Avoiding Human Error in Mission Operations – Cassini Flight Experience

Thomas A. Burk*
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109

Operating spacecraft is a never-ending challenge and the risk of human error is ever-present. Many missions have been significantly affected by human error on the part of ground controllers. The Cassini mission at Saturn has not been immune to human error, but Cassini operations engineers use tools and follow processes that find and correct most human errors before they reach the spacecraft. What is needed are skilled engineers with good technical knowledge, good interpersonal communications, quality ground software, regular peer reviews, up-to-date procedures, as well as careful attention to detail and the discipline to test and verify all commands that will be sent to the spacecraft. Two areas of special concern are changes to flight software and response to in-flight anomalies. The Cassini team has a lot of practical experience in all these areas and they have found that well-trained engineers with good tools who follow clear procedures can catch most errors before they get into command sequences to be sent to the spacecraft. Finally, having a robust and fault-tolerant spacecraft that allows ground controllers excellent visibility of its condition is the most important way to ensure human error does not compromise the mission.

Acronyms

AACS	=	Attitude and Articulation Control Subsystem
AFC	=	AACS Flight Computer
AU	=	Astronomical Unit
CDS	=	Command and Data Subsystem
FSDS	=	Flight Software Development System
FSW	=	Flight Software
HGA	=	High Gain Antenna
IRU	=	Inertial Reference Unit
ITL	=	Integrated Test Laboratory
IVP	=	Inertial Vector Propagation
IVT	=	Inertial Vector Table
KPT	=	Kinematic Predictor Tool
OTM	=	Orbit Trim Maneuver
RAM	=	Random Access Memory
RCS	=	Reaction Control System
ROM	=	Read-Only Memory
RWA	=	Reaction Wheel Assembly
SID	=	Star Identification
SRU	=	Stellar Reference Unit
SSR	=	Solid State Recorder

I. Introduction

Avoiding human error in spacecraft operations is a vital aspect of mission success, but is often overlooked when a mission's quantitative requirements and capabilities are established. The number of space missions that have been degraded, or ended, due to human error is very sobering. The Clementine mission after it surveyed the Moon, the Near Earth Asteroid Rendezvous (NEAR) mission, the Mars Climate Orbiter, the Yohkoh Solar Observatory, the Solar Heliospheric Observatory (SOHO), Mars Global Surveyor, these (and many other missions) were significantly affected by human error during operations.

The Cassini attitude and articulation control system (AACS) spacecraft operations team is especially cognizant of the need to minimize the effects of human error. Cassini was launched on October 15, 1997 and is nearing completion of its 15th year in space. For any mission of long duration, ground engineers or controllers will sometimes make mistakes, perhaps just through complacency. The goal is to establish a mission operations system – meaning an operations team and their tools and processes – that identifies and corrects mistakes before any incorrect commanding reaches the spacecraft.

A fundamental way to reduce the risk of human error is a robust and fault-tolerant design of the spacecraft itself. For example, the Cassini attitude control flight computer (AFC) is programmed to reject a command to power off a

* Cassini Attitude Control Product Delivery Manager, Guidance and Control Section, M/S 230-104, 4800 Oak Grove Dr., Pasadena, California 91109, thomas.a.burk@jpl.nasa.gov

required resource (for example, the prime inertial reference unit). Another example is a design that provides good visibility of spacecraft states. If an anomaly occurs onboard, the spacecraft is designed to go into safe mode and thus protect itself. A common problem on some missions is the failure to understand what really happened to cause the safe mode situation. More than once, on some spacecraft, ground controllers have made a mild fault much worse by inappropriate ground commanding. On Cassini, a large dedicated buffer in the AFC, called the fault protection log, contains the entire history, with time tags, of any onboard fault response. This kind of visibility greatly aids in understanding the root cause of a spacecraft anomaly.

An important way to track down and fix a human error is to test a spacecraft command sequence on the ground before it is uplinked to the spacecraft. A ground simulator, whether it is a laboratory with actual flight computers and avionics, or a pure software simulation, will normally detect an error in a command sequence. On one mission, such a test was performed and a violation in the form of a plot was generated, but ground operators missed the violation because the simulation output did not clearly highlight the violation. The sequence was uplinked and the spacecraft went into safe mode as a result.

There are innumerable ways human error can creep into ground control operations, so most space missions keep an updated database of anomalies for the life of the mission – from early in flight software and hardware development pre-launch, all the way through the end of the mission. This database documents not just flight anomalies, but operations and ground software anomalies too. If a ground simulator detects a violation, but an engineer misreads it, the problem needs to be documented in the ground software anomaly database and corrected in a timely fashion. The anomaly should not be closed unless some direct action is taken to rectify it.

Some mistakes occur because the ground controller did not follow a procedure correctly. Others occur because there is no documented procedure at all. An experienced engineer will go on vacation and not have an adequately trained backup. A new engineer will come on board and make a mistake that no one detects. Adequate training, having an experienced backup to check all flight products, regularly scheduled peer reviews of each command sequence product, and not rushing through a verification process to meet a deadline, all these and more have to be part of standard operating procedure. Attention to detail, flying what you test and testing what you are about to fly, these are not just truisms but are the only real way to successfully operate an expensive spacecraft – and all spacecraft are expensive relative to doing these many, but straightforward, checks.

This paper discusses how these processes, and the spacecraft itself, were designed to help avoid risks to mission success due to human error throughout the lifetime of the Cassini mission. Significant real-life examples involving the Cassini attitude control operations team will be discussed including lessons learned. Automating as many checks and tests as possible are vital to mission success. No matter how many checks, though, it remains true that engineering judgment – having experienced well-trained engineers who have as much knowledge at their fingertips as possible – is always vital. An engineer should not just "turn the crank" like a drone. Nothing breeds success like thoughtful and engaged engineers, open to new ideas, doing their job carefully and with adequate time to do a good job.

II. Background

The Cassini mission at Saturn has had three phases. The nominal mission was designed prior to arrival at Saturn in July 2004. This nominal mission tour extended from June 2004 through June of 2008. One highlight of this 4-year "Tour" of the Saturn system was the successful release of the European Space Agency Huygens Probe on December 24, 2004 and its subsequent descent to the surface of the Titan atmosphere on January 14, 2005.

An extended mission (Equinox) was funded and spanned two years of exploration through September of 2010. The final phase of the mission, called Solstice, is currently executing and, if the spacecraft remains healthy, will continue through mid-September of 2017. Consumables are expected to be nearly depleted by that time so the plan is to adjust the spacecraft trajectory so that Cassini will steeply enter the atmosphere of Saturn, ending the mission.

