

METHODOLOGY FOR PROTOTYPING INCREASED  
LEVELS OF AUTOMATION  
FOR SPACECRAFT RENDEZVOUS FUNCTIONS

A Thesis

by

JEREMY JAY HART

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

May 2007

Major Subject: Aerospace Engineering

METHODOLOGY FOR PROTOTYPING INCREASED  
LEVELS OF AUTOMATION  
FOR SPACECRAFT RENDEZVOUS FUNCTIONS

A Thesis

by

JEREMY JAY HART

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee, John Valasek  
Committee Members, Thomas R. Ioerger  
Srinivas R. Vadali

Head of Department, Helen L. Reed

May 2007

Major Subject: Aerospace Engineering

## ABSTRACT

Methodology for Prototyping Increased Levels of Automation

for Spacecraft Rendezvous Functions. (May 2007)

Jeremy Jay Hart, B.S., University of Wisconsin - Madison

Chair of Advisory Committee: Dr. John Valasek

The Crew Exploration Vehicle (CEV) necessitates higher levels of automation than previous NASA vehicles due to program requirements for automation, including Automated Rendezvous and Docking (AR&D). Studies of spacecraft development often point to the locus of decision-making authority between humans and computers (i.e. automation) as a prime driver for cost, safety, and mission success. Therefore, a critical component in the CEV development is the determination of the correct level of automation. To identify the appropriate levels of automation and autonomy to design into a human space flight vehicle, NASA has created the Function-specific Level of Autonomy and Automation Tool (FLOAAT).

This research develops a methodology for prototyping increased levels of automation for spacecraft rendezvous functions. This methodology was used to evaluate the accuracy of the FLOAAT-specified levels of automation, via prototyping. Two spacecraft rendezvous planning tasks were selected and then prototyped in Matlab using Fuzzy Logic (FL) techniques and existing Shuttle rendezvous trajectory algorithms. The prototyped functions are the determination of the maximum allowable Time-of-Ignition (TIG) slip for a rendezvous phasing burn and the evaluation of vehicle position relative to Transition initiation (Ti) position constraints.

The methodology for prototyping rendezvous functions at higher levels of automation is judged to be a promising technique. The results of the prototype indicate that the FLOAAT recommended level of automation is reasonably accurate and that

FL can be effectively used to model human decision-making used in spacecraft rendezvous. FL has many desirable attributes for modeling human decision-making, which makes it an excellent candidate for additional spaceflight automation applications. These conclusions are described in detail as well as recommendations for future improvements to the FLOAAT method and prototyped rendezvous functions.

To my loving family. Thank you for all of your support over the years.

## ACKNOWLEDGMENTS

I would like to thank Dr. John Valasek for his encouragement and support as my graduate advisor. His confidence in me inspired the completion of my research and thesis despite the very busy schedule that I encountered upon returning to work at NASA. Thank you to my committee members, Dr. Srinivas Vadali and Dr. Thomas Ioerger.

I would like to express my thanks to NASA for allowing me to pursue my degree as part of the JSC fellowship program. I also appreciate the support of my management in this endeavor, specifically Mark Hammerschmidt, Howard Hu, Matt Ondler, and Dave Kanipe.

Several colleagues at NASA provided me with invaluable advice and guidance. In particular, I would like to thank Dustin Hamm and Brian Kubena for their assistance in implementing the Matlab version of the PISCES simulation. During the initial phases of my research several people were instrumental in helping me determine appropriate functions to prototype, including Jennifer Mendeck, Jack Brazzel, Ryan Proud, Chris D'Souza, and Mark Jackson.

I would also like to thank Theresa Spaeth for her much needed editorial and grammatical review. Finally, my acknowledgments cannot be complete without thanking Lesley Weitz for her endless patience during the thesis writing process, especially for her LaTeX and editing advice.

## TABLE OF CONTENTS

| CHAPTER |  | Page |
|---------|--|------|
| I       | INTRODUCTION . . . . .   | 1    |
|         | A. Automation in Human Spaceflight . . . . .   | 2    |
|         | B. Automated Rendezvous and Docking . . . . .  | 3    |
|         | C. Function-specific Level of Automation and Autonomy Tool                                       | 4    |
|         | D. Research Objectives and Approach . . . . .  | 7    |
| II      | TECHNICAL BACKGROUND . . . . .   | 9    |
|         | A. Introduction . . . . .  | 9    |
|         | B. NASA Flight Rules . . . . .   | 10   |
|         | 1. History of NASA Flight Rules . . . . .  | 10   |
|         | 2. Space Shuttle Orbiter and International Space Sta-<br>tion (ISS) Rendezvous Profile . . . . . | 11   |
|         | 3. Space Shuttle and ISS Rendezvous Flight Rules . . . . .                                       | 13   |
|         | 4. Crew Exploration Vehicle (CEV) Project and Flight<br>Rules . . . . .                          | 14   |
|         | C. Function-specific Level of Automation and Autonomy Tool                                       | 16   |
|         | 1. Levels of Automation (LOA) . . . . .  | 16   |
|         | 2. Rendezvous, Proximity Operations, and Docking<br>(RPOD) FLOAAT Study . . . . .                | 18   |
|         | a. Reference Levels of Autonomy and Automation . . . . .   | 19   |
|         | b. Resultant Requirements . . . . .  | 19   |
|         | D. Role of Artificial Intelligence (AI) in Modeling Human<br>Decision-Making . . . . .           | 20   |
| III     | CANDIDATE RENDEZVOUS FUNCTIONS FOR PROTO-<br>TYPING . . . . .                                    | 23   |
|         | A. Selection Criteria . . . . .  | 23   |
|         | B. Candidate Functions . . . . .   | 25   |
|         | C. Flight Rules . . . . .  | 27   |
|         | D. Functions Selected for Prototyping . . . . .  | 29   |
|         | 1. Time of Ignition (TIG) Slip Planning . . . . .  | 30   |
|         | 2. Transition Initiation (Ti) Position Constraints . . . . .                                     | 34   |

| CHAPTER | Page   |
|---------|--|
| IV      | CANDIDATE ARTIFICIAL INTELLIGENCE DECISION-<br>MAKING METHODS . . . . . 38 |
|         | A. Selection Criteria . . . . . 38   |
|         | B. Candidate Decision-Making Methods . . . . . 39                          |
|         | 1. Neural Networks . . . . . 39  |
|         | 2. Expert Systems . . . . . 42   |
|         | 3. Fuzzy Logic (FL) . . . . . 45   |
|         | 4. Other Methods . . . . . 47  |
|         | C. Selected Method, Fuzzy Logic . . . . . 47                               |
| V       | MODELING RENDEZVOUS FUNCTIONS USING FUZZY<br>LOGIC . . . . . 49            |
|         | A. Fuzzy Logic . . . . . 49  |
|         | 1. Fuzzy Sets . . . . . 49   |
|         | 2. Fuzzy Rules . . . . . 51  |
|         | 3. Fuzzy Logic Input/Output Mapping Example . . . . . 53                   |
|         | B. TIG-slip Planning . . . . . 58  |
|         | 1. TIG-slip Problem . . . . . 58   |
|         | 2. Modeling TIG-slip Planning Using Fuzzy Logic . . . . . 59               |
|         | a. Inertial TIG-slip Iteration Model . . . . . 59                          |
|         | b. LVLH TIG-slip Iteration Model . . . . . 64                              |
|         | c. TIG-slip Comparison Model . . . . . 66                                  |
|         | C. Evaluation of Vehicle Position at $T_i$ . . . . . 71                    |
|         | 1. $T_i$ Position Constraints . . . . . 73                                 |
|         | 2. Modeling $T_i$ Position Constraints Using Fuzzy Logic . . . . . 73      |
|         | a. Modeling the $3\text{-}\sigma$ Position Constraint . . . . . 74         |
|         | b. Modeling the MC-2 Lighting Constraint . . . . . 80                      |
|         | c. Modeling Navigation Accuracy . . . . . 85                               |
|         | d. Combined $T_i$ Position Constraint Model . . . . . 87                   |
| VI      | EXPERIMENT DESIGN . . . . . 92   |
|         | A. TIG-slip Planning . . . . . 92  |
|         | 1. Objectives . . . . . 92   |
|         | 2. Assumptions . . . . . 93  |
|         | 3. Simulation Environment . . . . . 94                                     |
|         | B. Evaluation of Vehicle Position at $T_i$ . . . . . 98                    |
|         | 1. Objectives . . . . . 98   |
|         | 2. Assumptions . . . . . 98  |



| CHAPTER  | Page |
|--|------|
| 3. Test Environment . . . . .                        | 99   |
| C. Hardware and Software Configuration . . . . .     | 100  |
| VII EXPERIMENTAL RESULTS . . . . .                   | 102  |
| A. TIG-slip Planning . . . . .                       | 102  |
| 1. Results, Nominal Trajectory . . . . .             | 102  |
| 2. Results, Dispersed Trajectories . . . . .         | 105  |
| 3. Comparison to Existing Methods . . . . .          | 108  |
| B. Evaluation of Vehicle Position at $T_i$ . . . . . | 110  |
| 1. Results . . . . .                                 | 110  |
| 2. Comparison to Existing Methods . . . . .          | 114  |
| C. Assessment of Level of Automation . . . . .       | 115  |
| D. Assessment of Fuzzy Logic . . . . .               | 117  |
| VIII CONCLUSIONS . . . . .                           | 119  |
| IX RECOMMENDATIONS . . . . .                         | 121  |
| REFERENCES . . . . .                                 | 124  |
| APPENDIX A . . . . .                                 | 129  |
| VITA . . . . .                                       | 131  |

## LIST OF TABLES

| TABLE |  | Page |
|-------|--|------|
| I     | Comparison of AI Decision-Making Methods . . . . .                   | 48   |
| II    | Rules for FL Season Based Temperature Model . . . . .                | 54   |
| III   | Rules for TIG-slip Iteration Models . . . . .                        | 64   |
| IV    | Rules for TIG-slip Comparison Model . . . . .                        | 70   |
| V     | Rules for Ti $3\text{-}\sigma$ Position Constraint . . . . .         | 78   |
| VI    | Rules for MC-2 Lighting Constraint . . . . .                         | 84   |
| VII   | Rules for Ti Position Constraint Model, Navigation = Precise . . . . | 89   |
| VIII  | Rules for Ti Position-Constraint Model, Navigation = Degraded . . .  | 90   |
| IX    | Rules for Ti Position-Constraint Model, Navigation = Inadequate . .  | 91   |
| X     | TIG-slip Planning Success Criteria . . . . .                         | 93   |
| XI    | Hardware and Software Configuration for Prototype Experiments . .    | 101  |

## LIST OF FIGURES

| FIGURE |  | Page |
|--------|--|------|
| 1      | FLOAAT Level of Autonomy Scales, v4.0 . . . . .            | 5    |
| 2      | FLOAAT Level of Automation Scales, v4.0 . . . . .          | 6    |
| 3      | Stable Orbit Rendezvous . . . . .                          | 12   |
| 4      | Inertial and LVLH Reference Frames . . . . .               | 13   |
| 5      | Shuttle Flight Rule Example . . . . .                      | 15   |
| 6      | Reference Levels of Automation and Autonomy . . . . .      | 21   |
| 7      | Requirement Based on FLOAAT Output . . . . .               | 22   |
| 8      | Candidate High-Level Rendezvous Functions . . . . .        | 26   |
| 9      | Inertial TIG slip . . . . .                                | 32   |
| 10     | LVLH TIG slip . . . . .                                    | 33   |
| 11     | Ti Position and Lighting Constraints . . . . .             | 35   |
| 12     | Ti Delay Trajectory . . . . .                              | 37   |
| 13     | Simplified Artificial Neuron . . . . .                     | 40   |
| 14     | Simplified Neural Network . . . . .                        | 41   |
| 15     | Crisp Set $C = [a,b]$ . . . . .                            | 50   |
| 16     | Membership Function for Fuzzy Set $F = [a,b]$ . . . . .    | 51   |
| 17     | Intersection Operator as Defined for Fuzzy Sets . . . . .  | 52   |
| 18     | Union Operator as Defined for Fuzzy Sets . . . . .         | 52   |
| 19     | FL Example, Membership Functions for the Seasons . . . . . | 54   |

| FIGURE | Page  |
|--------|---|
| 20     | FL Example, Membership Functions for Temperature . . . . . 55               |
| 21     | FL Example, Seasons Model Output for April . . . . . 56                     |
| 22     | FL Example, Seasons Model Output for July . . . . . 57                      |
| 23     | FL Example, Seasons Model Input-Output Summary . . . . . 58                 |
| 24     | Inertial TIG slip of 30 seconds, Position-Offset . . . . . 60               |
| 25     | Inertial TIG slip of 180 seconds, Position-Offset . . . . . 61              |
| 26     | Inertial TIG-slip Iteration Model, Input Membership Functions . . . 62      |
| 27     | Inertial TIG-slip Iteration Model, Output Membership Functions . . 63       |
| 28     | Inertial TIG-slip Iteration Model, Input-Output Mapping . . . . . 65        |
| 29     | LVLH TIG-slip Iteration Model, Output Membership Functions . . . 66         |
| 30     | LVLH TIG-slip Iteration Model, Input-Output Mapping . . . . . 67            |
| 31     | TIG-slip Comparison, Propellant Input Membership Functions. . . . 68        |
| 32     | TIG-slip Comparison, Time Input Membership Functions. . . . . 69            |
| 33     | TIG-slip Comparison, Propellant Output Membership Functions. . . 70         |
| 34     | TIG-slip Comparison Model, Input-Output Mapping . . . . . 72                |
| 35     | TIG-slip Comparison Model, Input-Output Mapping (Contour Plot) 72           |
| 36     | Ti $3\text{-}\sigma$ Position Constraint . . . . . 75                       |
| 37     | Assessment Number for an Elliptical Constraint . . . . . 75                 |
| 38     | Ellipse-Assessment Input Membership Functions . . . . . 76                  |
| 39     | Ti Position-Constraint Model, Output Membership Functions . . . . 77        |
| 40     | Ti $3\text{-}\sigma$ Position Constraint, Input-Output Mapping . . . . . 78 |
| 41     | Severity Output Surface for an Example Elliptical Constraint . . . . 79     |

| FIGURE | Page   |
|--------|--|
| 42     | Ti Position and Lighting Constraints . . . . . 80  |
| 43     | Determine MC-2 Lighting Constraint Boundary . . . . . 82   |
| 44     | MC-2 Lighting Constraint, MC2-Assessment Surface Plot . . . . . 82   |
| 45     | MC2-Assessment Input Membership Functions . . . . . 83   |
| 46     | Ti MC-2 Lighting Constraint, Input-Output Mapping . . . . . 85   |
| 47     | MC-2 Lighting Constraint, Input-Output Surface . . . . . 86  |
| 48     | Navigation-Quality Input Membership Functions . . . . . 87   |
| 49     | Ti Position Constraint Model, Input-Output Mapping, Navigation-<br>Quality = 5.0 ('precise') . . . . . 89  |
| 50     | Ti Position-Constraint Model, Input-Output Mapping, Navigation-<br>Quality = 2.5 ('degraded') . . . . . 91 |
| 51     | TIG-slip Trajectory Example, Nominal and TIG-slip cases . . . . . 95                                       |
| 52     | TIG-slip Planning Flow-Chart . . . . . 97  |
| 53     | Compute Inertial TIG-slip, Flow-Chart . . . . . 97   |
| 54     | Ti Position Constraint FL Model, Flow-Chart . . . . . 100  |
| 55     | Inertial TIG-slip Results, Nominal Trajectory . . . . . 103  |
| 56     | LVLH TIG-slip Results, Nominal Trajectory . . . . . 104  |
| 57     | Inertial and LVLH Maximum TIG-slip Comparison (Nominal Run) . 104  |
| 58     | Dispersed Initial Conditions for NC TIG slip . . . . . 106   |
| 59     | Inertial TIG slip for Dispersed Initial Conditions . . . . . 107   |
| 60     | LVLH TIG slip for Dispersed Initial Conditions . . . . . 108   |
| 61     | FL Ti Constraint Model Output (Surface Plot), Nav Accuracy = 5.0 111                                       |
| 62     | FL Ti Constraint Model Output, Nav Accuracy = 5.0 . . . . . 112  |

| FIGURE | Page  |
|--------|---|
| 63     | FL Ti Constraint Model Output, Nav Accuracy = 4.0 . . . . . 113 |
| 64     | FL Ti Constraint Model Output, Nav Accuracy = 2.5 . . . . . 113 |
| 65     | FL Ti Constraint Model Output, Nav Accuracy = 1.0 . . . . . 114 |

## CHAPTER I

## INTRODUCTION

The National Aeronautics and Space Administration (NASA) recently established a new vision for space exploration that calls for the design of the next generation of spacecraft to explore the solar system[1]. The new spacecraft, called the Crew Exploration Vehicle (CEV), will be capable of rendezvousing with the International Space Station (ISS), returning to the moon, and eventually enabling human exploration of Mars. These missions present unique challenges such as increased communication delays and spacecraft rendezvous in lunar and Martian orbits. To meet these challenges there must be an increased level of vehicle autonomy<sup>1</sup> over previous human spacecraft[2]. Because of limited crew sizes many of the increases in autonomy will be realized by the use of on-board automation<sup>2</sup>. As a result, the CEV necessitates higher levels of automation than previous NASA vehicles. A key technology to the success of the CEV is developing Automated Rendezvous and Docking (AR&D)[3].

The precise breakdown of responsibility between the crew and on-board computers, or level of automation, has not been formally established for the CEV. One critical area is in the division of authority for decision-making tasks. Studies of spacecraft development often point to the locus of decision-making authority between humans and computers (i.e. automation) as a prime driver for cost, safety, and mission success[4].

---

The journal model is *IEEE Transactions on Automatic Control*.

<sup>1</sup>Autonomy is defined as the ability for a vehicle and its on-board systems to perform a function without external support. On-board systems include humans that are on-board. The level of autonomy is the degree to which the function can be performed by on-board systems without ground systems support.

<sup>2</sup>Automation is defined as the ability for computer systems to perform a function without human support. The level of automation is the degree to which the function can be performed by computer systems without human support.

Therefore, a critical component in the CEV development is the determination of the appropriate level of automation.

#### A. Automation in Human Spaceflight

Historically, NASA has operated at low levels of automation and relied heavily on manual control and ground based planning. In early spacecraft such as Mercury, Gemini, and Apollo, computer technology limited the amount of automation. However, some routine and repetitive tasks were performed automatically. In some cases the automated functions were inhibited by the ground or crew due to a lack of trust in the automation.

The Space Shuttle has a variety of automated functions for both ground and on-board systems. There are automated responses for many single systems-failure cases, requiring limited human interaction, but multiple failure cases are not automated. During the design of the Shuttle there were plans for on-board automation of numerous functions, but many of these plans were eliminated because of cost and schedule pressures. Since the first launch of the Shuttle, many functions have been automated with mixed results. Overall, the Shuttle relies heavily on humans for execution of virtually all of its on-board and ground-controlled functionality.

The ISS was intended to have increased levels of automation for many major functions in order to meet the needs of continuous operations. Many of the space station's subsystems include automated functionality to maintain and conduct nominal operations. However, much of the automated functionality is difficult and costly to modify. The result is that many functions are disabled or bypassed via operational workarounds. There have been some recent improvements to ISS automated operations such as the inclusion of the Timeliner software used for command and control



functions[1].

For the CEV, new approaches must be used to determine the correct levels of automation. One particular area is the automation of rendezvous and docking functions.

## B. Automated Rendezvous and Docking

Rendezvous of spacecraft in orbit has been a critical task throughout the history of spaceflight. It was identified as a necessary activity early on in the development of the United States space program and was the primary technical objective of the Gemini missions[5]. In the Apollo program, the Lunar Module (LM) had to successfully rendezvous with the Command and Service Module (CSM) on its return from the lunar surface. Rendezvous also allows for on-orbit assembly, which provides flexibility in mission design by eliminating the requirement for one large booster rocket to carry every spacecraft component in a single launch.

Early Space Shuttle design studies included high levels of autonomy and automation for rendezvous capabilities due to rendezvous experience gained during Apollo and significant advancements in on-board computer capabilities. As late as 1976 there existed requirements for nominal rendezvous planning to occur using on-board computers with little or no support from Mission Control [5]. However, budget and schedule issues limited the on-board computer capability, which made these requirements difficult to meet. It was decided to reduce on-board targeting to include only burns supported by on-board relative navigation sensors. The automation of Shuttle rendezvous tasks was further complicated by a wide variety of missions. Early rendezvous missions were deploy/retrieval of satellites, missions with multiple rendezvous, and retrieval or servicing of un-cooperative target satellites.

As the role of the Shuttle changed to primarily rendezvous with the ISS, procedures became more standardized. This allowed for automated planning capability of proximity operations (prox ops) to be developed such as the Rendezvous and Prox Ops Program (RPOP) tool. The RPOP tool is hosted on a laptop computer and used to provide the crew a relative motion display and piloting cues. There have also been increases in the automation of ground-based tools used in the Mission Control Center (MCC). However, much of the planning and execution of Shuttle rendezvous and prox ops remain at low levels of automation[6]. The lowest levels of automation are for decision-making functions, which can be the most challenging to automate.

All of the CEV missions will require successful rendezvous, and the CEV requirements call for automated rendezvous and docking[7]. The requirements include uncrewed docking to the ISS, safe return without communication with the ground, and operation of the CEV with only a single crew member. These requirements result in a significant amount of on-board automation for rendezvous and docking functions. Since the existing levels of automation for these functions is low, this is a risk area for CEV development. In particular, the automation of decision-making functions will be critical to the success of automated rendezvous and docking for the CEV.

### C. Function-specific Level of Automation and Autonomy Tool

By finding the correct levels of automation, NASA can vastly improve the probability of mission success, increase safety, and decrease overall cost. To identify the appropriate levels of automation and autonomy to design into a human space flight vehicle, NASA has created a method called the Function-specific Level of Autonomy and Automation Tool (FLOAAT)[8, 4].

The backbone of FLOAAT is a practical construct of separate levels of automa-

tion and autonomy for each of the 4 stages of decision-making (Observe, Orient, Decide, and Act)[9], which leverages off theoretical constructs [8, 4]. These Levels of Autonomy and Automation (LOAAs) are divided into a 5-point scale for autonomy as shown in Figure 1, with level 1 corresponding to complete ground authority and level 5 corresponding to complete on-board authority, and an 8-point scale for automation as shown in Figure 2, with level 1 corresponding to complete human authority and level 8 corresponding to complete computer authority. The FLOAAT process employs a survey in which domain-area experts evaluate a variety of issues that would each lead to more or less autonomy or automation for a particular function (or task). These results are then mapped onto the corresponding LOAA scales. The output of FLOAAT is a level of automation and autonomy for each function (or task) evaluated in the process.

| Level | Observe  | Orient  | Decide  | Act  |
|-------|--|---|---|--|
| 5     | The data is monitored onboard without assistance on the ground.          | The calculations are performed onboard without assistance on the ground.        | The decision is made onboard without assistance on the ground.            | The task is executed onboard without assistance on the ground.               |
| 4     | The data is monitored onboard with available assistance on the ground.   | The calculations are performed onboard with available assistance on the ground. | The decision is made onboard with available assistance on the ground.     | The task is executed onboard with available assistance on the ground.        |
| 3     | Both the ground and the onboard have the capability to monitor the data. | Both the ground and the onboard have the capability to perform calculations.    | Both the ground and the onboard have the capability to make the decision. | Both the ground and the onboard have the capability to execute the decision. |
| 2     | The data is monitored on the ground with available assistance onboard.   | The calculations are performed on the ground with available assistance onboard. | The decision is made on the ground with available assistance onboard.     | The task is executed on the ground with available assistance onboard.        |
| 1     | The data is monitored on the ground without assistance onboard.          | The calculations are performed on the ground without assistance onboard.        | The decision is made on the ground without assistance onboard.            | The task is executed on the ground without assistance onboard.               |

\* "Without assistance" does not preclude data access

Fig. 1. FLOAAT Level of Autonomy Scales, v4.0

| Level | Observe   | Orient   | Decide   | Act   |
|-------|---|--|--|---|
| 8     | The computer is responsible for gathering and filtering data without displaying any information to the human.   | The computer overlays predictions with analysis and interprets data for a result that is not displayed to the human.   | The computer performs the final ranking task, and does not display the result to the human.  | The computer executes the decision and does not allow any human interaction.  |
| 7     | The computer is responsible for gathering and filtering data without displaying any information to the human. Though, a "program status indicator" is displayed.  | The computer overlays predictions with analysis and interprets data for a result which is only displayed to the human if result fits programmed context (context dependant summaries). | The computer performs the final ranking task and displays a reduced set of ranked options without displaying "why" the decision was made to the human. | The computer executes the decision and only informs the human if required by context. The human is given override ability after execution when physically possible.   |
| 6     | The computer is responsible for gathering, filtering, and prioritizing information displayed to the human.  | The computer overlays predictions with analysis and interprets the data. The human is shown all results for potential override.  | The computer performs the ranking task and displays a reduced set of ranked options while displaying "why" the decision was made to the human.         | The computer executes the decision, informs the human, and allows for override ability after execution when physically possible. In the event of a contingency, the human can independently execute the decision. |
| 5     | The computer is responsible for gathering and displaying unprioritized information for the human. The computer filters out the unhighlighted data for the human to monitor.                               | The computer overlays predictions with analysis and interprets the data. The human is the backup for interpreting data.  | The computer performs the ranking task. All results, including "why" the decision was made, are displayed to the human.                                | The computer allows the human a context-dependant time-to-veto before executing the decision. In the event of a contingency, the human can independently execute the decision.                                    |
| 4     | The computer is responsible for gathering and displaying unfiltered, unprioritized information for the human. The computer highlights the relevant non-prioritized information for the human to monitor.  | The computer is the prime source for analyzing data and making predictions as a trusted calculator. The human is the prime source for interpreting data.                               | Both the human and the computer perform the ranking task, the results from the computer are considered prime.  | The computer allows the human a pre-programmed time-to-veto before executing the decision. In the event of a contingency, the human can independently execute the decision.                                       |
| 3     | The computer is responsible for gathering and displaying unfiltered, unhighlighted, and unprioritized information for the human. The human is the prime monitor for all information with computer backup. | The computer is the prime source for analyzing data and making predictions with human checks of the calculations. The human is the only source for interpreting data.                  | Both the human and the computer perform the ranking task, the results from the human are considered prime.   | The computer executes the decision after human grants authority-to-proceed. In the event of a contingency, the human can independently execute the decision.  |
| 2     | The human is the prime source for gathering and monitoring data, with computer backup.  | The human is the prime source for analyzing data and making predictions, with computer verification when needed. The human is the only source for interpreting data.                   | The human is the only source for performing the ranking task, but the computer can be used as a tool for assistance.                                   | The human is the prime source for executing the decision, with computer backup for contingencies (e.g. deconditioned humans).   |
| 1     | The human is the only source for gathering and monitoring (defined as filtering, prioritizing and understanding) data.  | The human is the only source for analyzing data, making predictions, and interpreting data.  | The human is the only source for performing the ranking task.  | The human is the only source for executing the decision.  |

\*Humans still have access to data at the highest Levels of Automation, but it is not displayed by default

Fig. 2. FLOAAT Level of Automation Scales, v4.0

#### D. Research Objectives and Approach

This research seeks to prototype a sub-set of the rendezvous and/or prox ops functions at the levels of automation specified using FLOAAT. By prototyping at these levels, the accuracy of the FLOAAT outputs can be evaluated. Modern decision-making algorithms will be used to help improve the efficiency, safety, and quality in the execution of selected rendezvous and prox ops planning tasks. This research only addresses the division of human versus computer responsibility (automation) and will not address the issue of ground versus on-board responsibility (autonomy). The issue of autonomy, although important, is difficult to prototype until a more detailed design of the ground-control architecture and on-board computing and display capabilities is available.

The research objectives are to:

1. Prototype selected rendezvous and/or prox ops functions at the levels of automation determined by the Function-specific Level of Autonomy and Automation Tool (FLOAAT) process.
2. Evaluate the prototype versions by comparing to Shuttle/ISS implementations of the same functions.
3. Use this comparison to evaluate the accuracy of the FLOAAT recommended level of automation (LOA).
4. Evaluate the selected decision-making algorithms as applied to the selected functions

A final evaluation will be made to determine if the level of automation was appropriate for each prototyped function and provide suggestions for improvement.

This includes an evaluation of the prototyping process, decision-making techniques used, and the effectiveness of operating at the levels of automation specified by the FLOAAT process. The results of the prototyping effort will be used to gauge the accuracy of the FLOAAT tool to select appropriate levels of automation. It will also determine the applicability of the selected decision-making algorithms for use in human spaceflight.

The methodology to prototype spacecraft rendezvous functions at increased levels of automation is described in this section. The first step is to select appropriate rendezvous functions to prototype at the FLOAAT specified LOAs. The objectives and constraints of candidate functions are captured in the flight rules and procedures used for Shuttle/ISS rendezvous and docking. Once a set of rendezvous functions is selected, a survey of available Artificial Intelligence (AI) decision-making techniques is conducted to determine which technique is the most suitable for prototyping the selected rendezvous functions. Then, a prototype is created to implement the selected rendezvous functions at the appropriate LOAs as specified by the FLOAAT process. The effectiveness of the prototype for nominal and off-nominal test cases is compared to current methods used in Shuttle/ISS rendezvous. This evaluation includes an assessment of the selected AI technique and the FLOAAT selected LOAs. The results and conclusions of this research are presented including recommendations for future work.

## CHAPTER II

### TECHNICAL BACKGROUND

#### A. Introduction

During a given NASA mission, numerous decisions are made that affect the success of the mission and the safety of the crew. To date, the vast majority of this decision-making is completed on the ground by flight controllers using operational guidelines and constraints captured in ‘flight rules’. Future NASA missions to the moon and Mars will require increased use of on-board computer-based decision-making because of communication delays with the ground and limited crew sizes [2]. Recent advancements in computing speed and the development of reliable computer-based decision-making methods can be used to meet this requirement. NASA developed the FLOAAT process to determine the correct levels of automation and autonomy for human spacecraft functions. This process was used to determine the appropriate levels of automation and autonomy for CEV rendezvous and prox ops in an early development effort [10].

This section will discuss the history of NASA flight rules and give examples of existing flight rules for the Space Shuttle and ISS rendezvous decision-making. A discussion of the CEV mission architecture and expected flight rules for rendezvous functions are discussed. The FLOAAT process is outlined and the results of the FLOAAT study to determine the LOAAs for CEV rendezvous functions is presented. Finally, the role of AI algorithms for human decision-making will be discussed with a focus on previous spaceflight applications and the potential use in future human spaceflight decision-making tasks.

## B. NASA Flight Rules

### 1. History of NASA Flight Rules

Throughout the history of NASA, decisions have been made by humans on the ground during the mission. These decisions are critical to the success of the mission and the safety of the astronauts. Most of the necessary decisions have been determined long before launch via analysis and simulation cases, and these decisions are captured in procedures and ‘flight rules’. As spacecraft and missions have become more complicated, the number of flight rules has grown. During the Apollo era they were called ‘flight mission rules’. The following definition is from the final version of the Apollo 17 flight mission rules document[11]:

‘MISSION RULES ARE PROCEDURAL STATEMENTS WHICH PROVIDE FLIGHT CONTROL PERSONNEL WITH GUIDELINES TO EXPEDITE THE DECISION-MAKING PROCESS. THE RULES ARE BASED ON AN ANALYSIS OF MISSION EQUIPMENT CONFIGURATION, SYSTEMS OPERATIONS AND CONSTRAINTS, FLIGHT CREW PROCEDURES, AND MISSION OBJECTIVES.’

These flight rules are the guidelines that flight controllers use in the MCC to operate the vehicle. In most cases, the flight rules are followed precisely and therefore provide a model for how the spacecraft is intended to be operated. In addition to nominal operations, flight rules capture many of the procedures for handling off-nominal performance and system failure cases. The rules are generated and rigorously tested via extensive human-in-the-loop simulation. The flight rules allow the flight controllers to have at hand a trusted guide for how to respond in the high-stress and time-critical operations of human spaceflight.



