ADVANCED ONBOARD SPACECRAFT GUIDANCE AND NAVIGATION CONSOLE

A Thesis

by

NIKITA K. BEEBE

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Gregory Chamitoff
Committee Members,	Ana Diaz Selva
	Casey Papovich
Head of Department,	Ivett Leyva

August 2022

Major Subject: Aerospace Engineering

ABSTRACT

This proposal defines an advanced onboard navigation interface and trajectory design tool for future astronauts. As the space industry expands and becomes more commercialized, future spaceflight will include long-duration exploration to more distant destinations. Mission tasks, such as rendezvous, docking, descent, landing, and in-space trajectory planning will become commonplace. Presently, many of these tasks are performed with the assistance of ground-based mission control. The onboard crew typically has limited control of vehicle guidance and navigation. This is in stark contrast with the more mature commercial and private aircraft industry, for which guidance, navigation and control are operated primarily by the crew. As the space industry continues to expand with more and more space vehicles, it will become necessary and desirable for the crew to have independent onboard guidance, navigation and control capability. It will not be possible nor efficient to have ground-control operations of every space vehicle, particularly those that are distant from Earth. Just as the pilot operates a suite of instruments and controls on a conventional aircraft, new concepts for spacecraft crew interfaces will be needed for future space pilots.

As such, this research proposes the conception and design of an onboard spacecraft pilot interface called the “Spaceflight Console.” The purpose is to provide the crew full autonomy for spacecraft navigation and guidance via touch-controls and holographic visual displays. The objective is to design an intuitive interface for onboard trajectory planning and a wide range of mission tasks that will become more commonplace in the future, such as interplanetary departure, trajectory corrections, orbital insertion, rendezvous, station keeping, landing site selection and targeting, and so on.

The proposed Spaceflight Console will be built and demonstrated using a virtual-reality (VR) engineering-design platform designed by the ASTRO Lab at Texas A&M, called SpaceCRAFT. Using SpaceCRAFT crew interfaces can be designed, evaluated in the context of a mission environment in VR, and revised easily. Control panels for crew selections or data entry will be coupled with 3D visual displays that enable crew situational awareness in an orbital or interplanetary context. Similar to an aircraft cockpit and the usual suite of flight instruments, the Spaceflight Console presents intuitive information to the crew while internally performing complex computations to support the mission tasks. In effect, the Spaceflight Console aims to translate many complex ground-control capabilities into a fully onboard system with a simple and intuitive interface that operates from the pilot's perspective.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Gregory Chamitoff and Professor Ana Diaz Artiles of the Department of Aerospace Engineering and Professor Casey Papovich of the Department of Astrophysics.

The work constructed in this thesis was generated with the resources available by the ASTROLab, spearheaded by Professor Gregory Chamitoff. The work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University and a dissertation research fellowship funded by X-Grant 2020: Texas A&M Lunar Surface Experiments Program (LSEP).

NOMENCLATURE

Acronyms and Abbreviations:

ATC	Air Traffic Control
BFS	Backup Flight Software
CDI	Course Deviation Indicator
CRT	Cathode Ray Tubes
DME	Distance Measuring Equipment
DSN	Deep Space Network
ECLSS	Environmental Control and Life Support System
EFIS	Electronic Flight Instrument System
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
GPC	General Purpose Computers
GUI	Graphical User Interface
HAINS	High Accuracy Inertial Navigation System
HSI	Horizontal Situation Indicator
ICRF	International Celestial Reference Frame
ILS	Instrument Landing Systems
ISS	International Space Station
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LEO	Low Earth Orbit

LVLH	Local Vertical, Local Horizontal
MEDS	Multifunctional Electronic Display System
MLS	Microwave Landing System
NASA	National Aeronautics and Space Administration
OBI	Omni-Bearing Indicator
OBS	Omni Bearing Selector
PASS	Primary Avionics Software System
RMI	Radio Magnetic Indicator
RNAV	Area Navigation System
RPOP	Rendezvous and Proximity Operations Program
SPICE	Spacecraft, Planet, Instrument, "C-matrix", Events
STDN	Space flight Tracking and Data Network
TACAN	Tactical Air Navigation System
UE4	Unreal Engine version 4
VHF	Very High Frequency
VNB	Velocity, Normal, Binormal
VOR	Very High Frequency OmniDirectional Range

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION AND MOTIVATION

For all human-rated spacecraft, NASA requires the spacecraft must be capable of being controlled manually and remotely, regardless of how automated the spacecraft is. “This serves to demonstrate that the crew can control it in the event of an emergency or if there’s an unexpected problem with the automated controls.” [1] However, this requirement only addresses current spaceflight missions. In the long term, as the number of spacecraft grows, autonomous guidance and navigation will become necessary. Dependence upon ground stations will become intractable, and due to communication delays, impossible. Ground control for a spacecraft system involves ground stations, mission control, ground networks, remote terminals, and test facilities in nearly all space missions, whether commercial, military or scientific [35]. Any lag in communication between a spacecraft and ground control can drastically impact a mission’s success. To put this in a wider perspective, aircraft pilots and air traffic controllers today are where astronauts and flight controllers will be in the next hundred years. If a pilot is unable to contact an air traffic controller, their aircraft does not become inoperable, nor is their mission or flight path drastically impaired. With a plethora of spacecraft entering the solar system every day, the ability for mission control centers to monitor, navigate, and guide every space vehicle will become impossible. With human spaceflight becoming as common as commercial airline flights, the need for autonomous navigation will continue to increase. As such, the need for onboard guidance and navigation tools for the crew to execute future missions will become a necessity for space travel.

Modern spaceflight is primarily controlled and directed by ground stations. Crew onboard does not ‘fly’ their spacecraft in the way a pilot might fly a commercial airplane. Furthermore, mission control handles the majority of planning, calculating, and directing the overall mission profile.

The fundamental limitation with how GNC is operated today, is the dependency pilots have to ground control. Crew are expected to manually guide and navigate their spacecraft only in emergency situations or specific validity demonstrations. Outside of these rare instances, mission control dictates the maneuvers and operations of the spacecraft. Not only do crew members have limited control of the spacecraft, they also do not have all the information ground control monitors, further restricting a pilot's situational awareness. This limitation exist across a range of missions, from docking with the International Space Station to immediately replanning a trajectory during a software system failure to descent and landing at a specified destination on Earth's moon.

Take the example of rendezvous and docking. The process currently uses onboard sensors to determine relative position and, for some vehicles, guidance to the docking mechanism is computed onboard. In fact, the shift from radar-based to optical-based sensors for rendezvous was driven by the desire to develop fully autonomous rendezvous, proximity operations, and docking functionality [2]. Ground operations assist with navigation and flight dynamics up to a point, but when dealing with guiding a spacecraft hundreds of kilometers away with the precision of only a few feet to properly dock, onboard sensors are required to accurately determine relative position and attitude.

Until recently, this concept of a rendezvous guidance tool was beyond anything astronauts had available on their spacecraft. On the International Space Station (ISS), crew members do not monitor or do anything to change or control the ISS trajectory or plan trajectories in general. For instance, when an engine fires, the corresponding trajectory was planned by the ground. The crew will monitor attitude and attitude control, but the ISS does not have a unified control panel for guidance (i.e., trajectory control) [32]. For the Space Shuttle, which at its time was the only spaceflight vehicle that even had a form of trajectory planning on board, there were no visual renderings of the orbits or easy ability to plan an orbital maneuver. The most encompassing system was the Rendezvous and Proximity Operations Program (RPOP), which was a 'laptop computer-based relative navigation tool and piloting aid that was developed during the Space Shuttle program.' [33] As depicted below, RPOP would show a graphical representation of the relative motion between two spacecraft for rendezvous and proximity operations.

Prop Age 3

R 220 $\dot{R}$ -0.21

X 220 $\dot{X}$ -0.22
Y 3 $\dot{Y}$ -0.02
Z -5 $\dot{Z}$ -0.03
Orb DP to Tgt DP

Raw TCS 2(CW)
Refl 1
Age 0
Rng 226
Rdot -0.20
Elv -0.71
Azi 0.11

MET: 1/20:57:06
Pitch 181
Alt 187

	Rng	Rdot	Age
HHL/dt	222	-0.26 231	7
HHLRaw	222	-0.24	7

	RESID	RATIO	ACPT	REJ
RNG	-0.0	0.01	389	0
RDOT	0.01	0.07	387	0
ELV	-0.04	0.09	389	0
AZI	-0.04	0.09	389	0

Range 225.4 ft R-Dot -0.20 ft/s Data Good Mode Track

Figure 1: RPOP onboard laptop display showing relative position data and providing 2D visualization of the Space Shuttles approach to docking with ISS [33]

It also displayed data such as the relative position and velocity (in the middle-top of the screen), elevation, pitch, and azimuth parameters for reference. As a navigation (relative motion) tool it was critical, but there were no intuitive interactions with the interface, and it was data only – not a trajectory control interface.

Moreover, there was not one instrument that displayed all the necessary maneuver information, showed a visual representation of the approach, and enabled the crew to enter guidance commands. Instead, the crew would follow a written checklist of actions, while monitoring instruments and manually firing thrusters to make corrections. The most recent display of docking guidance was during the SpaceX Dragon capsule launch to the ISS, wherein the crew guided the capsule using a visual recreation of the spacecraft's approach corridor. A virtual rendering like this provides the beginning foundation of situational awareness to crew. With the Dragon capsule, most of the needed information for proximity and rendezvous guidance is displayed on screen to the crew in an intuitive manner. No communication with the ground is required, and control of the spacecraft is effectively autonomous.

Another past case study would be during events of an emergency or system failure. One of the simplest cases of an emergency is when communication between ground control and the spacecraft is lost. A long communication outage can interfere with safe operation of the spacecraft from mission control. Take for instance the Boeing Starliner, a reusable spacecraft capsule designed to accommodate seven passengers, or a mix of crew and cargo, for missions to low-Earth orbit [4]. The Starliner failed to dock with the ISS in late 2019 due to a faulty onboard timer. This is likely the result of only 'a few errors in over one million lines of code,' but one that caused the system to 'believe it was in a different segment of the mission than it really was,' executing a burn that maintained control and precision rather than the needed engine burn for the target orbit [5]. Further complicating the issue, NASA was unable to communicate with the unmanned spacecraft. As reported, 'NASA attempted to send commands to the spacecraft to rectify the situation, but mission control found itself in a signaling blind spot when it came to accessing the communications satellites in orbit.' Had the capsule been carrying astronauts, they could have taken control to override the automation of the spacecraft and insert it into proper orbit. This would have resolved the first issue, but in the instance of lost communication, crew members need to have an intuitive enough navigation/guidance instrument that can provide the needed commands for the desired trajectory. Specifically, for the Starliner capsule, a replica of the interior control panel is pictured below [36].

Figure 2: Boeing Starliner Capsule Mock-Up Control Panel [36]

Even with this modern display, the control panel itself still resembles a plethora of displays and input controls. Ideally, a trained crew would have been able to understand the error in thruster burns and could have entered the correct maneuver to execute a successful trajectory to dock. Yet the intuitiveness of the shown display can be much improved. Possible improvements this Spaceflight Console hopes to achieve is implementing a mixture of visual displays - touchscreen, manual controls, and holographic representations. Furthermore, limiting the multitude of different screens to only one or two per crew member can reduce an overload of unnecessary data per the mission profile by letting the crew customize what each screen shows to provide only the relevant and needed information for that profile. Not only will this onboard tool be needed in near and future-term, but as can be seen, this type of instrument currently doesn't exist.

Another past example wherein the crew did not have the necessary tools onboard was when executing descent and landing. This applies to targeting and landing on the lunar surface or, in the future, landing on Mars and eventually other planetary bodies, such as moons or asteroids. The first human landing on the moon, Apollo 11, was guided by commander Neil Armstrong and lunar module pilot Buzz Aldrin, who controlled the capsule in the final stages of descent and landing with only rudimentary instrumentation that provided X-Y-Z coordinates. Once their spacecraft entered into lunar orbit and the crew began final preparations for landing, the lunar module (LM) guidance computer relayed program alerts indicating that the guidance could not complete all of its tasks in real-time and had to postpone some of them. [6] Moreover, Armstrong had to look outside the spacecraft's window to determine where the computer's landing target was located, leaving him to then take semi-automatic control when he realized the chosen landing site was set in a boulder-strewn area. Throughout the descent, while guiding the lander to a new location, Armstrong had to rely on Aldrin to call out navigation data in order to land safely. As the spacecraft descended below 100 ft, lunar dust kicked up by the LM's engine impaired Armstrong's ability to see the ground or determine the spacecraft's position. While Apollo 11 successfully landed on the lunar surface, the crew had to rely on limited situational awareness for their spacecraft. A Spaceflight Console can help enhance the crew's knowledge of the environment and perspective of how the spacecraft is operating. This onboard instrument can display the needed trajectory information, such as ground speed, acceleration, thrust, remaining fuel and time to surface, coupled with a 3D graphic of the spacecraft on a guided path to the target landing site would have made the entire Apollo process much more efficient and safer.

II. OBJECTIVES AND CONTRIBUTIONS

The thesis presented aims to detail the specific capabilities envisioned with the Spacecraft Console. This involves explaining the flow process and implementation of the eight mission profiles previously mentioned. Next, a demonstration of the flight deck design and interaction will be shown for a number of the said mission objectives. And finally, a conclusion with future work and current limitations of this first concept. This will explain the objectives that were not demonstrated and will lay out what is left to include for a fully encompassed guidance and navigation console.

Thus, the thesis will present the following objectives:

1. Outline of envisioned capabilities for the spaceflight console
2. Demonstration of some of the detailed capabilities in a range of mission profiles
3. Detail the future work to create a conclusive version of the console

Section 1: Objectives

Such a Spaceflight Console should at a minimum encompass the following objectives for onboard crew to utilize; Planetary Orbital Maneuvers, Rendezvous and Docking, Interplanetary Target Planning, Interplanetary General Guidance, Gravity and Flyby Assist , Planetary Capture and Orbital Insertion, Aerocapture and Aerobraking Maneuvers, Remote Scanning and Prospecting, Descent and Landing. Each capability takes into account the crew point of view, what their inputs and outputs would be, and how to convey that information.

Beginning with planetary orbital maneuvers, a pilot might want to change their orbital inclination or altitude above a given central body, like depicted below.

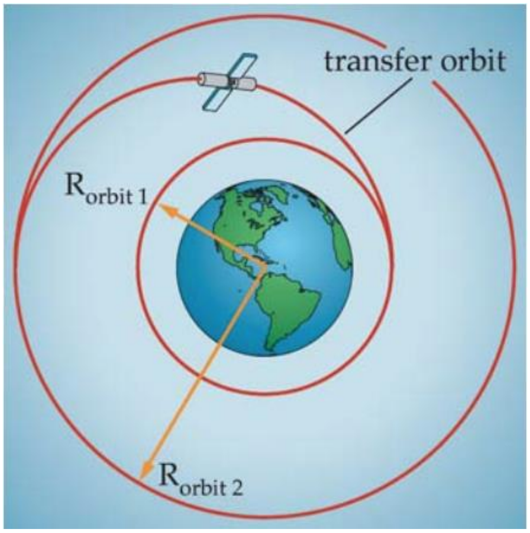


Figure 3: Maneuver Change in Orbit About Earth

From a high-level perspective, the crew might input the ideal maneuver location to execute, the new targeting trajectory elements or the general purpose of the new orbit. Additional inputs can include maximum fuel allowance or timeframe of when to apply the maneuver.

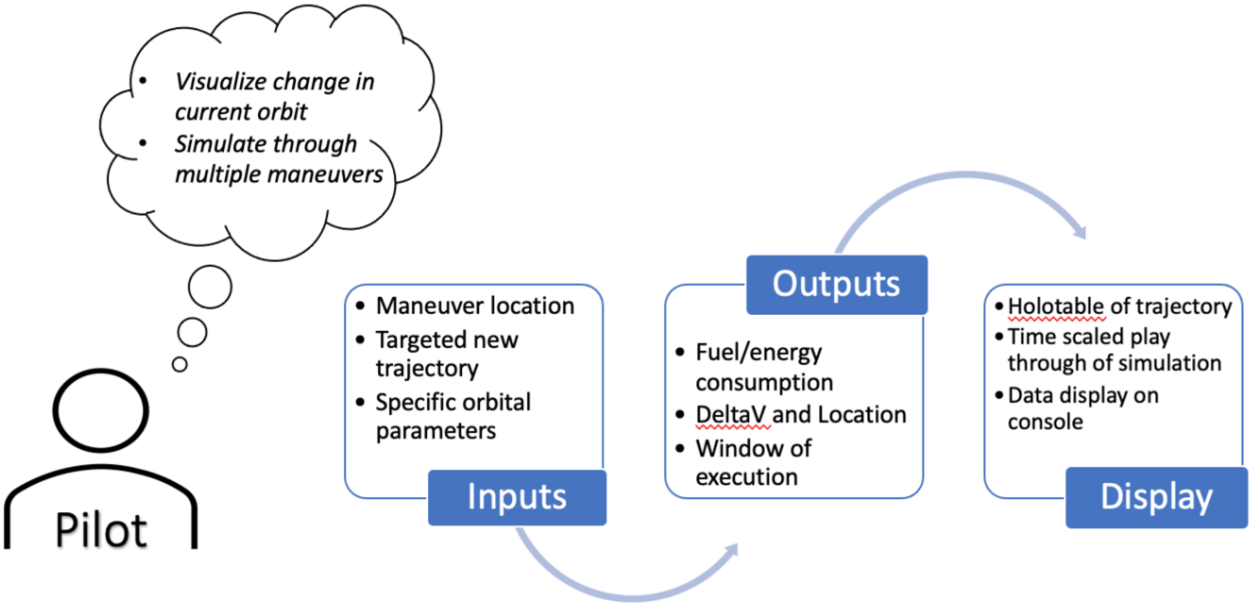


Figure 4: Planetary Orbital Maneuvers Breakdown Visual

Using the inputs, a pilot can obtain the required deltaV or number of maneuvers to accomplish their targeted trajectory. They can also view a 3D plot or heat map of a range of options for a given timeframe, factoring in the different deltaV or fuel consumption based on when they execute the maneuver. These data can also then be displayed holographically as a simulated play through of each maneuver option to give the crew a full understanding of the change in paths without having to actually execute the maneuver.

