DRAFT!! The Loss of a Milstar Satellite

Nancy G. Leveson

On April 30, 1999, at 12:30 EDT, a Titan IV B-32 booster equipped with a Centaur TC-14 upper stage was launched from Cape Canaveral. The mission was to place a Milstar-3 satellite in geosynchronous orbit. Milstar is a joint services satellite communications system that provides secure, jam resistant, worldwide communications to meet wartime requirements. It was the most advanced military communications satellite system to that date. The first Milstar satellite was launched February 7, 1994 and the second was launched November 5, 1995. This mission was to be the third launch.

As a result of some anomalous events, the Milstar satellite was placed in an incorrect and unusable low elliptical final orbit, as opposed to the intended geosynchronous orbit. Media interest was high due to this mishap being the third straight Titan IV failure and due to recent failures of other commercial space launches. In addition, this accident is believed to be one of the most costly unmanned losses in the history of Cape Canaveral Launch Operations. The Milstar satellite cost about $800 million and the launcher an additional $433 million.

Lockheed Martin Astronautics (LMA) was the prime contractor for the mission. The Space and Missile Systems Center Launch Directorate (SMC) was responsible for insight and administration of the LMA contract.

The Accident Investigation Board concluded that:

> Failure of the Titan IV B-32 mission is due to a failed software development, testing, and quality assurance process for the Centaur upper stage. That failed process did not detect and correct a human error in the manual entry of the I1(25) roll rate filter constant entered in the Inertial Measurement System flight software file. The value should have been entered as -1.992476, but was entered as -0.1992476. Evidence of the incorrect I1(25) constant appeared during launch processing and the launch countdown, but its impact was not sufficiently recognized or understood and, consequently, not corrected before launch. The incorrect roll rate filter constant zeroed any roll rate data, resulting in the loss of roll axis control, which then caused loss of yaw and pitch control. The loss of attitude control caused excessive firings of the Reaction Control system and subsequent hydrazine depletion. Erratic vehicle flight during the Centaur main engine burns caused the Centaur to achieve an orbit apogee and perigee much lower than desired, which resulted in the Milstar separating in a useless low final orbit.

To fully understand this accident, we need to understand why the error in the roll rate filter constant was introduced in the load tape, why it was not found during the tape production process and internal review processes, why it was not found in IV&V, and why it was not detected during operations at the launch site. In other words, why the safety control structure was ineffective in each of these instances.

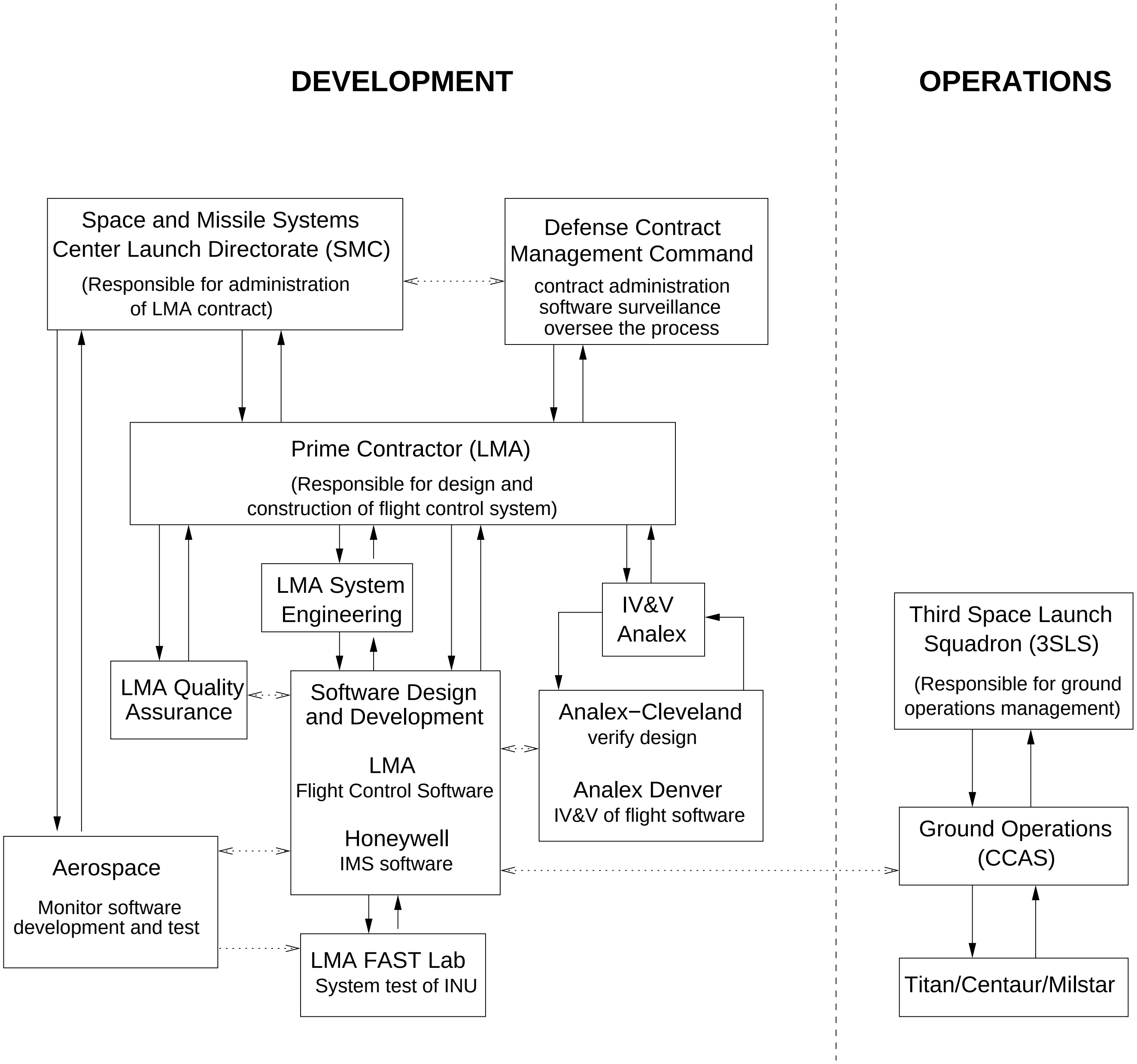

Figure 1: Hierarchical Control Structure

Figure 1 shows the hierarchical control model of the accident, or at least those parts that can be gleaned from the official accident report.[1] Besides SMC and LMA, the Defense Contract Management Command (DCMC) played some type of oversight role, but the report is not clear about what exactly this role was beyond a general statement about responsibility for contract management, software surveillance, and overseeing the development process.

LMA designed and developed the flight control software, while Honeywell was responsible for the IMS software. This separation of control, combined with poor coordination, accounts for some of the problems that occurred. Analex was the IV&V contractor, while Aerospace Corporation provided independent monitoring and evaluation. Ground launch operations at Cape Canaveral Air Station (CCAS) were managed by the Third Space Launch Squadron (3SLS).

Starting from the bottom and working up the levels of control, each level is examined for the flaws in the process at that level that provided inadequate control of safety in the process level below. The process flaws at each level are then examined and explained in terms of a potential mismatch in models between the controller's model of the process and the real process, incorrect design of the control algorithm, lack of coordination among the control activities, deficiencies in the reference channel, and deficiencies in the feedback or monitoring channel. This accident, as well as most others, includes examples of asynchronous evolution and adaptation.

One general thing to note in this accident is that there were a large number of redundancies in each part of the process to prevent the loss, but they were not effective. Sometimes, built-in redundancy itself causes complacency and overconfidence and is a critical factor in the accident process, as in this case.

