A Fault Management Strategy for Autonomous Rendezvous and Capture with the ISS

Russell Sargent[1], Ian Mitchell[2], Louis Breger[3],
David Benson[4], Chris Bessette[5] and Renato Zanetti[6]
Draper Laboratory, Cambridge, MA 02139

Joseph E. Groszkiewicz[7]
Orbital Sciences Corp., 21839 Atlantic Blvd., Dulles, VA 20166

The Cygnus vehicle, built by the Orbital Sciences Corporation, is being developed to perform autonomous rendezvous with International Space Station (ISS) and to provide cargo services to the faculty. Safety through fault management has been a primary consideration in the design of the trajectories, GN&C algorithms, and software. This paper describes the approach used to design and validate a fault management system required to meet the ISS visiting vehicle safety requirements. The nominal mission trajectory and abort maneuvers were designed using linear covariance analysis and were validated to provide a combination of passive and active collision avoidance and requirement satisfaction through semi-analytical methods and extensive simulation. Hardware faults, such as IMU, LIDAR, and GPS sensor faults, are managed using a highly reliable backup propagation system and Fault Detection and Isolation (FDI) algorithms. These algorithms ensure that key sensor systems are fault tolerant and that faulty measurement sources are detected and identified before they significantly corrupt the navigation system. Critically, the FDI system ensures that sufficient measurements will be available to execute an abort maneuver safely at any time. The IMU, LIDAR and GPS FDI algorithms employ standard parity methods to detect sensor measurement errors and a Maximum Likelihood Estimation (MLE) approach for fault identification when sufficient measurement redundancy exists. In the case of GPS FDI, parity space algorithms are utilized for GPS receiver redundancy as well as measurement redundancy within a particular GPS receiver. This allows independent detection and identification of receiver faults and measurement faults arising from GPS satellite faults or a degraded multipath environment. Fault thresholds for FDI were validated using Monte Carlo analysis in a high fidelity 6DOF simulation.

Nomenclature

AE	=	Approach Ellipsoid
AI	=	Approach Initiation
ATP	=	Authority To Proceed
ATV	=	Automated Transfer Vehicle
CCDL	=	Cross-Channel-Data-Link
CM	=	Crew Module
COTS	=	Commercial Orbital Transportation Services
CW	=	Clohessy Wiltshire
FDI	=	Fault Detection and Isolation

[1] Member of Technical Staff, Mission Design, 555 Technology Sq. Cambridge, MA, 02139
[2] Distinguished Member of Technical Staff, GN&C Systems, 17629 El Camino Real, #47, Houston TX, 77059
[3] Senior Member of Technical Staff, Tactical ISR, 555 Technology Sq. Cambridge, MA, 02139, AIAA member
[4] Senior Member of Technical Staff, Aerospace Guidance & Control, 555 Technology Sq. Cambridge, MA, 02139
[5] Senior Member of Technical Staff, Aerospace Guidance & Control, 555 Technology Sq. Cambridge, MA, 02139
[6] Senior Member of Technical Staff, Mission Design, 17629 El Camino Real, #47, Houston TX, 77059
[7] GN&C Lead Engineer, Orbital Sciences Corp., 21839 Atlantic Blvd., Dulles, VA 20166

GN&C	=	Guidance, Navigation, and Control
GPS	=	Global Positioning System
HP	=	Hold Point
HTV	=	H-II Transfer Vehicle
IRD	=	Interface Requirements Document
ISS	=	International Space Station
JEM	=	Japanese Experiment Module
KDR	=	Key Driving Requirement
KOS	=	Keep Out Sphere
LIDAR	=	LIght Detection And Ranging
LinCov	=	Linear Covariance
MLE	=	Maximum Likelihood Estimator
RAIM	=	Receiver Autonomous Integrity Monitoring
SCM	=	Safe Corridor Monitoring
SIGI	=	Space Integrated GPS/INS
SM	=	Service Module

I. Introduction

This paper demonstrates how the Cygnus addressed critical safety requirements using two fault tolerant sensors, and anytime abort trajectories. Key fault management requirements are explained and the validation actives are discussed.

Section II provides a mission overview which outlines Cygnus' approach to the ISS. This section will discuss the vehicle layout including sensors, actuators, and computer systems. In addition, this section will cover sensor utilization and the nominal approach trajectory in the context of driving safety requirements.

Section III focuses on navigation fault management. First an overview is provided of the navigation filter software and its interface to the sensor systems. This is followed by a description of the fault detection system for the IMU, LIDAR and GPS measurements. Key mathematical foundations are provided, as well as validation methodology.

Section IV is dedicated to the design of the abort maneuvers. This section explains how the onboard Safe Corridor Monitoring (SCM) software uses the fault tolerant navigation estimates from Section III to trigger aborts necessary to meet the key safety requirements in Section II. An overview of the abort verification methodology is also presented.

II. Cygnus Mission Overview

The Commercial Orbital Transportation Services (COTS) Cygnus vehicle will conduct an autonomous rendezvous with the ISS. The rendezvous and approach trajectory has been determined by the ISS to COTS Interface Requirements Document (IRD) requirements,[1] by the capabilities of the unmanned vehicle, and by the desire to remain safe in the event of an off-nominal situation. The mission concept of operations is shown in Figure 1.

A. Cygnus Architecture Overview

The Cygnus vehicle is composed of a Service Module (SM) and a Cargo Module (CM). The Cygnus GN&C subsystem consists of the two-fault tolerant flight control computer, Star Tracker, LIDAR Assembly, Space Integrated GPS/INS (SIGI), Three Axis Magnetometer, three independent strings of reaction control jets, and one main engine. During proximity operations the GN&C sensors are selected as a function of range. Figure 2 shows the navigation sensor utilization as a function of distance from ISS. The link with the PROX system onboard the ISS will be established approximately 50 km from ISS. Once the link is established, the PROX system transmits the JEM SIGI raw measurements to Cygnus. The onboard *Relative GPS* filter starts fusing the JEM SIGI and Cygnus SIGI raw measurements to generate the relative navigation solution. The Relative GPS filter is the primary source of relative measurements up to about 700 m, at which point the LIDAR starts providing range and bearing measurements to the *Relative LIDAR* filter. Once the quality of the LIDAR measurement has been established and the filter is converged to a stable solution, the LIDAR filter becomes the primary navigation system.

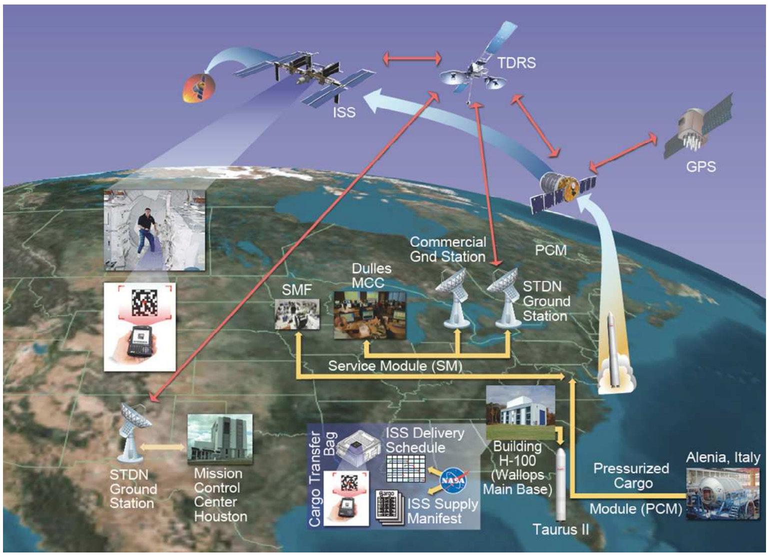


Figure 1 Mission Overview

The Flight Computer consists of four BAE RAD750 single-board computers interconnected via the Draper Network Element (NE) card and the Cross-Channel-Data-Link (CCDL). The NE asserts that all four computers operate upon identical (congruent) data and that all commands to effectors are agreed upon (voted) prior to being issued. All single and dual failures within the processing system are detected, isolated and removed.

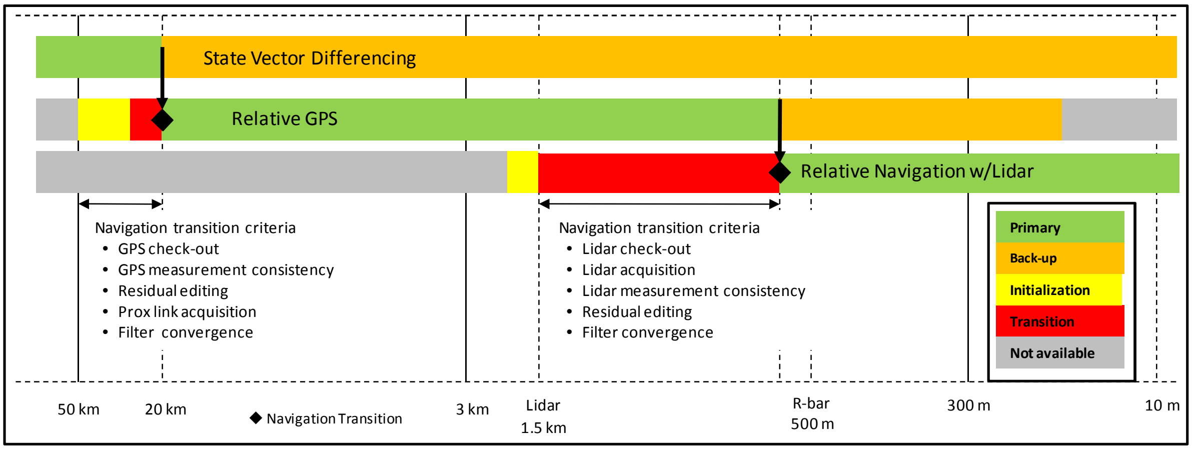


Figure 2 Navigation Sensor Utilization & Transition

B. Cygnus Approach to ISS

The Cygnus trajectory design and analysis considers three distinct mission phases: *Long Range Rendezvous*, *Approach*, and *Departure*. Each of the trajectories within these mission phases are designed to satisfy the requirements in the ISS to COTS IRD. Figure 3 shows the nominal approach trajectory and its major events.