Another capability a pilot would need is to rendezvous and dock with another vehicle or point target, like shown in the following diagram.

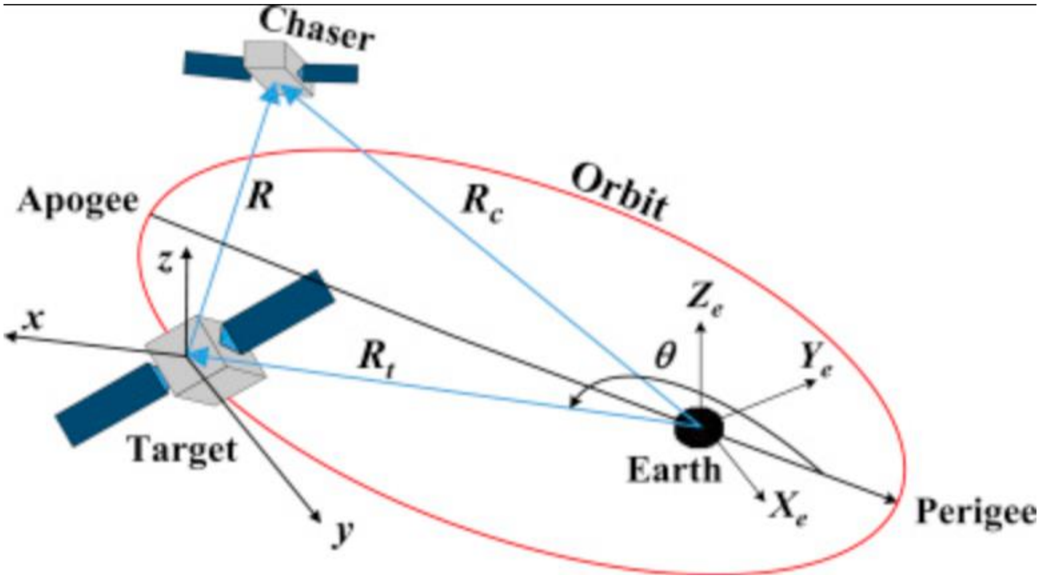


Figure 5: Rendezvous Maneuver with Target and Chaser About Earth Orbit

Rendezvous with an option to dock would be a necessity for future manned missions, even in the case that the rendezvous point is not necessarily another vehicle. From the pilot's perspective, they may simply want to input the targeting vehicle without needing to know prior information about its location or velocity or trajectory.

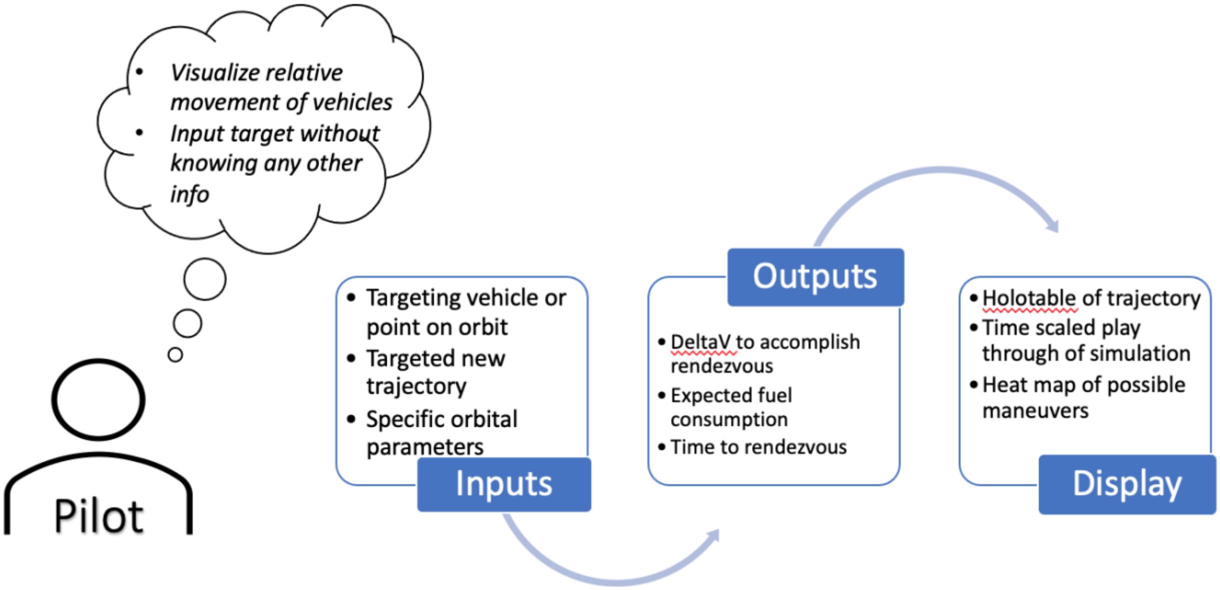


Figure 6: Rendezvous and Docking Breakdown Visual

Based on inputs of the vehicle, known trajectory information, or if there are constraints on the timeline to rendezvous, the outputs can include the expected fuel consumption and specific deltaV pertaining to each possible maneuver option. Then for a range of option, this can also be displayed on a heat map where pilots can select from a specific collection of data values and visualize the resulting maneuver and trajectory on a holographic display along with the pertaining mission details.

Interplanetary trajectory planning, primarily with optimized lambert algorithms, can be used by pilots to plan a direct transfer from their spacecraft to any targeted body or solar coordinate.

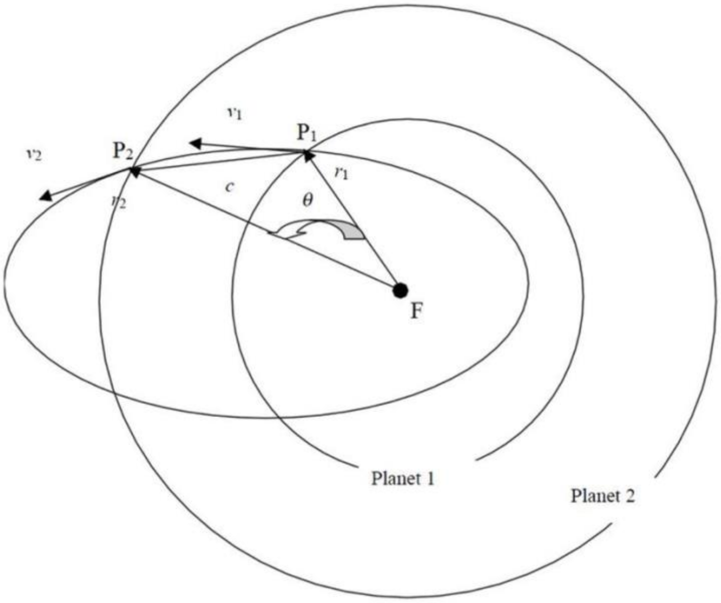


Figure 7: Geometry of Lambert Targeting Trajectory Maneuver

Crew can input the targeted body as well as any constraints on their mission profile, such as amount of fuel expenditure or time of flight. Based on the data entered, a pilot can then view a range of possible maneuver paths and specific values like the deltaV and true anomaly to apply at for each possibility.

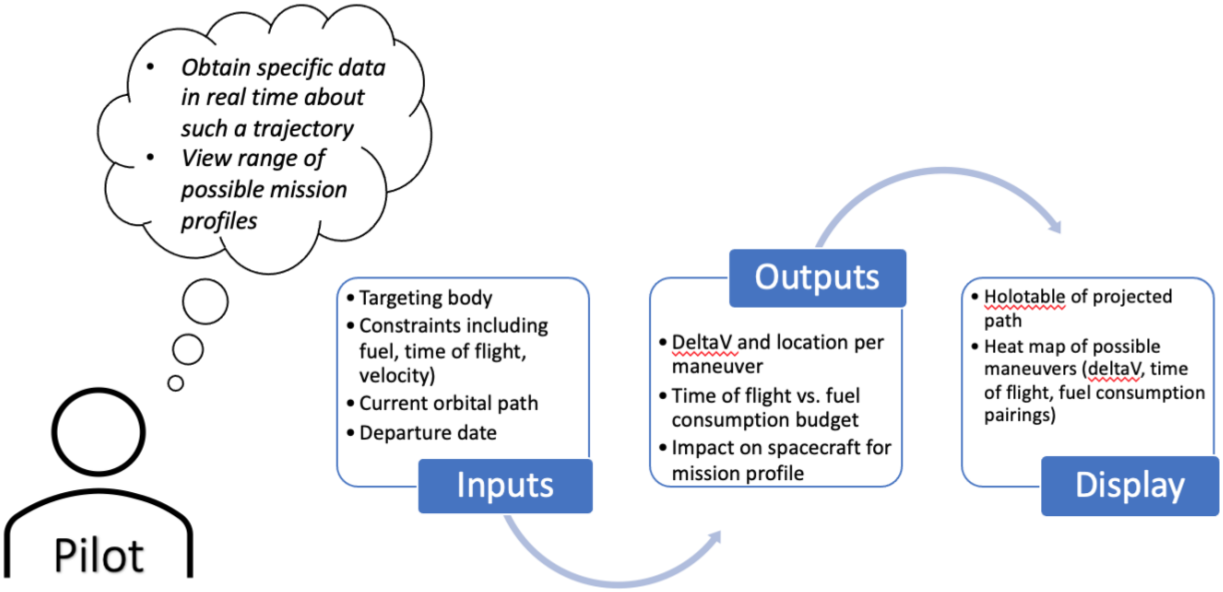


Figure 8: Interplanetary Target Planning Visual Breakdown

Each mission output of deltaV or expected impact on the spacecraft for a long duration flight can be displayed in real time on the pilot's console. Having the onboard console allows crew to recalculate iteratively for different maneuver options, and then visualize each projected path on the holotable to further aid in their visual understanding of the entire mission path.

Aside from specific interplanetary targeting, crew also has the ability to envision ideal conditions of a trajectory, namely planetary alignment or requirements for minimum/maximum variable inputs. Such an example would be a Hohmann transfer, wherein the location and timing of an impulse burn and the orbital paths of the bodies involved is very restricted.

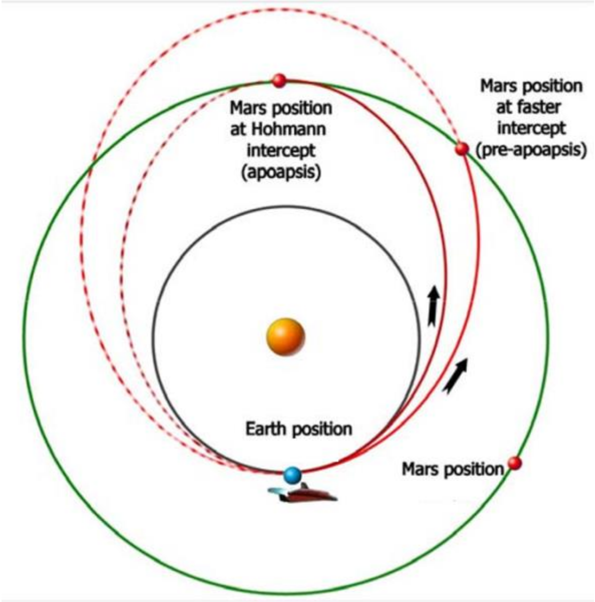


Figure 9: Hohmann Transfer Alignment of Earth to Mars Trajectory

To help the pilot simply understand the alignment of the bodies involved and the conditions to meet an optimized design, the inputs from the pilot can be very minimal.

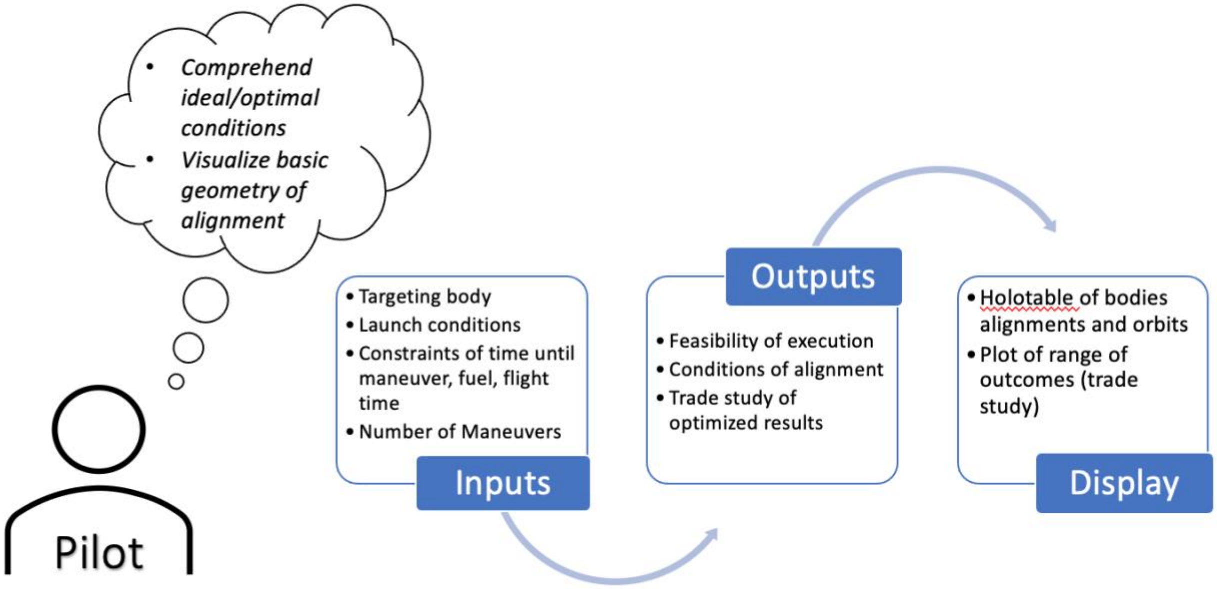


Figure 10: Interplanetary General Guidance Breakdown Visual

From a general idea, pilots can then view how feasible such an orientation would be for the planets and vehicles and analyze a trade study for any input constraints. This can be relayed to crew in plots and data tables on their console, as well as holographically to compare the optimal conditions to the actual maneuvers they would execute.

Once an interplanetary trajectory has been chosen, a pilot can also determine the maneuvers to insert into a parking orbit about a targeted planet. As shown in the image below for an orbital insertion, crew would have the ability to select arriving conditions far out in advance of physically entering into the planets sphere of influence.

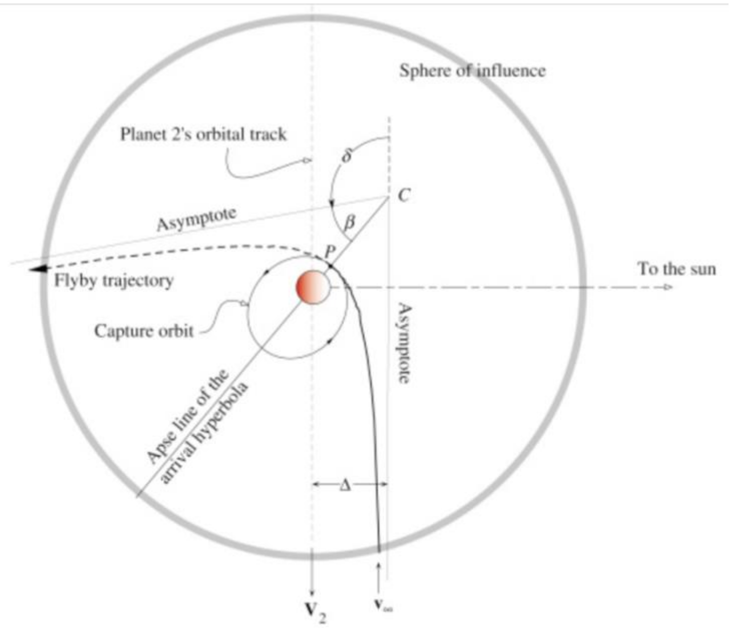


Figure 11: Projected arrival hyperbola into parking orbit about planet

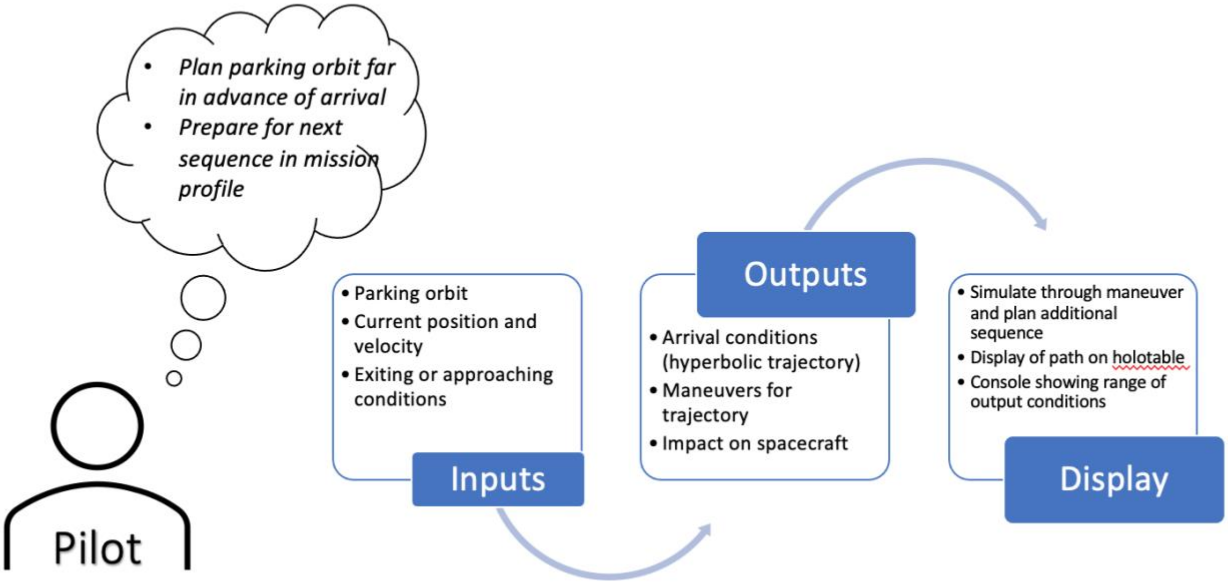


Figure 12: Planetary Capture and Orbital Insertion Breakdown Visual

With this tool a pilot can then sequence additional maneuvers after their arrival conditions are projected. This allows for future planning of either remote sensing about the planet, an eventual descent to land, or to simple orbit the planet and exit on another interplanetary hyperbola path. The information pertaining to the maneuver, like deltaV or impact on spacecraft entering into the sphere of influence can be shown on the pilot's console while the holotable can simulate the motion of the spacecraft from their current location and into the selected parking orbit.