Cassini is a 3-axis stabilized spacecraft with an 11-m magnetometer boom and three 10-m Radio and Plasma Wave Science antennas. There are twelve science instruments mounted on the spacecraft and there is no scan platform. Most science instruments require the whole spacecraft to be slewed in order to track a target of interest. The Cassini High Gain Antenna (HGA) is mounted on top of the spacecraft stack and points parallel to the spacecraft –Z body axis (see Fig. 1). The HGA is used as a Sun shield and also as a shield during hazardous Saturn ring plane crossings. The HGA provides downlink to the NASA Deep Space Network (34m and 70m stations in Goldstone, California; Madrid, Spain; and Canberra, Australia) at up to 166 kbps for science data at 9 AU. The engineering allocation of the telemetry is 1896 bps. Cassini has 2 backup Low-Gain Antennas for emergency communication.

On Cassini, inertial attitude knowledge sensors include a star tracker, known as a stellar reference unit (SRU), and an inertial reference unit (IRU). Both the SRU and IRU have identical backup units. Star identification (SID) flight software turns the star data into a 3-axis attitude quaternion based on this data every 5 seconds. The star-derived attitude estimate is sent to the attitude estimator flight software which uses an extended Kalman-Bucy filter[1]. In between star updates, including during periods where SID is "suspended" due to Saturn, the Rings, or other bright bodies obscuring the SRU field-of-view, the spacecraft's attitude is propagated using IRU data only.

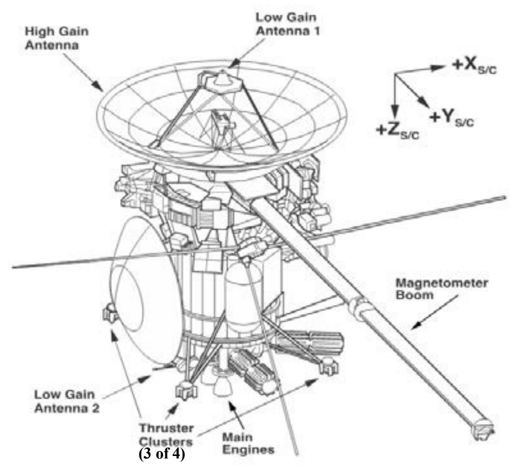

Figure 1. Cassini Configuration at Saturn Arrival

Cassini has dual redundant IRUs each containing four Hemispherical Resonator Gyroscopes (HRGs). The HRG is a rotation sensor that uses a resonating structure – analogous in some ways to the bowl of a wine glass – to determine the spacecraft rotation rate about an axis. The prime IRU on Cassini has been powered on continuously since launch, demonstrating outstanding durability. Gyro scale factor errors and misalignments are calibrated onboard every year.

Attitude control is provided by either 3 reaction wheel assemblies (RWAs) or by 8 hydrazine thrusters. There are 2 branches of 8 thrusters each. The A-branch was used from launch until March of 2009, but degraded thrust in two A-branch thrusters led ground controllers to switch to using the B-branch since that time. Mixed-branch thruster operation is also supported. Four of the 8 thrusters fire parallel to the spacecraft –Z body axis and provide attitude control about the spacecraft ±X and ±Y body axes. The other 4 thrusters fire as couples in the ±Y spacecraft body direction and provide control about the ±Z axis (Fig. 2).

For precise and stable pointing, the Cassini spacecraft uses 3 strapped-down reaction wheel assemblies (RWAs). Each RWA spin axis is fixed in the spacecraft body frame but is not co-aligned with the spacecraft body axes. Together the three prime RWAs form an orthogonal set. Each RWA has a 34 Nms momentum storage capability and a motor capable of .165 Nm of torque. A fourth RWA is articulable and can thus be oriented to match any of the other three RWA's spin axis orientation. This 4th RWA was originally the backup RWA, but since July of 2003 it has been used as one of the three prime RWAs. Only three RWAs can be used for control simultaneously, so one RWA is normally powered off.

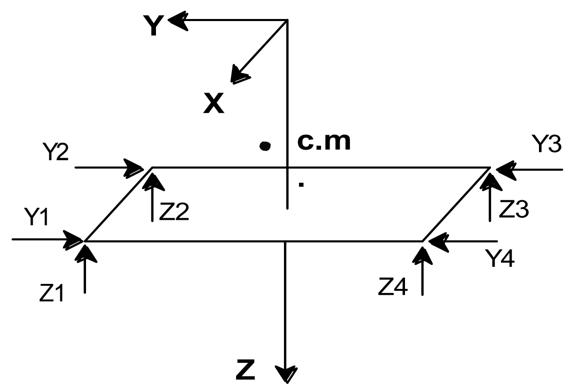

Figure 2. Cassini Thruster Configuration

The Cassini propulsion system includes the RCS system (4 co-aligned thrusters) for small (< 0.3 m/s) ΔV maneuvers and a bi-propellant main engine system for larger ΔV burns. Although there is a backup main engine, it has not been used in flight. Both main engines use two linear engine gimbal actuators to provide 2-axis thrust vector attitude control and the third axis (roll about the thrust vector) is controlled by the RCS thrusters. One or more ΔV maneuvers -- at Saturn these are called Orbit Trim Maneuvers (OTMs) -- are typically required for each targeted encounter with Titan (or another moon).

Cassini has dual redundant MIL-STD-1750A AFCs running Ada-compiled AACS flight software. The prime AFC is the controlling computer; the backup AFC has identical FSW and normally does minimal housekeeping. Each AFC communicates to peripherals (e.g. SRUs, IRUs, RWAs, thruster drivers) via dual redundant AACS data buses. AACS is considered a peripheral on the dual redundant Command and Data System (CDS) data buses. The prime and backup MIL-STD-1750A CDS flight computers, each with 512 KB of RAM storage, contain system fault protection algorithms and control all communication with the two 2-Gbit solid state recorders. The CDS computers store and execute all spacecraft command sequences transmitted from Earth. These sequences are lengthy sets of time-ordered commands that control the non-autonomous operation of the CDS and AACS flight computers as well as all the science instruments. All spacecraft pointing, data recording, playback, camera triggers, hardware on/off power commanding, etc. are designed on the ground, tested, uplinked, stored, and executed as part of onboard stored sequences.

