

Proximity Operations for Space Situational Awareness

Spacecraft Closed-Loop Maneuvering Using Numerical Simulations and Fuzzy Logic

Timothy Carrico

Analytical Graphics, Inc.

Travis Langster

Analytical Graphics, Inc.

John Carrico

Applied Defense Solutions

David Vallado

Center for Space Standards & Innovation

Mike Loucks

Space Exploration Engineering

Salvatore Alfano

Center for Space Standards & Innovation

Abstract

Maintaining space situational awareness requires an understanding of friendly and opposing force capabilities. This paper presents work towards a flexible and accurate framework for modeling rendezvous and proximity operations (RPO) within an existing simulation environment. The authors present several spacecraft close proximity maneuvering and perching techniques modeled with a high-precision numerical integrator using full force models and closed-loop control with a fuzzy logic intelligent controller to command the engines. Maneuvers, fuel use, and other parameters are documented and compared. An innovative application to design, simulate and analyze proximity and perching maneuvers, already in use for operational satellites performing other maneuvers, has been built. The system has been extended to develop closed-loop control laws to maneuver spacecraft in close proximity to another, perch and stare, conduct self-inspection, docking and other operations applicable to space situational awareness, space based surveillance and operational satellite modeling. The fully integrated end-to-end trajectory ephemerides are available from the authors in electronic ASCII text by request.

Introduction

The advances in onboard processing and high-speed communication links are enabling a new generation of semi-autonomous and fully autonomous spacecraft that are capable of operating in close proximity to other spacecraft. Proximity operations offer a tremendous opportunity to inspect, repair and monitor another spacecraft. To utilize such spacecraft to maintain space awareness and to understand the movements of foreign spacecraft performing advanced maneuvering operations, these trajectories must be modeled accurately. To support the needs of analysts, operators and commanders, this modeling must be done quickly and conveniently in a robust and flexible simulation environment. It is critical that friendly assets can respond to a changing environment which requires an adaptable and responsive support system.

The authors have used the standard application program interface (API) to an existing simulation environment, *Satellite Tool Kit (STK)* [1], to create a framework in which closed-loop orbit control for proximity operations and docking can be modeled and studied. This framework is reusable and adaptable to various mission needs. The simulation is built around the *STK/Astrogator* module which is currently used to support maneuver analyses and operations for many spacecraft worldwide. Because the simulation is within the *STK* application suite, other analyses can be performed after the proximity operations trajectory is created. For example, after modeling a proximity operations trajectory, the analyst can determine times of communication links, radio interference, sun lighting conditions for power and image collection, and create predicted tracking data to support navigation studies.

This study focuses on the portion of a mission after a rendezvous has occurred within, for instance, a one kilometer range. The maneuvering spacecraft is called “Chase” and the other, usually passive spacecraft, termed the “Target.”

A closed-loop controller was created to maneuver the spacecraft to various points around the Target spacecraft, perch and simulate close inspection.

Background

Traditional operations using maneuvering spacecraft are usually governed by fuel efficiency requirements. The cost of launching mass into space requires performing maneuvers in an optimal fashion. As a result, it's not surprising that a spacecraft transferring from its initial post-launch orbit to its final on-station orbit would perform a series of optimal maneuvers. From a situational awareness point-of-view, it's likely that maneuvers would be performed at known positions in the orbits, such as apogee and perigee. In addition, these maneuvers are often separated by hours and days so that tracking can occur, giving both Chase and Target spacecraft sufficient time to recover positional information. The maneuvers are usually planned in an open-loop fashion, and system biases are analyzed and compensated for manually. With such missions, a few dozen maneuvers are considered excessive, and those that require maneuvers weekly or monthly are considered "busy" missions.

Recent discussions and technology advancements are now focusing on and robotic spacecraft tasked to perform surveying activity on Target spacecraft [2]. When a spacecraft operates in close proximity to another, it must perform hundreds or thousands of maneuvers; more if the Target spacecraft is maneuvering. Fuel efficiency may still be important for these missions, but it is somewhat sacrificed to support the primary requirements of the mission. Whereas it may be most efficient to allow the natural orbital dynamics to cause the Chase spacecraft to drift towards the Target spacecraft, it may be more important to speed up the process, or to approach from a different direction. As a result, the maneuvers are performed at times and in directions not based on orbit mechanics, but rather in response to primary mission requirements. In addition, the maneuvers must be performed in such rapid succession, possibly hundreds in a matter of minutes, requiring closed-loop control laws to maintain the desired trajectory [3]. These systems rely on near instantaneous feedback from sensors to monitor how well mission goals are being met, often with human controllers providing little if any input.

An example of a required inefficiency is a straight line approach along the velocity vector, referred to as the "Vbar" [4]. A Vbar approach is a maneuver where the Chase spacecraft maneuvers to approach the Target along a line, aligned with velocity vector of the Target spacecraft. In the coordinate system of the Target spacecraft it appears as a straight line between the two. In the image below (Fig. 1), *STK* was set up to perform a Vbar approach of the Crew Exploration Vehicle (CEV) to extract the lunar landing module.

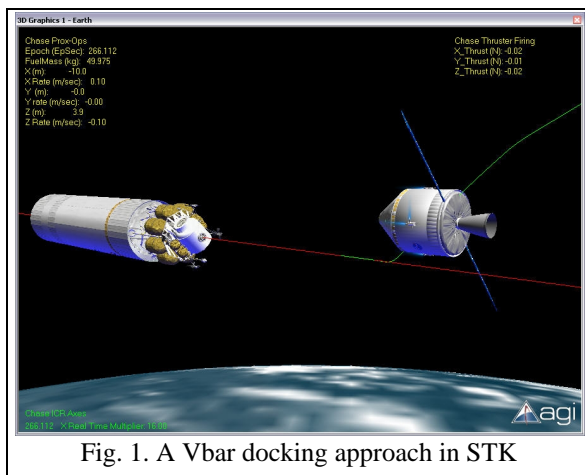


Fig. 1. A Vbar docking approach in STK

Docking in space traditionally uses this approach as one that "makes sense" in terms of how a human would want to dock two spacecraft. In such an approach the ΔV is traded for increased situational awareness allowing astronauts to be in the loop in command and control; comprehension of the maneuver necessitates a familiar geometry.