Along with theorizing ideal conditions, a pilot might want to determine if a gravity assist, or flyby trajectory would be possible to help conserve energy and provide an alternative option to evaluate.

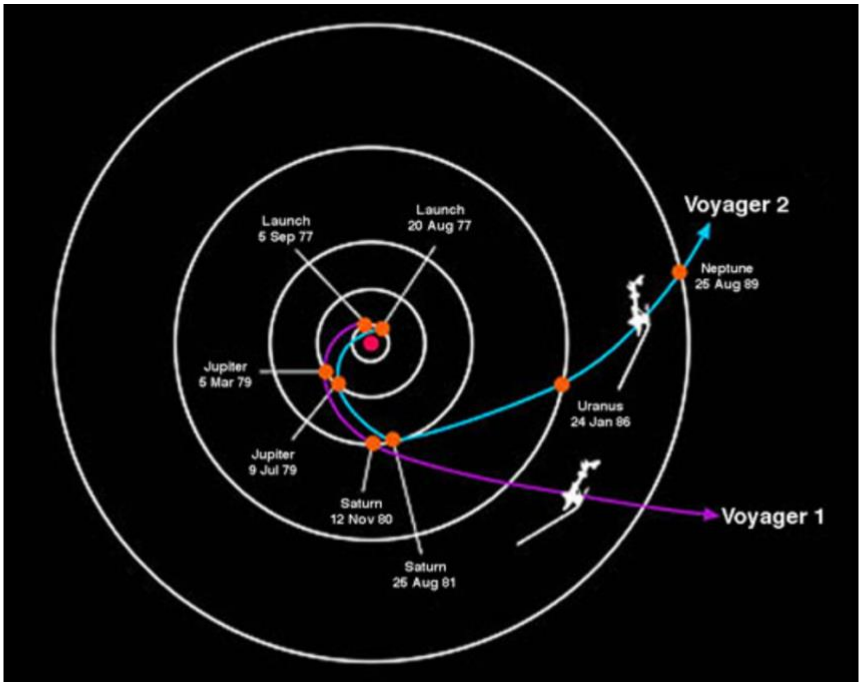


Figure 13: Voyager Gravity Assist Trajectory Comparisons

Gravity assist have not been executed for manned missions and are typically not done post-launch. But from a crew's perspective, they might want to explore the option of an assist for conservation of resources.

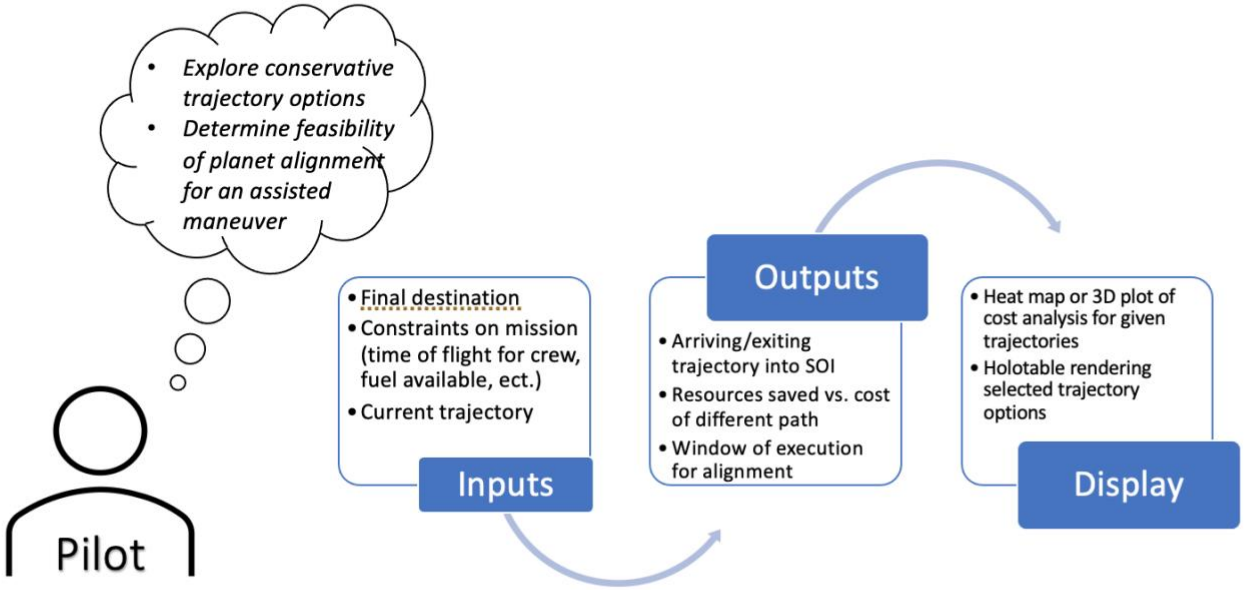


Figure 14: Gravity Assist Visual Breakdown

Based on their input, crew can calculate the possibility of a gravity assist while on their current trajectory. They can view information regarding when to apply an impulse burn, how the new trajectory compares against their current path, and visualize the entire mission execution played out on a holotable at sped up timescales.

Another option a pilot can explore is aerobraking or aerocapture trajectories, as depicted in the following figure comparison:

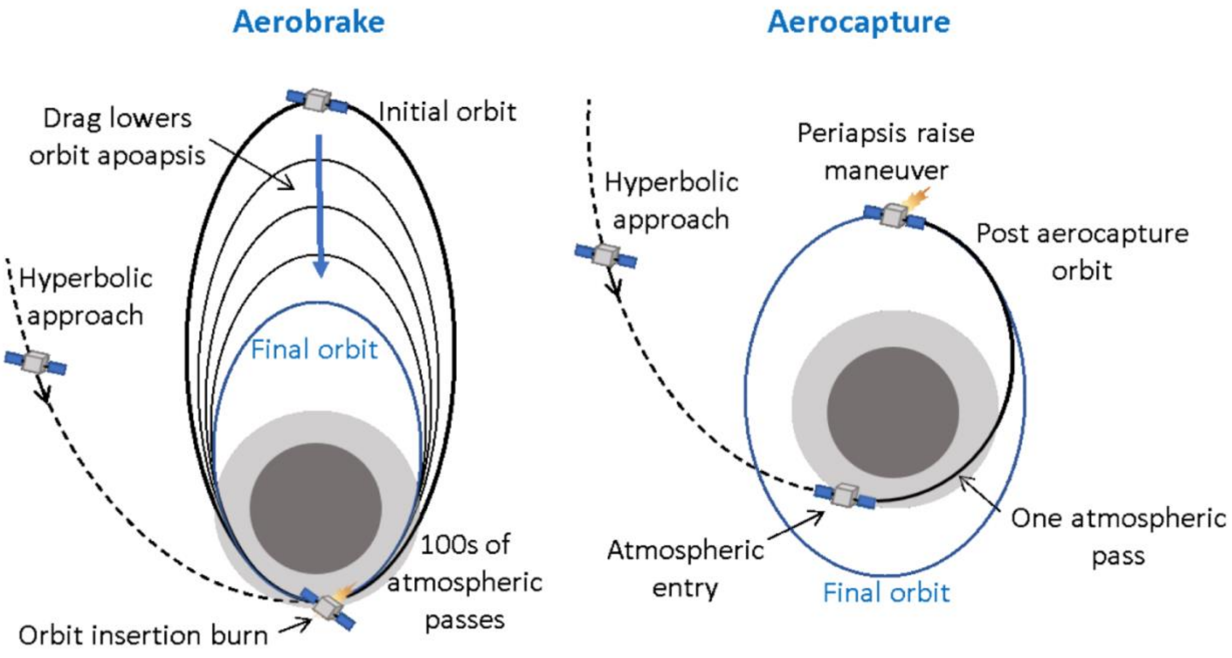


Figure 15: Comparison of Aerobraking vs. Aerocapture maneuvers into a final orbit

While aerobraking has not been executed with a manned mission, future flights and pilots should have the option to evaluate the restrictions and feasibility of the different insertions.

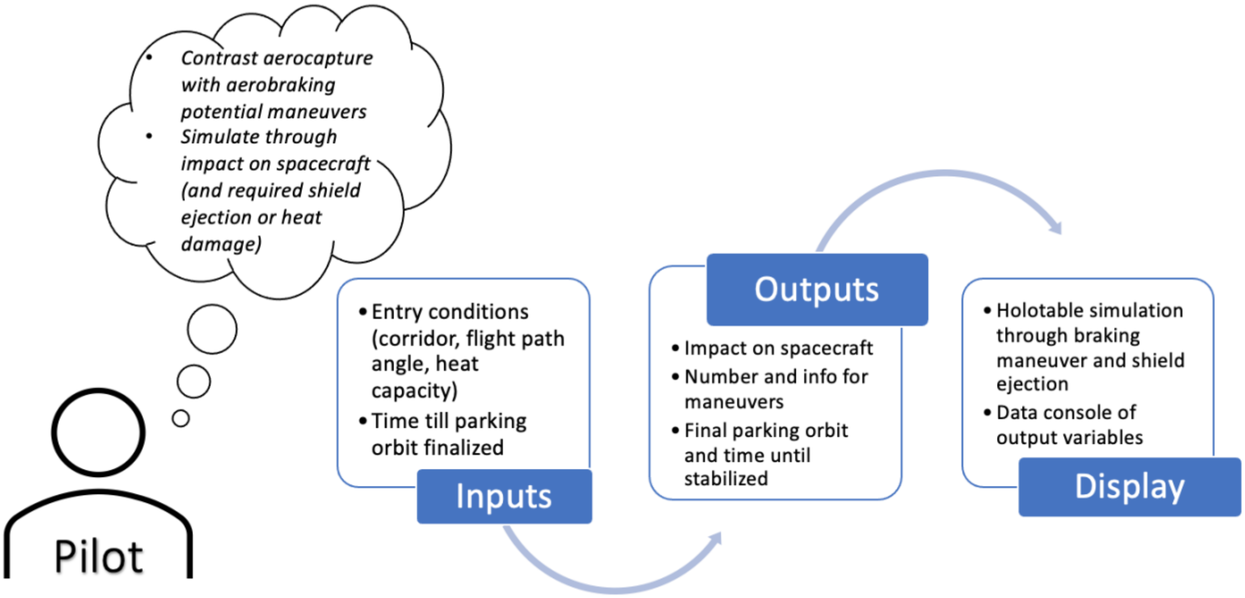


Figure 16: Aerocapture/Aerobraking Visual Breakdown

They can input a range of entry conditions, like a targeted entry corridor or flight path angle to target, and then evaluate the returning trajectory. Along with an orbit insertion, the pilot can simulate through ejecting a protective heat shield for an aerocapture and play through the resulting orbital path. Planning and visualizing these options, as well as quickly recalculating their options, provides the crew with additional maneuvers outside of traditional orbit insertion designs.

With a targeted body or parking orbit selected, crew can initialize remote sensing, as depicted in the following impression.

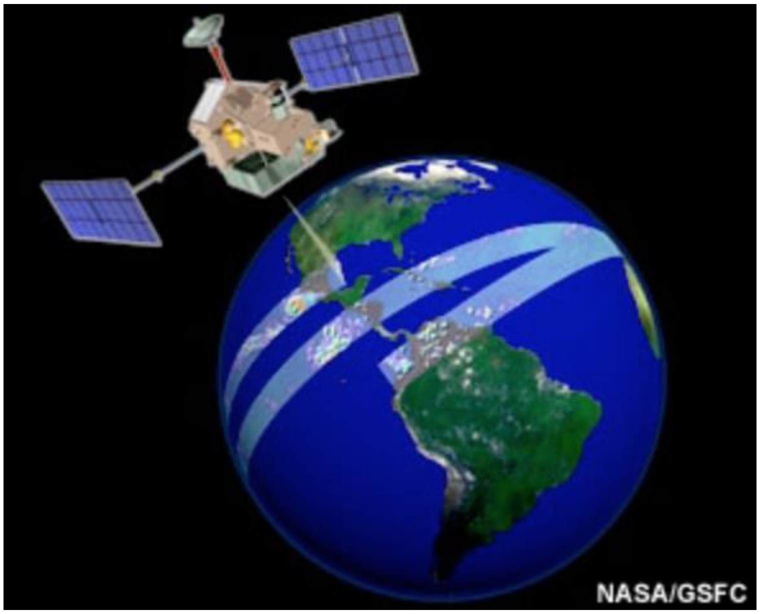


Figure 17: Artistic Impression of TRMM Satellite and Orbit Swath Across the Tropics

Planning out the orbit swath prior to inputting the maneuver allows a pilot to better determine if the selected trajectory will provide them with the necessary data.

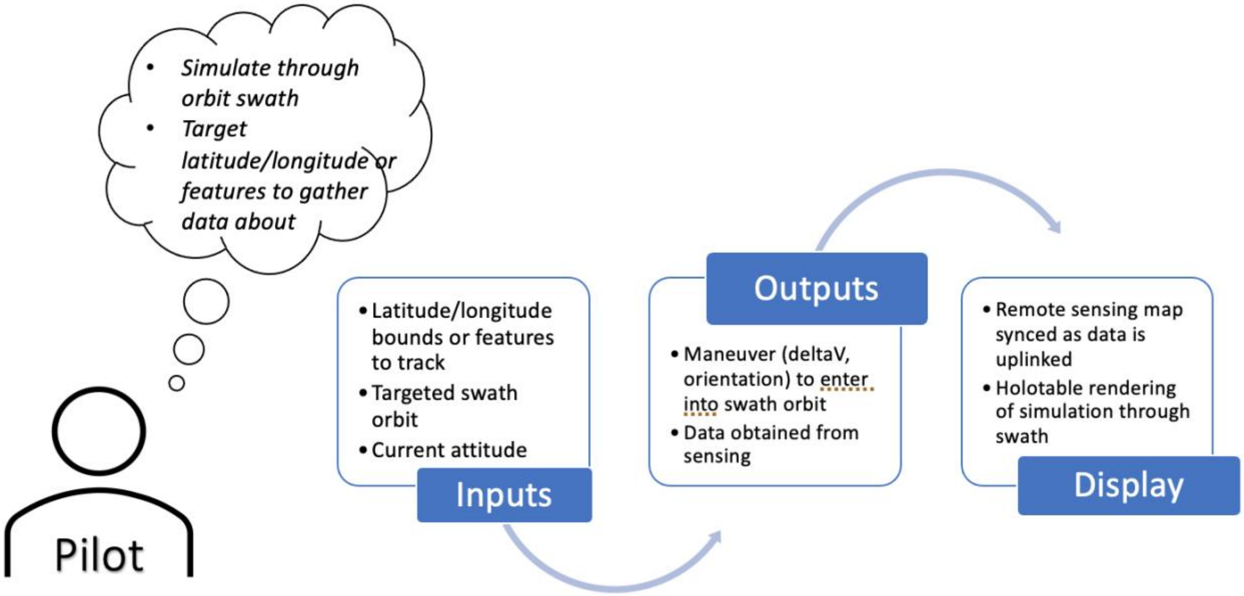


Figure 18: Remote Sensing Visual Breakdown

Pilots can plan out the expected remote sensing by choosing latitude and longitude bounds or specific landmarks to target. From a few inputs, the crew can obtain the resulting maneuver and orbital path, rendered either on the holotable or with specific data shown on their consoles. From the range of information given to pilots, they can easily recalculate different paths and compare expected data gathered without having to execute a prior maneuver.

Nearing the end of a trajectory, pilots can map out an entry, descent, and landing maneuver that their spacecraft will follow.

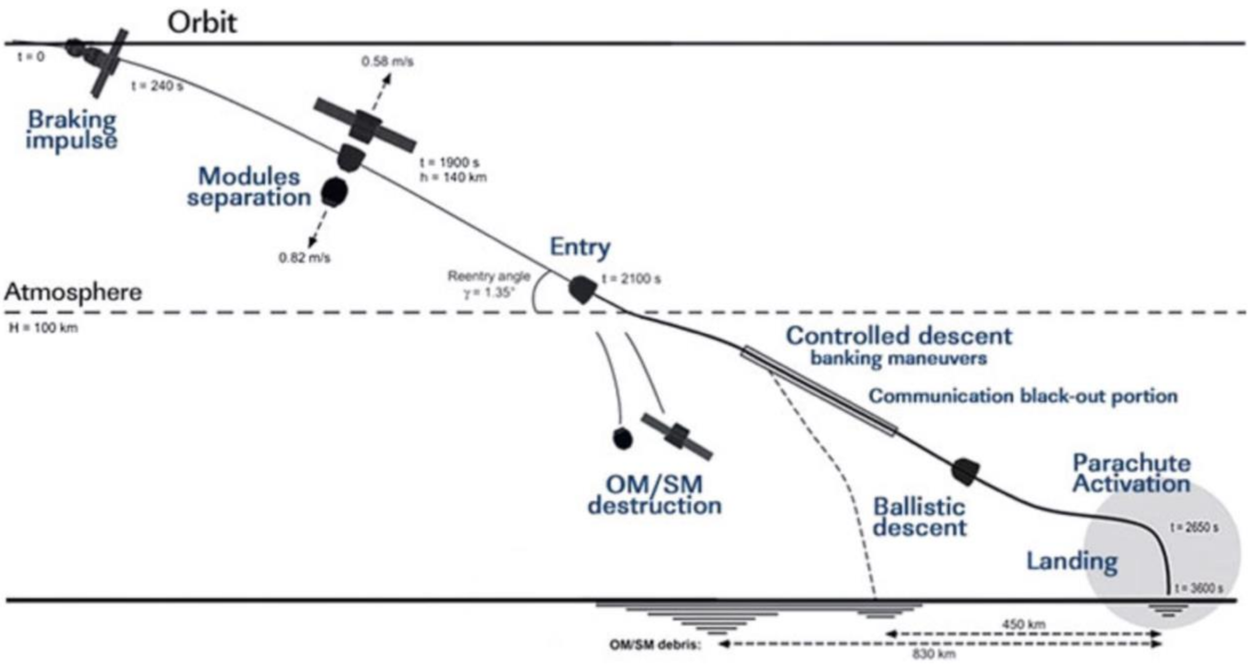


Figure 19: Entry, Descent, and Landing Trajectory Planned Path of Soyuz Craft

Even though the final descent and landing is typically autonomous, crew can input a different landing site than originally planned prior to descent and recalculate a new trajectory path to follow.

