

Spacecraft guidance, navigation, and control requirements for an intelligent plug-n-play avionics (PAPA) architecture

Nilesh Kulkarni*

QSS Group, Inc., NASA Ames Research Center, Moffett Field, CA 94035

Kalmanje Krishnakumar†

NASA Ames Research Center, Moffett Field, CA 94035

Abstract

The objective of this research is to design an intelligent plug-n-play avionics system that provides a reconfigurable platform for supporting the guidance, navigation and control (GN&C) requirements for different elements of the space exploration mission. The focus of this study is to look at the specific requirements for a spacecraft that needs to go from earth to moon and back. In this regard we will identify the different GN&C problems in various phases of flight that need to be addressed for designing such a plug-n-play avionics system. The Apollo and the Space Shuttle programs provide rich literature in terms of understanding some of the general GN&C requirements for a space vehicle. The relevant literature is reviewed which helps in narrowing down the different GN&C algorithms that need to be supported along with their individual requirements.

I. Introduction

An intelligent plug-n-play avionics (PAPA) system can provide critical capabilities for different space exploration missions. Any exploration mission involves the guidance, navigation and control of different systems and their sub-systems. Designing individual avionics solutions for each of these systems substantially increases the overall program costs. A plug-n-play system that can support the different GN&C requirements of all these systems in their different flight/ground phases provides an attractive affordable solution to this problem. Further, as the need emerges for designing new systems for exploration, the PAPA system can provide a sustainable technology solution. Besides working across different systems, the PAPA system can have on-board intelligence for adapting the controllers for unforeseen system failures and changes in mission. This provides an important analytical fault-tolerant capability that can be augmented with hardware fault-tolerance by using devices such as field programmable gate arrays (FPGAs).

The PAPA system design involves two elements, a plug-n-play component and an intelligent plug-n-play architecture that supports the component. The plug-n-play architecture must be able to recognize the devices and establish communication with these devices. This requiring a level of standardization to be imposed on the architecture that specifies the communication protocols, data bus specifications, power requirements, physical interface requirements, etc. Protocols must also be established to allow the necessary modifications to the avionics system as well as the plug-and-play modules to accommodate a previously unknown module configuration in a wide spectrum of possible platforms such as rovers and spacecrafts. For instance, sensor information and actuator configurations will be often radically different on each platform. Standardization requirements definitions will establish the communication protocols to allow this information to be passed between the system and modules, and must define the tradeoff between flexibility and complexity of the architecture to ensure usability without curtailing functionality.

* AIAA Member, nilesh@email.arc.nasa.gov

† Group Lead, Adaptive Control Technologies, AIAA Associate Fellow, kkumar@mail.arc.nasa.gov

Figure 1 illustrates the relationships between the PAPA architectural manager, the guidance, the navigation and the control component of an avionics system and their interaction with the physical system. The PAPA architectural manager identifies the physical system as well as the flight/ground operational phase of the system. It then uses this information to reconfigure the individual GN&C components from a set of available components for that particular system and its phase of operation.

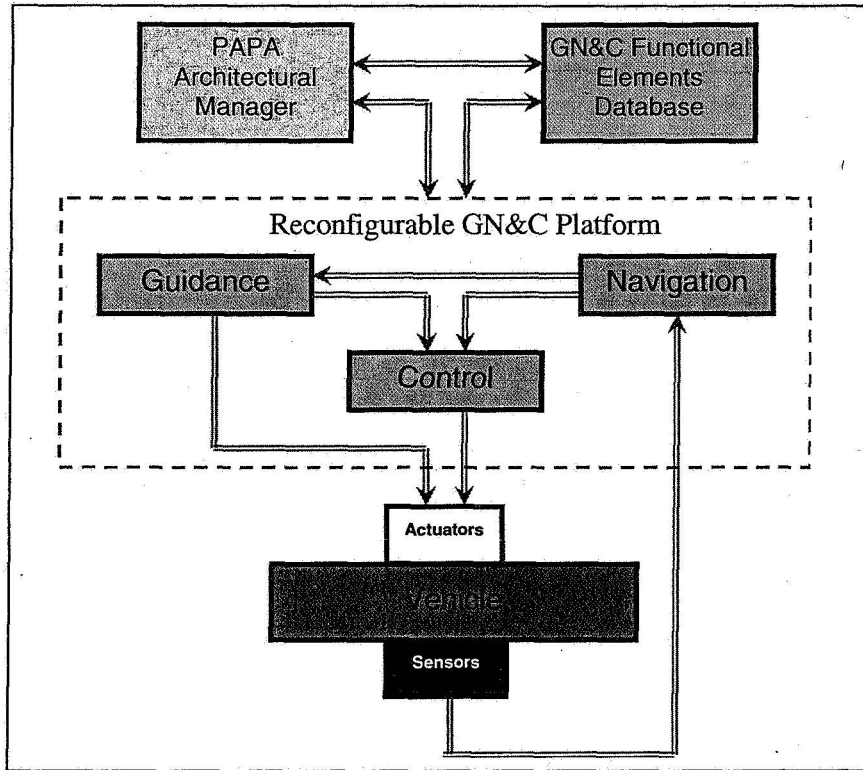


Figure 1: PAPA Architectural Flow

This needs a data-base of the GN&C functional elements that consists of all the different guidance algorithms, the navigation algorithms and control algorithms which will be used by the reconfigurable plug-n-play component as functional blocks. The PAPA architectural manager will use these functional blocks for constructing the GN&C system on the reconfigurable plug-n-play component knowing the system and its operational phase. It is therefore essential to understand the GN&C problems for relevant space systems in all their operational phases and use this information for generating the necessary set of *downloadable* GN&C functional blocks.

Section II reviews the GN&C problems in the Apollo and the Space Shuttle programs. Section III identifies the different GN&C blocks used in various phases of flight. Finally the conclusions of the study are presented in section IV.

II. GN&C Flight Phases

The Apollo and the Space Shuttle programs provide a rich set of literature in understanding the different GN&C problems for a space vehicle. The different phases of flight can be broadly classified as earth vicinity operations, space operations and lunar vicinity operations. The earth vicinity operations can be sub-classified into the ascent/boost phase that include its various abort phases depending on the position of the vehicle, the atmospheric reentry phase, the entry cruise phase and the different terminal area/landing phases. In a similar manner the lunar vicinity operations can be sub-classified into the lunar descent and the lunar ascent phase. The space operations include the earth and lunar orbital phases along with the travel between these two phases. Space operations also include special flight critical phases such as the autonomous rendezvous and docking phase. In the subsequent subsections various flight phases will be listed and the GN&C architectures associated with their operations reviewed.

A. Boost/Ascent Phase¹

The boost/ascent phase of the space shuttle orbiter consists of several individual stages. At lift-off, two solid rocket boosters (SRBs) are used to take the vehicle to an altitude of roughly 140,000 feet. The SRBs can be gimballed to control the attitude of the vehicle. After the SRBs deplete their source of the solid propellant, the casing is jettisoned. The Space Shuttle main engines, then take the orbiter to an orbit at which the external tank falls off. At this point, the orbital maneuvering system (OMS) engines are fired to insert the orbiter into an appropriate orbit. These also involve two separate burns. The first OMS engine burn raises the orbit apogee to the desired orbit altitude and the second burn inserts the orbiter into the desired parking orbit. The boost phase also includes three abort stages that can be invoked depending on the position of the orbiter. The return to landing site (RTL) abort stage is available for the first 270 seconds after lift-off when the main engines can be used to guide the vehicle back to its landing site. The abort once around (AOA) sub-phase is available prior to the second OMS burn when the vehicle traverses one orbit to return to the landing site. Finally the last abort sub-phase, abort to orbit (ATO) is available during the second OMS burn when the orbiter goes to an appropriate orbit from which it can return to earth.

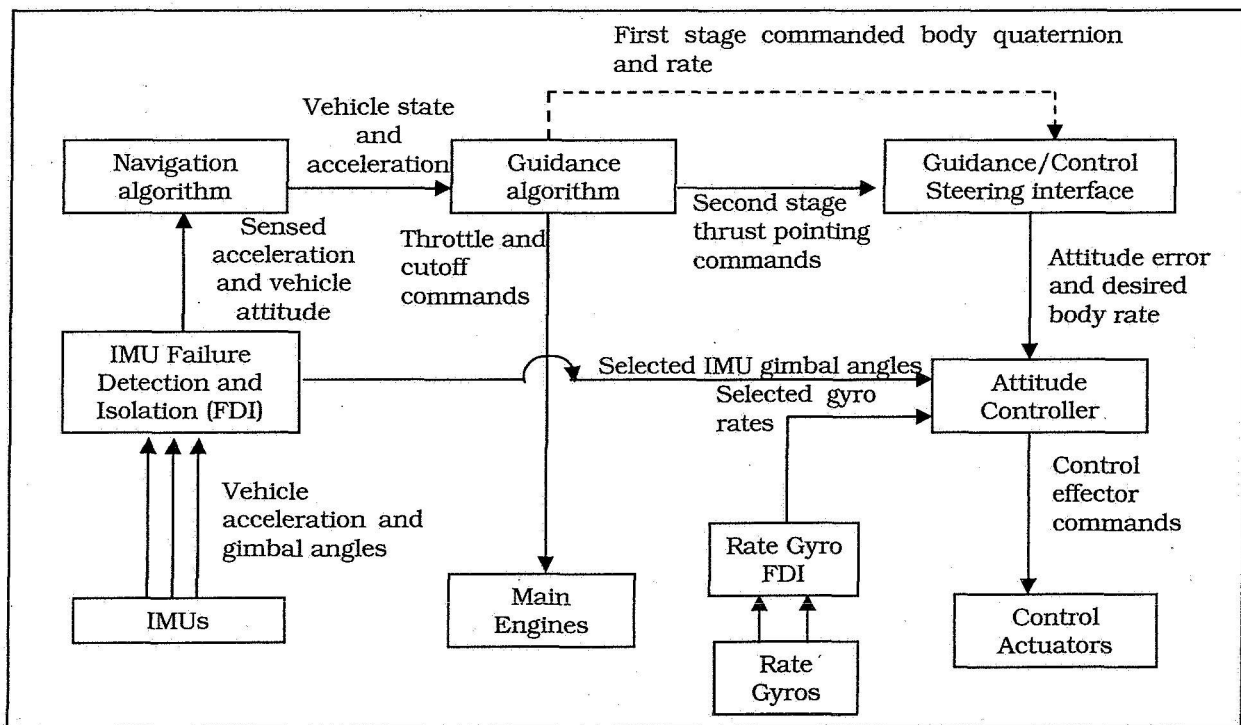


Figure 2: Boost/Ascent Phase Guidance, Navigation and Control Architecture¹

Figure 2 illustrates the GN&C flow for the boost phase of the space shuttle orbiter¹. The orbiter sensors consist of three sets of inertial measurement units (IMUs) that provide the vehicle acceleration and the platform gimbal angles, and a set of two rate gyros that provide the orbiter angular rate information. A failure detection and isolation algorithm compares the outputs of the different IMUs and sends data from the chosen IMU and rate gyro to the navigation algorithm. The navigation algorithm processes these inputs to provide the estimates of the states of the orbiter. During the first stage lift-off while the SRBs are in operation, the guidance algorithm provides commanded values of body quaternion and body angular rates. After the SRBs have been jettisoned, the guidance algorithm provides the commanded values for the thrust vector of the main engine and OMS engine. Since the attitude controller is designed to correct for small errors, a steering function is designed that takes the commanded values given by the guidance block (that could be potentially involve large error corrections) and provides smoothed desired values of the body angles and rates. The attitude controller therefore always sees small error corrections and provides actuator commands for zeroing them.