## 2. Space Shuttle Orbiter and International Space Station (ISS) Rendezvous Profile

This section discusses the Space Shuttle rendezvous profile for background necessary to understand Shuttle rendezvous flight rules. The baseline Shuttle rendezvous profile is known as ‘stable orbit rendezvous’. This trajectory profile has been used in the Shuttle program since 1983 for rendezvous with satellites, the Mir Space Station, and the ISS [5]. Figure 3 shows the relative motion of the Space Shuttle Orbiter with respect to the target spacecraft in a Local Vertical Local Horizontal (LVLH) reference frame. The origin of the LVLH frame is the target vehicle with the V-bar indicating the direction of the target spacecraft orbit and the R-bar directed toward the center of the body being orbited (i.e. Earth), this is shown in Figure 4. Several orbital burns are conducted by the Orbiter during the rendezvous profile to change the relative motion of the two spacecraft. The burns are computed as velocity changes ( $\Delta V$ 's) that the Orbiter executes using Orbiter Maneuvering System (OMS) thrusters and/or Reaction Control System (RCS) thrusters. The burns are executed in order from right to left and indicated by black squares with labels denoting the type of burn executed. The burn sequence and associated Shuttle nomenclature is as follows [6]:

- Nth Central phasing burn (NC). The NC burn allows the Orbiter to catch up with the target at the proper rate.
- Nth Corrective Combination burn (NCC). NCC targets the desired downtrack, out of plane position, and height at a future point (e.g. Ti).
- Transition initiation burn (Ti). This burn targets the Orbiter for a near-intercept trajectory with respect to the target spacecraft.

- Midcourse Correction (MC1 - MC4). The MC burns are small correction burns executed between  $T_i$  and the manual prox ops phase.

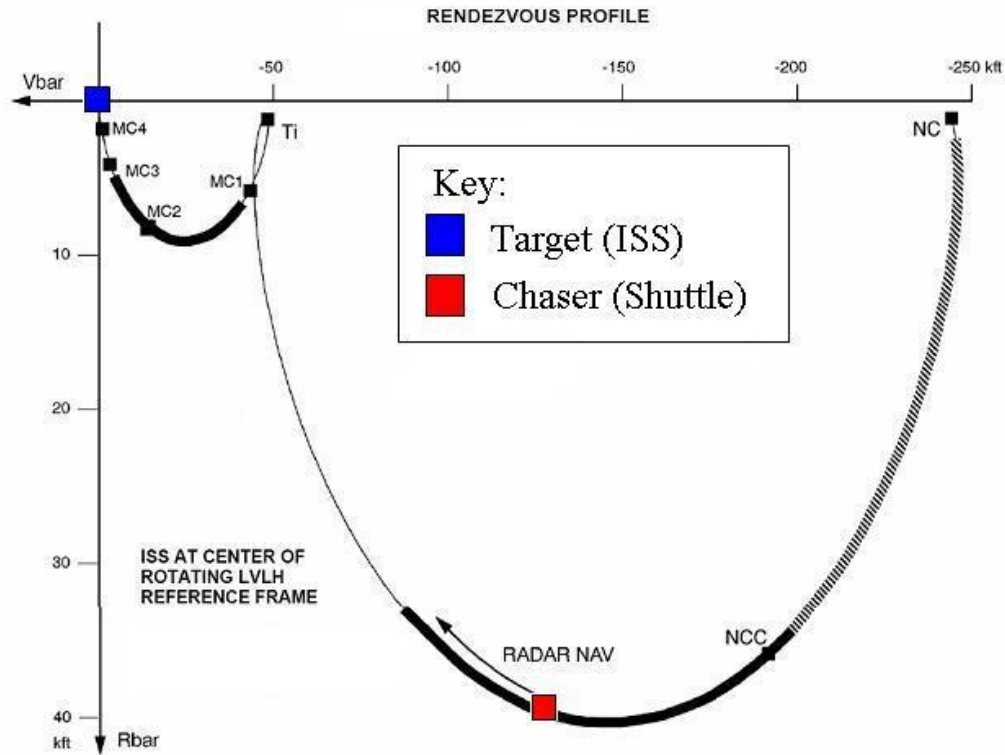


Fig. 3. Stable Orbit Rendezvous

This research focuses on the near-field rendezvous phase of the Rendezvous, Proximity Operations, and Docking (RPOD) operations. The rendezvous phase occurs after insertion into orbit following launch and concludes at the prox ops phase. Figure 3 shows the burns that comprise the near-field portion of the rendezvous profile, which occurs in the hours just prior to docking. A large portion of the RPOD decision-making occurs during the near-field rendezvous portion of the flight. This figure does not show the prox ops phase of docking. The prox ops phase occurs when the chaser vehicle (Shuttle or CEV) is in close proximity with the target vehicle (ISS). This phase begins when the range is less than 1000 ft and LVLH relative velocity is less

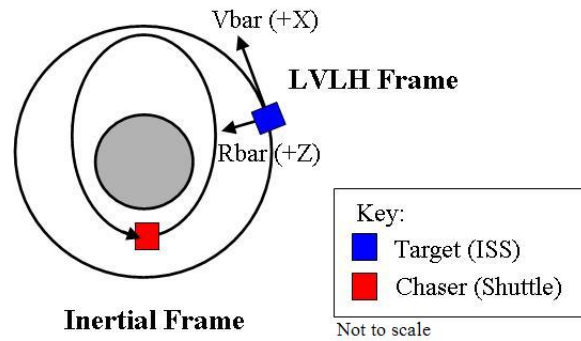


Fig. 4. Inertial and LVLH Reference Frames

than 1 ft/sec in each axis.

During prox ops, different techniques are used to control the orbiter trajectory than those used during rendezvous operations [10]. These techniques rely on crew visual observations and piloting techniques to achieve a desired relative state. Therefore, Prox Ops operations are primarily a guidance task which does not include the type of decision-making this research seeks to automate. The focus of this research will be the decision-making functions performed during near-field rendezvous.

### 3. Space Shuttle and ISS Rendezvous Flight Rules

There are currently nine flight rules for Shuttle rendezvous [12] and numerous operational procedures documented in the Flight Dynamics Officer On-Orbit Handbook [6]. A summary of the flight rules is listed here:

- Definitions of rendezvous and prox ops, A2-116
- Systems required for executing the Ti burn, A2-117
- Priority (preference) of rendezvous sensors, A2-119
- Propulsion limits and burn recommendations, A2-121

- Source of burn solutions (ground or on-board), A2-122
- Rendezvous maneuver solution selection criteria, A2-123
- Maneuver trim and time of execution requirements, A2-124
- Rules for delay of Ti burn, A2-125
- Plan for executing a breakout (abort) maneuver, if necessary, A2-126

These flight rules and associated procedures will be used to define the objectives and constraints of the rendezvous functions selected for the prototype. The example flight rule shown in Figure 5 provides the details of how a rendezvous delay should be executed [12]. Each flight rule includes rationale and additional details.

#### 4. Crew Exploration Vehicle (CEV) Project and Flight Rules

The CEV will be capable of rendezvous with the ISS, returning to the moon, and eventually enabling human exploration of Mars. There are two competing designs for the CEV rendezvous profile: 1) the ‘stable orbit rendezvous’ used in the Shuttle Program and 2) the ‘coelliptic profile’ used for lunar orbit rendezvous of the LM and CSM during the Apollo program[5]. The process is underway within NASA to select the rendezvous profile for the CEV[13]. Despite the differences in the profile, many of the CEV flight rules will be similar to the Shuttle flight rules. The specific details will change, but large decisions such as when to initiate the burn to create the near-intercept trajectory (Ti burn) will be based on similar criteria. Therefore, prototyping based on the Shuttle flight rules will be useful for automation of CEV rendezvous functions.

**A2-125****RNDZ OPS DELAY**

- A. FOR DELAYS PRIOR TO TI, IF CONVENIENT, THE NOMINAL MANEUVERS WILL BE ADJUSTED TO SLOW PHASING SO THAT THE ORBITER ARRIVES AT THE TI POINT AT THE DESIRED DELAY TIME. OTHERWISE, THE NOMINAL RENDEZVOUS PLAN WILL BE EXECUTED UNTIL TI WHERE A DELAY WILL BE DONE.

*The most efficient way, from a propellant standpoint, to delay a rendezvous is to slow the catch-up rate such that the orbiter will rendezvous with the target at the desired time, in an integral number of orbits. Desirable lighting conditions will be maintained since there is an integral number of orbits. This can be done at any preplanned phasing maneuver (NC), typically with no propellant penalty, or an additional maneuver can be inserted into the rendezvous profile. Mission activities and propellant budgets will determine which option to use.*

*In order to efficiently perform a phasing adjustment, an apogee point must be available prior to TI where the perigee can be raised and the catchup rates slowed. Any point other than apogee will necessitate a larger burn and fuel consumption. Communication availability and time to adjust the maneuver must also be considered. If any problems in these areas arise, it is no longer convenient to adjust phasing, and a TI delay should therefore be executed.*

- B. FOR SHORT TI DELAYS (ONE OR TWO ORBITS), MANEUVERS WILL BE EXECUTED TO RETURN TO THE TI POINT WITH THE DESIRED LIGHTING CONDITIONS. FOR LONG TI DELAYS (GREATER THAN TWO ORBITS), MANEUVERS WILL BE EXECUTED TO GUARANTEE A SAFE OPENING RATE OF AT LEAST 1 NM PER ORBIT.

*TI is the last convenient point to delay the rendezvous. This delay may be desirable to recover lost capability prior to committing to intercept. For short delays, the ground will compute a phasing maneuver to return to the TI position based on the most current onboard relative state. Desirable lighting conditions will be maintained since there is an integral number of orbits between the original and new TI maneuvers.*

*A long delay requires an opening rate to eliminate the need for the crew to continuously monitor the target. Long delays usually involve at least one crew sleep cycle. In this case, if propellant is available, the delay maneuver is targeted to a point in the following morning equivalent to where the original final phasing maneuver was executed. This range minimizes the possibility of awakening the crew to execute a trajectory control maneuver; shorter ranges may require interruption of crew sleep. In this manner, the second rendezvous trajectory will be similar to the original trajectory.*

...

Fig. 5. Shuttle Flight Rule Example

### C. Function-specific Level of Automation and Autonomy Tool

FLOAAT was developed to determine the appropriate Levels of Autonomy and Automation (LOAAs) for functions (tasks) performed in human spaceflight applications. It is a survey-based tool that requires respondents to address issues associated with automation and autonomy. The results of the process are recommended LOAA numbers for each function evaluated that correspond to the FLOAAT Level of Autonomy and Automation (LOAA) Scales (shown in Figures 1 and 2). The FLOAAT process has been proposed for the CEV program as a standard method to determine the appropriate balance of decision-making authority for human versus computers (automation) and ground versus on-board (autonomy)[8, 4].

In the FLOAAT process, domain-specific experts qualitatively evaluate a set of functionally decomposed vehicle requirements in a manner that yields a quantitative solution similar to the Cooper-Harper Scale [14] or the Bedford Workload Scale [15]. This quantitative solution maps directly to the FLOAAT LOAA Scales. The LOAA scales provide an easy-to-understand definition for the different levels of autonomy and automation. Application of the FLOAAT Process will result in clear and concise requirements that specify decision-making authority. This research focuses on prototyping the FLOAAT specified levels of automation for near-field rendezvous functions.

#### 1. Levels of Automation (LOA)

The FLOAAT LOA Scales provide precise and clear delineations of the different levels of automation. The intent of the scales is to aid system designers in easily and correctly identifying the appropriate level of automation to design into each function within the system. The LOA scales are available for either identifying the level of

automation of an existing function or for proposing an appropriate level of automation during the design of a new system.

The LOA Scales were derived from a mix of research into external automation applications (Sheridan [16], Parasuraman[17]) and the NASA internal automation scale development[8]. Each automation scale is tailored to fit the tasks encompassed by that function type (Observe, Orient, Decide, or Act (OODA)). This functional breakdown is designed to match the OODA loop[9]. The OODA loop was conceived by Col. John R. Boyd in the 1970s as an air-to-air combat strategy for military fighter pilots. Essentially, the OODA loop provides a model for how humans make tactical decisions. This concept lends itself well to the design of an autonomous and automated system, as its purpose is to mimic decisions made by human flight controller, crew, and computer teams today. The four categories in an OODA loop are:

- ‘Observe’, refers to gathering, monitoring, and filtering data;
- ‘Orient’, refers to deriving a list of options through analysis, trend prediction, interpretation and integration;
- ‘Decide’, refers to decision-making based on ranking available options; and
- ‘Act’, refers to execution or authority to act on the chosen option.

The OODA category aspect of this scale is advantageous because: 1) it allows more specific verbal description of the level of automation of a specific function than previous scales, and 2) it allows the function types to be weighted differently across a particular level.

The FLOAAT LOA Scales (Figure 2) are bounded by levels 1 and 8, which correspond to complete human and complete computer responsibility, respectively.

The individual levels in each scale are relatively consistent in magnitude of change between levels across the function types. Generally, the levels of automation can be broken down into three sections. In levels 1-2, the human is primary and the computer is secondary. In levels 3-5, the computer operates with human interaction. In levels 6-8, the computer operates independently of the human and the human has decreasing access to information and decreasing override capability. Understanding the differences between the levels is critical to interpreting them correctly. These scales allow system designers to precisely define different levels of automation for individual functions or tasks within an existing or planned design.

## 2. Rendezvous, Proximity Operations, and Docking (RPOD) FLOAAT Study

In 2005, during early CEV requirements development, NASA Johnson Space Center coordinated a study to evaluate the FLOAAT process[10]. The study goal was to use FLOAAT to develop Level 2 requirements for Rendezvous, Proximity Operations, and Docking (RPOD) functions. Upon completion of the study, 21 function-specific RPOD requirements were developed with clear decision-making authority specified. Potential improvements to the FLOAAT Process were identified and completed. As a result, the FLOAAT process was recommended as the methodology for development and analysis of Autonomy and Automation requirements for the CEV.



#### a. Reference Levels of Autonomy and Automation

One of the key outputs of this study are the current levels of automation and autonomy for Shuttle and ISS rendezvous missions. The required levels of automation and autonomy based on preliminary CEV requirements are also captured in this study. Collectively, these are referred to as the ‘reference levels of automation and autonomy’. These levels are helpful in evaluating how the FLOAAT outputs compare to the current Shuttle/ISS implementation and the CEV requirements. For the example in Figure 6, the text of the RPOD requirement is shown in yellow, the current Shuttle/ISS autonomy and automation values and associated reference text are shown in green, and the CEV-directed levels are shown in blue[4].

#### b. Resultant Requirements

After the FLOAAT process is complete, the original requirements are modified to specifically state the decision-making authority. The example given in Figure 7 shows the original requirement in yellow, the LOAA definitions in white, and the resultant requirement in blue. In this case the level of automation calls for both the human and computer to perform this task, with the computer considered prime. When the requirements are complete, a final review is conducted by the domain-area experts to verify that the requirements are at the appropriate LOAAs and that they meet the intent of the initial requirements.

The results of the FLOAAT RPOD study are a valuable tool for determining the correct level of automation for CEV rendezvous functions. This research uses the results of the FLOAAT RPOD study to determine a set of functions to prototype. The prototype will be used to verify the accuracy of the FLOAAT output and determine AI decision-making techniques for use in automation of CEV rendezvous functions.

#### D. Role of Artificial Intelligence (AI) in Modeling Human Decision-Making

The CEV will have increased levels of vehicle autonomy over previous human spacecraft[2]. Because of limited crew sizes much of the increases in autonomy will be realized by the use of on-board automation. To meet these challenges, new techniques must be adopted for use in spacecraft automation. By modeling human decision-making using automated software, ground support costs can be reduced and spacecraft will be able to overcome issues associated with communication delays between the spacecraft and ground support.

Numerous advances have been made since the development of the Space Shuttle in both computer speed and Artificial Intelligence (AI) techniques. More capable on-board computers and mature AI techniques will play a critical role in automation for the CEV. This research will select an AI technique to prototype human decision-making functions for spacecraft rendezvous.

| From: FLOAT RPOD Functional Requirements Document Baseline Version |   |   |  |  |
|--|---|---|--|--|
| Task   | Requirement   | Rationale   | Potential Implementation   | OODA Type  |
| 5.3.2<br>Rendezvous<br>Trajectory<br>Maintenance                   | [FLOAT_CE_V_0190]<br>The CEV Systems shall decide whether to continue on the current trajectory plan or to modify the current trajectory plan during the Rendezvous Flight Phase.   | Rationale: After determining an alternate plan, the system must determine whether it is appropriate to switch to this plan. This requirement allows for the possibility that humans would be involved in determining whether to switch from the previous plan to an alternative plan. | Potential Implementation: If the current plan does not meet rendezvous objectives and constraints, and an alternate plan does meet them, switch to the alternate plan. | OODA Type: Decide  |
|  | Current Shuttle/ISS RPOD Reference  |   |  |  |
|  | Automation Level  | Automation Reference  | Autonomy Level   | Autonomy Reference   |
|  | 2   | The FDO and the Prop officer are responsible for deciding if a change to the current plan is necessary. This may include using a computer as a tool, but not as a partner in the decision process.  | 1  | This is only performed on the ground during the Rendezvous Flight Phase. |
|  | CEV/EDS RPOD as Described in the Level 1 Documents Reference  |   |  |  |
| Automation Level   | Automation Reference  | Autonomy Level  | Autonomy Reference   |  |
| 6  | The human should be shown why the highest ranked option was chosen.<br><br>CTS0300H The CTS shall provide autonomous operations linked to the mission phase and function with an objective of autonomous operations throughout the mission. Rationale : A | 3   | CTS0300H The CTS shall provide autonomous operations linked to the mission phase and function with an objective of autonomous operations throughout the mission.       |  |

Fig. 6. Reference Levels of Automation and Autonomy

| <b>Original Requirement</b>  | <b>Automation Level</b> | <b>Level Definition</b>  | <b>Derived Requirement</b>   |
|--|-------------------------|--|--|
| [FLOAAT_CEV_0190] The CEV Systems shall decide whether to continue on the current trajectory plan or to modify the current trajectory plan during the Rendezvous Flight Phase. | <b>4</b>                | <b>Both the human and the computer perform ranking tasks, the results from the human are considered prime.</b> | [FLOAAT_CEV_0190] The CEV onboard crew, onboard computers, ground flight controllers, and ground computers shall decide whether to continue on the current trajectory plan or to modify the current trajectory plan during the Rendezvous Flight Phase with computer results considered prime. |
|  | <b>Autonomy Level</b>   | <b>Level Definition</b>  |  |
|  | <b>3</b>                | <b>The decision is made both onboard and on the ground and the final decision is negotiated between them</b>   |  |

Fig. 7. Requirement Based on FLOAAT Output

## CHAPTER III

### CANDIDATE RENDEZVOUS FUNCTIONS FOR PROTOTYPING

This section describes the process for selecting rendezvous functions to prototype. First, the selection criteria and desirable attributes are detailed. Candidate high-level functions are listed including a brief description of the function, current Shuttle/ISS levels of automation, and FLOAAT recommended levels of automation. By examining the rendezvous flight rules[12], additional details captured in the FDO On-Orbit Handbook [6] and discussions with NASA rendezvous experts, the high-level functions are then decomposed into more specific candidate functions suitable for prototyping. Next, the final selection process of functions for prototyping is described. Finally, the selected functions are described in detail including associated decision-making criteria and flight rules.

#### A. Selection Criteria

The selection process of functions to prototype was based on numerous factors described in this section. The main criteria for selection of functions to be prototyped is the comparison between levels of automation output by FLOAAT and those that exist today for Shuttle/ISS docking. Good candidates have an increased level of automation specified by FLOAAT outputs compared to current levels of automation. In other words, desirable functions are not currently automated at high levels, but the FLOAAT process recommends them for higher levels of automation. Also, the procedures for these functions should be explicitly detailed in flight rules and associated documents. In order to test and compare to existing Shuttle rendezvous methods, another criteria is that algorithms exist to provide the data that the decisions are based upon (it is beyond the scope of this research to design new trajectory plan-

ning algorithms). The selected functions should also lend themselves well to available computed based decision-making approaches. Therefore, it was an iterative process between selection of rendezvous functions and AI methods. Discussions with experienced Shuttle rendezvous experts were also critical to identify functions that they recommend for increased automation. Summary of the selected criteria:

- Higher FLOAAT recommended LOA than current Shuttle levels
- Procedures for the functions are detailed in existing Shuttle flight rules and procedures
- Functions are based on readily available trajectory data from existing trajectory algorithms
- AI decision-making techniques are available that will likely produce a feasible solution
- Recommended for automation by Shuttle rendezvous experts

## B. Candidate Functions

The candidate high-level functional requirements are taken from the FLOAAT RPOD study[10]. Figure 8 is the complete list of rendezvous functional requirements and associated levels of automation. This list includes only decision-making requirements. Each task is broken down into the four OODA components. For example, the trajectory maintenance task is divided into monitoring state (FLOAAT\_CEV\_0170), calculating the health of the current trajectory (FLOAAT\_CEV\_0180, FLOAAT\_CEV\_0185), deciding if a change is needed (FLOAAT\_CEV\_0190), and executing any changes (FLOAAT\_CEV\_0200). Many detailed functions are necessary for each of these high-level functions. The details of how the Shuttle is operated currently is provided in the ‘Automation Reference’ column. These details summarize the information that was captured and used to determine the reference level of automation.

|  |  |           | Current Shuttle/ISS RPOD Reference |  | FLOAAT Output    |
|--|--|-----------|------------------------------------|--|------------------|
| Task   | Requirement  | OODA Type | Automation Level                   | Automation Reference   | Automation Level |
| <b>Rendezvous Trajectory Maintenance</b>   | [FLOAAT_CEV_0170] The CEV Systems shall monitor the real-time navigation state and consumables of the chaser and the navigation state of the target, as required for a successful Rendezvous Flight Phase.                     | Observe   | 5                                  | Radar ground tracking is at level 5 for Shuttle. There is also some computer monitoring of propellant.   | 5                |
|  | [FLOAAT_CEV_0180] The CEV Systems shall calculate if the current trajectory plan and consumables budget are sufficient to continue within constraints on the current rendezvous plan, and shall plan the next burn parameters. | Orient    | 3                                  | A computer is used as a "calculator" only. Human selects inputs, initiates analysis, and interprets results. The frequency of this analysis is up the FDO. It is typically performed at least twice per 8-hr shift.  | 4                |
|  | [FLOAAT_CEV_0185] The CEV Systems shall calculate, when necessary, if an alternative trajectory plan or consumables budget is capable of achieving the rendezvous mission objectives and operating within constraints.         | Orient    | 3                                  | A computer is used as a "calculator" only. Human selects inputs, initiates analysis, and interprets results. The frequency of this analysis is up the FDO. It is typically performed at least twice per 8-hr shift.  | 4                |
|  | [FLOAAT_CEV_0190] The CEV Systems shall decide whether to continue on the current trajectory plan or to modify the current trajectory plan during the Rendezvous Flight Phase.   | Decide    | 2                                  | The FDO and the Prop officer are responsible for deciding if a change to the current plan is necessary. This may include using a computer as a tool, but not as a partner in the decision process.   | 4                |
|  | [FLOAAT_CEV_0200] The CEV Systems shall command the GN&C system to follow the chosen rendezvous plan.  | Act       | 1                                  | There is no computer involvement in issuing the commands.  | 4                |
| <b>Plan Potential Aborts During Rendezvous</b><br><br>The decision of which abort mode is best is performed in FLOAAT_CEV_0270.<br><br>No Act requirement is necessary. The abort action is taken in FLOAAT_CEV_0280.  | [FLOAAT_CEV_0210] The CEV Systems shall monitor abort availability and maintain a list of abort modes or categories during the Rendezvous Flight Phase.  | Observe   | 1                                  | The list is maintained in flight rules and procedures.   | 4                |
|  | [FLOAAT_CEV_0220] The CEV Systems shall calculate if potential abort modes are achievable during the Rendezvous Flight Phase.  | Orient    | 1-3 Depending on the type of Abort | Some aborts are more automated than others. For example TIG Slip is calculated by the computer, but the human interprets the results before any decisions are made.  | 4                |
| <b>Determine if an Abort is Necessary During Rendezvous</b><br><br>The monitor functions for this task are accomplished by FLOAAT_CEV_0130 and FLOAAT_CEV_0170<br><br>The Analysis of which abort modes are achievable is performed in the Orient Requirement FLOAAT_CEV_0220. | [FLOAAT_CEV_0250] The CEV Systems shall monitor the current need for abort during the Rendezvous Flight Phase.   | Observe   | 1                                  | The need for abort is determined by flight rules, systems status, opinion, etc.  | 4                |
|  | [FLOAAT_CEV_0270] If an abort is necessary during the Rendezvous Flight Phase, the CEV Systems shall decide which abort plan is necessary.   | Decide    | 2                                  | Which plan to follow comes from the flight rules. Which are derived pre-launch. In real-time this is a human only function that reads from the flight rules. A computer may be used a tool to help determine which flight rules is applicable for this scenario. | 4                |
|  | [FLOAAT_CEV_0280] The CEV Systems shall command GN&C to execute the chosen abort plan when necessary during the Rendezvous Flight Phase.   | Act       | 1                                  | All commands come from the human   | 4                |

Fig. 8. Candidate High-Level Rendezvous Functions



### C. Flight Rules

The rendezvous flight rules explicitly detail how decisions should be made during the on-orbit execution of rendezvous. As a result these rendezvous flight rules apply primarily to the ‘Decide’ functions of the OODA loop. After reviewing the rendezvous and prox ops flight rules it became clear that the majority of them apply to the execution of burns. In particular, flight rules provide criteria for when burns can be executed, when burns can be delayed and the duration of the delay, and what systems are required to execute burns. A detailed summary of the rendezvous flight rules is included below[12]. The FDO on-orbit handbook[6] provides additional explanation of the rules.

- Definitions of Rendezvous (RNDZ) and Proximity Operations (PROX OPS), A2-116
  - PROX OPS begins when range < 1000 ft and LVLH relative velocity is < 1 fps in each axis
- Systems required for executing the Ti burn, A2-117
  - Properly functioning hand controllers (used for manual burn commands)
  - Two functioning Inertial Measurement Units (IMUs)
- Priority (preference) of rendezvous sensors, A2-119
  - Delay of Ti burn may be allowed to regain a higher priority sensor
- Propulsion minimum requirements (limits) for burn execution, A2-121
  - Red-line reserve requirements for payload, day of landing weather buffer, allowance for nose/tail-only jet configuration

- Minimum propellant quantities for GO for Ti and Ti-delay
- Thruster guidelines to determine which thruster configuration to use (multi-axis; attitude burn to +X jets; Orbiter Maneuvering System (OMS) jets)
- Source of burn solutions (ground or on-board), A2-122
  - Ground solutions will be executed for all maneuvers up to NCC
  - Orbiter will compute targets for NCC and subsequent maneuvers
- Rendezvous Maneuver Solution Selection Criteria, A2-123
  - NCC and Ti will be computed both on-board and on the ground with on-board prime (selection criteria listed in the flight rule)
- Maneuver trim and time of execution requirements, A2-124
  - Maneuver components will be trimmed to  $< 0.2$  fps in all axes
  - Lambert Guidance Burns (NCC, Ti, and MC burns) must be completed by Time of Ignition (TIG) + 90 seconds
- Rules for delay of Ti burn, A2-125
  - Criteria and recommendation of a Ti delay
- Plan for executing a breakout (abort) maneuver, if necessary, A2-126
  - Prior to Ti, a ground-computed breakout maneuver will be executed, if necessary, to guarantee a safe miss distance between the Orbiter and the target.

#### D. Functions Selected for Prototyping

The candidate high-level functional requirements shown in Figure 8 indicate that rendezvous is not a highly automated flight phase. However, the FLOAAT results recommend a higher level of automation for almost all of the functions. Therefore, almost all of the high-level rendezvous functions meet the criteria of a higher FLOAAT recommended level of automation than the current implementation.

Since the flight rules are most applicable to the ‘decide’ functions, two high-level decide functions have been selected.

- [FLOAAT\_CEV\_0190] The CEV Systems shall decide whether to continue on the current trajectory plan or to modify the current trajectory plan during the Rendezvous Flight Phase.
- [FLOAAT\_CEV\_0270] If an abort is necessary during the Rendezvous Flight Phase, the CEV Systems shall decide which abort plan is necessary.

By evaluating these high-level functions and the flight rules, a list of potential prototype functionality was generated. This list includes calculation of a breakout burn for an abort (flight rule A2-126), the duration a burn can be slipped (delayed) before the burn can no longer be executed (detailed in FDO handbook[6]), and when to execute a burn maneuver to delay the Ti maneuver (Ti-delay, flight rule A2-125). All of these functions use data from existing trajectory algorithms so they are valid functions to prototype. These functions are also evaluated during selection of decision-making algorithms and are applicable for the techniques surveyed. The prototype concepts matured during personal conversations with rendezvous experts[18, 19, 20, 21]. After these conversations and additional consideration to determine the right level of detail, two functions were selected for prototype. One function was selected

for each FLOAAT requirement [FLOAAT\_CEV\_0190] and [FLOAAT\_CEV\_0270].

### 1. Time of Ignition (TIG) Slip Planning

For requirement [FLOAAT\_CEV\_0270] ('...decide which abort plan is necessary'), the function selected for prototyping is to determine the duration a burn can be slipped (delayed) before the burn can no longer be executed. In this context 'abort' is considered to mean executing a burn after the planned Time of IGnition (TIG). This is known as a 'TIG slip', and the maximum TIG-slip duration is calculated for every burn in the rendezvous plan. A TIG slip could be necessary if a burn needs to be delayed for any of several reasons including chaser vehicle system issues, target-vehicle system issues, etc.. Since the burn now occurs at a different time and location in the trajectory, slipping a planned burn will result in different relative motion than originally planned. It also results in increased propellant usage to return to the planned trajectory. Typically, the duration of a TIG slip is limited by deviations in the resulting relative motion or increased propellant usage. For most burns, the TIG-slip duration is less than 5 minutes.

In the current implementation of this function, a computer program is used as a tool, and a human flight controller iteratively runs the program to determine the maximum slip duration for each burn[6] (level of automation of 2 on the 'decide' scale). The FLOAAT recommended level of automation for [FLOAAT\_CEV\_0270] is 4, 'Both the human and the computer perform ranking tasks, the results from the computer are considered prime'. Therefore, the prototype should result in an automated process that determines maximum TIG-slip duration with this result considered primary while still allowing crew/flight controller back-up and override capability. The prototype automates the determination of maximum TIG-slip for the NC (phase change) burn shown in Figure 3. There are two ways to execute a TIG slip. These two methods

are called an ‘inertial TIG slip’ and an ‘LVLH TIG slip’.

In an inertial TIG slip, the burn targets (target  $\Delta V$ 's computed in the inertial frame) for the original burn are used with the Orbiter in an inertial attitude hold. The resulting burn will be slightly different than originally planned when the TIG is delayed. This is because the LVLH and inertial frame are slowly drifting out of alignment. The frames are only equivalent for the planned TIG time, not for the new TIG time. The LVLH frame rotates at the orbit rotation rate of the ISS, which is equal to 4 degrees per minute. Therefore, the longer the TIG slip is the bigger the difference will be between the planned and actual trajectories. As the difference in trajectories increases, the cost in terms of propellant also increases.[6]

Figure 9 plots a family of relative motion trajectories used for TIG-slip planning of the NC burn for an inertial TIG slip. This is an example of the TIG-slip analysis that flight controllers perform for each rendezvous burn during every Shuttle mission. The hand written markings denote the nominal trajectory (labeled ‘NOM’), 1-minute (labeled ‘1’), 2-minute (labeled ‘2’), and 3-minute (labeled ‘3’) inertial TIG slips. In each case, the NC burn must be successfully executed to reach the desired relative position at the correct time to execute the Ti burn. Also included are the propellant costs for each of the slips written in terms of  $\Delta V$  in feet per second (ft/sec). For each TIG-slip duration, the flight controller uses the relative motion plot to determine if the trajectory violates a 4 nautical mile (nm) constraint on relative distance between the Orbiter and ISS (located at the origin). In the example given in Figure 9, this constraint is shown as a solid gray line. For this case, it indicates that the 2-minute TIG slip will just barely violate the relative motion constraint. In terms of relative motion, the maximum TIG slip would be slightly less than 2 minutes. For this case, the inertial TIG slip of slightly less than 2-minutes will cost approximately an additional 13 feet per second of  $\Delta V$  over the nominal burn plan. After evaluating the

maximum TIG slip based on relative motion, the maximum slip duration and  $\Delta V$  cost is captured to use for comparison to the LVLH TIG-slip method.