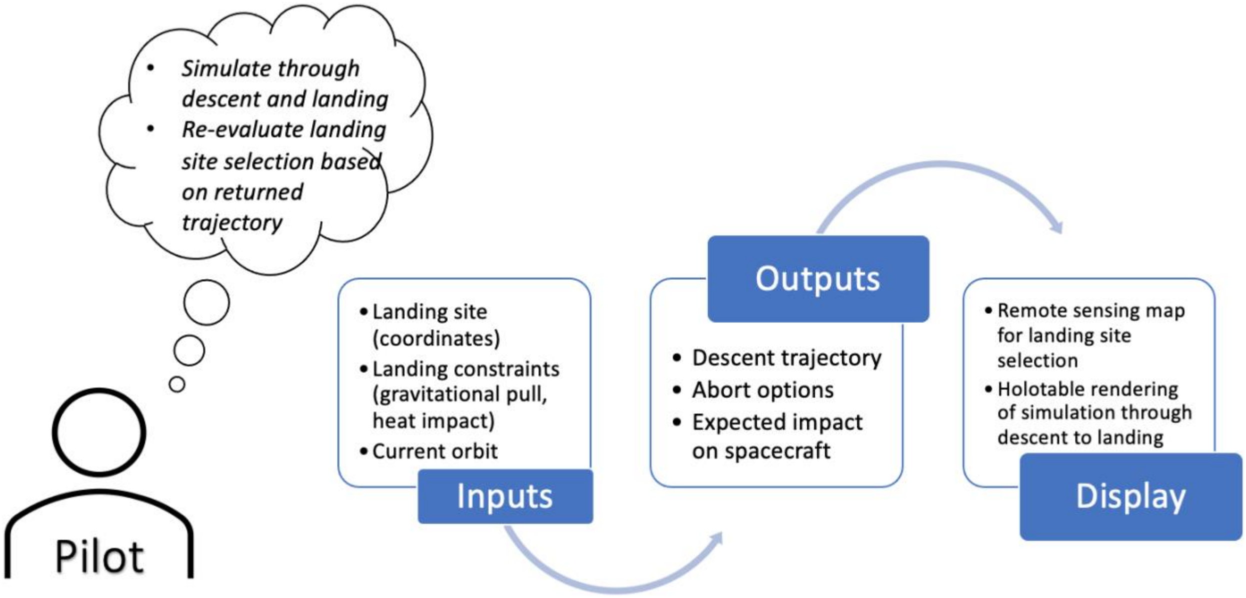


Figure 20: Descent and Landing Visual Breakdown

This gives pilots the flexibility to account for different constraints and to evaluate the impact on the spacecraft or determine what abort conditions might exist. This can be conveyed in plots and numerical figures on console, while also allowing crew to watch a real-time simulation of the landing maneuver to aid in preparing the crew for the final descent.

Section 2: Contributions

The primary contributions of the spaceflight console concept are;

1. Visualization - Future spacecraft piloting scenarios (solar-system wide) and the tools and interfaces that will be most useful for a range of mission profiles.
2. Fully Onboard Operations - Real-time planning and replanning of maneuvers that a spacecraft pilot will need to execute based on mission goals and in-situ considerations.
3. Conceptual Design - An advanced onboard spaceflight control console and holographic display for 3D situational awareness of guidance, navigation, and control.
4. Multidimensional Maneuver Optimization - Tools to aid in pilot decision making based on real-time constraints and objectives.
5. VR Simulation - Implementation of a future spacecraft cockpit for concept evaluation, testing, and training.

For future spacecraft piloting scenarios, whether this is within a sphere of influence or in a solar orbit, pilots will need the tools and interfaces that will be more useful for their given objectives, again, from a pilot’s perspective. This encompasses the visualization, both with a hologram and touchscreen displays, that the Spaceflight Console aims to provide.

By implementing this console onboard, all the operations crew would perform are completed independent on ground control. This allows real-time planning and replanning of maneuvers that a pilot will need to execute based on their mission goals and in-situ considerations. Thus, this console operates fully onboard without additional mission control communications.

Moreover, constructing the demonstrated console in Unreal Engine allows the design to be conceptual and more advanced than what can quickly be physically built. An onboard spaceflight control console can encompass a mixture of holographic projections, touchscreen displays, and manual control sticks. This conceptual design provides both the relevant data and 3D situational awareness of guidance, navigation, and control.

A further contribution of the proposed console addresses the optimization provided to crew members. Providing interactive plots to conduct trade-off analysis from the pilot's perspective and further allowing comparison studies between different methods of maneuvers allows multidimensional maneuver operations.

Lastly, this console has been constructed in a virtual reality (VR) environment. This helps in demonstrating the implementation of a future spacecraft cockpit for concept evaluation, testing, and training. VR enables quick redesigns of the console based on feedback from users and has the capability to coach and test future pilots.

III. BACKGROUND

The foundation of onboard spacecraft navigation and guidance instruments stems from the design of how airplanes are controlled. Aircraft navigate autonomously in the sense that all sensors and equipment needed to determine the position and attitude is onboard. The Air Traffic Control (ATC) towers monitor the flight path of aircraft, but ATC is not needed to navigate or guide the airplane. Instead, pilots rely on radio navigation systems such as the Very High Frequency OmniDirectional Range (VOR) for civilian aircraft or the Tactical Air Navigation System (TACAN) for military aircraft, as well as the Instrument Landing Systems (ILS) that enable aircraft to land on a runway safely even with extremely limited visual contact. [11]

Both VOR and TACAN use fixed ground-based beacons that provide the pilot with bearing and distance to a ground or ship-borne station. [12] The primary difference between the two is that TACAN is a more accurate version of VOR. TACAN provides an operational 3-fold increase in accuracy since it makes use of a two-frequency principle, with 15 Hz and 135 Hz components, and because Ultra High Frequency (UHF) transmissions are less prone to signal bending than Very High Frequency (VHF) transmissions that VOR uses. TACAN also provides ground speed and time-to-station data displayed to the pilot. But in many cases, VOR stations have collocated distance measuring equipment (DME) or TACANs in combination, resulting in either VOR-DME on civilian aircraft or VORTAC on military aircraft.

All this beacon data is conveyed to the pilot onboard through a variety of indicators. The most common indicators are an omni-bearing indicator (OBI) that consist of a knob used to rotate the omni bearing selector (OBS) and the course deviation indicator (CDI) that set the course of the aircraft. Other indicators include a radio magnetic indicator (RMI) that features a course arrow superimposed on a rotating card that shows the aircraft’s current heading, a horizontal situation indicator (HIS) that combines heading information with the navigation display to provide a simplified moving map of the aircraft, and an area navigation system (RNAV) that can provide an up-to-date navigation database through an onboard computer display.

A TACAN indicator or display, as shown below for a Boeing electronic flight instrument system (EFIS), can include additional parameters such as the slant range distance to the TACAN or DME ground station, an alphanumeric ground station identifier, ground speed or time to station, bearing to or from the TACAN station, and the selected TACAN channel or paired VOR frequency the pilot is currently operating on (managed via the selector switch). [13]

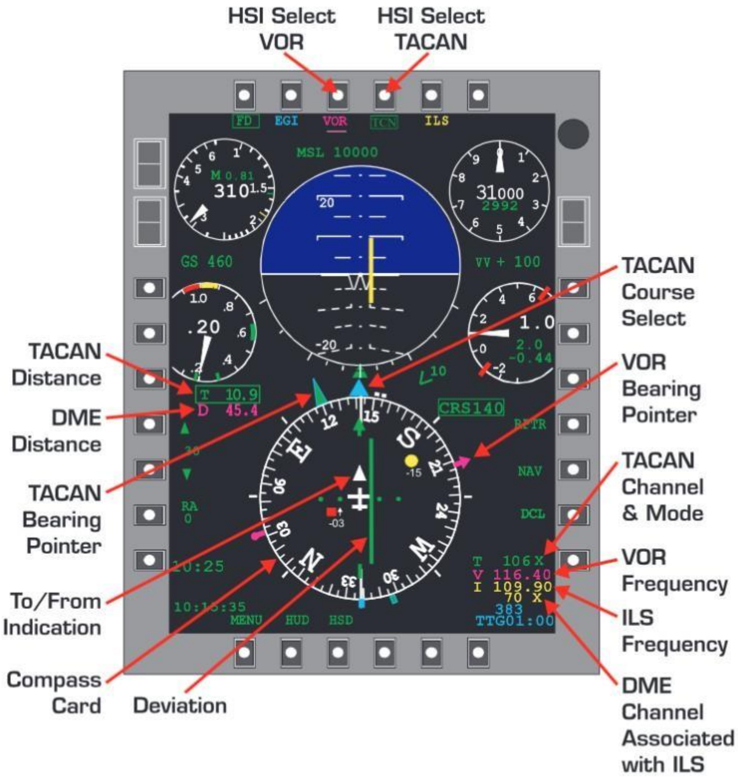


Figure 21: TACAN Information on a Boeing EFIS displaying parameters of distance, frequencies, and bearings to provide the pilot with a comprehensive understanding of their flight path [13]

Translating this to the Space Shuttle navigation required slightly different displays and parameters, although the Space Shuttle was originally designed to use TACAN navigation before upgrading to Global Positioning System (GPS) as a replacement. This occurred in the late 1990's and early 2000's, when the introduction of GPS promised to ensure better performance and reduced operating cost than TACAN. This is due in part because GPS can provide geolocation and time information to a receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more satellites. Further, GPS does not require the user to transmit any data, and it operates independently of any telephonic or internet reception.

Using GPS provided the Space Shuttle with the information to know where it was and how fast it was going. This data was then fed into the flight computers for rendezvous and docking maneuvers and provided mission control with the position of the spacecraft. Communication to ground control included information about 'orbiter operating conditions and configurations, systems and payloads' that is then printed out on the orbiter's teleprinter or text graphics system for the flight crew onboard. [REF]

Specifically for the Space Shuttle, direct communication occurred via a ground network and a deep space network (DSN). The ground network, maintained by Goddard Space Flight Center, included the Space flight Tracking and Data Network (STDN) ground stations for NASA missions. Direct signals of communication from the ground to the Space Shuttle were called uplinks, and signals from the Space Shuttle to ground were called downlinks. The Space Shuttle communication system was further divided into several smaller systems to provide information transfer within the vehicle itself and from the vehicle to ground.

The other communication network is NASA's deep space network (DSN), which is an international network of large antennas that provide the ability for the ground to communicate with satellites and other spacecraft missions, as well as provide radio and radar astronomy observations for deep space exploration. The DSN is composed of three deep-space facilities places approximately 120 degrees apart across the globe to provide constant observation of a spacecraft as the Earth rotates. This ensures there is always a ground station that can send and receive signals at any point in a spacecraft position. The DSN thus provides ‘the vital two-way communications link that guides and controls these spacecraft and collects images and scientific information sent by them. Among other things, the DSN make is possible to; acquire data from spacecraft, transmit commands to spacecraft, …[and] track spacecraft position and velocity’. [15]

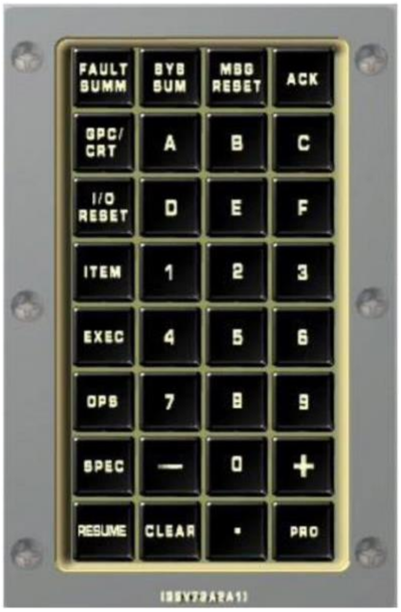
Yet communication between either the ground network or the deep space network is poised for potential failures that can leave a spacecraft without the ability for navigation and guidance. Even during nominal operations, latency already poses a challenge for distances beyond the moon. For travel to Mars, the communication delay is between 4 minutes at closest approach to 24 minutes at farthest approach. In other words, the astronauts would need to wait between 4 and 24 minutes for their messages to reach mission control, and another 4 to 24 minutes to receive a response. This time lag interferes with real-time guidance of a spacecraft, especially if the mission profile changes rapidly and the crew needs an immediate action. Aside from general latency, as the time between transmissions increases the quality of the data can also deteriorate due to solar interference. This includes radiation from the sun or celestial bodies. Signals can also be affected, though to a smaller degree, by other forms of energy from outside the galaxy, such as pulsars or quasars, general noise along its transmission path, or electromagnetic interference.

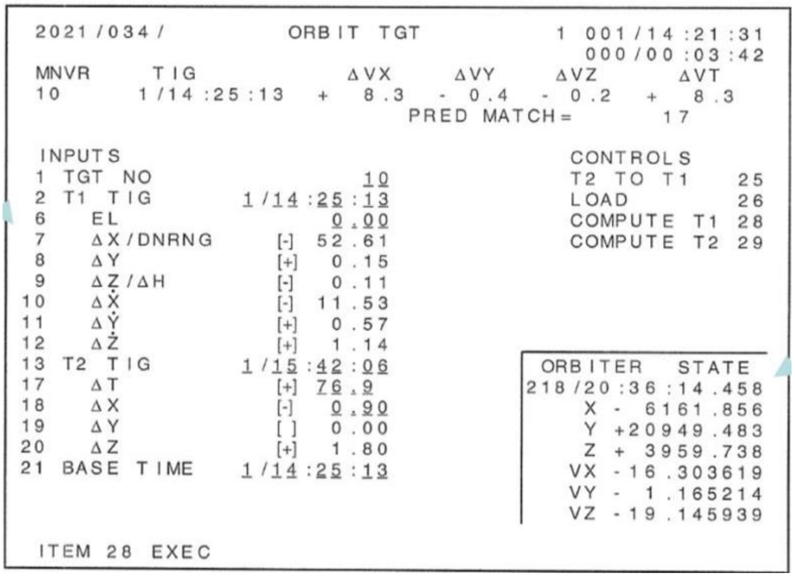
This time lag assumes nominal operations. If the DSN were to encounter an error or component failure on-orbit, it would be impossible for ground engineers to repair anything that breaks. If the communication system breaks down completely, no data would be received, and we could lose all forms of contact. [15] This could result from a small error line in code, or unforeseeable impacts with tiny space debris collisions. In short, while communication is embedded into the current way missions operate, there is much potential to improve on the manner in which spacecraft can operate and communicate as needed.

In a more general sense, the navigation system of the Space Shuttle for other maneuvers and mission profile needs were also onboard. [17] This system architecture encompassed redundant computers to handle the event of a computer failure, namely five different Guidance, Navigation, and Control (GNC) computers onboard the Space Shuttle, also referred to as General Purpose Computers (GPC). Four of those computers contained the Primary Avionics Software System (PASS), and a fifth ran the Backup Flight Software (BFS) that contained a subset of PASS functionality that enabled the vehicle to finish nominal ascent, an abort, or landing in the event of a generic PASS software failure. [18] The Space Shuttle also included accelerometer and gyro inertial measurement units (IMU), entry navigation sensors including High Accuracy Inertial Navigation System (HAINS) IMU’s, barometric altimeters, and Microwave Landing System (MLS) units. Additional radar altimeter data was used for pilot situational awareness but was not incorporated into the Space Shuttle navigation state. All of these systems integrated together to construct a three step GNC process: guidance equipment and software first computed the orbiter location required to satisfy mission requirements, navigation then tracked the vehicle's actual location, and flight control then transported the orbiter to the required location.

This GNC system consisted of two operational modes: auto and manual. ‘In the automatic mode, the primary avionics software system essentially allows the GPCs to fly the vehicle; the flight crew simply selected the various operational sequences. The flight crew may control the vehicle in the control stick steering mode using hand controls, such as the rotational hand controller, translational hand controller, speed brake/thrust controller and rudder pedals. The translational hand controller was available only for the commander, but both the commander and pilot had a rotational hand controller. The vehicle controls included attitude processing, steering, thrust vector control, and digital autopilots. This flight control received vehicle dynamics commands from the guidance software or flight crew controllers, and then processed the commands into effector commands such as engine fire or gimbal changes.

All of this navigation data and guidance was not displayed to the crew on one single screen or in a manner that was easy to read and interpret. Take the objective of lambert targeting for example. This type of algorithm took in two position vector states and a time of flight to then calculate an orbital trajectory path from the initial position to the final position. While this is considered an orbit determination problem, it can also be used as a rendezvous or intercept technique. [19] The current design onboard the Space Shuttle is shown below (figure on the right), wherein a list of initial position and times are entered with a manual keypad (figure on the left), and the resulting orbiter [Space Shuttle] state to achieve that target is displayed.





```
2021/034/              ORBIT TGT              1 001/14:21:31
                                                000/00:03:42
MNVR      TIG              ΔVX      ΔVY      ΔVZ       ΔVT
10      1/14:25:13      +   8.3   -  0.4   -  0.2   +   8.3
                                 PRED MATCH=         17

 INPUTS                                       CONTROLS
 1 TGT NO                       10            T2 TO T1     25
 2 T1 TIG               1/14:25:13            LOAD         26
 6   EL                       0.00            COMPUTE T1   28
 7   ΔX/DNRNG           [-]  52.61            COMPUTE T2   29
 8   ΔY                 [+]   0.15
 9   ΔZ/ΔH              [-]   0.11
10   ΔẊ                 [-]  11.53
11   ΔẎ                 [+]   0.57
12   ΔŻ                 [+]   1.14
13 T2 TIG               1/15:42:06          ORBITER   STATE
17   ΔT                 [+]  76.9          218/20:36:14.458
18   ΔX                 [-]   0.90               X -  6161.856
19   ΔY                 [ ]   0.00               Y +20949.483
20   ΔZ                 [+]   1.80               Z +  3959.738
21 BASE TIME            1/14:25:13              VX -16.303619
                                                VY -  1.165214
                                                VZ -19.145939

 ITEM 28 EXEC
```

Figure 22: Space Shuttle Keypad and Terminal for calculating general lambert problem for a given target [19]

None of this included any type of visual trajectory for the resulting orbiter state and does not include additional parameters such as fuel consumption or alternative velocity burns that can be used for a more optimal orbital path. For rendezvous and proximity operations, crew members are forced to rely on visual monitoring, like aft and overhead windows or closed-circuit television to guide their spacecraft.