The AACS flight software includes all the logic for attitude estimation, control, hardware management, and fault protection. Stored sequences uplinked to the spacecraft include all the time-ordered commands for individual science or engineering activities. Commanded pointing of the spacecraft is achieved by a combination of sequence commands and various flight software objects. One notable FSW object central to all pointing on Cassini is called Inertial Vector Propagation. FSW memory limitations do not allow complete and continuous S/C ephemeris to be stored onboard. Instead, all pointing is achieved using IVP vectors in command sequences, combined with onboard propagation of those vectors. This elegant approach[2] allows pointing command accuracy better than 40 µrad when pointing at any feature on Saturn, the Rings, on a moon, or any desired target. Onboard inertial vector propagation allows "tracking" or "target motion compensation" to occur automatically so that image smear is virtually eliminated, even when close to a moving body.

The Cassini Science teams are distributed at JPL and at other NASA or contractor sites as well as overseas. All their observations need to be integrated into a single entity – the background sequence. Sequence development, integration, and verification requires several months and is a collective responsibility of many teams that are building science observations, maintaining the scientific instruments, playing back the data to Earth, maintaining the attitude control subsystem flight hardware and software, and accomplishing all this within the capabilities – including consumables usage – of the spacecraft. Once a sequence has been integrated, tested and verified on the ground, it is uplinked to Cassini about 3 days before it is scheduled to begin execution. These sequences are called "background" sequences because they are active throughout many weeks of continuous operation. During XXM, these sequences are typically 7-10 weeks in duration.

Some activities, for example ΔV maneuvers which require late-breaking orbit determination knowledge, are uplinked while the background sequence is already executing onboard. These "real-time" sequences can execute in parallel with background sequences, and in fact the background sequence usually has 4-6 hour "gaps" (with the spacecraft typically quiescent and at Earth-point) so that a ΔV maneuver mini-sequence can be uplinked a few hours to a few days before the actual ΔV "burn".

III. Design the Spacecraft with Flight Operations in Mind

Human error in spacecraft operation is best minimized by robust and fault-tolerant spacecraft hardware and software design. Each component of the Cassini AACS subsystem defaults to a safe state upon any detected deviation, and recovery is aided by ample diagnostic information[3]. AFC bus communication watchdog timers are used as well as a heartbeat signal that AACS sends to CDS every second. Were a heartbeat loss to occur, or a peripheral watchdog timer to expire, the AFC or peripheral would be commanded to reset. All flight hardware is designed to reset to a safe state after any malfunction. To date, no autonomous resets of any AACS flight hardware have occurred on Cassini since launch. In 2010, the prime CDS flight computer reset while receiving and checking a real-time command sequence. This did lead to safe mode, but AACS performance remained nominal and high data rate ground communication was re-established in one hour.

Commands on Cassini are protected by bit error detection at every step from transmission through command sequence execution. Bus addresses are protected against errors so no peripheral will mistakenly act upon a command intended for another unit. Key hardware such as actuators, propulsion, and power are protected even from their own local bus interface units. This is done by requiring a correct sequence of actions, unlikely to be generated by accident, to achieve the protected function.

Cassini has vast hardware monitoring onboard so that the flight software and ground controllers have excellent visibility of not just commanded hardware states but also the hardware response. In many cases, hardware was added to Cassini explicitly for this purpose.

Cassini was built to allow exercising backup hardware without altering the prime operating configuration. No failure modes of the backup units can interfere with the operation of the prime hardware. The backup RWA can be exercised in RCS or in RWA control (in practice, this exercise is done every three months in RWA control). The backup sun sensor is powered on before, and validity checked by AACS engineers after, each hazardous Saturn ring plane crossing. The prime and backup star trackers are calibrated every year, as are the prime and backup inertial reference units. A single sequence command is all that is need to perform many of these calibrations.

Sequenced commands are high-level, goal-oriented commands that alert the flight software that an activity is starting. For example, a single command initiates an IRU or accelerometer calibration. A multi-revolution spacecraft slew for science fields and particle can be accomplished with just two commands: the first to command the rate and acceleration for the activity; the second to start the actual turn and tell the FSW the total magnitude and slew axis. The rest of the turn (profiling the acceleration, coasting, and deceleration phases) is handled by the flight software. Sequenced commands have built-in checks to ensure non-corruption and mode and range checks to ensure validity. Over 400,000 sequenced AACS commands have been uplinked and executed on Cassini since reaching Saturn.

None of these features were "add-ons" to an already-completed design. Each was built into the system early in spacecraft development. For example, the AACS flight system architecture included these and other robustness features at least three years before launch.

IV. Flight Operations are Sometimes Undervalued Compared to Design

Operating a complex spacecraft like Cassini is a significant challenge for engineers, especially given the inevitable aging of the spacecraft hardware after almost 15 years in flight. Just as challenging, is the need to maintain a team of experienced engineers to operate Cassini.

Operations engineers are often categorized differently than design engineers. Designers are crucial to putting together a quality spacecraft that is durable. Operations engineers are usually not experts in the design and may have limited understanding of flight hardware or software details. Design engineers tend to work with cutting-edge technology where new designs or approaches are encouraged, while operations people tend to "stick with what works". Designers are often labeled more "creative", while operators are sometimes considered more "turn the crank" users without much technical acumen.

However, real world experience suggests that actual designs can have many "traps" unless they are carefully informed by operational knowledge. A design that is easily broken or mishandled is a poor design. In spacecraft operations, it is the operators who come to know the actual performance of flight hardware and software. Operators on some space missions have to build a database of "idiosyncrasies" and troubleshoot around unwieldy designs. It can be hard to predict the behavior of some designs. A few spacecraft have onboard intelligent agent flight software that might override a commanded slew path, for example. But if real world experience demonstrates that the behavior of the agent cannot be routinely predicted, it may be of little value and can actually detract from stable operation.

One example of operations having to overcome a design flaw is the Huygens Probe Relay Doppler problem. During a probe relay S-Band demonstration test in February of 2000, more than two years after launch, it was found that the Doppler shift in the link from the Probe simulator on Earth to the receiver onboard Cassini caused the data signal to fall outside the bandwidth of the detector[4]. The relative motion of Cassini with respect to Titan would cause this Doppler shift, but the receiver design did not take this into account. Without a workaround, the Huygens Probe Relay would have been seriously degraded or lost completely. The workaround involved a major re-design of the Cassini trajectory in the first year of its mission at Saturn to reduce the Doppler shift during Probe Relay. The result was excellent data reception throughout the relay on January 14, 2005.