1. Boost/Ascent Guidance algorithm

The guidance algorithm has to provide command signals in different stages of the boost phase, each of which involve different objectives and constraints. In the first stage after lift-off, the objective of the guidance algorithm is to take the vehicle through the atmosphere while minimizing the fuel spent and keeping the structural loads in their design range. To achieve that, the algorithm provides attitude commands as a function of the vehicle velocity. These values are pre-computed and stored in the computer memory. During the first stage, the guidance algorithm also provides pre-computed values of throttle commands to the main engine depending on the vehicle attitude to restrain the total g-forces felt by astronauts.

The later stages of the ascent phase use a closed-loop guidance algorithm. The guidance algorithm provides steering and throttle values for the thrust vector while satisfying constraints in terms of the vehicle altitude, velocity, flight path angle and orbital plane. The algorithm is closed-loop in a sense that instead of using a table-lookup based approach as in the first stage of ascent, given the estimate of the state of the vehicle; it uses a prediction-correction based iterative procedure in an online manner to meet the imposed constraints. The algorithm uses the linear tangent steering law, developed for the Saturn V guidance, which imposes a linear time-varying functional form for the thrust vector².

$$\lambda_f = \lambda_v + \dot{\lambda}(t - t_\lambda) \quad (1)$$

Here λ_f is the commanded thrust direction vector, λ_v represents the unit vector in the direction of the velocity that needs to be gained by the S/C, $\dot{\lambda}$ represents the rate of the change of the commanded thrust direction, t is the current time and t_λ represents the time when the commanded thrust direction vector will align with the velocity to be gained vector. A prediction-correction scheme computes the variables λ_v , $\dot{\lambda}$ and t_λ in an iterative manner to satisfy the various constraints on the S/C trajectory. These values then provide the commanded thrust direction vector. In addition to providing the thrust direction vector, the guidance algorithm also provides roll commands that are based on a table look-up scheme.

Steering:

The commanded thrust direction vector computed by the guidance algorithm translates into commanded values of the body attitude represented by the quaternion, q_i^{CB} . Similarly, the vectors λ_v and $\dot{\lambda}$ provide commanded body rate, ω_i^{CB} , for the vehicle to achieve its commanded body attitude.

$$\omega_i^{CB} = \lambda_v \times \dot{\lambda} \quad (2)$$

These values can correspond to big changes from the actual body attitude and rates, and the controller may not be able to correct these as it is designed using linear techniques that assume small errors. The steering architecture takes the commanded body attitude and rate values and converts them into desired body attitude and rate values. Compared to the commanded values that could represent discontinuous jumps in time, the desired values vary sufficiently slowly with time and dynamically reach the commanded values. These desired values always produce small error corrections that are achieved by the controller. Traditionally steering has been considered as a separate functional form in addition to guidance, navigation and control. However in this discussion, steering is considered as part of the guidance block sets.

Figure 3 describes the steering architecture¹. The sensed change in velocity unit vector from the IMU provides the sensed unit thrust vector. This is filtered to give the estimated unit thrust vector, which along with the unit commanded thrust vector and commanded roll angle, is used to compute the commanded body quaternion. This is subtracted from the current value of the desired body quaternion to produce the steering error. A maneuver function takes the rate error and the steering error to produce desired body acceleration.

$$\alpha + 2\xi\omega_n\omega_e + \omega_n^2\theta_e = 0 \quad (3)$$

θ_e represents the steering error, ω_e , the steering rate error and α , the desired body acceleration. ξ and ω_n represent the damping ratio and natural frequency of the second order response. The maneuver function uses

appropriate values of damping and natural frequency to produce smoothed values of desired body quaternion and body rates that allow the controller to correct for continuously varying errors that are always small in size.

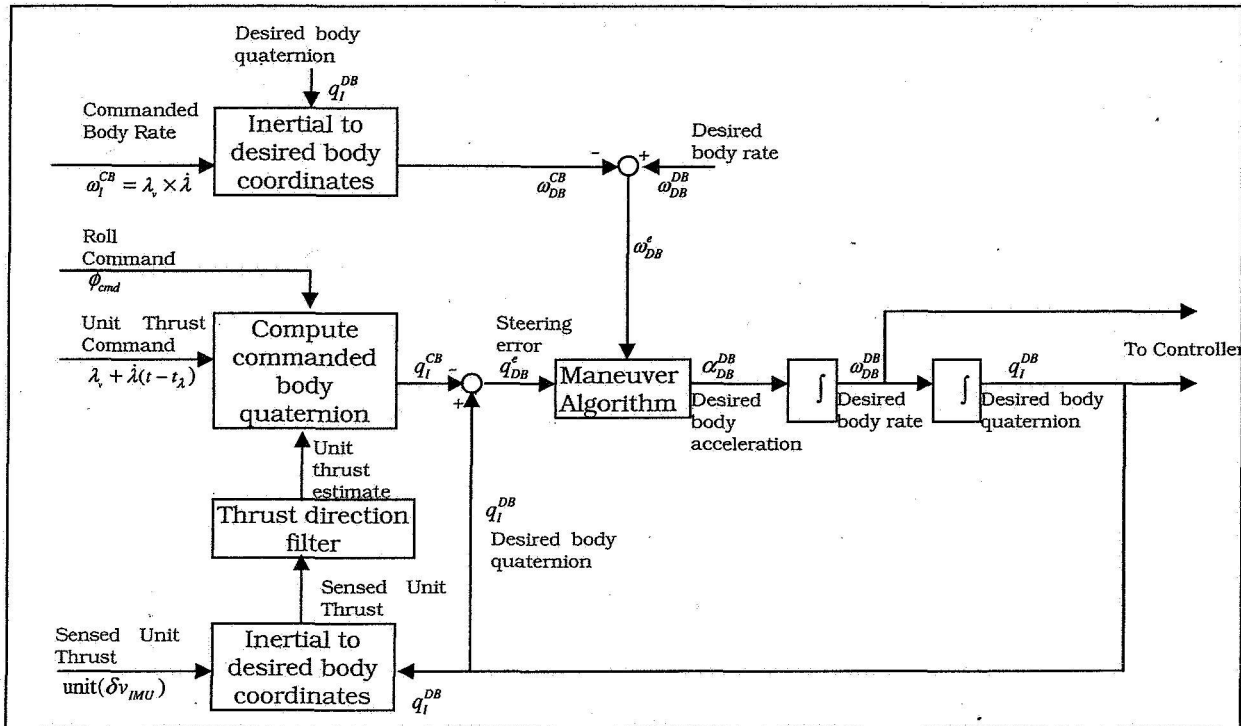


Figure 3: Boost/Ascent Phase Steering Architecture¹

2. Boost/Ascent Navigation Algorithm

The navigation algorithm consists of two separate elements. The first part of the algorithm gets the outputs of all the sensors that consist of three IMUs and three rate gyros and checks for sensor failures using a failure detection and isolation algorithm (FDI). The IMUs provide the vehicle attitude relative to the inertial frame of reference. The IMUs also have accelerometers that provide the inertial acceleration. The IMUs are placed on platforms that are skewed with respect to each other. This lets the FDI algorithm check for the relative angular attitude between the three platform orientations and identify if any of the IMU has failed. The same check holds for the accelerometer measurements.

After choosing a particular IMU, the navigation algorithm computes the vehicle position and velocity by integrating the IMU acceleration measurement and the gravitation acceleration obtained using the chosen gravity model.

3. Boost/Ascent Control algorithm

The control architecture is responsible for computing the engine deflection commands given the desired body quaternion and rates. This control architecture is valid for the first two stages of the boost/ascent phase corresponding to the space shuttle main engine cut-off followed by the external tank separation. The remaining two stages corresponding to the two OMS engine burns are similar to the on-orbit maneuvers that use simplified control architecture and are treated in the next section that deals with that flight phase.

Figure 4 illustrates the boost phase control architecture for the pitch channel³. The control architecture consists of two distinct elements, a PID control logic and control allocation implemented by the actuator mixing logic. Attitude and rate errors are computed using the desired body quaternion and rates from the steering blocks and the sensed body attitude and rates given by the IMU and the rate gyros. The attitude error is summed with the acceleration error to compute the composite attitude error. This error is then taken through a gain and an integrator block to compute the proportional and integral components of the PID control. Finally the computed rate errors multiplied by a gain provide the derivative component. The PID gains are scheduled as a function of time to provide

the appropriate response characteristics. The control architecture for the yaw and roll channel are similar to the pitch channel with an exception that the roll channel attitude error does not receive the acceleration error input.

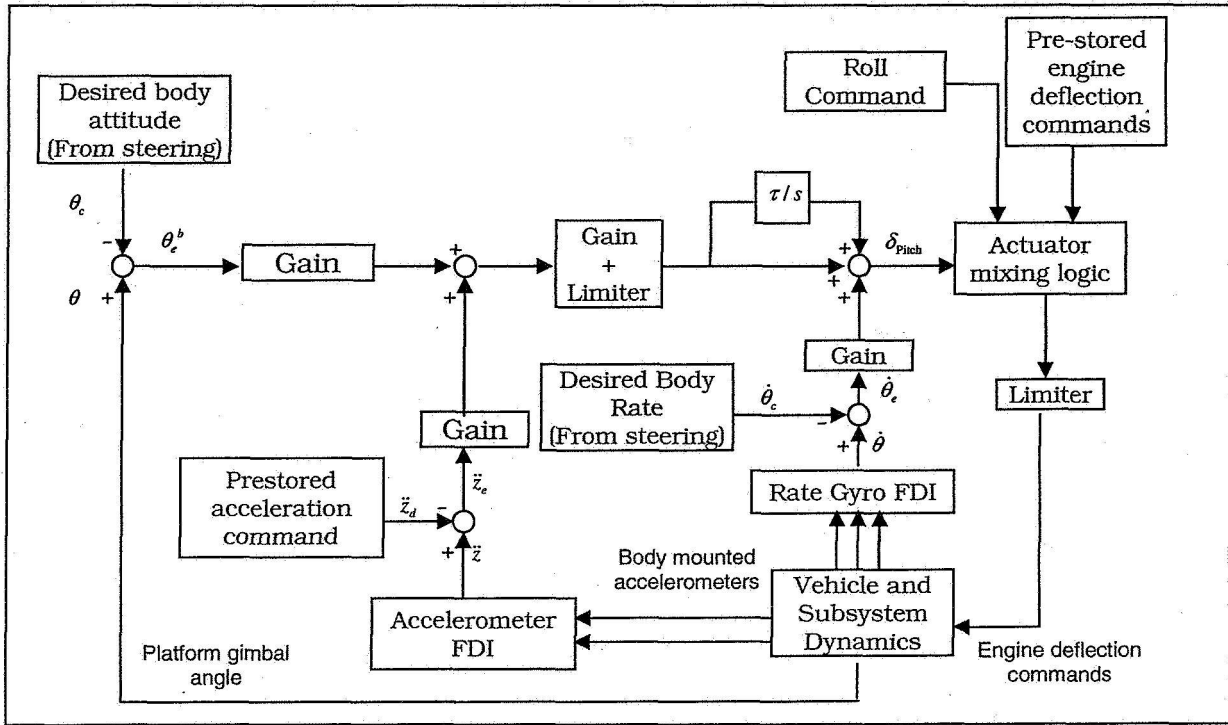


Figure 4: Boost/Ascent Phase Pitch Control Architecture³

B. On-orbit Phase

The on-orbit phase starts with the two OMS engine burns that place the shuttle in its appropriate orbit. This phase uses two distinct control architectures depending on the nature of the required maneuver. For changing orbits or changes in the velocity vector, the shuttle uses thrust vector control (TVC) of the OMS engine. For attitude changes in its given orbit, the shuttle uses the attitude control propulsion system (ACPS). The ACPS uses reaction jets to reorient the spacecraft.