This type of rudimentary design for onboard guidance has already begun to see updates and changes within the public and private industry. Since the Space Shuttle as designed in the 1970's, it was first upgraded with the Multifunctional Electronic Display System (MEDS) that helped to 'remedy the obsolescence of the original cockpit components', like replacing electromechanical gauges and cathode ray tubes (CRTs) with color LCD screens to improve the reliability and maintenance of the onboard displays. However, MEDS did not resolve the human factors drawbacks of the legacy cockpit displays since the upgrade was primarily driven by 'concerns over hardware obsolescence and maintenance, [so] few human factors limitations of the original design were addressed.' [34] This led to another upgrade, deemed the 'cockpit avionics upgrade' that aimed to 'redesign the displays to improve situation awareness, reduce workload, and improve performance.' This included factoring in color principles 'to enable the crew to differentiate classes of data and information, particularly during off-nominal conditions', graphic principles that are 'constructed from simple but effective symbologies representing components such as valves, pipes, and tanks' with the goal that the resulting display would 'match with the operator's mental model or system structure and system functioning'.

In summary, the two upgrades sought to consolidate the displays shown to crew and thus reduce the workload required to build situational awareness of system functioning and capability. The final updated Space Shuttle cockpit is shown below.

Figure 23: Updated Space Shuttle Endeavour cockpit with LCD screens [33]

Regardless of the modifications, the number of screens, control buttons, and monitors the crew is responsible to keep track of is still scattered throughout the flight deck, leaving no singular control panel that allows the type of guidance and navigation aid modern and future space flight will require.

In contrast to the public industry, the private sector has seen the type of upgrades more in line with what future instrumentation will encompass. The SpaceX Dragon capsule shown previously is one such example of an innovation to the flight deck crew interacts with. SpaceX is exploring the use of touchscreens and computer displays to convey the necessary flight information and control options to the crew. As shown below, three screens provide the crew with various options and displays. [20] This can include a system overview of their spacecraft, a virtual rendering of the Earth's surface they are orbiting directly over, a control interface to guide the spacecraft during docking or maneuvers, and a map of their spacecraft on their chosen trajectory.

Figure 24: SpaceX display of Earth surface and spacecraft components from pilot's chair point of view [20]

Figure 25: SpaceX display of flight deck for trajectory and control panel [20]

This more futuristic design is not the only type of flight deck being constructed currently. Boeing has announced new updates to its Starliner flight deck controls, aiming to replicate a modernized Space Shuttle cockpit.

Figure 26: Boeing Display of Starliner Cockpit that a pilot would control during mission profiles [21]

This design has the benefit of being more in line with what pilots interact with – and given NASA requires Space Shuttle pilots and commands meet a minimum of 1,000 hours of experience as a pilot-in-command, this would cater to the vast range of pilot applicants aiming to become astronauts. [21] Another critique of the touch-screen approach comes from Doug Hurley, an astronaut who flew Space Shuttle missions in the early 2000's and who then flew the SpaceX Dragon capsule. When piloting the Space Shuttle, he was 'used to seeing every available surface crammed with buttons, switches, and hand controls to maneuver the spacecraft. The touchscreens on the crew Dragon don't give the same kind of tactile feedback, and that's kinda of a big deal. "Growing up as a pilot my whole career, having a certain way to control the vehicle, this [touchscreen] is certainly different. The difference is you have to be very deliberate when you're putting an input in with a touchscreen relative to what you would do with a stick because you know when you’re flying an airplane for example, if I push the stick forward it’s going to go down. [Whereas on the Dragon] I have to actually make a concerted effort to do that with a touchscreen.”’ [22] Yet as spacecraft becomes more commercialized, astronauts who navigate and guide the vehicle may not necessarily be former pilots who are used to physical control gears or sticks.

These competing designs are only two of the possible concepts constructed so far. Both clearly have advantages, such as the ability for the Dragon flight deck easily visualize the needed mission information and limit the number of ‘buttons, switches, and hand controls’ crew would be responsible for managing. Yet Boeing has the benefit of being more intuitive and tactile, especially to former pilots. Moving forward into commercialized spaceflight, the need for an innovative, intuitive, and interactable flight deck will likely produce a more futuristic design akin to SpaceX. Flight commanders in the future may never see the inside of an aircraft cockpit, and hence would not intuitively understand the complex structure of current day Space Shuttle flight decks or Boeings proposed design. Instead, constructing a touchscreen layout that provides a list of the most crucial mission parameters - current velocity and position, fuel consumption, time of flight towards a destination, proximity to other space debris or vehicles - along with a 3D representation of the spacecraft on its trajectory will be imperative for future spaceflight.

IV. SPACEFLIGHT CONSOLE BACKGROUND COMPUTATION

Before delving into the spaceflight console interface and capabilities, it is important to understand the physics behind the output of results the crew interacts with. This is divided into: 1) fundamental Keplerian elements and initial orbit architecture, 2) general and lambert algorithm for trajectory planning, 3) gravity assist and flybys for specialized interplanetary trajectory design, 4) reference frames for detailing various perspective views and impulse maneuvers as conceptualized by the crew.

Keplerian Elements

To mathematically describe an orbit, one must define at least six orbital parameters, referred to as orbital elements or Keplerian elements. These include the semi-major axis (a), eccentricity (e), inclination (*i*), argument of periapsis (ω), true anomaly (f), and longitude of ascending node (Ω). Additional terms used in the equations include periapsis – the point in an orbit closest to the primary – and apoapsis – the point in an orbit farther from the primary – where primary is defined as the body being orbited.

Semi-Major Axis is defined as one half of the major axis and represents satellites mean distance from its primary:

$$a = \frac{(R_P + R_A)}{2} = \frac{1}{\frac{2}{r} - \frac{v^2}{G * M}}$$

Eccentricity is defined as ‘a dimensionless parameter (either a vector or scalar) that determines the amount by which its orbit around another body deviates from a perfect circle’:

$$e = \frac{R_P * {V_P}^2}{G * M} - 1 = \frac{R_A}{a} - 1 = 1 - \frac{R_P}{a}$$

Inclination is defined as the angular distance between a satellite's orbital plane and the equator of its primary:

$$i = \cos^{-1}(\frac{h_Z}{h})$$

True anomaly is the angular parameter that defines the position of a body moving along a Keplerian orbit:[*]

$$f = \cos^{-1}(\frac{\boldsymbol{e}*\boldsymbol{r}}{e*r})$$

Argument of periapsis is the angular distance between the ascending node and the point of periapsis:

$$\omega = \cos^{-1}(\frac{\boldsymbol{N}*\boldsymbol{e}}{N*e})$$

Longitude of ascending node (Ω) is the node (point where an orbit crosses a plane) of a satellite crossing from south to north's celestial longitude. In other words, is the angle from a specified reference direction, called the origin of longitude, to the direction of the ascending node (crossing north to south), as measured in a specified reference plane:

$$\Omega = \begin{cases} \cos^{-1}(\frac{N_X}{N}), & N_Y \geq 0 \\ 360° - \cos^{-1}\left(\frac{N_X}{N}\right), & N_Y < 0 \end{cases}$$

---

[*] Bold face text represents vector

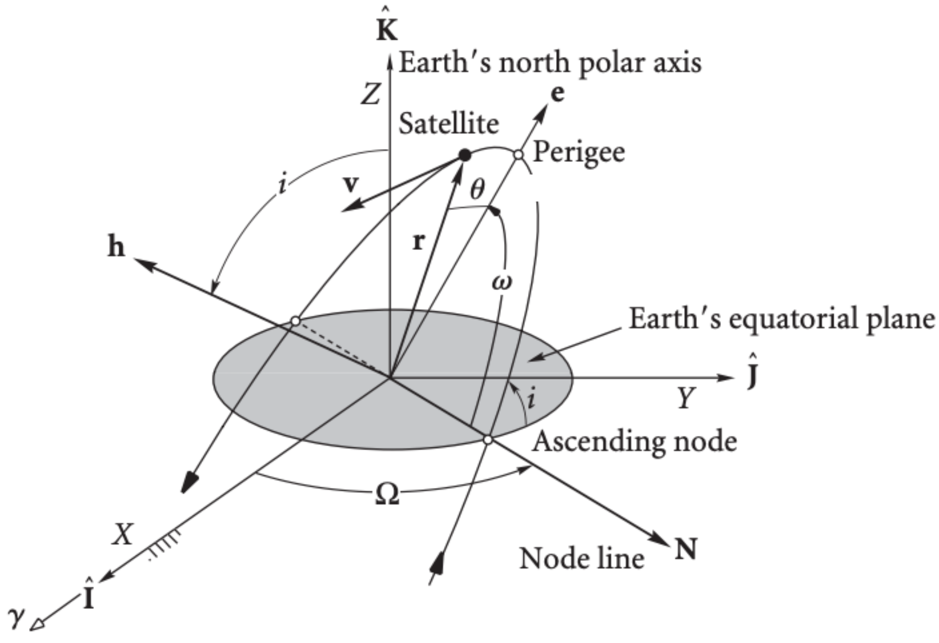


Figure 27: Geocentric equatorial frame and the orbital elements representation.

Lambert Algorithm

Suppose the position vectors $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ of two points $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ on the path of mass m around mass M are all defined for a spacecraft mission. Then according to a theorem of Lambert, the transfer time Δt from $\boldsymbol{P_1}$ to $\boldsymbol{P_2}$ is independent of the orbit's eccentricity and depends only on the sum $r_1 + r_2$ of the magnitudes of the position vectors, the semimajor axis a and the length c of the chord joining $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$. Then using the time of flight (Δt) from $\boldsymbol{P_1}$ to $\boldsymbol{P_2}$, Lambert's problem is to find the trajectory joining $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$.

The trajectory is determined once $\boldsymbol{v_1}$ is calculated, because the position and velocity of any point on the path are determined by $\boldsymbol{r_1}$ and $\boldsymbol{v_1}$. The algorithm used in this thesis is taken from Curtis [26], and is consolidated in a general method below for guidance:

Step 1: Calculate the magnitude of the given position vectors, $\boldsymbol{r_1}$ *and* $\boldsymbol{r_2}$

$$r_1 = \sqrt{{r_{x1}}^2 + {r_{y1}}^2 + {r_{z1}}^2}, \quad r_2 = \sqrt{{r_{x2}}^2 + {r_{y2}}^2 + {r_{z2}}^2},$$

Step 2: Choose prograde or retrograde and calculate the change in true anomaly (Δf):

$$\Delta f = \cos^{-1}(\frac{\boldsymbol{r_1} * \boldsymbol{r_2}}{r_1 * r_2})$$

Step 3: Calculate (iterate with) intermediate variables A, y, and z in functions F(z)

$$A = \sin(\Delta\theta) * \sqrt{\frac{r_1 * r_2}{1 - \cos(\Delta\theta)}}$$

$$^{*}F(z) = \left[\frac{y(z)}{C(z)}\right]^{3/2} * S(z) + A * \sqrt{y(z)} - \sqrt{\mu}\Delta t$$

$$z_{i+1} = z_i - \frac{F(z_i)}{F'(z_i)}$$

$$y = r_1 + r_2 + A * \frac{z * S(z) - 1}{\sqrt{C(z)}}$$

Step 4: Calculate the Lagrange coefficients f, *g, and* $\dot{g}$

$$\mathrm{f} = 1 - \frac{y}{r_1}, \quad g = A * \sqrt{\frac{y}{\mu}}, \quad \dot{g} = 1 - \frac{y}{r_2}$$

*Step 5: Calculate the target velocity needed (**v₁**) and the arrival velocity (**v₂**) – if desired*

$$v_1 = \frac{1}{g}(r_2 - \mathrm{f} * r_1), \quad v_2 = \frac{1}{g}(\dot{g} * r_2 - r_1)$$

*Step 6: Use **r₁** and **v₁**, or **r₂** and **v₂**, to calculate the orbital elements of the trajectory path.*

In result, all that needs to be inputted from the crew is the destination and time of flight. Known entities, like planets or moons, can have their location in the solar system pre-programmed to allow a simple drop-down selection for the destination selection. Adding on to this, if a time of flight is not inputted, a default minimum time, relative to the target distance, can be used to begin

---

[*] Each term, function, and derivation is provided on page 204 of Curtis Orbital Mechanics Textbook

the lambert algorithm. This would then iterate through, calculating the velocity vectors and corresponding orbital elements, until a default maximum time is achieved. With this array pair of time and velocity, a plot of the possible trajectories can be provided to the crew for selection.

Gravity Assist – Planetary Flyby

A gravity assist maneuver is the use of relative motion of an orbit and the gravitational pull of a planet or astronomical object about that orbit to alter the path and speed of a satellite, typically used to save propellant and reduce expense. More specifically, ‘a gravity assist around a planet changes a spacecraft's velocity (relative to the Sun) by entering and leaving the gravitational sphere of influence of a planet. The spacecraft's speed increases as it approaches the planet and decreases while escaping its gravitational pull (which is approximately the same), but because the planet orbits the Sun, the spacecraft is affected by this motion during the maneuver. To increase speed, the spacecraft approaches the planet from the direction of the planet's orbital velocity and departs in the opposite direction. To decrease speed, the spacecraft approaches from the planet from a direction away from the planet's orbital velocity – in both types of maneuvers the energy transfer compared to the planet's total orbital energy is negligible.’ [41]

Calculating a gravity assist maneuver simply factors in an additional step with the general equations of patched conics. In patched conics, the mission objective of launching from one planetary body to another is broken down into three sections: elliptical transfer orbit, hyperbolic planetary departure, and hyperbolic planetary arrival. The gravity assist calculations factor on the last section, wherein the arriving hyperbolic velocity is added with the planets velocity to create a second exiting hyperbolic departure. The patched conics and modified gravitational assist calculations are consolidated and outlined below following the three sections mentioned.

A preliminary calculation is first shown for reference on the derivation for the patched conics. A perfectly ideal trajectory would be a Hohmann transfer, which is defined as an elliptical orbit used to transfer between two circular orbits of different radii around a central body in the same plane. The Hohmann transfer often uses the lowest possible amount of propellant in traveling between these orbits, and can be easily calculated as follows:

Using the modified vis-viva equation for a Hohmann-transfer, the spacecrafts velocity about its initial orbiting body and at the periapsis of the transfer orbit for the deltaV:

$$\Delta \boldsymbol{v_1} = \sqrt{\frac{\mu}{\boldsymbol{r_1}}} * \left(\sqrt{\frac{2\boldsymbol{r_2}}{\boldsymbol{r_1} + \boldsymbol{r_2}}} - 1 \right)$$

Where $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are respectively the radii of the departure and arrival circular orbits (larger of the two being the periapsis of the transfer orbit, and the smaller being the apoapsis of the transfer orbit). The deltaV ($\Delta \boldsymbol{v}_1$) of the first impulse burn represents the ejection from the initial orbiting body into the transfer trajectory. And for the second impulse burn, a similar equation is used but with the gravitational parameter of the target body and at the apoapsis of the transfer trajectory ($\Delta \boldsymbol{v}_2$). Thus:

$$\Sigma \Delta \boldsymbol{v} = \Delta \boldsymbol{v_1} + \Delta \boldsymbol{v_2} = \sqrt{\frac{\mu}{\boldsymbol{r_1}}} * \left(\sqrt{\frac{2\boldsymbol{r_2}}{\boldsymbol{r_1} + \boldsymbol{r_2}}} - 1 \right) + \sqrt{\frac{\mu}{\boldsymbol{r_2}}} * \left(1 - \sqrt{\frac{2\boldsymbol{r_1}}{\boldsymbol{r_1} + \boldsymbol{r_2}}} \right)$$

The reason for using a patched conics approach instead is that a Hohmann transfer is incredibly simplified. For example, a mission to Mars would require the spacecraft to start in a parking orbit around Earth, perform a maneuver, and escape Earth's sphere of influence first. Furthermore, a mission designer would also need to consider the effects of Mars's gravity during the final portion of the problem. Thus, solving the deltaV values is more realistically and accurately done by breaking the problem down into multiple two-body problems.