1 The Physical Process (Titan/Centaur/Milstar)

Components of the Physical Process: The Lockheed Martin Astronautics (LMA) Titan IV B is a heavy-lift space launch vehicle used to carry government payloads such as Defense Support Program, Milstar, and National Reconnaisance Office satellites into space. It can carry up to 47,800 pounds into low-earth orbit and up to 12,700 pounds into a geosynchronous orbit. The vehicle can be launched with no upper stage or with one of two optional upper stages, providing greater and varied capability.

The LMA Centaur is a cryogenic, high-energy upper stage. It carries its own guidance, navigation, and control system, which measures the Centaur's position and velocity on a continuing basis throughout flight. It also determines the desired orientation of the vehicle in terms of pitch, yaw, and roll axis vectors. It then issues commands to the required control components to orient the vehicle in the proper attitude and position, using the main engine or the Reaction Control System (RCS) engines (Figure 2). The main engines are used to control thrust and velocity. The RCS provides thrust for vehicle pitch, yaw, and roll control, for post-injection separation and orientation maneuvers, and for propellant settling prior to engine restart.

System Hazar (2) (1) The satellite does not reach a useful geosynchronous orbit the satellite is damaged during orbit insertion maneuvers and cannot provide its intended function.

: (INU) **Description of Process Controller** The Inertial Navigation (Figure 2): (1) the Guidance, Navigation, and Control System (the Flight Control Software or FCS) and (2) an Inertial Measurement System (IMS). The Flight Control Software computes the desired

[1] Some details of the control structure may be incorrect because I had to guess at them, but the model should suffice for this example analysis.

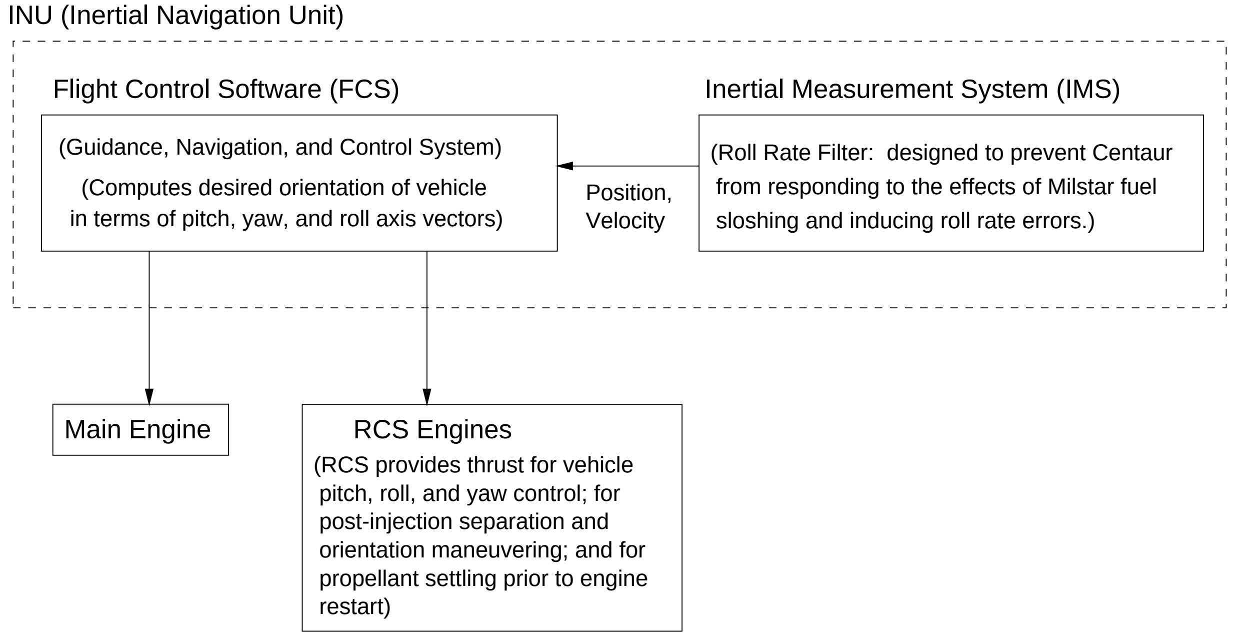
INU (Inertial Navigation Unit)
Flight Control Software (FCS)
(Guidance, Navigation, and Control System)
(Computes desired orientation of vehicle in terms of pitch, yaw, and roll axis vectors)
Inertial Measurement System (IMS)
(Roll Rate Filter: designed to prevent Centaur from responding to the effects of Milstar fuel sloshing and inducing roll rate errors.)
Position, Velocity
Main Engine
RCS Engines
(RCS provides thrust for vehicle pitch, roll, and yaw control; for post-injection separation and orientation maneuvering; and for propellant settling prior to engine restart)

Figure 2: TechnicUal Process Control Structure for IN

orientation of the vehicle in terms of the pitch, yaw, and roll axis vectors and issues commands to the main engines and the reaction control system to control vehicle orientation and thrust. To accomplish this goal, the FCS uses position and velocity information provided by the IMS. The component of the IMS involved in the loss is a roll rate filter, which is designed to prevent the Centaur from responding to the effects of Milstar fuel sloshing and thus inducing roll rate errors.

CS:Safety Constraint on The FCS must provide the attitude control, separation, and orientation maneuvering commands to the main engines and the RCS system necessary to attain geosynchronous orbit.

Safety IMConstraint on S: The position and velocity values provided to the FCS must not be capable of leading to a hazardous control action. The roll rate filter must prevent the Centaur from responding to the effects of fuel sloshing and inducing roll rate errors.

2 Descri of the Pro al Loss the

There were three planned burns during the Centaur flight. The first burn was intended to put the Centaur into a parking orbit. The second would move the Centaur into an elliptical transfer orbit that was to carry the Centaur and the satellite to geosynchronous orbit. The third and final burn would circularize the Centaur in its intended geosynchronous orbit. A coast phase was planned between each burn. During the coast phase, the Centaur was to progress under its own momentum to the proper point in the orbit for the next burn. The Centaur would also exercise a roll sequence and an attitude control maneuver during the coast periods to provide passive thermal control and to settle the main engine propellants in the bottom of the tanks.

First Burn: The first burn was intended to put the Centaur into a parking orbit. The IMS transmitted a zero or near zero roll rate to the Flight Control software, however, due to the use of

With an incorrect roll rate filter constant. Without roll rate feedback, the FCS provided inappropriate control commands that caused the Centaur to become unstable about the roll axis and not to roll to the desired first burn orientation. The Centaur began to roll back and forth, eventually creating sloshing of the vehicle liquid fuel in the tanks, which created unpredictable forces on the vehicle and adversely affected flow of fuel to the engines. By the end of the first burn (approximately 11 minutes and 35 seconds after liftoff), the roll oscillation began to affect the pitch and yaw rates of the vehicle as well. The FCS predicted an incorrect time for main engine shutdown due to the effect on the acceleration of the vehicle's tumbling and fuel sloshing. The incorrect shutdown in turn resulted in the Centaur not achieving its intended velocity during the first burn, and the vehicle was placed in an unintended park orbit.

<u>*First Coast Phase*</u>: During the coast phases, the Centaur was to progress under its own momentum to the proper point in the orbit for the next burn. During this coasting period, the FCS was supposed to command a roll sequence and an attitude control maneuver to provide passive thermal control and to settle the main engine propellants in the bottom of the tanks. Because of the roll instability and transients created by the engine shutdown, the Centaur entered this first coast phase tumbling. The FCS directed the RCS to stabilize the vehicle. Late in the park orbit, the Centaur was finally stablized about the pitch and yaw axes, although it continued to oscillate about the roll axis. In stabilizing the vehicle, however, the RCS expended almost 85 percent of the RCS system propellant (hydrazine).