The *Approach* mission phase begins approximately one hour before Cygnus is within range of the Proximity Communication System (PROX). Cygnus is on a co-elliptic orbit 4 km below ISS, having previously performed several maneuvers to adjust the height, orbit plane and phase with respect to the ISS. Upon entering PROX range, Cygnus establishes communication and begins to perform relative GPS navigation using the GPS receiver within the PROX subsystem on the ISS and the Cygnus GPS receiver. Cygnus will compute and execute a maneuver sequence to acquire a co-elliptic orbit 1.4 km below the ISS provided an Authority to Proceed (ATP-1) has been granted. Once on the 1.4 km co-elliptic, Cygnus will drift until arriving at the Approach Initiation (AI) point.

The final approach begins when ATP-2 is given for the ADV-3 burn to transfer to the R-bar. The Cygnus onboard targeting will compute and execute the ADV-3 maneuver. The ADV-3 maneuver, shown in Figure 4, is targeted to bring the Cygnus inside the AE and to arrive with a predefined velocity at a designated location on the R-bar below ISS. Prior to arriving at the R-bar, the LIDAR will have been powered on and the Cygnus vehicle will be appropriately oriented toward ISS in order to acquire and track the target reflectors. The transition from relative GPS navigation to LIDAR relative navigation will occur in the vicinity of the R-bar acquisition point. The Cygnus vehicle will ascend the R-bar at a predefined closing rate. Multiple Hold Points (HPs) are positioned on the R-bar prior to entering the Keep Out Sphere (KOS) and reaching the capture volume. Once in the capture volume, the SSRMS grapples Cygnus and securely connects the two vehicles. The two sides of the CBM then form a pressure tight seal, and the Cygnus vehicle will be electrically connected to ISS.

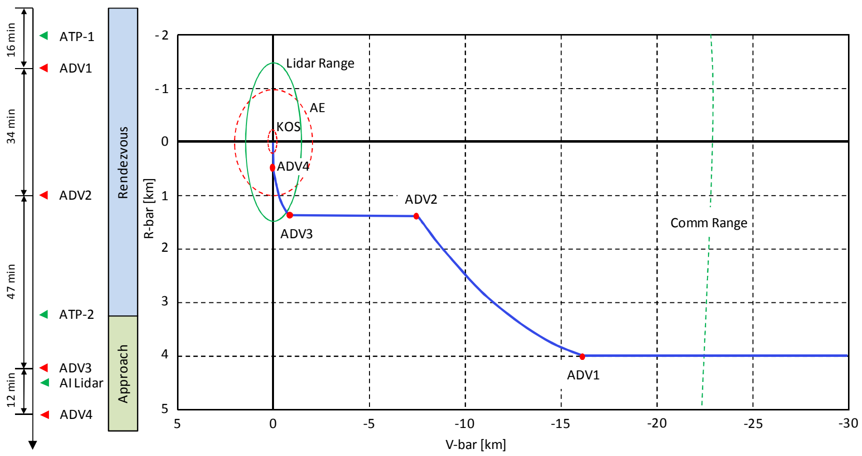


Figure 3 Cygnus Nominal Approach Trajectory

The departure mission phase begins with release of Cygnus from the Space Station Remote Manipulator System (SSRMS). The Cygnus will stay in free drift for about five minutes and after that it will perform a predefined burn. The burn is pre-calculated and designed to bring the Cygnus vehicle in front and above the ISS.

C. Overview of Safe Trajectory Design

Although the unmanned vehicles ATV, HTV, and Cygnus, follow different trajectories to rendezvous with ISS, all of those trajectories are subject to many of the same safety considerations. In this paper, a safe trajectory will be defined as a trajectory that will not violate trajectory requirements in the presence of a class of off-nominal conditions. Although the ATV follows a Vbar approach while the HTV and Cygnus follow an Rbar approach, the trajectories of all three vehicles must consider several common issues in order to meet the safety requirements. These elements are missed burns, partial burns, drag uncertainty, navigation error, and the ability to perform an abort safely. A missed burn is a burn that is planned, but does not occur. Missed burns are a critical consideration during early phases of the rendezvous approach. During the Cygnus mission, missed burns that occur within a certain distance of the ISS must be guaranteed to passively avoid the ISS for a minimum of 24 hours. A partial burn is a thrust that ends early or late, resulting in an underburn or an overburn, respectively. Trajectories are designed to meet minimum separation requirements from ISS in the presence of missed or partial burns by using orbits with drift biases that naturally cause the vehicles to separate. This task is complicated by atmospheric drag, which will either accelerate the drift rate or cause one vehicle to reverse its motion relative to the other. To compensate for drag effects, the drift biases must be adjusted. This increase is often at the expense of maneuver cost. The trajectory design is further complicated by uncertainty in the atmospheric drag and navigation error, both of which will result in trajectories that deviate from the nominal trajectory.

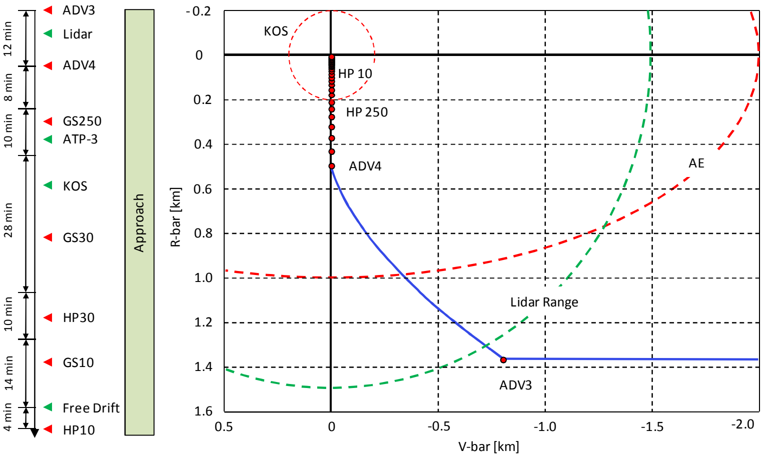


Figure 4 Cygnus Nominal R-bar Ascent Trajectory

1. *Cygnus Proximity Operation Trajectory Overview*

The Cygnus proximity operations trajectory is implemented using three types of trajectory generation and the following algorithms: Co-elliptic Transfers, Rbar Acquisition, and the Rbar Ascent. Co-elliptic transfers are planned using a converging algorithm that employs long-term horizon planning and a high fidelity dynamics model. The Rbar Acquisition Guidance Algorithm is used to implement ADV3 and is comprised of several different closed loop guidance modes, which require minimal computation and rely on the CW equations. The Rbar Ascent is performed using the Glideslope algorithm, a low-computation, closed-loop guidance routine, which yields predictable, constrained trajectories with low fuel use.

2. *Co-elliptic Transfers*

A targeting routine is used to calculate co-elliptic transfers on-orbit. This routine is first called in advance of the anticipated burn time in order to determine a burn attitude. As the Cygnus approaches the burn time, the targeting algorithm is called again to refine the burn duration estimate. Each co-elliptic transfer is comprised of an initiation burn to place the Cygnus on a transfer orbit and a termination burn to place the Cygnus on a new co-elliptic orbit relative to the ISS.

The key driving requirements during the rendezvous phase in which the targeting algorithm is used are:

1. The GN&C subsystem shall maintain a trajectory prior to Approach Initiation (AI) that remains outside of the Approach Ellipsoid (AE) including expected dispersions (99.73 percent of the trajectories with 50 percent confidence level).
2. The GN&C subsystem shall perform an Approach Initiation (AI) maneuver that results in the vehicle trajectory remaining outside of the Keep Out Sphere (KOS), including expected dispersions (99.73 percent of the trajectories with 50 percent confidence level).
3. The GN&C subsystem shall compute translational maneuvers prior to Approach Initiation (AI) such that any resulting free drift trajectory during or after the execution of such a maneuver remains outside of the AE for a minimum of 24 hours.
4. The GN&C subsystem shall perform rendezvous and proximity maneuvers to arrive at the capture volume within 6 hours or less from the start of Integrated Operations (IO).
5. The GN&C subsystem shall perform the dual Co-Elliptic Transfer (CT) maneuver provided authorization to proceed has been received from mission control.
6. The GN&C subsystem shall compute translational maneuvers prior to receiving authorization to enter the KOS such that any resulting free-drift trajectory during or after the execution of such a maneuver remains outside of the KOS for a minimum of 4 orbits.

3. *Rbar Acquisition*

The ADV3 maneuver is executed using Rbar Acquisition Closed-loop Guidance. This routine operates by calculating delta-V commands to place a vehicle on a reference trajectory.[2] The reference trajectory is chosen by back-propagating using the CW equations from a position and velocity state on the glideslope path. When the Cygnus is behind (i.e., the Vbar direction) the reference trajectory, a phantom-targeting algorithm is used to shift the targeted point after each time step to correct for any dispersions that occur due to process noise or navigation error. If the Cygnus is ahead of the reference trajectory, vertical tangent targeting routine is used to guide the vehicle back to the glideslope using a bisection search algorithm.

The key driving requirements during the Rbar Acquisition phase are:

1. The GN&C subsystem shall perform the Approach Initiation (AI) maneuver provided authorization to proceed has been received from mission control.
2. The GN&C subsystem shall perform an Approach Initiation (AI) maneuver that results in the vehicle trajectory remaining outside of the Keep Out Sphere (KOS) including expected dispersions.

4. *Rbar Ascent*

The Rbar Ascent portion of the approach trajectory uses the glideslope algorithm[2]. The glideslope algorithm provides a computationally-minimal, economical, and easily implemented method for adhering to a steadily shrinking approach corridor using a fixed burn rate. The reference trajectory for a glideslope approach is found by reverse propagating from the final range and range-rate along the glideslope path using the CW equations.

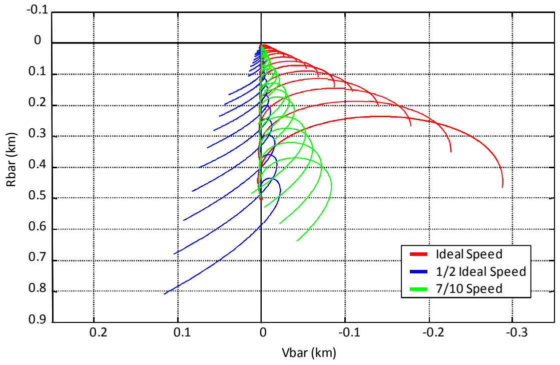


Figure 5 Free-drift Trajectories for Varying Glideslope Ascent Rates

The key driving requirements during the Rbar Ascent phase in which the glideslope algorithm is used are:

1. The GN&C subsystem shall perform the Approach Continuation (AC) maneuver provided authorization to proceed has been received from mission control.
2. The GN&C subsystem shall maintain a trajectory prior to Approach Continuation (AC) that remains outside of the Keep Out Sphere (KOS) including expected dispersions.
3. The GN&C subsystem shall perform approach maneuvers to arrive at the capture volume within 80 minutes or less from initial R-Bar acquisition.
4. The GN&C subsystem shall maintain a trajectory after Approach Continuation (AC) that remains within the assigned approach corridor during nominal approach.