TVC mode

Figure 5 illustrates the GN&C architecture for the TVC mode⁴. The architecture represents a simplified version of the boost/ascent GN&C architecture. The guidance algorithm for this phase is a simplified form of the boost phase algorithm given by Eq. (1). While Eq. (1) is solved with several constraints on altitude, velocity, flight path angle and orbital plane, the guidance algorithm for the on-orbit phase has fewer path constraints. Similarly, the boost phase involves thrust to weight (T/W) ratios around 1 through 3 and burn times on the order of a few minutes, the on-orbit phase has T/W ratios around 0.06 through 0.02. The burn times for the OMS engines can last up to 20 minutes.

The guidance algorithm computes the velocity to be gained and the throttle command using the relevant set of constraints for the particular maneuver. The commanded angular velocity is then computed using the cross product of V_g and the change in the spacecraft velocity, ΔV , given by the accelerometer and is representative of the thrust vector. The commanded angular velocity then goes through the steering loop that produces a desired angular velocity value. The TVC mode does not use rate gyros, but uses back-differencing of the IMU gimbal angles to compute the estimated change in the vehicle attitude. This is differenced from the desired attitude change given by the desired angular velocity times the sample time, to compute the attitude error. The attitude error is input to the controller that consists of two separate elements, a digital compensation filter and a thrust misalignment correction (TMC) filter.

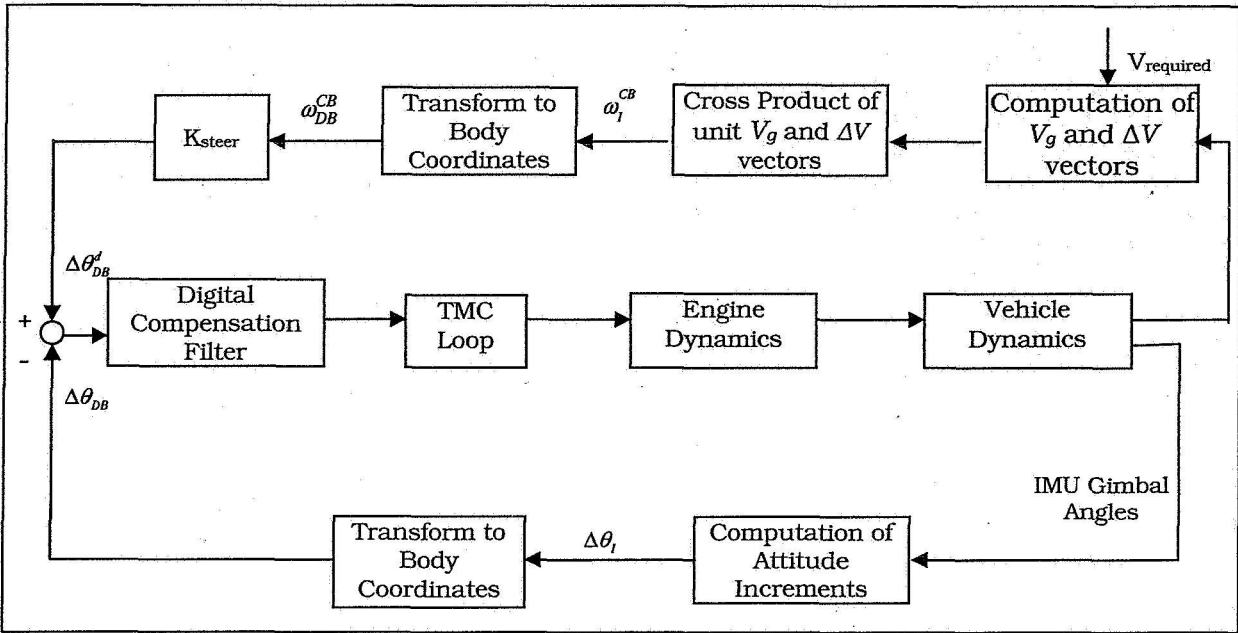


Figure 5: On-orbit Thrust vector control architecture⁴

ACPS mode

The ACPS mode is used in the re-entry phase as well as the lunar vicinity operations apart from on-orbit maneuvers. It involves firing a set of reaction jets in order to achieve the desired attitude and angular velocities. The control system is responsible for the scheduling and timing of the firing sequences. Given the inertia characteristics of the vehicle and the position of the jets, the control algorithm uses a set of control axes to decouple the sixth order rigid body dynamics of the vehicle into three independent second order dynamic systems. Each of these second order dynamics are then treated as a separate control problem.

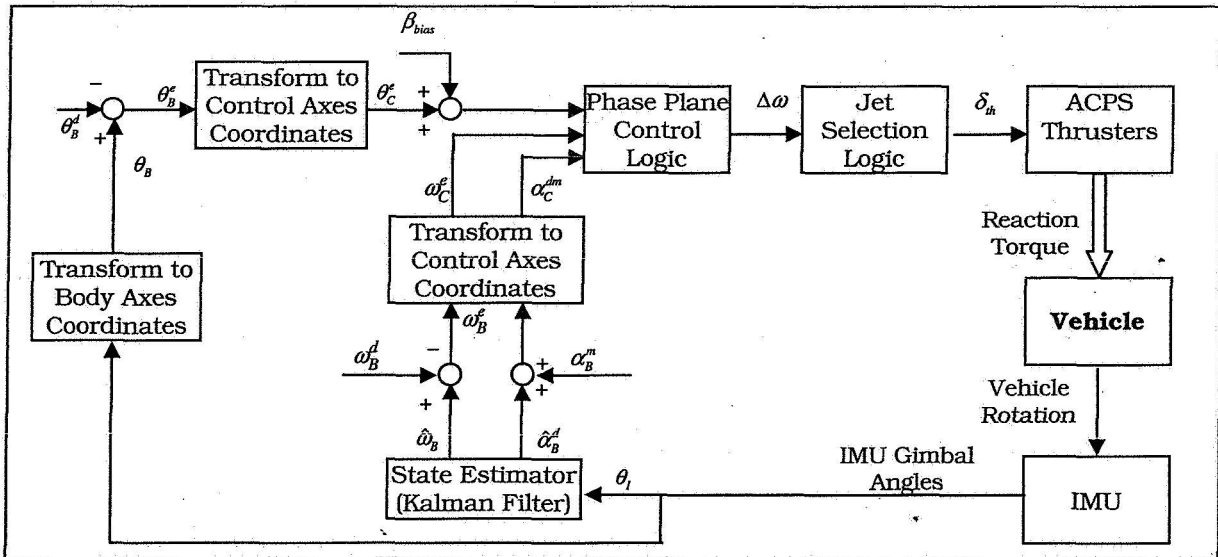


Figure 6: On-orbit ACPS control architecture⁴

The main component of the guidance algorithm for the ACPS mode involves the steering loop similar to the one illustrated in figure 3. The commanded values of the body attitude and rate are mission driven. The steering

architecture then converts these to smoothed desired values using a maneuver function. The ACPS mode does not have access to the entire system state. It uses a Kalman filter to estimate the state vector by using sensor measurements consisting of the IMU gimbal angles. The desired and the sensed values of the body attitude and rates are input to the control architecture which computes the jet firing sequence.

Figure 6 illustrates the control architecture for the ACPS mode⁴. The control architecture takes the desired attitude and attitude rate from the steering loop and the sensed values from the IMU and the Kalman filter. The attitude and rate errors are then converted to the control axes so as to decouple the control problem into three independent control sub-problems. The main elements of the control architecture are the phase plane switching logic and the jet selection logic. The phase plane switching logic provides optimal control paths in the attitude error-attitude rate phase plane that the spacecraft needs to traverse that minimize the fuel-time performance cost. While these optimal paths cannot take the spacecraft exactly to desired attitude and attitude rate, the spacecraft limit cycles around these desired values. The limit cycles can be chosen as close to the desired values as possible however, closer the limit cycle, higher is the reaction jet activity leading to increased fuel consumption.

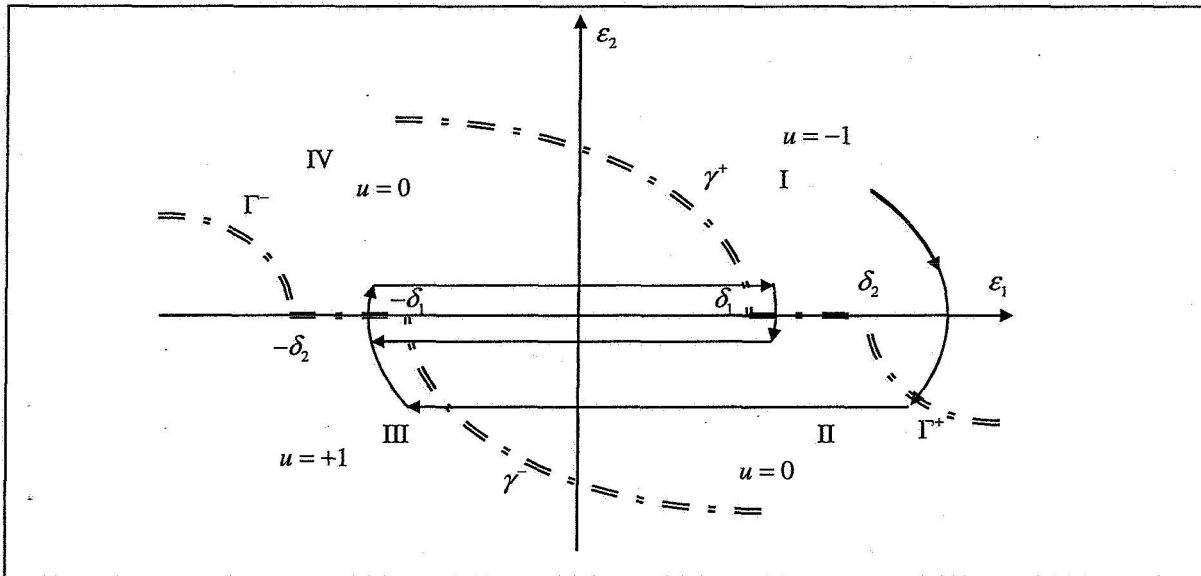


Figure 7: Simplified phase plane switching logic⁴

Figure 7 illustrates the operation of such a simplified phase plane switching logic⁴. The design of this phase plane logic involves designing the switch curves, γ^+ , γ^- , Γ^+ and Γ^- . ϵ_1 and ϵ_2 represent the angular error and the angular rate error respectively. The switch curves divide the two-dimensional phase plane into four regions. The control given by the reaction jets is represented by u . δ_1 and δ_2 represent parameters that describe the limit cycle behavior around the origin. The behavior of a typical trajectory is illustrated where it starts in Region I and the control actuation is given by $u = -1$. So given this constant negative torque, the trajectory follows the typical parabolic profile obtained by integrating the angular error and angular rate error equations. When the trajectory cuts the switch curve Γ^+ , no control is applied and the spacecraft coasts with a constant angular rate error. On cutting the switch curve γ^- , control $u = 1$ is applied corresponding to reaction jets opposite of the ones firing in region I. This makes the trajectory follow the parabolic curve that is a mirror image of the first parabolic curve about the ϵ_2 axis. Finally when the trajectory enters region IV, all reaction jet firing is ceased letting the spacecraft coast. From this point on, the spacecraft remains constrained to the limit cycle around the origin.

The above phase plane switching logic gets shifted if there is external disturbance acting on the system. The Kalman filter as shown in figure 6 actually estimates this external disturbance and sends it as an input to the phase plane switching logic which adjusts the switch curves depending on this disturbance. For the on-orbit phase, the phase plane logic is divided into two distinct regions, the small error logic and the large error logic. The small error logic is similar to the one illustrated in figure 7. The objective of the large error logic is to get the error trajectory into the small error region. Similar to the phase plane laws illustrated in figure 7, the actual phase plane laws are stated for both the small error and large error regions.