Technical knowledge is a key factor in successful mission operations, but the importance of human interaction – communication with colleagues – is demonstrably just as important. The international nature of the Cassini Huygens mission required a lot of careful communication to ensure that standards and conventions were always clearly understood by American and European engineering teams. Even so, the fact that the Probe Relay critical sequence required major integration by both teams still led to an oversight in the final sequence: key hardware on the Cassini orbiter was controlled by the European team, but the critical sequence was managed by the American team. An oversight led to some probe receiver hardware on Cassini not being powered on and half of the Huygens Probe images from its descent to Titan were not received by Cassini due to this error[5].

V. Cassini Integrated Test Laboratory (ITL)

All missions utilize a hardware integrated test laboratory for both flight hardware and flight software development and test. Flight spares of many flight hardware components provide outstanding fidelity and make the Cassini ITL an essential operational tool after launch. When combined with robust ground support equipment, the Cassini ITL is the perfect platform to test to-be-flown command sequences. The input products to ITL should be the actual products that will be uplinked, where possible. The ground telemetry system needs to be fully integrated into ITL so that ground operators get a preview of their visibility of the actual activities on the spacecraft.

All critical and first-time events should be tested in an integrated test laboratory. ITL should have easily accessible ways to extract telemetry, as well as environment "truth" simulation parameters. Standard ITL output products should also include fault protection logs, and records of detailed inputs used in simulation environment initialization.

Cassini always schedules ITL procedure walk-throughs so that both the ITL test team and the flight operations team have a common reference to follow for test initialization. Test initialization is especially important because an effective test must match the states of the actual spacecraft. This is especially true for flight software fidelity and as many subsystem parameters as possible. Test review must be a team responsibility and flight team reports on the outcome of each test need to supplement the ITL team output products. *A test not carefully evaluated provides little value-added*.

The ITL is also a great resource for in-flight anomaly evaluation too. To supplement the ITL, the Flight Software Development System (FSDS) is an "all software" simulator that was used to build and check out the flight software pre-launch[6]. It is used extensively in operations because it contains all the flight software (including fault protection), runs faster than real-time, and runs on an analyst's workstation. FSDS is used for detailed anomaly or engineering investigations, OTM testing, hardware-related fault insertion, and flight software development and regression testing. FSDS also contains updated models of Titan atmosphere and Enceladus plume models. *A key aspect of operational success on Cassini is the careful design of test cases using FSDS or ITL*[7]. Testing that emphasizes actual operational scenarios is especially valuable in tracking down human errors that could occur in flight.

VI. Flight Operations Training

Cassini operations engineers are responsible for maintaining the spacecraft, keeping it on course, testing and uplinking commands to achieve science and engineering objectives, monitoring telemetry, trending spacecraft performance, and responding to anomalies when and if they occur. How should spacecraft operations engineers be trained to perform the complex tasks that are regularly encountered during a mission?

The early Cassini spacecraft operations team was assembled several years before launch. One of the most valuable learning experiences that an attitude control operations engineer can have is participating in the actual spacecraft testing that occurs during pre-launch assembly, test, and launch operations (ATLO). Pre-launch is when the greatest number of flight hardware and flight software problems are encountered. This is the ideal stage for in-depth learning about the spacecraft. And it is knowledge, more than anything, that helps a spacecraft operations engineer do his/her job most effectively. Pre-launch is when an operations engineer can most easily interact with the cognizant flight hardware and flight software experts. Pre-launch is when most of the essential capabilities of the spacecraft are physically tested. It takes time and training to properly command and monitor a spacecraft. Ideally, this is best accomplished in parallel with the final integration and testing of the spacecraft itself pre-launch.

In the first days and weeks after launch, the operations team is usually supported by many of the cognizant flight hardware and flight software experts. On Cassini, by 30 days after launch, the flight operations team became the sole team responsible for the spacecraft, with flight hardware and flight software experts used thereafter as consultants rather than intimately involved in operational duties. However, continuing consultation with the "designers" is the only real way operations engineers can develop the expertise to operate the spacecraft wisely. It is essential to capture this expertise early, because designers will inevitably move on to other things.

The AACS operations team provides formal and informal tutorials to new AACS engineers to help them learn about the flight hardware and software, how the spacecraft is commanded and what telemetry is produced on a daily basis. He/she becomes acquainted with the ground software tools and processes to build and validate command sequences. She supports many meetings relating to science and engineering activities for a specific background sequence. He supports meetings on OTM preparation, approval and post-burn wrap-up. She monitors the actual OTM burn to develop a familiarity with real-time telemetry monitoring. He learns about ΔV maneuver execution error and how flight software parameter updates help minimize it. She learns under what conditions a fault protection response in the onboard flight software can become activated and autonomously issue commands onboard to recover after a hardware fault. He learns about how RWA health is optimized by judicious selection of RWA momentum commands at key intervals during a sequence.

A new AACS analyst learns about a key method of avoiding operational mistakes on Cassini – flight rules. Flight rules are operational limitations imposed by the spacecraft system design, hardware and software, violation of which would possibly result in spacecraft damage, loss of consumables, loss of mission objectives, loss and/or degradation of science, and less than optimal performance. Flight rules are an important way to capture the knowledge of both designers and lessons learned from previous operations on Cassini[8]. Flight operations engineers need to become steeped in the mission flight rules. They need to know not only what the flight rules are and why they exist, but also how they are checked. This is discussed further in the ground software section.

VII. Flight Operations Engineers – Knowledge, Care, Training, and Review

Human error is more likely when flight operations engineers do not have adequate tools. This is discussed more in the next section. Besides having fully capable tools, human error is more likely to creep in when an engineer:

1. Is not given enough time to complete the given task.
2. Does not adequately understand the task and its impact.
3. Does not take enough care in setting up or preparing for their task.
4. Does not follow a clear procedure.
5. Does not review his/her verification results carefully enough, or with others.

An example of knowledge deficit occurred when an AACS engineer designed an IRU calibration about one year after Cassini reached Saturn. To precisely estimate gyro scale factor errors during slews, good attitude estimation from star identification is essential. The AACS analyst mistakenly designed the slews for the calibration in a way that caused Saturn and Rings bright bodies to enter the SRU field of view. This caused star identification to be suspended and led to an inadequate gyro calibration.

A new flight rule was added to avoid the same mistake happening again. At the same time, the rule was explicitly added to ground software so an automatic check would detect the error in the future. Peer reviews were added as a standard review step for all future sequences. At the peer review, the entire AACS team (including cognizant FSW and fault protection leads) reviews all the verification results that an AACS lead analyst has done. The lead systems engineer for the sequence also participates. The lead AACS analyst reviews all his inputs and analysis for the sequence with the rest of the AACS team, including the AACS team lead. The informal nature of the meeting – about 6 to 8 people participate – is conducive to asking questions and allowing each engineer to learn in detail about issues that inevitably are relevant to their own lead responsibilities too. Peer reviews definitely help in finding errors.