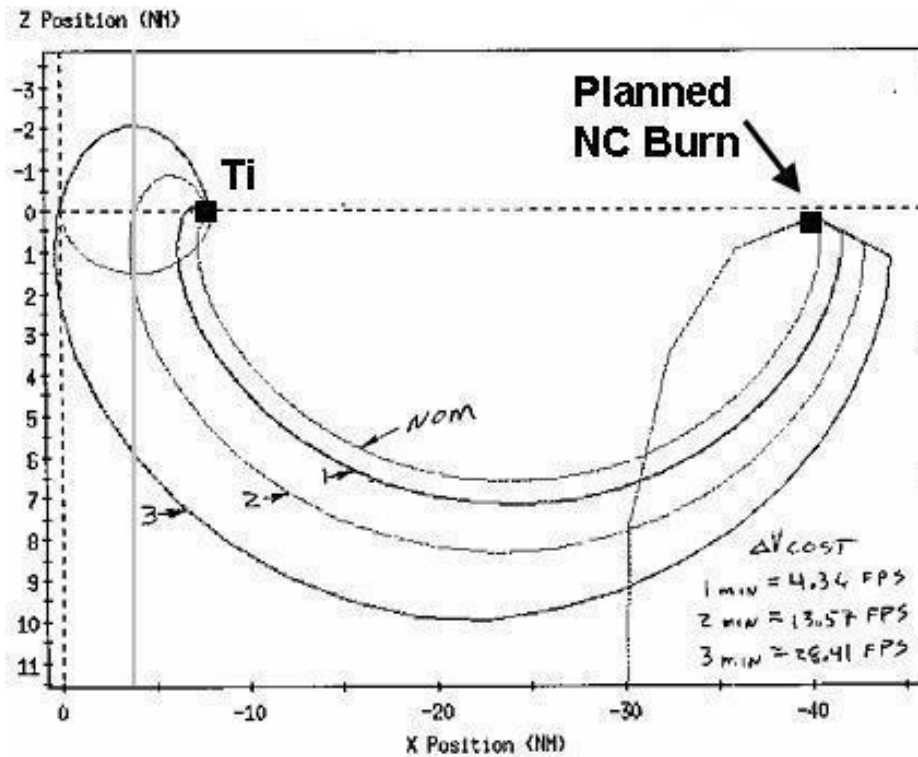


Fig. 9. Inertial TIG slip

For an LVLH TIG slip the burn targets stay the same, but the new TIG results in a different inertial burn attitude. Unlike an inertial TIG slip where the burn attitude is inertially fixed, for an LVLH TIG slip, the Orbiter's burn attitude is changed to the new inertial attitude when the burn is executed [6]. Despite this difference in the maneuvers, the process to determine the maximum TIG slip is the same. Figure 10 shows the nominal trajectory (labeled 'NOM'), 1-minute (labeled '1'), 2-minute (labeled '2'), and 3-minute (labeled '3') LVLH TIG slips of the NC burn. The propellant costs for each of the slip cases written in terms of  $\Delta V$  in ft/sec are also included in Figure 10. Just as for the inertial TIG slip, the maximum LVLH

TIG-slip duration is determined by evaluating the relative motion plots for a violation of the 4-nm constraint (shown as the solid gray line). For this example, the maximum LVLH TIG slip is slightly over 1-minute. To execute a 1-minute LVLH TIG slip, there is a cost of an additional 12 feet per second of  $\Delta V$  over the nominal burn plan.

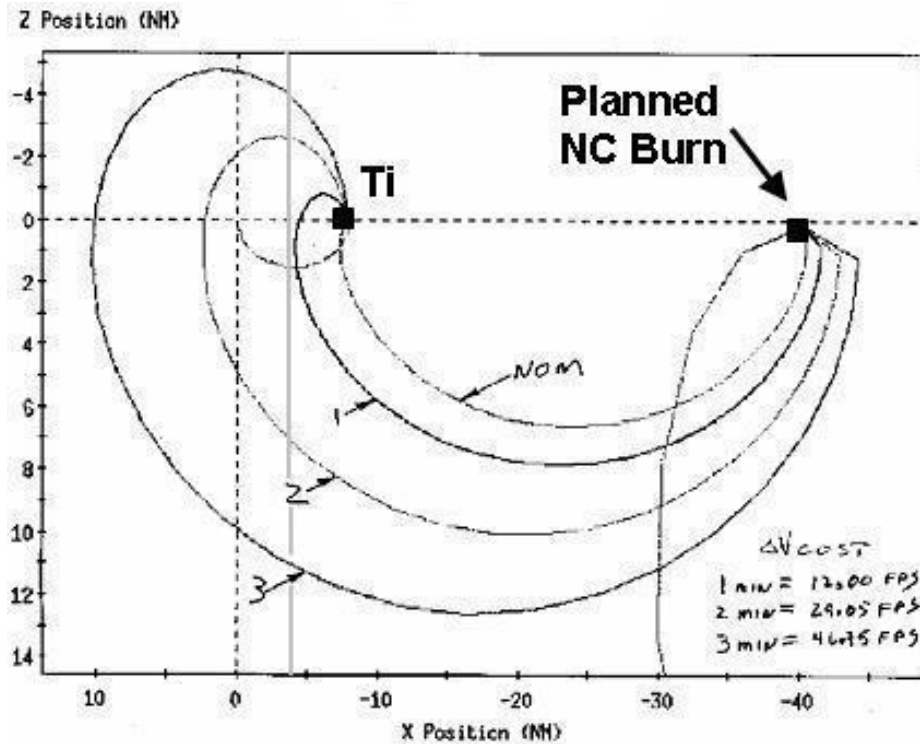


Fig. 10. LVLH TIG slip

After both methods for executing the TIG slip are evaluated, the results are compared to select which method to use and to specify the maximum TIG-slip duration. If the results are equal for relative motion and propellant, the inertial TIG slip is preferred because it is easier for the crew to execute since the inertial burn attitude is unchanged. However, if the LVLH slip has a longer maximum duration or lower propellant costs, then it could be selected over the inertial TIG slip. The prototype will be used to determine the maximum duration for both inertial and LVLH TIG

slips and provide a recommendation of the TIG-slip method to execute.

## 2. Transition Initiation (Ti) Position Constraints

Many of the rendezvous flight rules focus on the execution of the Ti burn. Since the Ti burn results in a rear intercept trajectory with the target spacecraft, it is critical to the safety of crews on-board both spacecraft. For requirement [FLOAAT\_CEV\_0190], ('...decide whether to continue on the current trajectory plan or to modify the current trajectory plan...'), the function selected for prototyping is determining when to execute a burn to delay the Ti maneuver based on position constraints and current navigation quality.

The decision to execute a Ti delay can be made in response to any number of system issues/failures. Executing a Ti-delay maneuver allows for time to understand failures, determine impacts of failures, generate workarounds, and plan an appropriate response. In addition to system failures, trajectory errors can result in a Ti-delay if the predicted relative position at the planned Ti TIG exceeds  $3\text{-}\sigma$  trajectory bounds or the position constraint based on lighting. Figure 11 shows an example nominal trajectory and the position constraints. During a Shuttle mission, the rendezvous flight controllers monitor the predicted Ti position and compare this position to the constraints. If these constraints are exceeded, then a Ti-delay burn can be used to delay the Ti so the trajectory can be corrected. The decision to execute a Ti delay depends on the severity of the position-constraint violation. In addition to the position constraints, the flight controllers evaluate the current navigation accuracy. If the navigation state is inaccurate, the predicted Ti position will also include inaccuracies. This is taken into account when comparing the predicted Ti position with the constraints. For large errors in navigation accuracy, a Ti-delay could be recommended even if the predicted position appears well within the constraint.



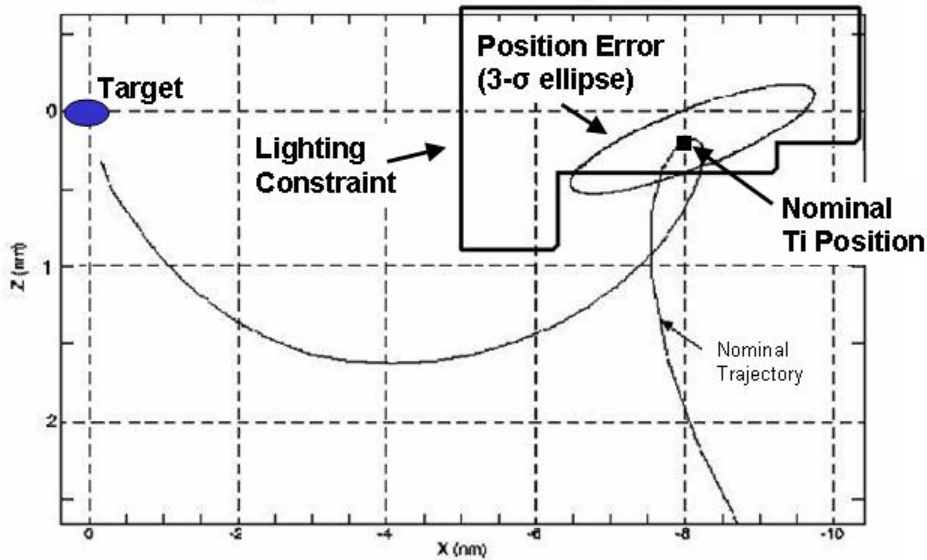


Fig. 11. Ti Position and Lighting Constraints

This assessment is also used to evaluate the trajectory during a Ti-delay maneuver. During a Ti-delay, a ‘football orbit’ is executed to return the chaser to the original relative position to execute the delayed Ti burn (Figure 12). As part of the Ti-delay trajectory, an NCC correction burn is executed to improve the accuracy of the Ti position. A prediction will be made during the Ti-delay to determine if the NCC burn is necessary. The ‘severity’ of the Ti position error is used to determine the priority of the execution of the NCC correction burn. For example, if a Ti-delay was executed because of a system error, then the crew will likely be occupied trying to troubleshoot and correct the error. In this case, the crew may not have time to execute the NCC burn unless the Ti position error is considered very severe.

In the current implementation of this function, ground computers are used to predict the location of the Orbiter when the Ti burn is to be executed and the flight controllers determine if this position is acceptable (level of automation of 2 on the ‘decide’ scale). The FLOAAT recommended level of automation for [FLOAAT\_CEV\_0190] is

4, ‘Both the human and the computer perform ranking tasks, the results from the computer are considered prime’. Therefore, the prototype effort should produce an automated process that determines the ‘severity’ of the predicted Ti position with this result considered primary while still allowing crew/flight controller back-up and override. When implemented, a threshold can be set for the severity that would result in a Ti-delay.

The three factors that are the primary drivers for determining if the predicted Ti position is acceptable are detailed here. The location relative to the Ti  $3\text{-}\sigma$  error ellipse is a constraint based on the resulting relative motion after the Ti burn is executed. If the error is beyond the  $3\text{-}\sigma$  boundary, relative motion between the target and chaser spacecraft can become unacceptable following the Ti burn. The lighting constraint, shown in Figure 11, is based on the lighting conditions at the MC-2 burn TIG. The burn TIG for MC-2 is based on elevation angle to the target (Figure 12). Any error in the Ti position will change the relative motion between Ti and MC-2. This will in turn result in a change in the time the MC-2 burn is executed and thus a difference in the lighting conditions that the crew will experience for the remainder of the trajectory. The acceptable limit on MC-2 TIG time based on lighting is between -3 minutes and +7 minutes of the nominal MC-2 TIG time. The boundary shown in figure 11 is the boundary on the Ti position that will keep the MC-2 TIG within the lighting constraints. The quality of the navigation solution also affects how these constraints are evaluated. If the current navigation state is accurate, then the constraint boundaries are evaluated as they appear in Figure 11. However, if there are some known inaccuracies in the navigation state, there must be some additional conservatism added to the constraint boundaries to account for navigation errors. In the extreme case, this might result in a delay of Ti with a predicted position that is well within the constraints.

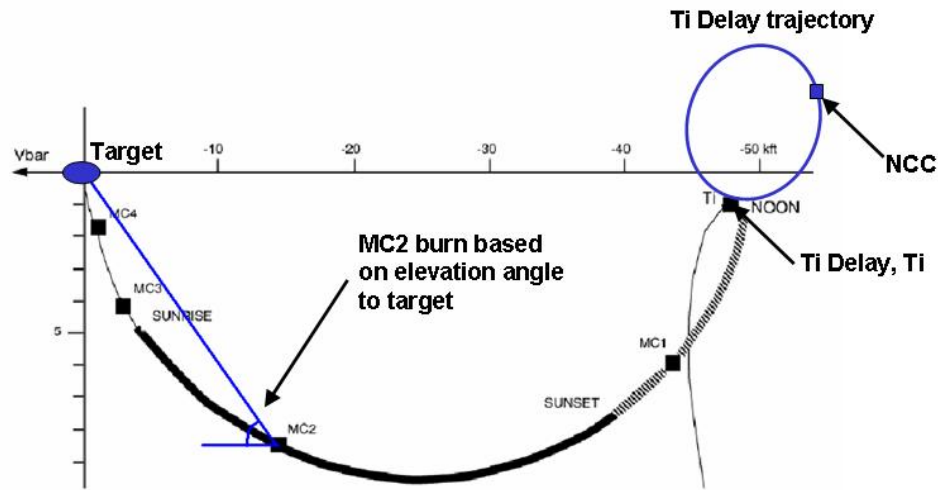


Fig. 12. Ti Delay Trajectory

When the predicted Ti position relative to the constraints and the navigation accuracy are taken into account, the flight controller makes a judgment based on the severity of the predicted Ti position. Then, based on their experience from Shuttle flights and simulation exercises, they determine if a Ti-delay should be executed. This is a subjective decision that could potentially vary depending on which flight controller is making this assessment. The prototype will seek to model this decision-making process and remove some of the subjectivity. The results of the prototype will be used to help make this decision but is not intended to replace the experience of the flight controllers.

## CHAPTER IV

### CANDIDATE ARTIFICIAL INTELLIGENCE DECISION-MAKING METHODS

The next step is to select an AI decision-making technique to use for prototyping the selected rendezvous functions. This section describes the selection process. The selection criteria and desirable attributes are detailed. Then, each method is briefly described, including a list of strengths and weaknesses. Candidate methods included, but were not limited to, neural networks, expert systems, and fuzzy logic. After the description of the methods, examples are given of how the method has been used in previous spaceflight applications. Finally, the final selection of Fuzzy Logic is described in detail.

#### A. Selection Criteria

This section describes the criteria used to select a decision-making method for the prototypes. As mentioned previously, the method chosen depends heavily on the selected functions. The selected method should be compatible with rendezvous decision-making processes described in the previous section. This decision-making is based on flight rules and procedures, which must be properly modeled in the prototype.

During the selection process, special considerations for human spaceflight must be addressed. In human spaceflight applications, safety of the crew is paramount and many steps are taken to ensure their safety. As a result, the operations of the vehicle are constantly being adjusted to maximize capability to ensure safety. For the prototype to be successful, it must have the flexibility to accommodate changes in the way the spacecraft is operated. Since the rendezvous functionality will not be fully automated, a human flight controller must be able to quickly and easily understand the output of the automated system. If the flight controller cannot understand the

output, then they are at risk of incorrect action that could jeopardize the safety of the crew. In order to allow for regular updates to the automated software, the selected method should be simple enough so the verification and validation process is streamlined. This is particularly important given the rigorous testing standards used for human spacecraft. A summary of the selection criteria is included below:

- Compatibility with the selected rendezvous functions
- Capability to model human decision-making captured in flight rules and procedures
- Flexibility to accommodate changes to procedures
- Easily understood by human flight controllers
- Simple software validation and verification

## B. Candidate Decision-Making Methods

Candidate decision-making methods were identified by evaluating available AI techniques. After reviewing the available methods and carefully considering their applicability to rendezvous decision-making, three were selected: Neural Networks, Expert Systems, and Fuzzy Logic [22, 23]. Each of these methods are detailed below.

### 1. Neural Networks

Neural Networks are modeled after the neurons that comprise the human brain. They are used to produce a desirable input-output mapping by creating a network of ‘neurons’ (Figure 13). Neural Networks are useful for modeling systems where the relationship between inputs and outputs is not fully known, and these networks can

model very complex systems. In particular, they can be useful in modeling systems that are not fully understood and are difficult to model analytically such as human decision-making.

In each individual neuron, the input signals are multiplied by weighting factors resulting in amplification or attenuation. The strength of connection (weighting factor) is given a numeric value between -1.0 for maximum attenuation and +1.0 for maximum amplification. The signals are then combined (typically added) to form the output of that neuron. A Neural Network consists of a series of neurons that are connected with the output of one neuron serving as the input of another. This network of neurons can contain many intermediate layers between the input signals and outputs (Figure 14).

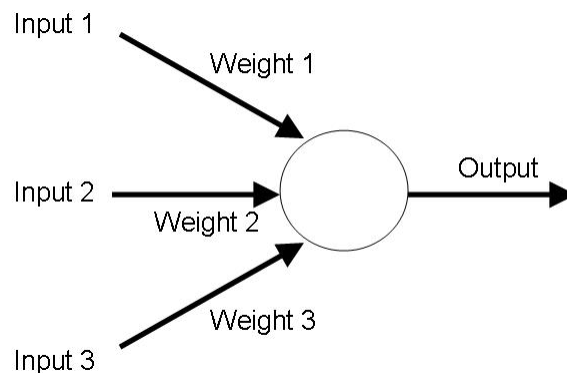


Fig. 13. Simplified Artificial Neuron

Training is used to properly set connection weights to produce the desired output for a given input. During training, characteristic inputs of the system being modeled are injected into the network, and the resulting outputs are compared to the desired values. Some connections between neurons are reinforced by increasing the weighting, while other connections weaken or disappear altogether. The weighting factors

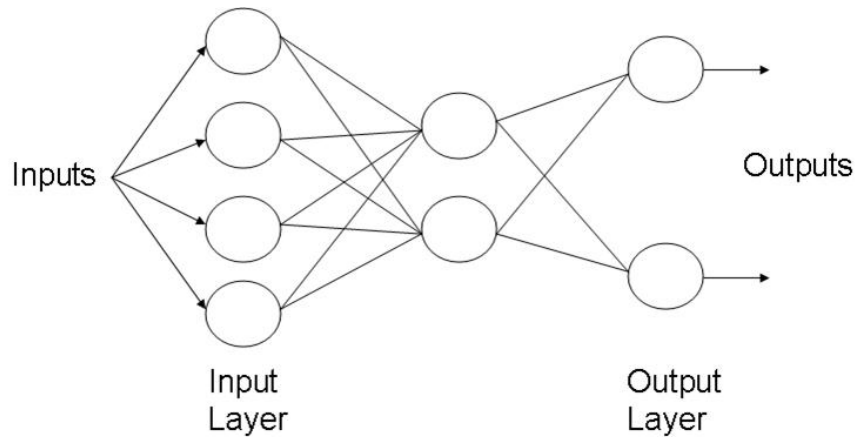


Fig. 14. Simplified Neural Network

between the neurons are adjusted and the training continues until the network outputs converge to the desired values. Once the network is trained, it can generalize its experiences to new cases. [24]

Strengths and weaknesses of Neural Networks include:

- Strengths:
  - Neural Networks are able to model relationships that are poorly understood or difficult to determine analytically.
  - Connection weights can indicate strength of the relationship between inputs and outputs.
- Weaknesses:
  - Training a Neural Network can be time consuming.
  - Data not previously seen in training sets may not exhibit desired behavior.
  - It is difficult to model formal reasoning and rule-following using Neural Networks.

A literature review found that Neural Networks have been infrequently used for human spaceflight applications. However, this technique was applied as a potential tool to determine when to execute an abort during the precision landing of the robotic lander called the Mars Science Laboratory (MSL)[25]. The resulting Neural Network was successful in properly triggering an abort in cases similar to the training data set, but did not recognize when to abort in dispersed cases outside of the training set. Future work was planned to improve the techniques used to properly train the Neural Network. The difficulty in properly handling cases that deviate from those seen in the training data is an issue for this technique, which would make it difficult to certify for use in human spaceflight because of safety concerns.

There are several concerns in using Neural Networks to model human decision-making for rendezvous functions. Since this technique is not well-suited for rule-following, it would be difficult to create a model that would capture the flight rule-based decisions of human flight controllers. Also, the intensive training process makes the models difficult to modify to meet changing operational procedures. The connection weights capture the details of the relationships between inputs and outputs, but are difficult for a user to easily understand.

## 2. Expert Systems

Expert Systems are used to capture the experience of domain experts. They can be effectively used to capture knowledge that is heuristic in nature and is based on ‘rules of thumb’. The expertise is captured in a Knowledge Base (KB) that typically consists of IF-THEN relations in the form of condition-action rules. Once the experience of the experts is captured in the KB, the Expert System can be used to diagnose and solve problems. They are typically most useful in modeling systems that require specialized expertise, but would only require an expert a short period of time to complete ( $< 1$



hour). [23]

When the expertise is captured in the form of condition-action rules, results are determined by using a technique called ‘forward-chaining’. In forward-chaining, conditions result in actions until all rules with satisfied conditions execute or a halt is reached. These actions can change the system outputs, result in additional actions, or trigger other events. The resulting outputs will reflect the domain expertise captured in the KB.

An Expert System shell program is used to provide an interface with the user, execute the problem solving by evaluating the rules (called an ‘inference engine’), and explain the reasoning behind the results. Several Expert Systems shells are available as public domain software including the NASA developed Expert System shell called C Language Integrated Production System (CLIPS). CLIPS was developed to provide an affordable expert system application that is compatible with diverse hardware and operating systems. The development of tools like CLIPS has helped to spread the use of Expert Systems throughout the public and private sectors[26].

Strengths and weaknesses of Expert Systems include:

- Strengths:
  - Human knowledge is often based on useful “rules of thumb” which can be modeled with an expert system.
  - Typically a suitable problem for an Expert System requires specialized expertise but would only require a short time for a human expert to complete (< 1 hour).
- Weaknesses:
  - It can be difficult to capture all of the information required to create a

accurate system.

- Expert Systems require experts who are willing and available to provide help.
- Expert System models become complex as the number of rules increase.

Expert Systems have been previously applied to the spacecraft rendezvous planning [27]. However, these applications were intended for use in preflight planning to generate the overall rendezvous plan. The purpose was to create a preliminary plan for the human rendezvous expert to work from and refine to create the final rendezvous plan. Such tools were abandoned in favor of analytical optimization techniques that do not rely on “rules of thumb” [28]. These applications were not used to model the rendezvous decision-making of flight controllers during on-orbit operations.

The Expert Systems techniques could apply well for modeling human decision-making since the expertise is captured as “rules of thumb” in the flight-rules and procedures. Also, the KB can be easily updated to reflect changes to the operational procedures. The results can be easily understood by a human user because most Expert Systems shells provide an explanation of the reasoning in addition to the outputs. A difficulty in implementing this type of system is that as the KB grows, the complexity of the system rapidly increases. This could lead to issues with testing and verification of the system. Also, since it is difficult to capture all of the subtle details of how an expert makes a decision, important details can be inadvertently omitted. Another issue is that the flight rules and procedures for the candidate rendezvous functions use approximate terms, such as ‘slightly outside the boundary’, which are difficult to model using conventional conditional logic.

### 3. Fuzzy Logic (FL)

Similar to Expert Systems, Fuzzy Logic (FL) uses IF-THEN rules to model relationships and produce a desirable input-output mapping. Both methods leverage off the experience of experts to model relationships. However, FL can model relationships that are more complicated than binary answers (yes=1 and no=0). Unlike classical binary logic, where some X must be either in ‘set A’ or in set ‘not-A’, FL entries can have a partial degree of membership in a set. For example, in classical binary logic the date June 12th would be a member of the set ‘Spring’, but not a member of the set ‘Summer’. In a FL model of the seasons, this same date could have partial membership in both Spring and Summer, which is a more accurate representation of how humans perceive seasons [29, 30].

A FL model maps input-space into output-space based on a degree of membership using IF-THEN type logic. This capability makes FL a powerful method for modeling human decision-making based on approximate terms that are common in natural language. Since FL is based on the use of natural-language terms, the models are intuitive to create and easy to maintain. This method also allows complex relationships to be modeled rapidly.

The similarities between Expert Systems and FL result in common weaknesses. It may be difficult to adequately capture all of the information necessary to correctly model the system. FL is more susceptible to the potentially prohibitive increase in the complexity of models as the rule-base increases. This is due to the increased complexity of rules that account for degrees of membership instead of binary logic.

Strengths and weaknesses of Fuzzy Logic include:

- Strengths:
  - Models leverage off the experience of experts.

- Models are based on natural language.
  - Results in models that are easy to create, understand, and modify.
  - Allows rapid modeling of complex rules and non-linear relationships.
  - Provides the flexibility to easily add new relationships to existing models.
- Weaknesses:
    - Difficulty in capturing all of the information required to create an accurate system.
    - Requires experts who are willing and available to provide help.
    - Defining the rule-base becomes complex if too many inputs and outputs are required.

The most common application of FL is in the creation of control systems that incorporate the subjective nature of human decision-making. FL control techniques have been applied to the autonomous rendezvous and docking control problem [31, 32]. FL has also been studied by NASA to model human piloting for simulation analysis of Space Shuttle rendezvous performance[33, 34, 35]. Another application of FL is automated navigation for mobile robots [36]. These applications show the varied applications of FL for spacecraft, but do not attempt to model human decision-making in an operational context.

FL is a promising technique for modeling human decision-making. It has the capability to capture the flight rules and procedures with IF-THEN relationships, and unlike Expert Systems, FL is able to model the approximate terms found in the flight rules and procedures. These models can be quickly updated to satisfy evolving operational procedures. By capturing relationships using natural language, FL models are easy for a human flight controller to understand. One issue is the

increase in model complexity as the number of rules is increased. This could lead to issues with ease of testing if the system becomes too complex. It is also sometimes difficult to ensure that all of the necessary details of how an expert makes a decision are captured.

#### 4. Other Methods

Other methods were also surveyed for potential implementation, including decision-trees, search methods such as A\*, and genetic algorithms [22, 23]. However, these methods were ruled out because of their difficulty in modeling the flight rules and procedures used in human-spaceflight decision-making. Whereas some of these techniques may be suitable for other spaceflight applications these methods are not compatible with the selected rendezvous functions.

##### C. Selected Method, Fuzzy Logic

Whereas both Expert Systems and FL are well-suited for modeling the selected rendezvous functions which are captured in flight rules and procedures, terms such as ‘slightly’ and ‘small’ are common in the procedures. Examples from the FDO handbook are given below[6]:

- ‘...it may be prudent to execute a Ti Delay burn...if the trajectory is just slightly outside limits.’
- ‘If a stable football has been achieved post Ti , and NCC delta-V is predicted to be small, then...consider waiving NCC...’[6]

These approximate terms cannot be easily modeled using binary logic but are well suited for modeling using FL. This indicates that FL can successfully model the

flight rules and procedures for the selected rendezvous functions. FL also satisfies the criteria for ease in modification, thus accommodating future changes to operational procedures. The outputs are easily understood by human flight controllers since the models are based on natural-language terminology.

After surveying the decision-making methods described above, FL was chosen because it best satisfies the selection criteria. A summary of selection criteria and methods is captured in Table I. One concern with the FL techniques is the verification and validation of the software due to model complexity. Nonetheless, if the models remain at a reasonable level of complexity, this issue is manageable. Since the prototypes are intended to model individual tasks and not the entire decision-making process, there is little risk of prohibitive model complexity. This concern should be considered if these types of models are accepted into wide use or are integrated into larger systems. Overall, FL is an excellent candidate for modeling the human decision-making for the selected rendezvous functionality.

Table I. Comparison of AI Decision-Making Methods

|  | <b>Neural Networks</b> | <b>Expert Systems</b> | <b>Fuzzy Logic</b> |
|--|------------------------|-----------------------|--------------------|
| <b>Compatible with the selected rendezvous functions</b> | -                      | -                     | X                  |
| <b>Capability to model flight rules and procedures</b>   | -                      | X*                    | X                  |
| <b>Flexibility to accommodate changes</b>                | -                      | X                     | X                  |
| <b>Easily understood by human flight controllers</b>     | -                      | X                     | X                  |
| <b>Simple software validation and verification</b>       | -                      | X**                   | X**                |

\* Unable to handle approximate (non-binary) terms

\*\* Model complexity issues for large rule-bases

## CHAPTER V

### MODELING RENDEZVOUS FUNCTIONS USING FUZZY LOGIC

This chapter provides a brief overview of FL, including an example of how it is used to model input-output relationships. The FL models used for TIG-slip planning are described in detail. Finally, there is a description of the FL model evaluation of the vehicle position relative to the position constraints for the Ti burn.

#### A. Fuzzy Logic

FL is a powerful technique that enables the mathematical representation of approximate terms such as ‘small’, ‘medium’, and ‘large’ with a continuous range over the closed interval,  $[0,1]$ . This technique, attributed to Zadeh, has become widely used for applications such as control systems, image processing, and to model human decision-making [30]. FL allows the modeling of relationships that are more complicated than binary logic (yes=1 and no=0). The foundation of FL is the concept of a ‘fuzzy set’ which allows intermediate values to be assigned, which fall between yes and no (true and false, medium and large, etc.). This concept is also known as multi-valued logic.

##### 1. Fuzzy Sets

To understand the concept of fuzzy sets and multi-valued logic, classical logic and the concept of a ‘crisp set’ must be understood. In classical two-valued (or binary) logic, the elements of the Universe of Discourse,  $X$ , are mapped to the two-element set  $\{0,1\}$ . In this logic, each element from  $X$  is either a member of the set  $C$  or not a member of the set  $C$ . Those elements that are members of the set  $C$  are assigned a value of 1, and those that are not in the set  $C$  are assigned a value of 0. This type of set is shown graphically in Figure 15. Since there is only full or zero membership in

the set  $C$  and no intermediate values in-between, this is known as a ‘crisp set’ [37].

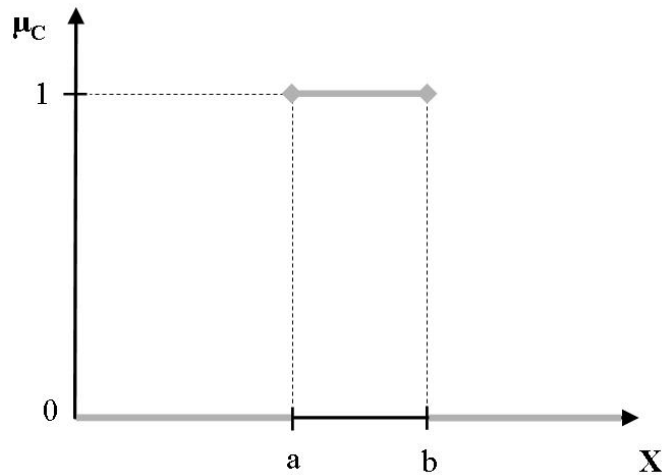


Fig. 15. Crisp Set  $C = [a,b]$

By contrast, a fuzzy set uses multi-valued logic, which results in elements that can have partial membership within a set. Similar to crisp sets, elements are mapped from the Universe of Discourse,  $X$ , to an interval between 0 and 1. However, for fuzzy sets these values are mapped to the continuous interval  $[0,1]$ . As before, elements that map to a value of 1 have full membership in the set  $F$  and elements that map to a value of 0 are not members of the set  $F$ . In addition, elements can now belong to the set  $F$  with a degree of membership ( $\mu$ ), which ranges from 0 to 1. The mapping of this relationship is captured in a function called a ‘Membership Function’ (MF). Figure 16 shows an example membership function which represents a fuzzy set, set  $F$ . In this example, all values within the interval  $[a,b]$  have at least partial membership in the set  $F$  ( $\mu_F > 0$ ). For an input value corresponding to point  $c$ , the membership is equal to 0.6 ( $\mu_F = 0.6$ ), and over the input range of  $[d,e]$  there is full membership in set  $F$  ( $\mu_F = 1.0$ ). There are numerous types of membership functions that can be used to represent membership in a particular set. Some of the most common



membership functions are triangular, trapezoidal (as shown in the example), and Gaussian distributions.

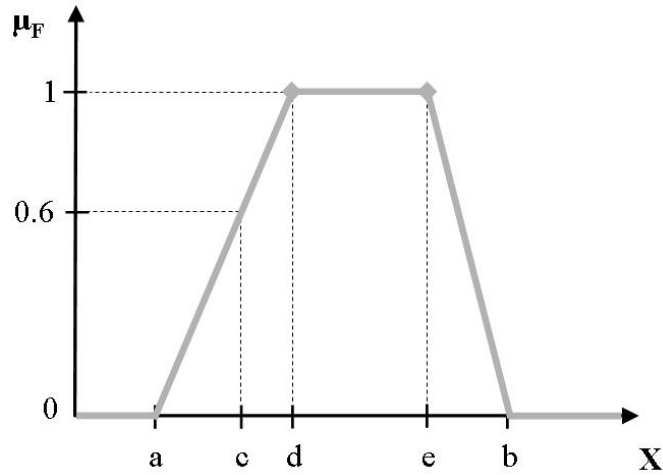


Fig. 16. Membership Function for Fuzzy Set  $F = [a, b]$

## 2. Fuzzy Rules

To model complex input-output relationships, fuzzy sets are often combined with IF-THEN rules. The rules are of the form, IF (X is A) THEN (Y is B), where the ‘IF’ portion of the rule is called the antecedent and the ‘THEN’ portion of the rule is called the consequent. The antecedent can also contain multiple clauses, which are combined with the logical operators AND (intersection) or OR (union). For fuzzy sets, the intersection and union operators are equivalent to the minimum and maximum of the membership functions as shown in Figures 17 and 18 [29].