The question of what point to use a Vbar approach vs. more efficient approaches is a subject suitable for study where human factor studies are compared with ΔV tradeoffs at 1 km, 500 m, 100 m etc. Such tradeoffs in efficiency vs. situational awareness are measurable. This paper does not address human factors, but the framework does support such analysis.

Proximity operations have constraints and requirements such as autonomy, urgency, criticality, detection and other non-orbital mechanical factors that must coexist with the physics requirements of mass, force, acceleration, light, radar and communications. To support such engineering and operational planning requires accurate toolsets and understandable visualizations.

Technical Approach

The *STK/Astrogator* software is a high-precision trajectory and maneuver simulation software program used operationally to plan maneuvers. This software has two major APIs that allow it to be customized. One API allows

for the remote control of the software, enabling tasks to be automated, such as parametric studies, Monte Carlo runs, and embedding it in other applications. The other API allows a user to “plug-in” their own algorithms to be used in the calculations. The details have been previously described [5,6]. For this study, the authors used the plug-in engine model interface to create a set of smart engines that turn themselves on and off in response to a closed-loop controller. Various closed-loop control laws can be employed in this framework, and for this paper the authors created a fuzzy logic algorithm to control the engines of the Chase spacecraft.

The API provided orbit and other spacecraft state data for input to the plug-in user control algorithms. These data were converted into simulated sensor data, the control laws called, and resulting engine thrusts returned to the calling *Astrogator* program. These thruster forces were included in the force model for numerical integration.

Another advantage of using the *Astrogator* architecture and API, was to allow the proximity operations to be modeled as a phase in an overall rendezvous and proximity operations (RPO) mission. *Astrogator* enables the user to build “Mission Control Sequences” so that conventional orbit ascent maneuvers after insertion can be created, followed by a series of maneuvers to affect a multi-maneuver rendezvous, followed by proximity maneuvers. After the proximity and perching maneuvers, subsequent maneuvers can be modeled to transfer to another spacecraft, de-orbit, or “park” in a holding orbit. These end-to-end trajectories are fully numerically integrated, and can be used for other analysis such as communications studies, interference analysis, sunlight, power, sensor collection and simulated tracking data generation.

Finally, using a commercially available product enables several originations to standardize and communicate in common terms, enabling greater collaboration and reducing the risk of modeling errors.

What is proximity?

Proximity means close, but how close? Relative spacecraft maneuvering is driven by mission objectives. For docking the distance is ultimately zero meters on physical connection. For spacecraft inspection, distances are driven by sensor performance. Some terms of reference are required to understand proximity maneuvering. Analogous to ships performing underway replenishment (UNREP) where ships “meet” in a designated area prior to going alongside each other, there are operational phases required to get spacecraft within range of each other for the conduct of proximity operations. For this paper the following RPO terms will be used:

Rendezvous: Used to describe getting a spacecraft from one orbit to a control box near, but offset from, another spacecraft.

Proximity Operations: Begins after rendezvous; very near another spacecraft (e.g., < 1 km).

This study employs standard rendezvous techniques [7,8,9,10,11] modeled in *STK* to rendezvous (within 1 km) the maneuvering spacecraft (labeled “Chase”) with the spacecraft of interest (labeled “Target”). Once at rendezvous, a closed-loop controller was modeled to maneuver the spacecraft to various points around the Target spacecraft, simulating a close inspection.

Fuzzy Logic Controller for Spacecraft Maneuvering

Fuzzy logic was selected for the mathematical model for calculating the accelerations to maneuver and perch the Chase spacecraft around the Target spacecraft and has been used for control law modeling spacecraft engines for lunar landing with *Satellite Toolkit* [6]. Fuzzy logic is a proven method for the development of control laws in human-rated and other critical systems. Much research has been done by NASA and others in the application of fuzzy logic to maneuver planning [12] and proximity operations controllers [13,14,15].

Applied to control law development, fuzzy logic provides a method for the mathematical computation of human expertise expressed in linguistic control decision terms, such as Near, Far, Slow, and Fast and the transitions between. Fuzzy logic is a heuristic method that lends itself to complex control and decision environments where multiple expertises are required. The process is deterministic, for every given set of inputs, the outputs will be the same; an important feature in fuzzy logic control for critical systems. Despite the name “Fuzzy,” the control is anything but, and provides a rigorous mathematical model based in discrete formulas, not probability.

The authors found this method of modeling proximity spacecraft maneuvering to be useful by providing an understandable, maintainable method to encode subject matter expertise. A language-based approach opened a dialog between simulation modelers, astrodynamics, astronautic, aerospace engineers and senior engineering decision makers during the development and demonstrations of this simulation.

Using *Satellite Tool Kit (STK)* with the *Astrogator* module, spacecraft can be modeled to a high level of fidelity and set into an accurate physics environment which includes gravitational forces, solar pressure and other space environmental conditions. *STK/Astrogator* is designed to model the physics of spacecraft flight and can fire engines based on various conditions or via external algorithms as in a closed-loop control law process. Controlling the engines in this simulation environment allows for accurate physics, numerous analytical tools and data outputs such as fuel consumption, burn rate and range rate.

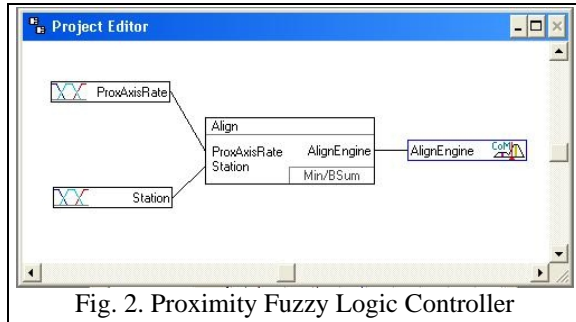


Fig. 2. Proximity Fuzzy Logic Controller

The authors extended the engine model in *STK* to call an external fuzzy logic algorithm to control the maneuvering spacecraft's engines in accordance with an expert rule base. The algorithm was developed in a commercial software product, *FuzzyTech* [16], and compiled into a function callable library which received real numbers regarding velocity and range and returned engine commands. The fuzzy logic algorithm consists of two input sets (velocity and position to a waypoint along an axis), a rule set and an output set (actuator command) (Fig. 2). The inputs are mapped to real numbers from the running simulation in

STK. In this closed-loop process, the control algorithm is called every propagation step with relative speed and distance data to the next waypoint.