<u>*Second Burn*</u>: The FCS successfully commanded the vehicle into the proper attitude for the second burn, which was to put the Centaur and the satellite into an elliptical transfer orbit that would carry them to geosynchronous orbit. The FCS ignited the main engines at approximately one hour, six minutes, and twenty-eight seconds after liftoff. Soon after entering the second burn phase, however, inadequate FCS control commands caused the vehicle to again become unstable about the roll axis and begin a diverging roll oscillation.

Because the second burn is longer than the first, the excess roll commands from the FCS eventually saturated the pitch and yaw channels. At approximately two minutes into the second burn, pitch and yaw control was lost (as well as roll), causing the vehicle to tumble for the remainder of the burn. Due to its uncontrolled tumbling during the burn, the vehicle did not achieve the planned acceleration for transfer orbit.

<u>*Second Coast Phase*</u> (transfer orbit): The RCS attempted to stabilize the vehicle but it continued to tumble. The RCS depleted its remaining propellant approximately twelve minutes after the FCS shut down the second burn.

<u>*Third Burn*</u>: The goal of the third burn was to circularize the Centaur in its intended geosynchronous orbit. The FCS started the third burn at two hours, thirty-four minutes, and fifteen seconds after liftoff. It was started earlier and was shorter than had been planned. The vehicle tumbled throughout the third burn, but without the RCS there was no way to control it. Space vehicle separation was commanded at approximately two hours after the third burn began, resulting in the Milstar being placed in a useless low elliptical orbit, as opposed to the desired geosynchronous orbit (Figure 3).

<u>*Post Separation*</u>: The Mission Director ordered early turn-on of the satellite in an attempt to save it, but the ground controllers were unable to contact the satellite for approximately three hours. Six hours and fourteen minutes after liftoff, control was acquired and various survival and emergency actions were taken. The satellite had been damaged from the uncontrolled vehicle pitch, yaw, and roll movements, however, and there were no possible actions the ground controllers could have taken in response to the anomalous events that would have saved the mission.

The mission was officially declared a failure on May 4, 1999, but personnel from LMA and the Air Force controlled the satellite for six additional days in order to place the satellite in a non-

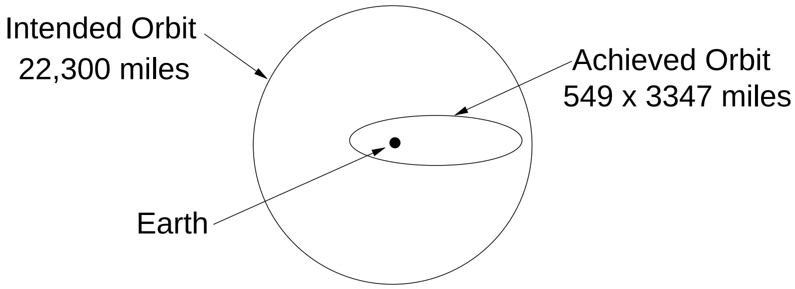

Figure 3: Achieved Orbit vs. Intended Orbit

interfering orbit with minimum risk to operational satellites. It appears the satellite performed as designed, despite the anomalous conditions. It was shut down by ground control on May 10, 1999.

3 Physical Process and Automated Controllers: Dysfunctional Interactions

Figure 4 shows the automated controller flaws leading to the accident. The Inertial Measurement System algorithm was incorrect, specifically, there was an incorrect roll rate filter constant in the IMS software file (Figure 4) that led to a dysfunctional interaction with the flight control software. However, the algorithm operated as designed (i.e., it did not fail).

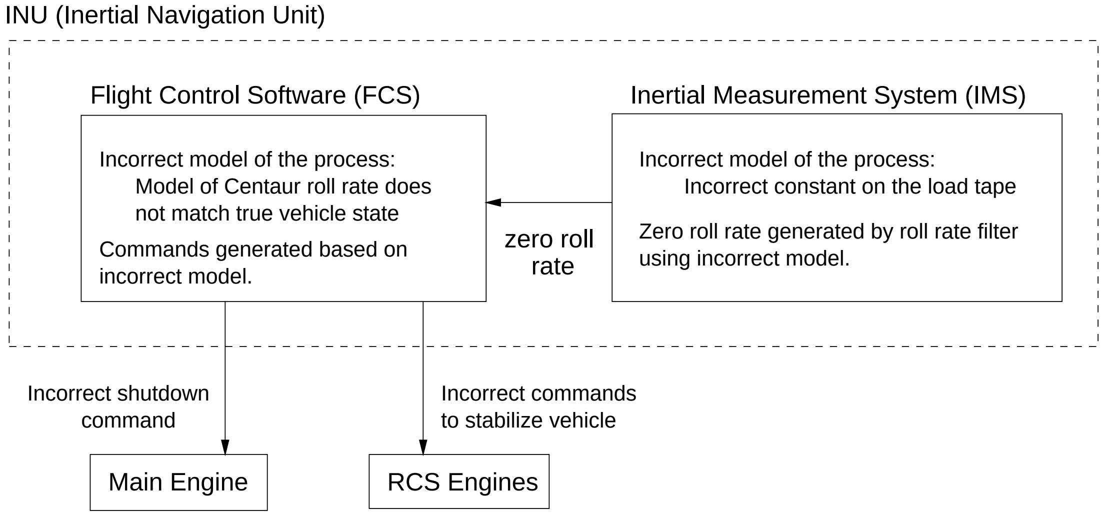
INU (Inertial Navigation Unit)
Flight Control Software (FCS)
Incorrect model of the process:
Model of Centaur roll rate does not match true vehicle state
Commands generated based on incorrect model.
Inertial Measurement System (IMS)
Incorrect model of the process:
Incorrect constant on the load tape
Zero roll rate generated by roll rate filter using incorrect model.
zero roll rate
Incorrect shutdown command
Incorrect commands to stabilize vehicle
Main Engine
RCS Engines

Figure 4: Control Flaws at the Physical Process Level

The Flight Control Software operated correctly (i.e., according to its requirements). However, it received incorrect input from the IMS, leading to an incorrect internal FCS software model of the process: the roll rate was thought to be zero or near zero when it was not. Thus there was a mismatch between the FCS internal model of the process state and the real process state. This mismatch led to the RCS issuing incorrect control commands to the main engine (to shutdown early) and to the RCS engines. Using STAMP terminology, the loss resulted from a dysfunctional interaction between the FCS and the IMS. Neither failed: they operated correctly with respect to the instructions (including constants) and data provided.

The accident report does not explore whether the FCS software could have included sanity checks on the roll rate or vehicle behavior to detect that incorrect roll rates were being provided by the IMS. Even if the FCS did detect it was getting anomalous roll rates, there may not have been any recovery or fail-safe behavior that could have been designed into the system. Without more information about the Centaur control requirements and design, it is not possible to speculate about whether the Inertial Navigation Unit software (the IMS and FCS) might have been designed to be fault tolerant with respect to filter constant errors.

This level of explanation of the flaws in the process (the vehicle and its flight behavior) as well as its immediate controller provides a description of the "symptom," but does not provide enough information about the factors involved to prevent reoccurrences. Simply fixing that particular flight tape is not enough. We need to look at the higher levels of the control structure for that. Figures 5 and 6 summarize the information in the rest of this paper.

4 Launch Site Operations

The function of launch site operations is to monitor launch pad behavior and tests and detect any critical anomalies prior to flight. Why was the roll rate error not detected during launch operations?

Safety Constraint Violated: Critical variables (including those in software) must be monitored and errors detected before launch. Potentially hazardous anomalies detected at the launch site must be formally logged and thoroughly investigated and handled.

Context: Management had greatly reduced the number of engineers working launch operations, and those remaining were provided with few guidelines as to how they should perform their job. The accident report says that their tasks were not defined by their management so they used their best engineering judgment to determine which tasks they should perform, which variables they should monitor, and how closely to analyze the data associated with each of their monitoring tasks.