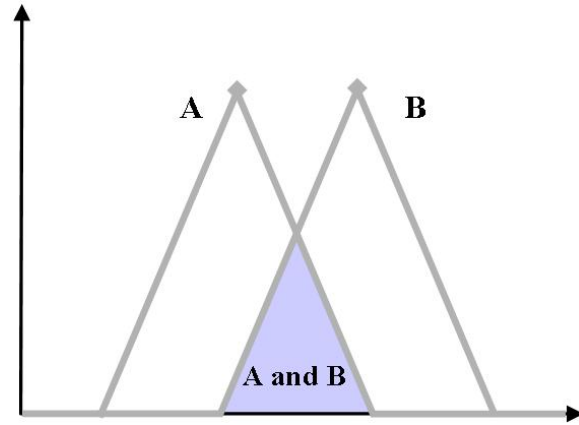


Fig. 17. Intersection Operator as Defined for Fuzzy Sets

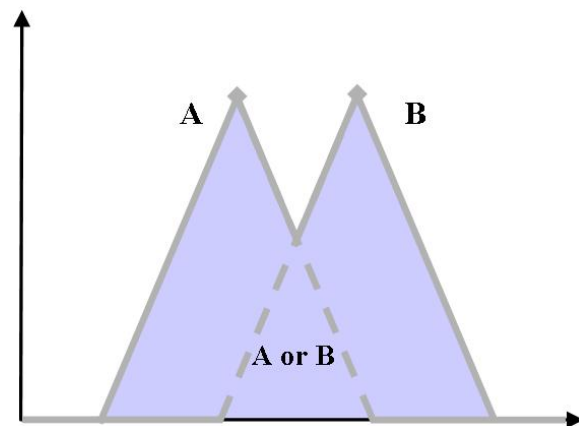


Fig. 18. Union Operator as Defined for Fuzzy Sets

### 3. Fuzzy Logic Input/Output Mapping Example

The example below demonstrates how FL is used to model input-output relationships. In the example, the input variable is the month of the year and the output variable is the expected temperature. This is accomplished by mapping the inputs to seasons and then using IF-THEN rules to relate the seasons to corresponding temperatures (using the terms ‘Cool’, ‘Warm’, and ‘HOT’). The model was created using the Fuzzy Logic toolbox, which is part of the Matlab application created by The MathWorks, Inc. [29].

The first step in creating the model is to select membership functions to represent the seasons. To show the flexibility in the FL technique, the model employs different types and sizes for the membership functions. The membership functions shown in Figure 19 are intended to represent a warm climate with an extended summer compared to the other seasons (as found in College Station, TX). The seasons of Spring, Fall, Winter are represented by triangular membership functions. The membership function for Winter is broken into two segments, which correspond to January through February (Winter-2) and October through December (Winter-1). The membership function for Summer is represented by a dual-sided Gaussian distribution.

The next step is to create a set of membership functions to represent the temperature using the terms Cool, Warm, and HOT, shown in Figure 20. The membership functions for Cool and Warm are represented with triangular membership functions, and the membership function for HOT is represented with a Gaussian distribution. Since this model represents a warm climate, the membership functions for temperature are biased toward high temperatures (in degrees Fahrenheit). The centroids of the membership functions are located at 55 degrees, 65 degrees, and 95 degrees for Cool, Warm, and HOT, respectively.

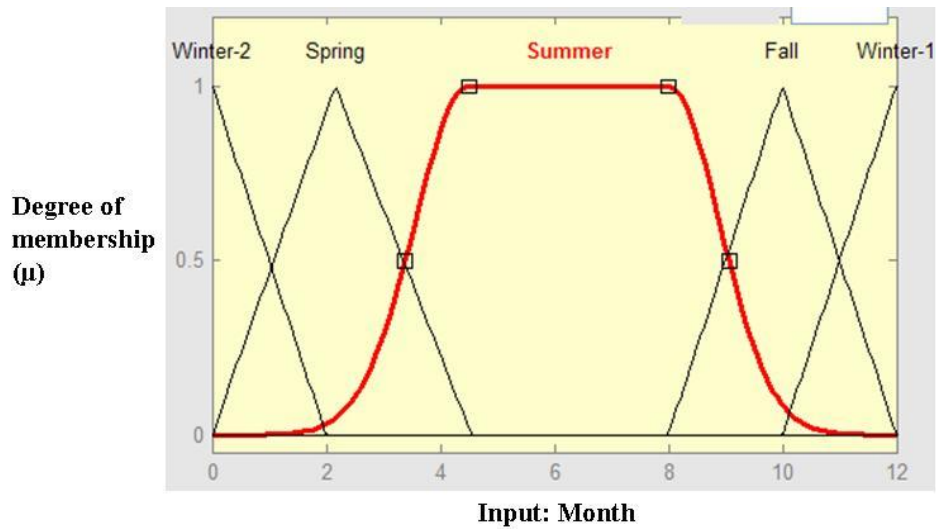


Fig. 19. FL Example, Membership Functions for the Seasons

A set of rules are created that relate the seasons to the temperature, which are captured in Table II. Capturing these rules should be simple since we are modeling the way humans understand the concept of seasons. In this case, the rules reflect that in Texas winter is cool, spring and fall are warm, and the summer is hot.

Table II. Rules for FL Season Based Temperature Model

| <b>IF</b> | <b>THEN</b>     |
|-----------|-----------------|
| Month is: | Temperature is: |
| Winter-1  | Cool            |
| Spring    | Warm            |
| Summer    | HOT             |
| Fall      | Warm            |
| Winter-2  | Cool            |

Once the membership functions and rules have been established, the input-output

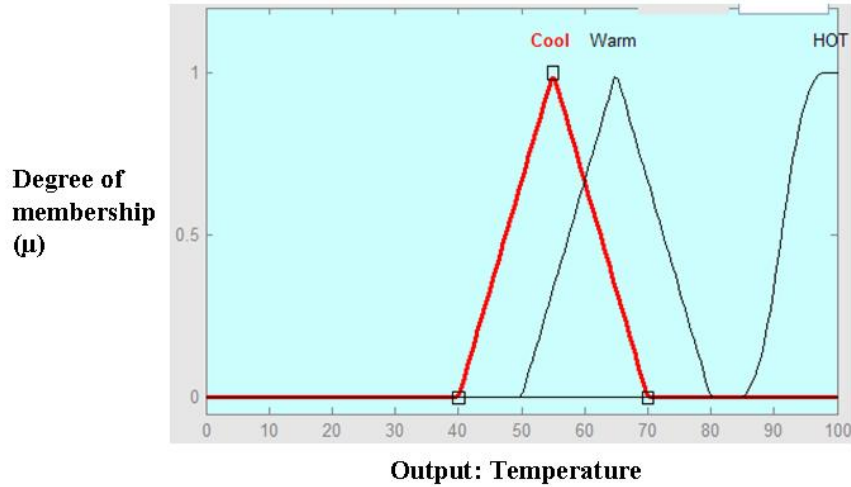


Fig. 20. FL Example, Membership Functions for Temperature

mapping is complete. Figures 21 and 22 are used to illustrate how the membership functions and rules that relate them result in an input-output mapping. In Figure 21, the input month is 4, which corresponds to the end of April. This input is evaluated to determine the degree of membership for each of the season membership functions. For an input of 4 there is partial membership in Spring and partial membership in Summer. In this case, the rules **IF** (Month is Spring) **THEN** (Temperature is Warm) and **IF** (Month is Summer) **THEN** (Temperature is HOT) are both active. These rules are then used to map the degree of membership in the season membership functions to equal degrees of membership in temperature membership functions (shown by the red lines in Figure 21). Next, the output membership functions are combined with an aggregation method, typically the union (max) operator is used (shown by the blue lines in Figure 21). Finally, this aggregated output membership function must be evaluated to produce a numerical output, in a process called ‘defuzzification’. The most common technique used for defuzzification is to take the centroid of the aggregated output membership function [29]. The centroid method is used in this

example and for all FL models created in this research. In Figure 21, the centroid is denoted by the thick red line in the center of the aggregated membership function. In this case, the centroid is located at 82.8 degrees Fahrenheit, which is the output corresponding to an input month of 4.

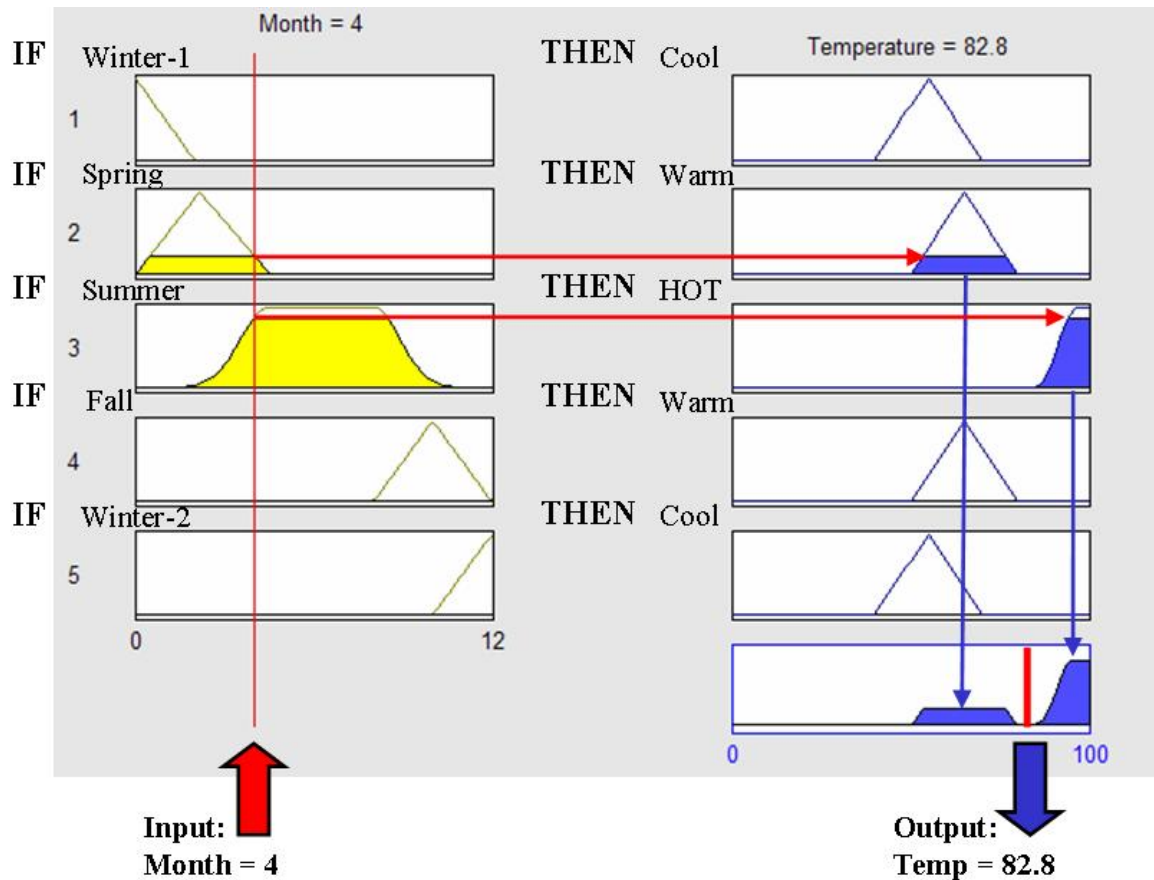


Fig. 21. FL Example, Seasons Model Output for April

Figure 22 is another example, this time for an input of 7, corresponding to the end of July. In this case, the input results in full membership in Summer and zero membership in all other seasons. Therefore, the only rule evaluated is **IF** (Month is Summer) **THEN** (Temperature is HOT). Since there is full membership in Summer, there is also full membership in the output membership function HOT. The

aggregated output membership function is simply equal to the membership function for HOT. The output is equal to the centroid, which has a value of 95.4 degrees Fahrenheit. This example illustrates that the region between input values of 4.5 and 8 will have a constant output of 95.4 degrees Fahrenheit, which is actually an accurate representation of Texas weather.

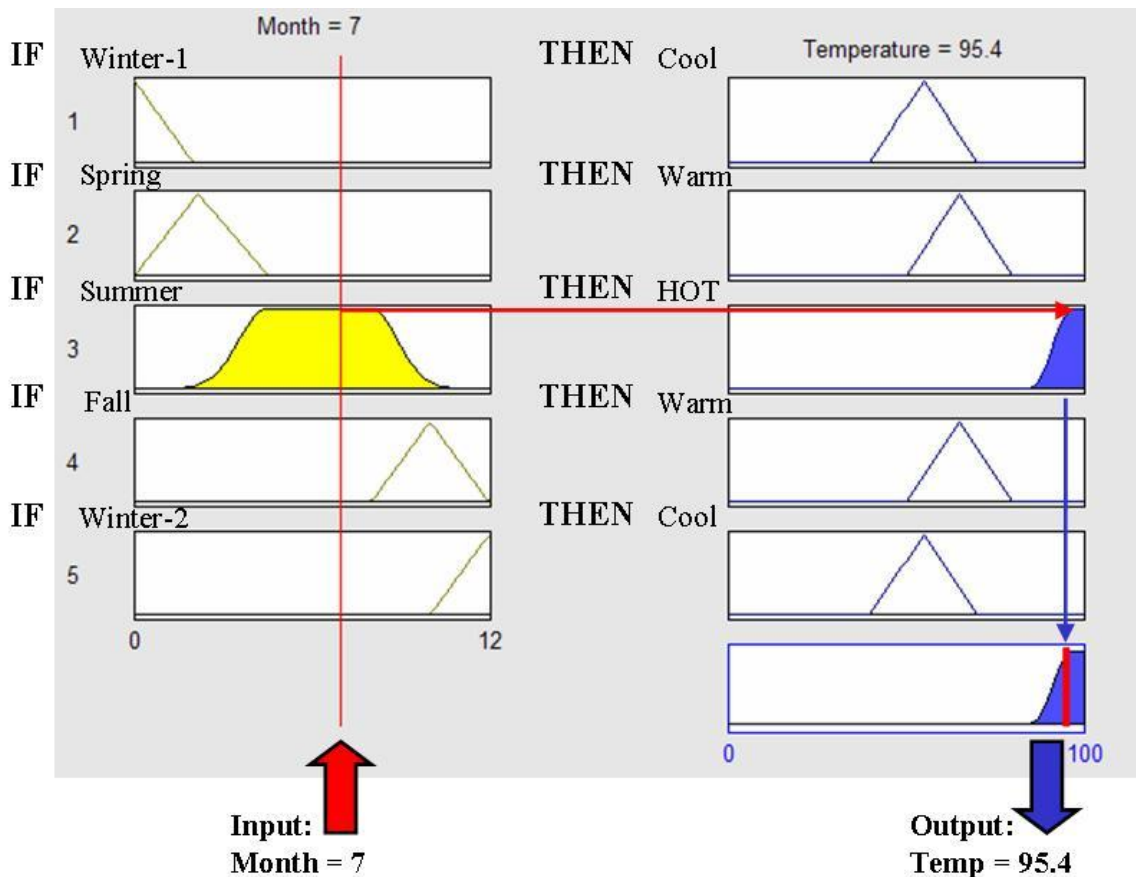


Fig. 22. FL Example, Seasons Model Output for July

The entire input-output mapping for this model is shown in Figure 23. As desired, the minimum temperature occurs around the turn of the year and the temperatures quickly increase and plateau for an extended hot summer.

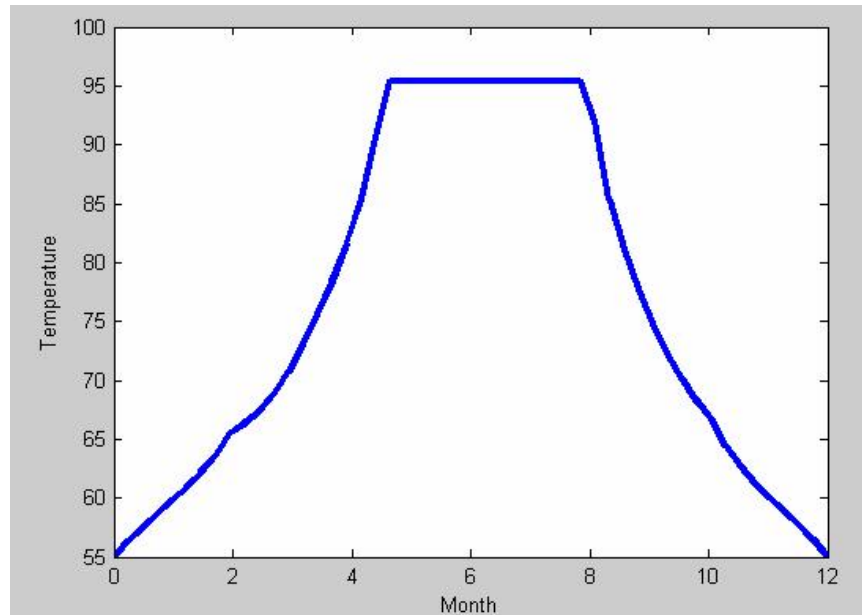


Fig. 23. FL Example, Seasons Model Input-Output Summary

## B. TIG-slip Planning

This section describes the FL models used to model the TIG-slip planning process. A brief description is provided that details the steps involved in the process. Then, the FL models used to complete the planning process are described in detail.

### 1. TIG-slip Problem

The TIG-slip planning problem involves the determination of the maximum amount of time the NC burn can be delayed (slipped) that will not result in a violation of a 4-nm relative position constraint (shown in Figure 9). This duration is computed for both the inertial and LVLH TIG-slip methods. The results using these two methods are then compared based on maximum TIG-slip duration, propellant costs, and a preference for the inertial TIG-slip method. The final output of the process is a recommended TIG-slip method (inertial or LVLH), maximum TIG-slip duration (in



seconds), and the additional propellant cost over the nominal burn ( $\Delta V$  in ft/sec). This process is described in detail in Chapter III, ‘Candidate Rendezvous Functions for Prototyping’.

## 2. Modeling TIG-slip Planning Using Fuzzy Logic

There are three separate FL models used to model the TIG-slip planning process. Two of the models are used to iteratively converge on the maximum TIG-slip duration; one is for the inertial TIG slip, and one is for the LVLH TIG slip. Both of these models are intended to emulate the way in which the human flight controller iteratively converges on the maximum TIG-slip duration. An initial guess is evaluated to determine how close the trajectory approaches the relative motion constraint. The distance from the constraint is provided as the input to the FL model. A small distance from the constraint will result in a small adjustment to the TIG-slip duration, and a large distance will result in a large adjustment. These models are run iteratively until they converge within a desired tolerance on duration. The third FL model is used to compare the two types of TIG slip based on the comparison of maximum TIG-slip duration and additional propellant cost. The output of this model is a recommendation of the TIG-slip method to execute. This model also includes a bias toward the inertial TIG-slip method, so if the TIG slip types are equal, the inertial TIG slip is recommended.

### a. Inertial TIG-slip Iteration Model

The input for the inertial TIG-slip iteration model is called the ‘position-offset’, in nautical miles. The position-offset is measured as the difference between the maximum relative motion between the NC and Ti points for the TIG slip trajectory and the 4-nm relative-motion constraint (equation 5.1).

$$\text{position-offset} = \text{max relative position} + 4 \text{ nm} \quad (5.1)$$

This distance is shown for a 30-second inertial TIG slip in Figure 24 and a 180-second inertial TIG slip in Figure 25. The input values to the TIG slips are -2.7 nm for the 30 second inertial TIG slip and 4.8 nm for the 180 seconds inertial TIG slip. It is clear that negative values of position-offset correspond to trajectories that do not reach the constraint and thus can be additionally slipped. Positive values of position-offset have exceeded the constraint, and the TIG-slip duration must be reduced to satisfy the constraint.

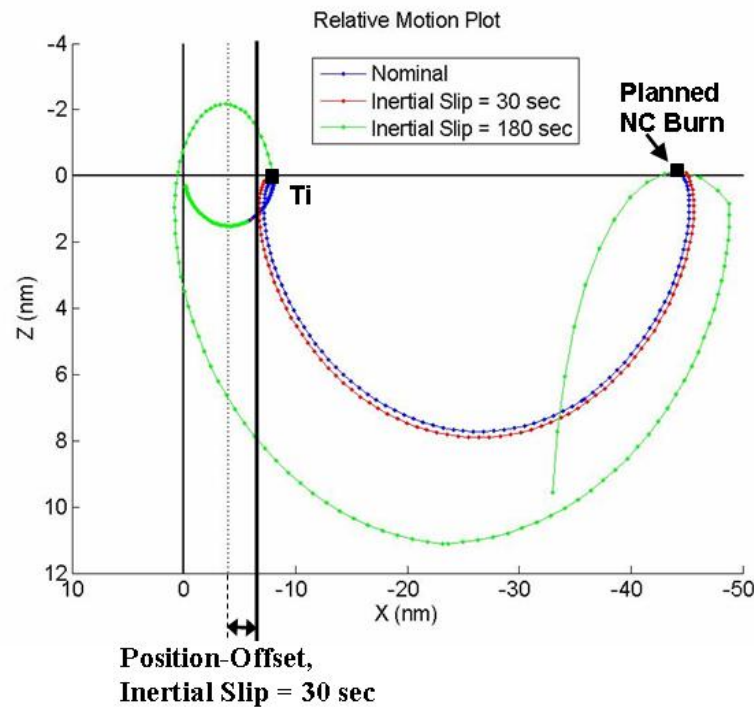


Fig. 24. Inertial TIG slip of 30 seconds, Position-Offset

The input membership functions of the inertial TIG-slip iteration model describe the position offset using the terms 'Negative-Large', 'Negative-Small', 'Zero',

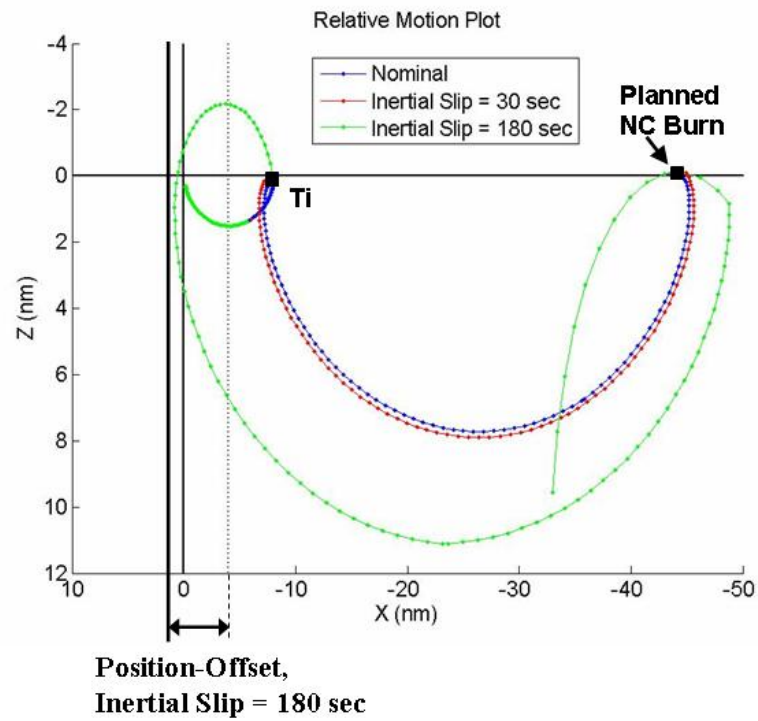


Fig. 25. Inertial TIG slip of 180 seconds, Position-Offset

‘Positive-Small’, and ‘Positive-Large’. The membership functions used to describe these linguistic terms are shown in Figure 26. The input range is limited to  $\pm 6$  nm to reflect a reasonable range of position-offset values. The membership functions for both the inputs and outputs were shaped by an empirical process to result in a successful iteration process for the inertial TIG-slip method.

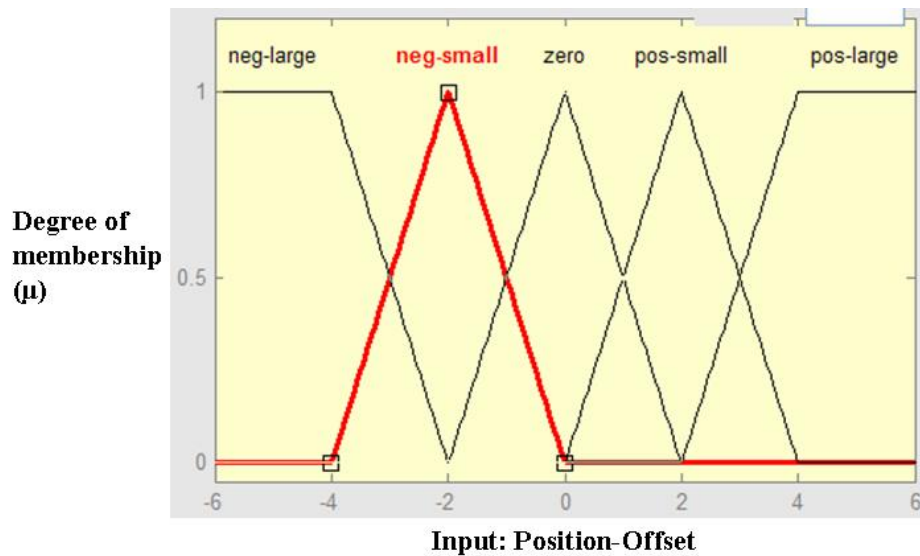


Fig. 26. Inertial TIG-slip Iteration Model, Input Membership Functions

The output of the inertial TIG-slip iteration model is a ‘time-delta’ from the TIG-slip duration for the previous iteration. The output range is between  $\pm 80$  seconds. The output membership functions use the same linguistic terms as the input membership functions, ‘Negative-Large’, ‘Negative-Small’, ‘Zero’, ‘Positive-Small’, and ‘Positive-Large’. The membership functions used to describe these linguistic terms are shown in Figure 27.

The rules for the inertial TIG-slip iteration model are described in Table III. Since a negative position-offset allows an increase in the duration of the TIG slip, as

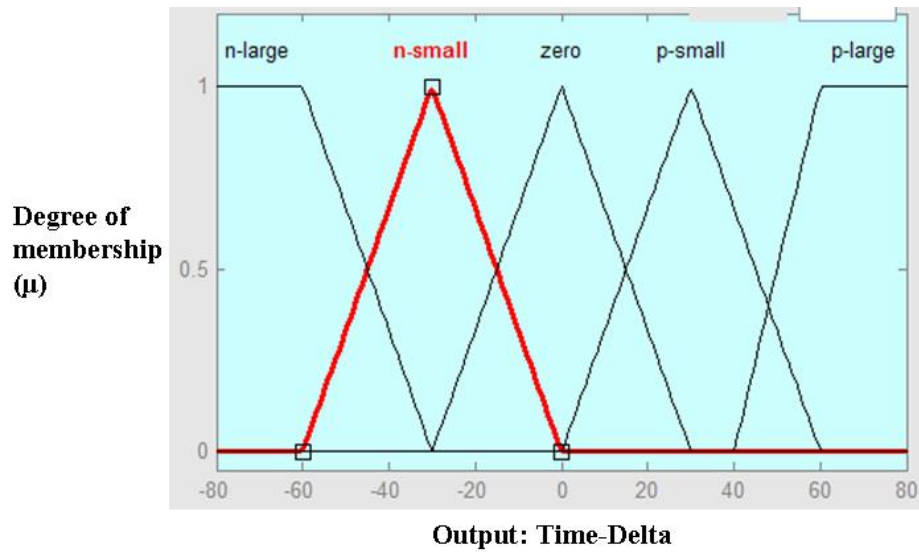


Fig. 27. Inertial TIG-slip Iteration Model, Output Membership Functions

described above, a negative position offset should result in a positive time-delta and vice versa. The rules reflect this relationship and result in small time changes for small offsets and large time changes for large offsets.

The complete model consists of the input and output membership functions and the rules. The result is an input-output mapping used to modify the TIG-slip duration by the time-delta output. Figure 28 shows the relationship between a current position-offset and the resulting time-delta used in the iteration process. The relationship is approximately linear between position-offsets of  $\pm 4$  nm, with smaller position-offsets resulting in smaller time-deltas, as desired. For inputs that exceed  $\pm 4$  nm, the output is a constant maximum time-delta of  $\pm 65$  seconds. This is the upper limit of reasonable time-deltas for the inertial TIG slips based upon empirical analysis. This input-output mapping accurately reflects the desired behavior during the iteration process.

Table III. Rules for TIG-slip Iteration Models

| <b>IF</b>           | <b>THEN</b>    |
|---------------------|----------------|
| Position-Offset is: | Time-Delta is: |
| Negative-Large      | Positive-Large |
| Negative-Small      | Positive-Small |
| Zero                | Zero           |
| Positive-Small      | Negative-Small |
| Positive-Large      | Negative-Large |

b. LVLH TIG-slip Iteration Model

The LVLH TIG slip and inertial TIG slip iteration models are identical except for differences in the output membership functions. The reason for this difference is that the dynamics of the LVLH TIG slip result in larger changes in relative motion than for an identical inertial TIG-slip case. This difference is evident in Figures 9 and 10, which show large relative motion differences between the inertial and LVLH TIG slips of identical durations. As a result, the output membership functions should reflect a smaller time-delta for a given position-offset than the inertial TIG-slip case.

The LVLH TIG-slip iteration model uses an identical input variable, position-offset, and membership functions as shown in Figure 26. The rules are also identical to the inertial TIG-slip iteration model, shown in Table III. The output membership functions use the same linguistic terms as the input membership functions, ‘Negative-Large’, ‘Negative-Small’, ‘Zero’, ‘Positive-Small’, and ‘Positive-Large’. However, the output membership functions, shown in Figure 29, have different shapes to produce the desired input-output mapping.

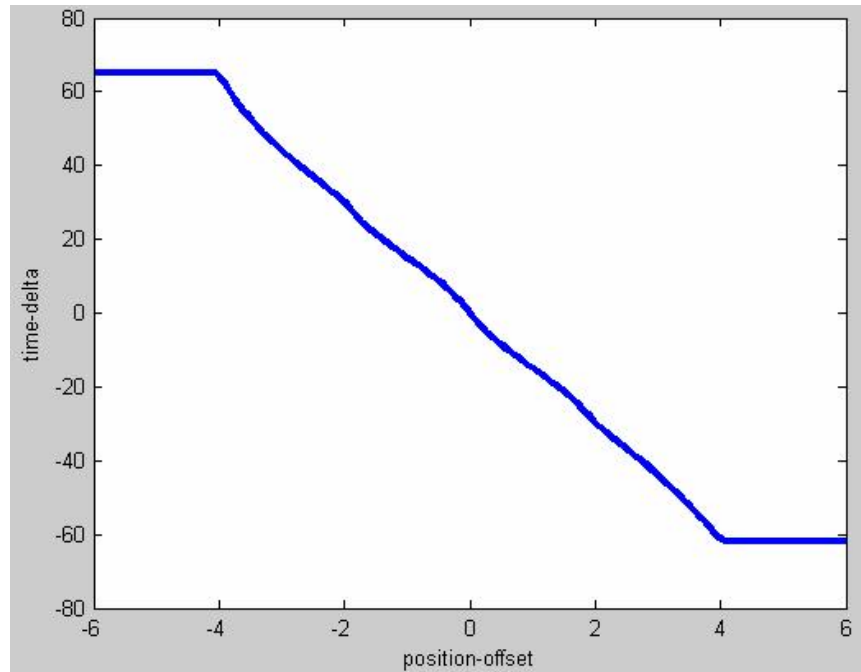


Fig. 28. Inertial TIG-slip Iteration Model, Input-Output Mapping

The output membership functions are designed to prevent outputs that result in a violation of the relative motion constraint and will produce outputs that return trajectories to the acceptable side of the position constraint if a violation has occurred. This is accomplished by having small membership functions for positive time-deltas, which correspond to cases that are on the acceptable side of the relative motion constraint (negative position-offsets). There are also large membership functions for negative time-deltas, which correspond to violations of the relative motion constraint (positive position-offsets).