A relative waypoint scheme was created to affect the proximity maneuvers. In this method the spacecraft is assigned relative waypoints and perching times around the Target spacecraft. The waypoint to waypoint maneuvering profile, loaded once at run time, is persisted in an Extensible Markup Language (XML) format. This allows the analyst to specify desired waypoints by simply editing a text document then propagating the proximity maneuver ephemerides and lends itself to automation for trade studies on proximity maneuvering strategies.

A local horizontal system (Fig. 3) was defined at the Target with in-track, cross-track and radial components. The coordinate system was set up for this environment to have the X+ along the velocity vector (in-track), Z+ along the radial and Y+ along the cross-track.

The controller is provided the velocity and distances in X, Y and Z separately and analyzes the motion on each axis for control to the engine aligned on the same axis. The simulation was set up so that every one second the controller receives an input of velocity and distance in the X direction relative to its current waypoint. By inputting all three axes simultaneously, the resultant engine commands maneuver the spacecraft in the direction of the next waypoint and hold it in assigned positions for specified periods of time (perch).

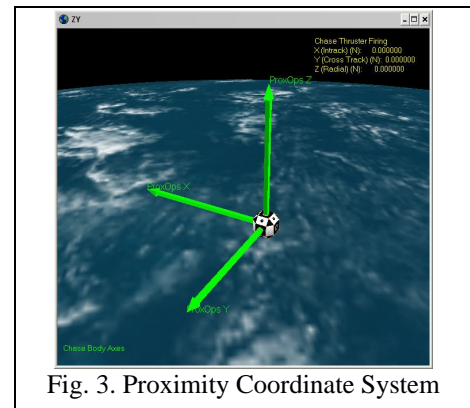


Fig. 3. Proximity Coordinate System

| Table 1 - Initial Spacecraft "Chase" Configuration | |
|--|----------------------------|
| Drag Area: | 1e-006 km ² |
| SRP Area: | 1e-006 km ² |
| Dry Mass: | 50 kg |
| Fuel Mass: | 50 kg |
| Total Mass: | 100 kg |
| Area/Mass Ratio: | 1e-008 km ² /kg |
| State Vector, Earth Inertial, Keplerian | |
| Semimajor Axis | 6778.0213414722066000 km |
| Right Ascension of Ascending Node | 0 deg |
| Eccentricity | 0.0001024111472531 |
| w: | 0 deg |
| Inclination | 56.10000000000004 deg |
| True Anomaly | 359.9999999999435 deg |

Simulation Environment and Spacecrafts' Initial Conditions

For the following example maneuvers, the spacecrafts' initial conditions and orbital positions are the same and provided below (Table 1). The simulations consist of two spacecraft that can be placed in any orbit close to each other. The *STK* environment supports a complete analysis from launch, to rendezvous, to proximity operations and safe orbit positioning. One use case (Case 5) involves a geostationary orbit proximity operation; the others (Cases 1-5) use the initial state vector (Table 1) for the Target spacecraft.

The simulation environment set up in *STK* consisted of a full force model. The position of each spacecraft is numerically integrated using Cowell's formulation of the equations of motion [17] (in inertial space). Once the environment was set up, several simulations were run to study various maneuvers around the Target spacecraft. These maneuvers were analyzed for Δv , fuel use, stability, closest point of approach, visual situational awareness and other values.

Proximity Operations Case 1 – Maneuver along Velocity Vector 50 Meters and Perch

In this initial study, the maneuvering spacecraft was set to maneuver along the velocity vector 50 meters away from the Target spacecraft and then hold that position for 60 seconds.

The simulation took approximately five seconds to run on a 1700MHz PC. The maneuvering spacecraft did perform as expected, maneuvering 50 meters along the velocity vector away from the booster section. *STK* was set up to display range, range rate and thruster firing data during animation. Additionally, the thruster firing itself was represented in the animation to provide situational awareness and greater understanding of the maneuver.

In the simulation, the spacecraft maneuvered away from the Target spacecraft with an average velocity of .1 m/sec to reach the assigned position 50 m on the velocity vector in 500 seconds consuming .011 kg of fuel (Fig. 4).

The fuzzy logic algorithm provided smooth, predictable control of the thruster firings ramping down from 3.3 N to 0.0 N in 18 seconds to accelerate the spacecraft moving along the velocity vector. At 20 meters from the assigned waypoint it began a gentle X engine firing (Fig. 5, black line) to decrease the relative velocity. The spacecraft did not overshoot the waypoint and remained perched for 60 seconds, 50 m from the Target (Fig. 5, green line).

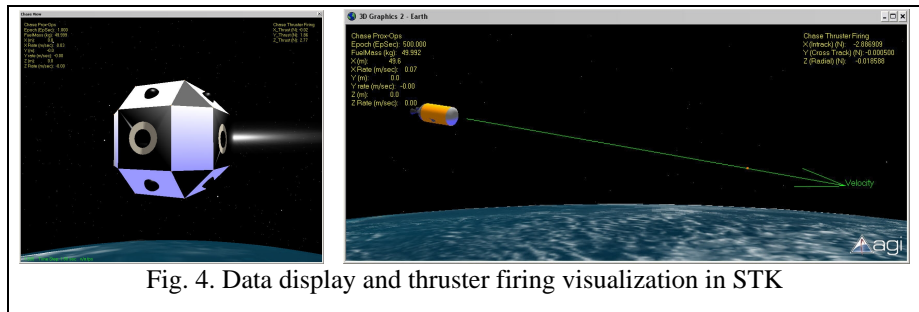


Fig. 4. Data display and thruster firing visualization in STK

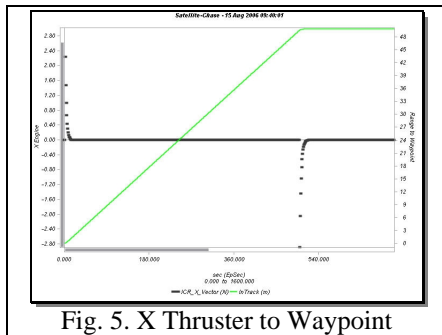


Fig. 5. X Thruster to Waypoint

To maneuver forward on the Target's velocity vector, the spacecraft accelerates forward in a series of engine thrusts parallel to the velocity vector of the initial circular orbit. The result is a change in semimajor axis with each engine firing increasing the apogee. To perform a straight line maneuver required proportional application of the Z engines (radial) to compensate for the change in the Chase spacecraft's orbit.

