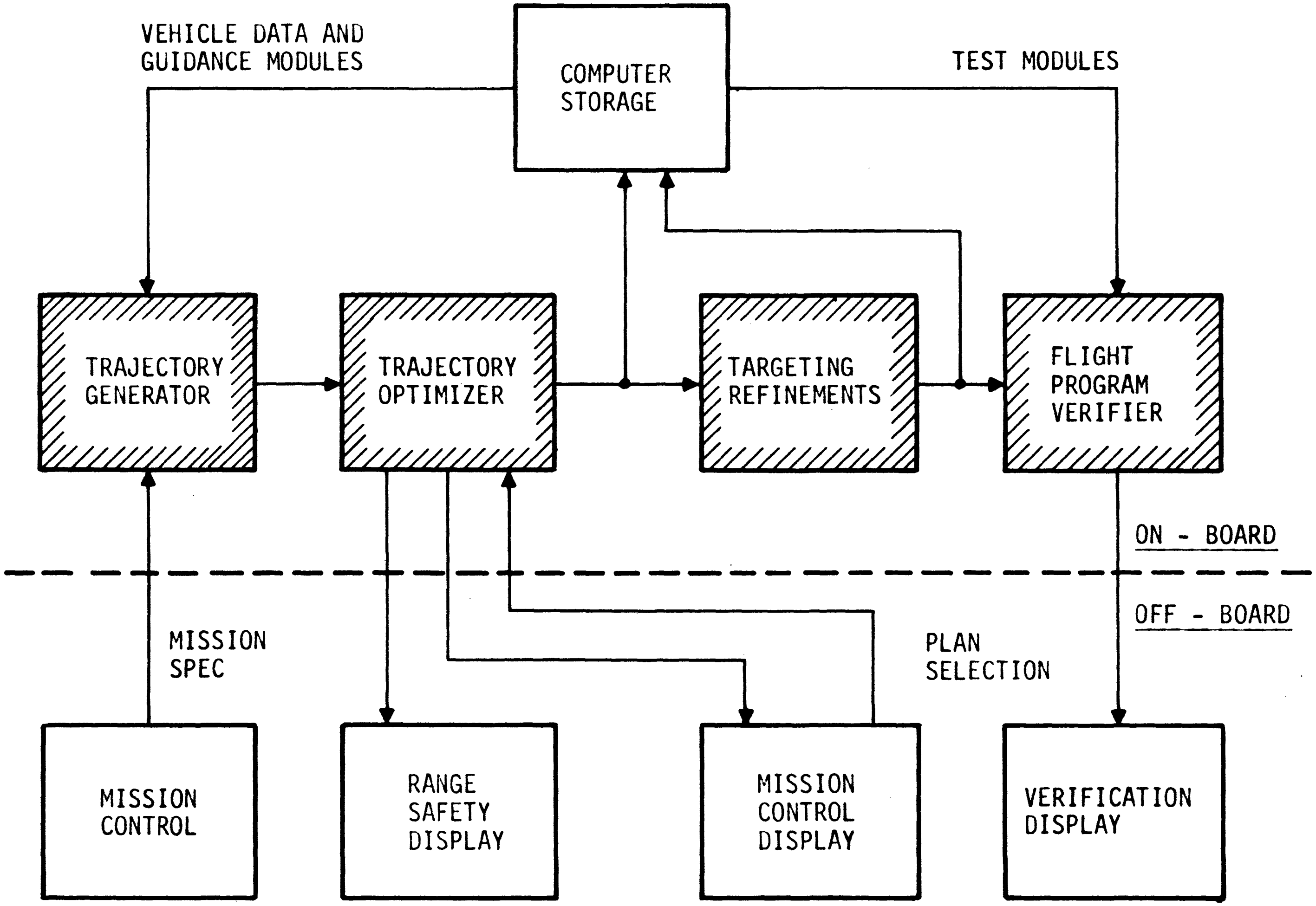

Figure 8 Semi-Automatic Mission Planning

The inputs to the Trajectory Generator are (see Figure 8): a mission specification (from Mission Control when on the pad or by the crew when in orbit), various guidance routines and planning programs, and vehicle data from the on-board computer. The idea of preliminary trajectory generation is to use planning constraints and approximations in order to rapidly generate trajectories which meet mission objectives. An example of a planning approximation is a spherical earth model; requiring an orbital burn to occur at the line-of-nodes is an example of a planning constraint.

Certain mission and trajectory constraints are introduced at the preliminary trajectory stage. These include: launch azimuth limits, earliest and latest possible launch time, maximum ΔV available, and latest mission completion time.

The type of mission and its particular objectives and constraints will determine the way in which the mission segments will be constructed and options used. A rescue mission, for example, will result in a direct ascent rendezvous planning option (as well as others) because of the desire for a quick rendezvous.

Planning routines available are:

- Analytical expressions for launch conditions (to minimize out-of-plane angle with the target orbit plane).
- VPG for generating three-burn rendezvous maneuvers from a parking orbit.
- RENDI (Rendezvous Intercept), a technique developed during Phase I for generating a nearly optimal two-burn rendezvous from a parking orbit.

Figure 9 illustrates how a typical flight plan would be constructed. The mission illustrated involves rendezvous with a target in a known orbit.

The plan involves launching at t_L and at azimuth A_L to minimize out-of-plane angle with the target's orbit. The atmospheric phase of ascent is calculated with analytical equations and vehicle-dependent (but not mission-dependent) parameters. The exo-atmospheric ascent involves injection into a circular parking (radius R_p, orientation $\underline{n}_p$). The calculations are done with the OP-EX guidance algorithm.

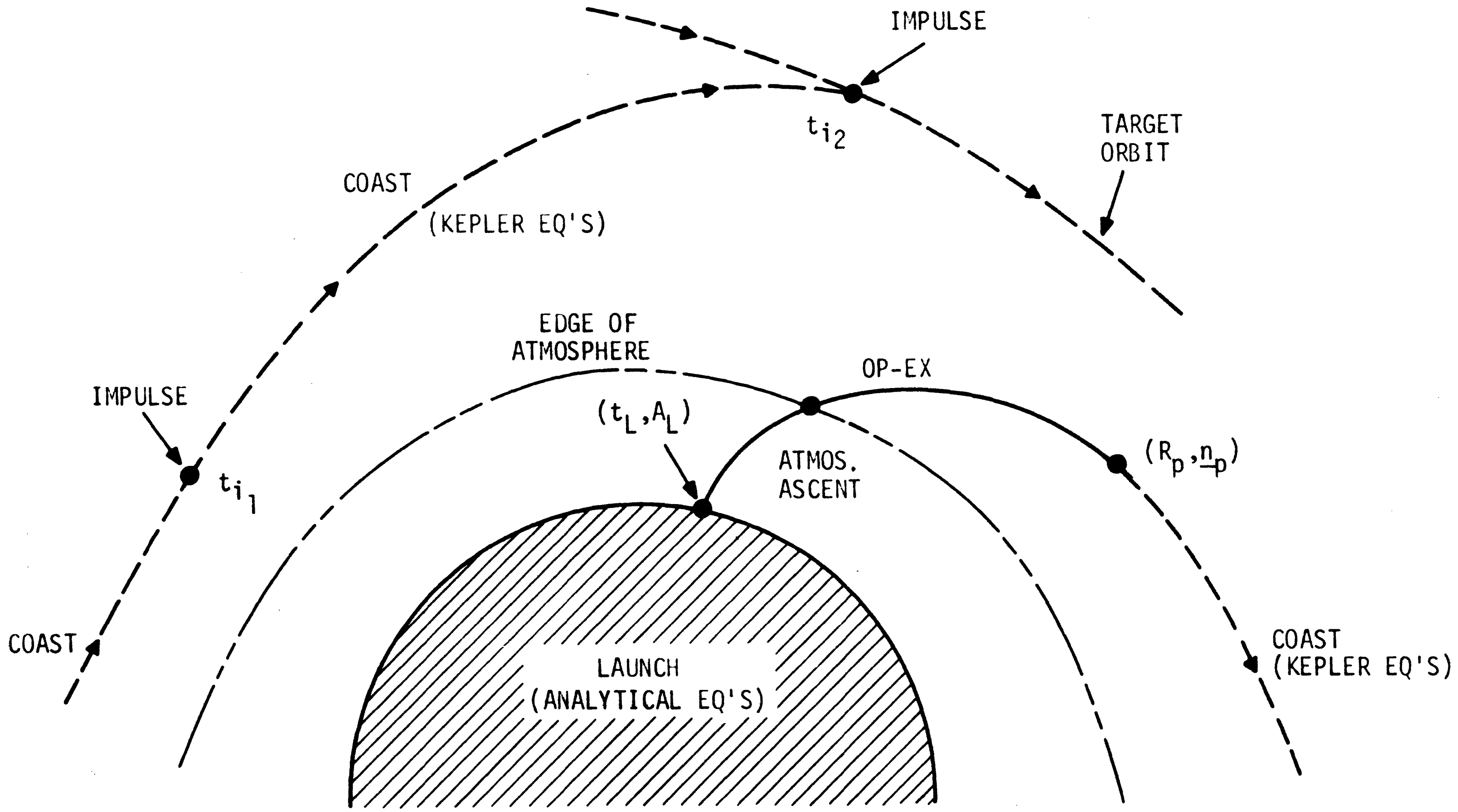

Figure 9 Trajectory Generation

After phasing in the parking orbit, a two-burn transfer is planned with RENDI which results in gross rendezvous at the target's altitude. This transfer is specified by the times of the two burns.

Three-burn orbital rendezvous maneuvers could also be planned using VPG to provide alternate plans.

The second step in the planning process involves optimization of the preliminary trajectories. The plans selected for optimization will result from preliminary tradeoffs of time and fuel. trajectory feasibility, and other factors such as Range Safety.

The optimization program allows for relaxation of the planning constraints in order to increase over-all performance and allows for enforcement of mission and trajectory constraints which could not be included in the preliminary trajectory generation.

The input to the optimization program involves a specification of the preliminary trajectory in terms of the parameter set (output from TGP) constraints on the parameters, type of minimization desired, (ΔV, time), step sizes, control data, and constants.

An initialization block sets up the paths for the appropriate payoff function and identified the parameters to be varied and their limits. The initial trajectory parameters are used in the payoff function to obtain the initial value of the payoff. A search routine is then called and the optimization proceeds; for each parameter variation, the corresponding payoff value is computed.

The program developed in Phase I represents a general function minimizer of the firect-search type. This allows both flexibility and computational efficiency because analytical derivatives are not required and a problem is completely defined by the equations for the function to be minimized and the constraints.

The results of the optimized trajectories are displayed to Range Safety and Mission Control for final selection. The selected trajectory is then refined to compute final guidance parameters.

The trajectory refinements for guidance account for the approximations that are made during trajectory generation and optimization. The major approximations are impulsive orbital burns and spherical earth. The former can be corrected by analytical techniques, the latter by various closed-form solutions or numerical integration of pertinent mission phases.

Two simulation programs, Trajectory Generator Program and Trajectory Optimizer Program were written and tested during Phase I. These programs have demonstrated the flexibility and performance required for mission planning but require extension and refinement in order to be suitable for operational use. The most important improvements are described below.

Trajectory Generation

The trajectory-generation program must be improved to increase the number of different types of missions it can handle. For example a more flexible set of options is needed for orbit insertion; besides insertion into an orbit with a fixed orientation in inertial space, there should be provisions for insertion into orbits of variable orientation (e.g., orbits unspecified except for perigee, apogee, and inclination). This capability can be obtained by use, in mission planning, of additional end-condition options available in the OP-EX guidance algorithm.

The Mission Planner uses analytic approximations for predicting position and velocity at the end of the atmospheric phase of ascent. At present, these approximations are based on a fixed kick angle. More general formulae must be developed to permit ascent planning with a variable kick angle, and a variable time of switches to exo-atmospheric guidance. These may be empirical formulae, or quasi-explicit formulae based on zero-lift ascent. (The actual guidance of ascent may employ trajectories with nonzero lift for greater optimality, but the difference is not significant for planning.)

The trajectory algorithms used in mission planning must be refined to account for the secular effects of earth oblateness and (possibly) atmospheric drag. Non-secular perturbations due to these causes can be ignored during preliminary planning and mission optimization, but must be considered during active guidance, or compensated by target offsets computed as the final step of mission planning. Computations of offsets may be by numerical integration, or explicit formulae, or a combination of the two. Explicit formulae for the position and velocity perturbations caused by the quadrapole part of the Earth's gravity field have recently been developed at IBM. These formulae are based on the equations developed by Lion and Handelsman (Reference 5) for the transition matrix associated with Kepler arcs. They are valid for circular, elliptic, and hyperbolic trajectories. They are also simple enough to be usable in active guidance if desired.

Constraints

The Phase I work relating to mission and vehicle constraints was preliminary and must be extended in Phase II. Present and future constraints relevant to mission planning must be identified and classified. Some present constraints are due to subsystem hardware limitations which will be reduced or eliminated by "next generation" hardware. These constraints need not be considered. Constraints expected to be of continuing importance must be analyzed to determine whether the constraint should be handled by planning or guidance. For example, a constraint specifying a required attitude at vehicle cutoff would be assigned to the guidance algorithm (OP-EX) whereas line-of=sight constraints for viewing of orbital maneuvers by tracking stations would be the responsibility of the planner.

Trajectory Optimization

To maintain compatability with the planning algorithm changes (and guidance changes), TOP must be updated. The new guidance algorithms must be incorporated into the payoff function; the search list extended to include new parameters (such as the kick angle) and provisions incorporated for enforcing a set of constraints of varied types (mostly argument bounds and nonlinear inequality constraints, but some equality constraints).

In addition to direct enforcement of argument bounds (a feature of the present TOP) two different techniques for handling constraints will be considered. One is a sophisticated form of the well-known "penalty function" method, which was tested with some success during Phase I. It is well suited for mission-planning situations where there is no way to tell in advance which inequality constraints will be active and which will not.

The other technique enforces identified constraints (equality constraints, or inequality constraints known to be active) by using a variant of Newton's method to compute some variables as dependent functions of the rest, thus reducing the dimensionality of the search space. This latter technique is favored when a solution has been found but must be continuously updated as the situation changes (e.g., on-line use). It is presently used in TOP via OP-EX, which iteratively varies steering-policy parameters to make the predicted cutoff state watch given requirements. The same technique, slightly generalized, can be incorporated directly in TOP.

Improvements in the convergence speed of the optimization program are lightly desirable. Considerable improvement is possible by changing the program logic to eliminate repetitious calculations in the evaluation of the payoff function. For example, if trajectory parameters for ascent to orbit are compared with their past values and found unchanged, the past values of Δ V, cutoff time, cutoff state, etc., can be used in lieu of recomputation.