Section 1: Elliptical Transfer Orbit

The most important factor in this section is calculating the phase angle (ϕ) and closest epoch at which this phase angle occurs. This accounts for the time of travel that occurs for the arrival planet while the spacecraft is in flight.

$$\phi = 180° - \frac{1}{2} * T_{Hoh} * \dot{\theta}$$

$$T_{Hoh} = 2\pi * \sqrt{\frac{a_t^3}{\mu}} \quad and \quad \dot{\theta} = \frac{360°}{2\pi} * \sqrt{\frac{\mu}{a^3}}$$

Section 2: Hyperbolic Planetary Departure

To enter the Hohmann transfer orbit, $\boldsymbol{v}_\infty$ must be the correct amount to place the spacecraft on the desired elliptical trajectory. The required $\boldsymbol{v}_\infty$ is equal to the difference between Earth's orbital velocity ($\boldsymbol{v}_E$ and $\boldsymbol{r_E}$ of the Earth's orbit around the sun) and the velocity the spacecraft needs to have at the periapsis of the transfer orbit:

$$\boldsymbol{v}_\infty = \sqrt{\frac{\mu}{\boldsymbol{r_E}}} * \left(\sqrt{2 - \frac{\boldsymbol{r_E}}{a_t}} - 1 \right)$$

This velocity is then used to calculate the semi-major axis of the hyperbola:

$$a_{hyp} = (\frac{2}{r_{E,SOI}} - \frac{\boldsymbol{v}_\infty{}^2}{\mu_E})^{-1}$$

Now with the hyperbolic semi-major axis, the velocity at periapsis ($\boldsymbol{v}_p$) is:

$$\boldsymbol{v_p} = \sqrt{\mu_E\left(\frac{2}{\boldsymbol{r_p}} - \frac{1}{a_{hyp}}\right)}$$

And finally, the deltaV for the first impulse burn ($\Delta\boldsymbol{v_1}$) can be solved as:

$$\Delta\boldsymbol{v_1} = \boldsymbol{v_p} - \sqrt{\frac{\boldsymbol{\mu_E}}{\boldsymbol{r_p}}}$$

Tying back to the phase angle for where to apply this first impulse burn, the angle between the apse line of the escape hyperbola and the line parallel to the planet's velocity vector is:

$$\beta = \cos^{-1}\left(\left(1 + \frac{\boldsymbol{r_p} * \boldsymbol{v}_{\infty}{}^{2}}{\boldsymbol{\mu_E}}\right)^{-1}\right)$$

Section 3: Hyperbolic Planetary Arrival or Gravity Assist

For arriving at the target planet from the hyperbolic trajectory, an orbital insertion impulse burn depends on the target orbit altitude on the given planet. For simplicity, the following calculation solves for a burn into a circular orbit with a radius equal to the periapsis distance of the hyperbolic arrival trajectory:

$$\Delta\boldsymbol{v_2} = \boldsymbol{v_p} - \sqrt{\frac{\boldsymbol{\mu_T}}{\boldsymbol{r_p}}}$$

If the maneuver is a gravity assist, then the entering heliocentric velocity vector $(\boldsymbol{v_1})$ is actually the sum of the entering hyperbolic excess velocity vector and the planet's velocity vector and the exiting heliocentric velocity vector $(\boldsymbol{v_2})$ is the sum of the exiting hyperbolic excess velocity vector and the planet's velocity vector.:

$$\boldsymbol{v_1} = \boldsymbol{v_p} + \boldsymbol{v}_{\infty,1} \;\; and \;\; \boldsymbol{v_2} = \boldsymbol{v_p} + \boldsymbol{v}_{\infty,2}$$

Then the deltaV from the gravity assist is simply the difference between $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, which based on the provided formulas is simplified to:

$$\Delta \boldsymbol{v} = \boldsymbol{v}_{\infty,2} - \boldsymbol{v}_{\infty,1}$$

Hence, the total $\Delta \boldsymbol{v}$ from a gravity assist is the difference between the exiting hyperbolic excess velocity vector, and the entering hyperbolic excess velocity vector.

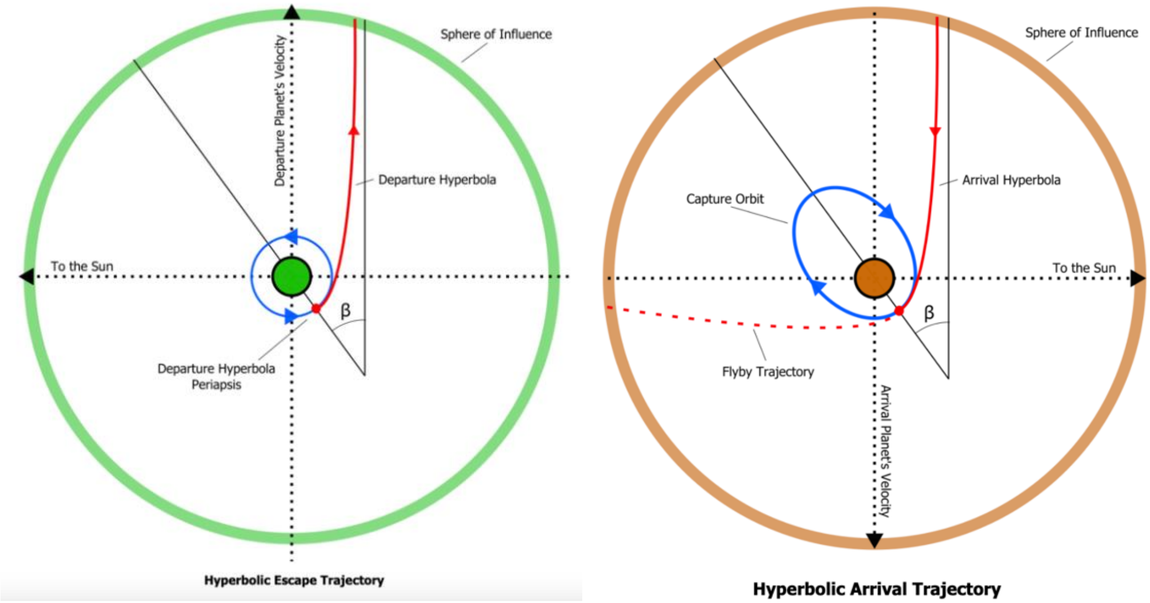


Figure 28 and 29: Hyperbolic Escape and Arrival Trajectory Drawing, respectively

Attitude Reference Frames

Velocity, Normal, Binormal (VNB) is a rotating reference frame where the origin is the center of the spacecraft. This is used for reference when the crew inputs an impulse velocity burn, they are doing so in the VNB frame. This frame is described by an x-axis vector oriented in the direction of vehicles velocity. A y-axis vector oriented in the direction of the orbit normal, or in the direction of angular momentum of the vehicle. And a z-axis oriented in the direction of the vehicles position vector, forming a right-handed coordinate system.

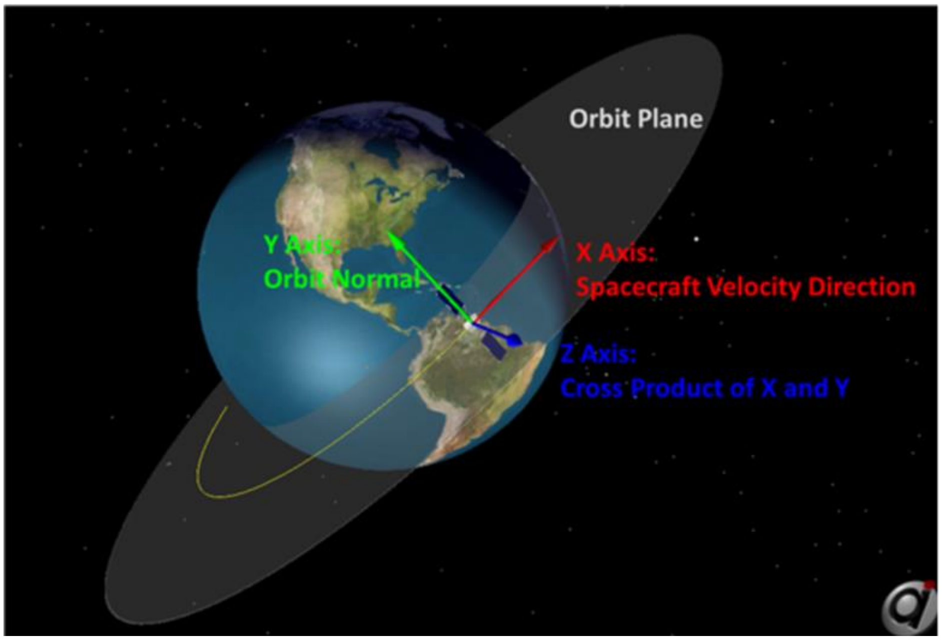


Figure 30: Vehicle-Normal-Body (VNB) Reference Frame from an Earth-centered view

Local Vertical, Local Horizontal (LVLH) is another rotating reference frame with the spacecraft as the origin. This frame is Earth pointing and is used in the Earth centered velocity conversions and deltaV calculations. It is defined as having an x-axis perpendicular to y-axis and z-axis, forming a right-handed coordinate system. A y-axis negative to the orbit normal, or in the negative angular momentum direction. And a z-axis oriented in the direction negative the vehicles position or pointing toward the center of the Earth.

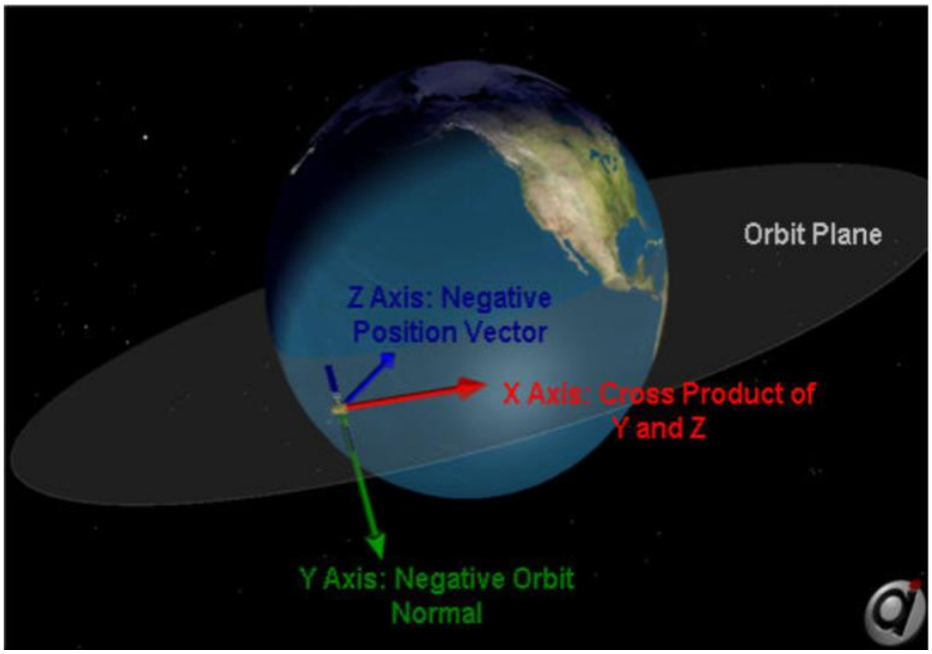


Figure 31: Local Vertical, Local Horizontal (LVLH) Reference Frame from an Earth-centered view

International Celestial Reference Frame (ICRF) is an inertial frame defined by the adopted, fixed locations of about 600 extragalactic radio sources with the origin at the solar system barycenter. This frame is nearly identical with the J2000 frame, a non-inertial reference frame with x-axis vector pointing from the Earth to the mean vernal equinox at Julian year 2000.0. A y-axis vector perpendicular to the x-axes and z-axes, forming a right-handed coordinate system. And a z-axis vector normal to the mean equatorial plane at Julian year 2000.0, pointing towards the Northern Hemisphere. The J2000 and ICRF frames are used for SPICE data, which provides space geometry, event data, and general navigation information for mission designs.

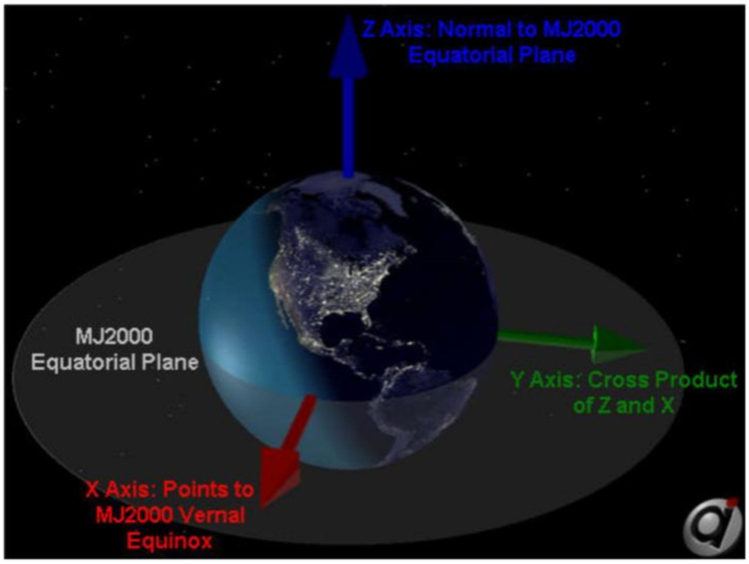


Figure 32: J2000 Reference Frame

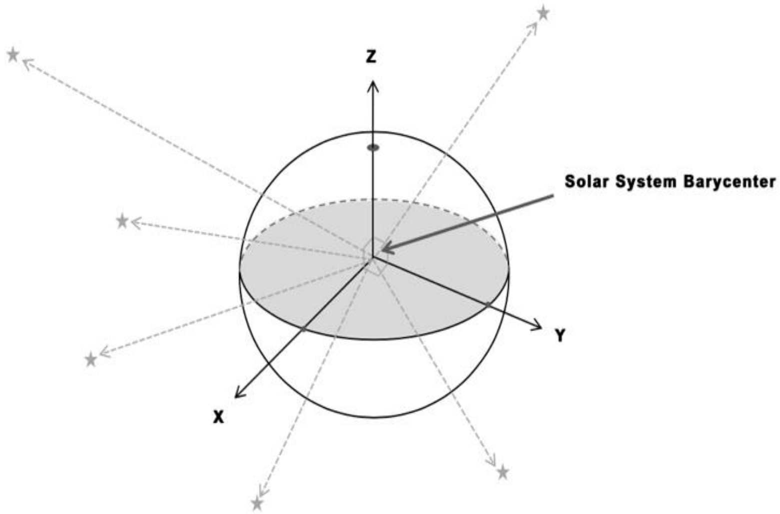


Figure 33: International Celestial Reference Frame (ICRF)

V. SPACEFLIGHT CONSOLE DESIGN CONCEPT

The concept of the Spaceflight Console encompasses controls and inputs, as well as the rendering and outputs for each given objective. The architecture of the flight deck will be a combination of touchscreens, akin to the SpaceX Dragon capsule, manual input controls, like the Space Shuttle flight deck, and ability to render certain mission profiles with a holographic display. This approach will allow all the information to be isolated to a select few screens, instead of being represented by different monitors or displays placed throughout the control panel like typical cockpit designs. Displaying the data on limited number of interactive monitors is more in line with the state of current technology. Smart phones, touchscreen computers, tablets, and the like provide a user with the controls and information on one single screen. Future interfaces will continue this trend, coupled with the advancement of voice control and gesture recognition. [23]

Yet integrating physical controls, like a joystick or push buttons, would be appropriate certain mission profiles such as close proximity and docking, especially from the perspective of pilots. As note by SpaceX Crew members onboard, the sensitivity of a touchscreen and possible discontinuous motions of each ‘button’ push can make a certain mission like docking more difficult. The loss of physicality with touchscreen causes some applications more difficult than if a control stick or hand controller were available, wherein a pilot has quicker and finer control over their motion. And lastly, for planning and better visualizing trajectories or general flight paths, a holographic representation could succinctly convey the situational awareness the crew might need. Virtual reality, used for showing a 3D model of the spacecraft, of a course destination, or overall station location, can provide a better understanding of the situation than parameters on a screen might.

In summary, the Spaceflight console design will encompass: a mixture of touchscreen, holographic renderings, and physical input controls, as well as the ability to plan and execute a variety of mission profiles. The reasoning for this design stems from the lessons learned of aircraft cockpits and the realization that future spaceflight will continuously grow as demand for space travel increases.

Beginning with the interactive display, past cockpit designs were extremely cluttered and relegated to manual controls with automatic modes. This restricted pilots from being able to easily pull up needed mission information directly in front of them, or to display various data points in a front centered manner that a touchscreen display can provide. For most aircraft cockpits, many of the buttons and switches are duplicated for both pilots to allow the aircraft to be flown from either seat. Hence, buttons and switches in the middle control aircraft systems like lights, climate, fire, and other equipment that are generally not used for primary flight control. [37] Multiple display screens can contain all the buttons and switches, along with a plethora of information that can give the crew a full understanding of their spacecraft. By using a few screens, there would be a drastic reduction in space and mass by moving control to digital displays. The multiple displays also allow for easy redundancy if one were to go awry. Another advantage is the streamlined development, training, testing, and iteration process, all of which are much more cumbersome when there are many physical components. [38]

Nevertheless, as has already been seen with SpaceX touchscreen flight deck, the Spaceflight console will also contain manual controls, akin to a joystick or sliders and throttle, in order to prevent accidental interactions with the screen and to provide more precise and smooth control of the spacecraft. Especially for close proximity maneuvers, touchscreen inputs cannot provide the tactile response pilots are accustomed to. Even the Dragon capsule is not solely touchscreen, as it has about 30 buttons alongside its three screens. This is still a far cry from the over 2000 manual buttons in the Space Shuttle, but it demonstrates even one of the most advanced designs of a flight deck is not reliant on screens alone. [40]

The other interaction, holographic display, also ties in with needs of flexible mission planning and providing situational awareness for every type of operation crew might be exposed to. NASA is already developing ‘Project Sidekick’ that will use HoloLens technology to its newly developed ‘Project Sidekick’ as a way for the crew of the ISS to expedite day-to-day operations and research while increasing overall efficiency. The headsets will enable astronauts to conference with experts here on Earth in real time and provide a holographic instruction manual for designated scientific experiments. [39] Taking this further, holographic renderings can provide the crew a unique visual understanding of where their spacecraft is relative to other satellites or where it is along a given trajectory path. Crew will not need to rely on looking out the deck windows to try to gauge their proximity or understand their relative location. It is imperative that the crew has situational awareness of their spacecraft throughout various mission tasks, and holographic renderings can provide a realistic display of complex situations.

Especially for distant and interplanetary planning, the ability for crew to understand their current situation, and then modify it based on provided information, will be a necessity for deeper space travel wherein communication might also be further limited. Take modern day aircraft pilots for example. They have a given flight path to follow, but if weather becomes a safety concern and they need to adjust flight path, they rely on air traffic controllers for a new heading. They might not be able to see the weather conditions, or understand visually their relative location, but air traffic controllers can. For spaceflight, this understanding needs to be in the hands of the crew. As space travel becomes more congested and at further distances, crew should have the ability to manage their trajectories and general mission planning. This Spaceflight console can provide both the information needed and ability to act on that information for a variety of objectives.

The capabilities of the overall spaceflight consoles are distinguished between short term and long-term applications that future crew would use as spaceflight becomes more developed and expansive.

