AIAA Guidance, Navigation and Control Conference and Exhibit
20 - 23 August 2007, Hilton Head, South Carolina

AIAA 2007-6789

Fault Tolerant Relative Navigation using Inertial and Relative Sensors*

Gabriel M. Hoffmann† Dimitry Gorinevsky‡ Robert W. Mah§
Claire J. Tomlin ¶ Jennifer D. Mitchell‖

Many emerging applications of space, ground, marine, and air vehicles require relative automated navigation with respect to other vehicles and objects. Disturbances in the environment may cause faults in relative navigation sensors. For sensors based on cameras or laser range finders, events as common as lighting changes, glint, or obstruction by debris could potentially cause spurious responses. Relative navigation is safety critical–fault tolerance must be addressed. We propose a fault detection, identification, and recovery architecture using multiple moving horizon estimators, each for a separate hypothesis of the fault state of the system. The hypothesis with maximum empirical likelihood is selected. Detected and identified faults are reported to the main navigation filter, which may then discard the relative navigation sensor data, and instead temporarily rely on the inertial navigation system. The guidance system may also act on the identified fault state, taking actions to recover the system to a safe state. This logic is demonstrated in simulation for the automated rendezvous and docking (AR&D) of spacecraft–a key technology for the near future demands of the space program. The simulation results demonstrate that faulty relative sensors may seriously affect the navigation solution. The proposed fault detection scheme has demonstrated an ability to identify faults in these sensors and take them offline before they disrupt navigation and lead to mission failure.

I. Introduction

Relative navigation sensors are used to measure the position, velocity, and attitude relative to other objects. Sensing is a critical capability for autonomous vehicles to interact with other vehicles and to account for obstacles. Relative measurements can be fused with absolute navigation data in a navigation filter to track the relative motion of each object being observed. Even if the other vehicle's position is known, absolute navigation is often insufficient in close proximity; accurate relative navigation sensing is required. Disturbances in the vehicle's environment may lead to faults of the relative navigation sensor. When this occurs, the fault must be detected so that more reliable absolute sensors can temporarily be relied upon entirely. This paper considers the problem of relative navigation for automated rendezvous and docking (AR&D) of spacecraft, but the problems with such sensors are general for ground, marine, air and space vehicles.

*This work was supported by NASA Exploration Systems Mission Directorate ETD Program, AR&D Sensors project, through Mitek Analytics research subaward #8020-011, USRA prime contract #NCC2-1426.

†Ph.D. Candidate, Department of Aeronautics and Astronautics, Stanford University. Mitek Analytics LLC. AIAA Member. gabeh@stanford.edu

‡Consulting Professor, Electrical Engineering Department, Stanford University. Principal, Mitek Analytics LLC. AIAA Member. gorin@stanford.edu

§Research Scientist; SSRL Group Lead, NASA Ames Research Center. Robert.W.Mah@nasa.gov

¶Professor, Department of Aeronautics and Astronautics; Director, Hybrid Systems Laboratory, Stanford University. Professor, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. AIAA Member. tomlin@stanford.edu

‖CEV Flight Dynamics Deputy Functional Area Manager, ETDP/AR&D Sensor Technology Project Manager, EG6/GN&C Autonomous Flight Systems Office, Aeroscience and Flight Mechanics Division, NASA Johnson Space Center. jennifer.d.mitchell@nasa.gov

Fault tolerant relative navigation is a vital design requirement of many systems. It is a key aspect of the airplanes which we trust daily to provide safe landings. Relative navigation with respect to the landing strip is at the core of the automated landing capability. The level of reliability and fault tolerance required for a Category III Autolanding System is necessarily extreme, to guarantee the safety of passengers and ground facilities.[1] A less developed automated function is for airplanes to perform closely spaced parallel approach to airports.[2] One of the limiting technologies is the ability to detect faults in the navigation systems to prevent collision in limited visibility conditions. As a final example, in autonomous automobiles, relative navigation is again found to be a critical technology. Modern automatic cruise control systems rely on radar measurements. Self driving automobiles recently competed in an offroad race, the DARPA Grand Challenge, that demonstrated the state of the art in autonomy, including the ability to drive around obstacles. However, even the vehicle that won the race, Stanley, encountered faulty perception data that led to driving blunders.[3]

Many current and near-future space system applications rely heavily on relative navigation. The near term goals of the NASA Exploration Program require robust relative navigation to support rendezvous, proximity operations and docking for both crewed and uncrewed vehicles. The near term missions require automated docking of the Orion Crew Exploration Vehicle (CEV) to the International Space Station. For lunar missions, in event of lost communication with ground, there could be a need to carry out the rendezvous, proximity operations and docking autonomously. The navigation performance critically affects the safety of the controlled spacecraft during such maneuvers. In the past, the ETS-VII program encountered problems due to interaction of the control system with sensors measurements.[4] Unexpected noise in the navigation sensors is largely responsible for the mishap in the DART AR&D demonstration.[5] In future space applications, such as lunar landing, relative navigation sensors will play a key role. Some technologies already developed for spaceflight include relative GPS navigation[6–8] and reflected laser positioning systems.[4]

In many relative navigation scenarios, the sensors may be categorized into relative navigation sensor packages and absolute navigation packages. The latter are based on Inertial Navigation Systems (INS). The INS are mature and reliable systems. Often, such systems are multiply redundant. On the other hand, relative sensing of the external environment is less mature and can be susceptible to external disturbances, such as lighting, glint, or obstruction caused by debris. These are inherently less reliable systems. The goal of this work is to formulate and demonstrate a fault tolerant architecture that relies on the INS to detect the faults of relative navigation sensors and to recover from these faults. The proposed analytical redundancy approach is somewhat different from voting and string selection techniques commonly employed in fault tolerant navigation systems, such as a fault tolerant redundant INS. The main advantage is that the fault tolerance could be achieved without having many redundant relative sensors.

The fault tolerant architecture proposed in this paper compares interpretations of sensor measurements based on multiple probabilistic fault-state hypotheses. With the most probable fault state of the system identified, the navigation filter may react accordingly, preventing navigation errors. The guidance system may invoke an abort logic to avoid unsafe system states. We formulate hypotheses estimators in a moving horizon setting. The proposed algorithm has low computational demands, with the most intensive requirement being the numerical inversion of sparse, banded matrices. The algorithm is demonstrated in the simulation of an automatic rendezvous and docking operation between two spacecraft in low Earth orbit.

The contributions of this paper are three-fold. First, a moving horizon estimation (MHE) technique is developed using multiple models of the fault state of the system to detect and identify the likelihood of faulty measurements. Second, the technique is applied to fault detection in the relative navigation problem. Specific models are developed assuming reliable absolute (inertial) navigation, and a relative navigation sensor that might fail. Third, the fault tolerant architecture is applied to relative navigation for spacecraft AR&D, and demonstrated to be effective in simulation.

There is significant literature on fault detection, identification and recovery (FDIR), with many examples of successful work. Dynamic Bayesian networks were used as a fault diagnosis technique,[9] which is well suited for discrete probability maps. Techniques to identify faulty behavior modes, without the availability of external sensor, were studied by Kleer and Williams.[10] Another strong body of work has been demonstrated using neural network methods.[11,12] These methods could be susceptible to failures when they encounter "new experiences", for which they were not trained, and may be difficult to validate formally.

Closer to our approach, is work of Hsiao and Tomizuka, who use a switching Extended Kalman Filter

(EKF) that is capable of tracking complex switching histories between system modes using a branching method.[13] In our algorithm, we use a window of recent measurements to compute the likelihood of fault states within that window that may have led to the measurement sequence, whereas branching methods use an EKF to compute the likelihood of the end points only. The branching EKF method does not account for the changing probability of the branches that lead to the end points, as more measurements become available. Allerton and Jia surveyed fault-tolerant architectures for aircraft navigation systems using banks of Kalman filters with multiple fault hypotheses and a moving window method for fault inference.[14] However, the architectures referenced in[14] do not use the moving window for estimation, so information contained in the past Kalman filter residuals is not improved by more recent measurements.

By taking into account the entire set of the data inside the horizon at once, the assessment of fault state likelihood can be improved over existing EKF methods. By using only a finite horizon of data, the computation remains tractable, and the system is able to identify when the system recovers from a fault state. The use of moving horizon estimation for fault detection allows trends in data, that take multiple time steps to appear, to be captured. The results demonstrate that an MHE excels over a simple EKF in estimation for relative navigation, and in doing so, with multiple hypotheses for fault state, provides a much more accurate prediction of a fault condition.

The application of moving horizon estimation (MHE) to interpret sensor data is rapidly developing, thanks to recent advances in computational capabilities, although its application to relative navigation appears to be new. MHE has demonstrated superior performance to extended Kalman filters (EKFs) in nonlinear estimation problems.[15,16] Moving horizon estimation has been used for fault detection in hybrid system using a mixed integer quadratic program (MIQP) formulation.[17] Although this work is applicable to fault detection, we address the problem in a different way, by making simplifying assumptions that allow the algorithm to be run online with limited computational resources. In other work, MHEs have been used in a similar approach to ours, but without an explicit use of a fault model.[18]

We proceed by formally stating the problem, and then giving details of the proposed fault-tolerant approach. We apply this approach to automated rendezvous and docking in a simulation with 6 degrees of freedom for both the chaser spacecraft and the target spacecraft. The results from experiments with the simulation demonstrate the capability of the algorithm to correctly identify faults, the improvements to the navigation filter when it uses the fault identification information, and the potential for catastrophe that occurs when there is no fault identification system.