Figure 5 shows the effect of going into a free-drift at various points in the glideslope approach trajectory for several gain scaling selections.

III. Navigation Hardware Fault Management

This section describes the three levels of navigation fault management applied to each measurement. First, each sensor system performs internal integrity tests to ensure that the measurement is meaningful and that the sensor is operating as expected. If all sensor system tests pass, a positive validity flag is included with the measurements. The second level of navigation fault management is the Fault Detection and Isolation (FDI) algorithms. These algorithms check that the redundant and independent measurements are self consistent using parity space techniques. If the measurements are not self consistent, FDI detects a fault and isolates the faulty sensor. After the measurements are successfully screened by the FDI algorithms, the selected sensor signal is passed on to the navigation filter, where the third level of navigation fault management is performed. Before the measurements are processed to provide a navigation solution, the navigation filter compares the solution to the expected measurement using a process known as residual edits.

A. Navigation Algorithm Overview

The navigation subsystem design consists of two filters, one processing GPS measurements and the other processing LIDAR measurements. The actual performance of the LIDAR sensor is known only within its operating range. Thus, for safety concerns, it is desirable to independently validate the results of the LIDAR navigation filter prior to utilizing its solution to guide the vehicle to the ISS. Therefore the two-filter architecture is an integral part of navigation fault management, in the sense that it is designed to identify LIDAR failures.

The GPS filter is designed to satisfy requirements that ensure passive collision avoidance, as well as mission success. The GPS filter requirements are derived such that the navigation estimate will not cause targeting to command a maneuver that will take the vehicle inside the approach ellipsoid. The LIDAR filter requirements are derived such that the vehicle passively avoids entering the keep out sphere if a failure occurs prior to the 250 meter hold point. At the capture volume, the requirements are driven by bump analysis to ensure collision avoidance with the ISS. The requirements are linearly interpolated between 250 meters and the capture volume.

The GPS filter is a dual inertial filter that provides an estimate of the relative position and velocity between Cygnus and ISS. The GPS filter can operate in absolute mode or in relative mode. When in absolute mode, only the inertial state of Cygnus is estimated. The filter operates in absolute mode before the ISS state can be initialized. The ISS state is initialized with a ground update, which consists of position and velocity of the ISS at a future time. When the ISS state from the ground update becomes current, the filter transitions to relative mode and accepts ISS GPS measurements through the communication link. At any time, the Cygnus state can be re-initialized using the PVT solution from one of the onboard receivers. The filter can also be reset at any time to either absolute mode or idle mode. A reset is needed to accommodate off-nominal scenarios, for example a faulty ground update that needs to be overwritten. When in idle mode, the Cygnus state can be initialized with a ground update. In order to validate the pseudorange measurements, FDI estimates the receivers' clock biases, which are also used by the filter. Two valid consecutive estimates of the bias from FDI are needed before the filter can process measurements; these values are used to initialize the filter's estimated clock bias and drift. Since the clock bias and drift are unique to each receiver, it is important that FDI always selects the same receiver unless a failure occurs.

The LIDAR filter is initialized using the GPS filter estimate. Biases on the LIDAR estimate are estimated and carried in the filter's state, and are initially set to zero. These biases are also set to zero each time FDI switches to a different LIDAR, because these errors are specific to each sensor. FDI always selects the same LIDAR unless a failure occurs.

The navigation filter carries an estimated state that is propagated between measurements, this integrated state is denoted as $\hat{\boldsymbol{x}}^-$. Together with the state, the filter also carries a propagated estimation error covariance matrix, $\boldsymbol{P}^-$. The covariance matrix is symmetric positive definite and represents the uncertainty of the propagated estimate; a larger covariance indicates a less precise knowledge of the state. A covariance matrix is also associated with the measurement; again a larger covariance indicates a less accurate measurement. The navigation filter performs a weighted average of the propagated state and the measurement. The weights are given by the inverses of the covariance matrixes. The filter processes one measurement at a time through a scalar update. The nonlinear model of the scalar measurement, *z*, is given by (1)

$$z = h(\boldsymbol{x}) + n \tag{1}$$

Where *h* is a known nonlinear function of the true state vector, $\boldsymbol{x}$, and *n* is the zero-mean measurement noise with variance $\sigma_n{}^2$. The expected value of the measurement is given by

$$\hat{z} = h(\hat{\boldsymbol{x}}^-) \tag{2}$$

When a measurement becomes available, its value is compared to the expected value to test its validity. This process is called residual editing[3]. The measurement residual is given by

$$\tilde{z} = z - \hat{z} \tag{4}$$

When the a priori estimation error $\boldsymbol{e}^- = \boldsymbol{x} - \hat{\boldsymbol{x}}^-$ is small and zero mean (two common assumptions), the measurement residual is also approximately zero mean and can be expanded in a Taylor series centered at $\hat{\boldsymbol{x}}^-$ to obtain

$$\tilde{z} = h(\boldsymbol{x}) + n - h(\hat{\boldsymbol{x}}^-) \approx h(\hat{\boldsymbol{x}}^-) + \boldsymbol{H}\boldsymbol{e}^- + n - h(\hat{\boldsymbol{x}}^-) = \boldsymbol{H}\boldsymbol{e}^- + n \tag{5}$$

where $\boldsymbol{H}$ is the observation partial matrix about $\hat{\boldsymbol{x}}^-$ (i.e. the measurement matrix). The measurement residual variance is therefore given by

$$\sigma_m{}^2 = \boldsymbol{H}\boldsymbol{P}^-\boldsymbol{H}^T + \sigma_n{}^2 \tag{6}$$

The numerical value of the residual is tested to assure it lies between a predetermined range of its standard deviation. If the test fails, the measurement is rejected, otherwise the measurement is processed. For LIDAR measurements, all three scalar components (range, azimuth, or elevation) are rejected when any single component is failed by FDI. For GPS measurements, only the failed pseudoranges are rejected. Any valid pseudoranges are not affected and are the processed by the navigation filter. In each filter a counter keeps track of how many measurements have been rejected; this information is available to the ground to identify failures.

B. Fault Detection and Isolation Algorithm

The purpose of this section is to provide a brief derivation of the FDI algorithm, which monitors redundant measurements and analyzes the discrepancies between them. This algorithm is detailed in depth in multiple sources[4,5,6], which describe the GPS Receiver Autonomous Integrity Monitoring (RAIM) for aeronautic applications. For convenience, the FDI algorithm will be partitioned into two sections: fault detection and fault identification. Fault detection strictly refers to the determination of a fault, without identifying the offending sensor. Fault identification is invoked when a fault has been detected and is tasked with determining the culpable instrument.

1. Fault Detection Algorithm

Presume in Eq. (1) there are *M* independent measurements, ***z***, of some set of *N* states, where *M* is greater than *N* and the measurements span the space of *N*. Furthermore, assume that measurements are susceptible to faults. A failure of measurement *i* is modeled by $\boldsymbol{b} = \boldsymbol{b}_i$ where $\boldsymbol{b}_i$ is an *M* x 1 vector with the *i*th element non-zero and zero elsewhere. Accordingly, the measurement vector, ***z***, can be approximated as:

$$\boldsymbol{z} \approx \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} + \boldsymbol{b} \tag{7}$$

The generalized inverse matrix of $\boldsymbol{H}$ is defined as,

$$\boldsymbol{H}^* = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T \tag{8}$$

The parity space has three key characteristics:

1. The parity space vector, ***p***, is independent of the state vector, ***x***
2. If there is no fault, then ***b*** = **0**, the expected value of ***p*** is 0.
3. If there is a fault, the expected value of ***p*** is a function of ***b***.

Note that the parity space method can identify excessive faults, ***b***, present in the measurement vector, ***z***, even when the sensor being analyzed is producing non-zero mean data.

The ***A***-matrix is used to partition the measurement space into the state space and parity space. The ***K***-matrix (not the Kalman gain) is the matrix which spans the null-space of ***H***-matrix, ***KH*** = 0. Additionally, ***K*** is defined such that rank [***K***] = *M* – *N*, and $\boldsymbol{K}\,\boldsymbol{K}^T$=0.

$$\left[\frac{\text{State Space}}{\text{Parity Space}}\right] = \boldsymbol{A}\left[\text{Measurement Space}\right]$$
$$\text{where} \tag{9}$$
$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{H}^*\\ \boldsymbol{K}\end{bmatrix}$$

When the inverse of the ***A***-matrix is used to transform the parity space vector to the measurement space, the fault vector, ***f***, is calculated.

$$\boldsymbol{f} = \begin{bmatrix}\boldsymbol{H} & \boldsymbol{K}^T\end{bmatrix}\begin{bmatrix}0\\ \boldsymbol{p}\end{bmatrix} = \boldsymbol{A}^{-1}\begin{bmatrix}0\\ \boldsymbol{p}\end{bmatrix}$$
$$\boldsymbol{f} = \boldsymbol{Sz} \tag{10}$$
$$\boldsymbol{S} = \boldsymbol{I}_M - \boldsymbol{HH}^*$$

Refer to Ref. 4 for a derivation of the *S*-matrix. Note that the fault vector can be calculated by the flight computer using only the measurement vector, ***z***, and the measurement matrix, ***H***.

The ***f***-vector is used to detect faults because its expected value is the measurement fault vector mapped from state space to parity space. The fault vector covariance is the covariance of the sensor errors mapped from state space into measurements space. Thus, for a set of redundant measurements, the fault vector will increase in magnitude when the fault is significantly larger than the measurement noise.

$$E[\,\boldsymbol{f}\,] = \boldsymbol{Sb}$$
$$COV[\,\boldsymbol{f}\,] = \sigma_n^2\boldsymbol{S} \tag{11}$$

Next compute the scalar decision variable, *D*, as the squared magnitude of the fault vector:

$$D = \boldsymbol{f}^T\boldsymbol{f} \tag{12}$$

Using hypothesis testing, the FDI algorithm determines that a fault has occurred when the fault vector is larger than a predetermined fault threshold, *T*. The fault threshold is a function of the measurement noise variance, the probability of false alarm, and the difference between the length of the measurement vector and the length of the state vector.

$$T = \sigma_n^2 \mathrm{X}^{-1}\left(P_{FA} \mid M - N\right)$$
$$\chi^{-1} \equiv \text{Inverse Chi - Squared Distribution} \tag{13}$$
$$P_{FA} \equiv \text{Probability of False Detection}$$

Note that the fault threshold is a function of the measurement redundancy (*M-N*). As a result there will be different fault thresholds for two redundant and single redundant measurement sets. Another important conclusion from Eq. (13) is that a fault can be detected if at least one redundant measurement exists. However, as will be learned in the subsequent section, fault identification requires two redundant measurements.