Analysis of this maneuver demonstrated success of the control law and concept of using *STK* in a closed-loop control simulation to perform relative maneuvers. Successive studies were built off this initial case.

Proximity Operations Case 2 – Maneuver Around the Target for Close Observation

In this example, the maneuvering spacecraft was set to maneuver completely around the Target spacecraft, simulating an inspection of the space shuttle's airframe. A series of waypoints were coded in XML representing a path around the space shuttle beginning from within the cargo hold. Several different paths were created and simulated.

The controller provided smooth control, maneuvering over a range of waypoints and perching tasks (Fig. 6). Since the controller was imbedded as a "plug-in" into *STK*, the thrust and

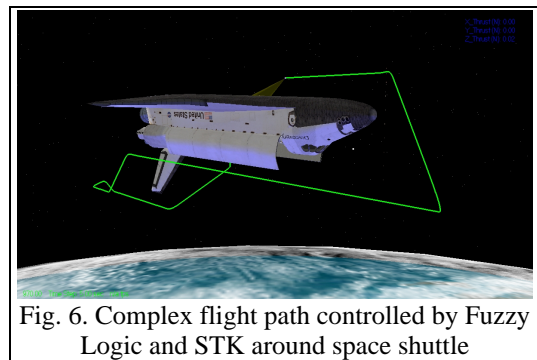


Fig. 6. Complex flight path controlled by Fuzzy Logic and STK around space shuttle

fuel consumption data were available in the graphing and reporting tools providing for quick analysis of the maneuver. Once set up, a change in a maneuver was easily made in seconds by modifying the XML file. Likewise, fuzzy logic rules could be edited in minutes providing for a productive workflow to examine issues such as actuator

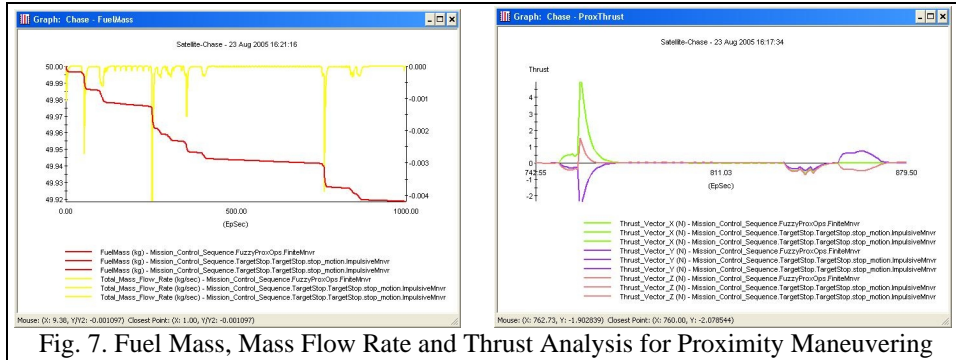


Fig. 7. Fuel Mass, Mass Flow Rate and Thrust Analysis for Proximity Maneuvering

sizing, sensor placement and proximity operation planning. Data from the maneuvers were plotted in graphical form to study fuel use and thruster set operation (Fig. 7).

The simulation and analysis environment provided situational lighting angles, relative velocity, position, azimuth, elevation, and range.

An additional test involved using the environment for actuator size trade studies. The thruster set in this mode had a maximum thrust of 10 N. The engine maximum thrust was adjusted until it could no longer maneuver through an assigned set of waypoints. The image (Fig. 8) shows the result of an original path in a square shape (red ephemeris) around the Target with progressively smaller sized engines. For each run, the fuzzy logic controller used the available thrust size to attempt maneuvering to the waypoints. Despite less and less elegant maneuvers, the engine was able to make the assigned waypoints (for this set of points only) until the engine was scaled down to about 1/100th of its original size (purple ephemeris).

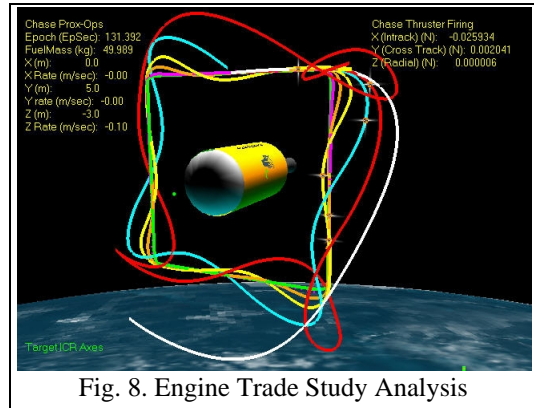


Fig. 8. Engine Trade Study Analysis

Proximity Operations Case 3 – Monte Carlos Runs for Control Law Stability Analysis

In this example, a 100 m by 100 m control box was placed around a point in orbit and the Chase spacecraft randomly placed within, with a fuzzy logic controller to maneuver to the center of the box (Fig. 9). The purpose of this case was to set up an environment for the analysis of control law stability by generating a statistically significant amount of data on maneuvering to a waypoint.