II. Problem Formulation

This section presents the models used for the estimation and fault detection algorithms in this work. We start by outlining the motivating problem, relative navigation. Then we define the motion and sensing models used in this work. Next, the Extended Kalman Filter (EKF) equations are presented for completeness; we use the EKF in this work. The linearized equations used in the EKF are also used in formulating the moving horizon estimator.

A. Relative Navigation Problem

For this work, the relative navigation problem is defined to be the task of estimating the relative state of a chaser vehicle with respect to a target vehicle. A motivating application is spacecraft rendezvous, as depicted in Figure 1. Here, GPS is used to maneuver the vehicles to within far field range, but near field approach requires finer precision than can be provided by current GPS systems. The typically considered near field approach sensor suite consists of an inertial navigation system and a relative navigation system. For the spacecraft rendezvous problem, the goal is to measure the relative position and attitude so that the large, heavy spacecraft can safely approach one another and dock. To accomplish this, upon terminal approach, sub-centimeter accuracy is often required.

To perform relative navigation, the inertial navigation system uses components that operate in a controlled environment, using mature, field tested technology. In space applications, inertial navigation systems often additionally have layers of redundancy. However, relative navigation sensors typically make measure-

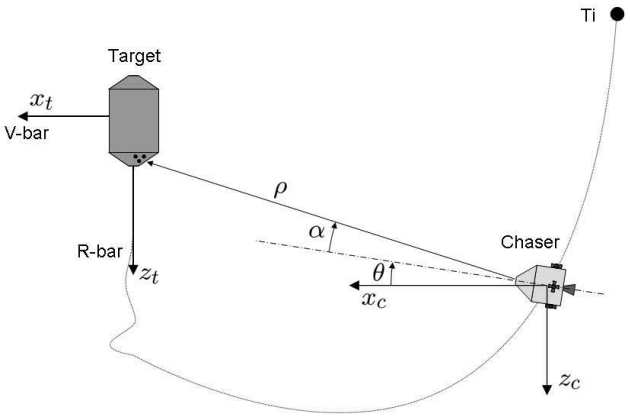

Figure 1. The relative navigation problem. An R-bar approach scenario is depicted, as used in Section IV for the simulated guidance system. The chaser spacecraft has inertial sensors and relative navigation sensors. The relative navigation sensor returns a measurement direction to the target spacecraft, with respect to the chaser, and the range, when the spacecraft are close. Although the inertial sensors are generally reliable, the challenges of relative navigation sensing make faults in relative navigation measurements possible. The depicted procedure is one procedure currently used, manually, for Space Shuttle Orbiter rendezvous with the International Space Station (ISS).

ments through an unknown environment, with disturbance such as glint, obstruction caused by debris, and ambient electrical noise interfering with optical or electrical measurements of the relative state. Due to the complexity of the measurements being made, large deviations in performance are both feasible and difficult to detect using the relative navigation sensor alone. Thus, an important component of the relative navigation problem is the ability for the system to be fault tolerant with respect to the output of the relative navigation sensor. By combining the sensor outputs, insight can be gained to the fault state of the system. When a fault occurs, if it is not caught before corrupted measurements are used in the navigation filter, the relative state estimate may incur large, unknown errors.

B. System Model

The state of the system is $\mathbf{x} \in \mathbb{R}^n$, where $\mathbf{x}^T = [\mathbf{x}_c^T \mathbf{x}_t^T]$, the states of the "chaser vehicle" and the "target vehicle", respectively (inspired by the automated rendezvous and docking example).[a] The discrete time equations of motion of the system at integer time t are given by

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \nu(t) \tag{1}$$

where the control inputs are $\mathbf{u} \in \mathbb{R}^r$, and $\mathbf{f}$ is a potentially nonlinear equation of motion, and $\nu(t) \in \mathbb{R}^n$ is a random disturbance defined later. Note that $\mathbf{u}$ can be determined from the control system's command history, or measured through the sensor suite. If the commands, and the measurements of their effects, are available, the commands can be used for the equations of motion, and the measurements can be treated as sensor outputs.

The sensor model is a stochastic model of the values reported by the sensors, as a function of the system states. Sensors include those that make direct measurements of the system state, such as GPS, and relative sensors, such as a relative optical navigation system. The complete vector of such measurements is given by the measurement model,

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) + \xi(t) \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^m$ (m is the number of measurements taken at each time step). Using the historical output data of the sensor, this model can be established for nominal, fault-free sensor performance.

[a] Bold face fonts are used for vector variables and functions which return vectors.

We consider a system with two types of sensors. One uses primarily inertial data, and is presumed to be reliable. The other uses relative measurements, and is potentially faulty. Note that GPS could fall in either category of sensor, depending on the level of redundancy and the operating environment. Measurements from the reliable system can be grouped into $\mathbf{y}_{base}$, and those from the potentially faulty systems are in $\mathbf{y}_{rel}$, such that

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{y}_{base}(t) \\ \mathbf{y}_{rel}(t) \end{bmatrix} \tag{3}$$

where $\mathbf{y}_{base} \in \mathbb{R}^{m_{base}}$ and $\mathbf{y}_{rel} \in \mathbb{R}^{m_{rel}}$. The number of base and relative measurements are m_{base} and m_{rel} respectively. Then, the measurement model, (2), can be separated into two components,

$$\mathbf{y}_{base}(t) = \mathbf{h}_{base}(\mathbf{x}(t)) + \xi_{base}(t) \tag{4}$$
$$\mathbf{y}_{rel}(t) = \mathbf{h}_{rel}(\mathbf{x}(t)) + \xi_{rel}(t) \tag{5}$$

For the relative navigation problem, the sensor measurements are nonlinear functions of the states being estimated, such as relative position and velocity. Additionally, the equations of motion are often nonlinear. This necessitates linearization of the equations of motion and the measurement model, as in an EKF. The linearized measurement model is

$$\mathbf{x}(t+1) \approx \mathbf{f}(\mathbf{x}_p(t), \mathbf{u}(t)) + A(\mathbf{x}(t) - \mathbf{x}_p(t)) + \nu(t) \tag{6}$$

where $\mathbf{x}_p(t) \in \mathbb{R}^n$ is the operating point about which the linearization is performed. As commonly done in navigation algorithm derivation, we assumed that, $\nu(t) \in \mathbb{R}^n$ is zero mean Gaussian noise with a covariance $Q \in \mathbb{R}^{n \times n}$, and $A \in \mathbb{R}^{n \times n}$ is given by the Jacobian of the dynamics.

$$A\left(\mathbf{x}_p(t)\right) = \left.\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_p(t)} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}_p(t))}{\partial x_1} & \cdots & \frac{\partial f(\mathbf{x}_p(t))}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x}_p(t))}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x}_p(t))}{\partial x_n} \end{bmatrix} \tag{7}$$

The linearized measurement model is

$$\mathbf{y}_{base}(\mathbf{x}(t)) \approx \mathbf{h}_{base}(\mathbf{x}_p(t)) + H_{base}(\mathbf{x}(t) - \mathbf{x}_p(t)) + \xi_{base}(t) \tag{8}$$
$$\mathbf{y}_{rel}(\mathbf{x}(t)) \approx \mathbf{h}_{rel}(\mathbf{x}_p(t)) + H_{rel}(\mathbf{x}(t) - \mathbf{x}_p(t)) + \xi_{rel}(t) \tag{9}$$

where $\xi_{base}(t) \in \mathbb{R}^{m_{base}}$ and $\xi_{rel}(t) \in \mathbb{R}^{m_{rel}}$ are additive Gaussian noise with zero mean and covariance $\Xi_{base} \in \mathbb{R}^{m_{base} \times m_{base}}$, and $H_{rel} \in \mathbb{R}^{m \times m_{rel}}$ are the Jacobian matrices of the measurement model,

$$H_{base}\left(\mathbf{x}_p(t)\right) = \left.\frac{\partial \mathbf{h}_{base}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_p(t)} = \begin{bmatrix} \frac{\partial h_{base,1}(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial h_{base,1}(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{base,m_{base}}(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial h_{base,m_{base}}(\mathbf{x})}{\partial x_n} \end{bmatrix} \tag{10}$$

$$H_{rel}\left(\mathbf{x}_p(t)\right) = \left.\frac{\partial \mathbf{h}_{rel}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_p(t)} = \begin{bmatrix} \frac{\partial h_{rel,1}(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial h_{rel,1}(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{rel,m_{rel}}(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial h_{rel,m_{rel}}(\mathbf{x})}{\partial x_n} \end{bmatrix} \tag{11}$$

The measurement model provides an expected value of the noise corrupted measurements, $\mathbf{z}(t) \in \mathbb{R}^m$.