2. *Fault Identification Algorithm: Maximum Likely Estimator*

The previous section explained how to determine if a fault occurred. This section explains the procedure for determining which instrument is culpable. The maximum likelihood estimator (MLE) identifies the ith measurement as being faulty if

$$P(\boldsymbol{p} \mid \boldsymbol{b}_i) = \max_j \left\{P(\boldsymbol{p} \mid \boldsymbol{b}_j)\right\} \tag{14}$$

The appendix in Ref. 4 shows that the measurement that maximizes $P(p \mid b_j)$ also maximizes $\frac{f_k^2}{S_{kk}}$. Thus, once a fault has been detected, *D* > *T*, the MLE identification algorithm is defined as:

$$i = \underset{k=1}{\overset{M}{MAX}}\left(\frac{\boldsymbol{f}_k^2}{\boldsymbol{S}_{kk}}\right) \quad \text{for } (M - N) \geq 2 \tag{15}$$

The $\boldsymbol{S}_{kk}$ values can be thought of as the relative parity space observability of the k^{th} measurement, which is the extent to which the k^{th} measurement is redundant with the other measurements. Thus, the MLE is essentially identifying the measurement with the largest fault vector element, as normalized according to its sensor geometry.

Note that two or more redundant measurements are required to perform fault identification. Imagine an inertial navigation system that contains three *x*-axis gyros for redundancy. If one of the gyros introduces a large error, the fault can be detected and identified because one of the three gyros will not be in agreement. However, if there are only two valid *x*-axis gyros, there is only one redundant measurement. If a fault is introduced, the two gyros measurements will not be in agreement. Thus a fault can be detected, but the offending gyro cannot be identified.

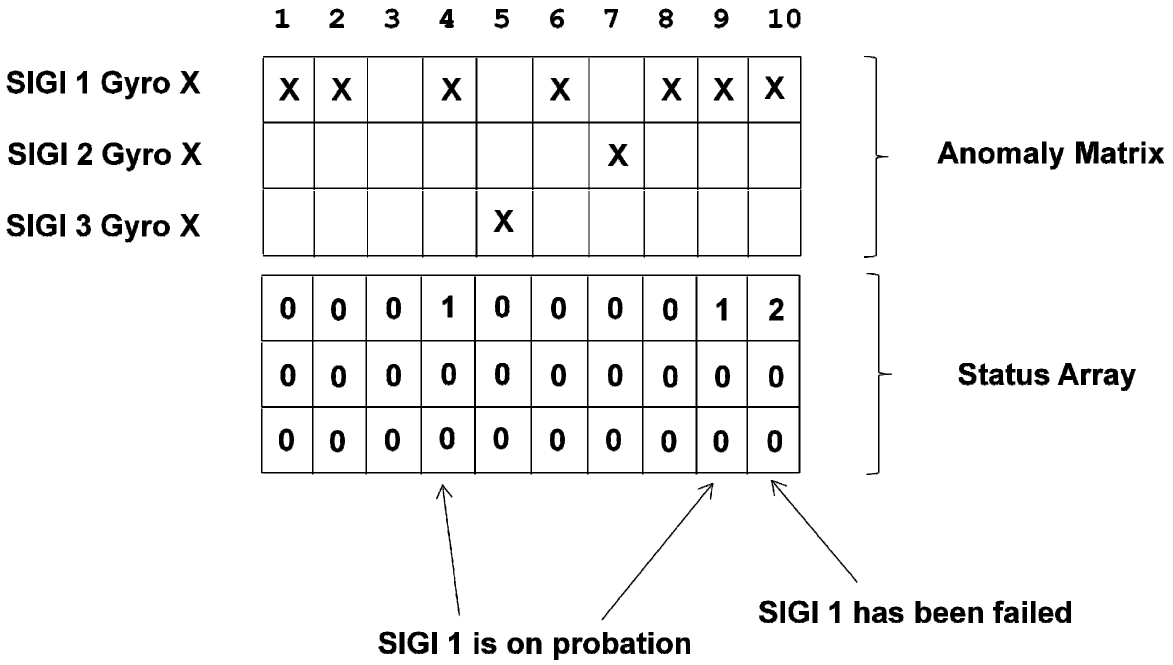
3. *Sensor Status*

The previous two sections explained how to determine if a fault has occurred and then how to determine which sensor is responsible. The COTS software contains several layers of fault monitoring. The first layer is self-monitoring by the sensor firmware, the second is the FDI routine, and the final layer is residual monitoring being performed by the navigation filter. The FDI routine will not typically fail a sensor permanently using a single “snapshot” of data containing an anomalous measurement. Instead a 'moving window” approach is used, which tracks the number of faults a sensor has accrued over a fixed interval. As a result, a two redundant sensor could have any of the following statuses:

1. Nominal
2. Probation
3. Failed
4. No Solution (GPS FDI only)

If a sensor has accrued more than n_{prbn} faults within m_{prbn} samples, then the sensor has been placed on probation. If a sensor has been placed on probation but subsequently outputs clean data, it is returned to nominal status. Likewise if a sensor has accrued more than n_{fail} faults within m_{fail} samples, the sensor is failed permanently. Once a sensor has been failed, FDI invalidates the measurements, prompting the mission manager to select another valid sensor. The only way for the sensor to return to nominal status is for the FDI algorithm to be reinitialized. If a single redundant measurement exists and $D > T$, the FDI routine will issue a secondary fault. Similar to the earlier case, a secondary fault flag will invalidate the measurements sent to the navigation filter. No solution applies only to GPS measurements.

In the following example there are three valid Gyro sensors, where $m_{\text{prbn}} = 3$, $n_{\text{prbn}} = 4$, $m_{\text{fail}} = 4$, and $n_{\text{fail}} = 5$. An “X” denotes that a single fault has occurred.



	1	2	3	4	5	6	7	8	9	10
SIGI 1 Gyro X	X	X		X		X		X	X	X
SIGI 2 Gyro X							X			
SIGI 3 Gyro X					X					

0	0	0	1	0	0	0	0	1	2
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure 6 Moving Window Example

In Figure 6, the X-Gyro in SIGI 1 is placed on probation after the fourth time step, as three of the four samples were faults. After nominal output on time step 5, SIGI 1 is returned to nominal status, as only two of the previous four samples were faults. After the ninth time step, SIGI 1 is placed on probation. Due to a subsequent fault on time step 10, SIGI 1 is placed on permanent failed status. To accommodate off-nominal scenarios the FDI algorithm contains a reset switch. If a reset is commanded, all anomaly matrix values are reset and all the sensor statuses are returned to nominal.

C. Fault Management Validation

The fault management algorithms for the accelerometer, gyro, LIDAR and GPS sensor systems are validated using a similar methodology. This section presents an overview of that validation process. The following section will describe the FDI application to individual sensor systems. This section is divided into three subsections: the first subsection discusses the process for determining the fault threshold values. The second subsection provides an overview of the COTS simulation tool. The third subsection describes the validation methodology.

1. Fault Threshold Analysis

The sensor fault thresholds are set using both analytical and Monte-Carlo analysis techniques. The lower bound threshold, T_{min}, for the each threshold is determined using Eq. 13, with a 10% probability of a false detection. Since the FDI routine is repeated for each measurement, threshold values lower than T_{min} will produce excessive false detections over the mission duration. The threshold upper bound T_{max}, is defined by the maximum amount of measurement error permitted before critical safety requirements begin to fail. T_{max} establishes the minimum sensor error that the FDI needs to detect and isolate from the system. Threshold values larger than T_{max} may not detect significant faults and will produce excessive missed detections.

The threshold upper bound, T_{max}, is determined using sets of Monte-Carlo runs with increasing levels of IMU sensor errors. The Monte-Carlo runs are performed using the COTS 6-DOF simulation tool described in the next section. Using the simulation tool, the select sensor noise sources are systematically increased to determine the maximum allowable measurement error. First, a baseline set of Monte-Carlo runs is performed with the nominal error level. The measurement error is systematically increased on one of the sensors, and the Monte-Carlo is then repeated. As shown in Figure 7, multiple Monte-Carlo sets are performed with increasing errors until the critical safety requirements begin to fail.

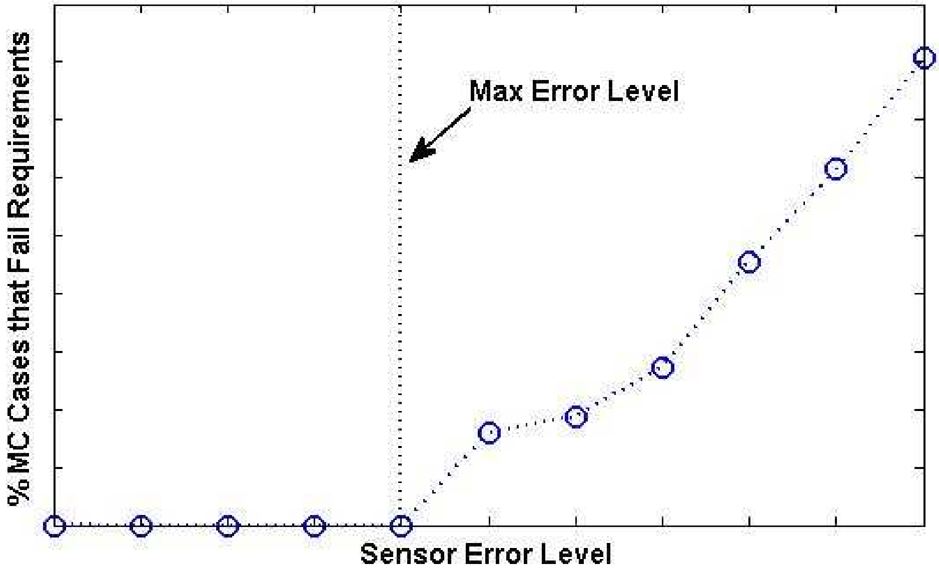


Figure 7 Maximum Sensor Error Determination

Once both T_{min} and T_{max} are determined, the sensor noise parameter in Eq. 13 and scan window size are tuned until an acceptable number of false detections and missed detections occur in a Monte-Carlo set in the presence of nominal GPS sensor noise. Establishing the threshold level in this way minimizes the risk of false detections and avoids overly conservative constraints on sensor performance.