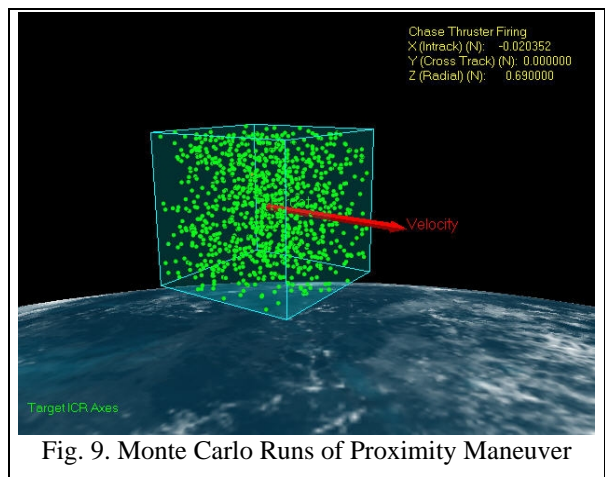


Fig. 9. Monte Carlo Runs of Proximity Maneuver

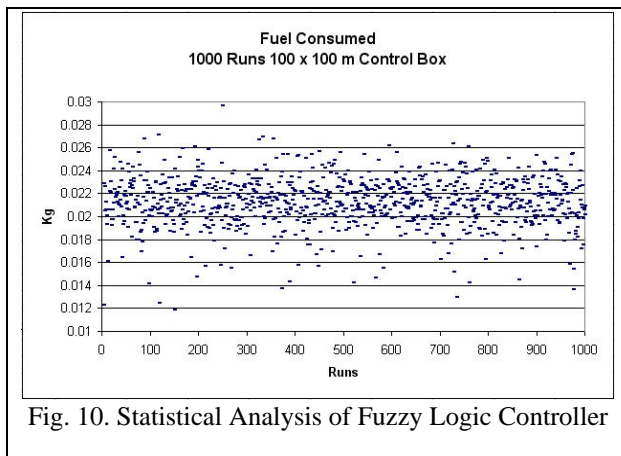


Fig. 10. Statistical Analysis of Fuzzy Logic Controller

STK provided the ability to visualize all of the Monte Carlo runs at once in an animation. Each run was saved as an ephemeris text file then loaded into a separate point object. With this capability we could see the progression of each run and visually detect anomalies as well as analyze them in data reports and graphs.

All runs reached the desired waypoint within approximately 500 seconds with no apparent anomalies in control. The algorithm tended to align the spacecraft on the X, Y or Z axis prior to closing into final position due to control laws designed around a three axis rule base (These laws are readily tunable to achieve different behavior.) Various parameters such as fuel consumption, ΔV and orbital elements were output to spreadsheets and graphs for analysis (Fig. 10). This use case demonstrated a suitable environment to perform control law analysis, including a unique ability to visualize statistical data in 3-D revealing patterns that may not have been noticed in tabular data analysis.

Proximity Operations Case 4 – Intelligent Control with Maneuver Cautionary Control Based on Sensor Fusion Error, Actuator Error and Positional Residual Error

In this example, a fuzzy logic controller was developed that will respond to error conditions to slow, hold or back up the maneuvering spacecraft as appropriate. Actuator failures, differences in sensor measurements (range), differences in estimated versus sensor position (residual) and sensor noise all provide input into a controller that determines which sensor to use and modifies engine commands.

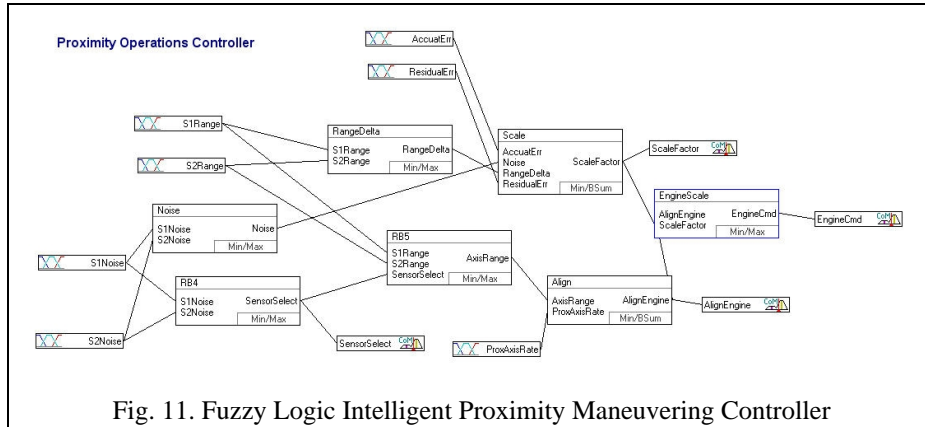


Fig. 11. Fuzzy Logic Intelligent Proximity Maneuvering Controller

If all systems are normal then the controls operate like the previous examples. As sensors' data diverges, noise increases, actuators fail or residual error builds, the controller first slows the maneuvers down and, when the system further deteriorates, will stop and even back away from the Target. If conditions improve, the maneuvers resume. To support this use case, some additional capabilities were programmed into the controller to control error conditions and provide additional input variables (Fig. 11).

As sensors' data diverges, noise increases, actuators fail or residual error builds, the controller first slows the maneuvers down and, when the system further deteriorates, will stop and even back away from the Target. If conditions improve, the maneuvers resume. To support this use case, some additional capabilities were programmed into the controller to control error conditions and provide additional input variables (Fig. 11).

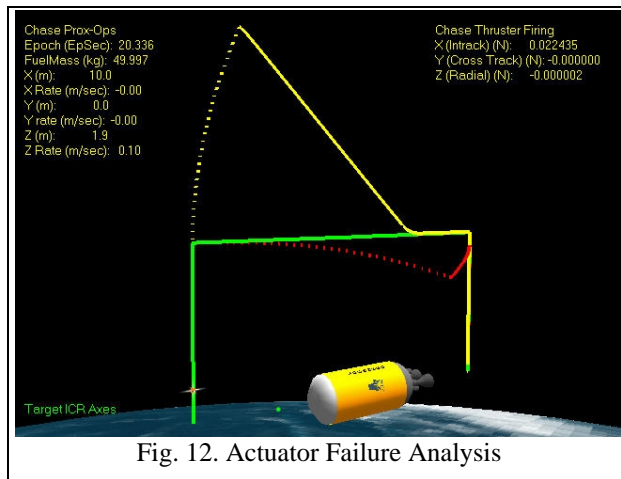


Fig. 12. Actuator Failure Analysis

The first test of this set up involved an actuator failure in the Z+ direction while the spacecraft maneuvered over the top of the Target. With no failures, the Chase spacecraft's ephemeris is a rectangular path over the Target. In the test, one of the radial (Z) engines, responsible for maintaining the straight line (see case 1) was programmed to fail completely from time T = 100 to 300 seconds into the maneuvers. The actuator failure should trigger a rule set that will take the orbital mechanical solution, based on distance and rates, and adjust it according to overall system status. The controller allows for reverse direction rules to back the spacecraft up when system errors warrant.