Written procedures to document all the steps that an analyst should follow throughout the sequence development and verification process have been a great aid. On Cassini, procedures are living documents – currently kept on a Wiki – that are amended and clarified so that new analysts have a roadmap to help them follow the numerous steps needed to fully verify the command sequence.

Since absences or sickness can happen anytime, fully capable "backup" engineers shadow the lead AACS engineer for all background and OTM sequence design and verification. These backups are a crucial additional set of "eyes" because everyone at some point overlooks a parameter value or might miss a procedural step. Backup engineers independently perform the sequence verification procedure as a check on the lead analyst's work. The backup engineer is especially important when the lead AACS analyst for a sequence is relatively inexperienced.

The Cassini team also uses detailed checklists for sequence verification that includes sanity checks along with formal flight rules. A different checklist is used for OTMs or other real-time activities. Certain unique steps are required during final verification and a checklist ensures that each analyst remembers these key steps. Both the lead and backup engineers should participate in checklist completion. These checklists evolved from previous experience where an analyst may have made a mistake, so checklists especially alert an analyst to areas of potential human error that require extra care.

Important activities benefit from operational readiness tests (ORTs). These involve the full spacecraft engineering team, often the navigation team, and science and uplink teams. One or more faults are introduced into the ITL (which represents the spacecraft in this case) which is configured for the activity, and the team must figure out how to deal with the – to them unknown – anomaly, often with certain additional "surprises" thrown in. Perhaps a hardware anomaly is introduced and downlink is lost. Or one of more key engineers is removed from the operational process so that other engineers need to demonstrate their own abilities to safely recover. Each ORT has a "lessons learned" review so that the flight team's performance is critiqued and processes improved. ORTs are great for training new team members too.

Command approval meetings need to be scheduled so that all teams have adequate time to evaluate the actual flight products. Avoid any changes in the command files after verification and before uplink. Standardize steps like file creation time review and simple file naming conventions to ensure there is no last-minute confusion about the specific files to be uplinked.

VIII. Ground Software – A Key To Avoiding Human Error

Ground software and procedures to verify command sequences, check flight rules, design science observations, and design ΔV maneuvers and other engineering activities are imperative for insuring errors do not get into uplinkable products and ultimately to the spacecraft. It cannot be emphasized enough that directing resources to ground software throughout the lifetime of a mission is key to safe operations. To date, over 130,000 spacecraft turns to point Cassini to accomplish science observations have occurred since reaching Saturn. Virtually all of these slews have been designed by the science teams themselves. Integrating all these activities together and insuring they meet all applicable flight rules and constraints requires robust ground software that both science observation designers and the engineering operations team utilize. The integration effort is huge and no hardware integration lab could possibly test more than a small percentage of these activities.

Spacecraft commanding requires a project-wide command database and a ground software tool used by all teams to generate commands for the spacecraft. At JPL, a tool called SEQGEN is used and the Cassini adaptation incorporates extensive flight rule checking as well as syntax and range checking on all command parameters. Outputs of this tool include a violations file and a "restricted" command summary. Certain commands are considered especially sensitive in that an incorrect argument could have dire consequences to spacecraft health and safety. Extra visual inspection must be done for any restricted command including physically signing a form allowing flight usage.

The bulk of ground software development on Cassini occurred pre-launch and during the 6.7-year cruise from Earth to Saturn. Two key AACS ground software tools are the Kinematic Predictor Tool (KPT) and the Inertial Vector Propagation (IVP) tool. IVP generates all the fixed and time-varying vector commands that populate the onboard inertial and body vector tables during sequence execution[9]. The KPT tool processes all the pointing-relating commands in a sequence, models all the pointing (slewing to and tracking targets) throughout an entire sequence, and models all flight rules that relate to spacecraft attitude (including the position of the sun and other celestial bodies). KPT emulates key AACS flight software but runs about 400 times real-time, allowing rapid verification of the 10-week background sequence.

Significant resources were devoted to validation and verification of each ground software tool. This verification effort was especially intense pre-launch and during the lengthy cruise to Saturn. Even during the Tour of Saturn, many tools required extensive updates or fixes. Each ground software tool is configuration-controlled by the project and upgrades and fixes require project manager approval. As an example, KPT and IVP ground software went through many iterations to ensure vector and turn modeling was completely consistent with actual spacecraft behavior, even for unexpected events like having one turn interrupt another. KPT implements the SRU bright body flight rules and creates the sequence commands to suspend star identification when needed. It also creates the commands to clear the AACS fault protection high water marks at the end of every downlink pass. IVP development continued during the Saturn Tour to enhance several science observation targeting categories, such as inertial vector definition commands for Radio Science observation requiring especially accurate time-varying pointing. KPT development continued during the Tour to enhance science instrument thermal modeling and improved ΔV predictions for RWA momentum changes. As recently as 2009, a new feature called Y-Biasing was added to KPT to allow slewing to an attitude tailored for that RWA momentum bias to reduce hydrazine usage and use the Y-firing thrusters more, saving the Z-thrusters for ΔV maneuvers and Titan flybys[10].

The software to design and verify ΔV maneuvers was required by launch because the first main engine burn occurred one month after launch. These ΔV maneuvers were recognized early in the mission to require a special design and verification process. Cassini's frequent flybys of Titan and the ever changing ground radiometric knowledge of the spacecraft orbital motion at Saturn dictates that an OTM must be designed only days before it must execute. Such rapid turnaround of a command sequence means that the OTM design and verification process has to be rapid and robust. The only way this can happen without unacceptable risk of an error is to automate a large amount of the process on the ground.

The development of maneuver automation software, and the process to go with it, was a long-term effort involving the whole project[11]. At least half a dozen normally-separate ground software tools had to be tied together, and a special process developed to allow exhaustive checks of the command sequence in a very compressed timeline (often less than 24 hours). A key part of this process is a rapid merge of the OTM product with the background sequence already on the spacecraft. The merged product is then run through extensive kinematic and flight rule checking simulations to ensure the OTM will seamlessly execute with the background sequence.

An important element of ground software is clear visibility of all flagged violations. KPT produces a violation summary report in a tabular format. This report documents all violations, when they occur, and list the science request in effect at the time of the violation. The KPT output log file is also important. This file should document all input and output files as well as include clear details about what caused each violation. One AACS analyst ignored a clear violation because the log file did not show him the input arguments of the command that caused the violation. The KPT log file needs to give the analyst as much visibility as possible into what led to each violation.