2. *Simulation and Monte-Carlo Analysis*

The Monte-Carlo runs are performed using the COTS 6-DOF simulation tool and the Monte-Carlo framework. The simulation tool was developed using a Model Based Design process in Simulink and independently simulates the dynamics, sensors, and control of both the ISS and Cygnus vehicles.[7] The simulated Cygnus vehicle flight software runs auto-generated C-code, created from the Simulink models that implement the flight algorithms for navigation, guidance, and targeting. C-code for the whole simulation is automatically generated and run within the Monte-Carlo framework to do performance and validation analysis. Running the simulation inside this framework permits analysis to be done in the presence of high-fidelity disturbances and uncertainty models. Disturbances and uncertainty models are randomized for each simulation and run the prescribed number of times to conduct a Monte-Carlo study.

The IMU accelerometer error model is composed of a variety of error sources, including: scale factor errors, g-squared, asymmetry, orthogonal and non-orthogonal alignment errors, long-term bias, short term bias, and velocity random walk. The short term bias is approximated using a discrete time Markov process characterized by the time scale and variance. The IMU gyro error model is also composed of several parts, including: scale factor errors, non-orthogonality alignment, long-term bias, short-term bias, and angle random walk. The LIDAR error model is composed of short-term bias and random noise for the azimuth, elevation and range measurements. The short term bias is approximated using a discrete time Markov process characterized by the time scale and variance.

The simulation tool models the GPS constellation's geometry and contains a GPS error model composed of a variety of error sources. Error sources include ionosphere signal delays, troposphere signal delays, receiver clock noise, channel clock noise, GPS satellite clock bias, multipath signal delays and GPS satellite ephemeris errors. Receiver clock noise and bias are unique to each receiver and are added equally to each channel. Channel clock noise and multipath delays are modeled by a Markov process and are unique to each channel within a receiver. A complete discussion of a GPS error model can be found in Ref. 8.

3. *Fault Management Validation*

Fault management validation is done using a large set of Monte-Carlo runs where measurement errors corresponding to T_{max} are injected in addition to the nominal measurement noise. Each Monte-Carlo case has one injected error that is placed on a sensor starting from a random time. The injected error continues to be added to the measurements for the remainder of the simulation. In addition, a second failure is placed on a different sensor at a second random time that occurs after the first time by some minimum amount. This is to avoid injecting the second error before the FDI has completed identification of the first error. For each Monte-Carlo run, the time that the FDI correctly detects and identifies the first injected failure is logged. Any false detection, missed detections or incorrect identifications are also logged. The FDI is determined to be successful if the first failure is correctly identified within a reasonable amount of time in a predetermined number of Monte-Carlo cases, as shown in Figure 8. Success of the FDI is also subject to correct detection of the second failure.

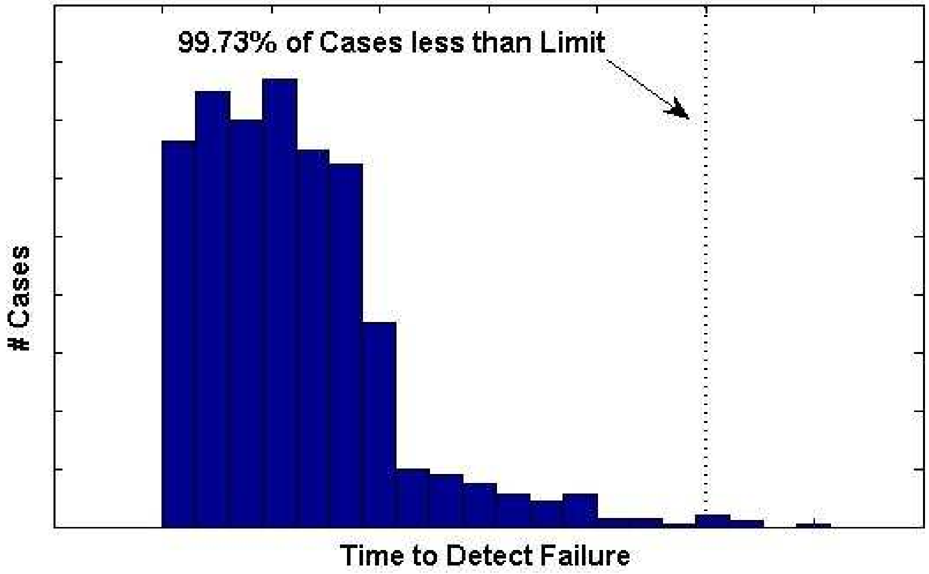


Figure 8 Time to Detect Failure Histogram

D. IMU and LIDAR Fault Detection and Isolation

For the IMU, fault detection is applied independently to each axis (X, Y, and Z in the IMU frame) of the accelerometers and gyros. Therefore, there are six independent applications of the FDI algorithm within the IMU FDI. The IMU sensor orientation within the body frame of the vehicle is known a priori and therefore the measurement matrix $\boldsymbol{H}$ in Eq. (7) is also known. This allows the $\boldsymbol{S}$ matrix in Eq. (10) to be pre-computed and loaded into the FDI algorithm as a parameter. To determine T_{max}, two types of IMU errors are systematically increased: the short-term bias and random walk noise for both accelerometers and gyros. Each error type on the two instruments is examined independently; axes are also examined independently. Therefore, there are 6 test runs total for accelerometers and 6 for gyros; one for each axis X, Y, Z and one for each error type. During IMU FDI validation, each Monte-Carlo case has one injected error that is placed on a random axis (X, Y, or Z), a random instrument, accelerometer or gyro, and a random sensor (1, 2, or 3). The second failure is placed on a different sensor, but on the same instrument.

For the LIDAR, fault detection is applied independently to the azimuth, elevation, and range measurements; therefore, there are three independent applications of the FDI algorithm within the LIDAR FDI. Similarly to the IMU FDI, the LIDAR sensor orientations are known a priori and therefore the S matrix can be pre-computed and stored as a parameter to the LIDAR FDI. To determine T_{max}, two types of LIDAR errors are systematically increased: the short-term bias and random noise for the azimuth, elevation and range measurements. Each error type is examined independently; measurement types are also examined independently. Therefore, there are 6 test runs total for the three measurement types and two for the error types. During the LIDAR validation, each Monte-Carlo case has one injected error that is placed on a random measurement type (azimuth, elevation, or range), and a random sensor (1, 2, or 3). The second failure is placed on a different sensor, but on the same measurement type.

E. GPS Receiver Fault Detection and Isolation

GPS measurements are screened by two separate levels of FDI: receiver level and channel level. The receiver level FDI checks for GPS receiver hardware failures by analyzing discrepancies between the sensors. However, receiver FDI alone does not ensure fault tolerance. Individual GPS signals can be corrupted due to multipath or GPS satellite clock errors. Since these errors are common to all GPS sensors, the receiver level FDI may not detect such faults. These faults can be detected by the GPS channel level FDI, which analyzes discrepancies between the GPS signals on a single receiver. This section describes the FDI algorithm application to the GPS receiver level analysis and validation methodology. The next section will cover the channel level FDI.

As described in the introduction, the Cygnus vehicle contains three separate SIGI units. Each SIGI contains a 12 channel GPS receiver, meaning that up to $M \leq 12$ measurements can occur. The FDI algorithm is applied to each of the 12 channels separately; if any GPS signal is not consistent across all valid receivers, receiver FDI detects an anomaly. Since the FDI algorithm is applied to the same GPS signal, $\boldsymbol{H}$ in Eq. (7) is a unity vector. This allows the $\boldsymbol{S}$ matrix in Eq. (10) to be pre-computed and loaded into the FDI algorithm as a parameter. Unlike the IMU and LIDAR FDI, the GPS measurements cannot be directly compared by FDI because each receiver has a different clock bias.

Before proceeding, it is necessary to briefly describe the GPS measurement process. The receiver matches the code pulse sequence to determine the signal transmission time. This is the time at which the signal was sent as measured by the GPS satellite's atomic clock (GPS system time). In addition, the receiver determinates the location of each satellite using the ephemeris information encoded on the signal. By comparing the transmission time with its own clock, the receiver computes the transit time and, multiplying by the speed of light, determines the range to each satellite. These measurements are termed pseudoranges because the actual range is corrupted by the receiver's clock bias. This is illustrated by the following equation:

$$\rho_k = \left|\boldsymbol{r} - \boldsymbol{s}_k\right| + b_{\text{clock}} \qquad (\text{k}=1,2,3,...\text{m}) \tag{16}$$

Thus, given the satellite positions and the pseudorange measurements ($\boldsymbol{s}_k$ and ρ_k), the GPS problem is to determine the receiver's position and clock bias ($\boldsymbol{r}$ and b_{clock}). The receiver clock bias contribution to the pseudorange measurement may be expressed as:

$$b_{\text{clock}} = c(t - \tau) \tag{17}$$

where c is the speed of light, t the user clock time, and τ the GPS transition time.

Before the pseudoranges are sent to the FDI algorithm, the receiver level must perform two operations. First the position and clock bias estimates ($\hat{\boldsymbol{r}}$ and $\hat{b}_{\text{clock}}$) are calculated for each valid receiver using Bancroft's method,[9] which is algebraic, computationally efficient, and numerically stable. Once the clock bias has been successfully estimated, the corrected pseudorange, $\rho_k - \hat{b}_{\text{clock}}$, is calculated. The corrected pseudorange defines the receiver level measurement vector, $\boldsymbol{z}$, in Eq. (7) .The receiver level state vector, $\boldsymbol{x}$, is defined as the true range, $|\boldsymbol{r}\text{-}\boldsymbol{s}_k|$. In addition, if the receiver signal time stamps are not identical, the corrected pseudoranges must be time synchronized. This is done using a forward Euler method and the range rate measurement, $\dot{\rho}_k$, provided by the SIGI. These two operations ensure that the FDI algorithm analyzes similar data across all valid receivers. As mentioned previously, the FDI clock bias estimate is sent as an input to the navigation filter.