The second element of the control architecture is the jet selection logic. This is equivalent to control allocation techniques found in other control architectures. Depending on the current phase plane law that is being implemented, the jet selection logic chooses the particular reaction jets to be fired for achieving the control torques.

C. Entry Phase

The objective of the entry phase GN&C architecture is to bring the spacecraft down through the atmosphere with acceptable energy characteristics to provide smooth landing capabilities. The guidance algorithm provides angle of attack and roll commands to the spacecraft in order to achieve that. The control algorithm is then responsible for achieving those angle of attack and roll commands. The GN&C architecture for the entry phase depends on the vehicle characteristics. The Apollo entry module used only the reaction jets to control the attitude and speed of the vehicle while the shuttle used a combination of the aerodynamic surfaces and reaction jets to achieve that. In this section the shuttle GN&C will be discussed.

4. Entry Guidance algorithm⁵

The guidance algorithm, as in the other phases, is designed to optimize the energy and trajectory characteristics of the vehicle while satisfying the design constraints. For the entry phase, the primary constraints involve the proper operation of the thermal protection system, limited structural load on the vehicle and appropriate demands on the flight control system to minimize propellant usage and structural constraints on the actuators. These constraints translate to limiting curves in the drag acceleration versus earth relative speed plane. The guidance algorithm defines a drag acceleration profile that the vehicle needs to follow that lies bounded within these limiting curves. This drag acceleration profile is built using five individual drag reference segments. The first two segments are quadratic functions of the speed and correspond to the high speed aerodynamic heating phase of the entry. The next two segments consist of a pseudo-equilibrium glide for which the flight path angle is constant, and a constant drag segment. These two segments are applied for intermediate speeds. Finally the last segment which is a linear function of the energy is applied for low speeds. The connected drag acceleration profile using these five segments defines the reference profile.

The motivation in defining these reference segments analytically using simple functional curves lies in that the range can be predicted in an analytically closed form and the commanded value of the lift to drag ratio can be computed easily. For a quadratic drag profile as in the high speed phase given by

$$D = C_1 + C_2V + C_3V^2 \quad (4)$$

where C_1 , C_2 and C_3 are appropriately chosen constants for the drag profile, the commanded (L/D) is given by the analytic expression,

$$\left(\frac{L}{D}\right)_0 = -\frac{1}{D_0} \left(\frac{V^2}{r} - g \right) - 4 \frac{h_s C_1}{V^2} - 4 \frac{h_s C_2}{V} - \frac{h_s \dot{C}_{D0}}{C_{D0} D_0} \left(\frac{\dot{C}_{D0}}{C_{D0}} - \frac{D_0}{V} \right) + \frac{h_s \dot{C}_{D0}}{C_{D0} D_0} \quad (5)$$

The commanded value of the lift to drag ratio gives the commanded value of the angle of attack for the shuttle. The subscript zero for the commanded value of the lift to drag ratio refers to this being the open-loop value. The guidance algorithm also provides a closed-loop guidance law for the drag profile to return to the reference profile given errors in modeling and the environment. So the actual commanded value of the lift to drag ratio is given by

$\left(\frac{L}{D}\right)_c$ where

$$\begin{aligned} \left(\frac{L}{D}\right)_c &= \left(\frac{L}{D}\right)_0 + \delta\left(\frac{L}{D}\right) \\ \delta\left(\frac{L}{D}\right) &= f_1(D - D_0) + f_2(\dot{h} - \dot{h}_0) + f_4 \int (D - D_0) dt \end{aligned} \quad (6)$$

The constants, f_1 and f_2 are chosen to provide a second order like response for the drag error dynamics. The integral term with the coefficient f_4 is introduced to remove any steady state error. While the open loop lift to drag ratio is achieved using the angle of attack, the perturbation value is obtained by modulating a combination of the angle of attack and the bank angle. The bank angle is perturbed in one direction till the vehicle cross range exceeds a chosen value at which point the bank angle is reversed and perturbed in the opposite direction.

As noted above, the functional forms of the drag profile allow the range to be calculated in an analytical form as well. The guidance algorithm can therefore alter the constants in the drag profile without changing the functional form to minimize the difference between the predicted range and the range to target. During entry, the postblackout footprint capability is maintained since the orbiter may need to correct for significant navigation errors that can go uncorrected during the communications blackout. The guidance algorithm, therefore, perturbs parts of the drag profile to correct for range errors while maintaining the postblackout state of the vehicle close to the nominal state.

5. Entry Control architecture

The entry phase is active from around 400,000 feet altitude down to around 80,000 feet altitude. At the beginning of the phase, as there is no atmosphere, the attitude control authority is provided by reaction jets. As the orbiter altitude decreases, the aerodynamic surfaces start becoming effective and towards the end of the entry phase, the control authority is completely taken by the aerodynamic surfaces. During the intermediate region, control using the reaction jets and the aerodynamic surfaces is blended to provide the necessary control authority and to minimize excessive use of the reaction jet propellant.

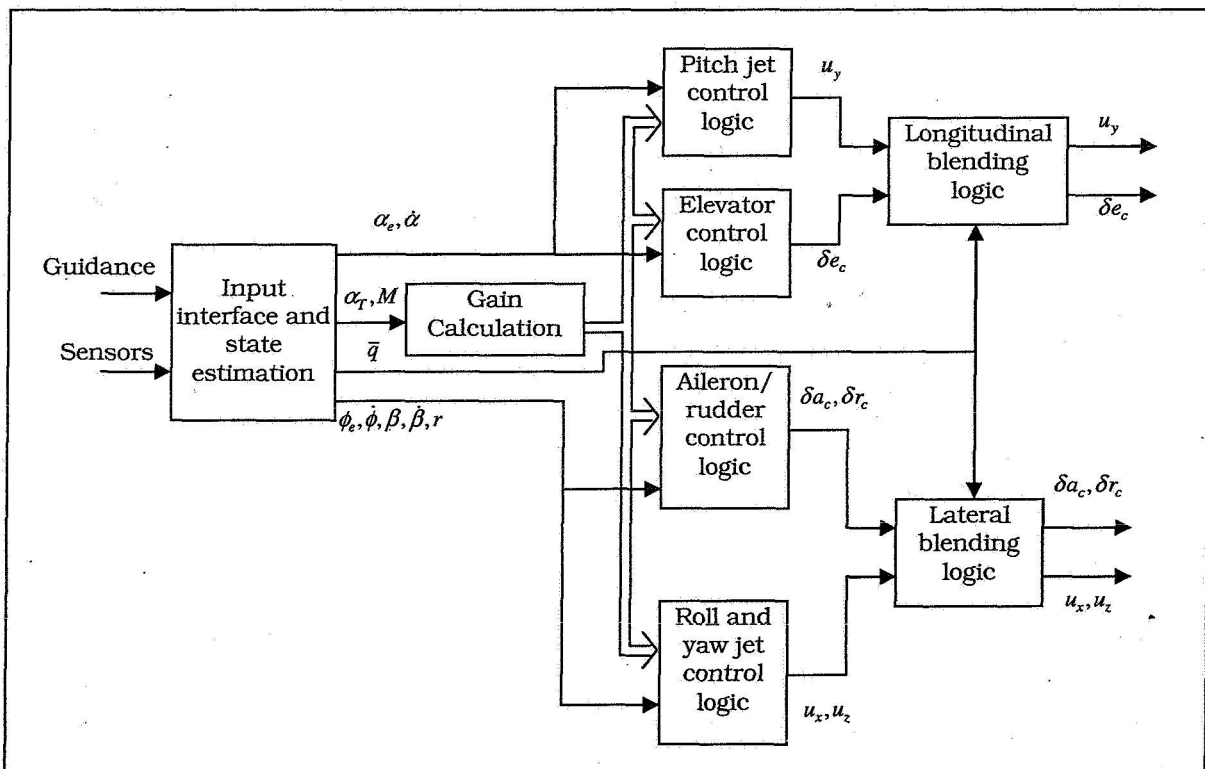


Figure 8: Entry Control Architecture⁶

Figure 8 illustrates the entry control architecture⁶. Body mounted rate gyros and IMU provide the attitude angles, attitude rates and the vehicle acceleration. A state estimator uses these sensor measurements to compute the estimates of the angle of attack, side slip, and bank angle errors from their desired values provided by the guidance algorithm and a steering function. Dividing these into the longitudinal and lateral components, they are provided to the aerodynamic surface control algorithm and the ACPS phase plane logic. Depending on the operational dynamic pressure of the orbiter, the outputs of the two control algorithms are blended to compute the effective control actuation by the aerodynamic surfaces and the reaction jets.

Longitudinal control

The longitudinal control is achieved by using the elevators and the pitch jets. The elevator operation involves two control sub-modes while the pitch jets have a single mode operation. During the early part of the entry only the pitch jets provide control authority for attitude corrections. In the intermediate part, when the dynamic pressure is low, the elevators are used to trim the vehicle while the pitch jets are used for any transient response. The elevators try to trim the vehicle by observing the pitch jet firing activity and try to compensate for that. Beyond a certain dynamic pressure value, the elevators start getting used for transient responses and the pitch jet activity is used in a limited capacity. It may be called for use if the demands on elevators start approaching their limits.

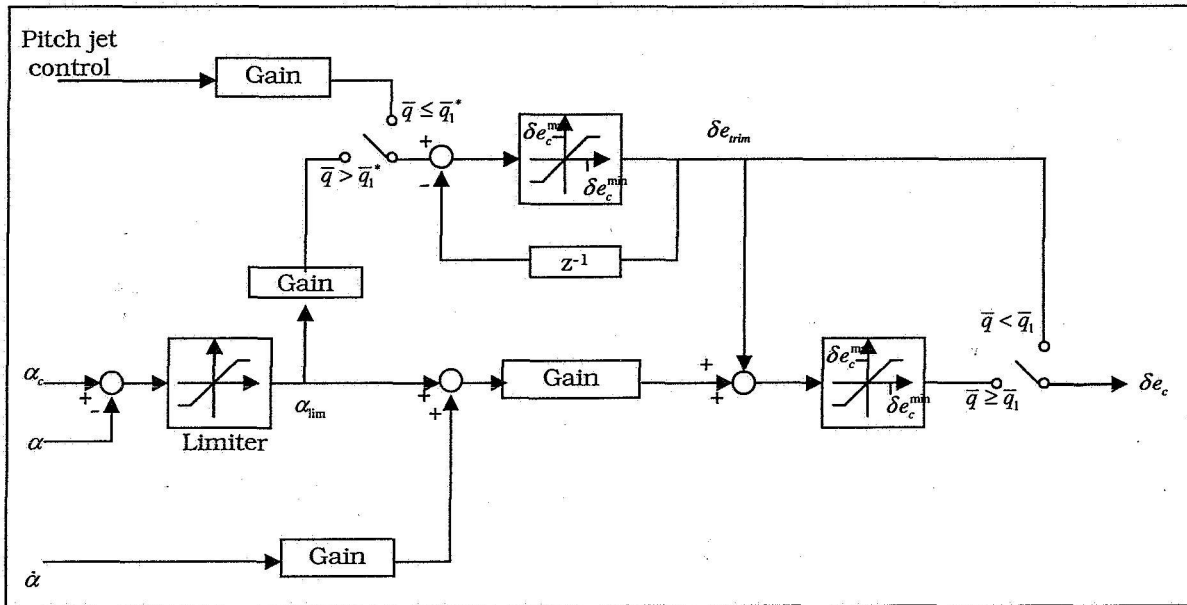


Figure 9: Elevator Control Architecture⁶

Figure 9 illustrates the elevator control architecture⁶. It represents a typical PID control loop with logic that brings in the dynamic pressure dependence. The PID gains are also scheduled as a function of the angle of attack and Mach number as illustrated in figure 8. \bar{q}_1 represents the dynamic pressure below which the elevators are used for trim-only control. \bar{q}_1^* represents the dynamic pressure beyond which the elevators are used as the primary control with limited activity from the pitch jets. The dynamic pressure range from \bar{q}_1 to \bar{q}_1^* uses control bending as discussed above.