Engineers must have confidence in the ground software tools they use. Configuration control is vital, especially making sure that outdated versions of ground software cannot be accidentally invoked by analysts. Just as important is to ensure that ground software does not incorrectly flag a violation. Real violations may be ignored if an analyst notes that the tool is flagging violations incorrectly.

One aspect of turn modeling that might introduce an error is a turn that is near a singularity. On Cassini, the flight software will choose the shortest path to achieve a desired 3-axis spacecraft attitude. When this turn angle approaches 180° degrees, a tiny difference in attitude can result in a turn in the direction opposite from what was expected. This occurred during a gyro calibration on Cassini in 2003 and led to trouble because the actual path chosen in flight required autonomous evasive action because of a boresight-to-Sun constraint violation[12]. A flight rule was written to ensure that turns near 180° were flagged as violations so that workarounds can be identified during sequence design.

IX. Patching the Flight Software – Counter Risk by Rigorous Testing

One way human error in operations can adversely affects a spacecraft is by changing flight computer software without sufficient rigor. Mars Global Surveyor encountered this in 2006[13]. Flight software configuration management on Cassini has been strictly followed throughout the mission. A formal process involving project manager approval is required to even begin the effort to make a change to the flight software source code. Then extensive ITL and FSDS regression testing must demonstrate that the change has no adverse impact on any operational scenario. Testing must clearly show the “before” and “after” effect of any change. Then procedures and command sequences must be generated and successfully executed in ITL to demonstrate the process of uplinking and loading the FSW can be seamlessly integrated with the onboard background sequence. Only after all these checks are passed does the project manager permit the flight software change to be made onboard.

Human error can occur if a small “patch” to the flight software is introduced. The normal way ground controllers command the AACS flight computer is via goal-oriented sequence commands. Sequence commands have built-in checks to ensure non-corruption and mode and range checks to ensure validity. Memory write commands (“patches”) bypass all these checks. That is why “patching” via memory write is strongly discouraged on Cassini.

In the rare case that a patch is required, the AACS team must follow the same rigorous process of testing and project approval as for a full flight software source upload. Cassini experience has shown that any direct patch to the FSW RAM should only change a variable, not an instruction or a constant. Patching an instruction or a constant changes the FSW checksum. This is important because were a flight computer reset to occur, this checksum must match the original FSW image checksum, otherwise the AFC would not progress from ROM to RAM. In February of 2003, this did happen once in-flight (to the Backup AFC) during a planned FSW upload. The problem was traced to the patching of a constant prior to 2003 that altered the FSW checksum. After this event occurred in 2003, a flight rule was created to always use the AFC and SSR “patch table” (limited to 304 words) if a constant or instruction needs to be changed. The patch table method is robust and ensures a consistent checksum, a new FSW version number, and smooth progression into normal RAM operation, as well as preservation of the patched parameters even if the FSW is autonomously loaded from the SSRs.

In the years since arrival at Saturn, patch table use has been limited to:

1. Increase the Delta V telemetry resolution
2. Reduce the commanded detumble acceleration to be consistent with lower RCS thrust force
3. Adjust IRU scale factor errors based on in-flight gyro calibrations
4. Change the secondary vectors in the safe table (see the next section)
5. Adjust a fault protection parameter to reduce the risk of a thruster branch swap during low-altitude Titan flybys
6. Reduce the default spacecraft mass and RCS thrust to be consistent with the latter stages of the mission at Saturn

A goal of the design architecture of command processing should be to minimize in flight patching of the flight software. Parameters should be changeable through standard goal-oriented commands. On Cassini, flight software loading, latch valve states, mass properties, thrust characteristics, turn angles, power states, primeness, burn execution, vector updates, health states, deadband changes, RWA momentum changes, channelized telemetry sample rate modifications, all utilize standard commands with full parsing, handshake, and validity checks.

Parameter update flight commands should not be long lists of required argument values. Changing a parameter value has risk and this risk is increased if operators have to include a whole set of parameters for the command to be parsed correctly. If more than a few parameters are required in a command, a "No Change" option for each is especially useful.

X. Preparing For Anomalies

The history of spacecraft operations shows that human error tends to be more likely, and probably more damaging, when ground controllers are confronted with an anomaly on the spacecraft. Responding to the unexpected requires discipline and cool, careful thought. Do not rush. The spacecraft is designed with robust fault protection to allow it to take care of itself.

Although anomalies cannot be predicted, anomaly response procedures need to be in place and used when and if an anomaly occurs. On Cassini, these procedures include tables of expected states, telecommunications and Deep Space Network station configurations, quick-look checklists for each subsystem, steps to re-establish onboard IVT vectors if needed, relevant library command files that pertain to safing conditions, steps to follow if an OTM is needed, and relevant examples of fault protection log contents (from flight or FSDS simulations).

The nominal strategy for the Cassini orbital flight path will, perhaps a few times per year, place the spacecraft on a collision course with Titan. This is per design and ground controllers have three regularly-scheduled ΔV maneuver opportunities to adjust the trajectory. An emergency ΔV maneuver has never been required on Cassini, but even if it were needed, by far the safest way of performing such a maneuver is to follow normal OTM processes. Standard OTM design will work regardless of whether the spacecraft is in RCS or RWA control. Typically, the main engine would be selected but a long RCS ΔV maneuver using thrusters is an option. Main engine and thruster swap contingency procedures exist and a library command file exists to turn on required main engine heaters ahead of the burn if the background sequence has been suspended.

Preparing for an anomaly also means insuring that the spacecraft has a safe 3-axis spacecraft attitude that it can autonomously go to. If safe mode occurs, the spacecraft will suspend all active or pending command sequences (including the background sequence) and will autonomously turn to the safe mode attitude. On Cassini, this attitude is defined in an onboard safe table which can be updated by standard sequence ground commands.

The AACS operations team ensures that the safe table is properly managed throughout the mission[14]. Excepting the special cases of the critical sequences (Saturn orbit insertion and Huygens Probe Relay), ground controllers select the −Z body vector (HGA boresight) and the Earth-line vector as the safe table primary targeting set (insuring good ground commandability and high data-rate downlink). Selection of the secondary vectors (twist about the HGA) depends on the orbital inclination of the spacecraft. When Cassini is in a low-inclination orbit around Saturn, the secondary vectors are chosen to keep Saturn and the Rings away from the star tracker bright body field-of-view. An additional factor in secondary vector selection is to keep the thermally-sensitive +X-side of the spacecraft shaded from the Sun. Different secondary vectors are used when Cassini is in a high-inclination orbit around Saturn. Additional management is required during some low-altitude Titan flybys to ensure robust attitude control, thermal safety, and a clear star field for the SRU. All safe table management is planned months or years in advance and scheduled as part of nominal background sequence commanding.