Fig. 12 shows the results of this case. The green (rectangular) line is the original ephemeris with no errors. The red (lowest) ephemeris is without cautionary rules. The yellow (upper most) ephemeris is the ephemeris when the system detects error states, and adjusts the engine commands according to non-orbital mechanical command and control principals of a prudent nature. The dotted portions of the ephemerides are during Z engine failure. Note that the fuzzy logic controller handles all three conditions, maneuvering the spacecraft along all waypoints while detecting anomalies and adjusting the maneuver sequences. However, the cautionary control is perhaps the most prudent; away from the Target during the period of the actuator failure.

Comparing the three conditions and maneuver control (Fig. 13), we can see the actuator failure in the data output and the controller reaction. The controller reacts by continuing to apply force upward (away from earth) along the Z radial with slow progression along the X axis until it regains a functioning radial engine. The rules move it into a safe position until fully functioning actuators are available.

Fig. 14 shows the fuzzy logic algorithm outputting an engine command as modified by a faulty actuator. The *AlignEngine* variable would normally command the engine to fire 10 N on this axis (lower right). However, the *ActuatorError* (lower left) has a significant error which caused rules to fire in the *EngineScale* rule set (upper right) modifying the engine command to 5.0152 N, which was returned to *Astrogator*.

Next, sensor noise was modeled to escalate to a point where both sensors are essentially useless to the spacecraft. This could be due to electronic problems, natural interference or jamming. From time $T = 100$ to 250 seconds, the two sensors which provide range, have increasing noise until

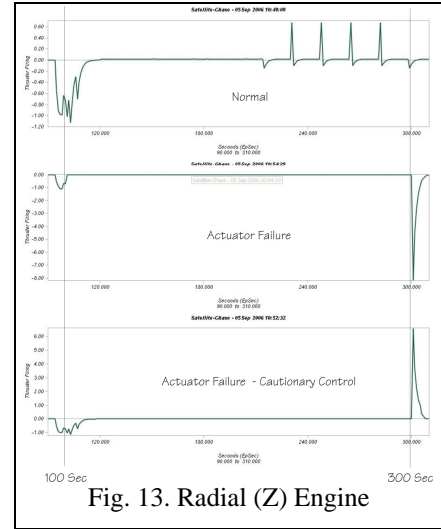


Fig. 13. Radial (Z) Engine

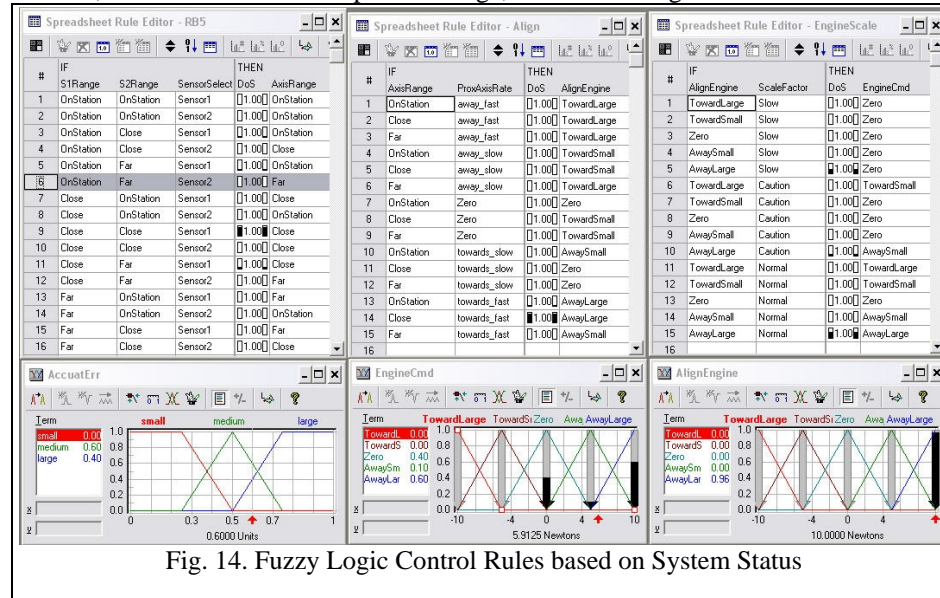


Fig. 14. Fuzzy Logic Control Rules based on System Status

seconds. The blue portion of the path shows the spacecraft resuming its flight path upon an acceptable noise threshold and sensor 1 selection.

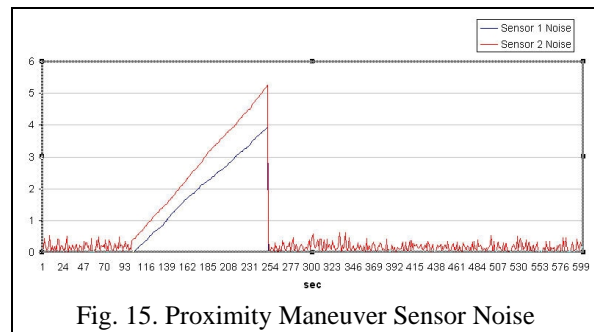


Fig. 15. Proximity Maneuver Sensor Noise

$T = 250$ when sensor 1 noise drops to 0, and sensor 2 noise remains Gaussian (Fig. 15).

The analysis shows the spacecraft slowing all actuators until the noise cleared up. The controller then chose the less noisy of the two sensors (sensor 1) for the remainder of the maneuver.

Fig. 16 shows the intended path in green. The red part of the ephemeris is during the period $T = 100$ to 250

This case demonstrated an ability to provide complex, non-orbital mechanical controls to the engine, based on human-like responses to changing conditions. As a complex system degrades temporarily or permanently, a human in command may adjust the plan to best achieve mission objectives; characteristics exhibited by this test. At every step of the simulation this type of controller takes into account system status in the calculation of engine commands to move forward, slow, hold or abort. The physics-based environment and 3-D display provides accurate understanding.

This model was very robust, responding well to reduced capability engines and required complete disabling of engines (failure = 100%) along the axis of travel to disrupt the maneuver significantly. Even then, the spacecraft was capable of reaching the assigned waypoints once the engine became less than 100% disabled. This model has a lot of potential for additional research, combining astrodynamics and operational command and control into intelligent controllers. These controllers are capable of maneuvering in close proximity, consistent with prudent operation given uncertainty and system degradation.