The pitch ACPS control uses phase plane logic as illustrated in figure 7. The angle of attack error and the angle of attack rate error correspond to the coordinate axes (ϵ_1, ϵ_2) . A hysteresis logic is designed that prevents chattering between the pitch ACPS and the elevator control since both are working towards a common objective.

Lateral control

The lateral control architecture while similar to the longitudinal architecture involves more sub-modes. The aerodynamic control is achieved using the ailerons and rudders, while the ACPS control is achieved using the yaw and roll reaction jets. Depending on the dynamic pressure, the aerodynamic surfaces operate in one of three control sub-modes. Similarly the roll reaction jets operate in three control sub-modes while the yaw jets operate in a single mode.

At the beginning of the entry phase, the control authority is entirely exercised by the yaw and roll jets. The yaw jets operate on the bank angle-bank angle rate errors' phase plane logic while the roll jets operate on side slip-side slip rate errors' phase plane logic. If both yaw and roll jets are operating simultaneously, then the architecture involves logic to compensate for coupling effects due to the two sets of jets. Further on, when the dynamic pressure increases, the roll jets operate in a rate damping mode when there are no yaw jets operating. In the presence of yaw jets, the roll jets operate in the previous side slip-side slip rate errors' phase plane logic mode.

As the dynamic pressure increases and aerodynamic surfaces start becoming effective, the ailerons are used for both transient and trim control. The rudders start getting used further on when the Mach number reaches a set value and they are proportionally deflected depending on the aileron deflections. The yaw jets and the roll jets are used in their previous modes with blending logic to account for the aerodynamic control. The blending logic is similar to the one used in the longitudinal mode. The overall control logic for this phase is termed as *reverse* logic.

When the dynamic pressure increases to the level where the aerodynamic surfaces are fully effective, the yaw and roll jet activity is limited to minimize their propellant usage. From this point on, the orbiter flies like an airplane with redundant control authority provided by the reaction jets if needed. The control logic for this phase is termed as *proverse* logic.

D. Lunar descent/ascent phase

The lunar vicinity operation as followed by the Apollo missions is characterized by several phases. These include the lunar excursion module (LEM) injection phase when the LEM leaves its coasting orbit around the moon. The descent phase has the descent engines fired to remove the vehicle velocity. This is followed by the lunar landing approach phase that corresponds to the final few miles of descent when the LEM is directed towards the landing site. After completion of its mission, the LEM begins its ascent phase before finally docking with the Command and Service Module (CSM).

In all these phase, the LEM uses a combination of the reaction jets and thrust vector control to control the attitude and velocity vector of the vehicle. The reaction control system consists of 16 reaction jets arranged in clusters of four. The descent engine can be throttled to upto 12% of its maximum thrust as well as gimballed to provide TVC authority. The ascent engine is fixed rigidly to the LEM without any gimbaling capability. Hence during the ascent phase, attitude control is provided using reaction jets.

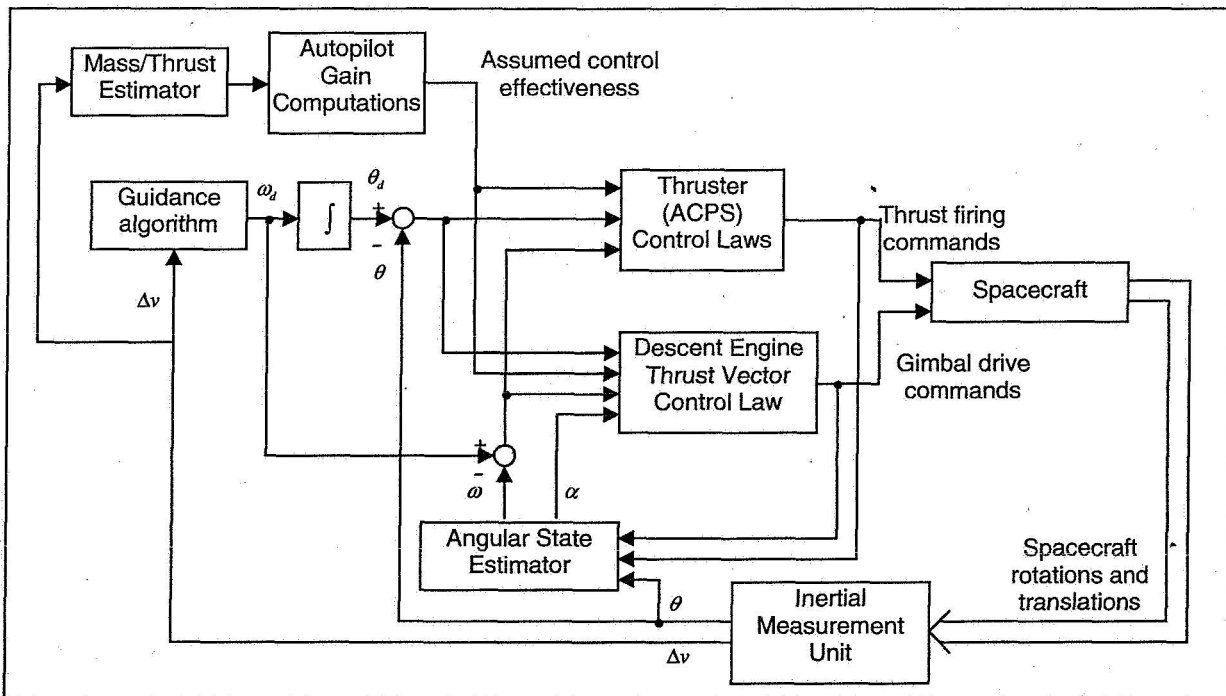


Figure 10: LEM GN&C Architecture⁷

Figure 10 illustrates the LEM GN&C architecture⁷. The LEM mission profile involves significant changes in its vehicle mass from the beginning of its descent phase to the ending of its ascent phase due to propellant depletion. This has significant effects on the vehicle translational and rotational inertia properties and subsequently in the effectiveness of the control actuators. The GN&C architecture therefore includes a vehicle mass estimator based on its velocity changes that are related to the thrust acting on it which is ultimately related to the propellant mass depletion. Based on the estimate of the vehicle mass, another block computes the effectiveness of both the TVC and ACPS control actuators on the vehicle. These measures of effectiveness are blended in both the control laws.

6. LEM Guidance algorithm

Apollo guidance laws formed the basis for the guidance algorithms used by the shuttle during its ACPS control phase and its TVC phase. The overall guidance approach used by the LEM during its various phases, therefore, remains the same as in on-orbit flight phase discussed for the shuttle. The difference in the guidance algorithms lie in the constraints and terminal conditions involved for the two problems. In both the descent and ascent phase of the LEM, solving the velocity required equations⁸ and given the current velocity of the LEM, the velocity to be gained is computed. The objective of the guidance laws is then to provide attitude rate commands to align the thrust vector with the velocity to be gained vector. For the descent phase, this is achieved by gimbaling the engine while the ACPS control achieves it in the ascent phase. In this section the constraints and terminal conditions used for the landing approach phase at the end of the descent phase are surveyed.

The LEM commander can choose and alter the landing site using his console. The objective of the guidance algorithm is to produce a reference trajectory to this chosen site while satisfying various visibility, attitude, path and velocity constraints. The reference trajectory has to be such that after the landing site re-designation, the landing site needs to remain visible through the window with a prescribed margin. Similarly the reference trajectory has to be such that the landing site has to remain in a particular plane with respect to the body axes. The pitch attitude of the vehicle throughout the maneuver has to remain with certain chosen limits. After landing site re-designation, the vehicle should approach this point from one side alone so that the curvature of the path lies on the same side of the path. Similarly both the forward and descent vehicle velocity magnitudes need to lie with a prescribed range. While these constraints increase the overall safety of the mission, they also provide a unique reference trajectory to the landing site.

The guidance algorithm commands the components of the total acceleration vector as quadratic functions of time⁹.

$$\bar{a}_c = \bar{a}_d + \bar{j}_f(t-t_f) + \frac{1}{2}\bar{p}_f(t-t_f)^2 \quad (7)$$

\bar{a}_c represents the commanded acceleration vector, \bar{a}_d , represents the desired acceleration vector at the end of the trajectory, \bar{j}_f , represents jerk or the derivative of acceleration at the end of the trajectory, and \bar{p}_f , represents snap or the second derivative of acceleration at the end of the trajectory. t represents the current time and t_f , the final time. \bar{a}_c , \bar{j}_f , \bar{p}_f and t_f represent the unknowns in this equation and are solved by specifying the end time constraints on the desired position, velocity and acceleration at the end time. Eq. (7) can be integrated to represent equations in terms of the desired position and velocity. The end time, t_f , that is also an unknown in this set, is found by specifying one component of jerk at the end time. Once the commanded total acceleration is computed, acceleration due to the moon's gravity is subtracted to get the commanded inertial acceleration and on integrating, the required velocity vector. The TVC guidance laws as outlined in the previous section are then applied to command an angular velocity in order to align the descent engine thrust with the velocity to be gained vector.

7. LEM Navigation

The LEM is navigated based on the sensor inputs given by the on-board IMU. During the lunar approach phase, landing radar provides distance to ground information. The navigation algorithm consists of a state estimator that uses measured attitude from the IMU gimbal angles, reaction jet firing information and the thrust vector command information to estimate the bias angular accelerations from the TVC and the vehicle angular velocities. Using the attitude at the previous time step and the reaction jet control and the thrust vector acceleration, the estimator predicts the vehicle attitude at the current time. This predicted value is compared with the measured attitude from the gimbal angles at the current time. The difference is used to update the estimate of the vehicle angular velocities and the thrust vector bias angular acceleration. The estimator gains are hand tuned to values that provide the best simulation results⁷.

8. LEM Control architecture:

The objective of the LEM control architecture is to use TVC and/or ACPS control logic depending on the flight phase, to achieve the desired angular velocity requirements provided by the guidance algorithm. The TVC actuator response is slower than the ACPS reaction jets response. For maneuvers requiring slow response time, therefore,

TVC is utilized while fast response time maneuvers use ACPS jets. The ACPS is based on single axis phase plane logic as illustrated in figure 7. For the LEM, the control axes are defined as P, U, V axes where the effect of the reaction torques about one of these axes has minimal effect on the other axes. The TVC system uses a minimum time control law that produces the fastest response possible for the TVC laws.