At least four separate GPS signals are required to estimate the receiver position vector and clock bias. As a result, occasionally a GPS solution cannot be calculated due to too few satellites in the receiver's field of view. In addition, poor GPS satellite geometry may significantly reduce the numerical precision of the position-clock bias estimate. As is widely known,[5] the effect of satellite geometry on position-clock bias estimate accuracy can be calculated using the Geometric Dilution of Precision (GDOP). Whenever the GDOP becomes larger than a predetermined threshold or the number of available satellites becomes less than four, no reliable solution is possible and the receiver level will issue an FDI status of no solution. In this situation, no data is added to the anomaly matrix and the GPS data is withheld from the GPS filter. Once the satellite geometry improves, the receiver FDI will resume screening receivers.

Both channel level and receiver level FDI determine the T_{max} by systematically increasing channel level clock noise. The additional noise is added to a single valid channel on the selected receiver. During channel level FDI validation, each Monte-Carlo case injected one error on a random GPS satellite in the receiver's field of view (1-12) on a random receiver (1, 2, or 3). The injected error is added to the GPS signal for as long as the selected GPS satellite is tracked by the receiver. The second failure is placed on a different GPS satellite and on a different receiver. If either failed GPS satellite leaves the receiver field of view within 50 seconds of adding the error, the Monte-Carlo run is considered invalid and the results are not included in the analysis. This ensures the receiver level FDI is permitted sufficient time to detect and ID the faulty receiver.

F. GPS Channel Integrity Monitoring

The previous section discussed the method through which the FDI algorithm is applied to the GPS receivers to ensure that the redundant SIGIs produce consistent measurements. The following channel level discussion describes how the FDI algorithm is applied to a single GPS receiver to ensure that the channel level information is self-consistent. This analysis can be performed, because the GPS receiver has more than four satellites in the field of view during the vast majority of proximity operations, creating redundant GPS pseudorange measurements which can be used to perform FDI. Unlike previous FDI applications discussed in this work, the GPS constellation geometry is constantly changing; therefore, the values of the ***H*** and ***S*** matrixes in equations 4.B.1.1 and 4.B.1-1 must be recalculated by channel level FDI at the beginning of each iteration of the algorithm.

As described by Sturza in Ref. 4, the GPS channel problem must be linearized before the FDI algorithms can be applied. The *M* x 1 channel level measurement vector, ***z***, contains the pseudorange residuals:

$$\boldsymbol{z} = \begin{bmatrix} \rho_1 - \left|\hat{\boldsymbol{r}} - \hat{\boldsymbol{s}}_1\right| \\ \vdots \\ \rho_m - \left|\hat{\boldsymbol{r}} - \hat{\boldsymbol{s}}_m\right| \end{bmatrix} \tag{18}$$

where $\hat{s}_k$ is defined as the GPS satellite position as estimated by the ephemeris information encoded in the signal. The position estimate $\hat{r}$ is provided by receiver level FDI. The 4 x 1 state vector, ***x***, consists of the Earth Centered Inertial (ECI) position errors and clock bias error:

$$\boldsymbol{x} = \begin{bmatrix} \hat{\boldsymbol{r}} - \boldsymbol{r} \\ b_{clock} - \hat{b}_{clock} \end{bmatrix} \tag{19}$$

The pseudorange measurement matrix, ***H***, is composed of the line-of-sight vectors from the receiver to the satellites with values of unity in the 4th column. In the channel level, there is a single application of the FDI algorithm which includes up to 12 pseudorange measurements. The channel level anomaly matrix contains 12 rows, each of which records the faults of one GPS satellite in the field of view. Whenever a new satellite enters the receiver's field of view, the anomaly history is cleared by replacing the associated anomaly row with zeros.

As discussed in the previous section, the receiver position and clock bias cannot be calculated in an environment with poor satellite geometry. If the receiver level was not able to calculate a position estimate or if there are only four GPS satellites the in the selected receiver field of view, the channel level FDI issues a status of no solution. If there are five satellites in the field of view, there is only one redundant measurement. In this instances, the FDI can detect, but not identify, a fault. With five valid satellites, the channel level FDI will verify the measurements to the navigation filter, provided no fault is detected. If six or more pseudorange measurements are available, the channel level FDI can detect and identify a fault. With six or more valid satellites the FDI algorithm will operate as described in Section II.B. Since the SIGI has a 12 channel receiver, eight (12 channel receivers minus 4 required pseudoranges) thresholds are set.

Channel level threshold tuning is particularly sensitive, because not all measurements are equally redundant. As discussed in Ref. 6, some GPS geometries are only partially observable in parity space. If this situation occurs, the decision variable D is systematically decreased causing the FDI to become less responsive to channel level faults. In order to maintain an acceptably low level of missed detections, the fault thresholds are tuned close to the measurement noise, resulting in more false detections. Since there are multiple redundant channels, a false detection in any single channel does not significantly affect the navigation filter.

During channel level FDI validation, each Monte-Carlo case has one injected error that is placed on a random valid channel (1-12) on the selected receiver, starting from a random time. The injected error continues to be added to the GPS signal for as long as the selected GPS satellite is tracked by the receiver. If the failed GPS satellite leaves the receiver field of view in less than 50 seconds, the Monte-Carlo run is considered invalid and the results are not included in the analysis. This ensures the channel level FDI is allowed sufficient time to detect and identify the faulty satellite.

IV. Safe Abort Maneuver Design

The Cygnus vehicle must be capable of safely performing an abort maneuver at any time during rendezvous. This requirement imposes design restrictions on both the design of the abort maneuver and on the design of the nominal trajectory.

The key driving requirements during the Rbar Ascent phase in which the glideslope algorithm is used are:

1. The GN&C subsystem shall perform abort maneuvers prior to Approach Initiation (AI) that place the vehicle on a trajectory that remains outside of the AE for a minimum of 24 hours.
2. The GN&C subsystem shall perform abort maneuvers that exit the bounds of the Approach Ellipsoid (AE) within 90 minutes after the abort maneuver execution.
3. The GN&C subsystem shall perform abort maneuvers outside the KOS that prevent the vehicle from entering the KOS.
4. The GN&C subsystem shall perform abort maneuvers that establish and maintain a positive opening rate after an abort maneuver initiated within the Keep Out Sphere (KOS) while inside the AE.
5. The GN&C subsystem shall perform abort maneuvers that exit the bounds of the Approach Ellipsoid (AE) and remain outside for at least 24 hours after the abort maneuver initiation.
6. The GN&C subsystem shall maintain a trajectory that remains at least 1.829 m away from any ISS element, other than the SSRMS, during free-flight operations.

A. Active and Passive Abort Overview

Two types of aborts are used throughout the Cygnus proximity operations trajectory: Abort Below and Abort Above (shown in Figure 9). The Abort Below is a fixed delta-v command opposing the velocity of the Cygnus which will cause the Cygnus to follow a trajectory that drifts below and ahead of the ISS. This abort type is used anytime an abort is commanded before the Cygnus has reached the initial approach phase, including the 4 km and 1.4 km coelliptics. The Abort above command is a fixed delta-v maneuver in the direction of the velocity with a sufficient radial component away from the ISS to guarantee an initially positive opening rate (see requirement 4). The Abort Above abort type will be used if an abort is required during final approach or during a retreat. The abort types were designed to minimize required delta-v while still satisfying the key driving requirements in the presence of substantial implementation error.

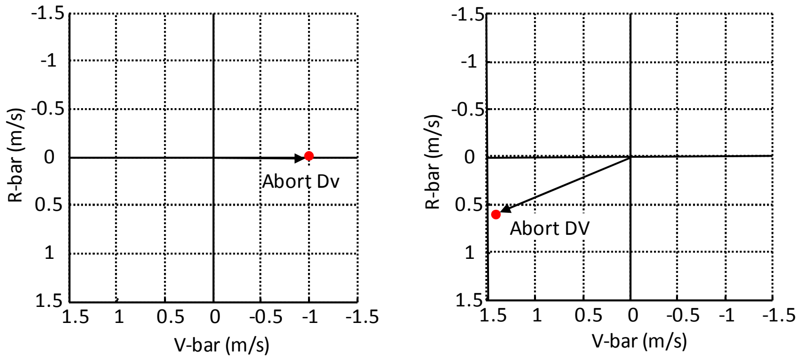


Figure 9 Abort Below Delta-v (left) and Abort Above Delta-v (right)

B. Safe Corridor Monitoring (SCM) for Abort

Aborts can be triggered by direct operator intervention or by the automatic onboard system. The automatic onboard system will trigger an abort in order to prevent the Cygnus from entering a region in which it is no longer safe to perform an abort. Thus, a safe corridor is defined in which an abort can be safely triggered. Safe corridor monitoring takes place onboard Cygnus to determine that the vehicle is still within the safe corridor. The nominal proximity operations rendezvous and ascent profile has been divided into phases and each phase has been assigned a set of thresholds within which an abort maneuver is guaranteed to meet the key driving requirements for abort maneuvers. These thresholds are defined as convex constraints on combinations of vehicle states. Testing if a vehicle lies within the safe corridor is performed by evaluating a matrix inequality representing the safe corridor constraint on the current state of the vehicle.

C. Choosing Abort Parameters

Designing safe corridors is accomplished by creating a wide margin around the anticipated trajectory dispersions. An example of a safe corridor is shown in Figure 10. In this figure, the blue lines show trajectories flown during a series of 1000 Monte Carlo Simulations. Safe corridors are chosen to constrain the following qualities:

1. Position Constraints
2. Velocity Constraints
3. Differential Semi-major Axis Constraints

Position and velocity constraints created independently would be insufficient to guarantee a safe corridor around anticipated dispersions. To couple position and velocity constraints, differential semi-major axis is also constrained. Because differential semi-major axis is directly related to free drift rate, this approach allows the dispersions to be encompassed with margin while still guaranteeing abort safety. After establishing safe corridors that surround the dispersions, those corridors are widened iteratively until they reach the limits of safety. This process is repeated independently for each constraint direction.

The safe corridor thresholds are tested by evaluating high fidelity propagation of abort maneuvers conducted from the threshold boundaries and verifying the resulting trajectories satisfied the abort requirements. In addition, these corridors are propagated as polytopes using the CW equations (see Figure 11) through the MPT software package[10] and those polytopes are tested for requirement violations.

D. Abort Trajectory Verification

A Linear Covariance (LinCov) tool was used to perform a preliminary design of the rendezvous, approach and departure. Subsequent detailed design of the trajectory was performed using both the LinCov tool and a high-fidelity 6-Degree-Of-Freedom (6-DOF) simulation tool. Final trajectory design verification was performed with the 6-DOF simulation via Monte-Carlo analysis.