C. Information for the Navigation Filter

To fuse the measurements, we assume that the vehicle is operating a navigation filter which is an Extended Kalman Filter (EKF). This is a well established approach to navigation system design, and broadly used in practice. The EKF is able to update the state estimate, $\mathbf{x}(t)$, using even partial measurements if portions

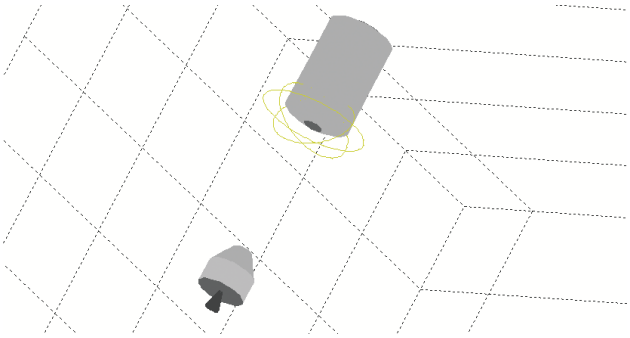

Figure 2. The output of the navigation filter for the application in Section IV. The depicted R-bar approach, along the radial line from the Earth to the target spacecraft, shows the true spacecraft positions at 750 meter range (spacecraft and error ellipses are magnified by 500x for visualization). The error ellipses show the estimated position and 1-sigma confidence volume for the target spacecraft as computed by the chaser spacecraft using relative navigation and inertial sensors. Note that if faulty data is not identified, the confidence volume will be inaccurately, dangerously small.

of sensor data are missing, and using measurements that provide incomplete state information. The EKF is optimal in a least squares sense for the estimate of the current state of the system, assuming the measurements are Gaussian, and the system is well approximated as locally linear. For a more detailed discussion on EKFs, please see.[19] Briefly, the key equations for updating the estimated mean, $\hat{\mathbf{x}}(t)$, and covariance, $\Sigma(t)$, are

$$\overline{\mathbf{x}}(t+1) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t)) \tag{12}$$

$$\overline{\mathbf{\Sigma}}(t+1) = A(\hat{\mathbf{x}}(t))\Sigma(t)A(\hat{\mathbf{x}}(t))^T + Q \tag{13}$$

$$K(t) = \overline{\mathbf{\Sigma}}(t+1)H(\overline{\mathbf{x}}(t+1))^T \left(H(\overline{\mathbf{x}}(t+1))\overline{\mathbf{\Sigma}}(t+1)H(\overline{\mathbf{x}}(t+1))^T + \Xi\right)^{-1} \tag{14}$$

$$\hat{\mathbf{x}}(t+1) = \overline{\mathbf{x}}(t) + K(t)\left(\mathbf{z}(t) - \mathbf{h}(\overline{\mathbf{x}}(t+1))\right) \tag{15}$$

$$\Sigma(t+1) = (I - K(t)H(\hat{\mathbf{x}}(t)))\overline{\Sigma}(t) \tag{16}$$

Note that (12) and (13) can be computed asynchronously with respect to (14), (15), and (16), as the relevant sensor data is available. Further, note that (14), (15), and (16) can be evaluated with only a subset of the sensor data, enabling data to be sampled at multiple rates. This also enables measurements to be entirely discarded when the sensor providing them is deemed faulty by the moving horizon fault detection logic. The output of a simulated EKF, from Section IV, is shown in Figure 2.

III. Fault Tolerant Relative Navigation Approach

This section focuses on the problem of finding whether a relative navigation sensor is healthy or faulty. As mentioned above, the relative navigation sensors are prone to common model failures, and physical diversity of such sensors would be likely insufficient for detecting the fault. Essentially, the approach discussed in this section is to use dead reckoning navigation based on inertial data as a reference for detecting relative navigation sensor faults. This section develops a fault tolerant relative navigation algorithm based on this idea. First, the fault modeling framework for relative navigation sensors is presented, and then the fault tolerant relative navigation algorithm is presented. The subsequent section demonstrates the algorithm in a detailed simulation of automated rendezvous and docking (AR&D).

The proposed approach to sensor fault detection is through a statistical evaluation of a healthy/faulty sensor hypothesis. Using an optimal estimation approach, we compute a likelihood ratio for the two hypotheses. The proposed optimal fault detection logic corresponds to multiple hypothesis testing approach commonly used in target tracking applications where a bank of extended Kalman filters is used for simultaneous evaluation of the hypothesis.[14] However, in this approach, a moving horizon is used, similar in principal to an iterated extended Kalman smoother[20] to provide the likelihood of historical data given past measurements, and those up to the present. This approach could be used as a baseline in trade studies even

if a simpler fault tolerant relative navigation logic would be deployed.

A. Fault Modeling

The derivation and analysis of the proposed fault tolerant relative navigation algorithm requires modeling of the faults of the relative navigation sensors. Ideally, the fault models should be simple, generic for different types of relative sensors, and adequately cover the faulty sensor behaviors encountered in practice.

We assume that the available modeling information about the faults includes the following

- Probability (frequency) of fault occurrence
- A statistical model of sensor noise in the normal (fault free) operation. If a normal distribution for the noise is assumed, the model is encoded by a covariance matrix and an expected value (mean), as in standard navigation filter design.
- Some statistical information about the sensor error magnitude in fail-active faults. Such information could be encoded in an empirically assumed distribution of the error.
- Fault persistency models. In the literature, there are six standard models of sensor faults including: (i) transient impulse, (ii) random fault, (iii) constant offset, (iv) stuck at zero, (v) stuck at the last value, and (vi) constant drift.[21] These are discussed in more detail below.

The first three types of fault models–transient impulse, random fault, and constant offset–are highly plausible for relative navigation sensors. These faults could be caused by an image of the target (or reflectors on the target) being deficient because of the lighting or extra reflections. The "transient impulse" corresponds, for instance, to spurious reflections distorting a single image frame. The "random fault" corresponds to the distortion changing randomly over an interval of time. The "constant offset" corresponds to occlusion or a spurious reflection being present, causing an offset error in the target pose measurement over time interval.

The "stuck at zero" and "stuck at the last value" faults are not specific to relative navigation sensors. These faults could possibly occur as a result of the processor hardware or software failure within the sensor, where the last value or a zero value is kept in the sensor output register. These faults are very unlikely since relative navigation sensors are expected to have sufficiently sophisticated electronics, with networked connections and heartbeat functionality. Finally, the "constant drift" fault is extremely unlikely since relative navigation sensors measure the position directly and possible faults are most likely to remain bounded. The constant drift with a small gain could potentially appear as result of the offset fault superimposed on the relative vehicle motion. As the chaser approaches the target, the same root cause of the offset could lead to changing offset value.

For a failed relative navigation sensor, we assume that equations (6) and (8) hold, whereas (9) is no longer true. Instead, we assume that

$$\mathbf{y}_{rel}(\mathbf{x}(t)) \approx \mathbf{h}_{rel}(\mathbf{x}_p(t)) + H_{rel}(\mathbf{x}(t) - \mathbf{x}_p(t)) + \xi_{fail}(t) \tag{17}$$

where $\xi_{fail}(t) \in \mathbb{R}^{m_{rel}}$ is the Gaussian noise model for the failed sensor, with zero mean and covariance $\Xi_{fail} \in \mathbb{R}^{m_{rel} \times m_{rel}}$. If the sensor output is unpredictable during a fault state, then a large value for Ξ_{fail} is used. Note that this demarcation of reliable and faulty sensors can be rearranged to test for faults in multiple relative sensor systems or subsets of their measurements.

B. Fault State Hypotheses

The first observation equation, (4), describes the inertial measurements. We will assume that the inertial measurement sensor, is reliable–it never fails, due to consistency of its operational environment, and redundancy. The inertial measurement has to be integrated to estimate the position. The second observation equation, (5), describes the relative navigation measurement. We assume that this sensor can fail. We need to detect from the measurement data if this is the case.

We need to discriminate between the following two statistical hypotheses

$$\mathbf{H_0}: \quad \{\text{No sensor fault, (6), (8), and (9) hold}\} \tag{18}$$

$$\mathbf{H_1}: \quad \{\text{Relative navigation sensor faulty, (6), (8), and (17) hold}\} \tag{19}$$

The decision is based on the models (6), (4), and (5), and on the measurement data available over a fixed time interval,

$$\mathbf{y}_{1:T} = \{\mathbf{y}(1), \ldots, \mathbf{y}(T)\} \tag{20}$$

The system state data on the same interval is

$$\mathbf{x}_{1:T} = \{\mathbf{x}(1), \ldots, \mathbf{x}(T)\} \tag{21}$$

where the state, $\mathbf{x}$, has components that are not necessarily directly observed.

The hypotheses (18) and (19) can be resolved by comparing the conditional probabilities for the system states, (21), conditional to the observations, (20),

$$p(\mathbf{X}_0|\mathbf{y}_{1:T}) \quad \lhd\rhd \quad p(\mathbf{X}_1|\mathbf{y}_{1:T}) \tag{22}$$

where $\mathbf{X}_0$ and $\mathbf{X}_1$ are the system state vectors estimated assuming the hypotheses $\mathbf{H_0}$ and $\mathbf{H_1}$ respectively. In what follows, we will estimate the ratio of the probabilities in (22) by using a standard Bayesian Maximum *a posteriori* Probability (MAP) formulation. This is a standard formulation.[19] Yet, an application to fault detection in aerospace engineering problem is not standard and we include the derivation for completeness.