Much more significant improvement is believed possible by use of sophisticated search techniques with accelerated convergence. Many such techniques are currently being developed by various individuals and organizations. In order to conserve development effort, novel techniques of this sort will not be developed during Phase II. Instead, optimization algorithms developed elsewhere will be considered for suitability, and one or two promising candidates will be incorporated in TEXT (the search-algorithm part of TOP).

2.2.3 GUIDANCE ALGORITHMS

The spaceborne guidance software must provide the following capabilities:

- The algorithms must provide control that is optimal or near-optimal with respect to fuel and/or time, and satisfy all trajectory constraints.

- The algorithms must enforce all constraints that are of frequenc occurrence, and allow for incorporation of additional constraints unique to particular missions or to particular vehicles and their navigation/guidance hardware.
- The algorithms must be capable of executing flight plans without significant degradation of optimality, in spite of off-nominal vehicle performance.
- The guidance function must be performed independently of the mission planning and navigation functions.
- Mission planning and guidance must take account of constraints caused by the navigation system.

The ultimate guidance comcept would be a single generalized law applicable to all guided phases of all vehicles and yielding optimal performance. A theoretically possible solution is in-flight use of an advanced, general method of trajectory optimization powerful enough to handle all cases, but this cannot presently be considered practical. The practical design goal is a set of simple guidance laws that result in near-optimal performance and can be applied with little or no precomputation to various vehicles for different missions. Whenever possible, the guidance laws should be adaptable for use in mission planning to minimize the number of computer programs required.

IBM's approach has been to develop guidance laws for three categories of flight; atmospheric ascent, exo-atmospheric maneuvers, and atmospheric re-entry. Due to limited funds, effort will be placed on atmospheric ascent and exo-atmospheric guidance only during the Phase II study.

2.2.3.1 Atmospheric Ascent Guidance

During atmospheric ascent, the structural loading and temperature limits of the booster constrain maneuverability and trajectory variations. The booster is nominally flown along a zero load or "gravity turn" trajectory after an initial vertical rise and velocity perturbation (kick-over).

Each booster is capable of flying any one of a family of trajectories similar to "gravity-turn" trajectories within its allowable ascent corridor. These trajectories may deviate from exact gravity-turn trajectories if this is advantageous. The bounds of the corridor are determined by structural and thermal constraints. A family of allowable trajectories is thus definable by two parameters - launch azimuth and kick angle. The selection of an atmospheric ascent giving over-all optimality requires a search over these parameters within the allowed corridor. This search can be performed as part of the trajectory optimization during mission planning.

The preliminary atmospheric ascent algorithm employed in Phase I involved a vertical rise, a kick-over maneuver (which was vehicle dependent but mission independent), and a gravity turn (zero lift) trajectory until the dynamic pressure had fallen below approximately 30 psf.

In Phase II, it is proposed to develop a more general and flexible scheme for atmospheric ascent. The kick parameter will be made mission dependent and included in the optimizer. The steering law will accommodate non gravity-turn trajectories and the time to switch to exo-atmospheric guidance will be calculated dynamically. This algorithm must be formulated so as to satisfy vehicle constraints during atmospheric ascent.

No attempt will be made to implement the strict theoretical optimum control for atmospheric ascent. This is known, both by theory and by experimental experience, to be extremely difficult, and computationally expensive. A much easier and more efficient alternative is available, namely, a quasi-optimal control policy based on the known form of the optimal solution. The trajectories generated by this policy consist of an initial segment in which the $Q\alpha$ constraint is not active, then a segment in which the vehicle flies the $Q\alpha$ constraint, and then switchover to the exo-atmospheric mode. The $Q\alpha$ constraint is explicitly enforced when applicable. For the initial segment, an empirical control law is used (e.g., trapezoidal pitch program) with parameters determined by the mission-planning algorithm.

2.2.3.2 Exo-Atmospheric Guidance

For exo-atmospheric ascent, IBM has developed OP-EX. As the name implies, OP-EX is both optimal and explicit. That is, it accepts present state and vehicle propulsion characteristics as input and generates an optimal (fuel minimal) trajectory satisfying explicitly stated constraints and final conditions. OP-EX requires no precomputations and is compatible with mission planning requirements.

Optimality is useful not only for its own sake but also because it allows for a consistent approach to all exo-atmospheric maneuvers (except possibly the final phase of rendezvous). The explicit aspect of the alforithm allows flexibility in the description of both vehicle and mission objectives. The operation of OP-EX is illustrated in Figure 10.

OP-EX is based on a numerical solution of the equations for optimal rocket-powered flight outside the atmosphere. The predicted trajectory is obtained by integrating the differential equation of state. The control parameter is the direction to orient the vehicle's thrust. For an optimal trajectory, in the sense of minimum burning time (or maximum payload), the control is related to the primer vector (λ) which itself must satisfy the equation $\dot{\lambda} = \underline{G}\ \underline{\lambda}$. Lambda ($\lambda$) is the adjoint of the velocity vector which Lawden entitled the primer vector.

Optimal guidance reduces to determining the vector function lambda satisfying the above equation. The problem is: (1) the initial conditions on λ are unkown (optimization theory tells us how to propagate the control policy but not where to start), and (2) the gravity gradient matrix ($\underline{G}$) in the equation for lambda depends on the trajectory. This leads to a two-point boundary value problem.

The particular numerical technique that has been employed by OP-EX for solving the problem involves integration of the equations from given and estimated initial conditions, and alteration of the estimated parameters to make the predicted final conditions equal the desired final conditions. The desired objectives are provided by the mission planner.

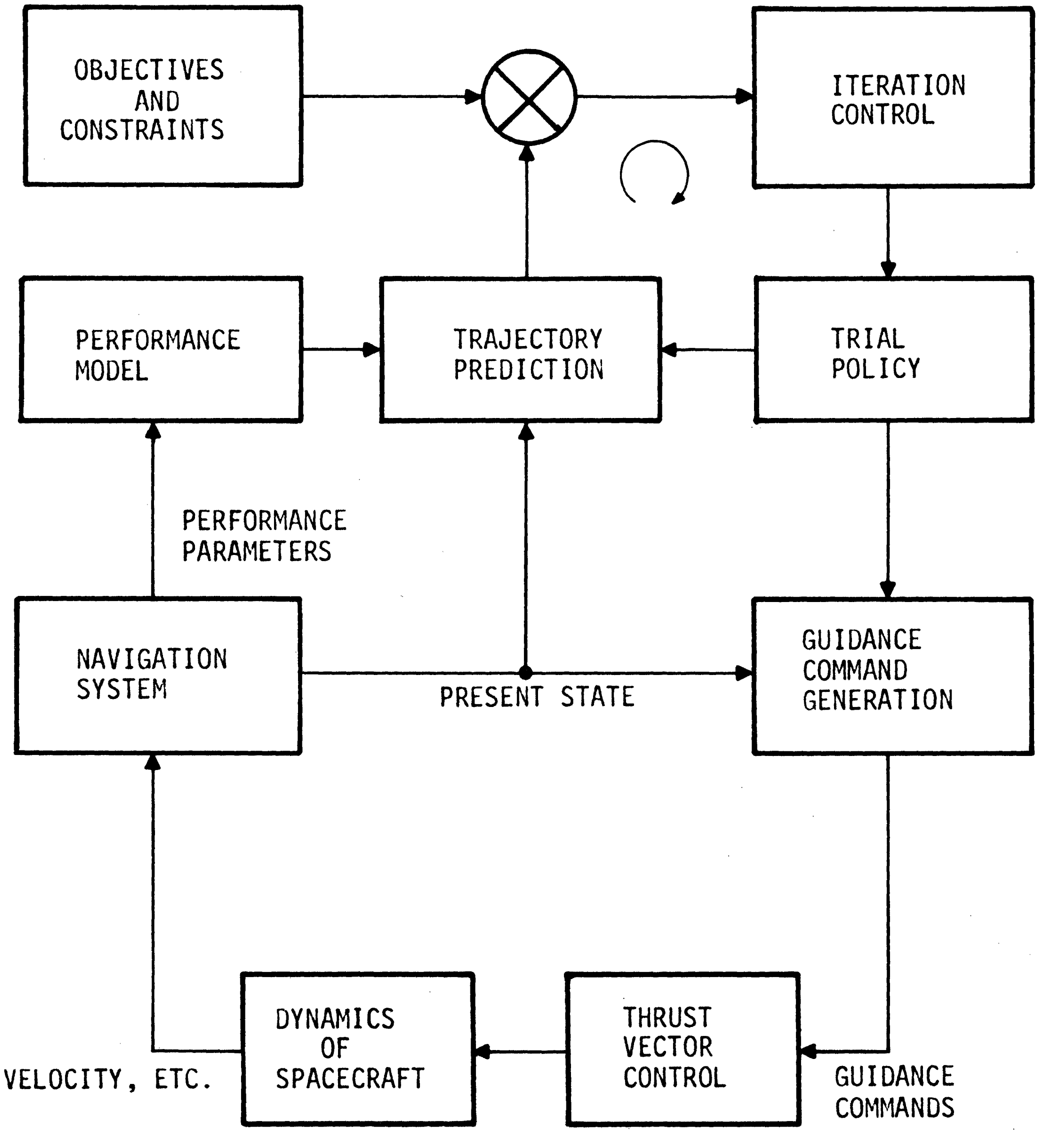

Figure 10 Quick-Reaction Guidance

The first aspect of the solution (trajectory integration) is handled with an Explicit-Predictor integrator using closed-form equations for integrals involving thrust acceleration. This makes accurate integration possible with large step sizes. Step sizes of 100 seconds and larger were successfully used in Phase I.

Estimation of the unknown parameters followed by the Explicit-Predictor integration produces a set of final conditions. The difference between these conditions and the desired final conditions is the error that must be nulled. This is done by an iterative adjustment of the unknown parameters.

In the iteration process, the matrix which relates changes in errors at final time to changes in the parameters of the problem is calculated using finite differences. This matrix is used to compute the change in a variable to null errors at the end of the mission phase.

OP-EX guidance is a versatile form usable for exo-atmospheric ascent, orbit transfers, gross rendezvous, intercept, deboost - in fact, for all powered phases outside the atmosphere. For each of these guidance requirements, the resulting fuel expenditure is minimal.

For any given powered phase a variety of alternative terminal conditions can be specified by the mission planner. For example, orbit insertions will be possible with specification of (1) orbital plane and orbit orientation, size, and shape in that plane, or (2) orbit inclination, latitude of perigee, size, and shape, or (3) orbit inclination and latitude, longitude, or altitude of orbit perigee. OP-EX has modes which provide optimal guidance for intercepting a given target vehicle, with or without velocity matching.

OP-EX has flexible provisions for propulsion performance description (thrusts, stage masses, mass rates). An unlimited number of stages can be specified and, in theory, any modelable function of time can be used to represent engine mass flow rate and exhaust velocity. Even exotic schemes like the Saturn V propellant-utilization system (which involves a change of thrust level at a variable time after stage ignition) can be implemented satisfactorily. In fact, the optimal time at which to switch the level of thrust could be specified by OP-EX.

The OP-EX algorithm is computationally efficient so as to be practical for real-time guidance, yet is sufficiently accurate to be used for rapid prediction of ΔV requirements for flight-planning purposes. It provides predicted vehicle states at future staging times (and other critical times) which can be used in spent-booster impact prediction.

The explicit formulae used in the computation of the required velocity for OP-EX are also used to generate candidate trajectories for mission planning and trajectory optimization.

The planned improvements and extensions of OP-EX, include (1) implementation of additional modes (i.e., additional options for the form of the final conditions to be satisfied at cutoff) in forms usable by the mission planner, (2) corrections for the effects of the Earth's quadrapole field, and (3) provisions for enforcing trajectory constraints such as attitude rate limits, prescribed attitude at cutoff, etc. Also, OP-EX will be interfaced with the atmospheric ascent routine.

OP-EX presently has several options for the form of the conditions to be satisfied at cutoff, but only three of these options have been used in mission planning. It is proposed to select and incorporate additional options for greater flexibility, and make all options usable by the planner.

The proposed corrections for the effects of the Earth's quadrapole field use recently-derived closed-form expressions which are simple enough for use during active guidance. An alternative is the use of target offsets computed by numerical integration, before active guidance begins. This is slightly less accurate, but more economical in computer requirements. The oblateness correction is only necessary when the cutoff conditions are such as to place the vehicle on a free-fall trajectory which satisfies certain future objectives; for example, rendezvous with a target vehicle. For orbit insertion, the closed-loop nature of the guidance will insure no injection errors due to oblateness, and the performance penalty is negligible.

The theoretically most elegant way to incorporate constraints (like attitude rates) into OP-EX is to formulate an expanded optimal control problem (with additional state and adjoint variables, etc.) whose solution gives an "optimal" trajectory satisfying the constraints. However, this is expensive computationally. Also, the theoretically "optimal" turn out to have some features that are undesirable from a practical standpoint. Therefore, a heuristic approach has been developed in which optimal control theory is used to suggest appropriate forms for the control policies. Implementation is not made in an exact form: instead, simple approximations are used where analysis indicated the resulting performance loss to be negligible, and qualitative practical considerations are taken into account in choosing a control law which is near-optimal, reasonable in other respects (e.g., does not call for abrupt changes of attitude rate just before cutoff, if the cutoff attitude or attitude rate is to be accurately controlled) and computationally convenient.

2.2.4 SVL CONFIGURATOR

Reusable software modules in a higher-order language source form are catalogued in the Multi-Mission Library. Prior to storage, the modules are tested to be sure they meet all performance and interface specifications.

The configurator extracts copies of multi-mission library modules, retrieves special-purpose modules, and combines these modules to form a smaller library of modules for a specific booster/spacecraft configuration called the Specific Vehicle Library (SVL). A simplified diagram of the operation of the SVL Configurator is shown in Figure 11.

The main input to the Configurator is a description of a specific vehicle configuration. Its output is a Specific Vehicle Library containing the flight programs necessary to perform all mission functions within the anticipated capability of the vehicle.