The resulting input-output mapping is shown in Figure 30. As desired, there is a gradual slope in time-delta for position-offsets with slightly negative values. This prevents cases that have not reached the relative-position constraint from exceeding the constraint. There is also a larger slope in time-delta for positive position-offsets,

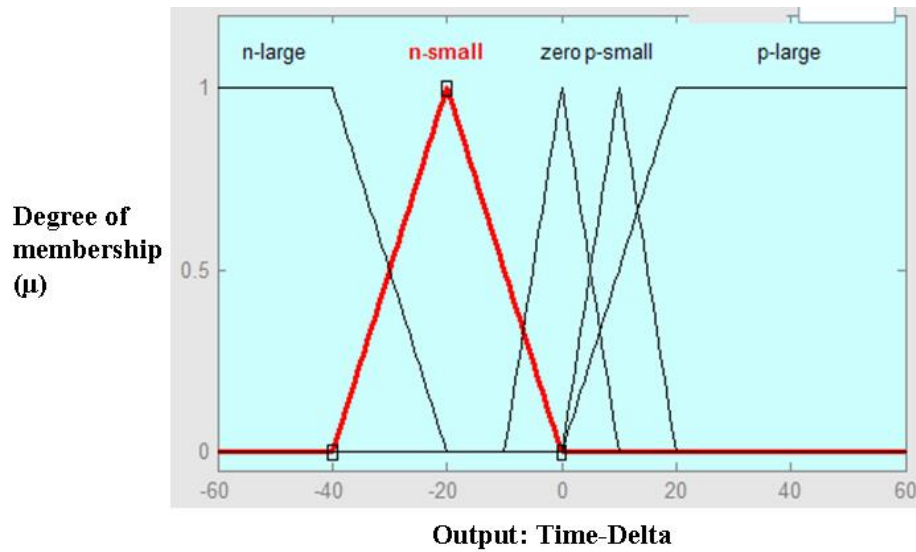


Fig. 29. LVLH TIG-slip Iteration Model, Output Membership Functions

which causes cases that exceed the position constraint to return to the acceptable side of the constraint in the next iteration. This also prevents the iteration process from jumping back-and-forth over the boundary. Results of the iteration process for both inertial and LVLH TIG-slip cases can be found in the ‘Experiment Results’ chapter.

### c. TIG-slip Comparison Model

Once the iteration process is complete for both TIG-slip methods, the results are compared using the TIG-slip comparison model described in this section. The output of this model is a value in the continuous interval  $[-1,+1]$ , with  $-1$  corresponding to a strong recommendation for the inertial TIG-slip method and  $+1$  corresponding to a strong recommendation for the LVLH TIG-slip method. This output is simply called ‘inertial-LVLH’. The inputs to this model are the difference in additional propellant and the difference in the maximum TIG-slip duration between the two methods. The input variables are called ‘propellant-difference’ and ‘time-difference’, respectively,



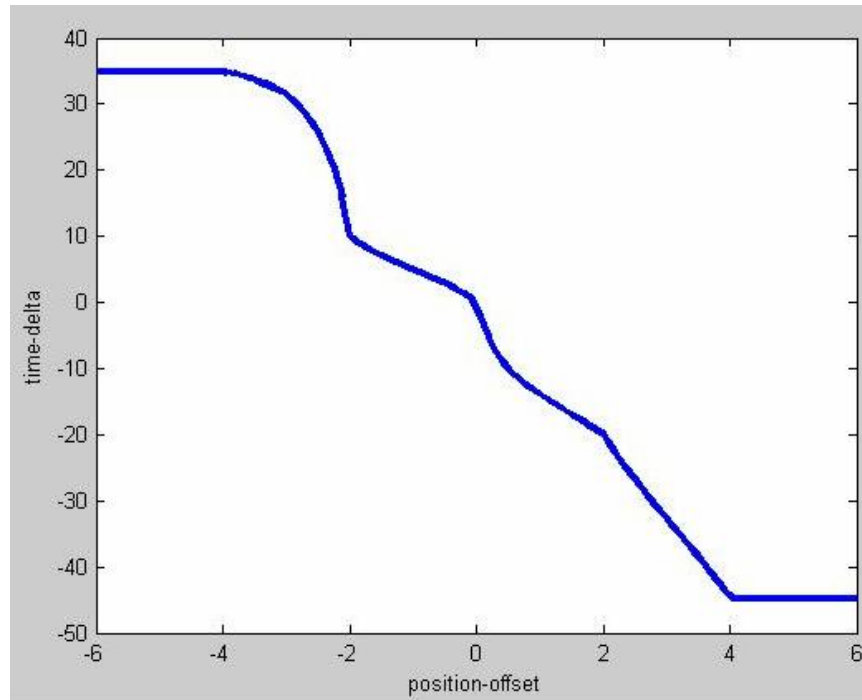


Fig. 30. LVLH TIG-slip Iteration Model, Input-Output Mapping

which are defined in equations 5.2 and 5.3.

$$\text{propellant-difference} = \text{propellant}_{LVLH} - \text{propellant}_{inertial} \quad (5.2)$$

$$\text{time-difference} = \text{time}_{LVLH} - \text{time}_{inertial} \quad (5.3)$$

A positive value of propellant-difference corresponds to a lower propellant cost for the inertial TIG-slip method, and a positive time-difference corresponds to a shorter maximum TIG-slip duration for the inertial TIG slip method.

The propellant-difference is modeled using membership functions with the linguistic terms ‘inertial-big-increase’, ‘inertial-small-increase’, ‘equal’, ‘LVLH-small-increase’, and ‘LVLH-big-increase’. These membership functions are shown in Figure 31. The input range of  $\pm 20$  ft/sec and the shape of the membership functions reflect

a reasonable range of propellant differences determined by evaluating dispersed TIG-slip cases. For these cases, a small difference is considered to be between 0 and 10 ft/sec and a large difference is between 10 and 20 ft/sec.

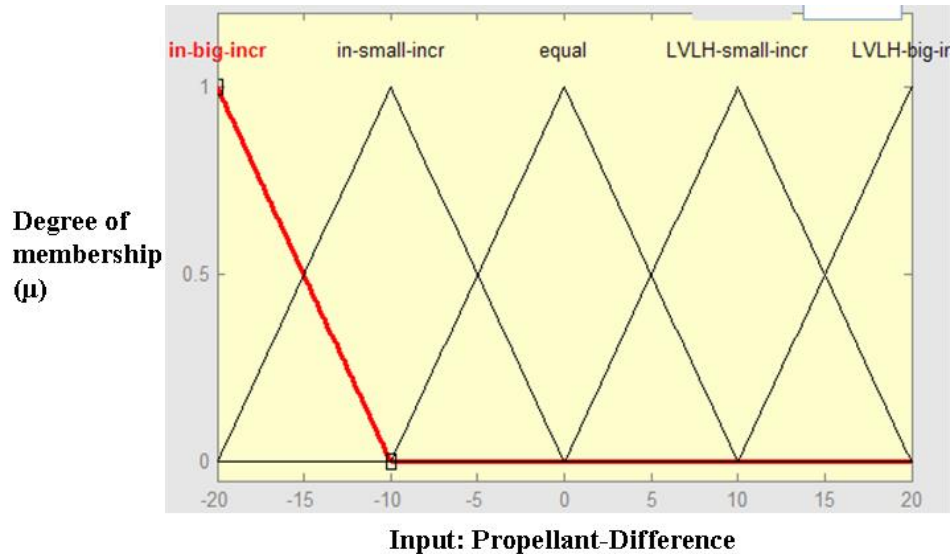


Fig. 31. TIG-slip Comparison, Propellant Input Membership Functions

The time-difference is modeled using membership functions with the linguistic terms ‘inertial-more-time’ and ‘LVLH-more-time’. The input range for time-difference is  $\pm 240$  seconds (4 minutes). The membership functions for the time-difference are used to model the preference for the inertial TIG slip method since it is easier for the Shuttle crew to execute. This preference is built into the model by creating membership functions that are unequally balanced to favor the inertial TIG slip. In Figure 32, the membership function for LVLH-more-time does not start until an input value of 0 seconds and it does not cross the inertial-more-time membership function until a time-difference of 120 seconds.

The output membership functions for the the TIG-slip comparison model are

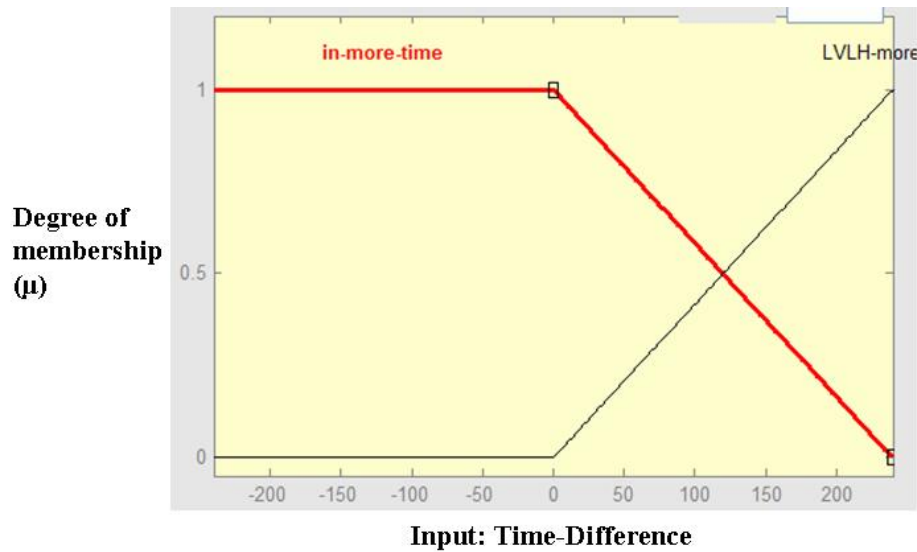


Fig. 32. TIG-slip Comparison, Time Input Membership Functions

shown in Figure 33. These membership functions use the linguistic terms ‘inertial-preference’, ‘inertial-slight-preference’, ‘equal’, ‘LVLH-slight-preference’, and ‘LVLH-preference’. The ‘inertial-preference’ and ‘LVLH-preference’ membership functions are designed to have a centroids at -1 and +1, respectively. This limits the output space to  $\pm 1$  and will result in an output of -1 for inertial TIG slip preference and +1 for LVLH TIG slip preference. The membership functions that refer to a slight preference are sized to allow for small adjustments to the final result, where appropriate.

The rules used to provide the recommended TIG-slip method are captured in Table IV. These rules are intended to reflect how a human flight controller would compare the TIG-slip methods based on the difference in duration and additional propellant cost. For example, if the LVLH method results a big increase in propellant cost and the inertial method has a longer TIG-slip duration, the inertial TIG slip would be strongly recommended. The remainder of the rules capture the relative preference for all combinations of inputs.

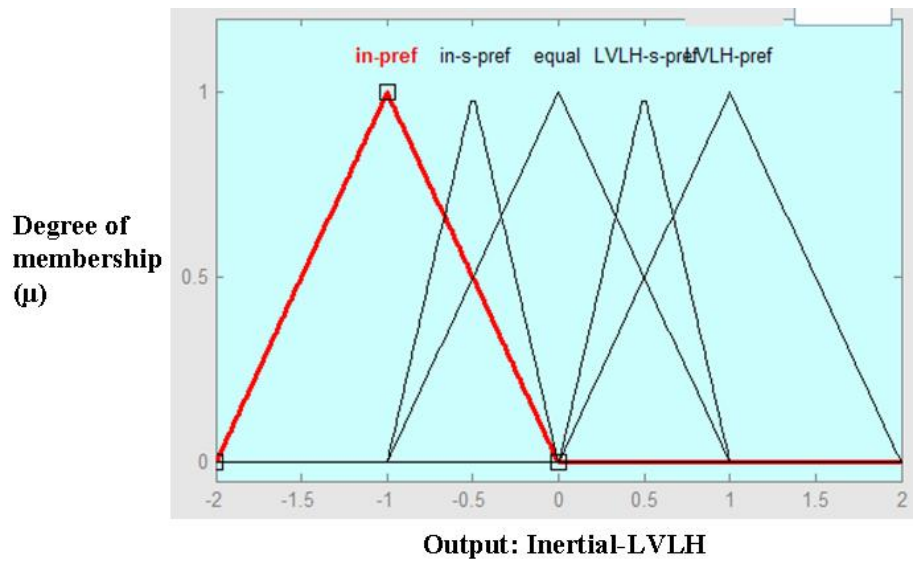


Fig. 33. TIG-slip Comparison, Propellant Output Membership Functions

Table IV. Rules for TIG-slip Comparison Model

| IF                        | AND                 | THEN                       |
|---------------------------|---------------------|----------------------------|
| Propellant-Difference is: | Time-Difference is: | Inertial-LVLH is:          |
| inertial-big-increase     | inertial-more-time  | LVLH-slight-preference     |
| inertial-big-increase     | LVLH-more-time      | LVLH-preference            |
| inertial-small-increase   | inertial-more-time  | equal                      |
| inertial-small-increase   | LVLH-more-time      | LVLH-slight-preference     |
| equal                     | inertial-more-time  | inertial-slight-preference |
| equal                     | LVLH-more-time      | LVLH-slight-preference     |
| LVLH-small-increase       | inertial-more-time  | inertial-slight-preference |
| LVLH-small-increase       | LVLH-more-time      | equal                      |
| LVLH-big-increase         | inertial-more-time  | inertial-preference        |
| LVLH-big-increase         | LVLH-more-time      | inertial-slight-preference |

The complete input-output mapping for the TIG-slip comparison model is shown in Figure 34. This model provides a continuous output surface over the entire input space. As expected, the surface has a minimum (maximum inertial preference) for a large propellant-difference (+20 ft/sec) and a large negative time-difference (-240 seconds, i.e., the inertial method provides 4 minutes of additional TIG-slip capability over the LVLH method). The maximum output value (maximum LVLH preference) corresponds to a large negative propellant-difference (-20 ft/sec) and a large time-difference (+240, i.e., the LVLH method provides 4 minutes of additional TIG-slip capability over the inertial method). The most interesting points of this input-output mapping are output values of zero, which denote the dividing line between recommending inertial and LVLH TIG-slip methods. The dashed line in Figure 35 shows the boundary between these recommendations. For cases with a longer allowable inertial TIG-slip duration (negative values of time-difference), the additional propellant cost of an inertial over an LVLH slip must exceed 10 ft/sec (-10 ft/sec propellant-difference) before an LVLH TIG slip is recommended. Another interesting feature is that the dividing line for equal propellant costs occurs at 120 seconds. These results confirm that the model reflects the desired bias toward inertial TIG slips.

### C. Evaluation of Vehicle Position at $T_i$

This section describes the FL model used to evaluate the vehicle position at the location of the  $T_i$  burn relative to position constraints. There is a brief description of the details of the process used to make this evaluation. Then, the FL model used for this prototype is described in detail. The description of the FL model includes a discussion for each input to the model individually and then the complete FL model.

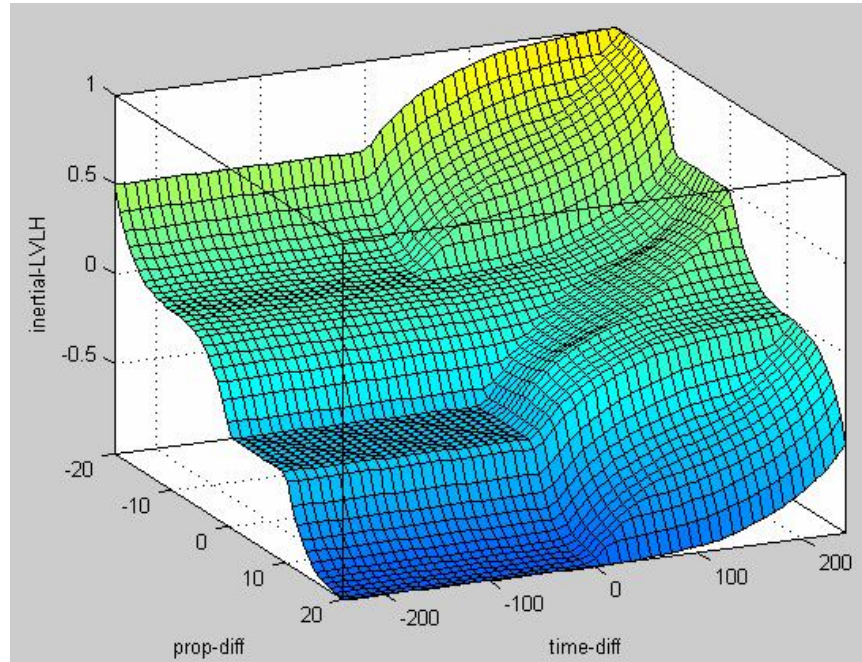


Fig. 34. TIG-slip Comparison Model, Input-Output Mapping

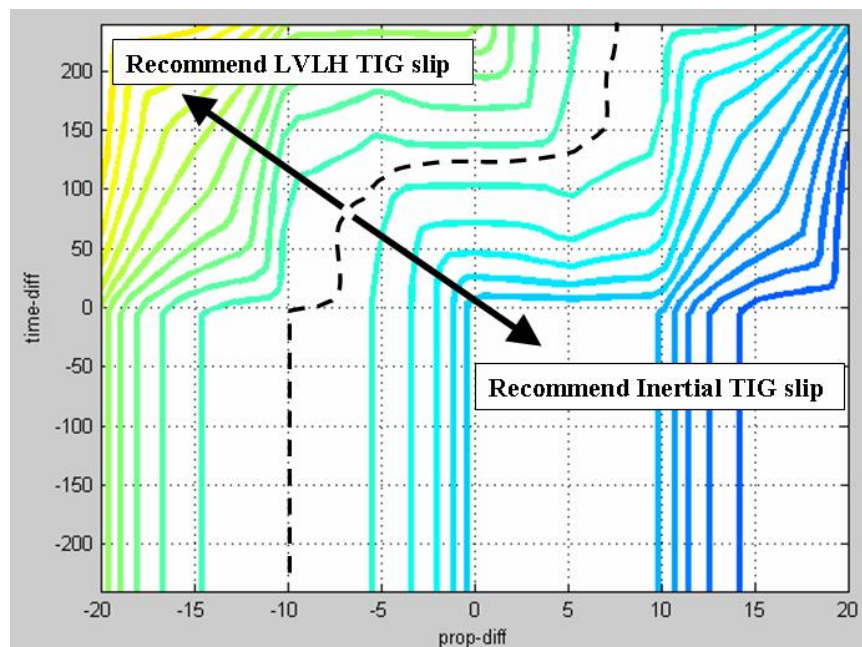


Fig. 35. TIG-slip Comparison Model, Input-Output Mapping (Contour Plot)

## 1. Ti Position Constraints

The Ti position constraints are used to ensure safe relative motion between the target and chaser spacecraft and to provide desirable lighting conditions during the rendezvous trajectory. These constraints are used to determine when to execute a burn to delay the Ti maneuver (or a correction burn during a Ti-delay trajectory) based on position constraints and current navigation accuracy.

The FL model must be able to assess the ‘severity’ of the predicted Ti position relative to  $3\text{-}\sigma$  position-dispersion bounds and the lighting constraint at the MC-2 burn location. In addition to the position constraints, the FL model must include the current navigation accuracy. When navigation inaccuracies are present, the model should produce a higher ‘severity’ value for a given relative position. Figure 11 shows an example nominal trajectory and the position constraints. The final output of this prototype is a ‘severity’ value that reflects the priority that should be assigned to executing a Ti-delay burn or a correction to an existing Ti-delay trajectory. This severity value can be used to assist the human flight controller or crew member in determining if a burn should be executed. This process is described in detail in Chapter III, Candidate Rendezvous Functions for Prototyping.

## 2. Modeling Ti Position Constraints Using Fuzzy Logic

The prototype for the evaluation of vehicle position at Ti consists of a FL model with three inputs. The inputs to the model represent the location relative to the elliptical  $3\text{-}\sigma$  position constraint, the position relative to the lighting constraint, and the current navigation accuracy. These inputs are then compared to their respective membership functions and the rules that govern the model to produce the ‘severity’ output. This model is described in detail below.

a. Modeling the  $3\text{-}\sigma$  Position Constraint

The first step in creating the Ti position constraint model is to evaluate position relative to the elliptical  $3\text{-}\sigma$  position constraint. This constraint is shown, along with the nominal trajectory in Figure 36. To evaluate the position relative to the center of an elliptical constraint, the term “assessment number” is defined. The assessment number for an ellipse is defined as the ratio of the radial distance to the point of interest (defined as the length ‘A’) and the radial distance to the edge of the ellipse along the same line (defined as ‘B’). This is shown graphically in Figure 37. It follows that points within the elliptical constraint have assessment numbers less than 1 and points outside will have assessment numbers greater than 1.

The procedure for computing the assessment number for a point of interest is straightforward. First, the coordinates of the location at Ti are rotated and translated into the ellipse relative frame that is aligned with the semi-major and semi-minor axes of the ellipse, shown in Figure 37. Next, the slope from the ellipse center to the point of interest  $(x_p, y_p)$  is calculated. Then, the point on the ellipse along that line  $(x_e, y_e)$  is computed using the equation of the ellipse and the slope of  $(x_p, y_p)$ . Finally, the lengths of vectors A and B are determined, and the assessment number is computed as the ratio A/B.

The ellipse assessment number (‘ellipse-assessment’) is an input of the FL Ti constraint model. The membership functions for this input are shown in Figure 38, which represent the linguistic terms ‘good’, ‘fair’, and ‘poor’. The membership function for ‘good’ assessment numbers represents values that are well within the elliptical position constraint. These values range from full membership for an ellipse assessment of 0.0 to zero membership for assessment values greater than 0.7. For the ‘fair’ membership function, the values represent points that are moderately within the



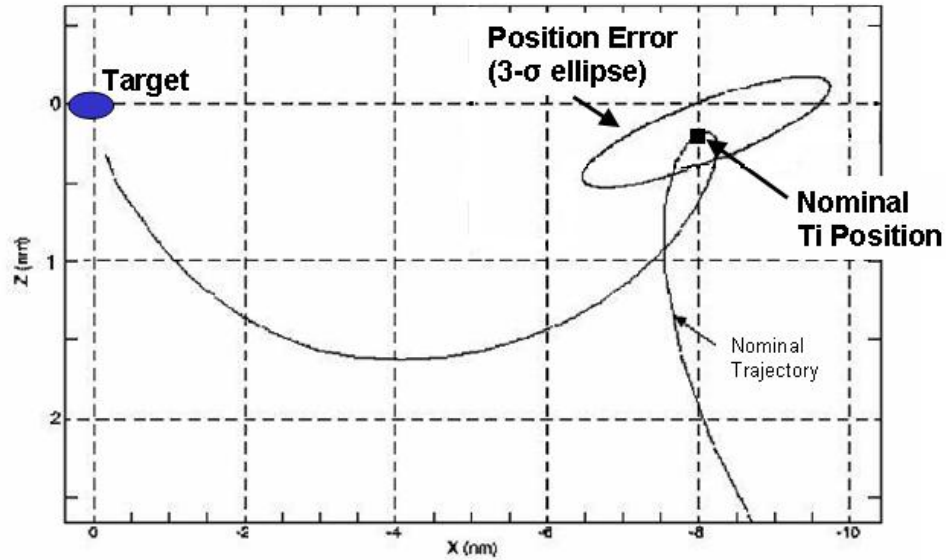
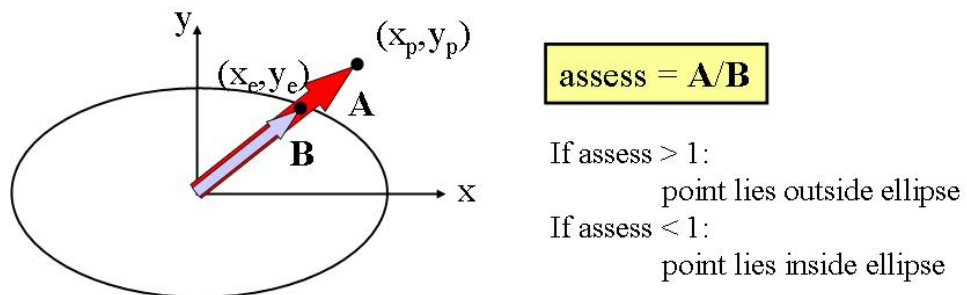
Fig. 36. Ti 3- $\sigma$  Position Constraint

Fig. 37. Assessment Number for an Elliptical Constraint

constraint or just outside the constraint. The membership function for ‘fair’ ranges between 0.3 and 1.1, with a maximum value for an ellipse-assessment at 0.7. The ‘poor’ region includes points that are just within the constraint boundary or outside of the boundary. The membership function for ‘poor’ begins at an ellipse-assessment of 0.7 and has full membership for values greater than or equal to 1.2.

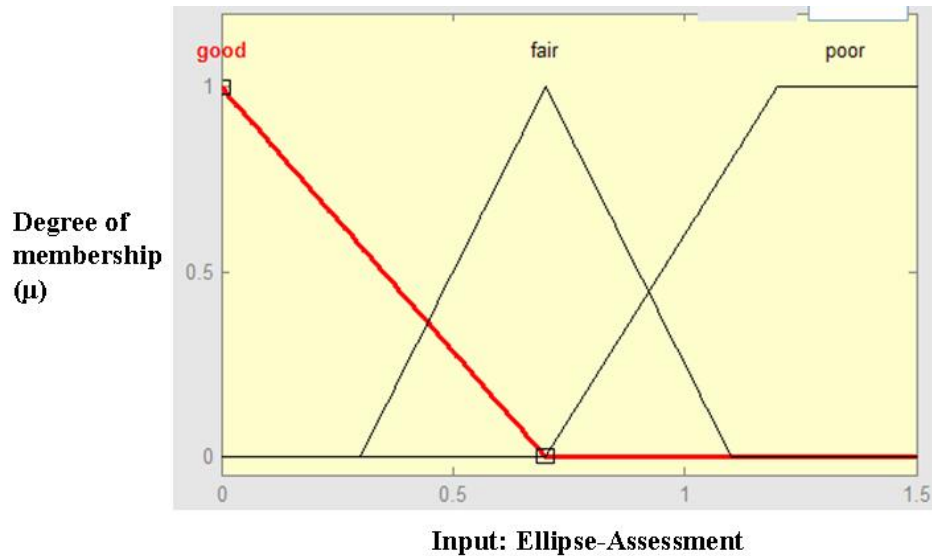


Fig. 38. Ellipse-Assessment Input Membership Functions

The output membership functions, shown in Figure 39, are used for all inputs to the Ti constraint model. The severity of the current position relative to constraint and navigation quality are captured using the linguistic terms ‘minor’, ‘moderate’, and ‘severe’ for the output membership functions. These membership functions are sized to produce outputs that represent severity values between 1 and 10 that are considered ‘minor’, ‘moderate’, and ‘severe’ when combined with the FL model inputs and rules.

The rules for the Ti 3- $\sigma$  position constraint, without considering the other inputs, are captured in Table V. When these rules are applied, the result is the input-output

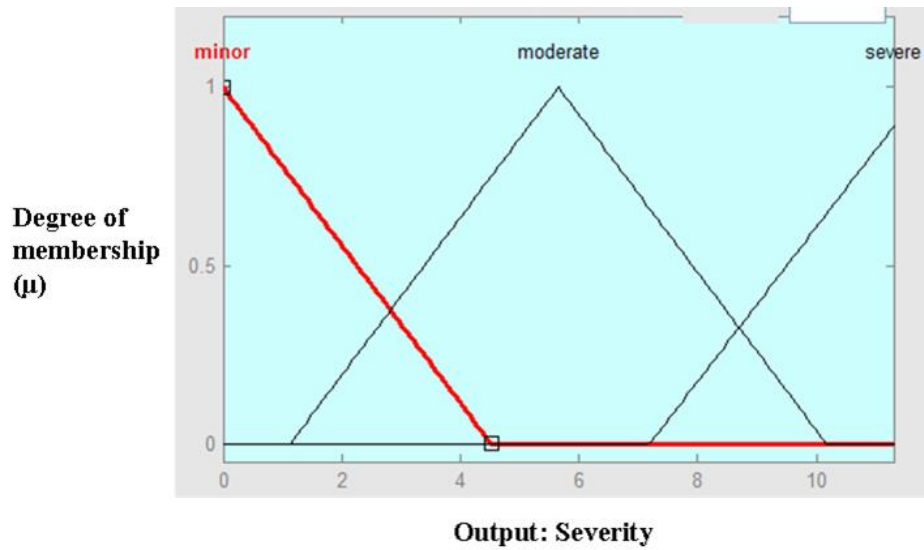
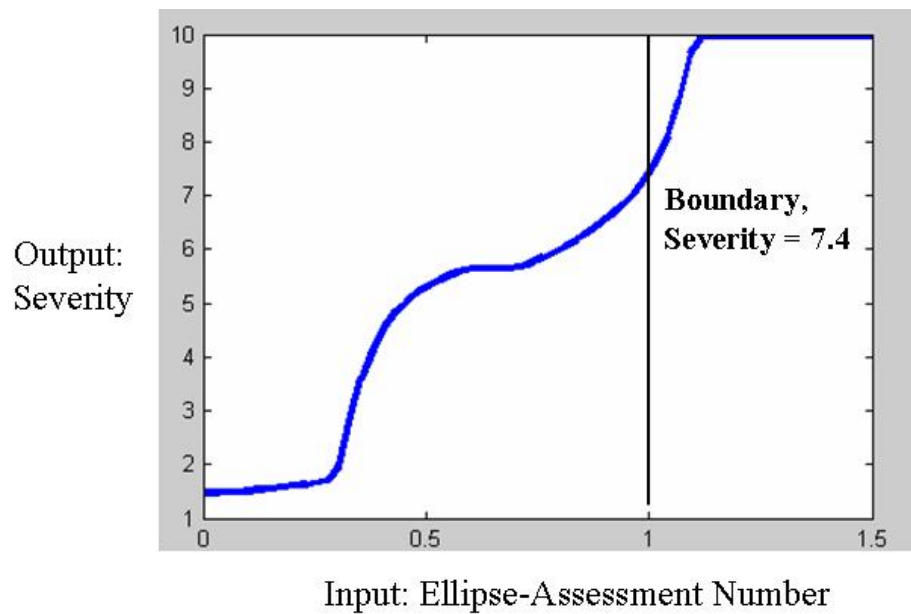


Fig. 39. Ti Position-Constraint Model, Output Membership Functions

mapping shown in Figure 40. For these input membership functions, inputs located solely in the ‘good’ region have low values of severity (1-2). When the ‘fair’ region of inputs is reached, the severity increases and then levels off with output values between 5 and 6. Finally, when the values in the ‘poor’ region are reached, the output increases to the maximum value of 10. The boundary of the  $3\text{-}\sigma$  position constraint corresponds a ‘severity’ output value of 7.4. Notice that the output value at the constraint boundary is not equal to the maximum severity output (10). This indicates that for this constraint it is acceptable to have cases that are just outside of the boundary. As expected, values that exceed an ellipse-assessment of 1.2 have full membership in the ‘poor’ region resulting in the maximum severity output value of 10. An example input-output mapping for a sample ellipse is shown in Figure 41, with the green line indicating the constraint boundary.

Table V. Rules for Ti 3- $\sigma$  Position Constraint

| <b>IF</b>              | <b>THEN</b>  |
|------------------------|--------------|
| Ellipse-Assessment is: | Severity is: |
| Good                   | Minor        |
| Fair                   | Moderate     |
| Poor                   | Severe       |

Fig. 40. Ti 3- $\sigma$  Position Constraint, Input-Output Mapping

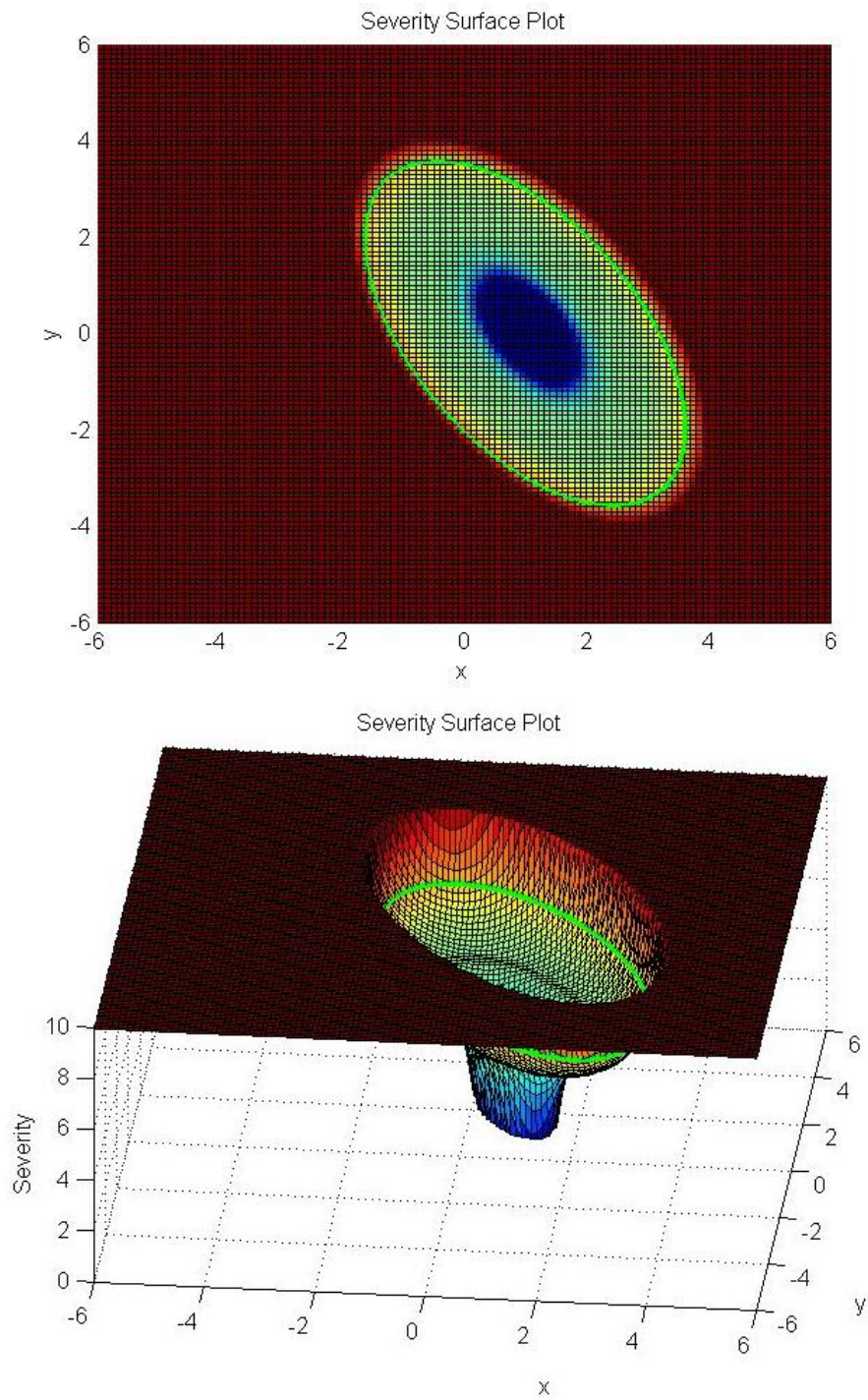


Fig. 41. Severity Output Surface for an Example Elliptical Constraint

### b. Modeling the MC-2 Lighting Constraint

The next step in creating the FL model of Ti position constraints is modeling the lighting constraint as shown in Figure 42. This constraint consists of several line segments that define the boundary of the constraint. Relative position values within this region result in acceptable lighting conditions when the MC-2 burn location is reached. The position with respect to this boundary is evaluated by using an “assessment number” that is defined similarly to the ellipse-assessment described previously. The ‘MC2-assessment’ number is defined as the distance to the closest point on the MC-2 constraint boundary, i.e., the shortest perpendicular distance from the boundary to the point of interest. The value of the MC2-assessment is equal to the distance from the constraint boundary in units of nm, with points within the boundary assigned negative numbers and points outside of the boundary assigned positive numbers. It follows that points along the boundary have an MC2-assessment of 0.0.