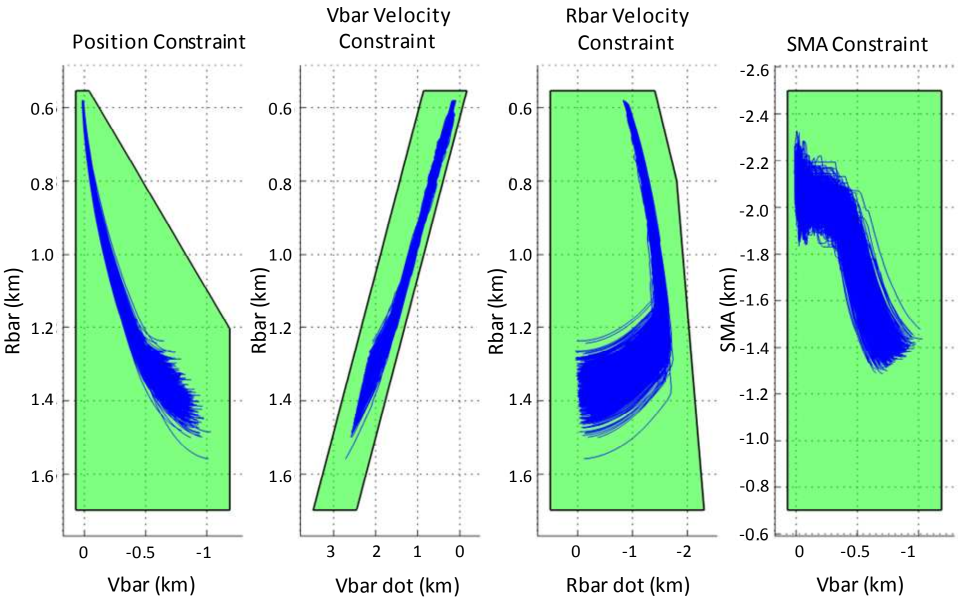


Figure 10 Safe Abort Threshold for Trajectory between ADV3 and ADV4

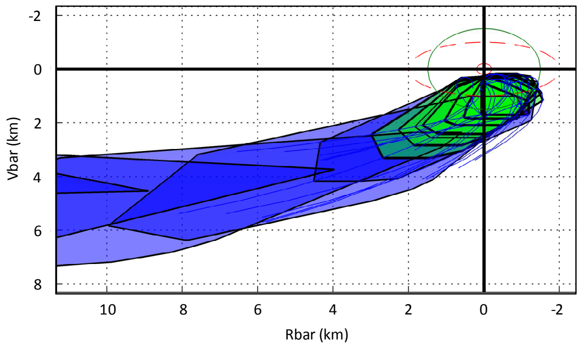


Figure 11 Propagation of ADV3-to-ADV4 Threshold using CW Equations

(Green represents 3 minute intervals, blue represents 10 minute intervals)

1. *Linear Covariance Analysis*

The Lincov tool was used to analyze the effect of navigation errors and control errors on the overall trajectory for the Cygnus vehicle. The trajectory design was evaluated in terms of the navigation dispersions and trajectory dispersions. The Lincov tool computes these dispersions using linear covariance analysis techniques and aims to reproduce the same statistical information that a non-linear closed-loop 6-DOF simulation would produce.

Navigation errors or dispersions are the differences between the actual state and the estimated state of the Cygnus vehicle and the LinCov tool characterizes how well the navigation system, with the sensors being utilized, is able to determine the current state. Trajectory errors or dispersions are the differences between the actual state and the nominal or desired state of the Cygnus vehicle. LinCov provides for a variety of guidance and targeting capabilities in order to perform rendezvous maneuvers and a sequence of maneuvers can be determined and then appropriately analyzed.

The trajectory dispersions are influenced by the proposed navigation system and the control system and vice versa. In other words a trajectory can be designed that will drive the requirements for the navigation system and control system. Alternatively, established navigation system and control system capabilities can drive the design of the trajectory. The LinCov tool provided a design and analysis environment where navigation system, control system and trajectory analysis trades could be performed quickly. The LinCov tool was used in this manner to support early system level trades for the selected Cygnus navigation sensors and thruster layout and sizing.

The LinCov tool was also used to design and analyze abort maneuvers with respect to the Key Driving Requirements (KDRs), which were previously summarized. At the conclusion of the preliminary design phase, LinCov provided preliminary verification of the trajectory design against an assumed performance of the relative navigation system and control system. This assumed performance for the relative navigation system and control system became the performance requirements that the relative navigation system and control system was then designed to meet. The subsequent detailed design activity evaluated the designed relative navigation system and control system for the nominal and contingency Cygnus vehicle trajectories using the high-fidelity 6-DOF simulation environment. The simulation results were then converted into three sigma dispersions and compared to LinCov.

The LinCov tool was used to perform the design of abort trajectories. As previously described, an abort strategy was developed that called for an ‘Abort Below’ prior to the Cygnus vehicle arriving at the 250 m Hold Point (HP) and receiving authorization to enter the KOS and an ‘Abort Above’ beyond this point.

There were two stages to overall abort maneuver verification. The first stage was the verification of the abort strategy at various points along the designed reference trajectory. The second stage was the verification of the abort thresholds from which an abort is triggered.

2. *Abort Strategy Verification*

A simple ‘canned’ abort maneuver for both an ‘Abort Below’ and an ‘Abort Above’ was selected for simplicity. The ‘Abort Below’ strategy imparts a retrograde acceleration to the Cygnus vehicle that causes the vehicle to drop below and drift in front of ISS. The ‘Abort Above’ strategy imparts a posigrade acceleration with a significant radial component to the Cygnus vehicle that causes the vehicle to initially move in front and above ISS and ultimately to drift behind ISS.

The ‘Abort Below’ trajectories at key points along the nominal trajectory leading up to the 250 m HP are shown in Figure 12, Figure 13, and Figure 14. These figures illustrate how several of the KDRs were verified. In these examples, the Cygnus vehicle must be prevented from entering the KOS, must leave the AE within 90 minutes, and must remain outside the AE for 24 hours. ‘Zoomed’ out plots not illustrated here illustrate the trajectory behavior over 24 hours and the verification of this requirement.

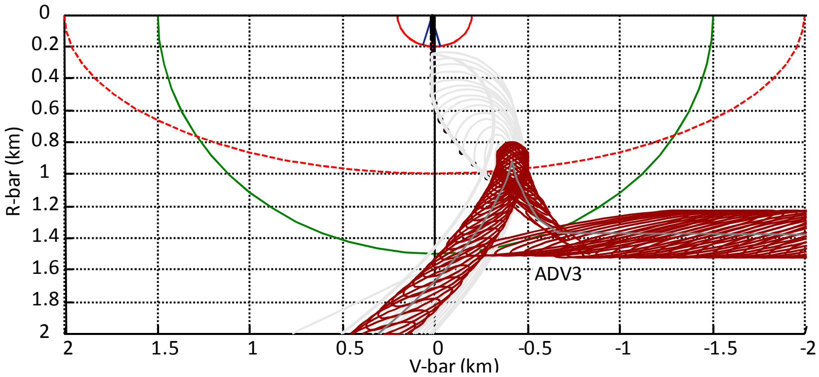


Figure 12 Abort Below Trajectory after ADV3 Maneuver Execution

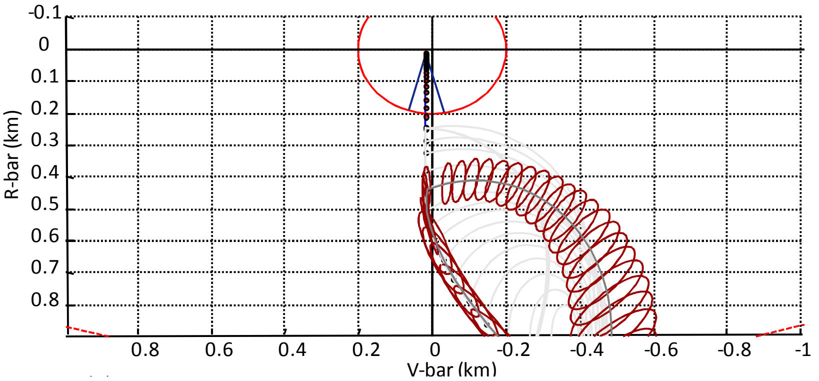


Figure 13 Abort Below Trajectory after ADV4 Maneuver Execution

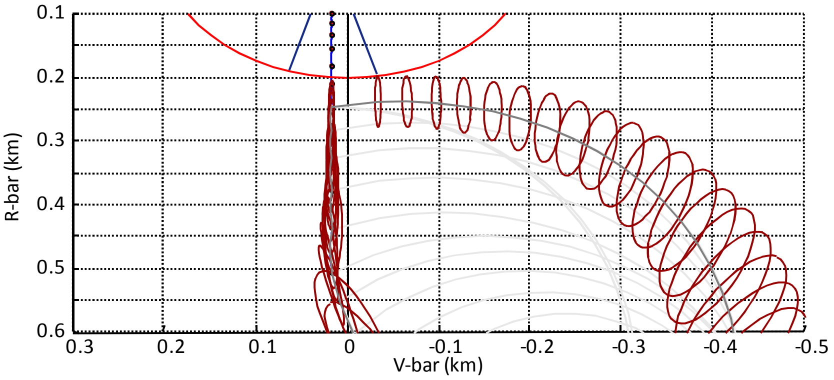


Figure 14 Abort Below Trajectory Prior to 250 m Station-Keeping

After analyzing the resultant free-drift trajectories and range-rate profiles associated with executing an 'Abort Above' at various locations along the trajectory, it was found that there were two abort scenarios that were especially critical. The first scenario is immediately following station-keeping at the 250 m HP and the second scenario is when the Cygnus vehicle is at or near the capture location in very close proximity to the ISS.