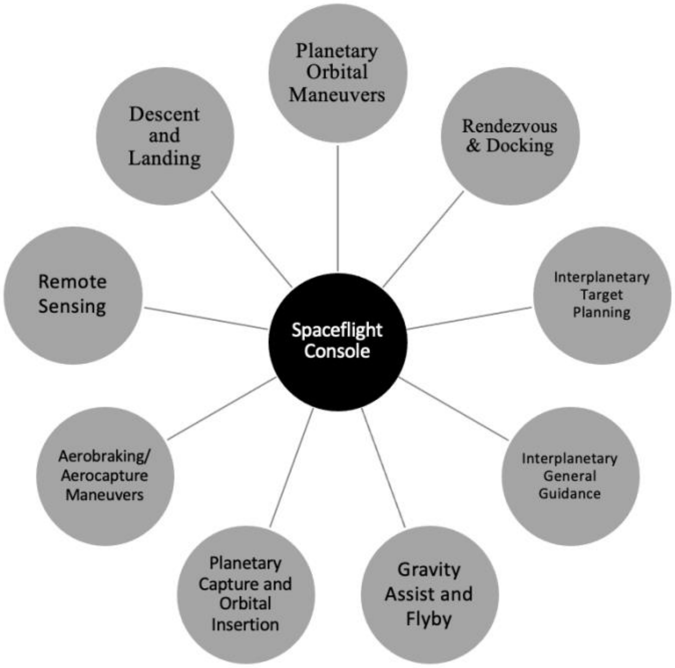


Figure 34: Diagram showing the top-level functions of proposed Spaceflight Console

In order to handle an array of the different objectives, each input control will be restricted to the appropriate menu selection. These various objectives can be grouped as follows:

I. Various Perspective Views about the Solar System

II. Initial Low Earth Orbit Design

III. Impulse Maneuver Planning and Sequencing

IV. Remote Sensing and Data Mapping

V. Proximity and Rendezvous Planning

VI. Trajectory Optimization (including interplanetary)

VII. Current and Past Data Storage Information

VIII. Entry, Descent, and Landing (EDL)

Beginning with the various perspective views, it is critical to the crew that they can conceptually understand the space environment they are traveling through.

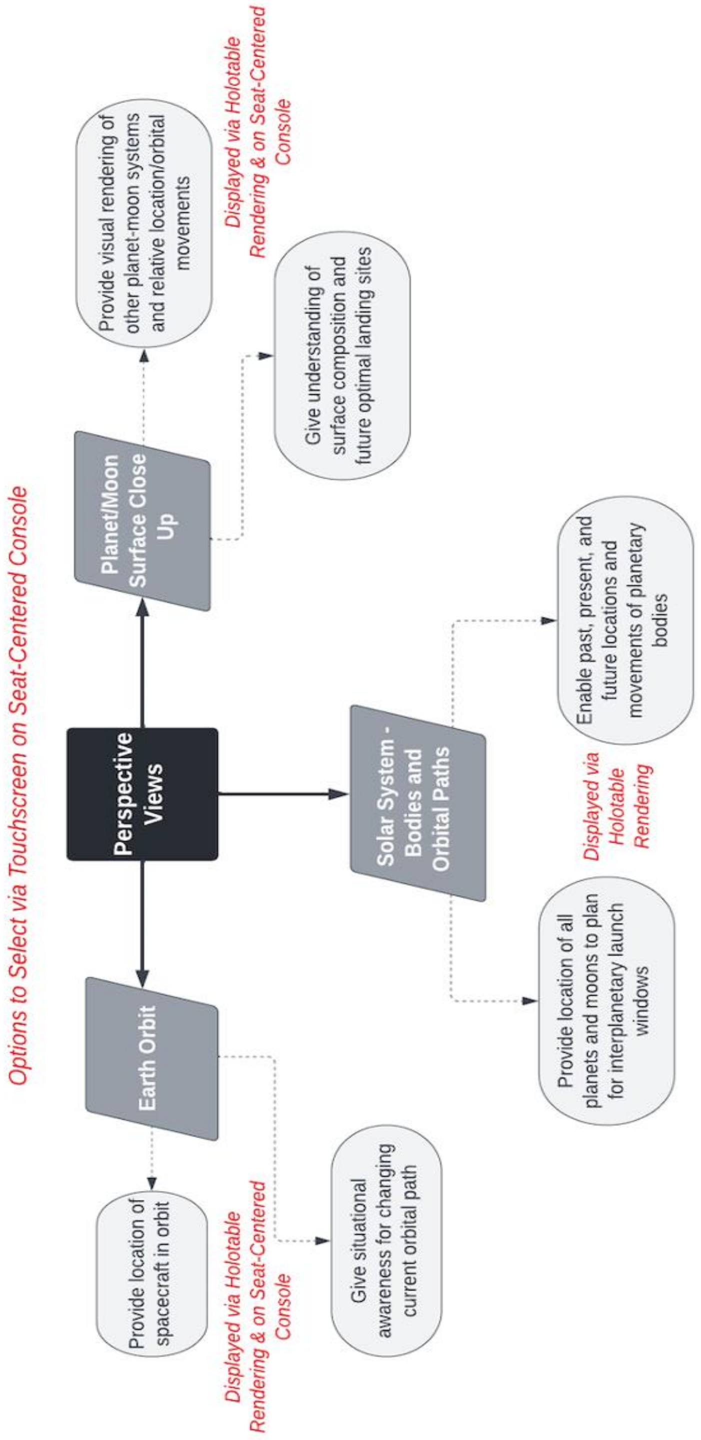


Figure 35: Block Diagram of Perspective Views Functionality of the proposed Spaceflight Console

Crew should be able to select a holographic rendering of the solar system in its entirety, displaying their spacecraft location in real time as planets and their respective moons orbit about. The other options to select from a drop-down display console include a close up of any planetary body, including their initial Earth launch orbit and maneuvers completed to obtain the spacecrafts current position. These perspectives provide the crew unique situational awareness that is not feasible by simply looking out the window or pulling up a static map of orbiting bodies. This is akin to how flight controllers monitor aircraft position and velocity relative to a given state space. This information is now rendered holographically and provided to the onboard crew to aid in future planning of their given mission profile.

The next key ability of the spaceflight console is designing an initial low earth parking orbit to build maneuvers or remote sensing off. The main driver behind designing and visualizing orbits about Earth is to give the crew attitude perspective of their spacecraft while in orbit. This tool provides crew the ability to manipulate Keplerian elements, defined primarily as semi-major axis, eccentricity, inclination, right ascension of ascending node, argument of periapsis, and true anomaly. Specifically with true anomaly, the crew can change the location of their spacecraft on the given orbital path, thereby changing the vehicles position and velocity. Each of these Keplerian elements can be displayed as an onscreen slider or value input. As the crew alters each element, the corresponding change in Earth orbit would be displayed on-screen or via a holographic model. In turn, crew can simulate various orbits such as sun synchronous or geostationary, or outline tracking paths their spacecraft would cross over given the custom orbit.

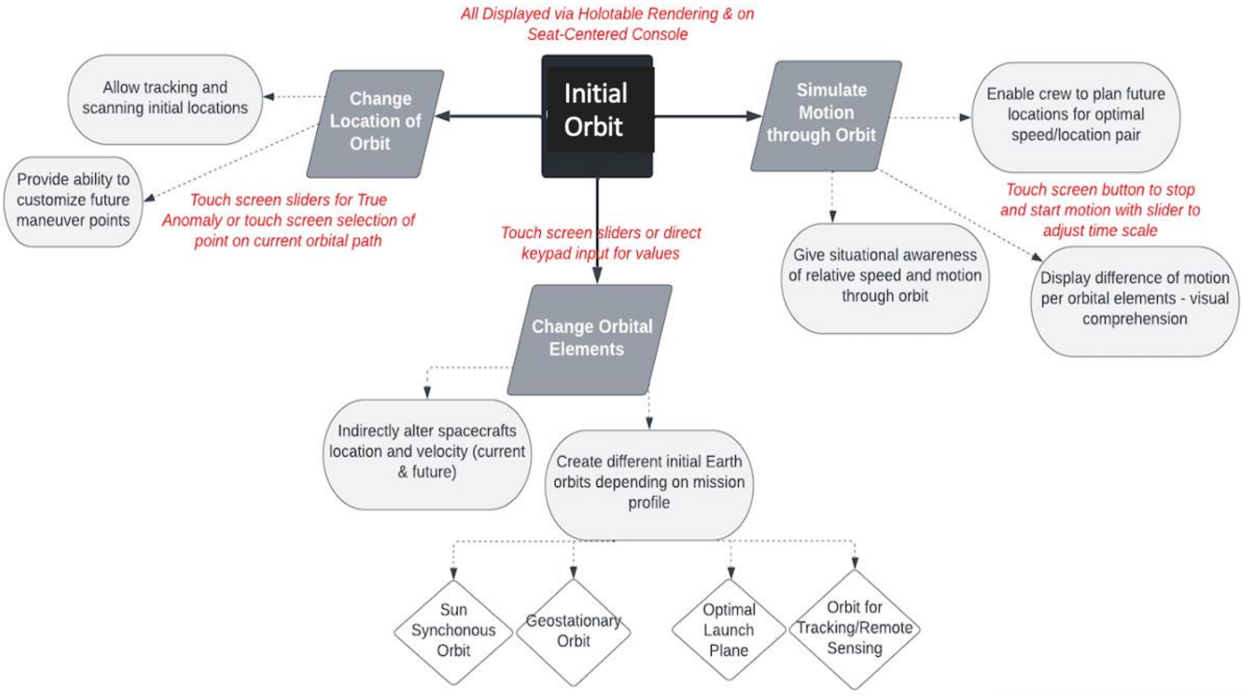


Figure 36: Block Diagram of Earth Orbit Design Functionality of the proposed Spaceflight Console

Building on the ability to change the spacecraft location, crew can also play through a holographic simulation showing their vehicle move through orbit at a custom time speed. This provides the crew the comprehension and ability to predict future paths as they see how their moves through the given trajectory. Being able to visualize and obtain data for movement about an orbit ties back to providing the crew situational awareness of their given mission profile.

With an initial orbit designed, crew can then plan out maneuvers based on their mission objective. As shown below, crew can select a point on their current trajectory path and input an impulse burn to construct a new corresponding trajectory.

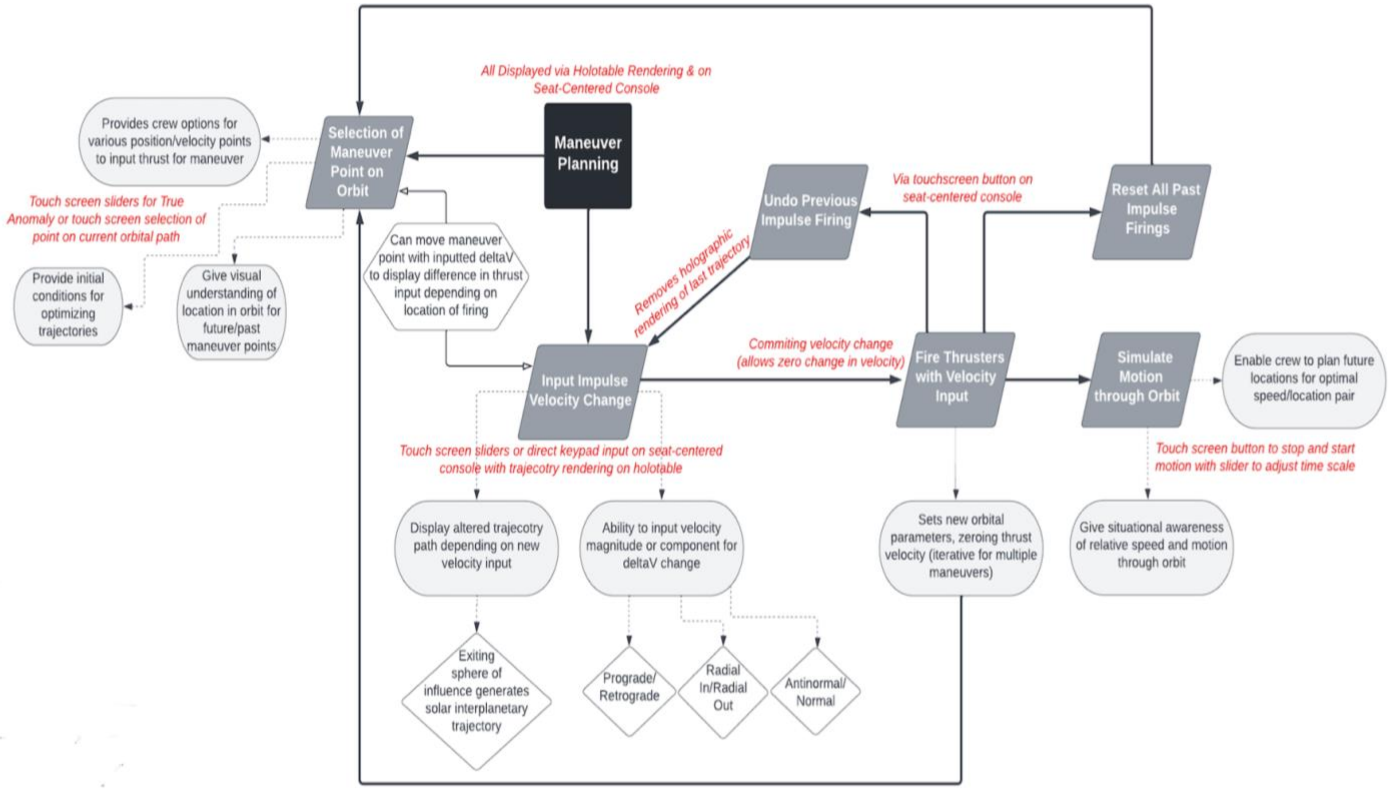


Figure 37: Block Diagram of Maneuver Planning Functionality of the proposed Spaceflight Console

Crew can select a point to conduct the impulse burn, then input the magnitude and direction of that burn, and finalize the maneuver. Once complete, information about the maneuver, such as current and predicted position and velocity, fuel burned, orbital elements, period or time of flight for the trajectory, is all displayed on screen while the trajectory is displayed in a holographic frame. This process can be iterative, allowing multiple maneuvers while also being able to undo or reset the entire design. Finally, as with the initial Earth design, the simulated trajectory can be ‘played through’, showing the spacecraft move through the given maneuvers in real time. With this tool, crew can visualize the impact of applying a deltaV to their current path and obtain new parameter values about the projected trajectory. Including both Earth based orbits and interplanetary orbits, this functionality aids in planning out multiple maneuvers, essentially mapping a sequence from point A to point B anywhere in the solar system, while also providing any necessary data points about the given profile.

Building off planning out impulse maneuvers, crew can construct proximity and rendezvous maneuvers. This is broken down into the selection of a specific body or vehicle to rendezvous with, or the selection to view a holographic map rendering planets and vehicles to potentially rendezvous with.

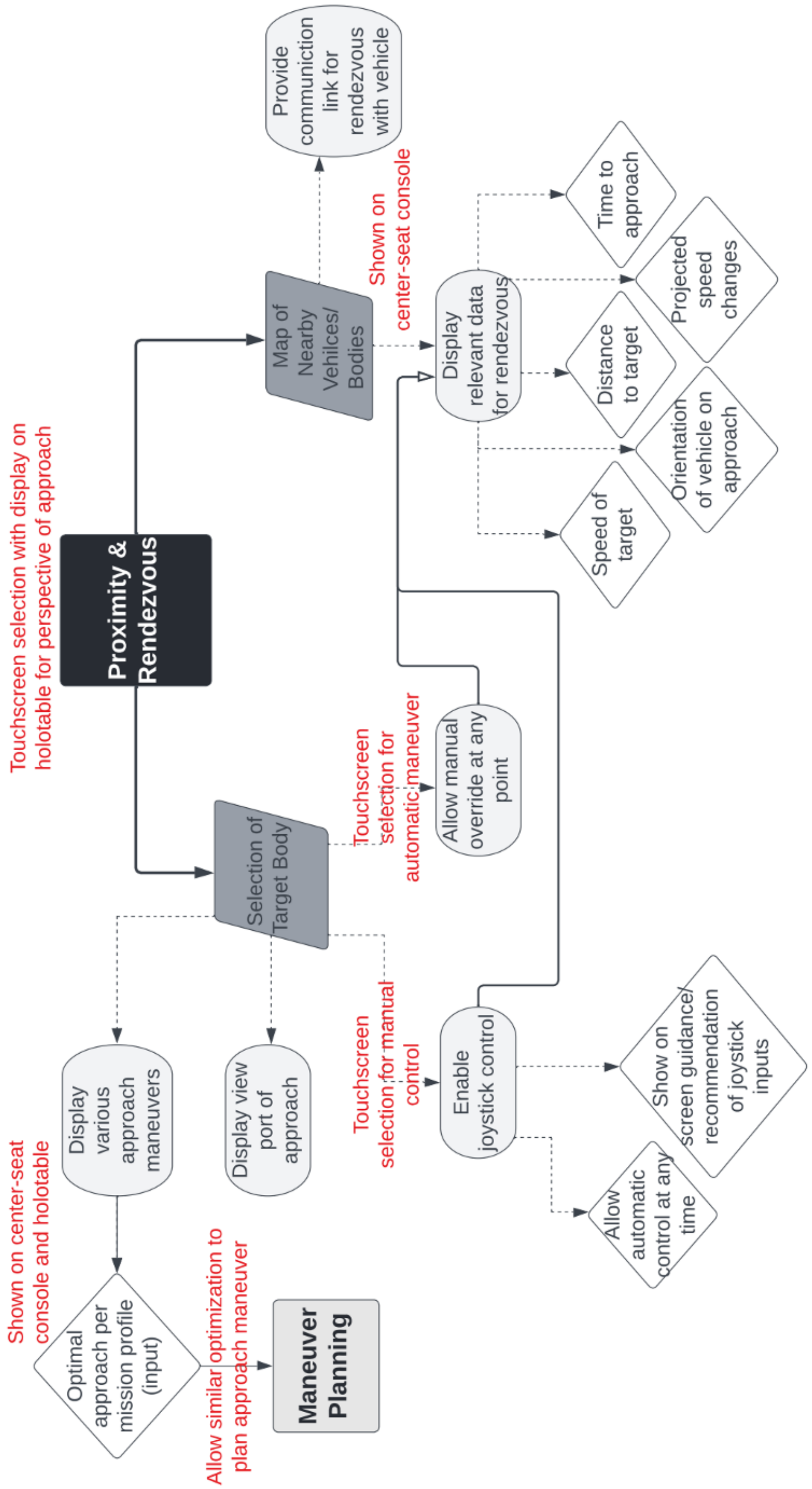


Figure 38: Block Diagram of Proximity and Rendezvous Functionality of the proposed Spaceflight Console

From a drop-down on-screen menu (or a manual input of solar coordinates), crew can view the possible approach maneuvers using predefined approach velocity and orientation to rendezvous properly. Crew can then select the optimal path for their mission objective and use the output deltaV vector and magnitude to conduct the maneuver manually or automatically. This option allows crew to enable joystick control with on screen guidance to conduct the proximity maneuver. Along with on screen guidance, the spaceflight console can also display a view port of the approaching target as if crew were looking out the front window, as shown in the figure below, curtesy of SpaceX designs. With this display crew can further select specific data parameters, including the speed of the target and speed of spacecraft, distance and time to reach the target, approaching planned attitude, and any projected changes in speed or path.