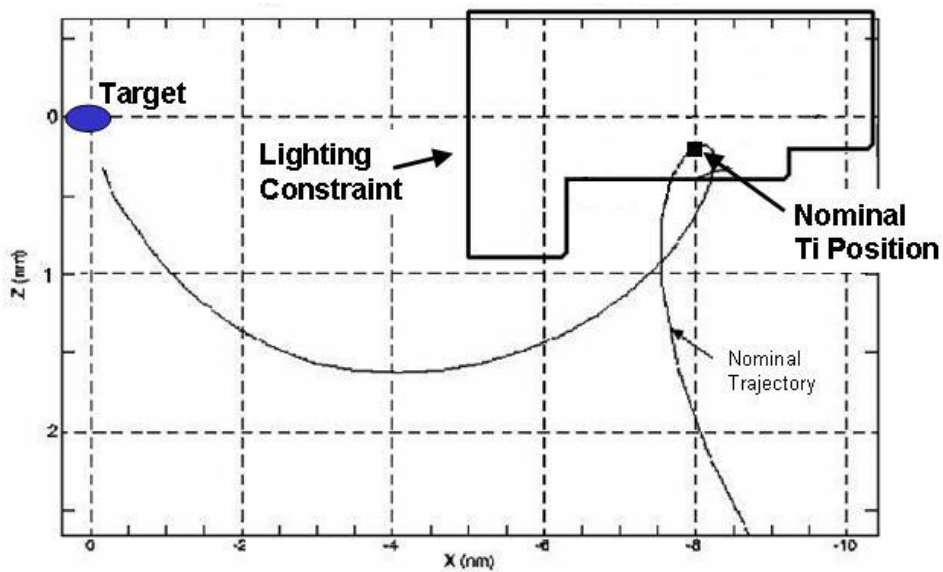


Fig. 42. Ti Position and Lighting Constraints

The MC-2 constraint boundary is modeled by assigning vertices at locations that are along the boundary of the constraint. An example of this is shown in Figure 43 for a simplified version of a linear constraint boundary with six vertices. A ‘type’ is also assigned to these vertices, which defines the quadrants relative to the vertex that are within the constraint boundary. In this example, for vertex 1, quadrant I is within the boundary and for vertex 5, quadrants I, II, and III are within the boundary, etc.

The calculation of the MC2-assessment is accomplished using the following procedure. First, the distance from the point of interest  $(x_p, y_p)$  to each vertex is computed to determine the closest vertex (see Figure 43). Then, the quadrant relative to the closest vertex is determined for the point of interest. The quadrant and vertex number are used to determine whether the point is inside or outside of the constraint boundary, based on the vertex ‘type’. The quadrant information is also used to determine the closest distance to the boundary. Finally, the proximity to the boundary and whether the point is inside or outside the boundary determines the ‘MC2-assessment number’. This process is used to determine the MC2-assessment number for points in the vicinity of the lighting constraint shown in Figure 42. The output of the process to determine the MC2-assessment number, denoted as “Boundary Distance”, is shown in Figure 44.

The input to the FL model for the MC-2 lighting constraint is the MC2-assessment number. The input membership functions for MC2-assessment are shown in Figure 45, which represent the linguistic terms ‘good’, ‘fair’, and ‘poor’. The membership function for ‘good’ assessment numbers represents values that are well within the lighting constraint. These values range from full membership for an MC2-assessment of less than -0.3 to zero membership for assessment values greater than -0.1. For the ‘fair’ membership function, the values represent points that are moderately within the



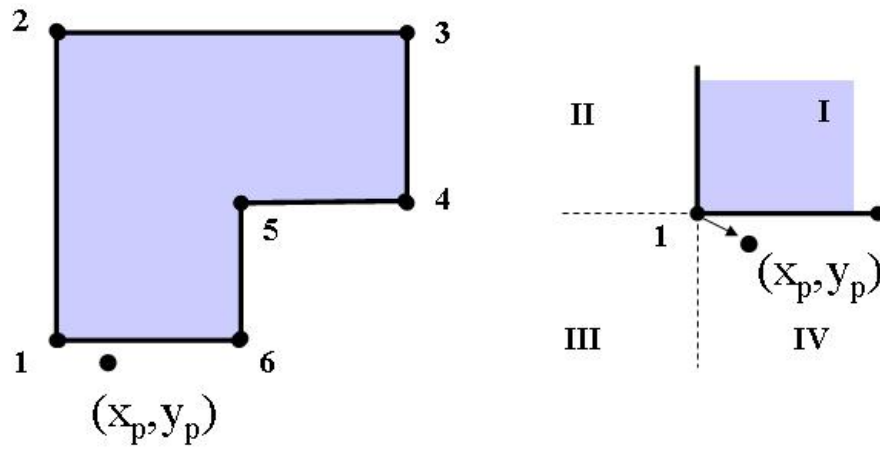


Fig. 43. Determine MC-2 Lighting Constraint Boundary

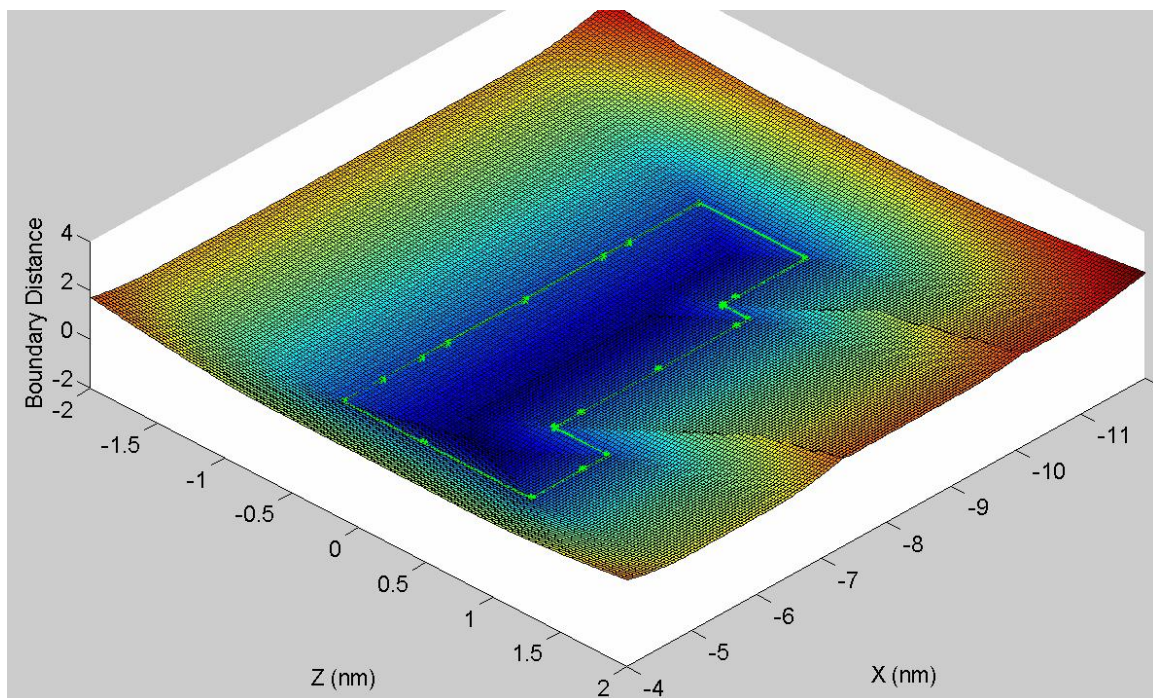


Fig. 44. MC-2 Lighting Constraint, MC2-Assessment Surface Plot



constraint or just outside the constraint. The membership function for ‘fair’ ranges between -0.25 and 0.05, with a maximum value for an MC2-assessment of -0.1. The ‘poor’ region includes points that are just within the constraint boundary or outside of the boundary. The membership function for ‘poor’ begins at an MC2-assessment of -0.1 and has full membership for values greater than or equal to 0.1. The output membership functions shown in Figure 39 are also used for the MC2-assessment input. The severity of the current position relative to constraint and navigation quality are captured using the linguistic terms ‘minor’, ‘moderate’, and ‘severe’ for the output membership functions.

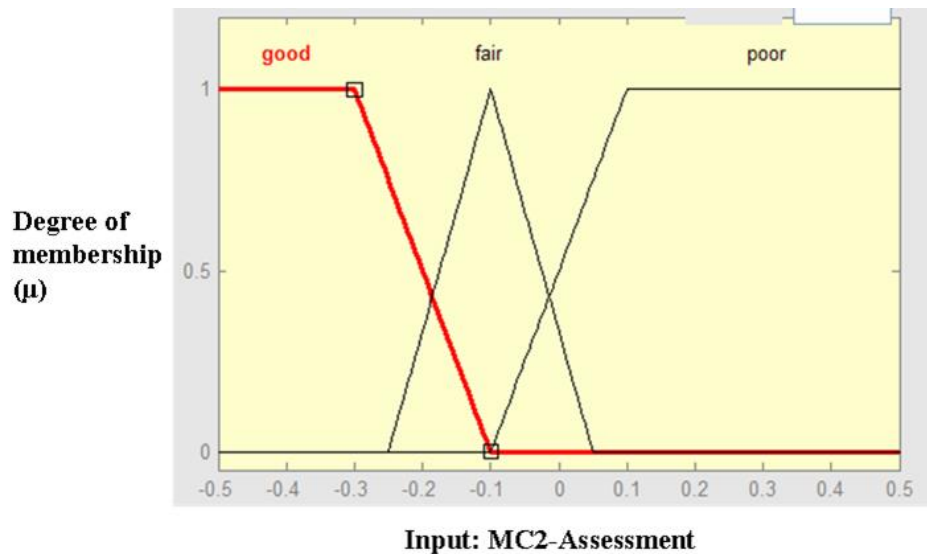


Fig. 45. MC2-Assessment Input Membership Functions

The rules for the Ti MC-2 lighting position constraint, without considering the other inputs, are identical to the rules for the elliptical constraint. These rules are captured in Table VI. When the rules are applied, the result is the input-output mapping shown in Figure 46. For these input membership functions, inputs located solely in the ‘good’ region have low values of severity (1-2). Then, when the ‘fair’

region of inputs is reached, the severity increases and then levels off with output values between 5 and 6. Finally, when the values in the ‘poor’ region are reached, the output increases to the maximum value of 10. The boundary of the lighting constraint corresponds a ‘severity’ output value of 6.6. Notice that the output value at the constraint boundary is not equal to the maximum severity output (10). This indicates that for this constraint it is acceptable to have cases that are just outside of the boundary, which is also the case for 3- $\sigma$  position constraint described above. However, for the 3- $\sigma$  position constraint the output value along the boundary was higher (7.4 instead of 6.6). The difference in the severity output values for 3- $\sigma$  position and lighting constraints is intended to represent that the lighting constraint is more flexible than the 3- $\sigma$  position constraint. As expected, values that exceed an MC2-assessment of 0.1 have full membership in the ‘poor’ region resulting in the maximum severity output value of 10. An input-output mapping for the region of points around the MC-2 lighting constraint is shown in Figure 47, with the black line indicating the constraint boundary.

Table VI. Rules for MC-2 Lighting Constraint

| <b>IF</b>          | <b>THEN</b>  |
|--------------------|--------------|
| MC2-Assessment is: | Severity is: |
| Good               | Minor        |
| Fair               | Moderate     |
| Poor               | Severe       |

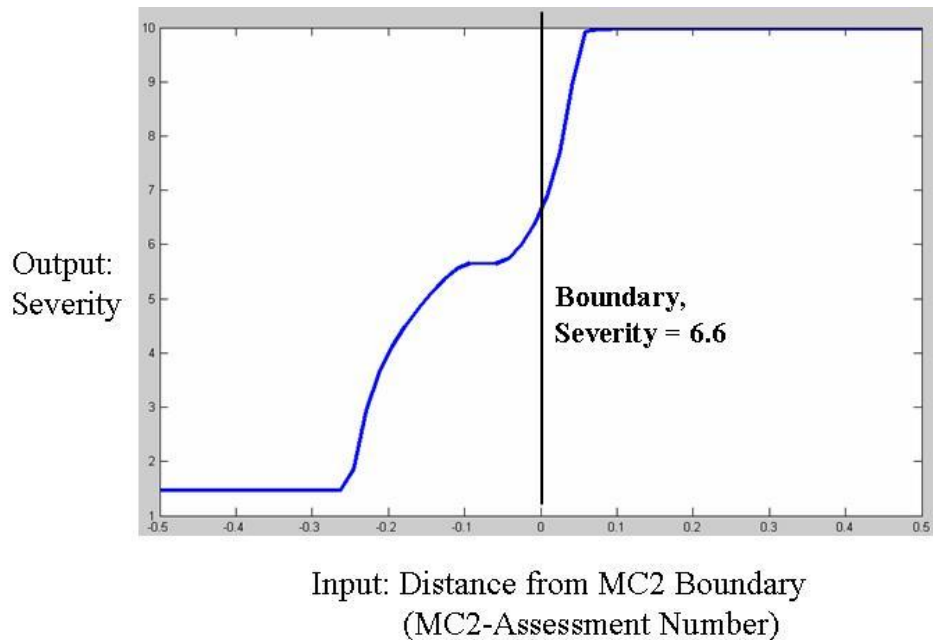


Fig. 46. Ti MC-2 Lighting Constraint, Input-Output Mapping

### c. Modeling Navigation Accuracy

Navigation accuracy is the final input to the Ti constraint model. This input will be provided as a subjective evaluation of the accuracy of the navigation solution. This evaluation must be made by a human flight controller, who has been monitoring the performance of the navigation system during the rendezvous trajectory. The input of navigation accuracy ranges from 0 to 5, with a value of 0 corresponding to very poor navigation accuracy and a value of 5 corresponding to very precise navigation accuracy.

The navigation accuracy input is called 'navigation-quality', which is modeled using the linguistic terms 'precise', 'degraded', and 'inadequate'. The input membership functions for navigation-quality are shown in Figure 48. The membership function for 'precise' represents very accurate navigation, with values ranging from zero membership for input values less than 2.5 to full membership for an input value

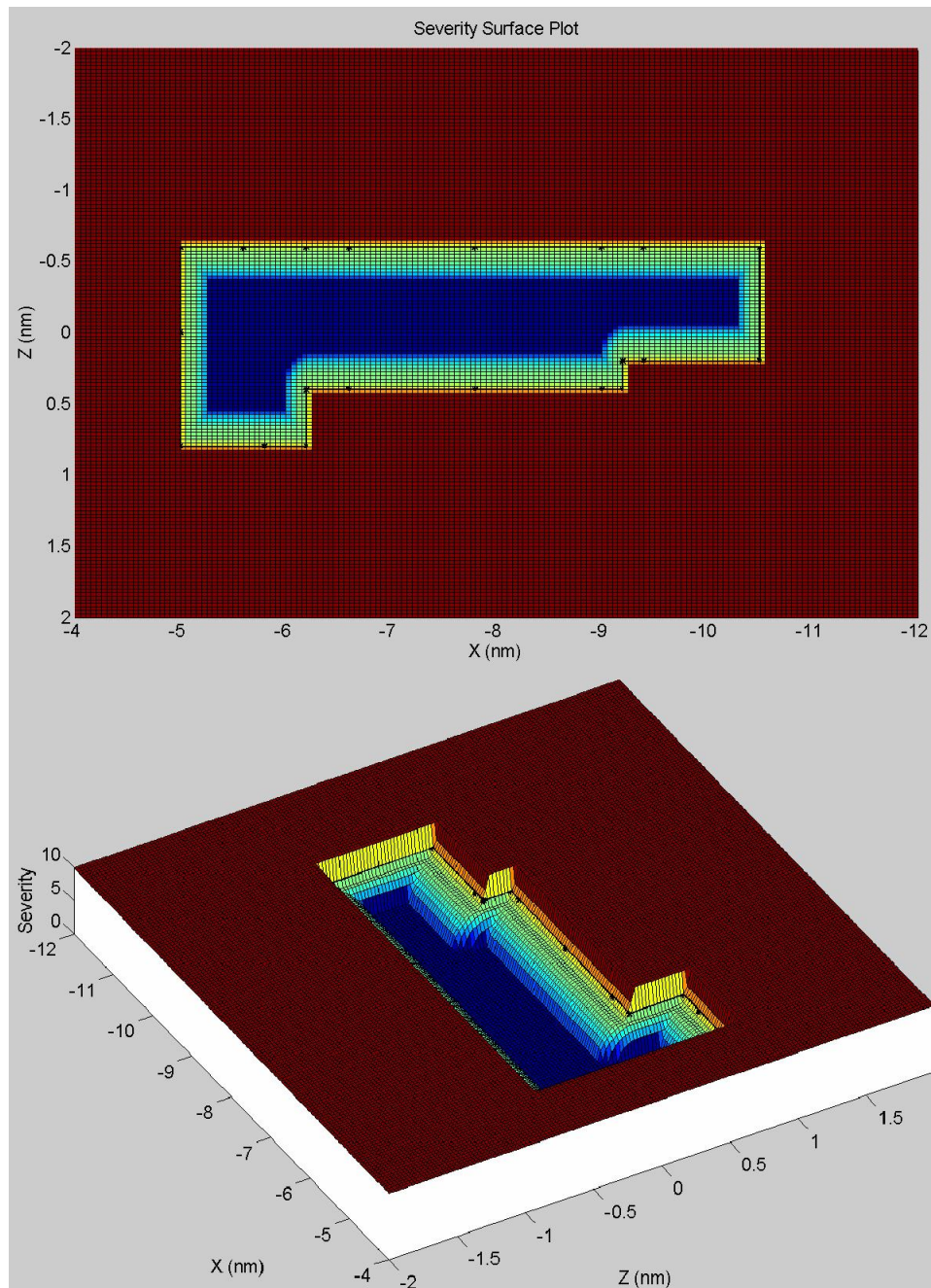


Fig. 47. MC-2 Lighting Constraint, Input-Output Surface

of 5.0. For the ‘degraded’ membership function, the values represent an increased navigation error. The membership function for ‘degraded’ ranges between 0.0 and 5.0, with a maximum value for a navigation-accuracy of 2.5. The ‘inadequate’ region represents navigation errors that result in unacceptable accuracy (and likely a recommendation for a Ti-delay burn). The membership function for ‘inadequate’ has full membership for navigation-accuracy of 0.0 and declines linearly until reaching zero membership for a navigation-accuracy of 2.5. These input membership functions will be used along with the ellipse-assessment and MC2-assessment inputs in the combined Ti position-constraint model described below.

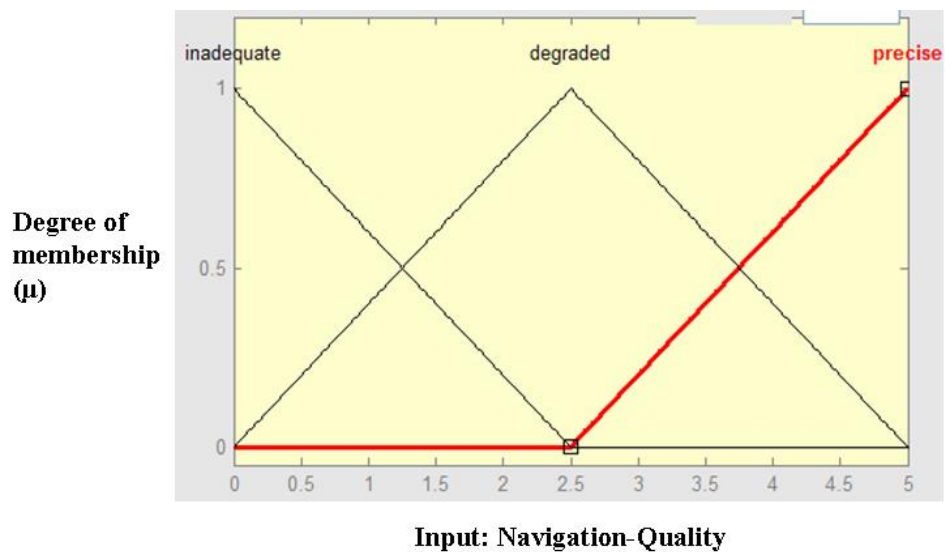


Fig. 48. Navigation-Quality Input Membership Functions

d. Combined Ti Position Constraint Model

The FL model of the Ti position constraints combines the three inputs described above to create an evaluation of vehicle position at Ti, which is relative to the position constraints and incorporates navigation accuracy. The inputs to the model are

the ellipse assessment, MC2 assessment, and navigation quality, which have input membership functions shown in Figures 38, 45, 48, respectively. These inputs are mapped to the corresponding output membership functions ('minor', 'moderate', and 'severe', shown in Figure 39) by the rules described in Tables VII, VIII, and IX.

For navigation-quality of 'precise' the rules are captured in Table VII. These rules represent the input-output mapping for the ellipse-assessment and MC2-assessment without any issues with the current navigation accuracy. When both inputs, ellipse-assessment and MC2-assessment, have values of 'good', then the severity output will be 'minor'. For cases where one input is 'fair' and the other input is 'good', the output will be 'moderate'. If both of the inputs are 'fair', then the output is 'severe'. The input-output mapping for these rules is captured in Figure 49. For cases with relative positions that are well within both boundaries, the outputs are very low values of severity (1-2). The severity outputs gradually increase as the input values begin to approach the constraint boundaries. The severity output approaches the maximum value of 10 when both input values are near the constraint boundaries. Notice that when one input is the minimum value, the input-output mapping corresponds exactly to the input-output mapping of the other input individually. For example, if ellipse-assessment is equal to 0.0, the input-output mapping reduces to the input-output mapping for MC2-assessment, as shown in Figure 46.

When the navigation-quality decreases and there is membership in the 'degraded' output membership function, the rules reflect a slightly more conservative evaluation of the position constraints. The rules for 'degraded' navigation-quality are captured in Table VIII. These rules are the same as the rules for 'precise' navigation-quality, except for the case when both inputs are 'good'. In this case, the output membership function is 'moderate' instead of 'minor' because of the uncertainty in the navigation



Table VII. Rules for Ti Position Constraint Model, Navigation = Precise

**IF** (Nav-Quality is Precise) **AND** ...

| 3- $\sigma$ Ellipse Assessment | MC2 Assessment |      |      |
|--------------------------------|----------------|------|------|
|                                | Good           | Fair | Poor |
| Good                           | M              | Mod  | S    |
| Fair                           | Mod            | S    | S    |
| Poor                           | S              | S    | S    |

Severity = [Minor (M), Moderate (Mod), Severe (S)]

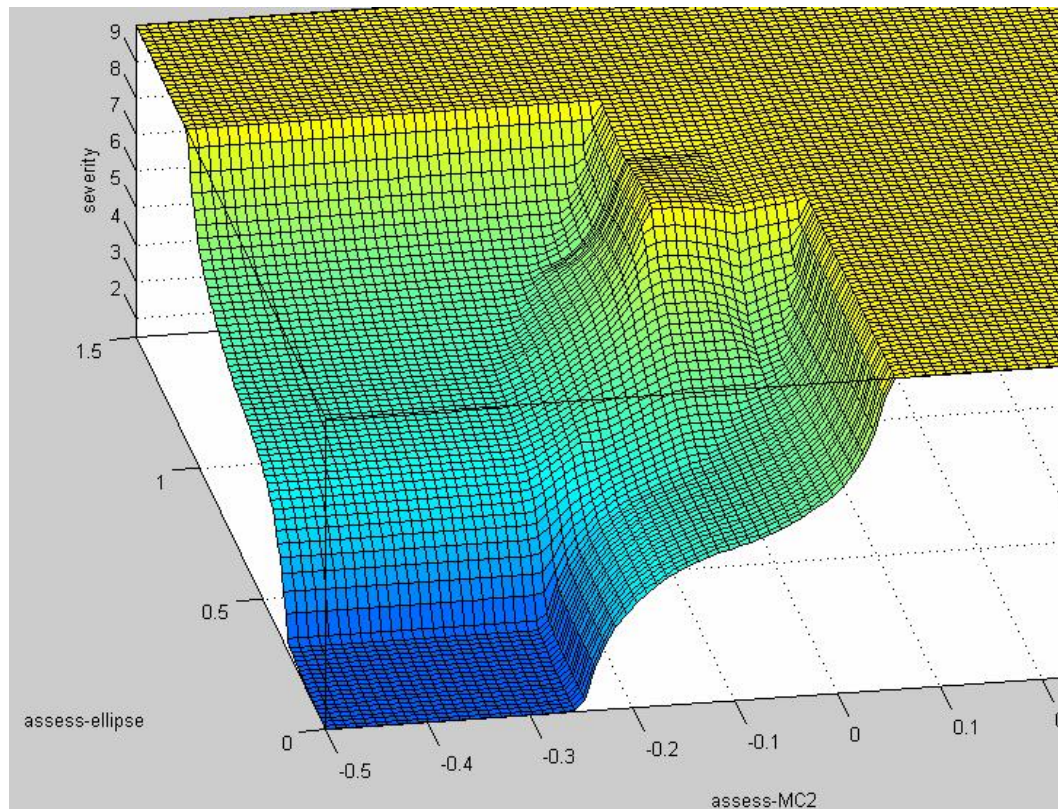


Fig. 49. Ti Position Constraint Model, Input-Output Mapping, Navigation-Quality = 5.0 ('precise')

accuracy. The input-output mapping for these rules is captured in Figure 50. Notice that the minimum output value is a severity of approximately 6, even for input values that represent relative positions that are well within both boundaries. As desired, this result will provide a more conservative evaluation of the constraints.

Table VIII. Rules for Ti Position-Constraint Model, Navigation = Degraded

**IF** (Nav-Quality is Degraded) **AND** ...

|   | <b>MC2 Assessment</b> |      |      |
|---|-----------------------|------|------|
| <b>3-<math>\sigma</math> Ellipse Assessment</b> | Good                  | Fair | Poor |
| Good  | Mod                   | Mod  | S    |
| Fair  | Mod                   | S    | S    |
| Poor  | S                     | S    | S    |

Severity = [Minor (M), Moderate (Mod), Severe (S)]

For ‘inadequate’ navigation-quality, the output membership function is ‘severe’ for all input values; these rules are captured in Table IX. ‘inadequate’ navigation-quality represents cases where the navigation errors are so large that very high severity outputs occur for all inputs (even for relative positions that are well within both boundaries). The input-output mapping for a navigation-accuracy equal to 0.0, corresponding to full membership in ‘inadequate’, will result in a severity output of 10.0 for all input values. The complete results of the Ti position constraint model are detailed in Chapter VII, ‘Experiment Results’.



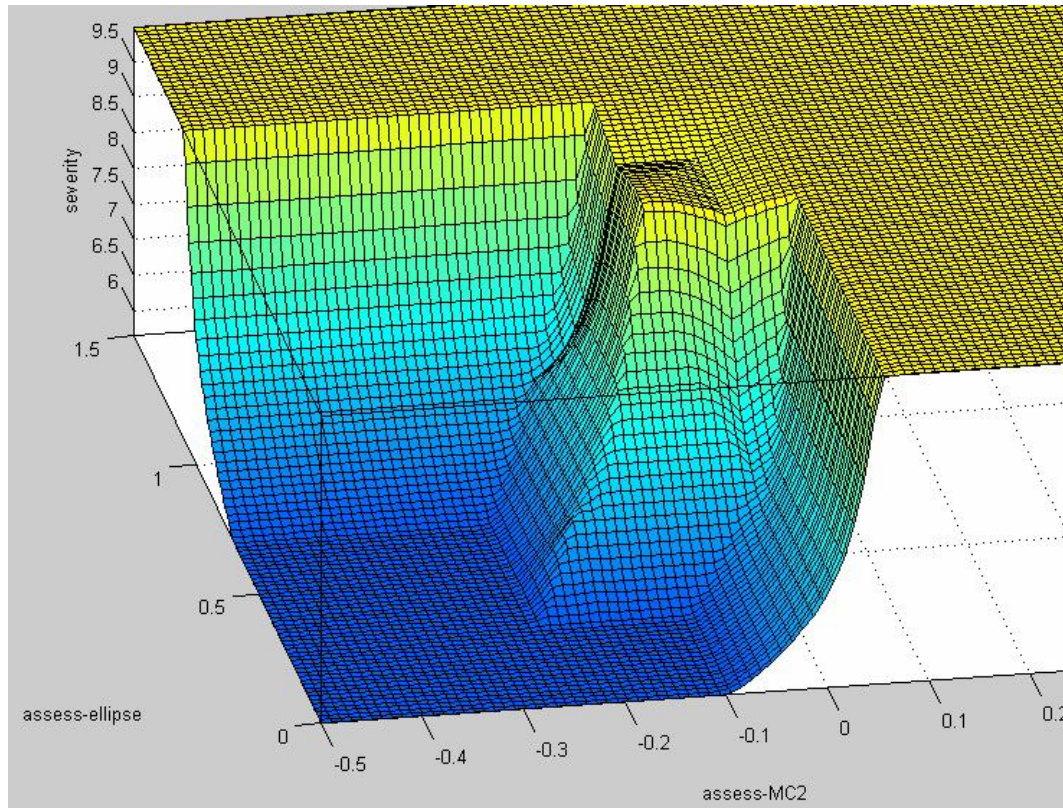


Fig. 50. Ti Position-Constraint Model, Input-Output Mapping, Navigation-Quality = 2.5 ('degraded')

Table IX. Rules for Ti Position-Constraint Model, Navigation = Inadequate

**IF** (Nav-Quality is Inadequate) **AND** ...

| 3- $\sigma$ Ellipse Assessment | MC2 Assessment |      |      |
|--------------------------------|----------------|------|------|
|                                | Good           | Fair | Poor |
| Good                           | S              | S    | S    |
| Fair                           | S              | S    | S    |
| Poor                           | S              | S    | S    |

Severity = [Minor (M), Moderate (Mod), Severe (S)]

## CHAPTER VI

### EXPERIMENT DESIGN

This section will discuss the test cases and simulations used to evaluate the prototype rendezvous-planning functions. The details are included for the TIG-slip-planning prototype, and the prototype that models the evaluation of vehicle position at  $T_i$  relative to constraints. The objectives of the test cases, the assumptions, and the test environment are discussed for each prototype.

#### A. TIG-slip Planning

##### 1. Objectives

The objective of the TIG-slip planning experiment is to evaluate the capabilities of the prototype by testing it in a realistic Space Shuttle rendezvous scenario. To be considered successful, the prototype must determine the maximum TIG-slip duration for both inertial and LVLH TIG-slip methods, compare the two methods, and provide a recommended TIG-slip method. The prototype must produce accurate results for both nominal and dispersed Shuttle rendezvous trajectories.