XI. Flight Operations during Anomalies

The Cassini ground telemetry system has key channels "red alarmed" so that immediate notification is sent to the spacecraft operations team in the event of an anomaly. Examples of red alarmed parameters include thresholds on SRU CCD temperatures, increments in AFC reset counters, rejected command counters, and fault protection error response counters.

After an anomaly, establishing communication with the spacecraft is the first priority, evaluating what may have occurred is close behind. Avoid jumping to premature conclusions. The team should get as much information as possible about what was occurring just before the anomaly, what is known to have occurred during the anomaly, and the current state of the spacecraft.

After an AACS fault response, Cassini AACS flight software is designed to record the detailed responses in an onboard fault protection log. The first 18 entries of this log are autonomously transmitted to Earth during all downlinks, regardless of anomaly status. So "first fault response" information is available to ground controllers as soon as Earth-pointed communications are established after an anomaly. The fault protection log "pointer" (an index of how many entries were written to the onboard log by the flight software during the anomaly) is also directly available to ground controllers when ground communications are established. Standard "library" command files are on the shelf to command the readout of as many entries in the fault protection log as needed to capture the complete history of the fault protection response. On Cassini this log is in a fixed location in the AFC memory map simplifying its readout.

Certain anomalies may require the ITL or FSDS be used to investigate the anomaly or test recovery command files. The simulation needs to be configured to reproduce, as faithfully as possible, the known state of the spacecraft given the onboard anomaly and response. Recovering the spacecraft to nominal operation should only be considered after due diligence. After the 2010 safing event on Cassini, it was decided to skip a week or two of science gathering in order to safely put the spacecraft back into a stable and safe flight configuration before returning to nominal operation.

XII. Avoid Late Changes and Minimize Real-Time Activities

It takes several months to develop and check a 10-week background sequence. The entire project is involved in completing checklists for all their activities in the background sequence. The complete background sequence is fully simulated in KPT and SEQGEN and other ground software tools. All violations must be fixed as early in the process as possible. If a science team chooses to allow one of their flight rules to be violated in one particular instance, they must explain to the project manager why it is safe and acceptable to do so. All such "waivers" are fully documented and reviewed by project lead personnel.

In October of 2007, the S35 background sequence on Cassini was slightly modified near the end of the development process to incorporate a late Deep Space Network availability issue. A change this late in the process is itself a red flag. To make the fix, a slightly revised playback strategy was merged with rest of the background sequence. But the uplink operations lead forgot to include the IVP vectors in this final merge. The background sequence was made available to all science and engineering teams, but as the team that generates all IVP vectors, the AACS team was responsible for checking the vectors. The AACS sequence lead mistakenly loaded all the IVP vectors as a separate input to KPT and the KPT results looked completely normal. The sequence was uplinked to the spacecraft.

In this instance, the AACS uplink team lead performed a separate visual check of the sequence after uplink and noticed that the IVP vectors were missing. To deal with the situation, the vectors were packaged as a real-time file (but configured so that each vector was issued at its nominally prescribed time) and uplinked hours before the background sequence was scheduled to begin. What would have been an immediate safing event (the sequence would have been suspended as soon as the first needed vector was not available) ended up being a completely nominal background sequence execution. Both AACS and Uplink Operations procedures were revised following this incident. In particular, the AACS process was changed to never separately load IVP vectors into KPT and to always run the final uplinkable product as a standalone input in KPT.

In general, real-time activities (not part of a background sequence) are inherently riskier due to their not being part of the normal background sequence development and verification process. It is especially challenging to ensure the real-time activity will seamlessly integrate with the nominal sequence. In May of 2003, it was the hand-over from a real-time activity to a new background sequence that led to a safe mode event. The cause was an IVP vector that had been deleted at the end of the real-time activity. This vector, which defines the north pole of Saturn and is used routinely in Saturn operations, was assumed to be onboard the spacecraft by the background sequence lead. The analyst for the real-time activity thought he was "cleaning up" at the end of his activity, and chose to delete the vector. This deletion was not properly noted in the background sequence lead's initial state table. Normal operations were soon re-established, but the IVP table "handover" process, where one AACS lead provides his ending states to the next AACS analyst's initial states, was changed to be more rigorously followed. New checklist items were added, and the sequence development procedure was updated to ensure this miscommunication never happens again.

XIII. Summary

Avoiding human error in flight operations requires skilled engineers with good technical knowledge, good interpersonal communications, quality ground software, regular peer reviews, up-to-date procedures, as well as careful attention to detail and the discipline to test and verify all commands that will be sent to the spacecraft. Each mission is different and human error in operations requires eternal vigilance due to the risk of complacency – especially on long missions. However, there are clear guideposts that a flight operations team should follow to reduce the chance of an incorrect command sequence being sent to the spacecraft. The list below summarizes the kinds of questions and checks that the Cassini operations team follows:

Spacecraft design and configuration

- The spacecraft should be designed with flight operations in mind, including ease of commanding and visibility of spacecraft states to ground controllers.
- Design the flight software and command system to make changing parameters simple. Avoid patching the flight software by having a good design.
- Operate the spacecraft with all fault protection monitors fully enabled. Give extra care to any exceptions.
- A dedicated fault protection log that can be easily read out helps ground controllers in the recovery from an anomaly.
- Fault protection high water marks should be easily clearable and have good ground visibility.
- Avoid operating near a fault monitor threshold during nominal operation. Allow easy update of threshold limits based on flight experience.

Command validation and testing

- Allocate sufficient time to check and test commands to be sent to the spacecraft.
- Test to-be-flown command sequences with configuration-controlled ground software tools that have been configured to match the actual state of the spacecraft.
- Important activities and first-time events require thorough testing in a hardware integrated test laboratory that has been carefully configured, both hardware and software, to match the actual state of the spacecraft.
- Actual flight products should be tested with detailed review of the results by the flight operations team.
- Use ground software, wherever possible, to create the commands for an activity. Ground-expanded blocks are especially useful to ensure validity, range-checking, and relative/absolute command timing constraints are enforced.
- Check as many flight rules and constraints as possible in ground software. Provide clearly visible evidence of violations including how close to a threshold key parameters are operating.