9. Manual LEM Attitude Control

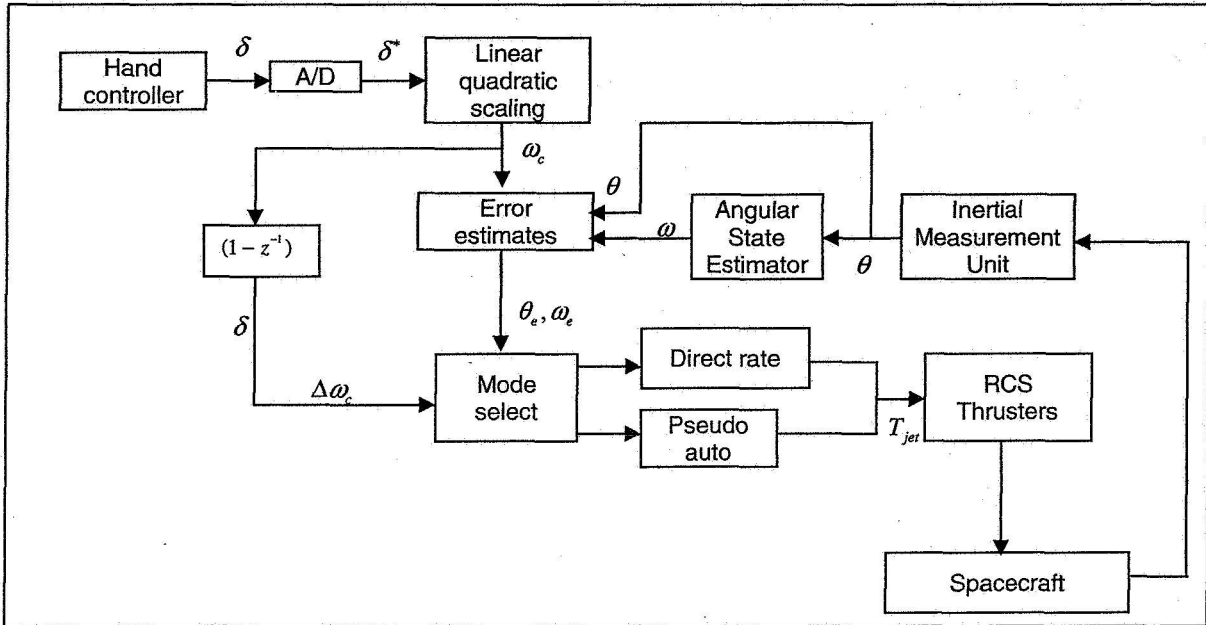


Figure 11: LEM Manual Attitude Control Architecture¹⁰

The guidance algorithm in the approach phase allows re-designation of the landing site as discussed in section 6. However the pilot does not have total control over the spacecraft trajectory which is generated by the guidance algorithm. At the same time, this guidance algorithm is active only upto 15 seconds before touch-down. A manual attitude control algorithm is therefore designed that gives the pilot the effective command of the vehicle whenever the need arises.

Figure 11 illustrates the LEM manual attitude control architecture¹⁰. The control algorithm employs a rate command/attitude hold mode so that whenever the pilot moves his hand controller, a rate command is issued to the control system. Similarly with no input from the pilot, the controller maintains the current attitude of the vehicle. The pilot essentially takes over the roll of the guidance system and cuts down on the processing time required to initiate the necessary maneuver. The manual attitude control uses only the reaction jets as they offer fast response time while the descent engine is being gimbaled to meet the required velocity.

The manual rate command logic includes two modes, the direct rate mode and the pseudo auto mode. The direct rate mode is provided for critical maneuvers when immediate action needs to be taken. If the input rate to the hand controller is more than a chosen value, the direct rate mode is used where the firing time for the reaction jets is given by

$$T_{jet} = -\frac{\omega_e}{a_{jet}} \quad (7)$$

where ω_e is the difference between the commanded and estimated angular velocity and a_{jet} is the total available rotational acceleration from the jets.

If the input rate to the controller is not higher than the critical value, a pseudo-auto mode is employed where phase plane logic is used to compute the jet firings. Given an input, δ , to the hand controller, the commanded angular velocity, ω_c , is given by a linear quadratic scaling law as

$$\omega_c = \frac{MCR}{40} \operatorname{sgn}(\delta) \left[|\delta| - 2 + \frac{(|\delta| - 2)^2}{2} \right] \quad (8)$$

Here MCR is the maximum commanded rate for the vehicle which is chosen to be 20 deg./s in normal scaling and 4° per sec in fine scaling. Fine scaling provides a finer control capability necessary for precise maneuvering as in the docking with the CSM after lunar ascent when the inertias are also much smaller.

E. Autonomous Rendezvous and Capture (AR&C)

AR&C capabilities form a strategic need for various space missions. These needs arise for ferrying supplies to and from the international space station, for planned crewed and un-crewed missions to moon and Mars, autonomous retrieval of expensive satellite hardware in space, repair and maintenance of orbiting satellites and so on [11]. AR&C GN&C requirements assume that the spacecraft designated as the chaser vehicle has translational as well as rotational maneuvering capabilities. The spacecraft that is designated as the target vehicle is assumed to have rotational maneuvering capabilities alone. For defining its GN&C needs, the AR&C flight phase is divided in three sub-phases. The flight phase that gets the chaser spacecraft within 40 km of the target spacecraft is defined as phasing. From a range of 40 km to 100 m, the flight phase is defined as proximity operation and the final flight phase from 100 m to capture is defined as the terminal phase.

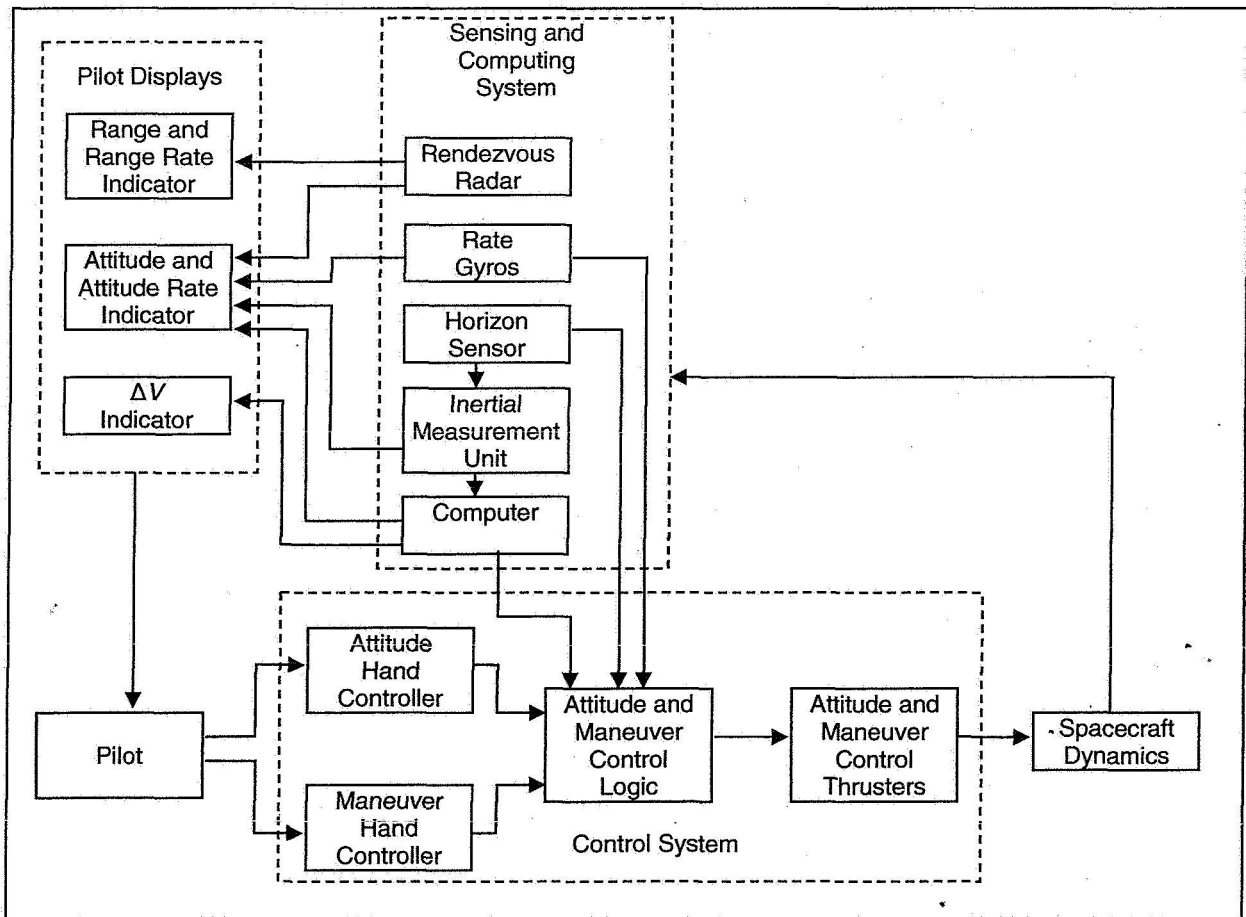


Figure 12: Gemini Rendezvous and Capture GN&C Architecture¹¹

Historically, the earliest rendezvous and capture technology was developed in the Gemini program in the 1960s. The Gemini spacecraft acted as the chaser vehicle and a modified Agena booster second stage was used as the target

vehicle. This program served as a testing precursor to the Apollo program that needed the capability for the docking of the LEM to the CSM in the lunar orbit. Figure 12 illustrates the GN&C architecture used for the astronaut-in-loop rendezvous and capture. While not autonomous in nature, the architecture illustrates various elements of the autonomous GN&C architecture.

The astronauts were provided sensor measurements of range, range rate, attitude and attitude rate. These indicators were used to control the Gemini range and attitude using hand controllers. The phasing operations were performed based on inputs from the ground controllers. The rendezvous radar was used for proximity operations where it provided the target range, range rate and relative bearing. Based on this information, the guidance algorithm in the digital guidance computer computed the incremental velocity needed for rendezvous. The horizon sensor was used for updating the IMU which provided the vehicle attitude and the rate gyros provided the attitude rate. All this information was displayed to the pilot who adjusted the vehicle attitude to point the thrust vector in the appropriate direction for achieving the incremental velocity. The rendezvous radar became ineffective in terms of the required range accuracies for the terminal phase where visual observations were used for navigating the vehicle. A similar principle was used for the Apollo mission as well as the docking of the space shuttle with the international space station and the Mir station. The only differences between these missions correspond to the use of more sophisticated radar systems and visual sensors for vehicle navigation.

The earliest AR&C system was demonstrated by the two un-crewed Russian Cosmos spacecraft in the 1960s. The GN&C architecture was similar to the one illustrated for the Gemini missions, however, the on-board computer provided autonomous guidance and control in place of the pilot or the mission control. This AR&C capability has been used for successive Russian missions involving the un-crewed Progress spacecraft and the crewed Soyuz spacecraft. The pilot or the mission control remains in a supervisory role and can take over control whenever the need be. Thus these AR&C missions always had the remotely piloted option. The first AR&C system demonstration between spacecrafts without the remote piloting capability was illustrated by the Japanese Engineering Test Satellite (ETS) VII [12]. This satellite consisted of both the chaser and the target components that were launched together as the ETS VII. At the start of the test, the chaser and the target were commanded to separate and then autonomously rendezvous and dock.

10. AR&C Guidance

The AR&C guidance algorithm computes the rendezvous trajectory for the chaser spacecraft in the face of several practical constraints. Lighting constraints involve the relative position of the sun that can play an important role in the terminal phase when video sensors are used for navigation. Plume impingement constraints correspond to having the chaser approach attitude such that any impingement of the plume from the chaser's thrusters onto the target vehicle is minimized. Similarly the docking port of the target vehicle may not be in the direction of its orbital velocity or may even be out of the orbital plane. There can also be other pointing and timing constraints. Along with these constraints, the computed rendezvous trajectory also needs to be optimized with respect to the fuel utilization.