Applying Bayes' rule for either of the conditional probabilities in (22) yields

$$p(\mathbf{X}|\mathbf{Y}) = p(\mathbf{Y}|\mathbf{X}) \cdot p(\mathbf{X}) \cdot c \tag{23}$$

where $p(\mathbf{Y}|\mathbf{X})$ is the observation model, $p(\mathbf{X})$ is the prior, and the last multiplier is the normalization constant, $c = (p(\mathbf{Y}))^{-1}$, that is the same for both hypotheses and, thus, of no consequence when computing the probability ratio.

Instead of (23), we will be dealing with the negative log-likelihood index, which has a minimum at the same value of $\mathbf{X}$ for which $p(\mathbf{Y}|\mathbf{X})$ is maximum.

$$L = -\log p(\mathbf{Y}|\mathbf{X}) - \log p(\mathbf{X}) \tag{24}$$

1. *Null Hypothesis*

For the Null Hypothesis, (18), the nominal system model (6), (4), and (5) can be used to compute the log-likelihood index, (24), as follows. First, consider the observation model, $p(\mathbf{Y}|\mathbf{X})$. Since the observation noises $\xi(t)$ are normally distributed, and independent between themselves and for different times t, we get, from the linearized models, (8) and (17),

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}) &= \textstyle\prod_{t=\tau}^{T} \left(P(\mathbf{y}_{base}(t)|\mathbf{x}(t))P(\mathbf{y}_{rel}(t)|\mathbf{x}(t))\right) \\ &= \eta_{(y|x),0} \textstyle\prod_{t=\tau}^{T} \Big(e^{-\frac{1}{2}\|[\mathbf{y}_{base}(t)-\mathbf{h}_{base}(\mathbf{x}_p(t))+H_{base}\cdot\mathbf{x}_p(t)]-H_{base}(\mathbf{x}_p(t))\cdot\mathbf{x}(t)\|^2_{\Xi_{base}}} \\ &\qquad \cdot e^{-\frac{1}{2}\|[\mathbf{y}_{rel}(t)-\mathbf{h}_{rel}(\mathbf{x}_p(t))+H_{rel}\cdot\mathbf{x}_p(t)]-H_{rel}(\mathbf{x}_p(t))\cdot\mathbf{x}(t)\|^2_{\Xi_{rel}}}\Big) \end{aligned} \tag{25}$$

where the normalization constant for the multivariate Gaussian is

$$\eta_{(y|x),0} = \frac{1}{(2\pi)^{\frac{m_{base}+m_{rel}}{2}} |\Xi_{base}|^{\frac{1}{2}} |\Xi_{rel}|^{\frac{1}{2}}} \tag{26}$$

Second, consider the prior model $p(\mathbf{X})$. Sequentially applying the model (6), and using the independence (whiteness) of the driving noise sequence, $e(t)$, yields

$$p(\mathbf{X}) = p_0 P(\mathbf{x}(1)) \prod_{t=1}^{T-1} P(\mathbf{x}(t+1)|\mathbf{x}(t)) = \eta_x p_0 \cdot e^{-\frac{1}{2}\|\mathbf{x}(1)-\mathbf{x}_0\|^2_{Q_0}} \cdot \prod_{t=1}^{T-1} e^{-\frac{1}{2}\|\mathbf{x}(t+1)-A\mathbf{x}(t)-[\mathbf{f}(\mathbf{x}_p(t),u(t))-A\mathbf{x}_p(t)]\|^2_{Q_t}} \tag{27}$$

with the normalization constant,

$$\eta_x = \frac{1}{(2\pi)^{T/2}|Q_0|^{\frac{1}{2}}(T-1)|Q_t|^{\frac{1}{2}}} \tag{28}$$

The initial state, $\mathbf{x}(1)$, is normally distributed as $\mathcal{N}(\mathbf{x}_0, Q_0)$, the state transition noise is distributed as $\mathcal{N}(0, Q_t)$, and p_0 denotes an *a priori* probability of the Null Hypothesis.

By substituting (25) and (27) into (24), we can compute the negative log-likelihood for the Null Hypothesis $\mathbf{H_0}$, (18),

$$\begin{aligned} L_0(\mathbf{X}) = \quad & \textstyle\sum_{t=1}^{T-1} -\frac{1}{2}\left\|\mathbf{x}(t+1) - A\mathbf{x}(t) - [\mathbf{f}(\mathbf{x}_p(t), u(t)) - A\mathbf{x}_p(t)]\right\|^2_{Q_t} \\ & + \textstyle\sum_{t=1}^{T} \frac{1}{2}\left\|[\mathbf{y}_{base}(t) - \mathbf{h}_{base}(\mathbf{x}_p(t)) + H_{base}\mathbf{x}_p(t)] - H_{base}(\mathbf{x}_p(t))\mathbf{x}(t)\right\|^2_{\Xi_{base}} \\ & + \textstyle\sum_{t=1}^{T} \frac{1}{2}\left\|[\mathbf{y}_{rel}(t) - \mathbf{h}_{rel}(\mathbf{x}_p(t)) + H_{rel}\mathbf{x}_p(t)] - H_{rel}(\mathbf{x}_p(t))\mathbf{x}(t)\right\|^2_{\Xi_{rel}} \\ & + \tfrac{1}{2}\left\|\mathbf{x}(1) - \mathbf{x}_0\right\|^2_{Q_0} - \log p_0 - \log \eta_{(y|x),0} - \log \eta_x \end{aligned} \tag{29}$$

Then, we can formulate the computation of the negative log-likelihood of the MAP estimate for $\mathbf{X}$ as an optimization problem

$$\mathbf{X}_{0,ls} = \arg\min_{\mathbf{X}} \left(L_0(\mathbf{X})\right) \tag{30}$$

The optimization problem leads to a linear system of equations that can be solved very efficiently for a large size of the data horizon T using standard sparse matrix packages. The last estimated point of the solution $\mathbf{x}(T)$ is the same as obtained by running an extended Kalman filter update, (12)-(16), for the estimation problem over the time horizon $t \in [1, T]$. Note that the sequence of states, (21), obtained as result of solving the batch problem, (30), will differ from a sequence of states obtained by running an extended Kalman filter update. This is because (30) is a *smoothing* problem, where the estimates of $\mathbf{x}(t)$ for $t < T$ are computed using the full data set, (20). The Kalman filter update is causal and uses the past data only at each step. Once the solution, (21), is found, the negative log-likelihood of the MAP estimate, $L_{0,ls}$ can be computed by substituting the solution, $\mathbf{X}_{0,ls}$ back into (29). This loss index, $L_{0,ls}$, is a measure of how improbable the observations are, assuming the Null Hypothesis.

2. *Fault Hypothesis*

Consider now computing the log-likelihood index, (24), for Fault Hypothesis $\mathbf{H_1}$, (19). The faulty system model is given by (6) and (4). The observation model, $p(\mathbf{Y}|\mathbf{X})$, can be computed using these models similar to how (25) is computed.

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}) \quad & = \textstyle\prod_{t=\tau}^{T} \left(P(\mathbf{y}_{base}(t)|\mathbf{x}(t))P(\mathbf{y}_{rel}(t)|\mathbf{x}(t))\right) \\ & = \eta(y|x), 1 \textstyle\prod_{t=\tau}^{T} \Big(e^{-\frac{1}{2}\|[\mathbf{y}_{base}(t) - \mathbf{h}_{base}(\mathbf{x}_p(t)) + H_{base}\cdot\mathbf{x}_p(t)] - H_{base}(\mathbf{x}_p(t))\cdot\mathbf{x}(t)\|^2_{\Xi_{base}}} \\ & \qquad \cdot e^{-\frac{1}{2}\|[\mathbf{y}_{rel}(t) - \mathbf{h}_{rel}(\mathbf{x}_p(t)) + H_{rel}\cdot\mathbf{x}_p(t)] - H_{rel}(\mathbf{x}_p(t))\cdot\mathbf{x}(t)\|^2_{\Xi_{fail}}} \Big) \end{aligned} \tag{31}$$

where the normalization constant is,

$$\eta_{(y|x),1} = \frac{1}{(2\pi)^{\frac{m_{base}+m_{rel}}{2}}|\Xi_{base}|^{\frac{1}{2}}|\Xi_{fail}|^{\frac{1}{2}}} \tag{32}$$

Note that the ratio $\frac{\eta_{(y|x),0}}{\eta_{(y|x),1}} = \sqrt{\frac{|\Xi_{fail}|}{|\Xi_{nom}|}}$, so the magnitude of this ratio increases with the magnitude of the noise introduced by the fault. Thus, the larger the assumed magnitude of noise during the fault state, the more the weight that is given to the null hypothesis, a tradeoff that probabilistically accounts for the expected size of noise distributions during nominal and fault states.