The directory search provides an internal list of all MML modules applicable to the vehicle specified. It then retrieves these modules from the MML and insures that all references to points outside the module are resolved. Those not resolved are (in normal operation) references to special purpose modules not included on the MML. These special modules are retrieved and merged with multi-mission

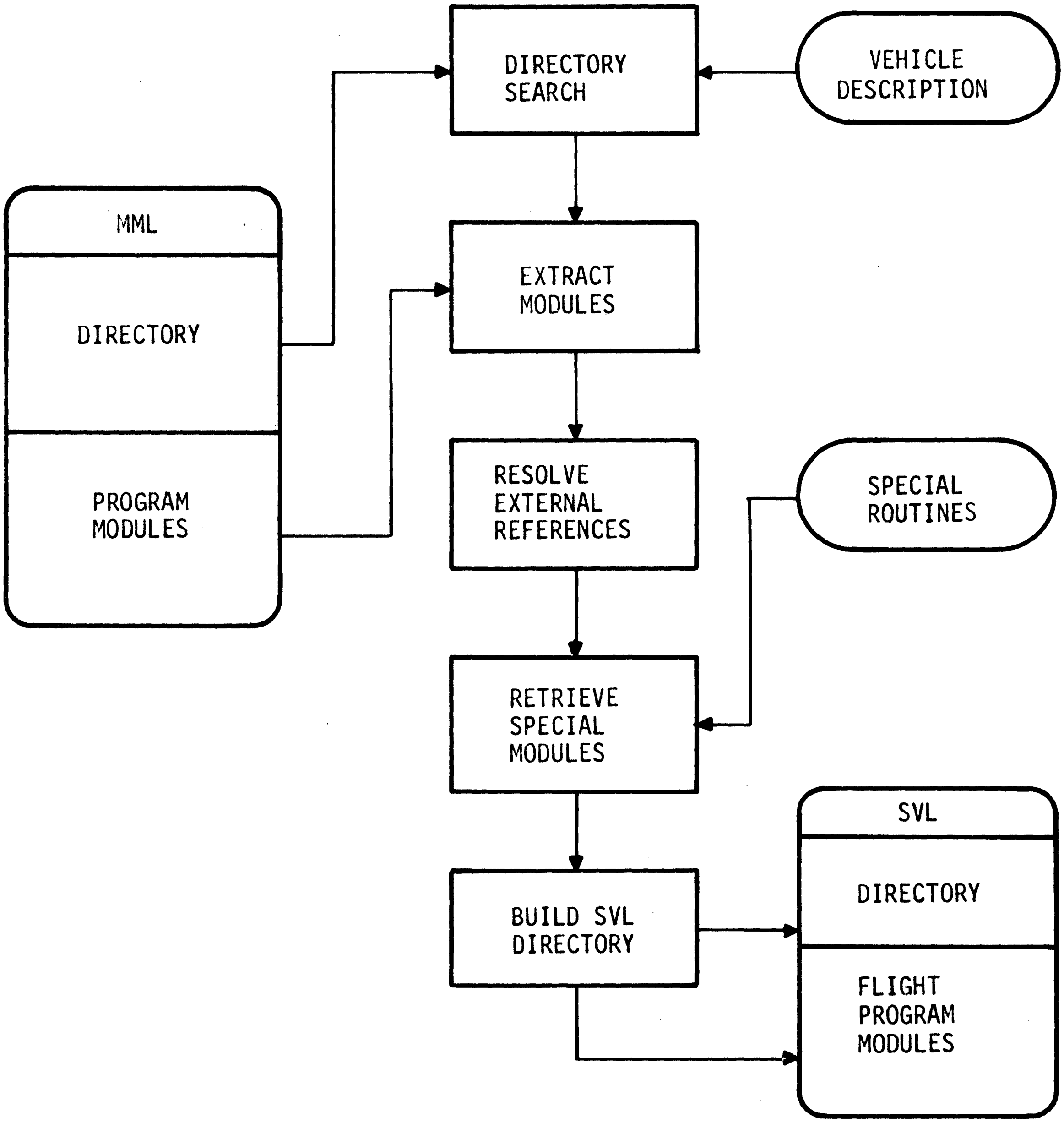

Figure 11 SVL Configurator

modules. All modules are then combined to form programs and are outputted, together with a directory, on the SVL device.

The Multi-Mission Library directory is necessary for library maintenance and for orderly search and configuration.

The directory consists of several indexes. Each index consists of several entries. Each entry contains a pointer to another index or (at the lowest level) a pointer to a physical module. Assuming the use of disc storage, each pointer is a track address consisting of head, cylinder, and cell numbers.

Figure 12 depicts the search of MML utilizing the directory. The illustration is top-to-bottom chaining to specify a module which computes velocity to be gained for the OP-EX function of the orbital phase of a T III-C rendezvous mission. This would be specified by the coding nomenclature shown at the bottom of Figure 12.

If the low order term (VG) of this specification were deleted, all elements required to implement the OP-EX function would be identified. The T III-C entry on the vehicle index denotes only one particular configuration of the T III-C vehicle. It is necessary to provide an entry for each configuration of interest since the SVL content depends on the sensor repertoire and final stage vehicle as well as on booster type.

The illustration shows unidirectional chaining from top to bottom. Actually, the chaining is bidirectional to permit downward search for all modules which constitute a higher level module, and upward search to find all higher level modules in which an element is used. The bidirectional chaining requires that each directory entry include pointers to both successor and predecessor nodes.

A useful analogy is Bill-of-Material processing, where a product is broken down into assemblies, subassemblies, and parts. The directory can also be used to build a where-used list to determine which higher level modules require each element in the MML.

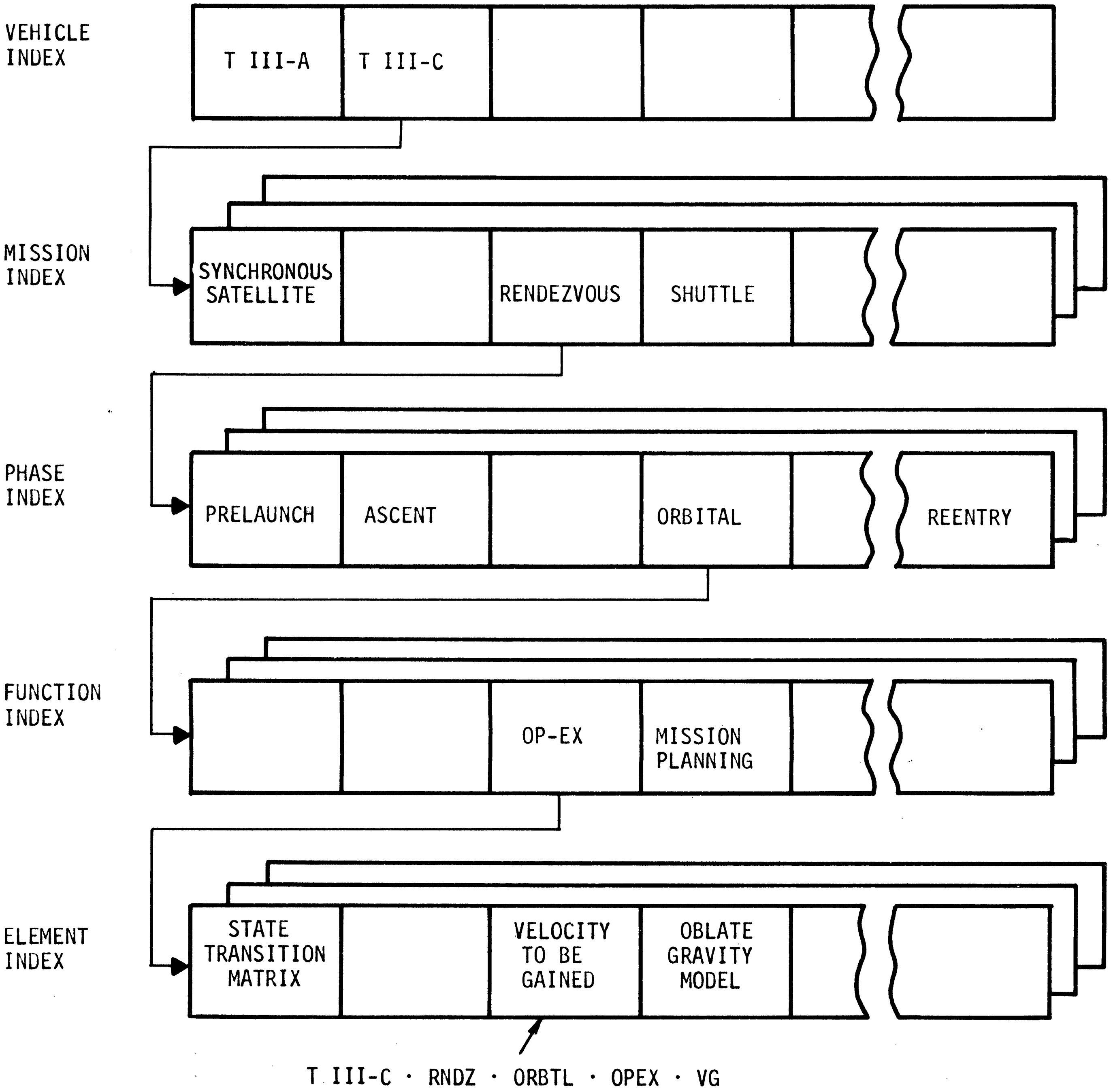

Figure 12 MML Directory Search

Figure 12 is limited to operational flight programs. A higher level index (called the program type index) could be considered. It would include entries for mission analysis, support programming, and flight programming - with the flight programming entry pointing to the vehicle index shown. The dominant node of the entire structure might be (for example) "Space Systems Programs".

The Phase I effort resulted in a definition of the MML and SVL Configurator concepts as described above. In the proposed Phase II effort, the SVL Configurator will be developed further and an experimental version will be demonstrated.

The development of the SVL Configurator includes the following:

- Establishment of a hierarchy of modules for the MML.
- Definition of standard format for modules.
- Definition of standard interface for major modules.
- Establishment of rules for modular combinations - or definition of factors affecting choice of combinations.
- Definition of the structure of the MML directory.
- Definition of the input spec requirements for configuration control.
- Definition of the interface with OBES.

Specific tasks associated with these efforts are described in Section 3.2

Some important problems which must be solved during Configuration development are described below.

- A convenient and concise method (language for satisfying control inputs to the Configurator must be devised. Ideally, only the identification of the vehicle and its major subsystems would be required; in practice, it may be required to choose only one of several modules which are applicable to the vehicle or subsystem. Thus, specification by module name may in some cases be required.

- The Configurator will have to provide for the retrieval and linkage of special-purpose routines not on the MML and not requiring compilation. Inevitably, some module needed will not appear on the MML either because it has not yet been written or because the on board computer required a special purpose routine. Routines that are heavily machine dependent and those for which storage and speed efficiency is critical cannot be written as general purpose routines in any existing HOL.

- The MML directory format and the method of chaining its entries must be devised to disassociate the physical organization of modules on the device from their logical organization in the directory. The directory must also facilitate (a) rapid search, (b) ease of inserting and linking new entries when modules are added to the MML, and (c) permit the use of an MML directory subset as the SVL directory.

- A means must be devised to insure that each module extracted leads also to the extraction of (a) all other modules which it invokes (calls), and (b) all other modules required to provide input data to each module extracted. The extraction of any given module must cause the extraction of all other modules which service it by providing either (a) specified (constant) input parameters, or (b) input data provided via computation in an invoked module.

2.2.5 ON-BOARD EXECUTIVE SYSTEM

The on-board executive system (OBES) is a control center for all on-board software. It operated during all mission phases to maintain control of the interfaces between application programs, vehicle devices, and crew members. It allocates and controls computer system resources in accordance with application program requirements, provides recovery capability in the event of errors or unusual conditions, and resolves conflicting demands for resources.

The executive control function is not new. It has existed in some form in previous space systems. The proposed OBES is, however, more centralized and is vital to the functions of the overall Flexible Guidance and Software System.

In the proposed software system of modules (MML and SVL), control must be isolated from the application programs. The modules or programs required at any given instant during a mission are not always known in advance. To insure that programs do not interfere with one another or issue conflicting demands on computer resources, their control must be centralized.

An executive control system has been designed to control and make efficient use of the computer memory space and CPU time. The OBES will perform the following functions:

- Interrupt Supervision
- Scheduling of Application Programs
- I/O Supervision
- Main Storage Supervision
- Handling of Emergency Conditions
- Utility Service

In providing the above services, the OBES utilizes some core storage and CPU time. But all of it is not directly chargeable to the executive system. Routines like Interrupt Handler, I/O Supervisor, and Utility Services would have to be included in the applications programs if not provided by the executor. In return for a small overhead cost in terms of core memory and CPU time, the on-board executive system provides the following advantages:

- Reusability of executive software for a variety of missions and vehicles.
- Ease of incorporating changes in the flight software by isolating control functions from application programs.

- Flexibility for handling program configurations which cannot be determined in advance by application programmers and which may change as the mission progresses.
- Avoidance or resolution of conflicting demands for computer resources by several application programs.
- More orderly and efficient allocation and control of main storage, I/O devices, CPU time, and shared software.
- Relief to the application programmer from detailed I/O programming, detailed linkage sequences, and routine computational functions which are provided as system services.

The executive system must be designed in a modular form to provide the capability of implementing different versions for various space missions and computers. The configuration will depend upon the complexity of the programs required to meet the mission requirements. The minimal system would have a simple scheduling routine and storage supervision would not provide for temporary storage allocations. Advanced systems would require all of the capability described above.

2.2.5.1 Interface with MML

Of particular importance to the OBES operation is the structure of the software modules. As a minimum each module must have:

- Program Control Table
- Program Common
- Text
- Relocation Dictionary

This structure is somewhat different from those used on present computers. To operate under the executive system, the Program Control Table must contain module priority, repetition rate, service class, and status. This information is used by the Scheduling routine. Program Common is data used by several routines within the module.

The Text contains instructions and data.

The Dictionary contains the information necessary for SVL configuration and the information required by the Main Storage Supervisor of OBES to relocate this program in memory when necessary.

Each module may be defined according to service class. This classification is determined at coding time by the nature of the program function. A typical classification scheme is as follows:

- Immediate response programs are those which must be executed as soon as the request for them has been recognized. The high priority interrupt processing routines fall in this category.
- Precisely cyclic are those programs which must be executed at exact intervals in the major cycle.
- Nominally cyclic programs are those which must be executed at a certain repetition rate, but not at exact intervals in the major cycle.
- Background programs are those which can be scheduled whenever there is time left over in a major cycle.

Application program scheduling and interrupt handling are two major areas of OBES requiring additional development. Efforts in the proposed Phase II program will be concentrated on these items.

2.2.5.1 Scheduler

The overall operation of the scheduler is shown, in simplified form, in Figure 13. The scheduler determines what supervisory services are required or which program is to be given CPU time. The application program proceeds until it is complete or until a predetermined time interval has elapsed. At either of these times, control is again given to the scheduler. If no interrupts have been stacked, the scheduler checks for service requests, such as program loading, or interrupt processing. When the appropriate service routines are executed, control is returned to the scheduler. This process is continued until all programs have been executed the proper number of times in a given major cycle.