The success criteria for the calculation of the maximum inertial and LVLH TIG-slip durations is as follows: solutions must converge to within a reasonable number of iterations, the execution time of the algorithms should be minimized, and the resulting TIG-slip trajectory shall not violate the 4-nm position constraint. This criteria is summarized in Table X. In addition, the results must include relative-motion plots for evaluation by a human user.

For the TIG-slip comparison model, the output must result in the recommendation of a TIG-slip method with a longer maximum TIG-slip duration and/or lower

propellant cost. The specific input-output relationship must satisfy the intent of the TIG-slip comparison FL model described in the previous chapter. In general, the recommendation should be to use the inertial TIG-slip method unless the LVLH TIG-slip method provides a much longer TIG-slip duration (approximately 2-minutes longer), or has a much lower propellant cost (approximately 10 ft/sec less). The success criteria for the comparison method is simply that the recommended TIG-slip method is consistent with the FL model input-output mapping.

Table X. TIG-slip Planning Success Criteria

| <b>Criteria</b>         | <b>Desired Value</b>       |
|-------------------------|----------------------------|
| Number of Iterations    | < 10 iterations (Minimize) |
| Execution Time          | < 1 minute (Minimize)      |
| Maximum LVLH X-Position | < -4 nautical miles        |

## 2. Assumptions

This section describes the assumptions for the experiments used to evaluate the TIG-slip-planning prototype. For the prototype testing, a nominal rendezvous profile for a typical Shuttle-ISS mission is used. The test cases use the nominal rendezvous plan for the Space Shuttle mission designated as STS-110, which successfully rendezvoused with the ISS in April 2002. The TIG-slip-planning prototype is designed to determine the maximum TIG-slip duration for the NC burn. The portion of the trajectory that is evaluated in the experiment is shown in Figure 51, which includes the rendezvous trajectory from just prior to the NC burn and concludes at the Ti burn location. The NC burn is triggered based on time and will execute at the nominal TIG for

the nominal trajectory. This burn will be delayed by the desired TIG slip duration and then executed as either an inertial TIG slip or LVLH TIG slip depending on the desired method. The burn plan also includes a correction burn, which is called NCC. The NCC burn is used to correct any errors and properly target the  $T_i$  point. For this experiment, it is assumed that the NCC burn is automatically targeted using the simulation environment described below.

The inputs to the TIG-slip-planning prototype are an initial state (nominal or dispersed initial conditions) and a nominal burn plan (in this case STS-110). The outputs of the experiment are the relative motion between the target (ISS) and chaser (Space Shuttle Orbiter) spacecraft, the recommended TIG-slip duration and method, and the additional propellant costs.

The accuracy requirement on the TIG-slip duration is assumed to be 2 seconds. Solutions that are more precise than 2 seconds far exceed the error inherent sources in this problem (such as navigation errors and the capability for the crew to execute a burn at a given TIG). Therefore, the iteration process will be terminated when the solutions for TIG-slip duration converge to within 2 seconds.

### 3. Simulation Environment

The simulation environment and software components of the prototype are described in detail in this section. The simulation environment for the prototype is created using Matlab scripts. The Matlab scripts are used to call the FL models and trajectory routines.

The relative motion trajectories are computed using a NASA developed tool called 'Platform Independent Software Components for the Exploration of Space' (PISCES). This is a Java-based application, which includes many of the trajectory

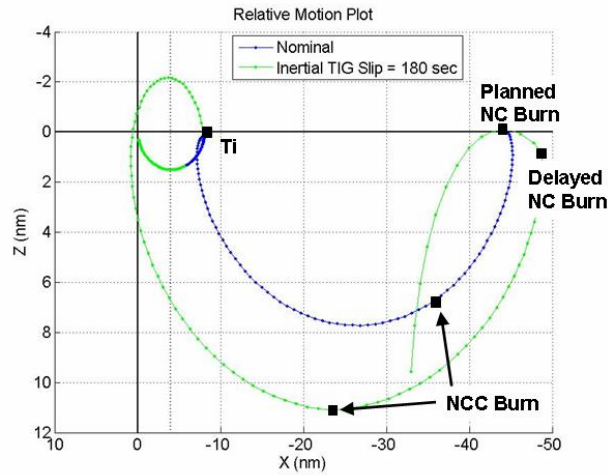


Fig. 51. TIG-slip Trajectory Example, Nominal and TIG-slip cases

planning tools used by NASA mission planners and flight controllers to plan and execute spacecraft rendezvous. The PISCES environment consists of a graphical interface as well as compiled Java libraries. The TIG-slip-planning prototype uses Matlab to call the PISCES Java libraries. These PISCES libraries are used to handle the execution of the relative-motion trajectories.

The process for executing the TIG-slip-planning prototype is outlined in Figures 52 and 53. These figures show flow-charts for the TIG-slip planning routines called in Matlab. The functions are color-coded to indicate if the step is a PISCES trajectory computation, a Matlab routine, or a FL model. Also indicated on the flow-charts are the inputs and outputs for each function.

When determining the maximum TIG slip, the first step is to execute the nominal trajectory (Figure 52). The nominal trajectory parameters will be used to create the TIG-slip trajectories. Once the nominal trajectory has been executed, the maximum TIG-slip durations are computed for the inertial and LVLH TIG-slip methods.

The procedure for computing the inertial and LVLH TIG-slip durations are identical. The only differences are the type of TIG slip executed by PISCES (inertial or

LVLH) and the FL model used to determine the TIG-slip duration for each iteration. The procedure for the inertial TIG-slip method is shown in Figure 53. The ‘Inertial TIG-slip Iteration’ FL model is called with the current value of ‘MAX\_X\_POSITION’ as an input, which is the location relative to the 4-nm position constraint. The output of the FL model is ‘DELTA\_TIME’, which is the amount the TIG-slip duration should be adjusted based on the proximity of the relative trajectory to the 4-nm constraint. This value is added to the previous TIG-slip duration (TIME\_OFFSET) to determine the new TIG-slip duration. Next, the relative motion trajectory is calculated for the new TIG-slip duration using PISCES (‘Execute TIG-slip Trajectory’). The outputs of the PISCES simulation are the new value of MAX\_X\_POSITION and the additional propellant cost over the nominal trajectory. The iteration process continues until the DELTA\_TIME value converges to an increment that is less than 2 seconds, assuming that the position constraint is also satisfied (MAX\_X\_POSITION is less than -4 nm). If the position constraint is not satisfied, then the iteration continues until acceptable relative motion is achieved. The FL models for the inertial and LVLH iteration process are sized to provide an efficient iteration process.

After the maximum TIG-slip durations are computed for both inertial and LVLH methods, the outputs are passed to the TIG-slip comparison model. The ‘TIG Slip Comparison Model’ is a FL model that compares the results of the inertial and LVLH TIG-slip methods and recommends one method. For negative output values, the inertial TIG-slip is recommended, and for positive outputs the LVLH TIG-slip is recommended. The magnitude of the output is a measure of the strength of the recommendation, with a maximum magnitude of  $\pm 1.0$ . The results of the prototype test cases and evaluation are captured in Chapter VII, ‘Experiment Results’.

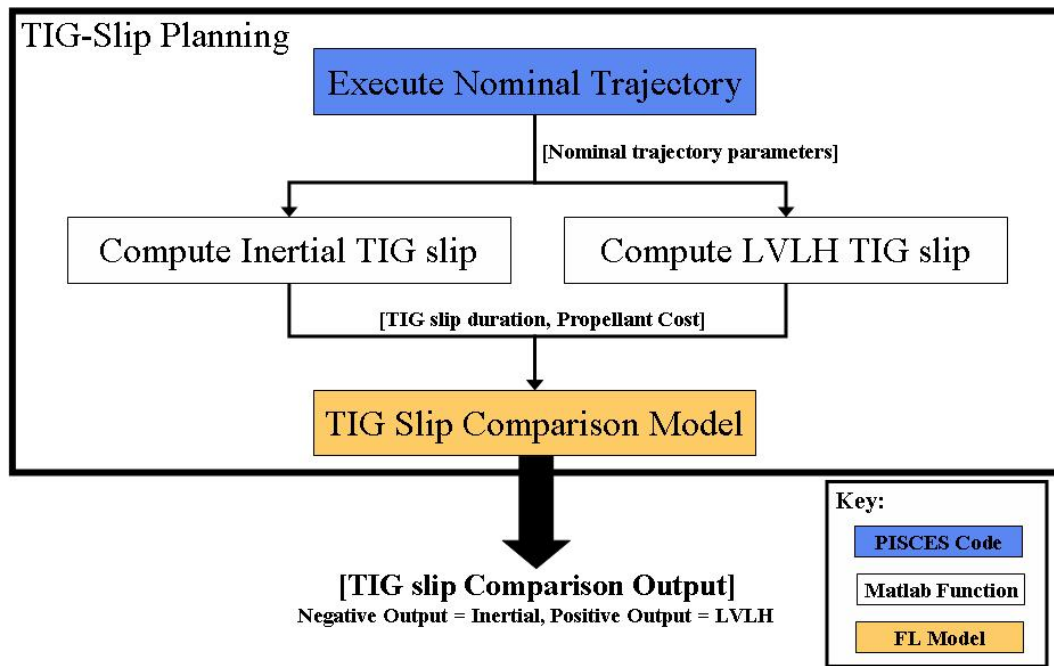


Fig. 52. TIG-slip Planning Flow-Chart

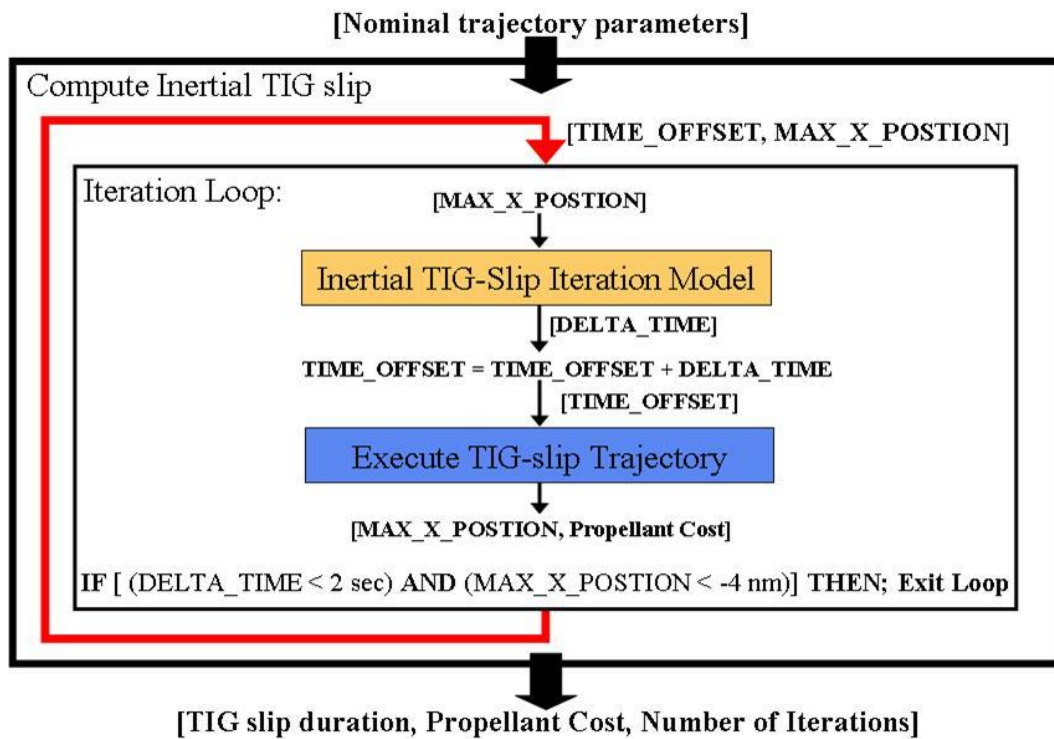


Fig. 53. Compute Inertial TIG-slip, Flow-Chart

## B. Evaluation of Vehicle Position at Ti

### 1. Objectives

The objective of this experiment is to evaluate the output of the ‘Ti Position-Constraint’ FL model for a range of relative positions in the vicinity of the constraint boundaries. This experiment will be used to determine if the FL model accurately models the ‘Severity’ of relative positions at the Ti burn time that are near position constraints. The success criteria is simply that the output is consistent with the intent of the FL membership functions and rules. This output will also be compared to an assessment of ‘Severity’ for a given relative position by a human flight controller. This comparison will be used to make future improvements to the accuracy of the FL model.

### 2. Assumptions

This section describes the assumptions used in the creation and evaluation of the Ti position-constraint model. The model is based on  $3\text{-}\sigma$  position and lighting constraint boundaries defined in the FDO Console Handbook [6]. The membership functions and rules of the FL model are used to determine the ‘Severity’ of the current Ti position relative to the position constraints and considering the current navigation accuracy. The FL model is described in detail in the previous chapter.

One important assumption in the modeling of these constraints is the ‘Severity’ output value assigned to relative positions that are located on the constraint boundary. Independent of other inputs, relative position values at Ti located on the  $3\text{-}\sigma$  and lighting constraint boundaries have severity values of 7.4 and 6.6, respectively. The selection of different values for the two boundaries reflects the potential for variations between the severity assigned for these constraints. Significant analysis would be required to identify the specific output values assigned to each constraint bound-



ary. This analysis task is beyond the scope of this research. The selected values are assumed to be reasonable outputs for the boundaries.

Another assumption is that the position constraints are static, i.e., they do not depend on the trajectory prior to  $T_i$ . This assumption allows for the evaluation of the constraints without requiring dynamic simulation. The constraint boundaries can change from flight-to-flight, but are not updated during the execution of the rendezvous trajectory [6].

### 3. Test Environment

The test environment and software components of the prototype are described in detail in this section. The prototype consists of Matlab scripts to create the ellipse-assessment and MC2-assessment input values for a position at  $T_i$  and a FL model used to generate the corresponding output. Matlab scripts are also used to test the prototype for a range of input values in the vicinity of the  $T_i$  position constraints.

The inputs to the  $T_i$  constraint prototype are the ellipse-assessment and MC2-assessment values for a given relative position at  $T_i$  and the current navigation-quality. The output of the ‘ $T_i$  Position-Constraint FL Model’ is the ‘Severity’ of the  $T_i$  position relative to the constraint, which can be used to determine if a  $T_i$ -delay burn should be executed (or  $T_i$ -delay correction, if a  $T_i$ -delay has already been performed).

The components of the  $T_i$  constraint model are captured in the flow-chart shown in Figure 54. In this model, Matlab functions ‘Compute Ellipse-Assessment’ and ‘Compute MC2-Assessment’ are used to create the assessment values that capture the position relative to the constraint boundaries. The ‘Navigation-Quality’ input is a subjective measure of the current navigation accuracy, which must be provided by a human user. These three inputs are passed to the ‘ $T_i$  Position-Constraint FL Model’, which generates the ‘Severity’ output.

A test script is used to cover the range of input values that encompass both position constraints. The input values for the LVLH Z-axis range from  $\pm 2$ -nm in increments of 0.025 nm, and the values for the LVLH X-axis range from 4 nm to 12 nm in increments of 0.05 nm. The input values are varied parametrically so that the entire input range is covered by a grid of points. This method allows for the evaluation of the input-output mapping for the entire range of interest. The results for this parametric evaluation are captured in Chapter VII, ‘Experiment Results’.

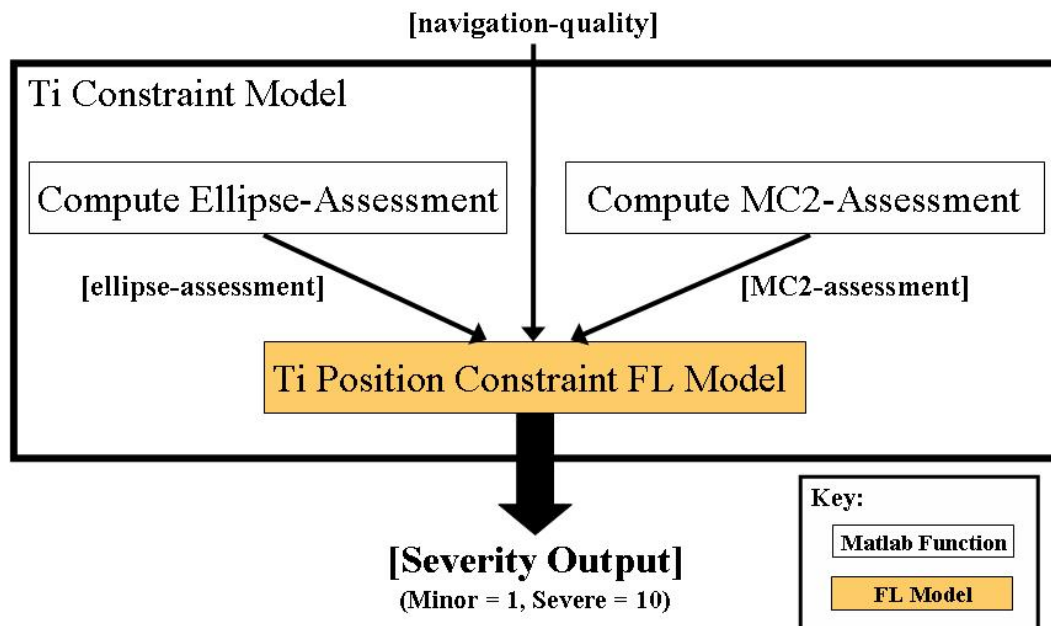


Fig. 54. Ti Position Constraint FL Model, Flow-Chart

### C. Hardware and Software Configuration

Table XI details the hardware and software configurations used for the development and testing of both prototypes.

Table XI. Hardware and Software Configuration for Prototype Experiments

|                   |                                   |
|-------------------|-----------------------------------|
| Computer:         | Dell Latitude 610, PC Laptop      |
| CPU Speed:        | 2.0 GHz                           |
| Memory (RAM):     | 1.0 GHz                           |
| Operating System: | Windows XP, 2002                  |
| Matlab:           | Version 7.2 (Release R2006a)      |
| Fuzzy Logic:      | Matlab, FL Toolbox, Version 2.2.3 |
| Java:             | Sun Microsystems, Version 1.5     |

## CHAPTER VII

## EXPERIMENTAL RESULTS

## A. TIG-slip Planning

The results of the TIG-slip planning prototype are detailed in this section. The TIG-slip duration and additional  $\Delta V$  costs are discussed for the inertial and LVLH TIG-slip cases for a non-dispersed nominal trajectory. The TIG-slip comparison model is used to recommend the type of TIG slip based on the inertial and LVLH TIG-slip results for the nominal trajectory. Results are also shown for cases with dispersed initial conditions to evaluate the robustness of the FL solution method for trajectory dispersions. The results of the prototype are then compared to existing methods used for TIG-slip planning to evaluate the prototype.

## 1. Results, Nominal Trajectory

For the nominal case, the maximum inertial TIG slip is 109 seconds, which results in an additional 10.7 ft/sec of additional  $\Delta V$  over the nominal trajectory. Figure 55 shows plots of the nominal trajectory, the iteration trajectories, and the trajectory for the maximum inertial TIG slip that satisfies the 4 nm X-relative position constraint. This solution is consistent with the results of the manually computed inertial TIG slip shown in Figure 9. The solution was found after 5 iterations with a total Matlab execution time of 5.4 seconds.

For the nominal case, the maximum LVLH TIG slip is 58 seconds, which results in an additional 9.6 ft/sec of additional  $\Delta V$  over the nominal trajectory. Figure 56 shows plots of the nominal trajectory, the iteration trajectories, and the trajectory for the maximum LVLH TIG slip that satisfies the 4-nm X-relative position constraint.

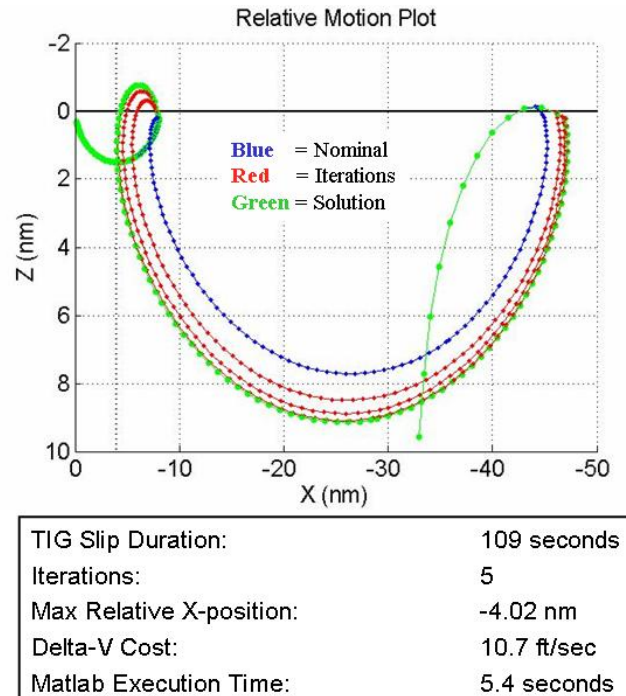


Fig. 55. Inertial TIG-slip Results, Nominal Trajectory

This solution is consistent with the results of the manually computed LVLH TIG slip shown in Figure 10. The solution was found after 3 iterations with a total Matlab execution time of 4.5 seconds.

A comparison of the maximum TIG-slip trajectories for inertial and LVLH cases is shown in Figure 57. In this figure, it is clear to see the difference in the trajectories where the longer slip duration of the inertial run results in additional relative motion away from the target vehicle. However, as expected both trajectories satisfy the 4-nm X-relative position constraint indicated on the plot by the dashed lined.

The maximum durations for inertial and LVLH TIG slips and their corresponding  $\Delta V$  costs are compared using the FL TIG-slip comparison model. The input to the comparison FL model is a difference in TIG slip duration of 51 seconds, favoring the inertial TIG slip and a difference in  $\Delta V$  cost of 1.1 ft/sec, slightly favoring the

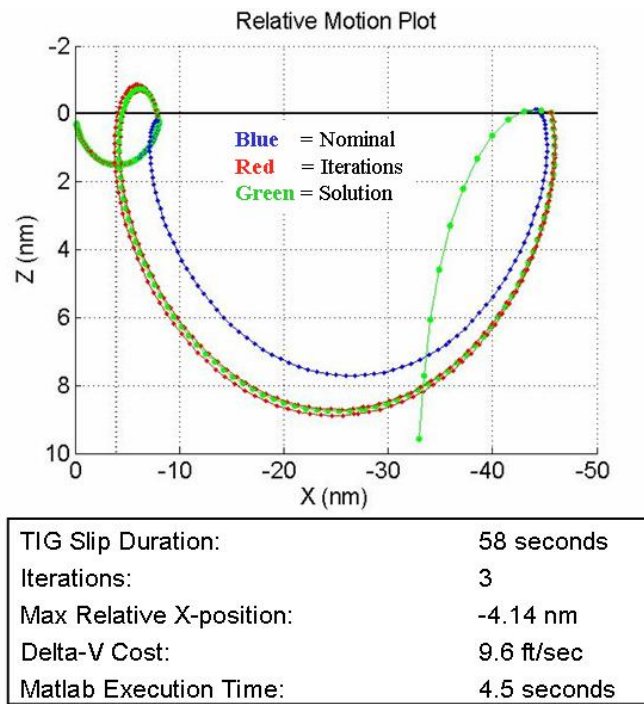


Fig. 56. LVLH TIG-slip Results, Nominal Trajectory

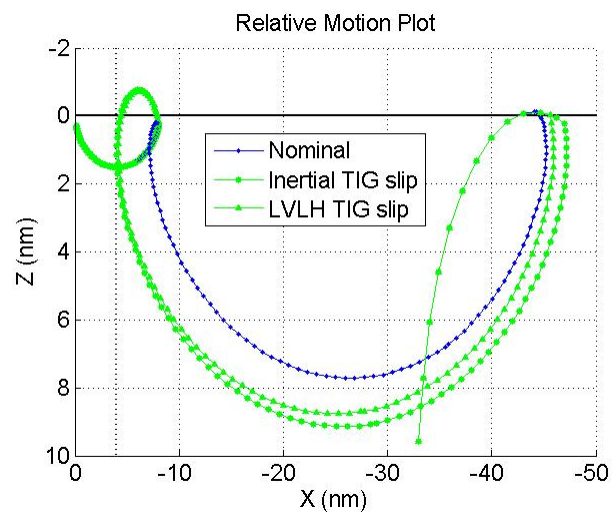


Fig. 57. Inertial and LVLH Maximum TIG-slip Comparison (Nominal Run)

LVLH TIG slip. The output of the model for these inputs is -0.33, which is a strong recommendation for the inertial TIG slip. For this case, the recommendation is that an inertial TIG slip of approximately 109 seconds is allowed with an expected  $\Delta V$  cost of approximately 10.7 ft/sec. This recommendation is consistent with the TIG-slip method that would be recommended by the FDO for these TIG-slip results.

## 2. Results, Dispersed Trajectories

A set of dispersed trajectories is used in order to test the capability of the TIG-slip prototype. This set of test cases consists of 100 trajectories with randomly dispersed initial conditions, with a  $1\text{-}\sigma$  distribution of 100 meters (m) in relative position and 0.1 m/sec in velocity. These dispersed cases are shown in Figure 58. For each of these dispersed cases, a maximum inertial and LVLH TIG-slip duration is calculated.

Figure 59 shows the maximum inertial TIG-slip trajectories calculated using the dispersed initial conditions shown in Figure 58. As expected all of the cases approach, but do not violate, the 4 nm X-relative position constraint. The maximum TIG-slip durations for these cases range from a minimum of 88 seconds to a maximum of 128 seconds. The mean TIG-slip duration is 109 seconds, which is equal to the TIG-slip duration for the nominal trajectory. Corresponding with these dispersed trajectories, the additional  $\Delta V$  costs above the nominal trajectory is a minimum of 9.4 ft/sec, a maximum of 11 ft/sec, and an average of 10.5 ft/sec. For these dispersed runs, the inertial TIG-slip solutions were found in 5 iterations.

The results for the LVLH TIG slip with dispersed initial conditions are shown in Figure 60. As with the inertial results, all of the cases approach, but do not violate, the 4 nm X-relative position constraint. The maximum TIG-slip durations for these cases range from a minimum of 46 seconds and a maximum of 70 seconds. The mean TIG-slip duration is 58 seconds, which is approximately equal to the TIG-slip

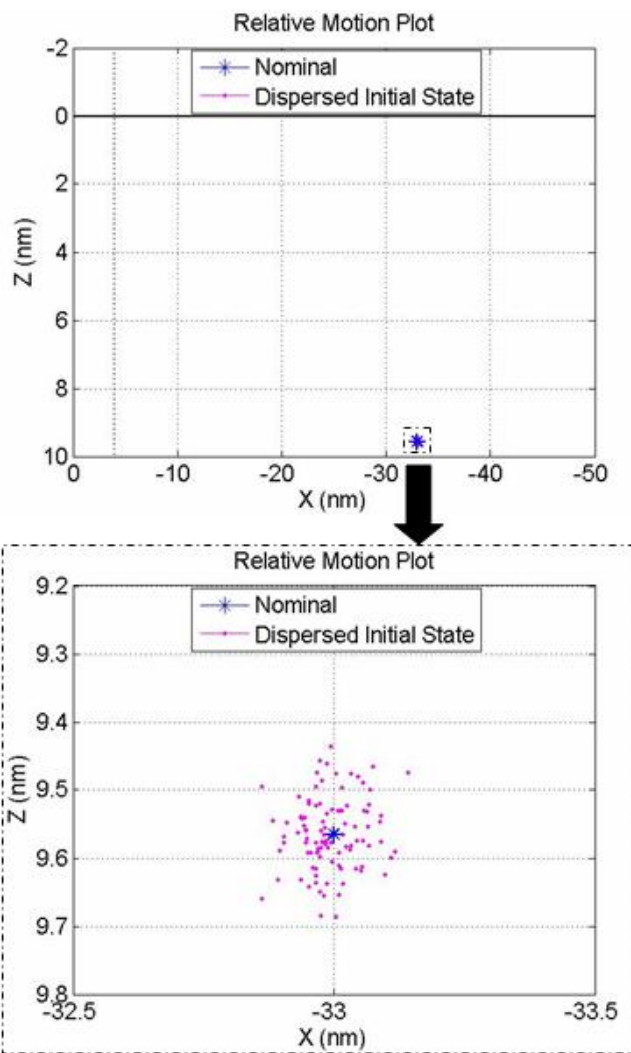


Fig. 58. Dispersed Initial Conditions for NC TIG slip



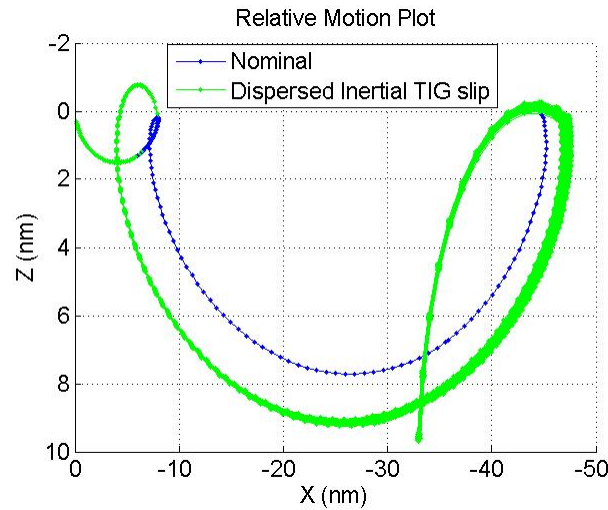


Fig. 59. Inertial TIG slip for Dispersed Initial Conditions

duration for the nominal trajectory. The range of additional  $\Delta V$  over the nominal trajectories ranges from 7.9 ft/sec to 12.2 ft/sec with a mean of 9.9 ft/sec. These solutions were found in a minimum and maximum of 2 and 4 iterations, respectively. These results and the results above confirm that the prototype for calculating the maximum inertial and LVLH TIG-slip durations is capable of handling dispersed trajectories, which result in a wide range of TIG-slip durations.

Results from these inertial and LVLH TIG-slip calculations are then compared using the FL TIG-slip comparison model. All of the dispersed trajectory cases result in a longer maximum TIG-slip duration for the inertial TIG-slip method. The average inertial TIG slip is 51 seconds longer than the average LVLH TIG slip and none of the trajectories have a longer LVLH than inertial TIG slip. Despite the difference in TIG-slip durations, both TIG-slip methods have similar  $\Delta V$  costs. The mean  $\Delta V$  cost for inertial is 1.1 ft/sec larger than the mean for LVLH TIG slips. Based on these results for the dispersed cases the FL TIG-slip comparison model properly recommends an inertial TIG slip for all of the dispersed trajectory cases.

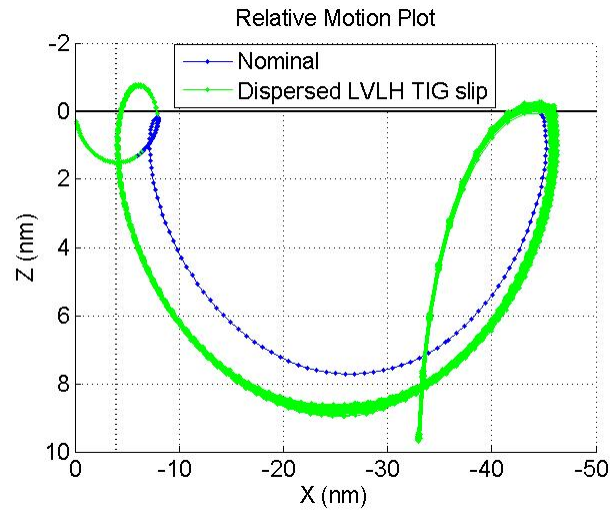


Fig. 60. LVLH TIG slip for Dispersed Initial Conditions

### 3. Comparison to Existing Methods

Recall that the existing method for determining the maximum duration and type of TIG slip involves a human flight controller iteratively running trajectory algorithms [6]. The flight controller computes the maximum TIG-slip duration and associated  $\Delta V$  cost for the inertial and LVLH TIG slips. Then, the recommended TIG-slip type, duration, and  $\Delta V$  cost is passed along to the Flight Dynamics Officer (FDO), who is in charge of the rendezvous maneuvers. These recommendations are typically approximate values, such as ‘inertial TIG slip of slightly less than 2-minutes’. Figures 9 and 10 show this analysis and the hand written markings used to denote the nominal trajectory (NOM), 1-minute (1), 2-minute (2), and 3-minute (3) TIG slips. Also included in these figures are the propellant costs for each of the slips written in terms of  $\Delta V$  in ft/sec. As part of this analysis, the flight controller takes into account the bias toward executing an inertial TIG slip over an LVLH TIG slip. This preference is reflected by recommending an inertial TIG slip unless the LVLH TIG slip provides around 2 minutes of additional TIG slip capability. The LVLH TIG slip could also be

recommended if it provides significant propellant savings over the inertial TIG slip. This analysis typically requires at least a few minutes to execute all of the runs and evaluate the results.