The rendezvous trajectories are computed and analyzed in the local vertical local horizontal (LVLH) reference frame centered at the target vehicle. In this reference frame, the positive x-axis is along the orbital velocity and this direction is also known as the V-bar direction. The positive z-axis is pointed towards the center of the earth and this direction is also known as the R-bar direction. The positive y-axis is perpendicular to the orbital plane to form a right handed coordinate system. The relative motion between the chaser vehicle and target vehicle is analyzed by linearizing the nonlinear trajectory equations that are based on the Keplerian dynamics. These linearized equations are known as the Hills-Clohessy-Wiltshire (HCW) equations.

$$\begin{aligned}\ddot{x} - 2\omega\dot{z} &= a_x \\ \ddot{y} + \omega^2 y &= a_y \\ \ddot{z} + 2\omega\dot{x} - 3\omega^2 z &= a_z\end{aligned}\tag{9}$$

These equations are integrable and therefore provide solutions for velocity increments for moving between two points in space. Let the vectors \mathbf{r} and $\dot{\mathbf{r}}$ correspond to the relative position and velocity of the chaser vehicle with respect to the target vehicle.

$$\begin{aligned}\mathbf{r} &= [x \quad y \quad z]^T \\ \dot{\mathbf{r}} &= [\dot{x} \quad \dot{y} \quad \dot{z}]^T\end{aligned}\tag{10}$$

If \mathbf{r}_0 and $\dot{\mathbf{r}}_0$ correspond to the relative position and velocity of the chaser at time $t = 0$, and if there are no accelerations on the chaser, then the chaser vehicle position and velocity at a later time t is given by,

$$\begin{aligned}\mathbf{r}_t &= \Phi_{rr}(t)\mathbf{r}_0 + \Phi_{rv}(t)\dot{\mathbf{r}}_0 \\ \dot{\mathbf{r}}_t &= \Phi_{rv}(t)\mathbf{r}_0 + \Phi_{vv}(t)\dot{\mathbf{r}}_0\end{aligned}\tag{11}$$

Here $\Phi_{rr}(t)$, $\Phi_{rv}(t)$, $\Phi_{vr}(t)$ and $\Phi_{vv}(t)$ correspond to the respective transition matrices that are functions of the orbital angular velocity, ω and the time interval t [13]. Knowing the initial position, the final position, and the final time, the value for the initial velocity that the chaser spacecraft needs to have can be computed as

$$\dot{\mathbf{r}}_0^+ = \Phi_{rv}(T)^{-1} [\mathbf{r}_T - \Phi_{rr}(T)\mathbf{r}_0]\tag{12}$$

If the spacecraft has an initial velocity, $\dot{\mathbf{r}}_0^-$ at time $t = 0$, then this provides the velocity increment that needs to be given to the spacecraft at time $t = 0$ for it rendezvous with the target at time T .

$$\Delta v = (\dot{\mathbf{r}}_0^+ - \dot{\mathbf{r}}_0^-)\tag{13}$$

Also knowing this velocity allows computation of the velocity that the spacecraft will have at time T . Since this velocity needs to be zero for docking, this corresponds to a velocity increment that needs to be given at time T given by

$$\Delta v_T = -\dot{\mathbf{r}}_T = -[\Phi_{rv}(T)\mathbf{r}_0 + \Phi_{vv}(T)\dot{\mathbf{r}}_0^+]\tag{14}$$

This trajectory is known as the classical two-impulse scheme that is directed along the straight line connecting the location of the chaser spacecraft at time $t = 0$ and the target spacecraft even though the actual trajectory of the target spacecraft does not coincide with this straight line. If this straight line is along the x-axis, then this scheme is termed as the V-bar approach while if it is along the y-axis, then it is termed as the R-bar approach. This classical two-impulse scheme can be generalized to a multi-impulse scheme where this straight line is divided into several smaller sections and a two impulse scheme for each of these sections. Ref. [13] presents the equations for such a multiple impulse scheme for approach and departure along any plane not restricted to the orbital plane as well as circling the target spacecraft where this circle is divided into multiple straight segments. The two-impulse scheme is then applied to each of these individual segments.

The Glideslope scheme is a specialized scheme that was developed for the space shuttle [14]. Glideslope is defined as the straight line path approach to the target spacecraft corresponding to a constant angle with respect to the V-bar and the R-bar directions. Let r be the magnitude of the relative position vector \mathbf{r} . The Glideslope scheme was designed by transforming the HCS equations into polar coordinates. It was illustrated that by canting the thrusters on the two ends of the shuttle by different angles α and β results in a solution for which the shuttle moves along the particular glide-slope given by an angle θ such that the magnitude, r , of the relative position vector follows second order dynamics whose damping ratio and natural frequency are a function of the angles, α , β and θ . So the relative range rate decays exponentially minimizing the thrusting required for slowing down the chaser near the target thereby reducing plume impingement on the target. The Glideslope scheme provided a straightforward approach for shuttle terminal guidance. However this approach leads to a high usage of fuel during the maneuver.

Ref. [15] illustrates a predictive control based guidance algorithm that minimizes the fuel usage while explicitly satisfying all the different constraints of this problem. While it provides much better results than the Glideslope approach, it is a computationally-intensive algorithm.

All of the AR&C guidance algorithms discussed above provide the incremental velocity needed for the chaser spacecraft to successfully rendezvous with the target spacecraft. This incremental velocity is taken as an input by the spacecraft steering architecture as in the one illustrated in figure 3, which, then computes the desired body attitude and body attitude rates and provides them to the control system.

11. AR&C Navigation

AR&C navigation algorithms are dependent on the sensor packages used for the mission. Apart from the specialized sensors, the AR&C phase also uses the IMU for inertial acceleration and attitude measurements. As in the other flight phases, FDI algorithms for selection of the particular IMU measurements are used and the selected sensor measurements are filtered using the extended Kalman filter.

Given the required range and range rate accuracies, the AR&C systems use different sensors in their different flight sub-phases. This choice is also dependent on the nature of the mission as to whether the AR&C requirement is in a Low Earth Orbit (LEO) where GPS can be used or in lunar orbits or Mars missions or earth orbits beyond GPS altitudes. For LEO, the phasing flight phase can use IMUs or GPS or combination of both (GPS/INS) upto around 7 km from the target spacecraft [16]. For proximity operations, relative GPS can be used that can provide upto 1 m accuracy in its range estimation. Precise navigation algorithms for relative GPS operation have been designed that use a 19 state Kalman filter for providing range and range-rate information [17]. The relative GPS can be used upto 100 meters from the target vehicle. For the terminal phase operation, the Video Guidance Sensor (VGS) can be used that can provide range accuracies upto 0.3 cm, range-rate accuracies upto 0.3 cm/s and relative attitude accuracies upto 0.25 degrees [18]. The VGS uses vision processing algorithms for relating the information obtained from the video camera to range, range-rate and relative attitude.

Beyond LEO operation, IMUs and star trackers can be used for inertial navigation and attitude estimation during the phasing flight phase. For proximity operations, there are a couple of sensor options that give similar accuracies. The trajectory control sensor (TCS) was developed for the shuttle-Mir docking capability [19]. Like VGS it has a laser that is reflected off the retro-reflectors attached to the target and is used to generate range range-rate and bearing information relative to the target. It provides an accuracy of 3 cm and 3 cm/s for the relative range and range-rate. This accuracy is sufficient for proximity operations from about 1.5 km upto 100 m. Another sensor that can be relevant for the proximity operations is the autonomous formation flying sensor [20-21]. As the name suggests, this sensor was developed for autonomous formation flying of satellites separated by distances of hundreds of meters but which require accuracies of a few centimeters. This sensor is based on GPS technology applicable in deep space with or without the network of GPS satellites. The measurements from these sensors will be filtered using extended Kalman filters. For terminal phase operations, the VGS can be used as in LEO.

Thus the navigation algorithms involve various forms of the extended Kalman filters for different sensors.

12. AR&C Control Architecture

The AR&C control architecture remains similar to the TVC architecture illustrated in figure 5. Similarly the terminal phase of the AR&C phase uses reaction jets for attitude control which use the phase plane control laws as illustrated in figure 6.

III. GN&C Block-sets

Surveying the GN&C problems in some of these flight phases brings out the requirements for the GN&C database needed for designing the PAPA component. This database can be classified in terms of different block-sets.

13. Guidance Block-set

The guidance algorithms in the different flight phases reviewed consist of

- ❖ algorithms supporting thrust vector control where the required velocity for the maneuver is computed. The required velocity vector and the estimated vehicle velocity vector provide the velocity to be gained vector. The guidance algorithm then commands an angular velocity for aligning the thrust vector with the velocity to be gained vector. This logic is used in the ascent/boost phase for thrust vectoring of the SRBs, and the

shuttle main engine, in the on-orbit phase for thrust vectoring of the shuttle OMS engines, in the Apollo lunar descent and approach phases for thrust vectoring of the descent engine as well as for all autonomous rendezvous and capture operations. The differences between the algorithms in these phases lie in their particular trajectory design constraints.

- ❖ earth entry phase algorithm where a reference lift/drag profile is chosen and the guidance algorithm provide open-loop and closed loop angle of attack and bank angle commands for maintaining this drag profile. This algorithm is general enough to be applicable for any atmospheric planetary entry.
- ❖ earth boost phase algorithm, when a table-lookup based logic is used for providing the commanded values of angle of attack and side slip as a function of the vehicle velocity
- ❖ steering algorithm that smoothes the commands given by the guidance algorithm and provide desired commands to the control architecture.

These algorithms consist of a part that is specific to the particular flight phase and a part that shares a commonality across different phases. For algorithms supporting TVC in various flight phases, the incremental velocity calculation can be specific to each phase. Computing the incremental velocity usually involves a constrained optimization and can be implemented as part of the flight specific code. However once ΔV is computed, the same elements consisting of computation of commanded body attitude and attitude rate and the steering architecture are employed by the guidance architecture to provide the desired body attitude and attitude rates to the control system. Depending on the phase, the steering function chooses its natural frequency and damping ratio to provide consistent response. A guidance block-set can be created that is capable of implementing these phase independent algorithms.

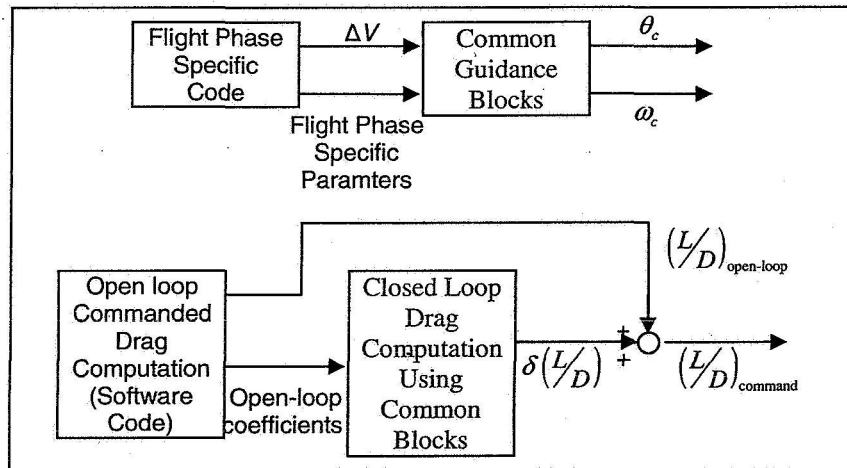


Figure 13: Decomposing guidance algorithms into flight phase specific and common blocks

For the earth entry phase, guidance provides a commanded L/D profile that consists of an open loop component and a closed loop component as illustrated in Eqs. (5-6). The open loop component can be computed with a software code. However the closed loop component shows similarities with PID control logic and can therefore be implemented using the common block set.

14. Navigation Block-set

The navigation algorithms in the different flight phases reviewed consist of

- ❖ state estimation algorithms that take inputs from different sensors to provide the state of the vehicle. These consist of the extended Kalman filter like algorithms that take inputs from IMUs and rate gyros. Vehicle attitude is also given by sensors such as the star trackers, sun and horizon sensors.

- ❖ sensor failure detection and identification algorithms that compare the inputs of redundant sensors and provide the appropriate sensor input to the estimation algorithms.
- ❖ vision processing algorithms that can compute relative range, range rate and attitude of the target spacecraft during AR&C flight phase
- ❖ general digital signal processing algorithms that act as static filtering algorithms for various sensors.

Some of these algorithms such as the vision processing algorithms are software intensive and can be coded accordingly. However the others can be built into a block-set library. A navigation block-set can therefore be created that is capable of implementing these algorithms.