The prior model, $p(\mathbf{X})$ is the same for either hypothesis, and has the general form similar to (27). For Fault Hypothesis $\mathbf{H_1}$, the prior model is

$$p(\mathbf{X}) = p_1 P(\mathbf{x}(1)) \prod_{t=1}^{T-1} P(\mathbf{x}(t+1)|\mathbf{x}(t)) = \eta_x p_1 \cdot e^{-\frac{1}{2}\|\mathbf{x}(1) - \mathbf{x}_0\|^2_{Q_0}} \cdot \prod_{t=1}^{T-1} e^{-\frac{1}{2}\|\mathbf{x}(t+1) - A\mathbf{x}(t) - [\mathbf{f}(\mathbf{x}_p(t), u(t)) - A\mathbf{x}_p(t)]\|^2_{Q_t}} \tag{33}$$

where $p_1 = 1 - p_0$ denotes a prior probability of the Fault Hypothesis. By substituting (33) and (31) into (24) we can formulate computation of the negative log-likelihood index for the Fault Hypothesis $\mathbf{H_1}$, (19),

$$\begin{aligned} L_1(\mathbf{X}) = \quad & \textstyle\sum_{t=1}^{T-1} -\frac{1}{2} \left\| \mathbf{x}(t+1) - A\mathbf{x}(t) - [\mathbf{f}(\mathbf{x}_p(t), u(t)) - A\mathbf{x}_p(t)] \right\|_{Q_t}^2 \\ & + \textstyle\sum_{t=1}^{T} \frac{1}{2} \left\| [\mathbf{y}_{base}(t) - \mathbf{h}_{base}(\mathbf{x}_p(t)) + H_{base}\mathbf{x}_p(t)] - H_{base}(\mathbf{x}_p(t))\mathbf{x}(t) \right\|_{\Xi_{base}}^2 \\ & + \textstyle\sum_{t=1}^{T} \frac{1}{2} \left\| [\mathbf{y}_{rel}(t) - \mathbf{h}_{rel}(\mathbf{x}_p(t)) + H_{rel}\mathbf{x}_p(t)] - H_{rel}(\mathbf{x}_p(t))\mathbf{x}(t) \right\|_{\Xi_{fail}}^2 \\ & + \tfrac{1}{2} \left\| \mathbf{x}(1) - \mathbf{x}_0 \right\|_{Q_0}^2 - \log p_1 - \log \eta_{(y|x),1} - \log \eta_x \end{aligned} \tag{34}$$

Then, in the form of an optimization problem similar (30), we compute the MAP estimate for $\mathbf{X}$ by solving the optimization problem

$$\mathbf{X}_{1,ls} = \arg\min_{\mathbf{X}} \left(L_1(\mathbf{X}) \right) \tag{35}$$

A linear system of equations corresponding to (35) can be solved similar to (30) to yield the loss index $L_{1,ls}$.

It is possible to compute the empirical likelihood of any permutation of sensor failures, though the computational complexity would grow exponentially with the number of sensors. Alternatively, one can consider an explicit subset of likely failure modes. For instance, two relative sensors may have some likelihood to fail individually, and some likelihood to fail simultaneously due to a common mode failure. By including the four permutations, we can perform effective but simple maximum likelihood discrimination on the system fault state.

C. Maximum Likelihood Estimation

In this section we give the solution to the maximum likelihood estimation problem, using sparse matrix inversion. The underlying optimization problems, (30) and (35), are unconstrained quadratic programs with positive definite objectives. Let the subscript N define sets of variables from time 1 to T, and the identity matrix of dimension p is $I(p) \in \mathbb{R}^{p \times p}$. Then,

$$R_N = I(N) \otimes R \qquad Q_N = I(N) \otimes Q \tag{36}$$

where R_N is the covariance matrix for sensor observations over the time horizon, given by a Kronecker tensor product, and Q_N is the covariance matrix for the state propagation noise. The states, measurements, and control inputs are grouped as,

$$\mathbf{X}_N = \begin{bmatrix} \mathbf{x}(1) \\ \vdots \\ \mathbf{x}(T) \end{bmatrix} \qquad \mathbf{Y}_N = \begin{bmatrix} \mathbf{y}(1) \\ \vdots \\ \mathbf{y}(T) \end{bmatrix} \qquad \mathbf{U}_N = \begin{bmatrix} \mathbf{u}(1) \\ \vdots \\ \mathbf{u}(T) \end{bmatrix} \tag{37}$$

The measurement Jacobians are grouped into H_N, and the matrix D_N takes the difference between each state and the predicted state based on the prior time step's unforced dynamics.

$$H_N = \begin{bmatrix} H(\mathbf{x}_p(1)) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots H(\mathbf{x}_p(T)) & \end{bmatrix} \qquad D_N = \begin{bmatrix} I(N) & -A(\mathbf{x}_p(1)) & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & I(N) & -A(\mathbf{x}_p(T-1)) \\ 0 & \cdots & 0 & I(N) \end{bmatrix} \tag{38}$$

The quadratic can be efficiently solved by solving for $\mathbf{X}_N$ such that the gradient of the objective is zero. This requires only a sparse matrix inversion, with the dimension scaling linearly with the length of the time history window. The gradient of (30) is

$$\nabla_{\mathbf{X}_N} L_0(\mathbf{X}) = \left(H_N^T Q_N H_N + D_N^T R_N D_N \right) \mathbf{X}_N + C_N^T Q_N \mathbf{Y}_N + D_N^T R_N \begin{bmatrix} \mathbf{x}_0 \\ B_1 \mathbf{u}_1 \\ \vdots \\ B_N \mathbf{u}_N \end{bmatrix} \tag{39}$$

Setting the gradient to zero, we obtain the least squares solution with respect to the negative log-likelihood for the null hypothesis,

$$\mathbf{X}_{0,ls} = \left(H_N^T Q_N H_N + D_N^T R_N D_N\right)^{-1} \left(C_N^T Q_N \mathbf{Y}_N + D_N^T R_N \begin{bmatrix} \mathbf{x}_0 \\ B_1\mathbf{u}_1 \\ \vdots \\ B_N\mathbf{u}_N \end{bmatrix} \right) \tag{40}$$

Similarly, for the fault hypothesis, taking the gradient of (35) leads to the least squares problem,

$$\mathbf{X}_{1,ls} = \left(D_N^T R_N D_N\right)^{-1} \left(D_N^T R_N \begin{bmatrix} \mathbf{x}_0 \\ B_1\mathbf{u}_1 \\ \vdots \\ B_N\mathbf{u}_N \end{bmatrix} \right) \tag{41}$$

These solutions are the state vectors with maximum likelihood. The value of the maximum likelihood is found by substituting the result of (40) into (30) and the result of (41) into (35). The resulting residuals are the negative log likelihood. Taking the exponent of their negative gives the relative likelihood of each hypothesis. The resulting likelihood ratio of the hypotheses allows for online decision logic to perform identification of faults and recovery of the system, as presented in the next section.

To analyze and validate the formulated approaches, trade studies were performed, motivated by a need in the manned spaceflight program to dock vehicles without pilots present, potentially even without GPS available. Such schemes are expected to be beneficial to lunar and planetary exploration missions, by allowing on-orbit build-up of spacecraft. In this section, we proceed by giving details about the effects and details considered in the simulation model. Then, we discuss relative navigation estimation equations. We present a scheme for recovering the system using an abort logic based on probabilistic measures. Finally, we give the results learned from running simulations. These results demonstrate a need to consider schemes such as the one presented in this paper in order to improve the safety and reliability of automated relative navigation systems.

IV. Spacecraft Automated Rendezvous and Docking

A. Simulation Model

The example application of fault tolerant relative navigation in the previous section was implemented in a full 6-DOF simulation, to estimate the relative position between two spacecraft in low Earth orbit, using a relative navigation sensor subject to various faults. A sensor modeled to be similar to current spacecraft rendezvous laser-based systems provides range, bearing (azimuth and elevation), range rate, and relative attitude measurements. The noise of the measurements increases with range. The inertial navigation system (INS) is simulated with the noise specifications for a Honeywell SIGI system. A Simulink block takes the output of the relative navigation sensor, as well as the IMU data from the INS, and solves for the most likely fault state of the system. This block provides the most likely fault state to the navigation filter, enhancing the utility of information from the relative navigation sensor. If the sensor data is deemed faulty, the navigation filter is able to reject those measurements, and prevent corruption of a filter which is designed to handle only Gaussian noise of limited magnitude. However, when the data is deemed fault free, the navigation filter can incorporate the data as usual, without modification to the algorithm, and further improve the relative state estimate of the system.

In the simulation implementation, the fault tolerant relative navigation settings can be turned on and off using switches while the simulation is running. This allows faults to be triggered, and ability to use or ignore the fault warning signal generated by the fault detection algorithm for the navigation filter.