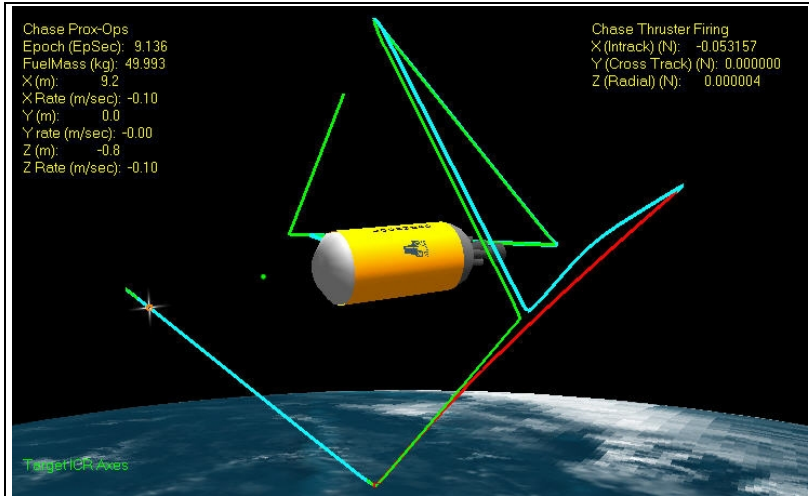


Fig. 16. Fuzzy Logic Controller Response to Sensor Noise

Proximity Operations Case 5 – Perch and Stare Maneuvering at Geostationary Orbit

As mentioned earlier, the current proximity maneuvering framework was constructed within an existing simulation to take advantage of other existing capabilities.

Using the previously discussed techniques described above, six waypoints were chosen around the Target geostationary spacecraft, one on each side of the Target. The waypoints included several perch and stare times (maintain position, image target) at each point, and halfway in between each waypoint. Fig. 17 shows the Chase’s trajectory relative to the Target. This is a single still frame from an animation produced in *STK*.

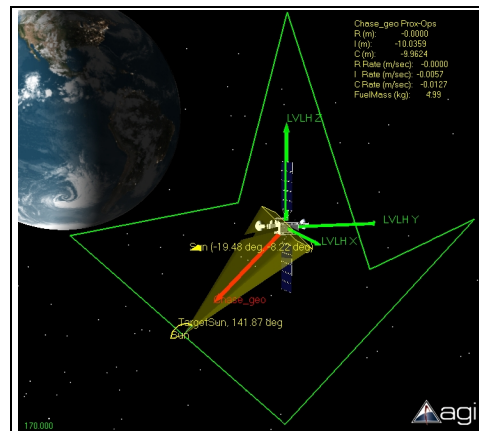


Fig. 17. Perch and Stare Maneuvering

To support situational awareness, several other objects have been added to this display, including the numeric data of the relative position and velocities, vectors showing the direction of the sun and relative positions, and the angle between the Chase’s camera bore sight and the sun. In addition, the field of view of the Chase’s camera is shown as a translucent square cone. These visual cues

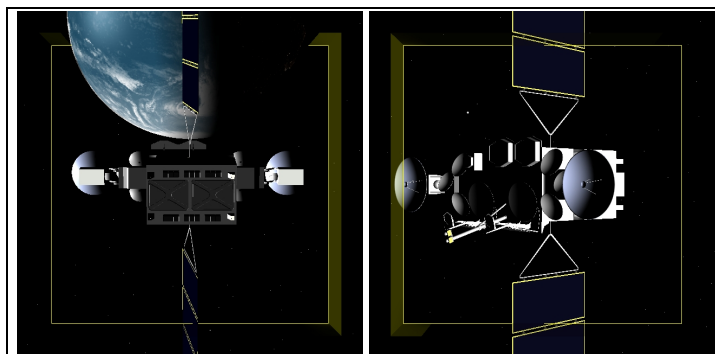


Fig. 18. Simulated Chase Camera Images, Different Sun Lighting

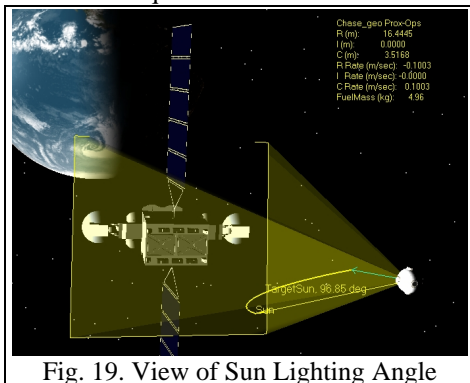


Fig. 19. View of Sun Lighting Angle

give invaluable insight to spacecraft controllers about the dynamic geometry. Without these pictures and animations, the controllers would only have telemetry numbers to look at and interpret. Realistic visualizations help assure that everyone has the same understanding of the situation, especially when dealing with complex and changing 3-D geometry.

Subsequent analyses are also possible. Fig. 18 shows simulated images from the Chase’s camera, using realistic geometry including

sun lighting and position; field of view; relative spacecraft position and attitude; and realistic Target solar array configuration. These images can help predict and understand what on-board sensors will and won't see, including sensor intrusion from the Earth, Sun, and Moon.

Fig. 19 shows a view of the Chase observing the Target when the sun to bore sight angle is just greater than 90 degrees. When planning an inspection mission, these angles are important [18], as well as how the angle changes in time. Fig. 20 shows the history of the sun angle, as well as the sun angle rate during the time span of the proximity operations.

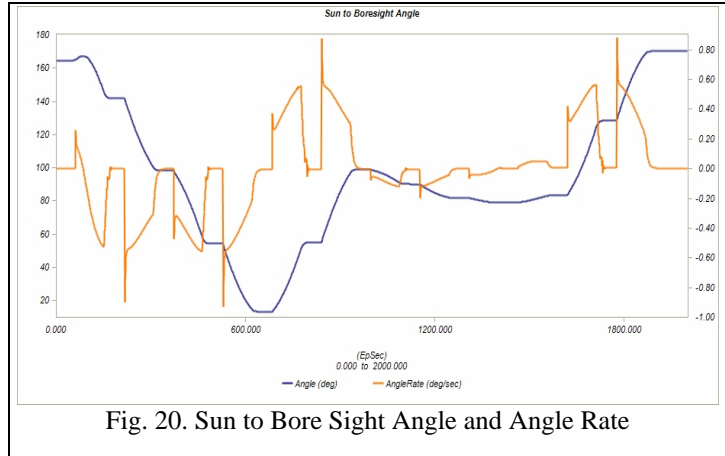


Fig. 20. Sun to Bore Sight Angle and Angle Rate

These data can be used for planning and interpreting camera performance. The perch times can be seen in the angle data as flat spots, and in the angle rate data when near zero.