15. Control Block-set

The control algorithms in the different flight phases reviewed consist of

- ❖ the PID control algorithm that is implemented in the shuttle earth boost phase for commanding the SRB deflections and in the shuttle entry phase for commanding the aerodynamic surfaces.
- ❖ phase plane logic that is used to fire the reaction jets in the on-orbit phase, in lunar descent and ascent phases as well as in the earth entry flight phase.
- ❖ thrust vector control laws that consist of digital compensation filters as used in the on-orbit maneuvers and in the lunar descent phase.
- ❖ manual control laws such as the rate command-attitude hold algorithm implemented for the LEM manual attitude control design.

Apart from the control algorithms, another important component of the control architecture consists of the control allocation algorithms that are used to distribute the command given by the control algorithm to the different actuators such as the reaction jets. Similar to the control allocation is the control blending logic that is used for distributing control between different function control actuators such as distributing the control authority between reaction jets and aerodynamic surfaces in the entry phase or the reaction jets and the descent engine in the lunar descent phase.

A control block-set can be created that implements these control algorithms as well as the control allocation and blending algorithms.

16. Coordinate Block-set

One of the most important functional forms in implementing any of the GN&C architectures is coordinate transformation. The space shuttle and the Apollo missions use several different coordinate systems such as the body axes system, earth-fixed system, the inertial system, the LVLH system, the lunar-fixed system, the lunar approach guidance coordinate system and so on. Apart from the different axes systems there are also exist different ways in which the orientation can be represented such as euler angles, quaternion, Gibbs vector and so on. Each of these may need to be implemented for certain flight phases. The GN&C database therefore needs to include a coordinate block-set that provides support for all its algorithms in various flight phases.

Figure 14 illustrates the design process of assembling the flight phase specific GN&C system using the block-sets enumerated above.

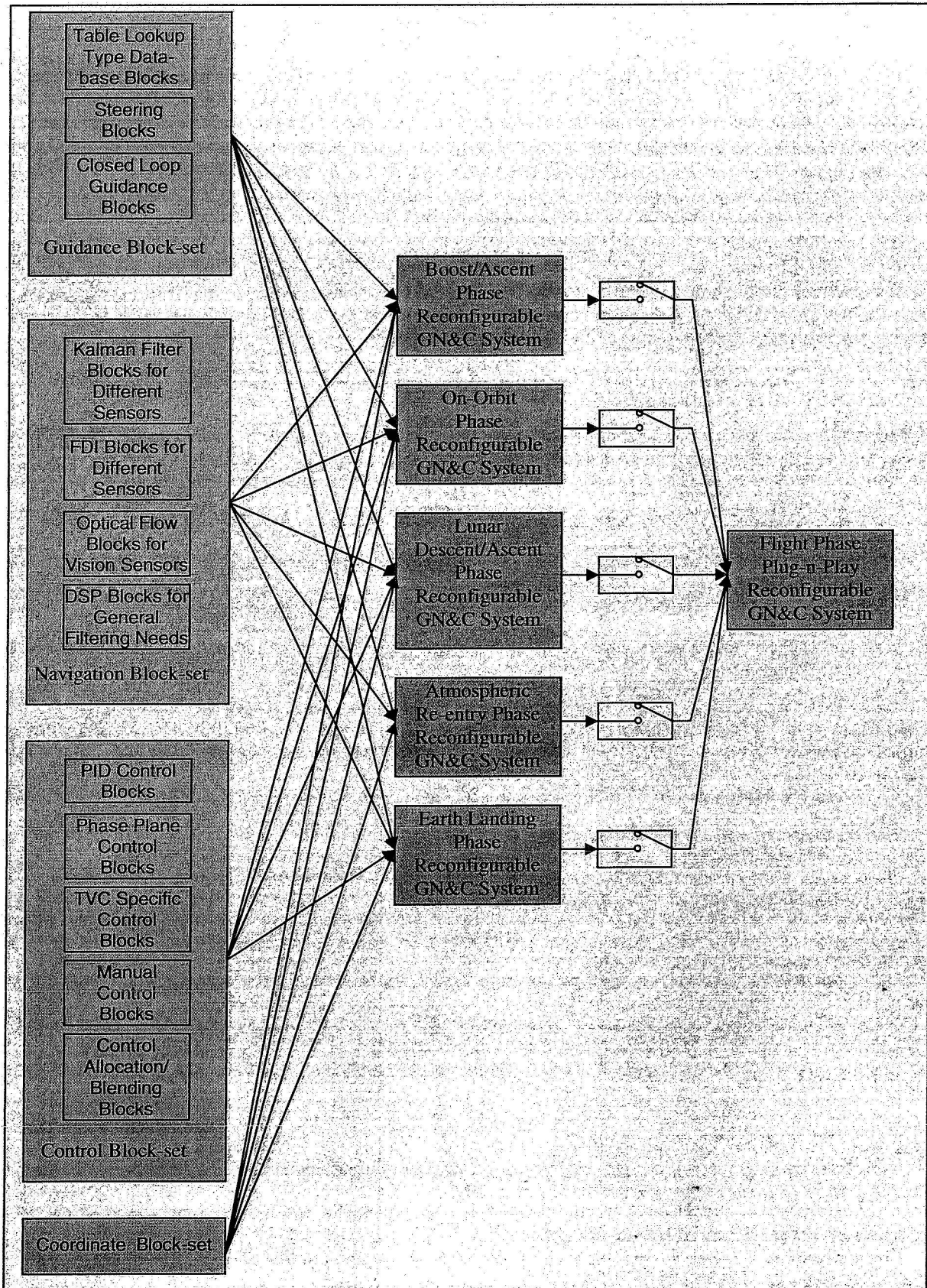


Figure 14: Plug-n-Play Reconfigurable GN&C System Design Flow

IV. Conclusions

In this survey, the Apollo and Space Shuttle literature has been reviewed towards understanding the spacecraft GN&C requirements. This survey is guided towards creating a block-set of the GN&C algorithms that can be realized on a plug-n-play reconfigurable hardware platform. Even though the problem of designing these GN&C algorithms is a complex problem given the various flight phases, a certain set of commonality can be noted between these algorithms. This can make it possible to reduce them to a level of abstraction corresponding to a minimal set that can be stored and utilized for all the flight phases. The survey has concentrated on the Apollo and Space Shuttle literature mainly because of the high technology readiness level (TRL) of their algorithms. However, since the onset of these missions, several robust and high-fidelity algorithms have been applied for several other aerospace problems. The literature since these missions therefore needs to be carefully reviewed to include these recent algorithms. An example is the dynamic inversion control architecture that has been well-established. Apart from realizing this minimal set of basic algorithms, adaptive implementation of these algorithms needs to be studied to provide the soft fault-tolerance functionality. This forms the direction of the future work.

References

- ¹McHenry, R. L., Brand, T. J., Long, A. D., Cockrell, B. F., and Thibodeau III, J. R., "Space Shuttle Ascent Guidance, Navigation, and Control," *Journal of the Astronautical Sciences*, Vol. XXVII, No. 1, pp. 1-38, Jan.-Mar., 1979.
- ²Martin, D. T., Stevens, R. F., O'Brien, R. M., and Rice, A. F., "Saturn V Guidance, Navigation, and Targeting," *Journal of Spacecraft*, Vol. 4, No. 7, July 1967.
- ³Cox, K. J., "Digital Flight Control Software Design Requirements: Chapter 7: Ascent/Boost Phase," JSC - 07759, NASA Space Shuttle Program Working Paper, July, 1973, Houston, Texas.
- ⁴Cox, K. J., "Digital Flight Control Software Design Requirements: Chapter 6: On-orbit Phase," JSC - 07759, NASA Space Shuttle Program Working Paper, July, 1973, Houston, Texas.
- ⁵Harpold, J. C., and Graves, C. A., "Shuttle Entry Guidance," *Journal of the Astronautical Sciences*, Vol. XXVII, No. 3, pp. 239-268, July-Sept. 1979.
- ⁶Cox, K. J., "Digital Flight Control Software Design Requirements: Chapter 5: Entry Phase," JSC - 07759, NASA Space Shuttle Program Working Paper, July, 1973, Houston, Texas.
- ⁷Widnall, W. S., "Lunar Module Digital Autopilot," *Journal of Spacecraft and Rockets*, Vol. 8, No. 1, 1971, pp. 56-62.
- ⁸Battin, R. H., "Astronautical Guidance," McGraw-Hill Book Company, Inc., New York, 1964.
- ⁹Klumpp, A. R., "A Manually Retargeted Automatic Descent and Landing System for LEM", *AIAA/JACC Guidance and Control Conference*, Seattle WA, August 15-17, 1966, pp. 713-734, available online as: R-539, Mar. 1966, M.I.T Instrumentation Lab., Cambridge, MA, <http://hrst.mit.edu/hrs/apollo/public/archive/R-539.pdf>.
- ¹⁰Stengel, R. F., "Manual Attitude Control of the Lunar Module," *Journal of Spacecraft and Rockets*, Vol. 7, No. 8, Aug. 1970, pp. 941-948.
- ¹¹Polites, M. E., "Technology of Automated Rendezvous and Capture," *Journal of Spacecraft and Rockets*, Vol. 36, No. 2, March-April, 1999, pp. 280-291.
- ¹²Kawano, I., Mokuno, M., Kasai, T., and Suzuki, T., "Result of Autonomous Rendezvous Docking Experiment of Engineering Test Satellite-VII," *Journal of Spacecraft and Rockets*, Vol. 38, No. 1, January-February 2001, pp. 105-111.
- ¹³Hablani, H. B., Tapper, M., and Dana-Bashian, D., "Guidance Algorithms for Autonomous Rendezvous of Spacecraft with a Target Vehicle in Circular Orbit," *AIAA Paper 2001-4393*, Proceedings of the *AIAA Guidance, Navigation and Control Conference*, 6-9 August 2001, Montreal, Canada.
- ¹⁴Pearson D. J., "The Glideslope Approach," *AAS Paper 89-162*, *Advances in the Astronautical Sciences*, Aug. 1989, pp. 109-123.
- ¹⁵Richards, A., and How, J., "Performance Evaluation of Rendezvous using Model Predictive Control," *AIAA Paper 2003-5507*, Proceedings of the *AIAA Guidance, Navigation and Control Conference*, 11-14 August 2003, Austin, Texas.
- ¹⁶Bunn R. L., "GPS/INS Autonomous Navigation Systems for Space Applications," *AAS Paper 98-016*, *Advances in the Astronautical Sciences*, February 1998.
- ¹⁷Upadhyay, T. N., Cotterill, S., and Deaton, A. W., "Autonomous Reconfigurable GPS/INS Navigation and Pointing System for Rendezvous and Docking," *AIAA Paper 1992-1390*, *Space Programs and Technologies Conference*, Huntsville, AL, Mar. 24-27, 1992.
- ¹⁸Howard R. T., Bryan T. C., Book, M. L., and Dabney, R. W., "The Video Guidance Sensor: A Flight Proven Technology," *AAS Paper 99-025*, *Advances in the Astronautical Sciences*, 1999.
- ¹⁹"User's Manual for the Trajectory Control Sensor Command and Display User Interface 6/5/97 Revision 5.8," NASA Johnson Space Center, JSC 27870, Houston, TX, 1997, p. 4.
- ²⁰Lau, K., Lichten, S., Young, L., and Haines, B., "An Innovative Deep Space Application of GPS Technology for Formation Flying Spacecraft," *AIAA Paper 1996-3819*, July 1996.

²¹Lichten, S., Purcell, G., Wu, S., Kuang, D., Srinivasan, J., Young, L., and Lau, K., "Autonomous Formation Flyer Sensor Technology Development," AAS Paper 98-062, *Advances in the Astronautical Sciences*, Feb. 1998.