Controls: The controls are not described well in the report. From what is included, it does not appear that controls were implemented to monitor or detect software errors at the launch site although a large number of vehicle variables were monitored.

Roles and Responsibilities: The report is also not explicit about the roles and responsibilities of those involved. LMA had launch personnel at CCAS, including Product Integrity Engineers (PIEs). 3SLS had launch personnel to control the launch process as well as software to check process variables and to assist the operators in evaluating observed data.

Failures, Dysfunctional Interactions, Flawed Decisions, and Inadequate Control Actions: Despite clear indications of a problem with the roll rate information being produced by the IMS, it was not detected by some launch personnel who should have and detected but mishandled by others. Specifically:

1. One week before launch, LMA personnel at CCAS observed much lower roll rate filter values than they expected. When they could not explain the differences at their level, they raised their concerns to Denver LMA Guidance Product Integrity Engineers (PIEs), who were now at CCAS. The on-site PIEs could not explain the differences either, so they directed the CCAS personnel to call the control dynamics (CD) design engineers in Denver. On Friday, April 23,

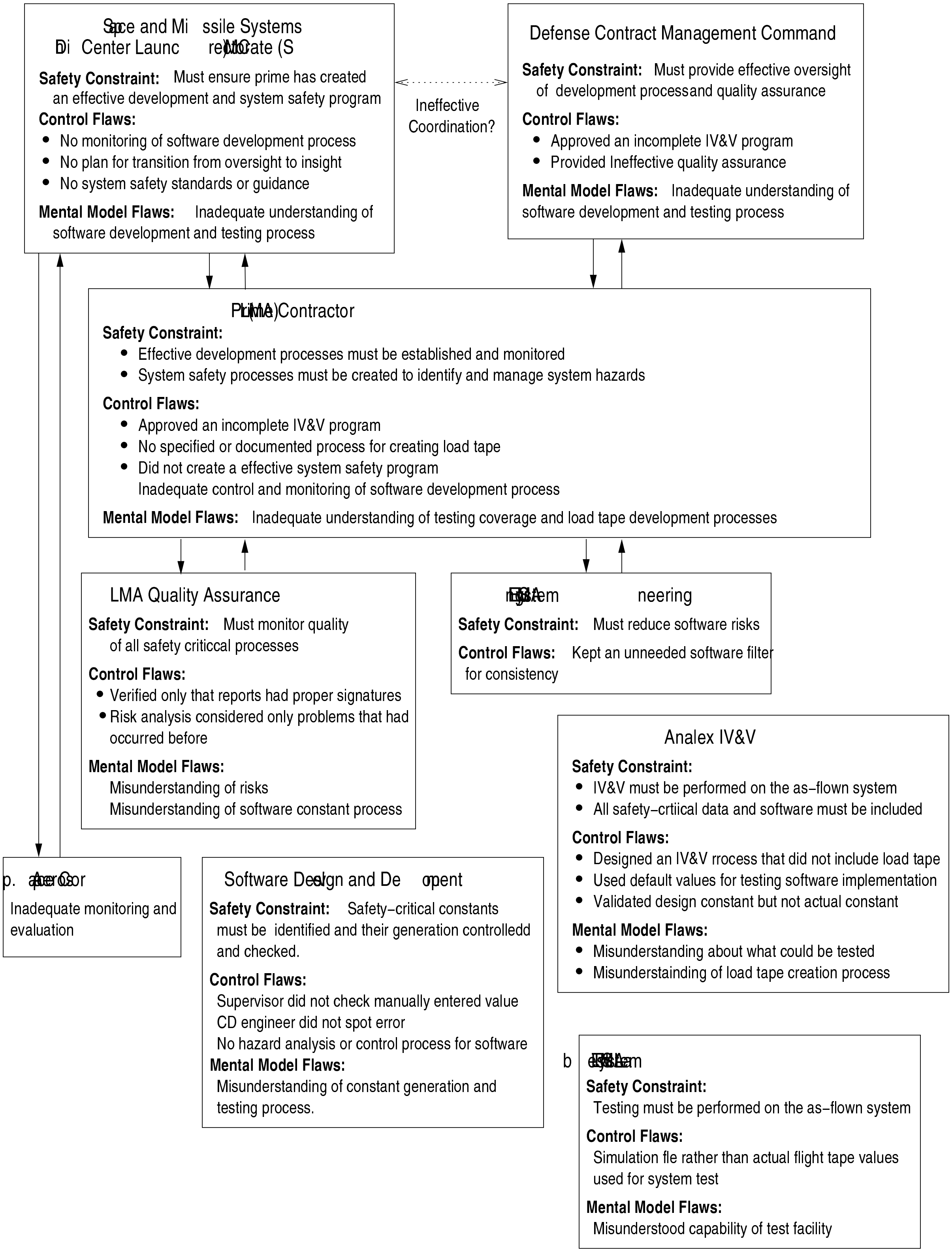
Space and Missile Systems Center Launch Directorate (SMC)
Safety Constraint: Must ensure prime has created an effective development and system safety program
Control Flaws:
• No monitoring of software development process
• No plan for transition from oversight to insight
• No system safety standards or guidance
Mental Model Flaws: Inadequate understanding of software development and testing process
Ineffective Coordination?
Defense Contract Management Command
Safety Constraint: Must provide effective oversight of development processand quality assurance
Control Flaws:
• Approved an incomplete IV&V program
• Provided Ineffective quality assurance
Mental Model Flaws: Inadequate understanding of software development and testing process
Prime Contractor (LMA)
Safety Constraint:
• Effective development processes must be established and monitored
• System safety processes must be created to identify and manage system hazards
Control Flaws:
• Approved an incomplete IV&V program
• No specified or documented process for creating load tape
• Did not create a effective system safety program
Inadequate control and monitoring of software development process
Mental Model Flaws: Inadequate understanding of testing coverage and load tape development processes
LMA Quality Assurance
Safety Constraint: Must monitor quality of all safety criticcal processes
Control Flaws:
• Verified only that reports had proper signatures
• Risk analysis considered only problems that had occurred before
Mental Model Flaws:
Misunderstanding of risks
Misunderstanding of software constant process
LMA System Engineering
Safety Constraint: Must reduce software risks
Control Flaws: Kept an unneeded software filter for consistency
Analex IV&V
Safety Constraint:
• IV&V must be performed on the as−flown system
• All safety−crtiical data and software must be included
Control Flaws:
• Designed an IV&V rrocess that did not include load tape
• Used default values for testing software implementation
• Validated design constant but not actual constant
Mental Model Flaws:
• Misunderstanding about what could be tested
• Misunderstainding of load tape creation process
Aerospace Corp.
Inadequate monitoring and evaluation
Software Design and Development
Safety Constraint: Safety−critical constants must be identified and their generation controlledd and checked.
Control Flaws:
Supervisor did not check manually entered value
CD engineer did not spot error
No hazard analysis or control process for software
Mental Model Flaws:
Misunderstanding of constant generation and testing process.
LMA System Test Lab
Safety Constraint:
Testing must be performed on the as−flown system
Control Flaws:
Simulation fle rather than actual flight tape values used for system test
Mental Model Flaws:
Misunderstood capability of test facility