Future Work

The work done in setting up *STK/Astrogator* with fuzzy logic for proximity maneuvering has been very promising. Future additions include attitude modeling and additional sensor and communications simulations. The intelligent controller was the most complex, but showed a great deal of promise in further refining and testing to include various non-orbital mechanical controls over a distant spacecraft. The *STK* environment will provide for such controls to have engines respond to signal to noise ratios, inter-object visibility, sun angles, and ground station links. Even variables such as crew rest can be programmed into the controller and analyzed in an accurate physical environment – this area offers the most promise and should be pursued with this preliminary work in setting up the environment as a baseline.

Conclusion

This paper presented an innovative application of an existing capability to design, simulate and analyze proximity maneuvers; currently in use for operational satellites performing other maneuvers. The benefits of this system include:

- A complete rendezvous to proximity operations (RPO) environment; model spacecraft missions from launch to end-of life.
- An accurate simulation and visualization to act as a decision aid; communicating the complexity, criticality, and risk of spacecraft proximity operations.
- An accurate trajectory model within an environment that already supports secondary mission requirements such as those concerning communications, sensor collection, interference, navigation, and power.
- A realistic physics-based simulation for the modeling, development and validation of control laws.
- A collaborative engineering environment for the design, development and tuning of spacecraft law parameters, sizing actuators (i.e., rocket engines) and sensor suite selection.
- A precise mathematical environment for research and development of future spacecraft maneuvering engineering tasks, operational planning and forensic analysis.
- A closed-loop, knowledge-based control example for proximity operations.

This RPO modeling and simulation environment provides a valuable adjunct to programs in space situational awareness and civil space exploration, engineering and decision making processes.

1. *Satellite Toolkit*, Analytical Graphics Inc., 220 Valley Creek Blvd., Exton, PA 19341, USA, Phone: 610-981-8000, E-mail: info@agi.com, www.agi.com

2. Overview of the DART Mishap Investigation Results, For Public Release, http://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf

-
3. A. S. Aljabri; T. Kia; J.Y. Lai, *Highly Autonomous Event-Driven Spacecraft Control*, Acta Astronautica vol. 35, 1995
 4. Oberg, James E., *Rendezvous And Proximity Operations Handbook*, JSC-10589, NASA Lyndon B. Johnson Space Center Mission Operations Directorate Flight Design and Dynamics Division, Houston, Texas, 1988
 5. John. Carrico, Emmet Fletcher, *Software Architecture and Use of Satellite Tool Kit's Astrogator Module for Libration Point Orbit Missions*, *Libration Point Orbits And Applications*, Parador d'Aiguablava, Girona, Spain, June 2002
 6. Mike Loucks, John Carrico, Timothy Carrico, Chuck Deiterich, *A Comparison Of Lunar Landing Trajectory Strategies Using Numerical Simulations*, International Lunar Conference, Toronto, Canada 2005
 7. John Carrico, *Spacecraft Rendezvous and Retrieval Missions Handbook for Flight Dynamics Analysts, Revision 1*. 554-FDD-91/042, Goddard Space Flight Center, Flight Dynamics Division, prepared by Computer Sciences Corporation, May, 1991
 8. James Herrstein, John Carrico, *Noncoplaner Rendezvous and the Biasing Technique*, CSC/TM-87/6013 89002, Goddard Space Flight Center, Flight Dynamics Division, prepared by Computer Sciences Corporation, June, 1989
 9. John Carrico, *Gamma Ray Observatory (GRO) Rendezvous Scenario Description*, CSC/TM-87/6013 89004, Goddard Space Flight Center, Flight Dynamics Division, prepared by Computer Sciences Corporation, September, 1989
 10. John Carrico, *Rendezvous and Phasing Computation Program (RENDEV) Requirements, System Description, and Mathematical Specification*, 554-FDD-90/195, Goddard Space Flight Center, Flight Dynamics Division, prepared by Computer Sciences Corporation, December 1990
 11. Jackie Kostoff, John Carrico, *RENDEV User's Guide*, 554-FDD-91/060, Goddard Space Flight Center, Flight Dynamics Division, prepared by Computer Sciences Corporation, March, 1991
 12. Darrel Conway, Robert Sperling, Dave Folta, Karen Richon, Robert DeFazio, *Automated Maneuver Planning Using a Fuzzy Logic Algorithm*, NASA/GSFC Flight Mechanics/Estimation Theory Symposium, Greenbelt, MD, USA, May, 1994
 13. Lea, R. N., *Automated Space Vehicle Control For Rendezvous Proximity Operations*. *Telemat. Inf.* 5, 3 (Nov. 1988), 179-185. DOI= [http://dx.doi.org/10.1016/S0736-5853\(88\)80022-4](http://dx.doi.org/10.1016/S0736-5853(88)80022-4)
 14. Lai, S. H. 1993. *Fuzzy Logic Controller Design for Spacecraft Proximity Operations*, *Comput. Ind. Eng.* 25, 1-4 (Sep. 1993), 13-16. DOI= [http://dx.doi.org/10.1016/0360-8352\(93\)90209-G](http://dx.doi.org/10.1016/0360-8352(93)90209-G)
 15. Suzuki, Takashi; Yasuda, Kinuharu; Yoshikawa, Shoji; Yamada, Katsuhiko; Yoshida, Norimasa, *An Application of Fuzzy Algorithm To Thruster Control System Of Spacecraft*, International Symposium on Space Technology and Science, 19th, Yokohama, Japan; 15-24 May 1994. pp. 303-310. 1994
 16. *FuzzyTech*, Inform GmbH, Pascalstrasse 23, D-52076 Aachen, Germany, Phone: 2408-945680, hotline@inform-ac.com, www.fuzzytech.com
 17. David Vallado, *Fundamentals of Astrodynamics and Applications*, 2nd. ed., Space Technology Library, Microcosm, Kluwer Academic Publishers, 2001
 18. Bo Naasz, *Safety Ellipse Motion with Coarse Sun Angle Optimization*, Flight Mechanics Symposium, Greenbelt, MD, USA, October, 2005