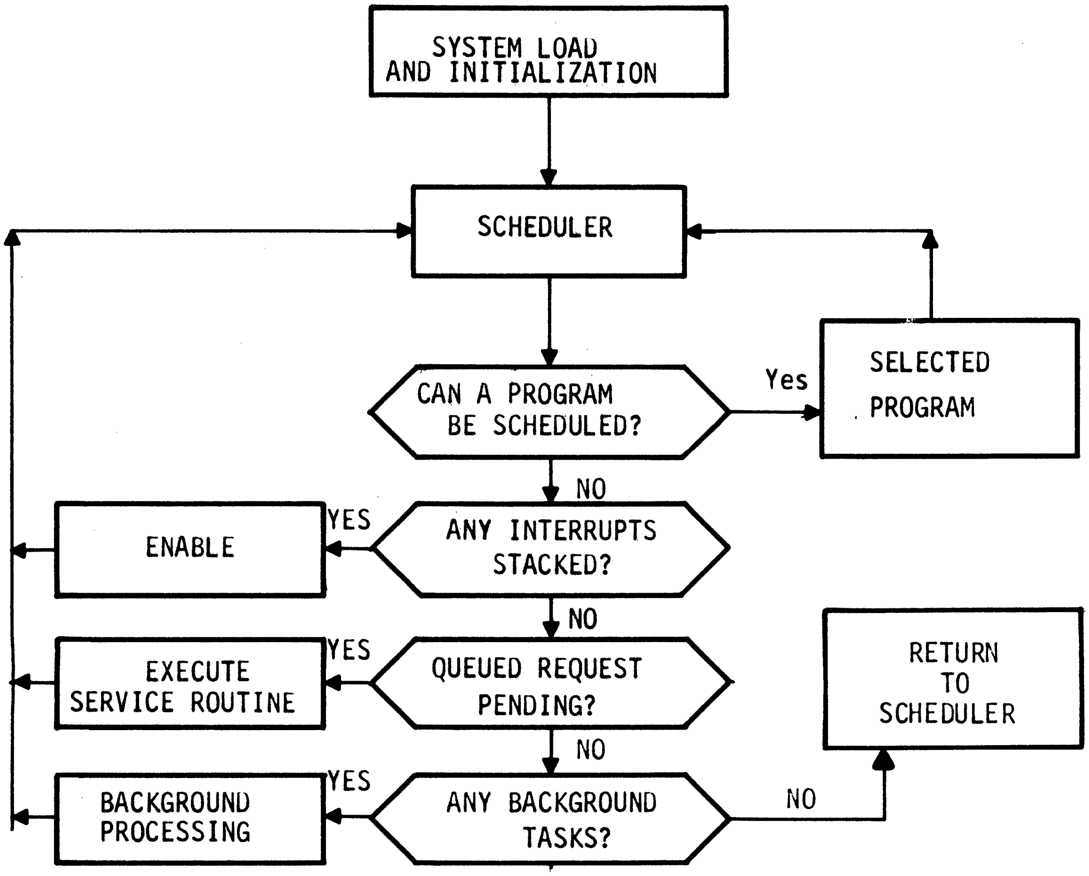

Figure 13 Program Scheduling

If there are no other programs to be executed, the scheduler executes the background tasks; that is, programs which do not require a rapid response time and can be executed whenever there is time available in a major cycle. An example of a background task might be a display routine which may only require response within one second of the request.

If the background tasks are completed and there is time remaining in the major cycle, the scheduler could place the computer in an idle status to conserve power.

Fundamental to the operation of the OBES scheduler is the recognition of program state. The state is kept in the program control table and is continually updated by the scheduling routine. Classification of states may be as follows:

- A selected program is the one currently in operation.
- An active program is in memory and currently a candidate for selection by the scheduling routine.
- An accommodated program is one which has performed its proper number of executions in the present major cycle.
- A program in the wait stage is one which is held up in its normal execution, waiting for an event to occur so that it may continue. For example, it could be waiting for an I/O operation to obtain data necessary for the next series of computations.

An inactive program is still in memory but not a candidate for selection by the scheduler. It has completed its function during this part of the mission and is no longer required. The area in memory which this program occupies could be used to load a new program.

When a program is in the initial delay status, it means that it has been requested from auxiliary storage, but is not yet in main storage. As soon as it is read in it would become an active program.

Dormant programs are those which are not in memory, but reside on the auxiliary memory.

During the Phase II study, scheduling algorithms will be prepared and tested. Details of the proposed tasks and documentation are given in Section 3.2.

Some of the problems which must be solved in the development of a scheduler algorithm are described below:

- A means must be provided to detect system overload, i.e., demands by programs for more storage or CPU time than is available, or for I/O data rates exceeding I/O channel capacity. Such overloads would not normally occur - they indicate that the computer system was initially undersized. However in the FGSS system, storage and speed requirements cannot be precisely determined in advance because of on-board planning and replanning. Thus, the scheduler must (a) detect overloads if they occur, (b) continue processing highest priority tasks when an overload occurs, and (c) provide an overload warning message to crew members or ground control when necessary.
- The scheduler must not only insure that the duty cycles and repetition rate requirements are met, but must also insure that the program entries are properly distributed over a major cycle. In some cases, entry into a program must be at a precise time and, still more demanding, it will be necessary to insure that a particular point (other than the normal entry point) is entered at a precise time.
- A criterion must be established for selecting the program to be executed during a given time slot. One alternative is to select programs on the basis of repetition rate, priority, and service class prespecified in a control table associated with each program. A second alternative is to allow each program to specify (to the scheduler) the precise time and entry point for its next entry. In the second method, the time/entry point specification would be treated as a request for CPU time from the scheduler, and the scheduler would honor all such requests if possible. When such requests conflict (e.g., 2 programs request entry at precisely the same time), the conflict would be resolved by the scheduler on the basis of priority

2.2.5.2 Interrupt Handlers

The cyclic process of scheduling is frequently suspended to service system interrupts. When this happens, an interrupt supervisor routine is initiated to determine what caused the interrupt, and what processor should be initiated. In some cases the interrupt is handled and processed immediately. In cases of low priority interrupts, the request may be processed at a later time when there are no higher priority programs to be executed.

Figure 14 is a functional diagram of the interrupt supervisor. In most cases, the process consists of saving the machine status, determining the services requested, executing the service routine, posting the completion, restoring the machine status, and returning to the point of interrupt.

When the required service routine is not available, the request is queued for later service. For example, if the interrupt is a request by an application program to perform an I/O operation and the I/O device or channel is busy, the requesting program would be put in the WAIT status.

Interrupt handlers will be designed and tested in Phase II. Details of the task are described in Section 3.2.

Interrupts originate in vehicle subsystems to signal a requirement for data, availability of data, subsystem status, emergency conditions, etc. When an interrupt is enabled, issuance of the signal changes instruction flow within the CPU by causing an automatic "transfer" to the fixed location in main storage associated with the interrupt type.

It is the function of the interrupt handler to save the machine status (primarily register contents) existing prior to the occurrence of the interrupt. This makes it possible for the interrupted program to regain proper control after the interrupt is processed. The interrupt handler must recognize the nature and urgency of the CPU processing commanded by the interrupt. This determined the entry point of the processing routine which must be executed. Upon completion of the processing routine, control is returned to the interrupt handler. The interrupt handler then restores the CPU to its state prior to the interruption.

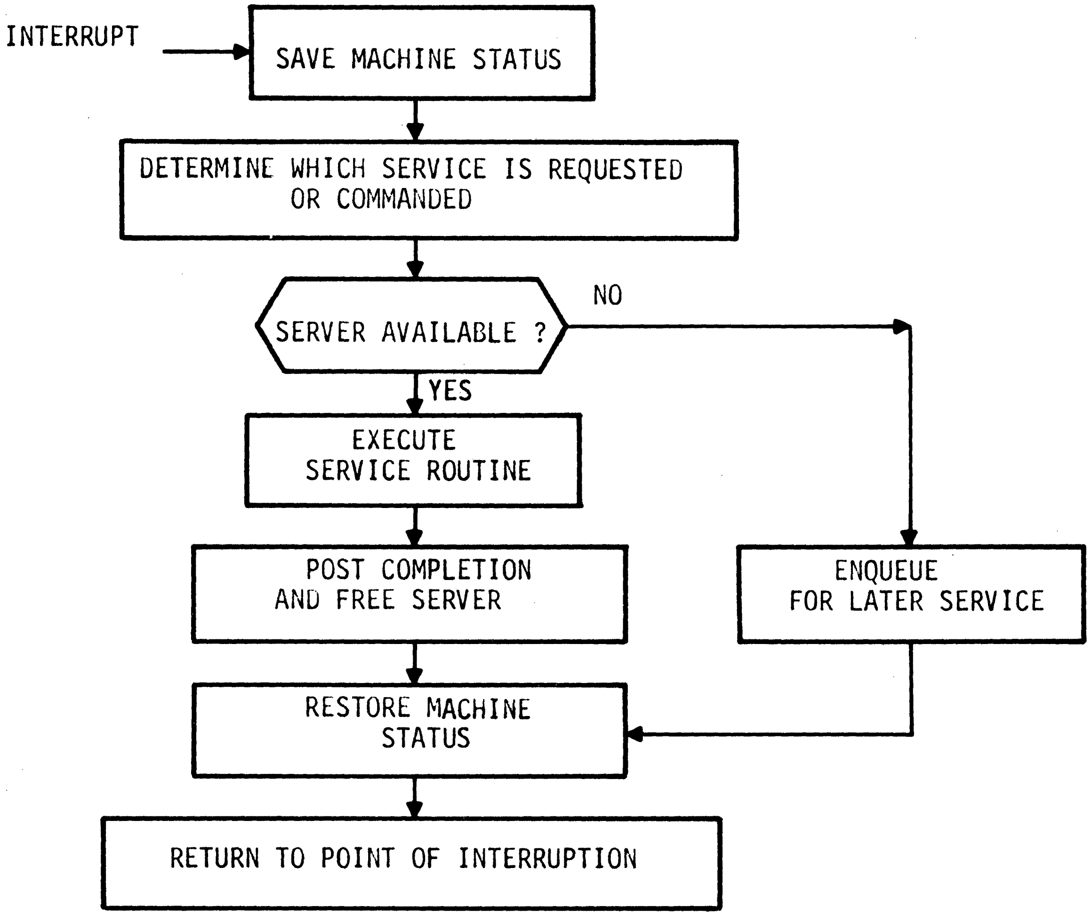

Figure 14 Interrupt Supervision

Interrupt handlers should be developed for timer, external, priority, and programmed interrupts since these are most commonly provided by advanced spaceborne computers. Unfortunately, interrupt handlers are generally machine dependent. The interrupt handlers will be developed for a specific spacecraft computer (IBM 4π-EP), and an effort will be made to provide meaningful specifications which apply to interrupt handlers for any computer.

In the employment of interrupt handlers, a problem arises when a second interrupt is received while a previous interrupt is being processed. If both interrupts are enabled, the second will cause an interruption in the processing of the first. If both interrupts require the services of the same processing routine or other resources - e.g., an I/O device, the conflict can only be resolved by assessing the priorities of each demand.

The above problem will be minimized by disabling all interrupts of lower priority than that which is currently being processed. Such disabled (stacked) interrupts can be later enabled and processed. The concept of priorities and their assessment may have to be applied to executive tasks and interrupt processors as well as to application programs.

2.3 SUPPORTING ANALYSES

To support the design of the key components of FGSS, analysis will be required of:

- Mission operations
- Man/Computer interface
- Guidance errors
- Space Programming Language

Due to the limited funds available for Phase II, these analyses will be limited to those efforts that are necessary to fulfill the basic objectives of key component development.

2.3.1 OPERATIONAL REQUIREMENTS

In addition to the requirements for minimizing reaction time and providing near optimal performance, the planning and guidance programs must meet the operational requirements inherent in each space flight. These requirements may take the form of restraints on trajectory, ground tracking or ground control needs, hardware limitations, or safety considerations. To avoid compromising the multi-mission capability of the mission planner and active guidance algorithms, means must be found for implementing those operational requirements that recur, in some form, on each flight.

Some missions may have unique measurements. Whenever possible, the basic planning and guidance algorithms should be designed to accommodate the special requirements by adding small, special purpose modules instead of resorting to complete, special purpose planning and guidance algorithms.

To provide the detailed design baseline for operational requirements, each mission and system of hardware, including booster and spacecraft, should be analyzed. Typical requirements to be categorized are:

- Ground tracking range and angle limits
- Location of telemetry and ground control stations
- Abort and back-up modes
- Booster and spacecraft aerodynamic load limits and aerodynamic heating limits
- Navigation system restraints
- Launch azimuth limits
- Allowable impact limits of booster
- Payload orientation limits

As a special form of operational requirement, the spaceborne software programs must be sized to fit the spacecraft computer, Therefore, it will be necessary to establish the spacecraft data processing hardware expected to be operational in the 1970-1975 time period.

2.3.2 MAN/MACHINE INTERFACE

The Flexible Guidance Software System will provide an opportunity for improved operational flexibility, via manned participation in quick-reaction alterations of the mission plan, on the pad before launch or during the actual mission. In order to exploit this opportunity, it is necessary to develop effective techniques for man/computer cooperation in rapid selection, modification, and verification of mission plans. This means effective cooperation between crew and computer in space, and similar cooperation between ground station personnel and their computer/software system.

The general problem of man's role in the operational use of a Flexible Guidance Software System was addressed in Phase I with emphasis on the onboard aspects. Crew functions were defined in the areas of mission planning system checkout and monitoring, malfunction detection and corrective action, and tasks in backup modes to be used in the event of certain equipment failures. Preliminary estimates were made of the display and control facilities needed for effective fulfillment of these functions.

In Phase II, it is proposed to concentrate on what is probably the most critical and novel problem, namely, the problem of crew/computer cooperation in mission planning and verification. This will be explored in more depth than in Phase I. Emphasis will be on procedures rather than on hardware requirements. Typical problems to be considered include the number of alternatives that should be presented for human decision, the degree to which some decisions should be automatic, the information needed for decision between alternate plans, and the way in which human wishes are to be communicated to the computer.

2.3.3 GUIDANCE ERRORS

To reduce computational requirements, certain approximations will generally be made in guidance equations and in navigation equations. In addition certain second order effects may be neglected that affect guidance or navigation accuracy. For example, during orbital flight at low altitudes, atmospheric drag affects the motion of the vehicle. If navigation is to be done with on-board equipment, a low-g accelerometer is required for precise, long-term navigation accuracy. In most cases a drag model is used to predict the effect of the atmosphere. This drag model will generally contain approximate relations and, as such, represent a source of error.

The design of error models (such as a drag model or a gyro drift model) is dependent on the specific application or specific hardware and is outside the scope of the proposed Phase II program. Also, due to limited funds, the accuracy of the guidance system will not be evaluated over the complete spectrum of missions.

The key questions affecting the design of FGSS are:

- What is the effect of the residual error sources on mission planning?
- Is it necessary to include error models in the mission planning routines?
- How should known error sources or disturbance effects be treated in launch-pad verifications?

These and related questions will be answered by analysis of error effects in conjunction with the mission planner and guidance algorithm developments. Error effects will be analyzed with the help of INEAD and MM programs. These computer programs are described in Section 5.2.

2.3.4 SPACE PROGRAMMING LANGUAGE

It was determined during Phase I of the QRGT Study that higher order programming languages should be used to realize the full potential of the FGSS software system. Additional benefits are gained by use of a common HOL.