Flight operations team

- Ensure engineers follow clear procedures in the design and analysis of flight products
- Analysts need to understand mission flight rules and constraints, why they exist, and how they are checked.
- Analysts need to demonstrate how they dealt with the issues they confronted in their design and verification of command sequences by conducting a peer review with relevant flight team members.
- A backup analyst needs to independently analyze and review all flight products
- Avoid "single point failures" where a single engineer has unique know-how. Document that unique knowledge and train backups.
- Incorporate all lessons learned from mistakes or anomalies into flight rules, ground software, or procedures.
- Analysts need to have ready access to a database of all onboard parameters and a history of all previous commanding.
- Analysts need to be especially careful when a process or ground software has changed. Extra review is warranted in these cases. Similar to checking key or first-time events, use the integrated test laboratory to ensure the effects of the change are completely understood.

Real-Time Activities

- It is preferred that activities be planned and placed in background sequences wherever possible to minimize the risk of human error in real-time activities. Real-time activities warrant extra review.
- Perform checks to ensure the activity does not clash with the background sequence and that it does not invalidate previous analysis or upcoming events on the spacecraft.
- There needs to be extra communication by analysts to ensure that the rest of the flight operations team understands the effects of the real-time activity

Changing The Flight Software

- Avoid memory writes (patching) to accomplish a flight software change whenever possible. If patching is required, verify its validity by reading out the "before" and "after" memory address contents in the integrated test laboratory and in flight.
- Any flight software changes should involve concept and readiness reviews by the entire project including functional and regression testing as well as integrated test laboratory checkout of the upload command sequences and procedures.
- Evaluate the effect of the flight software change on all ground procedures, simulators, ground software, databases, and command files already on the shelf.

Planning For Anomalies

- Ensure that tested command files are available to re-establish two-way communication with the spacecraft.
- Ensure there is adequate planning for conjunctions, occultations, and other periods where the operations team and/or spacecraft operate in a non-standard environment.
- Ensure that the autonomous safe mode 3-axis attitude for all upcoming periods meets all thermal, controllability, star tracker bright body, or other constraints
- Up-to-date procedures need to be available to aid in recovery after an anomaly. Especially important is early planning for recovery from a major hardware fault, before it actually occurs.
- Automate estimation and tracking of key hardware performance like individual thruster forces or RWA drag torque to ensure timely recognition of degraded behavior.
- Fault protection log readout and related command files to enhance visibility and aid recovery need to be in place well before an actual anomaly.
- Anomaly evaluation and recovery readiness tests should be performed periodically to ensure the team retains proficiency throughout the mission.

After An Anomaly

- During recovery, establish priorities for activities to be performed so that key expertise is focused on the highest priority tasks.
- Establish what lingering effects the anomaly may have on hardware or software states and ensure recovery corrects or mitigates them.
- Use ground simulators to understand and reproduce the fault – and to ensure the correctness of the recovery.

In summary, careful testing of flight products, keeping procedures and checklists up to date, having motivated and well-trained operations engineers, insuring backup analysts check the work of the lead analyst, insuring sufficient time to do a thorough job, minimizing flight software changes – all these things are important to getting the job done right. Complacency can develop in operations teams so the right mindset – check, test, and test again – is crucial for successful operations. Finally, having a robust and fault-tolerant spacecraft that allows ground controllers excellent visibility of its condition is the most important way to ensure human error does not compromise the mission.

Acknowledgments

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Special thanks to Dr. Allan Y. Lee and Mr. Todd S. Brown for their review of this paper. Thanks to Robert D. Rasmussen for internal JPL information about the design of the Cassini AACS hardware and flight software.

References

[1]Lee, A. Y., Hanover, G. "Cassini Spacecraft Attitude Control System Flight Performance", Paper AIAA-2005-6269, AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, August 15-18, 2005.

[2]Rasmussen, R. D., Singh, G., Rathbun, D. B., and Macala, G. A., "Behavioral Model Pointing on Cassini Using Target Vectors," *Proceedings of the Annual Rocky Mountain Guidance and Control Conference*, Keystone, Colorado, February 1–5, 1995, pp. 91–110.

[3]Brown, G. M., Bernard, D. E., and Rasmussen, R. D., "Attitude and Articulation Control for the Cassini Spacecraft: A Fault Tolerance Overview,"14th AIAA/IEEE Digital Avionics System Conference, Cambridge, MA, November 5-9, 1995.

[4]Clausen, K., Deutsch, L., *Huygens Recovery Task Force Final Report.* HUY-RP-12241, ESTEC, Noordwijk, The Netherlands, July 2001.

[5]Allestad, D., Standley, S., "Systems Overview of the Cassini-Huygens Probe Relay Critical Sequence", Paper AIAA-2005-6388, AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA, August 15-18, 2009.

[6]Brown, J., Lam, D., Chang, L., Burk, T., and Wette, M., "The role of the Flight Software Development System simulator Throughout the Cassini mission," AIAA 2005-6389, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, San Francisco, California, August 15-18, 2005.

[7]Brown, J., Wang, E., Hernandez, J., Lee, A.Y., "Importance of Model Simulation in Cassini In-Flight Mission Events", Paper AIAA-2009-5762, AIAA Guidance, Navigation, and Control Conference, Chicago, IL, August 10-13, 2009.

[8]Burk, T., and Bates, D. M., "Cassini Attitude Control Operations: Flight Rules and How They are Enforced," AIAA 2008-6808, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Honolulu, Hawaii, August 18-21, 2008.

[9]Burk, T., "Cassini Pointing Operations Flight Experience Using Inertial Vector Propagation", Paper AIAA-2007-6339, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Hilton Head, South Carolina, August 20-23, 2007.

[10]Brown, T., "Y-Biasing: A New Operational Method For Cassini to Control Thruster Usage while Managing Reaction Wheel Momentum", Paper AIAA-2010-7560, AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, August 2-5, 2010.

[11]Velarde Yang, G, Kirby, C., Mohr, D., "Cassini's Maneuver Automation Software (MAS) Process: How to Successfully Command 200 Navigation Maneuvers," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Honolulu, Hawaii, August 18-21, 2008.

[12]Singh, G., Macala, Glenn A., Wong, Edward C., and Rasmussen, R. D., "A Constraint Monitor Algorithm for the Cassini Spacecraft", AIAA Paper 97-3526, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, Louisiana, August 11-13, 1997.

[13]Perkins, D., NASA Internal Review Board Report on Mars Global Surveyor (MGS) Spacecraft Loss of Contact, April 13, 2007.

[14]Burk, T., "Managing Cassini Safe Mode Attitude at Saturn", Paper AIAA-2010-7558, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, August 2-5, 2010.