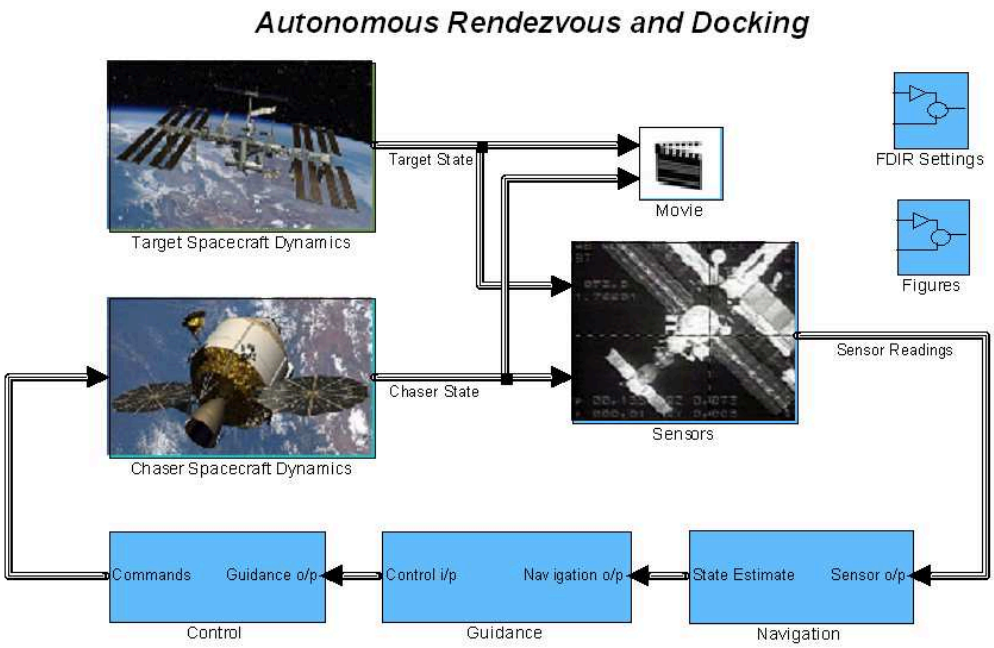

Figure 3. Simulink simulation used for the AR&D application. The spacecraft dynamics of a chaser spacecraft and a target spacecraft are modeled in 6 degrees of freedom, and relative navigation and inertial sensors are modeled. The guidance scheme transitions the control system between operating states and trajectories, and the control system computes optimal thruster configurations. The fault tolerant relative navigation component, an augmentation to the relative navigation sensor suite, provides information to the navigation system, which runs a standard extended Kalman filter (EKF).

B. Relative Navigation Estimation

The guidance logic in the simulation is programmed to perform the terminal phase of an R-bar approach of the chaser vehicle to the target vehicle, along the radial line from the Earth to the target spacecraft. For instance, this is one approach method of the Space Shuttle Orbiter to the ISS.[22] It's assumed here that the chaser spacecraft has relative navigation sensors with a fixed pointing direction and limited field of view, so throughout docking, the spacecraft is controlled to point at the target.

Consider a subset of measurements made by the relative navigation sensor, measuring the in-orbital-plane elevation angle, α, and range of the target spacecraft, ρ, from the chaser spacecraft's perspective, as shown in Figure 1. The corresponding measurement model is given by simple trigonometry, relating the measurement to the real-world states.

$$\mathbf{h}_{rel}(\mathbf{x}(t)) = \begin{bmatrix} \alpha(\mathbf{x}(t)) \\ \rho(\mathbf{x}(t)) \end{bmatrix} = \begin{bmatrix} -\arctan\left(\frac{z_{c/t}}{x_{c/t}}\right) - \theta_c \\ \sqrt{x_{c/t}^2 + z_{c/t}^2} \end{bmatrix} + \begin{bmatrix} \xi_\alpha(t) \\ \xi_\rho(t) \end{bmatrix} \tag{42}$$

where the independent white Gaussian measurement noises $\xi_\alpha(t)$ and $\xi_\rho(t)$ are zero mean and have covariance Ξ_{rel}. The relative navigation coordinates are given by $x_{c/t} = x_c - x_t$ and $z_{c/t} = z_c - z_t$, where x_c, z_c, x_t, and z_t are the coordinates of the chaser and target spacecraft, respectively, in the LVLH coordinate frame. The Jacobian of the measurement model in (42) is

$$H_{rel}(\mathbf{x}_p(t)) = \begin{bmatrix} \frac{z_{c/t}}{x_{c/t}^2+z_{c/t}^2} & \frac{-x_{c/t}}{x_{c/t}^2+z_{c/t}^2} & -1 & 0 & 0 \\ \frac{x_{c/t}}{\sqrt{x_{ct}^2+z_{ct}^2}} & \frac{z_{c/t}}{\sqrt{x_{ct}^2+z_{ct}^2}} & 0 & 0 & 0 \end{bmatrix} \tag{43}$$

The motion model for this system, extending (1), needs to include inertial measurements, which serve to

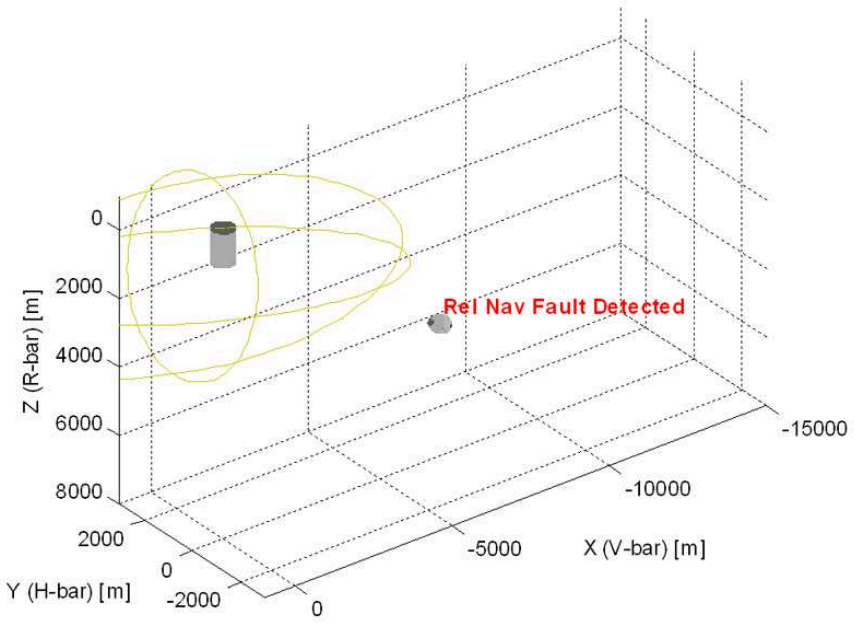

Figure 4. Visualization of R-bar docking procedure, along the radial line from the Earth to the target spacecraft, with a fault detected in the relative navigation sensor data. Note, the spacecraft are shown at 500x magnification.

predict the change in state of the system. The modified equation is

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{e}(t) \tag{44}$$

The A matrix represents the unforced dynamics of the system, and the B matrix maps the observed specific inertial forces onto their correspond states. For predicting in-orbital-plane motion, the dynamics can be found using the Clohessy-Whilshire equations,

$$\begin{bmatrix} \dot{x}_{c/t} \\ \dot{z}_{c/t} \\ \dot{\theta}_{c/t} \\ \ddot{x}_{c/t} \\ \ddot{z}_{c/t} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\omega \\ 0 & 3\omega^2 & 0 & -2\omega & 0 \end{bmatrix} \begin{bmatrix} x_{c/t} \\ z_{c/t} \\ \theta_{c/t} \\ \dot{x}_{c/t} \\ \dot{z}_{c/t} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_y(t) \\ a_x(t) \\ a_z(t) \end{bmatrix} \tag{45}$$

By taking the matrix exponential of the two matrices times the time step, we obtain the A and B matrices needed for (6). The noise of the inertial sensor measurements is the source of the white noise driving the system dynamics described in Section B. It has covariance Q.

C. Online Decision Logic

By computing the two loss indexes $L_{0,ls}$ and $L_{1,ls}$, the smallest index can be selected, which corresponds to the highest probability hypothesis. Knowing which hypothesis has been selected, one can switch to using the respective extended Kalman filter (EKF) for the estimation. Either a STRING 1 EKF using data from inertial, base, and relative sensors can be employed, or a STRING 2 dead reckoning EKF can be employed using data from the inertial and base sensors only.

So far, the discussion assumed that the batch estimation problems, (30) and (35) include the entire set of data from $t = 1$ to $t = T$. This is not suitable for on-line implementation because the size of the batch data

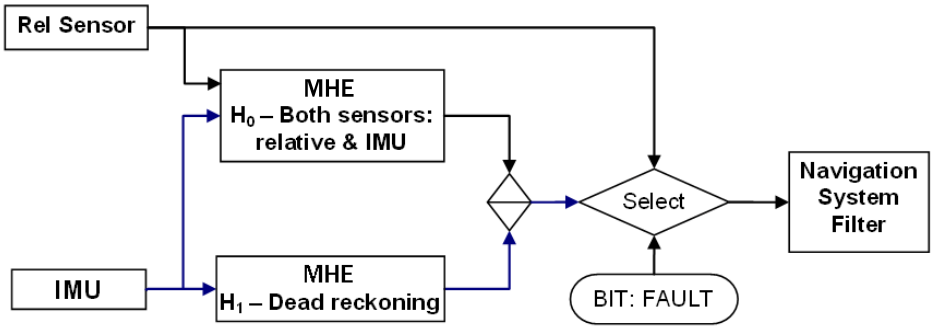

Figure 5. Relative sensor fault detection architecture for AR&D.

set will outgrow the available computational capacity. Therefore, a proposed practical implementation of the approach uses moving horizon data by retaining the last N data samples, where N is a design parameter that could be anything from 10 to 1000, for example. Using a fixed moving horizon automatically ensures a "healing" from the faults that have occurred more than the horizon length ago and have since disappeared. The further details of this algorithm depend are application specific.