A common HOL would make it possible to generate and maintain a single, machine-independent MML for all spaceborne computers. It would also reduce coding effort and thus reduce turnaround time. A common language would increase flexibility in the assignment of programmers. More programmers would be qualified for flight programming eliminating, to a great extent, the high degree of specialization which has been characteristic of flight programming.

A common HOL would improve communication between mission analysts and programmers, and among programmers. It would result in less distinction between programs written for mission analysis and spacecraft application. A common language would also reduce the number of compilers required to service the spacecraft computers.

There are many important considerations in the specification and implementation of a common programming language. In the area of language specification, it is important to select features and specify syntax which represent the lease complexity compatible with the facility required for flight programming. The language must be ease to use if it is to significantly increase programmer productivity; it must be easy to read if it is to serve as an effective communication tool. It must avoid or standardize features and syntax which are highly machine dependent. In the area of implementation, the translator must produce highly efficient object code. It must interface simply and effectively with other components of the FGSS ground operating system. The translator must produce machine code modules whose structure is compatible with the structure required by the FGSS onboard executive system under whose control they will be executed.

System Development Corporation is developing a HOL for SAMSO called Space Programming Language (SPL). To insure compatibility of this development and FGSS, IBM will analyze the use of SPL for software preparation, validation and maintenance and identify any special requirements imposed by FGSS.

Section 3

PROGRAM PLAN

3.1 SCOPE OF WORK

The proposed program is outlined below. Numbers in parenthesis correspond to paragraph numbers in the Statement of Work to RFP No. F04701-68-R-0113.

Task I (Integrated Guidance System)

- Simulate and prepare design specifications for Trajectory Generation.
- Implement vehicle, range safety, and navigation system hardware constraints in mission planning and active guidance.
- Simulate and prepare design specifications for Trajectory Optimization.
- Simulate and prepare design specifications for Atmospheric Ascent Guidance and Exo-Atmospheric Guidance.

Task 2 (Software System)

- Prepare design specifications and test scheduling algorithm for the On-Board Executive System (OBES).
- Develop and demonstrate interrupt supervision routines for the OBES.
- Develop and demonstrate an experimental Configurator to assembly Specific Vehicle Libraries (SVL) from the Multi-Mission Library (MML) of the Ground Operating System (GOS).
- Examine compatibility of the proposed Space Programming Language (SPL) being developed by SDC and the advanced FGSS concepts.

Task 3 (Man/Computer Interface)

- Determine the mission control tasks during the pre-launch mission planning phase.
- Determine the manner in which these tasks are to be implemented.

Task 4 (Error Analysis)

- Determine the error sources inherent in the functional elements of the guidance system.
- Determine the effect of residual errors on mission planning.
- Provide, if necessary, the capability for including error analysis in mission planning.
- Establish techniques for assessing the accuracy of system performance as part of the launch-pad verification process.

The mission planner, guidance algorithms, and the support software will be designed to meet the requirements of the Statement of Work as discussed in Section 2.0 of this proposal.

The activities planned to fulfill the requirements of each task are outlined below in Section 3.2. Documentation to be delivered on the program is described in Section 3.3.

The schedule for the program and the proposed distribution of manpower are given in Figure 15 and Table 1 respectively.

3.2 TASK DESCRIPTIONS

The program activities are organized in a task structure corresponding to Section 5.0 of the Statement of Work to the RFP. The objectives of each task will be accomplished by coordinated activities on specific subtasks as described below.

Task 1 Integrated System Definition (5.1)

Task 1.1 Mission Planner (5.1.1)

The Mission Planner developed during Phase I and implemented as TGP (Trajectory Generator Program) and TOP (Trajectory Optimizer Program) will form the basis for the Phase II work. Additions and refinements to be included involve three major activities: trajectory generation, constraints, and trajectory optimization. These additions and refinements will be closely coordinated with related changes in the guidance algorithms, Task 1.2.

TASK	MONTH								
	1	2	3	4	5	6	7	8	9
1. INTEGRATED SYSTEM DEFINITION									
1.1 MISSION PLANNING	■	■	■	■	■	■	■	■	
1.2 GUIDANCE	■	■	■	■	■	■	■	■	
1.3 OPERATIONAL REQUIREMENTS		■	■	■	■	■	■		
2. SOFTWARE SYSTEM									
2.1 OBES SCHEDULER	■	■	■	■	■	■	■	■	
2.2 OBES INTERRUPT HANDLER	■	■	■	■	■	■	■	■	
2.3 SVL CONFIGURATOR	■	■	■	■	■	■	■	■	
2.4 SPL REVIEW		■	■	■	■	■			
3. MAN/COMPUTER INTERFACE				■	■	■			
4. ERROR ANALYSIS		■	■	■	■	■	■	■	

Figure 15 FGSS Task Completion Schedule

TASK	MONTH									TOTAL
	1	2	3	4	5	6	7	8	9	MAN MONTHS
(1) INTEGRATED SYSTEMS DEFINITION	2.5	3.0	3.0	4.1	4.1	3.4	3.0	3.7		26.8
(2) SOFTWARE SYSTEM	1.5	3.0	3.0	3.0	3.0	3.0	3.0	3.5		23.0
(3) MAN/COMPUTER INTERFACE				0.5	0.5	0.5	0.5			2.0
(4) ERROR ANALYSIS		0.5	0.5	0.5	0.5	1.0	1.0	1.0		5.0
PROGRAM MANAGEMENT & DOCUMENTATION	1.3	1.0	0.8	0.8	0.6	0.6	0.9	1.0	0.5	7.5
TOTAL	5.3	7.5	7.3	8.9	8.7	8.5	8.4	9.2	0.5	64.3

TABLE 1 PROPOSED MANPOWER DISTRIBUTION

Task 1.1.1 Trajectory Generation (5.1.1.1)

In order to increase the capabilities of the Mission Planner, the present orbit-insertion mode will be supplemented with additional orbit-insertion options available in OP-EX. Only minor changes will be necessary in the logic of the Trajectory Generator Program, but the new modes of planning must be tested experimentally.

Improved analytic approximations for predicting position and velocity at the end of the atmospheric phase of ascent will be developed. Several zero-lift ascent trajectories will be computed and fitted with linear or quadratic functions of kick angle. The resulting formulae will be tested on several ascent trajectories and the predicted burning times will be computed with the actual burning times.

Consideration will also be given to the use of simple explicit or quasi-explicit formulae for zero-lift ascent trajectories. If promising equations can be developed, this approach will be adopted in place of the empirical approximations proposed above.

The trajectory routines used in mission planning will be modified to include corrections for the secular effects of gravity harmonics. Simple formulae for the corrections are available. Also, a routine will be developed for computing offsets to compensate for the effects of the quadrapole field. This routine will use numerical integration, or the closed formulae mentioned in Section 2.2.2.

Task 1.1.2 Constraints (5.1.1.2)

The work done during Phase I on mission and vehicle constraints will be extended. Present constraints and probably future constraints from Task 1.3 will be identified and classified so that implementation techniques can be developed. Relevant constraints will be implemented either in the planner (see Task 1.1.3) or in the guidance algorithm (Task 1.2.1) or by other techniques (e.g., precomputation of parameter bounds for acceptable atmospheric ascent trajectories).

Task 1.1.3 Trajectory Optimization (5.1.1.3)

The Trajectory Optimization Program (TOP) developed during Phase I will be updated to include the new parameters and constraint relations resulting from the activity in Tasks 1.1.1 and 1.1.2. For example, the kick angle parameter of atmospheric ascent will be added to the list of independent variables used in optimization. TOP will be modified to improve its ability to enforce constraints of varied types.

Constraints expressable as argument bounds will be directly enforced by the search logic (this is a feature of the present TOP). Other inequality constraints, or equality constraints which need not be satisfied with high accuracy, will be enforced by penalty functions. Equality constraints requiring precision will be enforced by designating some variables as dependent variables to be iteratively varied (by a technique similar to Newton's method) to satisfy the constraints.

The operating efficiency of TOP will be improved by certain changes in its logic, which eliminate redundant computations. Also, one or two new search algorithms will be tested if suitable candidates (developed elsewhere) can be found.

Task 1.2 Guidance Algorithms (5.1.2)

Guidance algorithm development in Phase II will be concerned with increasing the capability of the generalized guidance law (OP-EX) developed in Phase I, and development of an algorithm for nearly-optimal atmospheric ascent, which satisfied structural and thermal constraints.

Task 1.2.1 OP-EX (5.1.2.1)

The flexibility of OP-EX will be increased by selecting and incorporating additional options for the form of the conditions to be satisfied at cutoff. All options will be available for use by the mission planner. Also, OP-EX will be interfaced with the atmospheric ascent routine (see Task 1.2.2) to permit coordinated planning and execution of an entire ascent targeting.

Provisions will be made for correcting the guidance to compensate for the effect of the Earth's quadrapole field on the coast trajectories of the spacecraft and the target vehicle (for rendezvous). These corrections will be computed during active guidance using closed formulae, or just before active guidance (offset concept) using closed formulae or numerical integration.

Modifications will be added to OP-EX to permit enforcement of constraints such as bounded attitude rate, and a specified attitude at cutoff. The forms of these modifications will be suggested by optimization theory, but will be only quasi-optimal; strict optimality will be satisfied in favor of simplicity and practicality.

Task 1.2.2 Atmospheric Ascent Guidance (5.1.2.2)

The atmospheric ascent algorithm employed in Phase I involved a vertical rise, a kick-over maneuver which was vehicle dependent but mission independent, and a gravity turn (zero lift) trajectory until the dynamic pressure had fallen below a set value (approximately 30 psf). For planning purposes the atmospheric phase was generated analytically with sperical trigonometry relations and empirical coefficients determined by previous simulations of the above guidance algorithm.

In Phase II, a flexible, near-optimal guidance algorithm will be developed for atmospheric ascent, and coordinated with mission planning and the exo-atmospheric guidance. A complete ascent trajectory will consist of three phases: an initial phase prior to the buildup of large dynamic pressures, a second phase (high-Q regime) dominated by aerodynamic constraints, and an exo-atmospheric phase that begins when dynamic pressure becomes small again. During the initial phase, the vehicle follows a tilt program governed by parameters (e.g., launch azimuth and kick angle) chosen by the mission planner as part of the mission optimization. This phase lasts until dynamic pressure becomes appreciable. For the second phase, two guidance options are proposed: an explicit gravity turn (zero-lift) trajectory or a non-gravity turn with explicitly enforced bounds on the magnitude of $Q\alpha$, the product of dynamic pressure and angle of attack. (Note: Q and α will be computed from the vehicle's position, velocity, and attitude, rather than measured by sensors.) After dynamic pressure has passed its peak, the steering commands for exo-atmospheric guidance (OP-EX) begin to be computed, but not used. Instead they are compared with the $Q\alpha$ limits. As soon as Q drops below a set value, and the exo-atmospheric steering becomes compatible with the $Q\alpha$ limits, the atmospheric phase ends and OP-EX takes over.

Task 1.3 Operational Requirements (5.1.1.2)

The Air Force missions of interest will be analyzed to establish the constraints that must be satisfied by the mission planning and guidance functions. This data will be provided to support the development of the mission planning and guidance algorithms in Tasks 1.1 and 1.2 respectively.

Analysis will include:

- Tracking and telemetry requirements
- Aerodynamic loads and heating limits
- Back up and abort modes
- Range safety
- Typical constraints of navigation systems
- Special mission functions

Assistance will be required from SAMSO in the collection of mission operational details, special mission functions, and hardware limits.

Task 2 Computer Software System

This task is concerned with the development of key components of the support software in FGSS namely the on-board executive system and the SVL configuration. Development of OBES will be limited to the Scheduler and Interrupt Handler.

Analysis will also be made of the proposed Space Programming Language (SPL).

Task 2.1 Scheduler Algorithm (5.3.1)

The purpose of this subtask is to design a scheduler algorithm which will process on-board application programs of the flight program on the CPU according to their repetition rate and priority requirements. This effort includes:

- The programming of at lease one scheduler alternative.
- Testing of the scheduler with profiles of typical spaceborne software.

- Iterative modification and improvement of the algorithm.
- The preparation of specifications for the scheduler algorithm.

Scheduler flow charts will be provided at two levels: (1) without reflecting machine dependence, and (2) charts for implementation using an IBM 4 Pi (EP) spaceborne computer.

Task 2.2 Interrupt Supervision (5.3.2)

The purpose of this subtask is to develop and demonstrate interrupt supervision routines.

Development of the interrupt supervision routines includes test and usage of the routines: (1) operating individually and independent of other executive routines and (2) operating in conjunction with the scheduling algorithm to be developed in Task 2.1.

For the purpose of this subtask, it is assumed that the computer hardware functions performed on occurrence of an interrupt are similar to those performed by the IBM 4 Pi (EP) spacecraft computer. Interrupt types to be considered are:

- Timer interrupts which occur when a specified time interval has elapsed.
- External interrupts resulting from manual activation of a control at the crew member console.
- Priority interrupts resulting automatically from an alarm or attention signal within a vehicle subsystem.
- Programmed interrupts resulting from a programmed request for services of the executive control program.

Task 2.3 SVL Configurator (5.3.3)

The purpose of this subtask is to develop and demonstrate experimental version of the configurator required to generate specific vehicle libraries from the multimission library. This effort includes the determination of an MML directory format suitable for indexing FGSS flight programs and the preparation of a skeletal MML consisting of the directory and profiles of typical spaceborne software modules.

The development of the experimental configurator includes its test and usage to perform the following tasks:

- Accept a user's specification of the vehicle to be configured.
- Refer to the MML directory and extract all modules (in either source or object form) required to service the specified vehicle.
- If in source form, the compilation (FORTRAN or PL/1) of the modules extracted.
- The resolution of all external references among the modules extracted.
- Combination of the modules extracted to form higher level modules.

Using module profiles, the programs formed will not be executable, but it will be shown by means of maps and cross reference lists that all required modules are extracted and properly combined.

Task 2.4 Space Programming Language (5.3.4)

The programming language being developed by System Development Corporation for SAMSO will be analyzed to insure compatibility of SPL and FGSS development. Potential problems will be identified together with recommended corrective action. Special requirements imposed by FGSS will be delineated.

Task 3 Man/Computer Interface (5.3)

In conjunction with the development of the mission planner in Task 1.1, the role of ground personnel will be studied and specific tasks established for mission specification, alternate plan generation, trajectory selection, and program verification. The functional requirements for ground displays and controls will also be established including range safety requirements.

Task 4 Error Analysis (5.4)

The effect of guidance and navigation errors will be studied to determine their impact on mission planning and on launch-pad verification. Typical error sources will be identified and their effect propagated on alternate flight plans for the same mission to establish the sensitivity of system performance to flight plan selection.

If the sensitivity is large for certain error sources, appropriate error models will be constructed for use during prelaunch mission planning.

An approach will be established for launch-pad verification of the selected flight plan. The need for error models to simulate system performance will be identified and typical error models will be formulated.