The TIG-slip prototype automates the determination of maximum TIG slip for the NC (phase change) burn shown in Figure 3. This method is able to quickly converge to the maximum TIG-slip durations for inertial and LVLH TIG slips. The iteration process results in very exact TIG-slip durations that actually are more precise than necessary. The iterations are terminated when the outputs converge to within 2-seconds of the TIG-slip duration that would result in relative motion that just satisfies the 4 nm X-relative position constraint. By providing the FDO with this very precise TIG-slip duration, they can decide how much conservatism they want to apply to the solution.

The success criteria for the prototype calls for the minimize the number of iterations. However, FL iteration method does not converge in the optimal number of iterations. Typical iteration numbers for the prototype are 5 iterations and 3 iterations for inertial and LVLH TIG slips, respectively. Since the execution time of the prototype is very quick, approximately 5 seconds, this is not an issue. A benefit of the FL iteration method is that it does not require an analytical model of the system dynamics which would be necessary for optimal convergence methods.

Once the TIG-slip durations are calculated, the durations and propellant costs are input into the FL TIG-slip comparison model. These inputs are used to determine the recommendation of TIG-slip method. This model takes into account the bias toward inertial TIG slips in its calculations. The output of this prototype is a value between -1 and 1, with negative numbers corresponding to a recommendation of an inertial TIG slip and positive numbers corresponding to a recommendation of an LVLH TIG slip. Numbers closer to the extrema of this range represent a stronger

recommendation for that type of TIG slip. This allows the human flight controller to understand the strength of the recommendation when evaluating the output of the FL comparison model.

This prototype successfully models the process used to compute TIG-slip durations and determine which type of TIG slip to recommend. The results are actually more precise than the existing method in terms of TIG-slip durations. All of the data used to make a recommendation is provided to the human user including TIG-slip duration and  $\Delta V$  costs for both TIG-slip methods, relative motion plots, and strength of the recommendation of the TIG-slip type. Since the data used in making the recommendation is output by the prototype, this implementation allows the human to evaluate the results and provide an alternate result, if necessary. This implementation matches the desired level of automation with the computer considered prime, with the human also making the calculations as a backup.

## B. Evaluation of Vehicle Position at $T_i$

This section discusses the results of the Fuzzy Logic model of  $T_i$  position constraints. The results are captured by showing surface and contour plots of the FL model output for all points in the vicinity of the position constraints. Results are shown for various navigation-quality values that reflect cases with precise navigation position accuracy to cases with inadequate position accuracy. The results are discussed and compared to existing methods.

### 1. Results

Recall that the position constraints are made up of two boundaries, the elliptical constraint representing the  $3\text{-}\sigma$  position dispersion around the nominal  $T_i$  burn location

and the linear constraint used to ensure that the proper lighting conditions occur when the MC-2 burn is executed. A plot of these constraints is shown in Figure 11.

Figure 61 shows a surface plot of the output of the FL model of Ti position constraints for precise navigation position accuracy (Navigation-Quality = 5.0). This figure captures the severity of all relative position input values in relation to the constraints. The warm colors (red, orange, yellow, etc.) correspond to high severity values and the cool colors correspond to low severity numbers (green, blue, etc.). As desired, positions that are outside both constraints have the maximum severity value of 10. As the positions approach the center of the constraints the severity decreases to lower values. This output data is also captured in Figure 62, which shows contours of the severity values.

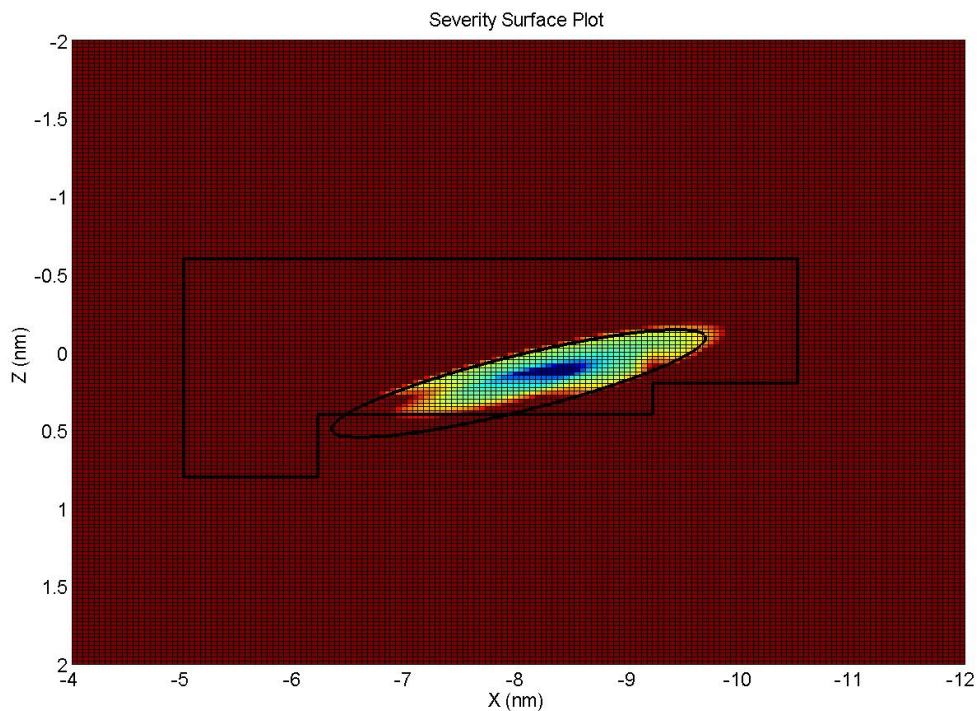


Fig. 61. FL Ti Constraint Model Output (Surface Plot), Nav Accuracy = 5.0

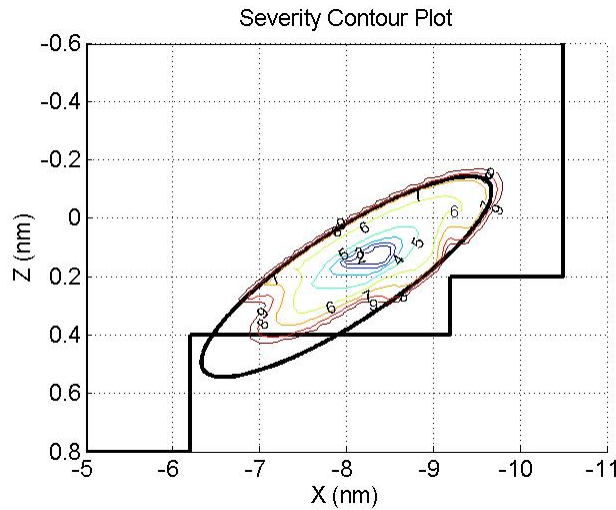


Fig. 62. FL Ti Constraint Model Output, Nav Accuracy = 5.0

Figure 63 shows the FL model output for slightly degraded position accuracy (Navigation-Quality = 4.0). For this case, the minimum severity output is increased to around 5.0. These results reflect that with uncertain position accuracy a correction burn could be desired for relative position values that fall within both position constraints.

As the navigation accuracy continues to degrade, the minimum severity increases and the severity contours expand toward the edge of the  $3\text{-}\sigma$  position constraint. This trend is evident in the results shown in Figures 64 and 65. In Figure 65 the severity is a minimum of approximately 8.0 even at the very center of the position constraints. Not pictured are the results for a navigation accuracy of 0.0. For this case, the severity output is equal to 10.0 for all inputs.

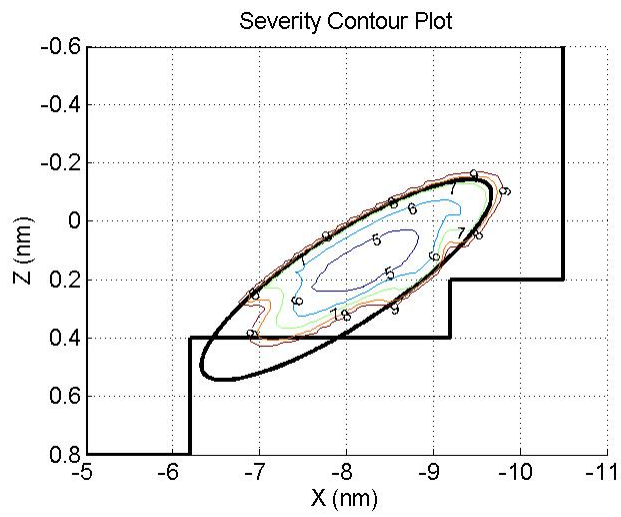


Fig. 63. FL Ti Constraint Model Output, Nav Accuracy = 4.0

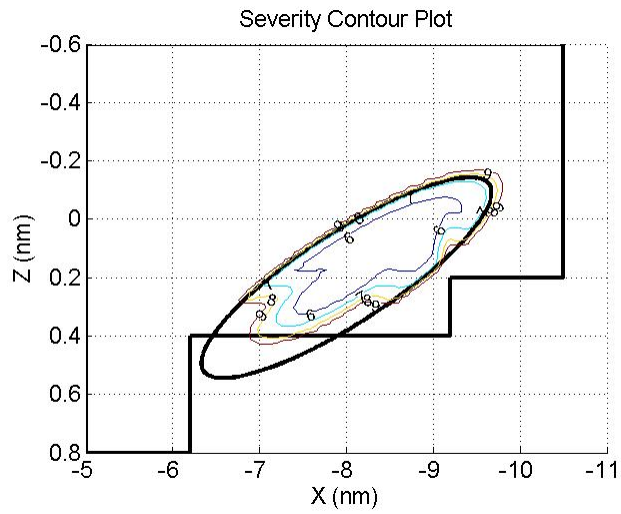


Fig. 64. FL Ti Constraint Model Output, Nav Accuracy = 2.5

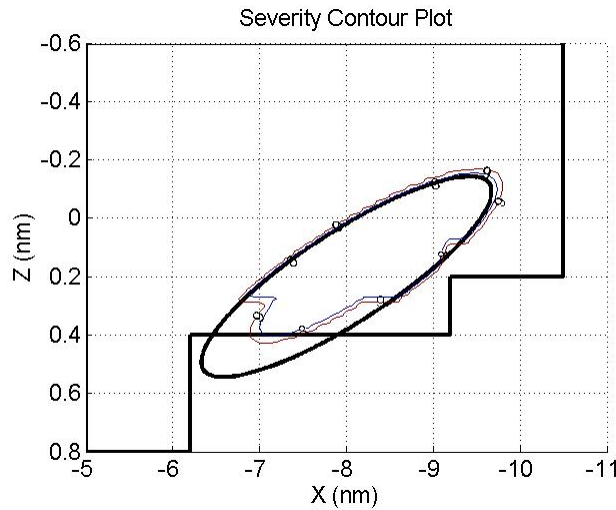


Fig. 65. FL Ti Constraint Model Output, Nav Accuracy = 1.0

## 2. Comparison to Existing Methods

In the current implementation of this function, ground computers are used to predict the location of the Orbiter when the Ti burn is to be executed and the flight controllers determine if this position is acceptable. This determination is based on the location relative to the position constraints and the current navigation accuracy. Feedback on the prototype suggests that in practice, flight controllers use larger boundaries than what is reflected in the FL model of Ti position constraints [38]. For some cases, it is acceptable to have relative positions outside of the  $3\text{-}\sigma$  boundary and not execute a correction burn. In these cases, it would be appropriate to execute a simulation to evaluate the relative motion plots for the case in question. Also, the current model results in severity values of 10 for some relative positions that are within, but near both boundaries. A more accurate model would have severity values less than the maximum for all relative positions that fall within both boundaries.

Since the current constraint boundaries are approximate and based on “rules-of-thumb”, another improvement to the model could be created by using Monte Carlo



techniques to create exact boundaries. These boundaries would represent the true constraints based on relative motion that are correlated to navigation accuracy. However, this could prove challenging to model because the boundaries would be complex, and thus it would be difficult to represent them analytically.

Another issue with the current model is that the severity increases equally for changes in the LVLH X and Z-axes. This is an issue since the change in relative motion for a change in X-relative position is not equal to the change in relative motion for an equal change in Z-relative position. Therefore, a more accurate model would take into account the difference between a relative position change between the X and Z LVLH axes.

The prototype results in an automated process that determines the ‘severity’ of the predicted Ti position relative to the position constraints and based on the current navigation accuracy. The severity output by the prototype can then be evaluated by the human user or compared to a threshold that would be used to determine if a Ti-delay or Ti-delay correction are necessary. The current prototype shows that FL can be effectively used to model constraints. This implementation requires that the human user to evaluate the output data and provide their own solution for comparison to the automated solution. Several issues were identified that can be used to make future improvements to more accurately capture the entire decision-making process for this rendezvous function.

### C. Assessment of Level of Automation

Both of the prototypes were designed to result in an implementation consistent with a level of automation of 4 on the ‘Decide’ scale (‘Both the human and the computer perform ranking tasks, the results from the computer are considered prime’, Figure

2). The prototype of the TIG-slip planning was successfully implemented to the desired level of automation. The computer provides a recommendation that can be used and can also be verified by the human user before implementation. It is likely that this prototype could be used to automate this functionality at a higher level of automation. However, since this function has been performed manually in the past, it would require time for the users to build trust in the automated functionality. Trust is most often the limiting factor in the implementation of automation[10]. The trust in the automation can best be built by implementing at a level where the human can still be used as a backup to evaluate the results. Another issue with implementing a higher level of automation is the ability to handle off-nominal cases that were not known or understood during the design process. This is a valid concern since the prototype is based on relative motion that is similar to the nominal trajectory. Cases that greatly deviate from these trajectories could result in unknown and undesirable solutions. By having a human backup, these cases can be identified and addressed if they appear. As the familiarity with the model increases with use, this concern can be lessened and the level of automation could be increased in the future. The results of this prototype indicated that the FLOAAT recommended level of automation is appropriate for this function. The prototype validates the feasibility of implementing an automated solution for this decision-making task. As familiarity with this method is increased during operational use, the level of automation could be increased in the future. However, at present, this level of automation is appropriate.

For the Ti position constraint, the prototype can provide an assessment of the severity of the position relative to the constraints. However, the implementation still requires the human to make a subjective assessment of the current navigation accuracy as an input to the model. The output must also be subjectively evaluated to determine the relative importance of the ‘severity’ output by the model, based on the current

availability of the crew to execute a Ti-delay or Ti-delay correction burn. Because of the subjective nature of this decision, these aspects are difficult to automate. Since the prototype is successful in automating a portion of the decision-making process (the evaluation of the constraints), but not the entire process, this prototype does not successfully implement the FLOAAT recommended level of automation. In this implementation, the human is still primarily responsible for performing the final evaluation of the output, and the automation can be used as a tool in this process. This implementation is consistent with a level of automation of 2 on the FLOAAT ‘Decide’ scale (Figure 2). However, the prototype does increase the level of automation to a level of 4 for a portion of the decision-making performed in the prototype. The issue with automating subjective decision-making tasks should be captured as feedback to the FLOAAT process.

#### D. Assessment of Fuzzy Logic

Fuzzy Logic has been successfully used to create prototypes for the selected rendezvous decision-making functions. This technique is well suited for the types of decisions made by human flight controllers, which are based on ‘rules-of-thumb’ captured in the flight rules and procedures. Much of the success of the FL technique is due to its ability to capture approximate terms often used by humans. This ability also results in models that are easy to understand and modify. These attributes and the success of the prototyping efforts show that FL is a useful method for modeling certain types of human decision-making tasks. There are some issues that limit the type of modeling that is successful, and these issues are discussed below.

One issue is that it is difficult to create and modify FL models of constraint boundaries that cannot be modeled analytically. This issue appeared in the creation

of the MC-2 lighting constraint for the Ti position constraint model. The boundaries of this constraint had to be determined and created by hand and hard-coded as part of the model. Any modifications to this constraint would result in tedious modification to the hard-coded values. This issue could be alleviated if the constraints could be approximated by functions that can be described analytically.

There is also an issue with capturing subjective decisions using FL. These decisions are often made without the use of quantifiable data, and are thus difficult to model using any automated decision-making technique. For example, the assessment of the current navigation is based on a variety of factors that are difficult to quantify, such as the recent navigation performance, or the occurrence of a burn or other propulsive event (such as a water dump or venting event) that is not yet reflected in the navigation solution. This could be overcome by attempting to quantify this information. However, this is a difficult and often an impractical task. As a result, these types of functions will often remain as tasks performed by humans.

Another issue is model complexity. Both prototypes are modeled with fairly straightforward and simple IF-THEN rule bases. However, the addition of one or two additional variables to the model would dramatically increase the model complexity. This issue likely limits the use of FL to individual functions similar to those prototyped here. For example, the model complexity would become too great if a model were needed to perform the entire rendezvous decision-making process. Since the prototypes are intended to model individual tasks, and not the entire decision-making process, there is little risk of prohibitive model complexity. However, this concern should be considered if these types of models are accepted into wide use or are integrated into larger systems. Overall, Fuzzy Logic is judged to be an excellent candidate for modeling the human decision-making for the selected rendezvous functionality.

## CHAPTER VIII

### CONCLUSIONS

Based on the results presented in this research, the following conclusions are drawn:

1. Prototypes of the selected Shuttle/ISS rendezvous decision-making tasks validate the feasibility of implementing higher levels of automation for such tasks. The TIG-slip-planning prototype automates the determination of the maximum TIG-slip duration for both inertial and LVLH TIG-slip methods and recommends the desired TIG-slip type producing accurate TIG-slip results that are comparable to existing methods but are calculated more quickly. The Ti-constraint prototype results in a partially automated process that can determine the ‘severity’ of the predicted Ti position relative to the position constraints and based on the current navigation accuracy.
2. The results of the prototype confirm that the FLOAAT recommended level of automation is accurate. The prototype of the TIG-slip planning was successfully implemented to the desired level of automation. The computer provides a recommendation that can be used and verified by the human user before implementation. The results of this prototype indicated that the FLOAAT recommended level of automation is appropriate for this function. The prototype implementation for Ti constraints is successful in automating a portion of the decision-making process but not the entire process. The prototype still requires the human to make a subjective assessment of the current navigation accuracy as an input to the model and determine the relative importance of the ‘severity’ output. However, the prototype does increase the level of automation to a level of 4 for a portion of the decision-making performed in the prototype.

3. The prototypes demonstrate that FL can be effectively used to model human decision-making used in spacecraft rendezvous. FL is well suited for the types of decisions made by human flight controllers, which are based on “rules-of-thumb” captured in the flight rules and procedures. Much of the success of the FL technique is due to its ability to capture approximate terms often used by humans. The FL iteration method used in TIG-slip planning demonstrates the capability of this technique to quickly and effectively converge on a solution without requiring an analytical model of the system dynamics. The prototype performs the iteration in a manner similar to how a human would perform the same function. By emulating the human iteration process, the chances of acceptance and trust in the automation are higher because the method is easy to understand and can be quickly adjusted. This method also provides outputs that can be easily understood by a human user. In the FL comparison model used in TIG-slip planning, output values close to the extrema of range represent a stronger recommendation for that type of TIG-slip. This allows the human flight controller to understand the strength of the recommendation when evaluating the output of the FL comparison model. Because the models are easy to understand, they are also easily modified, which allows for future improvements to the model accuracy.
4. The methodology for prototyping rendezvous functions at higher levels of automation is judged to be a promising technique. The FLOAAT tool can be used to accurately identify functions that can be implemented at an increased level of automation. FL has many desirable attributes for modeling human decision-making, which make it an excellent candidate for additional spaceflight automation applications.

## CHAPTER IX

## RECOMMENDATIONS

The results and conclusions indicate several issues and areas for future work and improvements.

1. **TIG-slip planning:** Future test cases should evaluate the accuracy of the prototype for NC burn locations originating at different relative positions. The NC burn can be executed at ranges on the order of  $\pm 20$  nm from the nominal location of 40 nm. Executing these cases would additionally test the robustness of the prototype.
2. **Ti Constraints model:** In the current model, there are inputs that are within both constraint boundaries but have severity values of 10. A more accurate model would have severity values less than the maximum for all relative positions that fall within both boundaries. In practice, flight controllers use larger boundaries than what is reflected in the FL model of Ti position constraints. For some cases, it is acceptable to have relative positions outside of the  $3\text{-}\sigma$  boundary and not execute a correction burn. Future work should focus on creating a more realistic representation of how these boundaries are interpreted by the human flight controller.

The constraint boundaries are currently approximations based on “rules-of-thumb”. Another future improvement to the model would be to create more accurate boundaries by using Monte Carlo techniques to create exact boundaries. These boundaries would represent the true constraints based on relative motion that are correlated to navigation accuracy. However, this could prove challenging to model because the boundaries would be complex, and thus it

would be difficult to represent them analytically. These boundaries could also include considerations for propellant requirements and jet failure cases.

In future versions, the  $3\text{-}\sigma$  position constraint should be modeled using Gaussian membership functions instead of triangular membership functions. By using Gaussian membership functions the values can be aligned with the  $1\text{-}\sigma$ ,  $2\text{-}\sigma$ , and  $3\text{-}\sigma$  contours. This will result in a more accurate and intuitive model of this constraint.

The current model treats position relative to the lighting constraint as equivalent for changes in X-position and Z-position. A more accurate model would also take into account the difference between a relative position change between the X and Z LVLH axes. Future work should include analysis to determine the sensitivity in the lighting conditions for changes in X-position compared to changes in Z-position. This analysis could be included with the Monte Carlo study to determine more accurate constraint boundaries.

3. **Fuzzy Logic:** It is difficult to create and modify FL models of constraint boundaries that cannot be modeled analytically. The boundaries of this MC2 lighting constraint had to be determined and created by hand and hard-coded as part of the model. Any modifications to this constraint would result in tedious modification to the hard-coded values. This issue could be alleviated if the constraints could be approximated by functions that can be described analytically. Future work should include investigation of how such boundaries could be more easily created and modified.
4. **FLOAAT recommended level of automation:** An issue was uncovered with automating subjective inputs and output for decision-making tasks. In the Ti constraint model the only source for the navigation-quality input is a



subjective evaluation by a human. These types of evaluations are often made without the use of easily quantifiable data and thus difficult to model using any automated decision-making techniques. As a result, functions with inputs or outputs that are measured subjectively will often remain as tasks performed by humans. Future versions of the FLOAAT process should capture this issue as a potential limitation to higher levels of automation.

The results of this research encompass only a small portion of the complete set of rendezvous planning functions. Additional prototyping should be used to provide additional confirmation of the accuracy of the FLOAAT specified levels of automation.

## REFERENCES

- [1] Douglas Zimpfer, Peter Kachmar, and Seamus Tuohy. Autonomous rendezvous, capture and in-space assembly: Past, present and future. In *A Collection of Technical Papers - 1st Space Exploration Conference: Continuing the Voyage of Discovery*, volume 1, pages 234 – 245, Orlando, FL, January 2005.
- [2] Alan R. Crocker. Operational considerations in the development of autonomy for human spaceflight. In *A Collection of Technical Papers - 1st Space Exploration Conference: Continuing the Voyage of Discovery*, volume 2, pages 1082 – 1089, Orlando, FL, January 2005.
- [3] Michael F. Machula and Gurpartap S. Sandhoo. Rendezvous and docking for space exploration. In *A Collection of Technical Papers - 1st Space Exploration Conference: Continuing the Voyage of Discovery*, volume 2, pages TBD–TBD, Orlando, FL, January 2005.
- [4] Ryan W. Proud and Jeremy J. Hart. FLOAAT, a tool for determining levels of autonomy and automation, applied to human-rated space systems. In *Proceedings of Infotech@Aerospace*, number AIAA-2005-7061, Arlington, VA, September 2005.
- [5] John Goodman. History of Space Shuttle Rendezvous. Technical Report JSC-63400, Flight Design and Dynamics Division, United Space Alliance, Oct 2006.
- [6] Flight Design and Dynamics Division. Flight Dynamics Officer (FDO) Console Handbook, V3 On-Orbit Trajectory Operations. Technical Report DM-CH-07, PCN-3, NASA, Johnson Space Center, October 2005.

- [7] Constellation Program Office. Constellation Architecture Requirements Document (CARD). Technical Report CxP 70000, NASA, Dec 2006.
- [8] Ryan W. Proud, Jeremy J. Hart, and Richard B. Mrozinski. Methods for determining the level of autonomy to design into a human spaceflight vehicle: A function specific approach. In *Proceedings of NIST Workshop on Performance Metrics for Intelligent Systems*, September 2003.
- [9] John R. Boyd. The Essence of Winning and Losing. Technical report, United States Air Force, January 1996.
- [10] Jeremy Hart, Ryan Proud, and Joshua Hardy. Autonomy and Automation Pilot Study: Rendezvous & Docking Final Report . Technical Report AC1L1-17, AFM-FLOAAT0004, NASA, Johnson Space Center, August 2005.
- [11] Flight Control Division. Final Mission Rules, Apollo 17. Technical Report MSC-01807, NASA, Manned Spacecraft Center, Sept 1972.
- [12] Flight Design and Dynamics Division. Space Shuttle Flight Rules, Section 2A Flight Operations. Technical Report PCN-4, NASA, Johnson Space Center, April 2005.
- [13] Chris D'Souza. Navigation and Dispersion Analysis of CEV ISS Rendezvous Scenarios. Technical Report EG-CEV-06-02, NASA, Johnson Space Center, March 2006.
- [14] G.E. Cooper and Harper. R.P. The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities. Technical Report NASA TN D-5153, NASA, April 1969.

- [15] A. H. Roscoe. Assessing pilot workload in flight. In *Advisory Group for Aerospace Research & Development Conference*, number 373: Flight Test Techniques, Neuilly-sur-Seine, France, 1984.
- [16] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. The MIT Press, Cambridge, MA, 1992.
- [17] R. Parasuraman, T. Sheridan, and C. Wickens. A model for types and levels of human interaction with automation. In *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, volume 30,3, 2000.
- [18] Personal Conversation with J. Mendeck, *NASA, Johnson Space Center, Orbit FDO*, January 2006.
- [19] Personal Conversation with J. Mendeck, *NASA, Johnson Space Center, Orbit FDO*, June 2006.
- [20] Personal Conversation with C. D'Souza, *NASA, Johnson Space Center*, March 2006.
- [21] Personal Conversation with J. Brazzel, *NASA, Johnson Space Center*, June 2006.
- [22] Stuart Russell and Peter Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, Upper Saddle, NJ, 2003.
- [23] Alison Cawsey. *The Essence of Artificial Intelligence*. Prentice Hall Europe, Harlow, England, 1998.
- [24] Simon Haykin. *Neural Networks A Comprehensive Foundation*. Prentice Hall, Upper Saddle, NJ, 1999.

- [25] Sarah R. Graybeal and Kara M. Kranzusch. Entry abort determination using non-adaptive neural network for mars precision landers. In *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, volume 8, pages 5846 – 5856, San Francisco, CA, August 2005.
- [26] Gary Riley. Clips: An expert system building tool. In *Technology 2001: The Second National Technology Transfer Conference and Exposition*, volume 2, pages 149 – 158, Washington, DC, 1991.
- [27] D. A. Bader. Project XFAS, an Expert System for Mission Maneuver Scheduling Design: Status Report. Technical Report JSC FM2-87-41, NASA, Johnson Space Center, May 1987.
- [28] Tom A. Mulder. Autonomous rendezvous trajectory planning. *Advances in the Astronautical Sciences*, 79(pt 2):709 – 719, 1992.
- [29] J.-S. Roger Jang and Ned Gulley. *Fuzzy Logic Toolbox User's Guide, Version 2.1*. The Math Works, Inc., Natick, MA, 2001.
- [30] T. J. Ross. *Fuzzy Logic, with Engineering Applications*. John Wiley & Sons Ltd., West Sussex, England, 2004.
- [31] V. Gopalan, A. Homaifar, M. R. Salami, B. Sayyarodsari, and R. W. Dabney. Fuzzy Genetic Controllers for the Autonomous Rendezvous and Docking Problem. In *ACM Symposium on Applied Computing*, number ACM 0-89791-658-1, pages 532 – 536, 1995.
- [32] Charles L. Karr and L. Michael Freeman. Genetic-algorithm-based fuzzy control of spacecraft autonomous rendezvous. *Engineering Applications of Artificial Intelligence*, 10(3):293 – 300, 1997.

- [33] R.N. Lea. Automated orbital rendezvous considerations. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, number 88CH2555-1), pages 1871 – 1872, Philadelphia, PA, 1988.
- [34] R.N. Lea. Applications of fuzzy sets to rule-based expert system development. *Telematics and Informatics*, 6(3-4):403 – 406, 1989.
- [35] R.N. Lea and Y. Jani. Fuzzy logic captures human skills. *Aerospace America*, 29(10):25 – 27, October 1991.
- [36] A. Howard, H. Seraji, and E. Tunstel. A rule-based fuzzy traversability index for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 3067– 3071, 2001.
- [37] Jie Rong. Intelligent executive guidance agent for general aviation aircraft under free flight. Master’s thesis, Aerospace Engineering, Texas A&M University, College Station, TX, August 2002.
- [38] Personal Conversation with J. Mendeck, *NASA, Johnson Space Center, Orbit FDO*, September 2006.

## APPENDIX A

## LIST OF ACRONYMS AND DEFINITIONS

| <b>Acronym</b>         | <b>Description</b>   |
|------------------------|--|
| AI                     | Artificial Intelligence  |
| AR&D                   | Automated Rendezvous & Docking   |
| CEV                    | Crew Exploration Vehicle   |
| CLIPS                  | C Language Integrated Production System, expert system shell.  |
| CSM                    | Command and Service Module   |
| FDO                    | Flight Dynamics Office (pronounced FIDO)   |
| FL                     | Fuzzy Logic  |
| FLOAAT                 | Function-specific Level of Autonomy and Automation Tool  |
| IMU                    | Inertial Measurement Unit, Navigation Sensor   |
| ISS                    | International Space Station  |
| KB                     | Knowledge Base   |
| LM                     | Lunar Module   |
| LOA                    | Level of Automation  |
| LOAA                   | Level of Autonomy and Automation   |
| LVLH                   | Local Vertical Local Horizontal reference frame  |
| MC-1, MC-2, MC-3, MC-4 | Midcourse Correction burns (MC1 - MC4).<br>The MC burns are small burn maneuvers executed<br>between Ti and the manual Proximity Operations phase. |
| MCC                    | Mission Control Center. Location of the spacecraft<br>operations support teams, in Houston, TX.  |
| MF                     | Membership Function, function describing a fuzzy set.  |
| MSL                    | Mars Science Laboratory  |
| NASA                   | National Aeronautics and Space Administration  |

| Acronym  | Description   |
|----------|---|
| NC       | Nth Central phasing burn (NC). The NC burn allows the Orbiter to catch up with the target at the proper rate.   |
| NCC      | Nth Corrective Combination burn (NCC). NCC targets the desired downtrack, out of plane position, and height at a future point (e.g. $T_i$ ).  |
| OMS      | Space Shuttle Orbiter Maneuvering System  |
| OODA     | Observe, Orient, Decide Act   |
| Orbiter  | Space Shuttle Orbiter   |
| PISCES   | Platform Independent Software Components for the Exploration of Space. On-orbit trajectory planning application.  |
| Prox Ops | Proximity Operations, the portion of the rendezvous and docking profile that begins when the range to the target is $< 1000$ ft and LVLH relative velocity is $< 1$ ft/sec. in each axis. |
| RNDZ     | Rendezvous. The rendezvous phase occurs after insertion into orbit following launch and concludes at the Proximity Operations phase.  |
| RCS      | Space Shuttle Reaction Control System   |
| RPOD     | Rendezvous, Proximity Operations, and Docking   |
| RPOP     | Rendezvous and Prox Ops Program   |
| Ti       | Transition Initiation burn ( $T_i$ ). This burn targets the Orbiter for a near-intercept trajectory with respect to the target spacecraft.  |
| TIG      | Time of IGnition  |
| TIG slip | Amount of time that the Time of IGnition for a burn can be delayed before violating a relative motion or propellant constraint.   |



## VITA

Jeremy Jay Hart is the son of Jeffrey and Elizabeth Hart of Minneapolis, Minnesota. Jeremy graduated from the University of Wisconsin-Madison with a Bachelor of Science degree in engineering mechanics in 2000. After working for NASA at the Johnson Space Center in Houston, TX, he entered the Aerospace Engineering Department at Texas A&M University under the supervision of Dr. John Valasek in 2005. Jeremy has returned to work at the Johnson Space Center in the Aeroscience and Flight Mechanics Division.

Permanent Address: Jeremy Hart; 2421 32nd Ave. S.; Minneapolis, MN 55406