For AR&D relative navigation, the fault detection scheme was found, in simulation, to perform well with a horizon of $N = 20$ relative navigation measurements. Note that many more inertial measurements fall into that time window, as the inertial measurement sampling rate is faster. The window size was found to demonstrate a reasonable trade-off between identifying faults and "healing" from the faults that have occurred more than the horizon length ago and have since disappeared. As shown in Figure 5, the navigation filter, an EKF, uses the maximum likelihood outputs computed for each hypothesis. If the null hypothesis is most likely, the relative navigation sensor's data is incorporated in the measurement update step in the EKF. However, if the fault hypothesis is more likely, then the relative navigation sensor's data stream is ignored, and dead-reckoning is performed. When sensor measurement updates are skipped due to a measured fault state, there is no method to reduce the covariance of the relative state error estimate. As a result, this error estimate can grow in time due to accrual of error during dead-reckoning. When the error becomes too large, abort logic is necessary to return the system to a safe state, to achieve a fixed likelihood value of avoiding collision, as described in the subsequent section.

D. Abort Logic

When a relative navigation sensor becomes faulty, the error in the navigation filter will grow with time. By incorporating fault detection logic, it is possible to reduce this error by taking the faulty sensor offline, and at the same time accurately estimate the true error of the system, without false confidence from incorporating faulty data that is treated as accurate. When the faulty sensor is taken offline, the estimated error will accumulate due to the prediction stage of the filter, Equations (12) and (13). The risk of collisions will grow with time as the uncertainty increases. To prevent this from happening, we consider abort logic, just before the collision becomes probable. This is determined in the worst case given the uncertainty. The approach which we propose for that is based on computing reachable sets, enabling verification of system capabilities.

One of the key technologies for design and analysis of safety critical and human-in-the-loop systems is verification, which allows for heightened confidence that the system will performed as desired. In the context of the present work, verification consists of proving that from an initial set of states, a system cannot reach an unsafe set of states (target). While the verification of discrete state systems is a relatively well-explored field for which efficient tools have been successfully developed, algorithms for verification of continuous state systems have been developed relatively recently:[23] verifying an uncountable (infinite) set of states represented by continuous variables requires a numerical treatment which is theoretically more difficult than for discrete systems, and harder to implement in practice. The approach used here incorporates the Hamilton-Jacobi partial differential equation (HJ PDE). The HJ PDE framework models the envelope as

the zero sublevel sets of a user defined function. This function is used as a terminal condition for an HJ PDE that is integrated backward in time. The result of the integration provides a new function, whose zero sublevel sets can be shown to be the set of points that can reach the target. The HJ PDE framework also provides a set-valued control law, which indicates the range of allowable control inputs that can be applied as a function of the continuous state, to keep the system inside the maximal controllable set. The benefit of this approach, sometimes called reachability analysis, is that it provides a proof (for the mathematical models used) that the system will not reach the target.

The abort logic uses the results of a reachable set analysis. Although the simulation captures the relative motion between chaser and target, for this analysis, the chaser is assumed stationary. A 2-degree of freedom (2DOF) model in the plane of the orbit is used to represent the chaser motion. Navigation uncertainty is modeled as a finite radius "target set" around the target vehicle, with the radius proportional to the level of uncertainty, as provided by the navigation filter and the fault detection logic. For a range of uncertainty levels, the set of configurations of the chaser vehicle which are guaranteed to be collision-free for a given time horizon are computed, using a publicly available software tool for computing reachable sets. The results are shown in Figure 6. In addition, the configurations which are guaranteed to be safe if the chaser decelerates at maximum rate are computed. The results are in terms of the minimum safe operating range of chaser with respect to target, to guarantee collision avoidance over this time horizon. Should the level of uncertainty change, as provided by the navigation filter and fault detection logic, the system can react accordingly, according to the reachable set analysis, using the safe control input.

E. Results

The application of the proposed fault tolerant relative navigation method to a series of test cases provided promising results, shown in Figures 7 and 8. Manually triggered faults were detected immediately, causing the navigation filter to ignore the measurements and use dead reckoning to span gaps in the data stream.

In Figure 8, a moving horizon estimator using the EKF framework described in Section C is shown with and without the probabilistic fault detection signal. When the signal is included, no substantial error is accrued, whereas when there is no fault detection, there is temporary divergence in the estimate. This divergence could have caused the control system to severely correct the spacecraft when such correction may have led to loss of view of the target vehicle–a potentially confusing state for some system designs.

A series of Monte Carlo simulations were also performed, with thousands of trials, for near field approach scenarios in the range of 100 m to 1 m. The results match the expected performance of the system. When the injected fault was randomly large with respect to the assumed fault noise magnitude, it was often detected

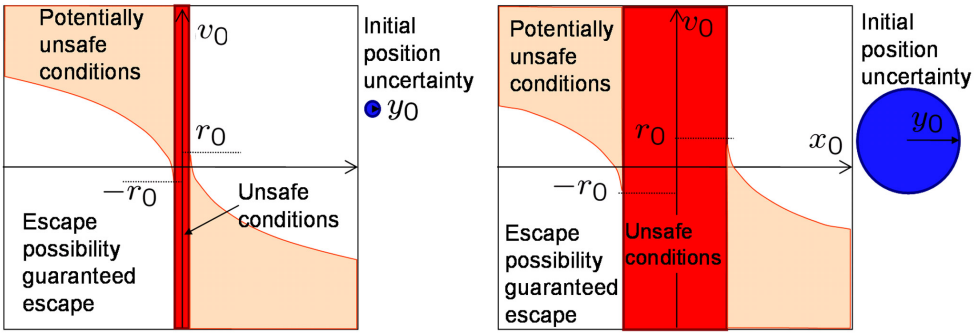

Figure 6. This plot shows the 2D sets for safety along 1 axis of motion, with position x_0 and velocity v_0. The shown set of unsafe conditions is where the estimated distance between the chaser and the target is less than r_0, a distance which is a function of the uncertainty. The Reachable Set (potentially unsafe conditions) contains the states for which the chaser might hit the target despite the best effort at deceleration. The Controlled Safe Set (escape possibility guaranteed) contains the states for which the chaser can avoid the target by decelerating at a maximum rate.

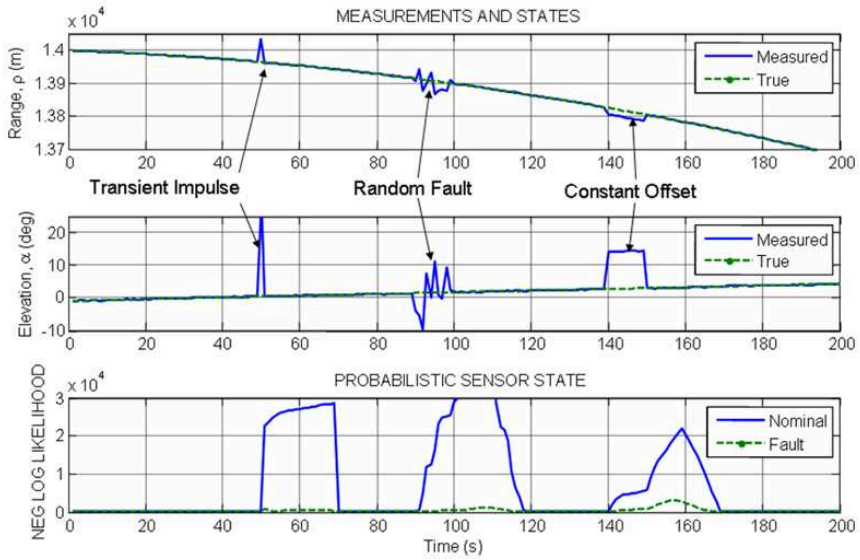

Figure 7. Probabilistic fault detection using moving horizons. The upper two plots show the faults injected into the simulated sensor data, and the bottom plot shows the negative log likelihood of the null and fault hypotheses. All three faults are quickly and accurately detected.

immediately. The time delay to detection was approximately described by an exponential distribution in time. This result is well supported by understanding the effect of performing the batch solution to obtain a moving horizon estimate. The older the data, the less the covariance on the related state estimate. This covariance decreases approximately exponentially, thus increasing the likelihood of capture faults as time progresses. When the injected fault was randomly small with respect to the assumed fault magnitude, however, the frequency of being identified immediately was most often less than the frequency of being captured a short number of time steps later, as shown in Figures 7 and 8. This occurs because these small faults would not be captured by an algorithm that only considers one time step, but can be captured later by this algorithm that considers the likelihood of a sequence of timesteps, making it possible to identify improbable sequences of measurements. With the particular settings chosen, faults of random magnitude were detected 70 to 80% of the time, with a false positive rate of under 0.01%. Note that some of the smaller faults were not only far smaller than the expected fault size, but also too small to be of consequence. Further Monte Carlo simulations could be performed for a specific system to calibrate the algorithm to obtain the desired tradeoff between false positives and detection likelihood.

V. Conclusion

This paper presents an architecture for fault tolerant relative navigation. A fault tolerant inertial navigation system is used as a reference for detecting faults of a relative navigation sensor and recovering from these faults. Models of multiple fault states of the system are used and empirical likelihood of each is computed. The fault detection and identification is accomplished by selecting the fault state of the system with maximum empirical likelihood. Fault tolerance of the navigation system is achieved by reporting the detected and identified faults to the main navigation filter which may discard the faulty relative navigation sensor data and, instead, temporarily use a reliable inertial navigation system. The guidance system may also act on the fault information, taking actions to recover the system to a safe state.