3.3 DOCUMENTATION

Reports and other data will be prepared and submitted in accordance with the Contract Data Requirements List.

3.3.1 PROGRAM PLANNING REPORT

A Program Plan will be submitted within thirty (30) days of contract go-ahead describing, in detail, the work to be performed to fulfill all requirements of the contract. Section 3.0 of this proposal will be used as a starting point for the Program Planning Report.

The report will contain, as a minimum:

- Definition of all end objectives and major milestones.
- A schedule for the accomplishment of all milestones and tasks.
- Identification and description of all tasks and subtasks necessary to accomplish the objectives and requirements of the contract work statement. Each task and subtask will be directly correlated with and referenced to a task in the contract statement of work.
- A list, by task, of significant interface decisions and the data required for the decision, together with need dates.
- A Task Descriptive Network, in the form of a PERT chart, identifying the activities, and their interdependencies, leading to the attainment of major milestones and end onjectives. The chart shall also show time estimates for each activity and the critical path.

The Program Plan may be modified with the approval, or at the direction of, the Air Force. The planning document will be maintained in a current status and updated versions of the Task Descriptive Network will be included in the Monthly Progress Reports.

3.3.2 MINUTES OF MEETINGS

IBM will prepare and submit minutes of working group meetings, technical coordination meetings, and formal Technical Direction conferences. These minutes will contain, as a minimum, attendance lists, agreements, action items, and discussion of the proceedings as necessary to detail the results of the meetings. These minutes will be mailed to Space Guidance Branch (SMTAG), SAMSO within five (5) working days for distribution to all participating agencies.

3.3.3 MONTHLY PROGRESS REPORTS

IBM will prepare and submit Progress Reports each month detailing status of the effort relative to the Program Plan. In addition to progress to date, this report will include descriptions of milestones accomplished, technical and managerial problems encountered, and changes in future plans, if any.

To minimize the cost of technical reporting, informal engineering reports (internal IBM documents) prepared or utilized on the program will be submitted to SAMSO. The Monthly Report will make reference to these informal reports as necessary to describe status and technical accomplishments.

The Monthly Report will also contain an updated version of the Task Descriptive Network and names of key personnel charging to the project.

Progress Reports will be submitted on or before the tenth (10) of each month following the reporting period.

3.3.4 DETAILED DESIGN SPECIFICATIONS

Design specifications will be prepared and submitted for:

- Mission Planner (Trajectory Generation and Optimization)
- Guidance Algorithm (Ascent and Exo-Atmospheric Guidance)

- SVL Configurator
- OBES Scheduler and Interrupt Handler

The specifications will include, as a minimum, equations, logic requirements, flow diagrams illustrating the processing of the equations, and other data as necessary to fully define the design configuration so that a program may be prepared for processing on a digital computer. To minimize the cost of documentation, these specifications will be prepared as informal engineering reports utilizing Exhibit XX to AFSCM 375-5 as a guide.

All specifications will be submitted within 240 days of contract go-ahead.

3.2.5 FINAL TECHNICAL REPORT

A Final Report will be prepared and submitted by IBM presenting all significant technical results. This report will be prepared in accordance with AFSCR 80-20 and 80-20A and distributed in accordance with the Contract Data Requirements List. One (1) reproducible copy will be submitted to SMTAG.

Draft copies of the report will be submitted to SMTAG for comments and revision requirements no later than 230 days after contract go-ahead. Final distribution will be made 270 days after contract go-ahead.

3.4 PROGRAM SCHEDULE

The proposed schedule is shown in Figure 15. This schedule will be amplified as necessary and made a part of the Program Planning Report.

The data delivery schedule is shown in Figure 16.

3.5 MANPOWER DISTRIBUTION

Effort on the proposed tasks has been distributed as shown in Table 1. This distribution may be varied as the needs of the program change or it may be adjusted during preparation of the Program Plan to reflect emphasis desired by SAMSO on particular tasks. Details of the distribution by labor grades are provided in the Cost Proposal.

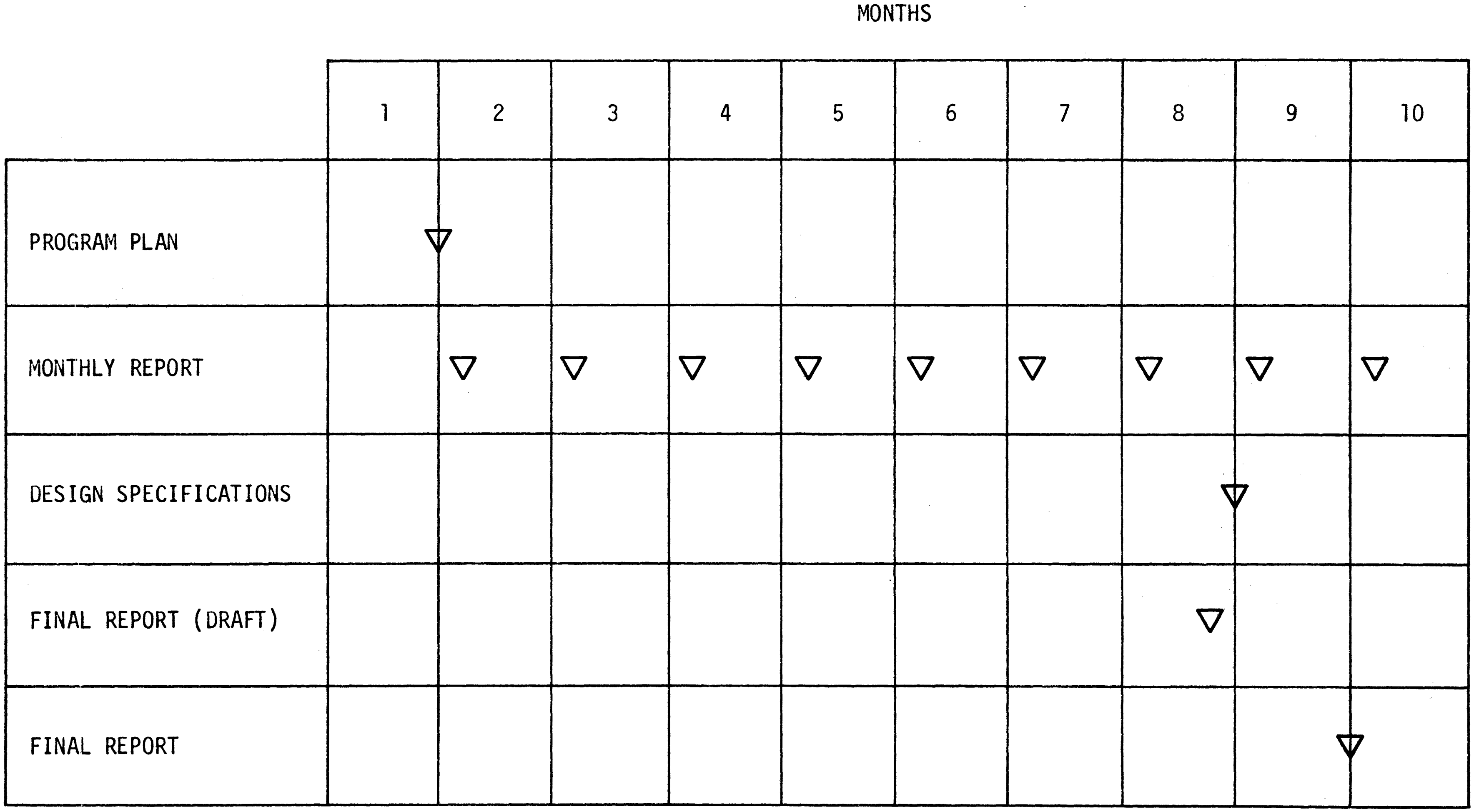

Figure 16 FGSS Data Delivery Schedule

Section 4

MANAGEMENT

4.1 ORGANIZATION

The FGSS study will be managed by Dr. Howard M. Robbins of the Military Space Engineering organization, Space Systems Center, Endicott, New York. Dr. Robbins was the manager of the Phase I study.

A special project team will be formed under Dr. Robbins to direct and supervise the work on the various tasks. Dr. Thomas Clancy, Mr. Joseph Constable, Mr. Fred Neumeyer, and Dr. Frank Schlee will lead tasks 1 through 4 respectively.

The study organization is shown in Figure 17. Resumes of key personnel are presented in Section 4.4 of this proposal.

4.2 STUDY MANAGEMENT

Project meetings will be held weekly with the Task Leaders to review task progress versus plans and schedules, identify problems and action items, and provide technical direction and coordination. Dr. Robbins will be assisted in these project meetings by Mr. Orrange's Technical Staff and by personnel from the Program Control office. Technical sessions will be held on a task level as necessary to provide guidance, resolve problems, and reformulate plans.

Program review meetings will be held monthly by Mr. Orrange to assess status versus plans and review technical approach.

Specialists in guidance and support software will be used as consultants and will be asked to review critical technical results and maintain technical liaison with similar efforts on other programs within IBM.

Labor and cost control data will be provided by IBM's MPACS reports (Management Planning and Control System). These reports provide weekly data on man

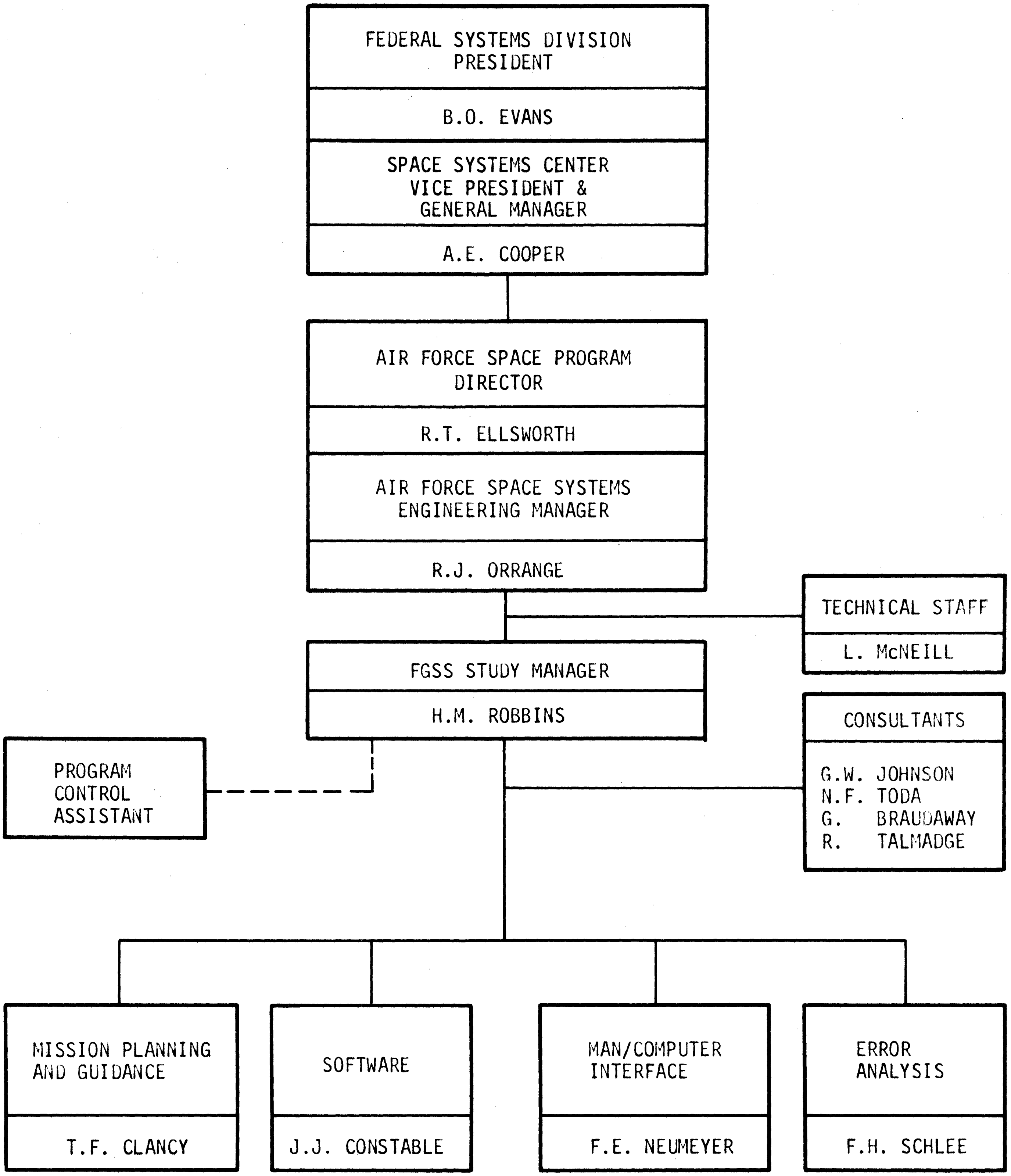

Figure 17 FGSS Study Organization

hours expended, travel costs, etc., for each work package in the program. An account code structure will be set up corresponding to task breakdown. Expenditures will be compared to budgets and work accomplished at the weekly Project Meetings and the monthly Program Review Meetings.

All contract documentation will be submitted by the Program Control office. Mr. F. Neumeyer will also act as special assistant to Dr. Robbins and will coordinate preparation of the Program Plan and technical reports and assist in technical liaison with SAMSO and program administration.

4.3 TECHNICAL LIAISON

In addition to progress reporting and telephoning communications, technical liaison will be maintained with SAMSO and Aerospace Corporation by working group conferences, technical coordination meetings, and formal Technical Direction Conferences.

Working group conferences and technical coordination meetings will be scheduled by IBM (or SAMSO or Aerospace Corporation) as necessary to resolve problems and review technical details.

T.D. Conferences are proposed for the third, fifth, and seventh months of the program. At these conferences, IBM will present task objectives, technical approach, and program progress. Approval and/or technical direction will be sought on major technical matters. Proposed changes in the Program Plan will also be presented. Copies of briefing material (charts as well as text) used by IBM at these T.D. Conferences will be distributed during the meeting.

A final presentation will be made by IBM at Headquarters, SAMSO, to report the significant results obtained under the contract. Charts and/or slides used for this presentation, as well as copies of prepared text, will be made available for Air Force Use. The presentation will be scheduled at a date mutually acceptable to SAMSO and IBM but not later than the last day of the contract.

The SPARS program in IBM is also being managed by personnel in the McLean Building. Therefore, an opportunity exists to coordinate technical liaison on SPARS and FGSS.

4.4 RESUMES OF KEY PERSONNEL

DR. HOWARD M. ROBBINS - Senior Engineer, Manager

FGSS Study Assignment - Study Manager

Basis for Selection - Dr. Robbins acted as study manager for Phase I (QRGT), and as task leader for Task I (ascent, orbit, rendezvous, reentry). He has expert knowledge of the mathematical theory of space guidance and navigation, and familiarity with all aspects of the FGSS problem.