Figure 5: STAMP model of Development Process

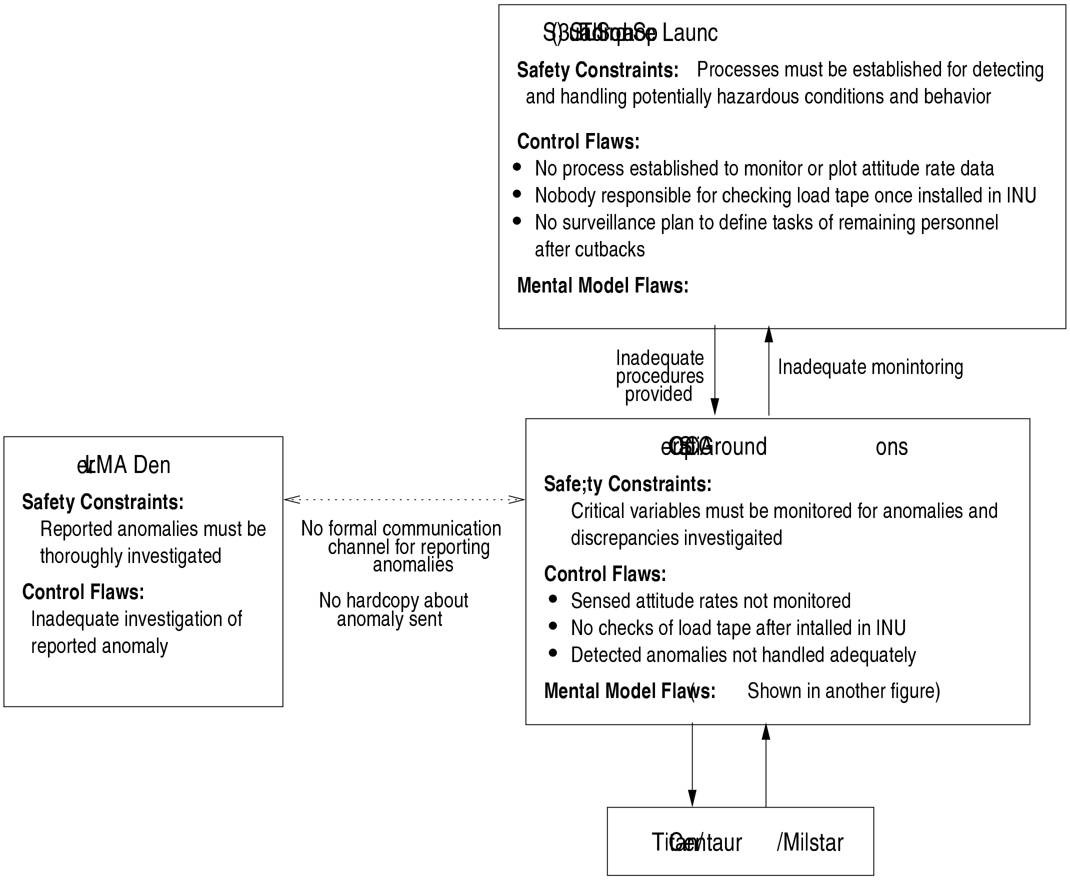
S()3SLSpaSe Launc
Safety Constraints: Processes must be established for detecting and handling potentially hazardous conditions and behavior
Control Flaws:
• No process established to monitor or plot attitude rate data
• Nobody responsible for checking load tape once installed in INU
• No surveillance plan to define tasks of remaining personnel after cutbacks
Mental Model Flaws:
Inadequate procedures provided
Inadequate monintoring
eLMA Den
Safety Constraints:
Reported anomalies must be thoroughly investigated
Control Flaws:
Inadequate investigation of reported anomaly
No formal communication channel for reporting anomalies
No hardcopy about anomaly sent
erCSGround ons
Safe;ty Constraints:
Critical variables must be monitored for anomalies and discrepancies investigaited
Control Flaws:
• Sensed attitude rates not monitored
• No checks of load tape after intalled in INU
• Detected anomalies not handled adequately
Mental Model Flaws: (Shown in another figure)
TitCenrtaur /Milstar

Figure 6: STAMP model of Launch Operations Process

the LMA Guidance Engineer telephoned the LMA CD lead. The CD lead was not in his office so the Guidance Engineer left a voice mail stating she noticed a significant change in roll rate when the latest filter rate coefficients were entered. She requested a return call to her or to her supervisor. The Guidance Engineer also left an email for her supervisor at CCAS explaining the situation. Her supervisor was on vacation and was due back at the office Monday morning April 26, when the Guidance Engineer was scheduled to work the second shift. The CD lead and the CD engineer who originally specified the filter values listened to the voice mail from the Guidance Engineer. They called her supervisor at CCAS who had just returned from vacation. He was initially unable to find the email during their conversation. He said he would call back, so the CD engineer left the CD lead's office. The CD lead subsequently talked to the Guidance Engineer's supervisor after he found and read the email. The CD lead told the supervisor at CCAS that the filter values had changed in the flight tape originally loaded on April 14, 1999, and the roll rate output should also be expected to change. Both parties believed the difference in roll rates observed were attributable to expected changes with the delivery of the flight tape.

2. On the day of the launch, a 3SLS INU Product Integrity Engineer (PIE) at CCAS noticed the low roll rates and performed a rate check to see if the gyros were operating properly. Unfortunately, the programmed rate check used a default set of I1 constants to filter the measured rate and consequently reported that the gyros were sensing the earth rate correctly. If the sensed attitude rates had been monitored at that time or if they had been summed and plotted to ensure they were properly sensing the earth's gravitational rate, the roll rate problem could have been identified.

3. A 3SLS engineer also saw the roll rate data at the time of tower rollback, but was not able to identify the problem with the low roll rate. He had no documented requirement or procedures to review the data and no reference to compare to the roll rate actually being produced.

The communication channel between LMA Denver and the LMA engineers at CCAS was clearly flawed. There is no information about any established reporting channel from the LMA CCAS or LMA Denver engineers to a safety organization or up the management chain. No “alarm” system adequate to detect the problem or that it was not being adequately handled seems to have existed. The report says there was confusion and uncertainty from the time the roll rate anomaly was first “raised” by the CCAS LMA engineer in email and voice mail until it was
it should be be reported, analyzed, documented, and tracked since it was a “concern” and not a “deviation.” There is no explanation of these terms nor any description of a formal problem reporting and handling system in the accident report.

[illegible] Control The accident report says that at this point in the prelaunch process, there was no process to monitor or plot attitude rate data, that is, to perform a check to see if the attitude filters were properly sensing the earth's rotation rate. Nobody was responsible for checking the load tape constants once the tape was installed in the IMU at the launch site. Therefore, nobody was able to question the anomalous rate data recorded or correlate it to the low roll rates observed about a week prior to launch and on the day of launch. In addition, the LMA engineers at Denver never asked to see a hard copy of the actual data observed at CCAS, nor did they talk to the guidance engineer or Data Station Monitor at CCAS who questioned the low filter rates. They simply explained it away as attributable to expected changes associated with the delivery of the flight tape.