This logic was implemented in simulation for the automated rendezvous and docking (AR&D) of spacecraft (The work supports the NASA Exploration Program). The completed study shows the need for rigorous development of the fault tolerance and redundancy management architecture and algorithms for AR&D. Many system trades are necessary with careful attention to failure modes as potential faults are not

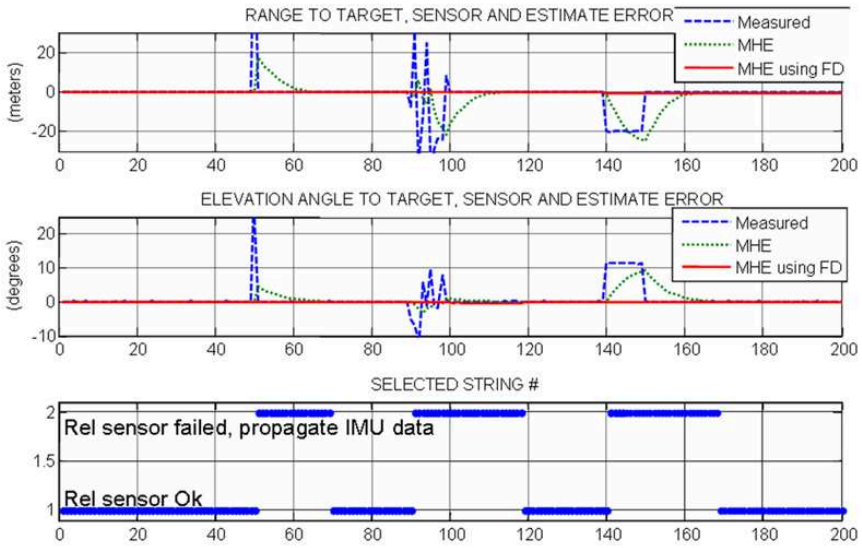

Figure 8. Effect of using the proposed online decision logic for fault tolerant relative navigation. The raw measurements are corrupted, as in Figure 7. The true state of the system is estimated using a moving horizon estimator (MHE). The results are given for a standard MHE, and for an MHE using the fault tolerant relative navigation algorithm to reject relative navigation measurements when a fault state is detected. This comparison shows the potential for the proposed fault tolerant algorithm to mitigate disturbances caused by faulty relative navigation sensors.

obvious. As an example, standard fault rejection schemes developed for Kalman filters can fail for an Extended Kalman Filter due to the bias incurred by linearizing over small windows of nonlinear measurements. On the other hand in this situation, the proposed moving horizon approach was able to detect the faults and prevent the navigation filter from incorporating the faulty measurements.

The developed fault tolerant relative navigation schemes could be used in many potential configurations and future applications. The fault tolerant navigation schemes presented here can be used in different configurations in a wide range of applications. Many emerging applications of automated/autonomous space, ground, marine, and air vehicles require relative navigation, both with respect to other vehicles, and with respect to other objects. Disturbances in vehicle environment such as lighting, glint, and debris can cause relative navigation sensors to fail. The fault tolerance approach and techniques presented in this paper address these problems.

References

[1]Everett, S., Markin, K., Wroblewski, P., and Zeltser, M., "Design considerations for achieving MLS Category III requirements," *Proceedings of IEEE*, Vol. 77, No. 11, 1989, pp. 1752–1761.

[2]Teo, R. and Tomlin, C. J., "Computing Danger Zones for Provably Safe Closely Spaced Parallel Approaches," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 3, 2003, pp. 434–443.

[3]Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P., "Winning the DARPA Grand Challenge," *Journal of Field Robotics*, Vol. 23, 2006, pp. 661–692.

[4]Polites, M. E., "Technology of Automated Rendezvous and Capture in Space," *Journal of Spacecraft and Rockets*, Vol. 36, No. 2, March-April 1999.

[5]NASA, "Overview of the DART Mishap Investigation Results. For Public Release." April 2006, http://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf.

[6]Busse, F. D. and How, J. P., "Real-Time Experimental Demonstration of Precise Decentralized Relative Navigation for Formation Flying Spacecraft," *In Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, August 2002.

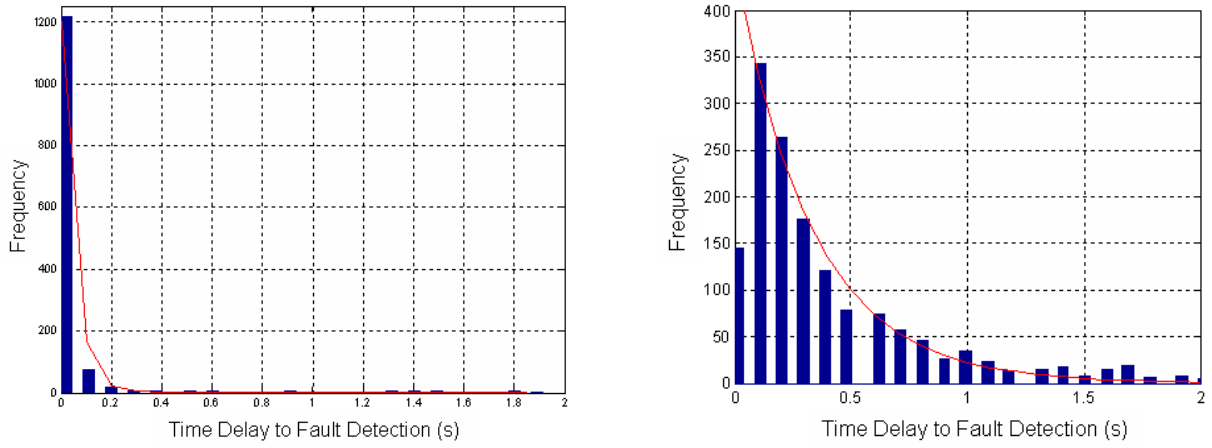

Figure 9. Histograms of the Monte Carlo simulation results for two near field approach scenarios. In (a), the random fault magnitude was large, leading to rapid detection, that can be modeled by a scaled exponential curve of e^{-20t}. However, in (b), the random fault magnitude was small, leading to a lower initial detection rate. The curve shown is a scaled version of e^{-3t}, though it does not capture the initially lower rate of detection. This result is as expected–small faults are difficult to detect, and require the consideration of several time steps to be accurately identified.

[7]Wolfe, J. D. and Speyer, J. L., "Effective Estimation of Relative Positions in Orbit Using Differential Carrier-Phase GPS," *In Proceedings of the AIAA Guidance, Navigation and Control Conference*, Providence, Rhode Island, August 2004.

[8]LeMaster, E. A. and Rock, S. M., "A Local-Area GPS Pseudolite-Based Navigation System for Mars Rovers," *Autonomous Robots*, Vol. 14, No. 2-3, March 2003.

[9]Lerner, U., Parr, R., Koller, D., and Biswas, G., "Bayesian fault detection and diagnosis in dynamics systems," *In Proceedings of AAAI*, Austin, TX, 2000.

[10]de Kleer, J. and Williams, B. C., "Diagnosis with Behavior Modes," 1992, pp. 124–130.

[11]Chen, J. and Patton, R., *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers, Norwell, MA, 1999.

[12]Simani, S., Fantuzzi, C., and Patton, R. J., *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*, Springer, New York, NY, 2003.

[13]Hsiao, T. and Tomizuka, M., "Sensor Fault Detection in Vehicle Lateral Control Systems," *In Proceedings of the AACC American Control Conference*, Portland, OR, June 2005, pp. 5009–5014.

[14]Allerton, D. J. and Jia, H., "A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems," *Journal of Navigation*, Vol. 58, 2005, pp. 405–417.

[15]Robertson, D. G., Lee, J. H., and Rawlings, J. B., "A Moving Horizon-Based Approach for Least-Squares Estimation," *AIChE Journal*, Vol. 42, No. 8, August 1996.

[16]Samar, S., Gorinevsky, D., and Boyd, S., "Moving Horizon Filter for Monotonic Trends," *In Proceedings of the IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[17]Ferrari-Trecate, G., Mignone, D., and Morari, M., "Moving Horizon Estimation for Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 47, No. 10, October 2002, pp. 1663–1676.

[18]Fretheim, T., Vincent, T. L., and Shoureshi, R., "Optimization Based Fault Detection for Nonlinear Systems," *In Proceedings of the AACC American Control Conference*, Arlington, VA, June 2001, pp. 1747–1752.

[19]Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2005.

[20]Bell, B. M., "The Iterated Kalman Smoother as a Gauss-Newton Method," *SIAM Journal of Optimization*, Vol. 4, No. 3, August 1994, pp. 626–636.

[21]Lala, J. and Harper, R., "Architectural principles for safety-critical real-time applications," *Proceedings of IEEE*, Vol. 82, No. 1, 1994, pp. 25–40.

[22]Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Cambridge University Press, New York, NY, 2003.

[23]Bayen, A., Mitchell, I., Oishi, M., and Tomlin, C., "Aircraft Autolander Safety through Optimal Control Based Reach Set Computation," *AIAA Journal on Guidance, Control and Dynamics*, Vol. 30, No. 1, Jan-Feb 2006, pp. 68–77.