Experience - At IBM, he defined the computer functions and appropriate equations for modification of the Titan II guidance computer to Saturn boost vehicle guidance. He directed a four-month analysis and simulation study for ABMA in connection with the guidance of an advanced IRBM, directed the Optimal Guidance Transfer Study, and the guidance equations part of the IBM SSGS Study, and has participated in numerous other studies.

Prior to joining IBM in 1960, he was employed for ten years with the Hughes Aircraft Company where he worked in game theory, air defense analysis, digital computer logical design, and computer application studies. He became a Senior Staff Physicist and head of a logical design group, with responsibilities for systems analysis to determine computer requirements of military systems, and synthesis of computer designs to match the requirements.

Dr. Robbins is a member of Phi Beta Kappa and the Americal Physical Society. He has several patents in the electronics field.

Relevant Publications - "Optimal Steering for Required - Velocity Guidance" Navigation 2, p 355-363 (1965); "An Analytical Study of the Impulsive Approximation" AIAA Journal 4, p 1417-1423 (1966); "Optimality of Intermediate - Thrust Arcs of Rocket Trajectories" AIAA Journal 3, p 1094-1098 (1965); "Optimal Rocket Trajectories with Intermediate - Thrust Arcs" Proc. 17th Congress IAF to appear; "A Generalized Legendre - Clebsch Condition for the Singular Cases of Optimal Control" IBM Journal of R&D 11, p 361-372 (1967).

Education - Dr. Robbins received a BS in Physics, in 1947 and an MA in Physics in 1948, both at the University of Minnesota; Ph. D in Physics (quantum field theory), 1952, at California Institute of Technology.

DR. THOMAS F. CLANCY - Advisory Engineer

FGSS Study Assignment - Task Leader - Mission Planning and Guidance

Basis for Selection - Dr. Clancy was responsible for mission planning and development of the Trajectory Simulator and Trajectory Optimizer Programs (TGP & TOP) during Phase I. He also performed an investigation of range safety requirements as related to mission planning.

Experience - Dr. Clancy joined IBM in 1962 and since that time has performed guidance and trajectory analysis for Saturn and Advanced Saturn, participated in the study of Optimal Guidance for Orbit Transfer, provided ascent guidance, analysis and equations for the Standardized Space Guidance System Study (SSGS), investigated earth satellite attitude problems in the MOSS study, and provided systems analysis support for in-house studies of SRAM, LFSW, and LEM Backup Attitude Reference System.

Dr. Clancy has also investigated explicit guidance schemes for multi-stage ascent vehicles and advanced targeting concepts for ICBM's. He has recently support the Apollo Backup Computer Study in the areas of computer equation definitions and guidance and control equation simulations.

Relevant Publications - "Explicit Guidance Scheme for Multistage Booster" February 1965, IBM 65-512-03.

Education - BSME - University of Pennsylvania, 1958; MSME - California Institute of Technology, 1959; Ph. D - (Applied Mechanics) Cornell University, 1962; Thesis - "Effects of Radiation Forces on the Attitude of an Artificial Earth Satellite" AIAA Journal, Vol. 2, No. 3, March 1964.

J. J. CONSTABLE - Advisory Programmer

FGSS Study Assignment - Task Leader - Software

Basis for Selection - Mr. Constable was responsible for the functional design of the Ground Operating System and the Onboard Executive System in the Phase I (QRGT) Study, and assisted in examination of validation requirements, methods, procedures, and facilities.

Experience - From 1952 to 1954 he was associated with the Fedders Corporation where he worked on design and test of refrigeration equipment while completing additional study in mathematics and mechanical engineering at the University of Buffalo. During 1956 and 1957 he was associated with the Beech Aircraft Corporation where he performed computer programming services for the structures and aerodynamics groups.

Since 1957 he has been associated with IBM and is a graduate of the IBM Systems Research Institute. Experience with IBM includes SAGE system test, test equipment engineering, and scientific programming. He has been primarily responsible for the development of computer programs for B-70 guidance simulation, Gemini reentry and rendezvous simulation, orbit computation, orbit transfer studies, and simulation of space reconnaissance/surveillance missions.

Most recently he has been assigned to lunar trajectory studies and to a MOL study effort for a computer/software system. His efforts in the lunar trajectory studies culminated with a modular software library suitable for (a) computing reference trajectories with choice of constraints at injection, perilune, and reentry, (b) computation of sensitivity matrices for the reference trajectories generated, (c) computation and simulation of midcourse velocity changes, (d) optimization and simulation of finite-thrust requirements for transferring from a parking orbit to a lunar trajectory, and (e) simulation and validation of proposed guidance schemes. His responsibilities in the MOL study effort included extensive analysis of mission computation requirements to determine the speed, storage, and I/O requirements imposed upon the onboard computer/software system.

Education - Mr. Constable obtained a BS from Temple University in 1952

F. E. NEUMEYER - Senior Associate Engineer

FGSS Study Assignment - Task Leader - Man/Computer Interface

Basis for Selection - Mr. Neumeyer has had extensive practical experience with man/machine interface problems in complex systems. He is familiar with navigation, guidance and control systems and assisted in the preparation of the FGSS program plan and proposal for Phase II.

Experience - Mr. Neumeyer has performed a photographic systems study for the Apollo Applications Program and has contributed to several studies and proposals in the area of optical instrumentation for photographic and navigation systems. He recently presented a paper in the AIAA Guidance and Control Conference (Huntsville 1967): "Drift Angle Acuity in Spacecraft Attitude Determination".

He was a military pilot, navigator, tactical air controller, and photo mapping superintendent in the U.S. Air Force for twenty-three years prior to joining IBM.

Education - Mr. Neumeyer received a BS (Military Science) in 1957 from the University of Maryland, and is presently engaged in graduate study (Engineering Administration) at Syracuse University.

DR. FRANK H. SCHLEE - Advisory Engineer

FGSS Study Assignment - Task Leader - Error Analysis

Basis for Selection - Dr. Schlee was responsible for the Error Analysis Task in Phase I of the Quick Reaction Guidance and Targeting (QRGT) Study. He has five years experience in missile and space field. The last three years have been spent at IBM in space guidance and navigation analysis

Experience - Prior to the QRGT assignment, Dr. Schlee participated in the design and analysis of software for the Gemini GT-10 autonavigation experiment. He participated in proposal studies for the navigation subsystem of the

Manned Orbital Laboratory. Prior to the MOL effort, Dr. Schlee performed mission analysis for the Standardized Space Guidance Study.

Dr. Schlee has published several reports on autonomous navigation and has presented seminars on this topic. He recently published three papers in the AIAA Guidance and Control Conference, Seattle 1966, and Huntsville 1967.

"Divergence in the Kalman Filter" with N. F. Toda

"Autonomous Orbital Navigation by Optical Tracking of Unknown Landmarks" with N. F. Toda and C. J. Standish

"The Region of Kalman Filter Convergence for Several Autonomous Navigation Modes" with N. F. Toda and P. Obsharsky

Education - After receiving a BSAE from Brooklyn Polytechnic, Dr. Schlee spent two years at Sperry Gyroscope. He then received his MS and Ph. D in Instrumentation Engineering from the University of Michigan.

NORMAN F. TODA - Senior Engineer

FGSS Study Assignment - Consultant to Dr. Robbins

Basis for Selection - Ten years experience in space guidance and navigation, significant accomplishments in the Phase I (QRGT) study, and participation in related studies.

Experience - Mr. Toda joined IBM Federal Systems Division in September 1964. He was assigned to the Image Velocity Sensor Subsystem (IVSS) Contract where he was responsible for analytical studies of image motion compensation.

Between 1965 and mid-1966 he was engaged in an in-house study of autonomous navigation. A computer program simulating the navigation process was developed. Optimal and suboptimal modifications of the Kalman filter were developed to yield good navigational performance in spite of computational errors induced by single precision (IBM 7090) word length and uncertainties in the gravity and drag models. The study also included an investigation of Preliminary Orbit Determination techniques suitable for use with onboard sensors.

The papers "Divergence in the Kalman Filter" by Schlee, F.H., Standish, C.J. and Toda, N.F., AIAA Journal, Vol. 5, No. 6, June 1967; and "Autonomous Orbital Navigation by Optical Tracking of Unknown Landmarks" by Toda, N.F. and Schlee, F.H. (to be published in Journal of Spacecraft and Rockets) were presented at the AIAA/JACC Guidance and Control Conference, August 1966. "Region of Kalman Filter Convergence for Several Autonomous Navigation Modes" by Toda, N.F., Schlee, F.H. and Obsharsky, P., was presented at the AIAA Guidance, Control and Flight Dynamics Conference, preprint No. 67-623, August 1967.

Equations for a deterministic navigation filter were developed for the Gemini Program. This effort included optimization of the observation schedule and developing a correction to account for the oblateness of the earth in star-horizon sextant measurements.

Mr. Toda was assigned to the QRGT Program between August 1966 and mid-1967. He was responsible for the synthesis of guidance routines for orbital maneuvers and rendezvous. Mr. Toda also developed an algorithm for determining two burn rendezvous maneuvers. These near optimal trajectories are employed to initialize the trajectory optimization program (TOP).

Between 1957 and 1964, Mr. Toda was employed at Sperry Gyroscope Company. His first assignment at Sperry was applied mechanics analysis in support of the B-58 Hustler Program and in the development of more advanced gyroscopes. He was then assigned to the development of ICBM and Space Vehicle Guidance techniques. He was responsible for the guidance equation development during the SSGS Contract.

He co-authored a set of notes for a two semester course in Trajectory Theory, Navigation, Guidance and Control of Space Vehicles. He has authored papers in Gyroscope Development, ICBM Guidance, trajectory perturbations induced by the oblate potential and gravity gradient attitude oscillations. He was awarded one patent.

Between 1950 and 1957 he was a full-time instructor in the Department of Engineering Mechanics, Cornell University.

Education - Mr. Toda received a BME and an MME from Cornell University in 1950 and 1953, respectively. He continued half-time graduate work through 1956 in Engineering Mechanics, Applied Mathematics, and Aerodynamics.

He was elected to Tau Beta Pi and Phi Kappa Phi.

He has taken in-house courses: Navigation Error Analysis, Optimization Techniques and Stability Theory. He has recently completed a course by Athans and Falb in Optimal Control.

Mr. Toda served two years in the U. S. Army. He is a member of AIAA.

MR. GORDON W. BRAUDAWAY - **Engineer** - Development Engineer

FGSS Study Assignment - Consultant to Dr. H.M. Robbins

Basis for Selection - In Phase I (QRGT) Mr. Braudaway developed the OP-EX (Optimal-Explicit) algorithm for guidance of exo-atmospheric flight, and participated in all parts of the guidance studies.

Experience - From 1958 to 1964, Mr. Braudaway was employed by the Boeing Company and did research and development of the Bomarc boost control system and a minimum complexity backup boost guidance technique for the X-20 (Dyna-Soar).

During this period he undertook research in the interaction between flexible vehicle structure and the flight control system and developed numerical techniques for the analysis of control systems for flexible vehicles. In addition, he was co-developer of an explicit multistage boost guidance technique which he later expanded to include the orbital rendezvous and intercept missions and was instrumental in the development of a general vehicle simulation program for the digital computer specifically suited to guidance analysis. He also developed a closed form solution of orbital motion over an oblate planet.

After joining IBM in 1964, he has continued research on explicit ascent guidance and was involved in the development of analytical programs for the Apollo Lunar Mission. He has also done simulation analysis of computational error propagation in numerical filters and navigation systems.

In addition to his analytical experience, Mr. Braudaway has extensive experience with both FORTRAN and machine language computer programming.

He is a member of Tau Beta Pi and Sigma Tau Engineering Fraternities and Sigma Pi Sigma Physics Fraternity.

Relevant Publications - "A Closed Form Solution for Satellite Orbits in the Gravitational Field of an Oblate Planet", October 1964, IBM #64-512-008.

Education - Mr. Braudaway received a BS in Engineering Physics in 1958 at the University of Colorado and an MS in Electrical Engineering in 1964 at the University of Washington.

DR. G. W. JOHNSON - Senior Engineer

FGSS Study Assignment - Consultant to Dr. H. M. Robbins

Basis for Selection - Recognized expert in the field of optimal control theory, and its application to spacecraft guidance and control.

Experience - Dr. Johnson joined IBM in 1956. At present he is a Senior Engineer with the Cambridge Space System Group. From 1962 to 1964, he served as the principal technical consultant to the engineering facility of the IBM Space Guidance Center in Huntsville, Alabama, where his studies were in the area of Liapunov stability for non-linear control systems and evaluation of optimal guidance systems using the maximum principle of Pontriagin. He also taught as part-time Associate Professor of Electrical Engineering at the Graduate Extension of the University of Alabama.

From 1950 to 1956, Dr. Johnson served as Instructor and Assistant Professor of Electrical Engineering at both of the above mentioned institutions, where as a member of the faculty of the Graduate School of the University of Connecticut, he taught a course in Feedback Control Theory.

He was a consultant to the N. W. Kellogg Company in 1952 and 1953 performing analytic and analog computer studies of rocket engine control dynamics.

From 1954 to 1956 he was a consultant to the Emerson Electrical Manufacturing Company performing analytic studies for airborne fire control systems. Dr. Johnson performed summer research from 1954 to 1956 for the IBM Airborne Computer Laboratory concerning analytic studies of digital bombing and navigation systems.

In 1956 he joined the IBM Electronics Systems Center in Owego, New York remaining there until 1962. He conducted research studies and served a principal investigator in the areas of ballistic missile flight control, hypersonic and orbital inertial guidance, control of nuclear power systems, craft oriented inertial navigation systems and design of digital accelerometers.

Dr. Johnson has served as a technical reviewer and researcher for books and technical journals in the field of automatic control theory and is a holder of a patent on a digital accelerometer.

Relevant Publications - Johnson, G.W., and Kilmer, F.G., "Summary of Analytical Studies concerning the Stability of the Path Adaptive Guidance Mode", January 1963; Johnson, G.W. and Brown, K.R., "Optimal Guidance for Orbital Transfer", August 1965; Johnson, G.W., "Pontriagin's Maximum Principle as a Theoretical Basis for Optimal Trajectory Determination", August 1963; Johnson, G.W. and Brown, K.R., "Real-Time Optimal Guidance", IEEE Transactions on Automatic Control, AC-12, #5, October 1967; Johnson, G.W., "Rapid Computation of Optimal Trajectories", IBM Journal of R&D, 11, p 373-382 (1967); Johnson, G.W. and Winfield, D., "On a Singular Control Problem in Optimal Rocket Guidance", AIAA Guidance, Control and Flight Dynamics Conference, Huntsville, Alabama, August 1967.

Education - BSEE, Rensselaer Polytechnic Institute; MSE, University of Connecticut, Ph. D, University of Connecticut.