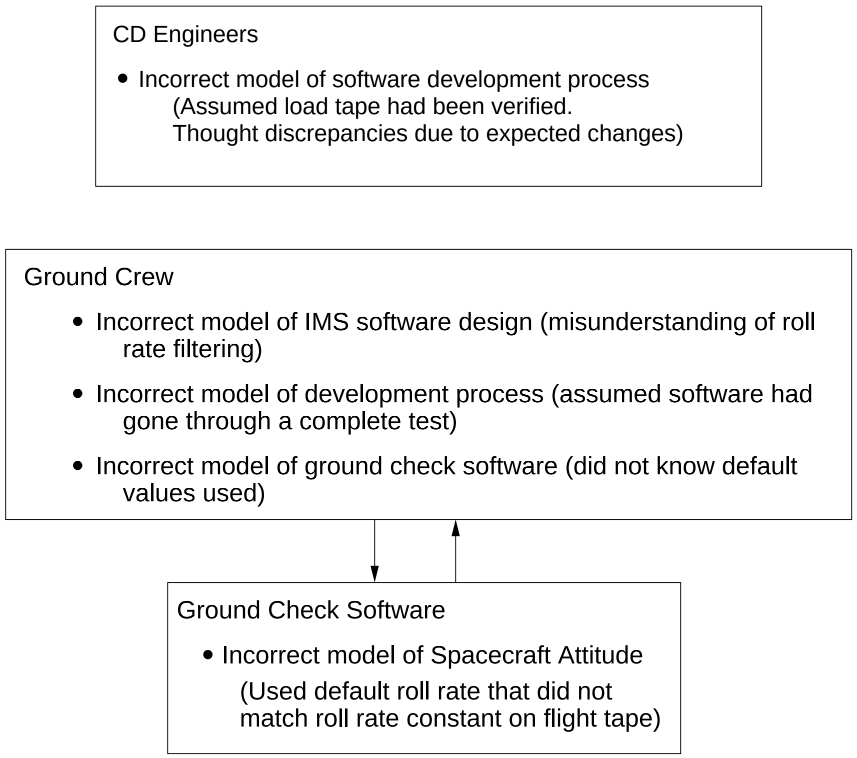
CD Engineers
• Incorrect model of software development process
(Assumed load tape had been verified.
Thought discrepancies due to expected changes)
Ground Crew
• Incorrect model of IMS software design (misunderstanding of roll rate filtering)
• Incorrect model of development process (assumed software had gone through a complete test)
• Incorrect model of ground check software (did not know default values used)
Ground Check Software
• Incorrect model of Spacecraft Attitude
(Used default roll rate that did not
match roll rate constant on flight tape)

Figure 7: The Flawed Process Models used by the Ground Personnel and Software

Process Mdb F Five models are involved here (see Figure 7):

1. Rate check software: The software used to do a rate check on the day of launch used default constants instead of the actual load tape. Thus there was a mismatch between the model used in the rate checking software and the model used by the IMS software.

2. Ground crew models of the development process: Although the report does not delve into this factor, it is very possible that complacency may have been involved and that the model of the thoroughness of the internal quality assurance and external IV&V development process in the minds of the ground operations personnel as well as the LMA guidance engineers who were informed of the observed anomalies right before launch did not match the real development process. There seemed to be no checking of the correctness of the software after the standard testing during development. Hardware failures are usually checked up to launch time, but often testing is assumed to have removed all software errors and therefore further checks are not needed.

3. Ground crew models of the IMS software design: The ground launch crew had an inadequate understanding of how the roll rate filters worked. No one other than the control dynamics engineers who designed the I1 roll rate constants understood their use or the impact of filtering the roll rate to zero. So when discrepancies were found before launch, nobody at the launch site understood the I1 roll rate filter design well enough to detect the error.

4. Ground crew models of the rate check software: Apparently, the ground crew was unaware that the checking software used default values for the filter constants.

5. CD engineers' model of the flight tape change: The control dynamics lead engineer at the launch site and her supervisor at LMA Denver thought that the roll rate anomalies were due to known changes in the flight tape. Neither went back to the engineers themselves to check this conclusion with those most expert in the details of the Centaur control dynamics.

Coordination: Despite several different groups being active at the launch site, nobody had been assigned responsibility for monitoring the software behavior after it was loaded into the INU. The accident report does not mention coordination problems, although it does say there was a lack of understanding of each other's responsibilities between the LMA launch personnel (at CCAS) and the development personnel at LMA Denver and that this led to the concerns of the LMA personnel at CCAS not being adequately addressed.

A more general question that might have been investigated was whether the failure to act properly after detecting the roll rate problem involved a lack of coordination and communication problems between LMA engineers at CCAS and 3SLS personnel. Why did several people notice the problem with the roll rate but do nothing and why were the anomalies they noticed not effectively communicated to those who could do something about it? Several types of coordination problems might have existed. For example, there might have been an overlap problem, with each person who saw the problem assuming that someone else was handling it.

Feedback: There was a missing or inadequate feedback channel from the launch personnel to the development organization.

Tests right before launch detected the zero roll rate, but there was no formal communication channel established for getting that information to those who could understand it. Instead voice mail and email were used. The report is not clear, but either there was no formal anomaly reporting and tracking system or it was not known or used by the process participants.[2]

The LMA (Denver) engineers requested no hardcopy information about the reported anomaly and did not speak directly with the Guidance engineer or Data Station Monitor at CCAS.

5 Air Force Launch Operations Management: Third Space Launch Squadron (3SLS)

Safety Constraint: Processes must be established for detecting and handling potentially hazardous conditions and behavior detected during launch preparations.

Context: 3SLS management was transitioning from an *oversight* role to an *insight* one without a clear definition of what such a transition might mean or require.

Control Algorithm Flaws: After the ground launch personnel cutbacks, 3SLS management did not create a master surveillance plan to define the tasks of the remaining personnel (the formal insight plan was still in draft). In particular, there were no formal processes established to check the validity of the I1 filter constants or to monitor attitude rates once the flight tape was loaded into the INU at Cape Canaveral Air Station (CCAS) prior to launch. 3SLS launch personnel were provided with no documented requirement nor procedures to review the data and no references with which to compare the observed data in order to detect anomalies.

[2] See a recent aerospace accident involving the bypassing of formal anomaly reporting channels and the substitution of informal email and other communication—with similar results.

Process Model Flaws: It is possible that misunderstandings (an incorrect model) about the thoroughness of the development process led to a failure to provide requirements and processes for performing software checks at the launch site. Complacency may also have been involved, i.e., the common assumption that software does not fail and that software testing is exhaustive and therefore additional software checking was not needed. However, this is speculation as the report does not explain why management did not provide documented requirements and procedures to review the launch data nor ensure the availability of references for comparison so that discrepancies could be discovered.

Coordination: The lack of oversight led to a process that did not ensure that anyone was responsible for some specific launch site tasks.

Feedback or Monitoring Channel: Apparently, launch operations management had no "insight" plan in place to monitor the performance of the launch operations process. There is no information included in the accident report about the process to monitor the performance of the launch operations process or what type of feedback was used (if any) to provide insight into the process.

6 System/Software Development of the Centaur Flight Control System

Too often, accident investigators stop at this point after identifying operational errors that, if they had not occurred, might have prevented the loss. Occasionally operations management is faulted. Operator errors provide a convenient place to stop in the backward chain of events from the loss event. To their credit, the accident investigation board in this case kept digging. To understand why an erroneous flight tape was created in the first place (and to learn how to prevent a similar occurrence in the future), the software and system development process associated with generating the tape needs to be examined.

Process Description: The INU, for which all the flight software is developed, consists of two major software components developed by different companies: LMA developed the Flight Control System software and was responsible for overall INU testing while Honeywell developed the IMS and was partially responsible for its software development and testing. The I1 constants are processed by the IMS, but were designed and tested by LMA.

Safety Constraint Violated: Safety-critical constants must be identified and their generation controlled and checked.

Dysfunctional Interactions, Flawed Decisions, and Inadequate Control Actions: Software Constants and Code: A Wordstream memo was generated by the LMA Control Dynamics (CD) group and sent to the LMA Centaur Flight Software (FS) group on December 23, 1997. It provided the intended and correct values for the first I1 constants in hardcopy form. The memo also allocated space for 10 additional constants to be provided by the LMA Avionics group at a later time and specified a path and file name for an electronic version of the first 30 constants. The memo did not specify or direct the use of either the hardcopy or the electronic version for creating the constants database.

In early February, 1999, the LMA Centaur FS group responsible for accumulating all the software and constants for the flight load tape was given discretion in choosing a baseline data file. The flight software engineer who created the database dealt with over 700 flight constants generated by multiple sources, in differing formats, and at varying time (some with multiple iterations) all of which had to be merged into a single database. Some constant values came from electronic files that could be merged into the database, while others came from paper memos manually input into the database.