The 'Abort Above' trajectories for these two scenarios are shown in Figure 15 and Figure 16. Figure 17 illustrates a 'Zoomed' out plot to illustrate the trajectory behavior over the long term. Figure 18 illustrates the range-rate of the Cygnus vehicle as it executes the abort maneuver and exits the KOS and AE. These figures illustrate how several of the KDRs were verified. In these examples, the Cygnus vehicle must establish and maintain a positive opening rate, must leave the AE within 90 minutes, and must remain outside the AE for 24 hours

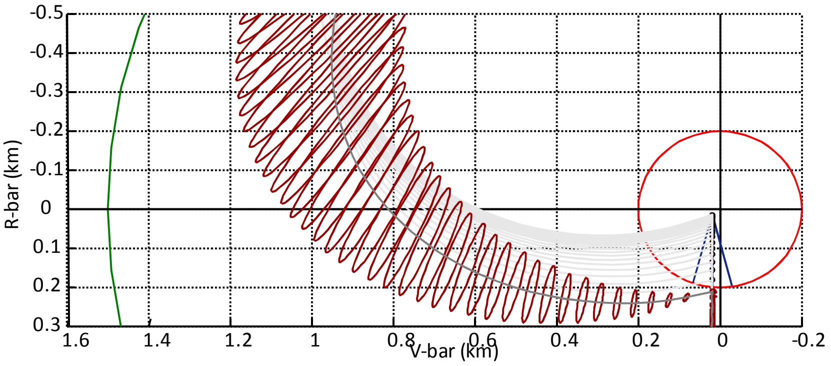


Figure 15 Abort Above Trajectory Immediately Following 250 m Station-Keeping

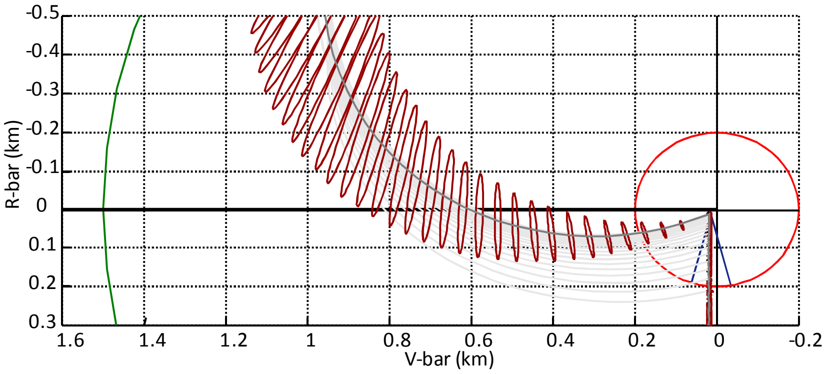


Figure 16 Abort Above Trajectory from the Capture Location

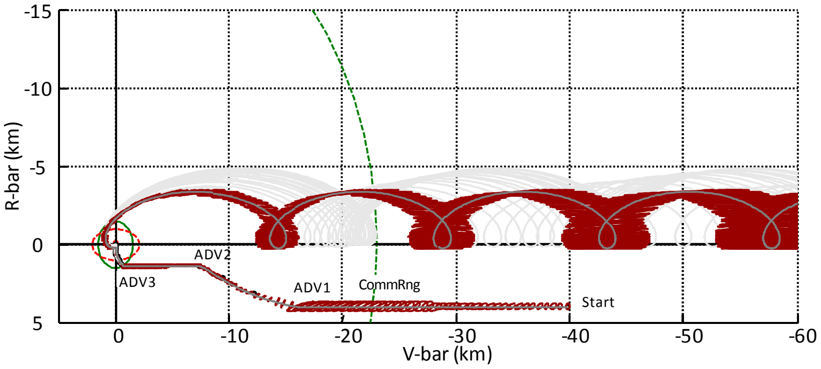


Figure 17 Abort Above Trajectory Immediately Following 250 m Station-Keeping (Zoom Out)

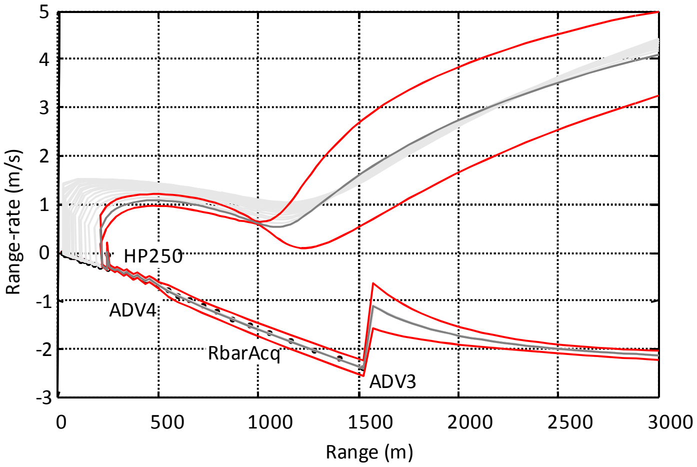


Figure 18 Abort Above Range-Rate Immediately Following 250 m Station-Keeping

E. Abort Trajectory Verification

Abort thresholds are intended to define the regions in which it is safe to perform abort maneuver and, as a result, where the Cygnus vehicle must remain at all times. If an abort maneuver is executed within appropriately defined abort regions, the resulting abort trajectory will meet the previously summarized KDRs. Abort thresholds are described as constraints on the Cygnus state relative to the ISS. These constraints represent restrictions on the relative position, the relative velocity, and the relative semi-major axis of the two vehicles. The abort thresholds are defined for specific regions of the rendezvous and approach profile.

Figure 10 illustrates abort thresholds for the trajectory region between ADV3 and ADV4. The green shapes that envelope the trajectory represents the abort thresholds on relative position, relative velocity and relative semi-major axis. The blue lines represent nominal trajectory dispersions for relative position, relative velocity and relative semi-major axis.

Verification of the abort threshold design was accomplished by initially sampling the initial conditions on the boundary of the threshold (in relative position, relative velocity, and relative semi-major axis), applying the abort maneuver according to the abort strategy, and then propagating the resulting trajectory in the presence of drag and other high-order effects. Clearly, this technique would require a significant amount of effort to verify all possible initial conditions and therefore a separate verification technique was developed.

A more exhaustive verification technique was applied by considering the set of valid states to be a fixed polytope defined in a frame relative to the ISS and then propagating the polytope after applying the abort maneuver according to the abort strategy. The polytope propagation captures the execution of an abort maneuver from all possible initial conditions. This method of polytope propagation is comparable to the approach used by ATV to validate safety in the presence of an abort maneuver with process noise.

Figure 19 shows the 24 hour propagation for a number of initial conditions sampled for the abort threshold boundaries for the trajectory region between ADV3 to ADV4. Figure 11 shows the short term propagation of the polytope region that represents the post-abort maneuver execution state at the abort threshold boundaries. The early portion of Figure 19 lies entirely within the polytope region establishing the validity of this approach.

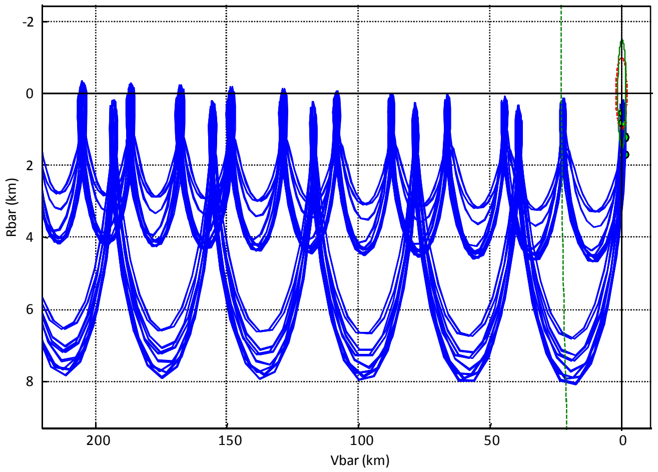


Figure 19 Day-long Propagation of Abort Maneuver Execution for ADV3 to ADV4 Threshold Boundary

V. Conclusions

The Cygnus vehicle employs several applications of fault management to meet the ISS visiting vehicle safety requirements for free-drift, safe abort, and minimum separation distances. During the approach and rendezvous mission phases, the IMU, LIDAR and GPS sensor measurements are independently screened by integrity monitoring routines. By comparing redundant measurements, these routines detect and isolate faulty sensor systems, thereby preventing corrupt measurements entering the navigation filter and ensuring sufficient navigation state accuracy to perform an abort. The abort monitoring corridors utilizes the fault tolerant navigation state to autonomously trigger an abort before safety requirements are violated. Together the sensor fault management routines and the abort monitoring corridors guarantee that the Cygnus is always capable of performing a safe abort. The abort corridors and the sensor fault management routines are verified using Monte-Carlo simulations with realistic disturbances.

The abort corridors performance is also validated using linear covariance analysis and a polytope propagation method.

Acknowledgments

The material presented in this paper is the result of the effort of a large team of people from the Orbital Sciences Corporation and the C. S. Draper Laboratory. We would like to thank David Woffinden and Stanley Shepperd from Draper Laboratory; Brian Keller from Orbital Sciences Corporation.

References

[1] SSP 50808 International Space Station (ISS) to Commercial Orbital Transportation Services (COTS) Interface Requirements Document (IRD).

[2] Clark, F.D., “Orion Final Approach Guidance Design,” FltDyn-CEV-08-084, CMT-08-022, GCD-08-484, 25 Apr 2008.

[3] Maybeck, P., *Stochastic Models, Estimation, and Control*, Volume 141 of Mathematics in Science and Engineering, Academic Press, 1979.

[4] Sturza, Mark, “Navigation System Integrity Monitoring Using Redundant Measurements,” *Journal of the Institute of Navigation*, Vol. 35, No. 4, 1988, pp. 69-87.

[5] Kaplan, K. D., Hegarty H.J., *Understanding GPS Principles and Applications*, 2nd ed., ArtTech, Boston, 2006, Chaps 7.

[6] Brown, R. G.,"A Baseline RAIM Scheme and a Note on the Equivalence of Three RAIM Methods." *Journal of The Institute of Navigation*, Vol. 39 No. 3, Winter 1992, pp. 301-316.

[7] Miotto, P., “Designing and Validating Proximity Operations Rendezvous and Approach Trajectories for the Cygnus Mission,” *AIAA Guidance, Navigation, and Control Conference* AIAA, Toronto, Ontario, Canada, 2010.

[8] Fritz, M., “A Comparative Study of Kalman Filter Implementations for Relative GPS Navigation,” MS Thesis, Aerospace Engineering Dept., Texas A&M Univ. College Station, TX, 2009.

[9] Bancroft, S., “An Algebraic Solution of the GPS Equations,” *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-21, NO. 7, 1985, pp. 56-59.

[10] M. Kvasnica and P. Grieder and M. Baoti, Multi-Parametric Toolbox (MPT), http://control.ee.ethz.ch/~mpt, 2004.