Section 5

IBM QUALIFICATIONS

Personnel in IBM's Space Systems Center have been engaged in advanced development of space systems for 9 years. This work began with the Gemini and Centaur programs in 1958 and currently includes effort on Saturn, MOL, Gemini B, OAO, and SPARS.

A brief description is given below of those past, or current, efforts that are directly related to the FGSS study or that involve similar technology or advanced development. Also described, in Section 5.2, are those computer programs, developed or utilized during the Phase I (QRGT) study, and available at IBM for immediate use in the proposed FGSS effort. The combination of qualified personnel with related experience and the direct involvement in the Phase I study will assure satisfactory completion of the FGSS program.

5.1 APPLICABLE EXPERIENCE

5.1.1 QUICK REACTION GUIDANCE AND TARGETING STUDY (QRGT)

The initial effort on Flexible Guidance and Software System was designated Quick Reaction Guidance and Targeting (QRGT). Contract AF 04(695)-1078 was awarded to IBM in August 1966 and was completed in June 1967. The Final Technical Report on Phase I was published as Air Force Report No. SSD-TR-67-122.

The results of the Phase I effort indicated that the concept of highly automated, self-contained, quick-reaction mission planning and execution was feasible, and that considerable reduction in software costs and lead times could be achieved. The major accomplishments of Phase I were:

- Preparation and test of Trajectory Generation Program (TGP) and Trajectory Optimizer Program (TOP). These programs demonstrated the flexibility and adaptability of the Mission Planner.

- Preparation and test of a Direct Search Optimization Program (DSOP) for use in TOP.
- Preparation and test of an optimal-explicit guidance algorithm (OP-EX). OP-EX is versatile and is usable for exo-atmospheric ascent, orbit transfers, rendezvous, intercept, and deboost.
- Development and test of a predictive guidance system for use during re-entry by lifting-body vehicles. A basic approach was developed for displaying the destination relative to maximum range and cross-range.
- Specification of performance requirements for components in autonomous navigation systems for three (3) classes of missions.
- Specification of basic spacecraft computer requirements for a sophisticated FGSS system.
- Functional design of an advanced Ground Operating System (GOS) for the preparation, integration and validation of space software.
- Functional design of a configurator to assemble Specific Vehicle Libraries (SVL) from the Multi-Mission Library (MML).
- Functional design of an On-Board Executive System (OBES) to manage resources and schedule application programs in real-time on the spacecraft computer.
- Determination of basic crew tasks and computer/display requirements for manned missions.
- Analysis of error propagation in the guidance and navigation systems.

This experience from Phase I provides IBM with a unique baseline for the proposed FGSS program.

5.1.2 SPACE PRECISION ATTITUDE REFERENCE SYSTEM (SPARS)

IBM has recently been awarded Contract F04701-68-C0067 from SAMSO to conduct a Phase 0 design study of precise attitude reference systems for spacecraft. The IBM approach utilizes star trackers and an inertial measurement unit operating with a spacecraft computer.

The Phase 0 study includes preliminary design of SPARS, preliminary design of PEPSY (a precise pointing system), design of laboratory and space experiments to evaluate SPARS and PEPSY, software for space operation and performance evaluation, integration of SPARS and PEPSY into a payload, and determination of AGE requirements.

The SPARS program will also be managed by personnel in the McLean Building under the direction of Mr. R. J. Orrange. All systems engineering effort will be performed in Endicott. Work on spacecraft system error models, determination of computer requirements, and payload integration will be applicable to the FGSS program. Study of the overall space and ground operation for experiment planning and determination of software requirements for advanced space hardware will also ensure a current state-of-the-art treatment of these factors in the FGSS program.

The SPARS and FGSS programs are both being conducted for the Space Guidance Branch of SAMSO. Therefore, the opportunity exists to reduce travel costs (and time) and minimize liaison problems by coordinating working group meetings and Technical Direction conferences.

5.1.3 MOL DATA COMPUTATION SUBSYSTEM GROUP (DCSG)

IBM is currently developing the Data Computation Subsystem for the Manned Orbital Laboratory Program under contract to the Douglas Aircraft Company. IBM efforts include design, development and manufacture of the Airborne Digital Computer, Laboratory Data Adapter Unit, Auxiliary Memory Unit, Printer, and Control and Display Units.

As part of the system engineering task during Phase IB, IBM analyzed all laboratory data processing requirements. Functional flow charts were prepared showing interface with the executive control programs and control program services and frequency required. Detailed math/logic flow diagrams were also prepared and trial programming was performed to estimate computer speed and storage requirements.

IBM also conducted man-machine interface analyses during Phase IB related to nominal and contingency tasks for the DCS. This included the determination of the sequence, frequency, and interrelation of specific tasks; the definition of human performance requirements; the identification of information requirements; and the identification of controls and communications required to implement these decisions.

5.1.4 MANEUVERABLE SPACECRAFT STUDY (S-5)

The Denver Division of Martin Marietta Corporation recently completed Study S-5 for the Air Force under Contract No. F04695-67-C-0124. The work includes design, performance, and operations study of maneuverable spacecraft which can perform a wide spectrum of space operations in the 1970-75 period. Space Systems Center personnel in the McLean Building provided design data on Data Management Systems and Guidance and Navigation Systems under a subcontract to Martin.

This effort is important to the FGSS program as it provides a broad interface with booster and spacecraft vehicle design requirements and advanced spacecraft system (hardware) requirements. Of particular importance are the integration of booster/spacecraft guidance functions and the study of re-entry navigation and guidance. The association with Martin Denver exposes SSC personnel to current state-of-the-art of booster and spacecraft technology.

5.1.5 OPTIMAL GUIDANCE FOR ORBIT TRANSFER INVESTIGATION (OGOTI)

IBM conducted this study for the Space Systems Division of the Air Force Systems Command under Contract AF 04(695)-398. The objective was to develop guidance laws for accurate and efficient transfer of a space vehicle from one free-fall orbit to another, primarily for rendezvous purposes. The study consisted of four main parts: a direct search for minimum-fuel transfer trajectories, optimization of finite-burn transfer trajectories, definition of efficient guidance laws, and simulation with nonstandard engine performance.

A Kepler arc program was developed to accept target and vehicle orbit parameters, and compute fuel for rendezvous at the arrival and departure time. The trajectory solutions were based on Lambert's theorem.

For finite thrust optimization a steepest-ascent trajectory optimization method was developed which minimized memory requirements. Details of this optimization technique are described in the paper, "A Steepest Ascent Trajectory Optimization Method which Reduces Memory Requirements" (R. H. Hillsley and H. M. Robbins), presented in the book, Computing Methods in Optimization Problems by A. Balakrishman and L. Neustadt, Academic Press, 1964, pp. 107-135.

The optimization techniques and the detailed guidance schemes developed in the OGOTI study were not utilized in the QRGT study since more recent developments (based in part on the OGOTI experience) have made more attractive alternatives available. In particular, much better trajectory-optimization techniques now exist, or are being developed, than the steepest-ascent optimization techniques now exist, or are being developed, than the steepest-ascent optimization technique developed in the OGOTI study.

5.1.6 STANDARDIZED SPACE GUIDANCE SYSTEM STUDY (SSGS)

This study, under Air Force Contract No. 04(695)-525, consisted of mission analysis, system synthesis, preparation of system specifications, and program planning. The basic objective was to design a Space Guidance System with good cost effectiveness across a broad spectrum and mixture of missions of interest to the Air Force.

Computer storage and speed requirements for solving the guidance, navigation, and control problem were derived. Various guidance schemes were studied together with the burden placed on the computer by their mechanization.

Guidance sensor requirements were derived and an extensive state-of-the-art survey was performed to determine the acceptability of off-the-shelf hardware and the need for new developments.

Since several of the planned SSGS missions were manned, the tasks of man were identified. In general, functions were designated for human participation when the human offered reliability, flexibility, or unique characteristics.

The scope and level of operator participation in subsystem operation was defined and trial configurations of the operator instruments were established.

5.1.7 IMAGE VELOCITY SENSOR SUBSYSTEM STUDY (IVSS)

Under Air Force Contract 04(695)-656, IBM conducted a Phase 0 study on the Image Velocity Sensor Subsystem for the MOL program.

Parametric studies were performed and simulations were made to determine what man could accomplish in acquiring and tracking visual targets and in compensating for image motion due to guidance, navigation, and control deficiencies. The basic sensor consisted of a direct-viewing pointing and tracking scope (PTS) with a coupled camera. A tracking servo system was used in conjunction with a general purpose computer.

Equations and math flows were generated for processing the PTS gimbal data for space navigation and control functions. Computer speed, storage, and flexibility requirements were established. Flight control sensor requirements and operational mission constraints were specified for optimum performance.

Several studies were made of the man-machine interface. These included techniques for presentation of target briefing material, a general purpose real-time display of viewing targets, operation of the PTS and star trackers, horizon sensors, and radar altimeters. Backup modes of operation of the PTS were investigated including analog instrumentation concepts to assume the digital computer was inoperative.

5.2 COMPUTER FACILITIES

5.2.1 COMPUTERS

All of the necessary computers are available in the Endicott/Owego area. These include IBM 7090's, 7094's, and System 360 Models 40, 50, and 65. It is anticipated that Tasks 1 and 4 will utilize mainly the 7090/94 facilities, while Task 2 will utilize mainly the System 360 facilities. Support software available includes FMS/FAP/FORTRAN and IBSYS/MAP/FORTRAN for the 7090/94's, and OS360/Assembler/PL-1/FORTRAN for the 360 systems. The Support Software Study (Task 2) will make heavy use of the facilities of Operating System 360 - particularly the linkage editor facilities, the utility programs, and the supervisor and data management macros for gaining access to OS360 system services. The 7094 installation at Owego includes a CALCOMP model 545 plotter and support software for preparing magnetic tape input to the plotter.

5.2.2 APPLICABLE COMPUTER PROGRAMS

Tasks 1, 2 and 4 will require computer simulation support during the proposed FGSS study. In the case of Tasks 1 and 2, the computer programs (in terms of Math Flows or Functional Flows) will also form a part of the design specifications.

In support of these tasks, use will be made of programs developed during Phase I as well as other IBM-developed programs. In the former category are TGP (Trajectory Generator Program), TOP (Trajectory Optimizer Program) and the OP-EX (Optimal Explicit) guidance algorithm. The latter category includes GISMO - a generalized 3 degree-of-freedom Trajectory Simulator and various error analysis programs (ANS, INEAD and MM).

A brief description of these programs is given below:

5.2.2.1 Trajectory Generator Program (TGP)

TGP is written in FORTRAN IV to run on IBM 7090/7094 computers. The program is employed to generate a preliminary trajectory (or trajectories) for near earth missions from launch to final orbit injection or gross rendezvous.

The program uses analytical equations for calculating launch conditions and the atmospheric phase of ascent. The vacuum phase of ascent (to orbit injection or intercept) is generated with OP-EX. All orbital burns are considered as impulses and are determined by two different techniques to provide alternate maneuvers. A spherical earth model is employed and coast arcs computed from Kepler-type equations. The trajectories generated by TGP are specified by parameters which identify the mission "phases" such as: launch conditions, orbit injection state, and time and ΔV for all orbital burns. These trajectory parameters are then used with TOP (Trajectory Optimizer Program) to minimize total ΔV (or time) by a direct-search technique. A more complete description of this program is given in Appendix I of the Phase I Final Report.

5.2.2.2 Trajectory Optimizer Program (TOP)

TOP is written in FORTRAN IV to run on IBM 7090/7094 computers. The program is used to generate local minimum fuel (or time) trajectories for near-earth missions. The mission can start at launch or from orbit and can include up to four orbital burns for final orbit injection or gross rendezvous.

The program employs a direct-search optimization algorithm (several options available) which systematically varies the parameters of an initial (input) trajectory in order to minimize the total ΔV or time required to complete the mission. Inequality constraints on trajectory parameters are enforced in the search routine (for certain options) while functional constraints are handled by penalty terms in the pay-off calculation. More details on this program can be found in Appendix II of the Phase I Final Report.

5.2.2.3 Optimal-Explicit Guidance (OP-EX)

The Optimal-Explicit guidance algorithm - OP-EX, developed in Phase I, has been implemented with various capabilities. This program will be employed in Task 1 and will be the basis for guidance algorithm improvements. A description of the program is given in Appendix III of the Final Report on Phase I.

5.2.2.4 General Integrated Simulation Model (GISMO)

A versatile point mass vehicle simulation program especially constructed to allow easy implementation of candidate guidance algorithms. It is easily employed to investigate the effects of guidance errors caused by uncertainty in drag, and either the neglect of or the use of simplified formulae for the oblateness correction. GISMO will be employed in Tasks 1 and 4.

5.2.2.5 Autonomous Navigation Simulation Program (ANS)

This IBM 7090/7094 program simulates autonomous navigation as performed by a Kalman Filter using data from selected combinations of horizon sensor, star tracker, landmark tracking telescope, radar altimeter and range (laser, radar) measurements. Two modes are programmed: an error analysis mode and a full simulation mode; both navigation accuracy and the time history of the time and estimated orbit parameters are computed in the full simulation mode.

5.2.2.6 Inertial Navigation Error Analyst and Diagnostician (INEAD)

This is a navigation error analysis program for Strapped Down and Inertial Systems. The program generates the sensitivity of position, velocity and attitude to errors to ninety-one errors in the navigation system. The program computes the error covariance matrix for position, velocity, and platform attitude, at the end of a boost, orbital or reentry guidance phase.

5.2.2.7 Matrix Manipulator (MM)

The matrix manipulator program performs addition, subtraction, multiplication and inversion of matrices. Several specialized matrix operations are included which facilitate error analysis. The program was designed to be used directly by engineers who have no familiarity with computer programming. This program will be employed in Task 4 as a supplement to the ANS and INEAD programs.

Programs are also available for navigation error analysis of ground tracking (radar) stations and boost guidance systems, and for the determination of the time interval a space vehicle may be visible to given ground stations.

Section 6

REFERENCES

1. Quick Reaction Guidance and Trajectory Study (U), SSD-TR-67-122, June 1967, Vol. I Secret, Vol. II (Appendices) Unclassified.

2. Robbins, H.M., "An Analytical Study of the Impulsive Approximation", AIAA Journal 4, August 1966, pp 1417-1423. Unclassified

3. McCue, G.A., "Quasilinearization Determination of Optimum Finite-Thrust Orbital Transfers", AIAA Journal 5, April 1967, pp 755-763. Unclassified

4. Caper, E.B., Camiel, J.J., McNaughton, R.V., et. al., Variable Point Guidance and targeting, Final Report, CR-67-588-Z, December 1966. Unclassified

5. Lion, P.M. and Handelsman, M., "The Primer Vector on Fixed-Time Impulsive Trajectories", AIAA Paper No. 67-54.

IBM FEDERAL SYSTEMS DIVISION
SPACE SYSTEMS CENTER
ENDICOTT NEW YORK