When the FS engineer tried to access the electronic file specified in the software Constants and Code Words Memo, he found it no longer existed at the specified location on the electronic file folder because it was now over a year after the file had been originally generated.[3] The FS engineer selected a different file as a baseline that only required him to change five I1 values for the digital roll rate filter (an algorithm with five constants). The filter was designed to prevent the Centaur from responding to the effects of Milstar fuel sloshing and inducing roll rate errors at 4 radians/second. During manual entry of those five I1 roll rate filter values, the LMA FS engineer incorrectly entered or missed the exponent for the I1(25) constant. The correct value of the I1(25) filter constant was -1.992476. The exponent should have been a one but instead was entered as a zero, making the entered constant one tenth of the intended value or -0.1992476. The flight software engineer's immediate supervisor did not check the manually entered values.

The only person who checked the manually input I1 filter rate values, besides the flight software engineer who actually input the data, was an LMA Control Dynamics engineer. The FS engineer who developed the Flight Load tape notified the CD engineer responsible for design of the first thirty I1 constants that the tape was completed and the printout of the constants was ready for inspection. The CD engineer went to the FS offices and looked at the hardcopy listing to perform the check and sign off the I1 constants. The manual and visual check consisted of comparing a list of I1 constants from Appendix C of the Software Constants and Code Words Memo to the paper printout from the Flight Load tape. The formats of the floating-point numbers (the decimal and exponent formats) were different on each of these paper documents for the three values cross-checked for each I1 constant. The CD engineer did not spot the exponent error for I1(25) and signed off that the I1 constants on the Flight Load tape were correct. He did not know that the design values had been inserted manually into the database used to build the flight tapes (remember, the values had been stored electronically but the original database no longer existed) and that they were never formally tested in any simulation prior to launch.

The CD engineer's immediate supervisor, the lead for the CD section, did not review the Signoff Report or catch the error. Once the incorrect filter constant went undetected in the Signoff Report, there were no other formal checks in the process to ensure the I1 filter rate values used in flight matched the designed filter.

[illegible]

- A process input was missing (the electronic file specified in the Software Constants and Code Words memo), so an engineer regenerated it, making a mistake in doing so.
- Inadequate control was exercised over the constants process. No specified or documented software process existed for electronically merging all the inputs into a single file. There was also no formal, documented process to check or verify the work of the flight software engineer in creating the file. Procedures for creating and updating the database were left up to the flight software engineer's discretion.

[3] Another example of migration of the system [illegible] design.

- Once the incorrect filter constant went undetected in the Signoff Report, there were no other formal checks in the process to ensure the I1 filter rate values used in flight matched the designed filter.

- The hazard analysis process was inadequate, and no control was exercised over the potential hazard of manually entering incorrect constants, a very common human error. If system safety engineers had identified the constants as critical, then a process would have existed for monitoring the generation of these critical variables. In fact, neither the existence of a system safety program nor any form of hazard analysis are mentioned in the accident report. If such a program had existed, one would think it would be mentioned.

  The report does say that Quality Assurance engineers performed a risk analysis, but they considered only those problems that had happened before.

  > Their risk analysis was not based on determining steps critical to mission success, but on how often problems previously surfaced in particular areas on past launches. They determined software constant generation was low risk because there had not been previous problems in that area. They only verified that the signoff report containing the constants had all the proper signatures [?].

  Considering only the causes of past accidents is not going to be effective for software problems or when new technology is introduced into a system. Computers are, in fact, introduced in order to make previously infeasible changes in functionality and design, which reduces the effectiveness of a "fly-fix-fly" approach to safety engineering. Proper hazard analyses examining all the ways the system components can contribute to an accident need to be performed.

Process Model F The accident report suggests that many of the various partners were confused about what the other groups were doing. The LMA software personnel who were responsible for creating the database (from which the flight tapes are generated) were not aware that IV&V testing did not use the as-flown (manually input) I1 filter constants in their verification and validation process. The LMA Control Dynamics engineer who designed the I1 rate filter also did not know that the design values were manually input into the database used to build the flight tapes and that the values were never formally tested in any simulation prior to launch.

While the failure of the LMA CD engineer who designed the I1 rate filter to find the error during his visual check was clearly related to the difficulty of checking long lists of differently formatted numbers, it also may have been partly due to less care being taken in the process due to an incorrect mental model, i.e., (1) he did not know the values were manually entered into the database (and were not from the electronic file he created), (2) he did not know the load tape was never formally tested in any simulation prior to launch, and (3) he was unaware the load tape constants were not used in the IV&V process.

Coordination: The fragmentation/stovepiping in the flight software development process, coupled with the lack of comprehensive and defined system and safety engineering processes, resulted in poor and inadequate communication and coordination among the many partners and subprocesses.

Because the IMS software was developed by Honeywell, most everyone (LMA control dynamics engineers, flight software engineers, product integrity engineers, SQA, IV&V, and DCMC personnel) focused on the FCS and had little knowledge of the IMS software.

7 Quality Assurance (QA)

Safety Constraint: QA must monitor the quality of all safety-critical processes.

Process Flaw: The internal LMA quality assurance processes did not detect the error in the role rate filter constant software file.

Control Algorithm Flaw: QA verified only that the signoff report containing the load tape constants had all the proper signatures, an obviously inadequate process. This accident is indicative of the problems with QA as generally practiced and why it is often ineffective. The LMA Quality Assurance Plan used was a top-level document that focused on verification of process completion, not on how the processes were executed or implemented. It was based on the original General Dynamics Quality Assurance Plan with recent updates to ensure compliance with ISO 9001. According to this plan, the LMA Software Quality Assurance staff was required only to verify that the signoff report containing the constants had all the proper signatures; they left the I1 constant generation and validation process to the flight software and control dynamics engineers. Software Quality Assurance involvement was limited to verification of software checksums and placing quality assurance stamps on the software products that were produced.

8 Developer Testing Process

Once the error was introduced into the load tape, it could potentially have been detected during verification and validation. Why did the very comprehensive and thorough developer and independent verification and validation process miss this error?

Safety Constraint Violated: Testing must be performed on the as-flown software (including load tape constants).

Flaws in the Process: The INU (FCS and IMS) was never tested using the actual constants on the load tape:

- Honeywell wrote and tested the IMS software, but they did not have the actual load tape.
- The LMA Flight Analogous Simulation Test (FAST) lab was responsible for system test, i.e., they tested the compatibility and functionality of the flight control software and the Honeywell IMS. But the FAST lab testing used a 300 Hertz filter simulation data file for IMS filters and not the flight tape values. The simulation data file was built from the original, correctly specified values of the designed constants (specified by the LMA CS engineer), not those entered by the software personnel in the generation of the flight load tape. Thus the mix of actual flight software and simulated filters used in the FAST testing did not contain the I1(25) error, and the error could not be detected by the internal LMA testing.

Process Model Mismatch: The testing capability that the current personnel thought the lab had did not match the real capability. The LMA FAST facility was used predominantly to test flight control software developed by LMA. The lab had been originally constructed with the capability to exercise the actual flight values for the I1 roll rate filter constants, but that capability was not widely known by the current FAST software engineers until after this accident; knowledge of the capability had been lost in the corporate consolidation/evolution process so the current software engineers used a set of default roll rate filter constants. Later it was determined that had they used the actual flight values in their simulations prior to launch, they would have caught the error.

9 Inpleffion an (I)V&V

dVSafety Constraint IV&V must be performed on the as-flown software and constants. All safety-critical data and software must be included in the IV&V process.

Dytefaciionalns: Each component of the IV&V process performed its function correctly, but the overall design of the process was flawed. In fact, it was designed in such a way that it was not capable of detecting the error in the role rate filter constant.

Analex was responsible for the overall IV&V effort of the flight software. In addition to designing the IV&V process, Analex-Denver performed the IV&V of the flight software to ensure the autopilot design was properly implemented in the software while Analex-Cleveland verified the design of the autopilot but not its implementation. The"ptrouthdedbaysliLnMeA, per agreement between LMA and Analex, was generated from the constants verified in the Signoff Report.

In testing the flight software implementation, Analex-Denver used IMS default values instead of the actual I1 constants contained on the flight tape. Generic or default I1 constants were used because they believed the actual I1 constants could not be adequately validated in their rigid body simulations, i.e., the rigid body simulation of the vehicle would not exercise the filters sufficiently.[4] They found out after the mission failure that had they used the actual I1 constants in their simulation, they would have found the order of magnitude error.

Analex-Denver also performed a range check of the program constants and the Class I flight constants and verified that format conversions were done correctly. However the process did not require Analex-Denver to check the accuracy of the numbers in the truth baseline, only to do a range check and a bit-to-bit comparison against the firing tables, which contained the wrong constant. Thus the format conversions they performed simply compared the incorrect I1(25) value in the firing tables to the incorrect I1(25) value after the conversion, and they matched. They did not verify that the designed I1 filter constants were the ones actually used on the flight tape.

Analex-Cleveland had responsibility for verifying the functionality of the design constant but not the actual constant loaded into the Centaur for flight. That is, they were validating the design onlyfpaltdendotsaiteonAnalex-Cleveland received the Flight Dynamics and Control Analysis Report (FDACAR) containing the correct value for the roll filter constant. Their function was to validate the autopilot design values provided in the FDACAR. That does not include IV&V of the I1 constants in the flight format. The original design work was correctly represented by the constants in the FDACAR. In other words, the filter constant in question was listed in the FDACAR with its correct value of -1.992476, and not the value on the flight tape (-0.1992476).

Cganithol Fwla Analex developed (with LMA and government approval) an IV&V program that did not verify or validate the I1 filter rate constants actually used in flight. The I1 constants file was not sent to Analex-Cleveland for autopilot validation because Analex-Cleveland only performed design validation. Analex-Denver used default values for testing and never validated the actual I1 constants used in flight.

[4] ewordsedNin tthat almost identical Ar5iaepeort.

Process Model Mismatches: The decision to use default values for testing (both by LMA FAST lab and by Analex-Denver) was based on a misunderstanding about the development and test environment and what was capable of being tested. Both the LMA FAST lab and Analex-Denver could have used the real load tape values, but did not think they could.

In addition, Analex-Denver, in designing the IV&V process, did not understand the generation or internal verification process for all the constants in the "baseline" provided to them by LMA. The Analex-Denver engineers were not aware that the I1 filter rate values provided originated from a manual input and might not be the same as those subjected to independent V&V by Analex-Cleveland.

None of the participants was aware that nobody was testing the software with the actual load tape values nor that the default values they used did not match the real values.

Coordination: This was a classic case of coordination problems. Responsibility was diffused among the various partners, without complete coverage. In the end, nobody tested the load tape and everyone thought someone else was doing it.

10 Systems Engineering

System engineering at LMA was responsible for the identification and allocation of the functionality to be included in the system. In fact, the software filter involved in the loss was not needed and should have been left out instead of being retained, yet another example of asynchronous evolution. Why was that decision made? The filter was designed to prevent the Centaur from responding to the effects of Milstar fuel sloshing and inducing roll rate errors at 4 radians/second. Early in the design phase of the first Milstar satellite, the manufacturer asked to filter that frequency. The satellite manufacturer subsequently determined filtering was not required at that frequency and informed LMA. However, LMA decided to leave the filter in place for the first and subsequent Milstar flights for "consistency."[5] No further explanation is included in the report.

11 LMA Project Management (as Prime Contractor)

Safety Constraint: Effective software development processes must be established and monitored. System safety processes must be created to identify and manage system hazards.

Context: The Centaur software process was developed early in the Titan/Centaur program. Many of the individuals who designed the original process were no longer involved in it due to corporate mergers and restructuring (e.g., Lockheed, Martin Marietta, General Dynamics) and the maturation and completion of the Titan IV design and development. Much of the system and process history and design rationale was lost with their departure.

Control Flaws:

- A flawed software development process was designed. For example, no process was provided for creating and validating the flight constants.

[5] This factor is similar to the Ariane 5 loss, where unnecessary software was retained "for consistency."

- LMA, as prime contractor, did not exert adequate control over the development process. The Accident Investigation Board could not identify a single process owner responsible for understanding, designing, documenting, or controlling configuration and ensuring proper execution of the process.
- An effective system safety program was not created.
- An inadequate IV&V program (designed by Analex-Denver) was approved and instituted that did not verify or validate the I1 filter rate constants used in flight.

Mental Nobody seemed to understand the overall software development process and apparently all had a misunderstanding about the coverage of the testing process.

12 Deefnse Contract Margan emnt Co mdan (IDCM C)

Cqeintsrol The report is vague about the role of DCMC, saying only that it was responsible for contract administration, software surveillance, and overseeing the development process. It does say that DCMC approved an IV&V process with incomplete coverage and that there was a software quality assurance function operating at DCMC, but it operated without a detailed understanding of the overall process or program and therefore was ineffective.

Cidnation: No information was provided in the accident report although coordination problems between SMC and DCMA may have been involved. Were each assuming the other was
?monitoring the overall process ? What role did Aerospace Corporation play Were there gaps in
?Howtheidrespsonsibilities assigned to each of the many groups providing oversight here
?oWvehrlaapp iofg freesdpbacnksi bdiildit iDeCsM fCi tugseetdh etro perform their
?process monitoring

13 Air Fgonce (Pro Office) Spadce an Missile Syms te s Center Launch DSiMrectorate (C)

Safety Constraint: SMC must ensure that the prime contractors creates an effective development and safety assurance program.

Cxonte Like 3SLS, the Air Forace Space and Missile System Center Launch Directorate was transitioning from a task oversight to a process insight role and had, at the same time, undergone personnel reductions.

Cgeiithol

- The SMC Launch Programs Directorate essentially had no personnel assigned to monitor or provide insight into the generation and verification of the software development process. The Program Office did have support from Aerospace to monitor the software development and test process, but that support had been cut by over 50 percent since 1994. The Titan Program Office had no permanently assigned civil service or military personnel nor full-time
Centsuppoar tsofwtwaka ethTeh eTyi tadnecided that because the Titan
"themsaotfutwrea,r ewsatsable, and had not experienced problems in the past
use their resources to address hardware issues.

- The transition from oversight to insight was not managed by a detailed plan. AF responsibilities under the insight concept had not been well defined, and requirements to perform those responsibilities had not been communicated to the workforce. In addition, implementation of the transition from an oversight role to an insight role was negatively affected by the lack of documentation and understanding of the software development and testing process. Similar flawed transitions to an “insight” role are a common factor in many recent aerospace accidents.

- The Titan Program Office did not impose any standards (e.g., Mil-Std-882) or process for safety. While one could argue about what particular safety standards and program could or should be imposed, it is clear from the complete lack of such a program that no guidance was provided. Effective control of safety requires that responsibility for safety be assigned at each level of the control structure. Eliminating this control leads to accidents. The report does not say whether responsibility for controlling safety was retained at the program office or whether it had been delegated to the prime contractor. But even if it had been delegated to LMA, the program office must provide overall leadership and monitoring of the effectiveness of the efforts. Clearly there was an inadequate safety program in this development and deployment project. Responsibility for detecting this omission lies with the program office.

In summary, understanding why this accident occurred and making the changes necessary to prevent future accidents requires more than simply identifying the proximate cause—a human error in transcribing long strings of digits. This type of error is well known and there should have been controls established throughout the process to detect and fix it. These controls were almost totally missing in the development and operations processes or they were inadequately designed and executed.