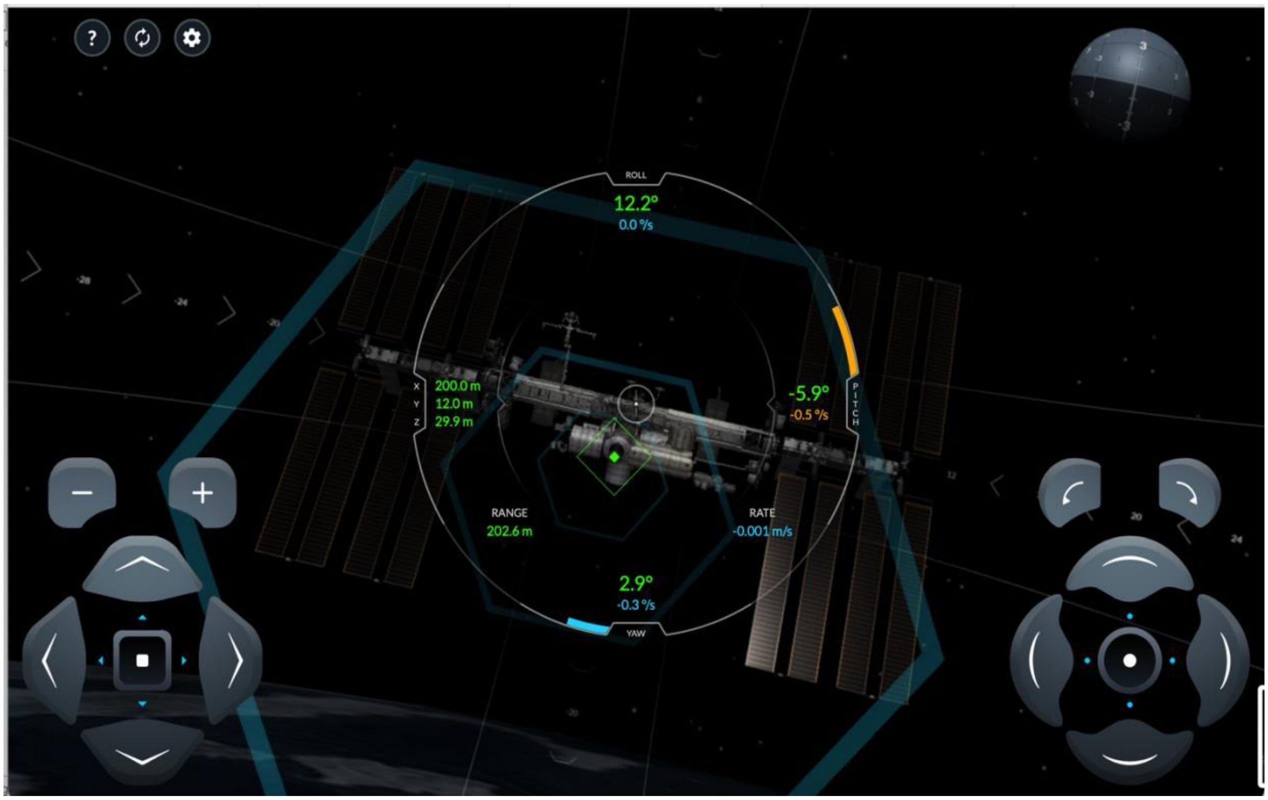


Figure 39: SpaceX Online Docking Simulation [24]

This set up allow for either a manual input control with a joystick, or on-screen selection of thrust firings to control yaw, pitch, and roll upon approach. The guidance assistance with manual control would be shown on screen (such as “Increase roll by five degrees”) as well as audibly to aid in the approach. Lastly, with the spaceflight console showing a front view port, the holotable can render a 3D mapping of the planned approach, including the spacecrafts orientation and the relative location and velocity of the target docking. Similar to how crew can view an interplanetary trajectory spline path, they can also view the optimal path to follow as they maneuver towards the target. This aids with the visual understanding of the spacecraft navigation and guidance, providing to the crew situational awareness of their approach.

The other option to assist the crew in comprehending their environment is to map out all nearby vehicles and bodies to possible rendezvous or dock with. The distance range is specific on screen by the crew, and the resulting array of targets is rendered on the holotable. From here crew can select a target, and view on screen the parameters required to rendezvous, such as time of travel for a given speed, maneuvers to apply, target bodies orbital path and current/future relative location and velocity. Supplying crew with all this information and ability to select specific inputs, situational understanding of their spacecraft relative to solar system entities can be easily conveyed to on board crew, without relying on mission control or ground-based systems to detail the mission environment.

Following maneuver planning, crew can add in remote sensing maneuvers and map out a bodies surface for resources and ideal landing conditions.

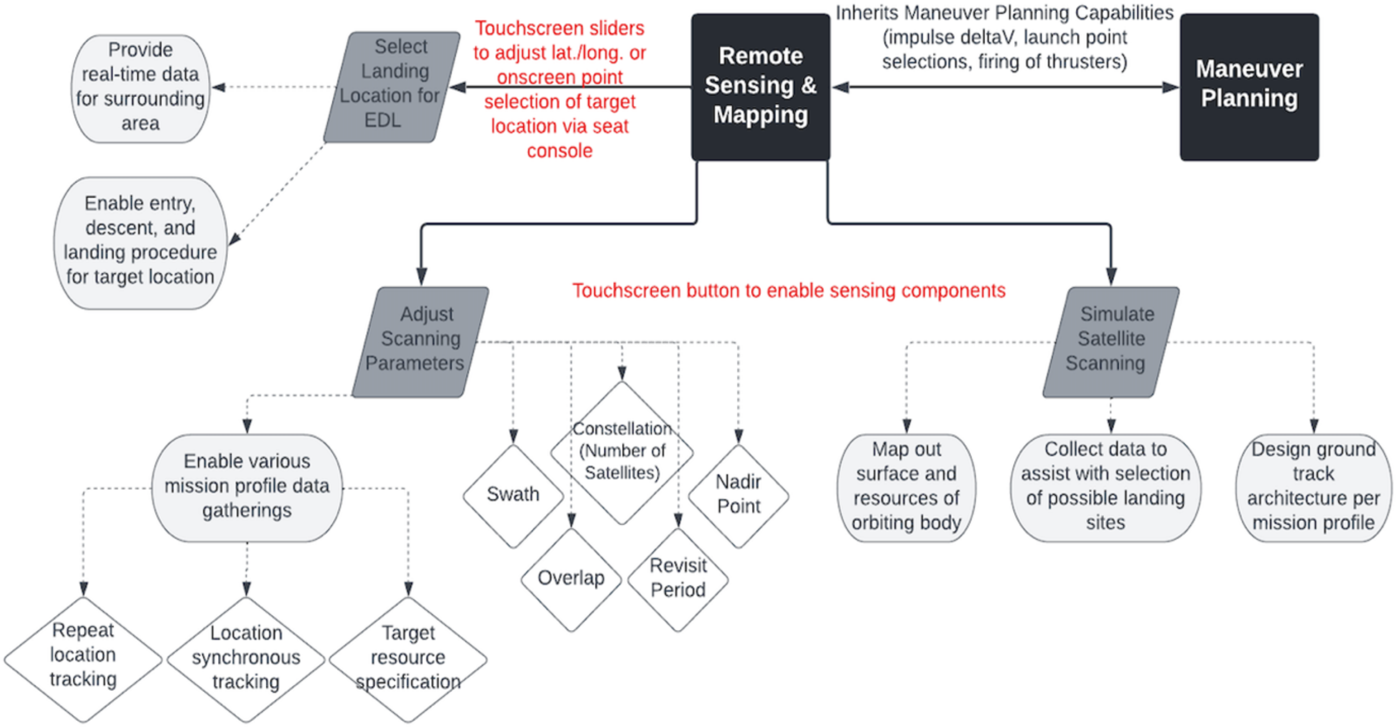


Figure 40: Block Diagram of Remote Sensing and Surface Mapping Functionality of the proposed Spaceflight Console

By adjusting on screen sliders or manually inputting landing coordinates, crew can select an initial target site to begin entry, descent, and landing operations. The crew can also enable scanning simulations, wherein they can construct constellation parameters, including swatch, revisit, overlap, etc.. From this, a simulated scanning profile can gather and display information about the surface of the orbiting body. This information is displayed on screen, and the surface and parameters of the body at hand is shown on a holographic table that the crew can then interact with to further understand the planet body. This provides real-time data that can aid in determining ideal landing conditions, properties of the planet or moon, and enable crew to better understand the environment they are in. Information like this can help crew determine where to land, or whether to stay in orbit, and allow them to construct further maneuvers from their remote sensing operation based on their given objective.

Outside of maneuver planning, the option to optimize a trajectory, without having to physically execute that trajectory, is critical for efficiently planning or reevaluating a given mission objective. As shown below, this optimization is broken down into four different aspects: 1) a direct lambert target algorithm for a specific, unique trajectory, 2) a generalized Hohmann transfer for optimal deltaV and fuel consumption usage, 3) gravity assist and flyby maneuvers to take advantage of planetary positions and gravitational energy, 4) optimal orbit insertion for remote sensing objectives and possible entry to descent maneuvers.

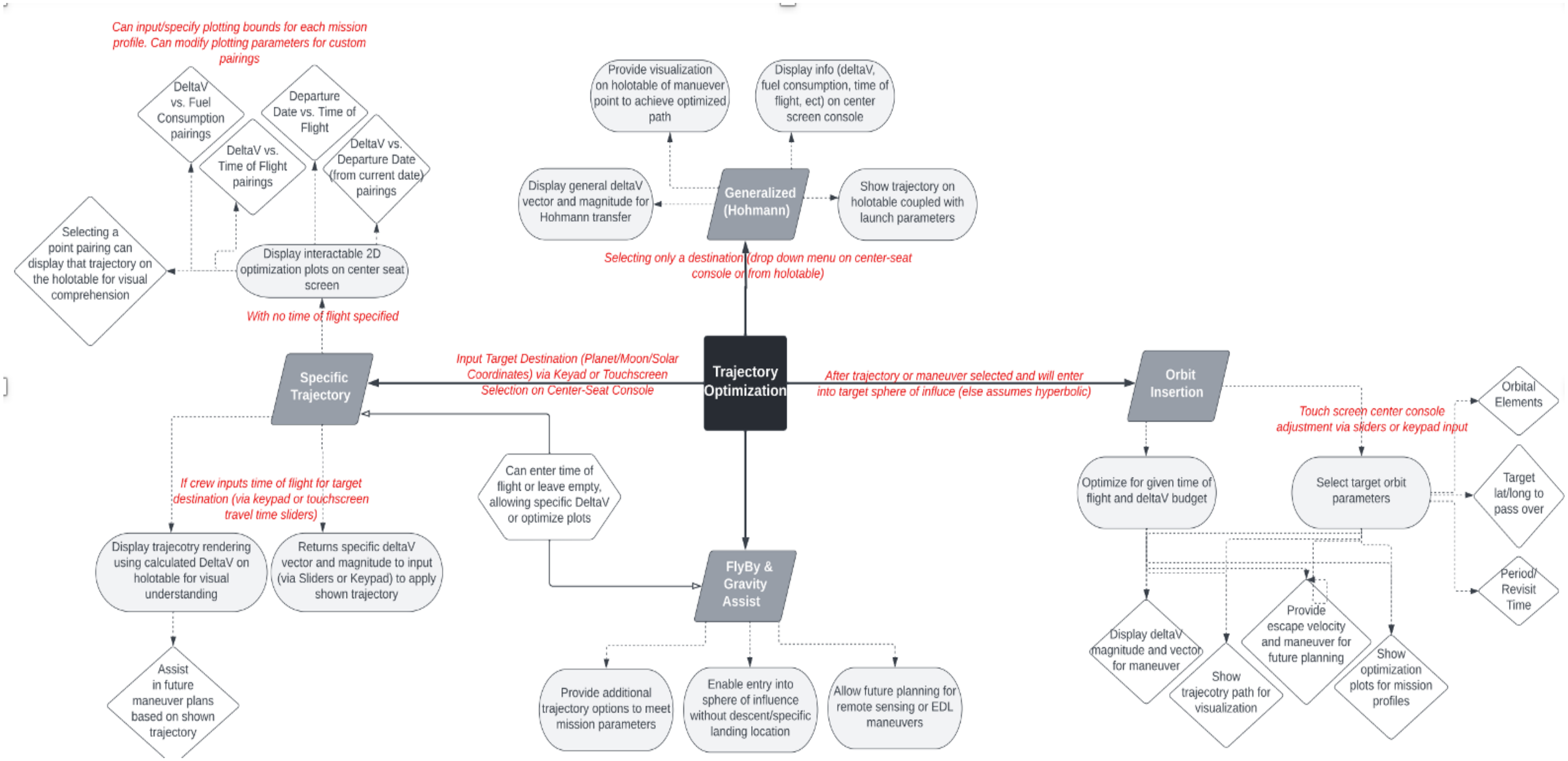


Figure 41: *Block Diagram of Trajectory Optimization Options of the Proposed Spaceflight Console*

For a specific trajectory design, crew can input a time of flight to get from one location (either their spacecrafts current location or another position within the solar system) to a final or intermediate destination. Inputting this information on their display screen, the crew can then render a holographic trajectory showing the given maneuver path as well as obtain data values on screen about the provided display. This includes the specific deltaV vector and magnitude for the resulting impulse burn, the expected fuel consumption based on their spacecraft, and additional orbital parameters such as the phase angle to launch at, ideal departure date, launch window to accomplish the shown path, etc.

Without a specified time of flight, crew can view an array of options based on various mission parameters including the departure date, deltaV, fuel consumption, and any possible additional in-flight maneuvers for plane change or inclination adjustments. This information is rendered on screen as either a heat map or a 3D surface plot, wherein crew can select a data point and display that custom parameter pairing on the holotable to view how the trajectory executes in a solar view frame.

Generalizing this approach, crew can also calculate an optimal Hohmann transfer between any two solar locations. By selecting from an onscreen drop down of pre-computed or known leaving destinations to arrival destinations, a Hohmann transfer provides the crew with a general understanding of the ideal conditions for interplanetary travel. This maneuver is based on specific conditions for the two selected locations, which is conveyed via the holotable by showing the ideal position of the target body at initial launch time and then at arrival time. Since these are ideal conditions, the holographic rendering does not have a time factor, but instead displays a static rendering of the orientation of the given design. Moreover, Hohmann transfer assumes perfectly circular orbits with the bodies in the same plane. These assumptions are detailed on screen to the crew by showing the orbital parameters, along with information detailing when such an alignment might occur. This option for a general Hohmann transfer is to provide the crew with background information and basic understanding of what a low deltaV trajectory design would encompass.

The third custom option for interplanetary travel planning encompasses gravity assist and fly-by maneuvers. By selecting on screen from a drop down of leaving location to final arrival destination, the crew does not need to specify or know what planetary body will optimize their trajectory for a gravity assist. Instead, this maneuver is computed based on the two input locations and is then displayed to the crew on screen by showing the impact of a flyby for the ideal planet corresponding to the inputs. For instance, should the crew want to plan a trajectory from an initial Earth orbit to a circularized parking orbit about Mercury, they can select these two leaving and arriving destinations, and the spaceflight console will return possible maneuver paths for the crew to select from. In this example, a gravitational slingshot about Venus would be one such possibility, and the crew would be able to view relevant information about such a mission plan, including: 1) launch window for when to launch from Earth in order to align with Venus and Mercury, 2) expected change in velocity and direction per the flyby maneuver, 3) time of flight and fuel consumption required for the maneuver, and 4) a comparison against no gravitational assist, should Venus not be utilized.

Akin to a direct transfer using lambert algorithms, crew can enter in a time of flight to accomplish their final arrival and view a specific trajectory path on the holotable. Or they can view a range of flight times to evaluate a trade study for different parameters such as fuel consumption, arrival velocity and orientation, phase window and departure date differences, etc. This gravity assist planning option aims to provide the crew with additional information and options to fulfil a mission objective. Crew should not be restricted to direct trajectories but should have the option to evaluate Hohmann transfer designs and fly-by maneuvers to accommodate all types of constraints when planning interplanetary travel.

The final portion regarding optimizing trajectories is planning an arrival orbit insertion. Crew can select the target body to orbit, and then specify on screen the orbital elements of the desired parking orbit. They also have the option to specify a latitude/longitude value to target for a pass over or specify a period or revisit time for the desired orbit. Using the crew inputs, the resulting deltaV impulse maneuver is shown on screen, coupled with the holographic rendering of the maneuvers and arriving orbit. This allows crew to then sequence an arrival orbit with either remote sensing, additional impulse maneuvers for a new target, or the initial conditions to begin entry, descent and landing. With orbit insertion, crew can finalize an entire mission procession beginning with an initial orbit to a calculated trajectory path and ending with an arrival parking orbit. This end-to-end simulation is critical for planning out entire mission designs and allows crew to understand the impact and limitations on current maneuvers, seeing the domino effect to later maneuvers and arrival conditions.

The last main application with the spaceflight console is planning entry, descent, and landing operations. Given a target body, crew can input specific coordinates or known surface landmarks to begin a controlled descent. They can specify the descent conditions, like the time till touch down, damage predicted to incur on the spacecraft, speed on approach, or ability to abort the descent. This information provides the crew an ability to conduct a trade space like study on board their spacecraft. And based on the customized inputs, the crew can then render a simulation of the fly down via the holotable.

In order to aid in selecting a landing site, the crew can also render a resource map of the surface of the target planet. This information can either come from previous remote sensing data gatherings, or from historically known mapping of the body. This gives the crew one more additional source of information to plan out their ideal landing site, based on what resources or capabilities they need upon a controlled landing.

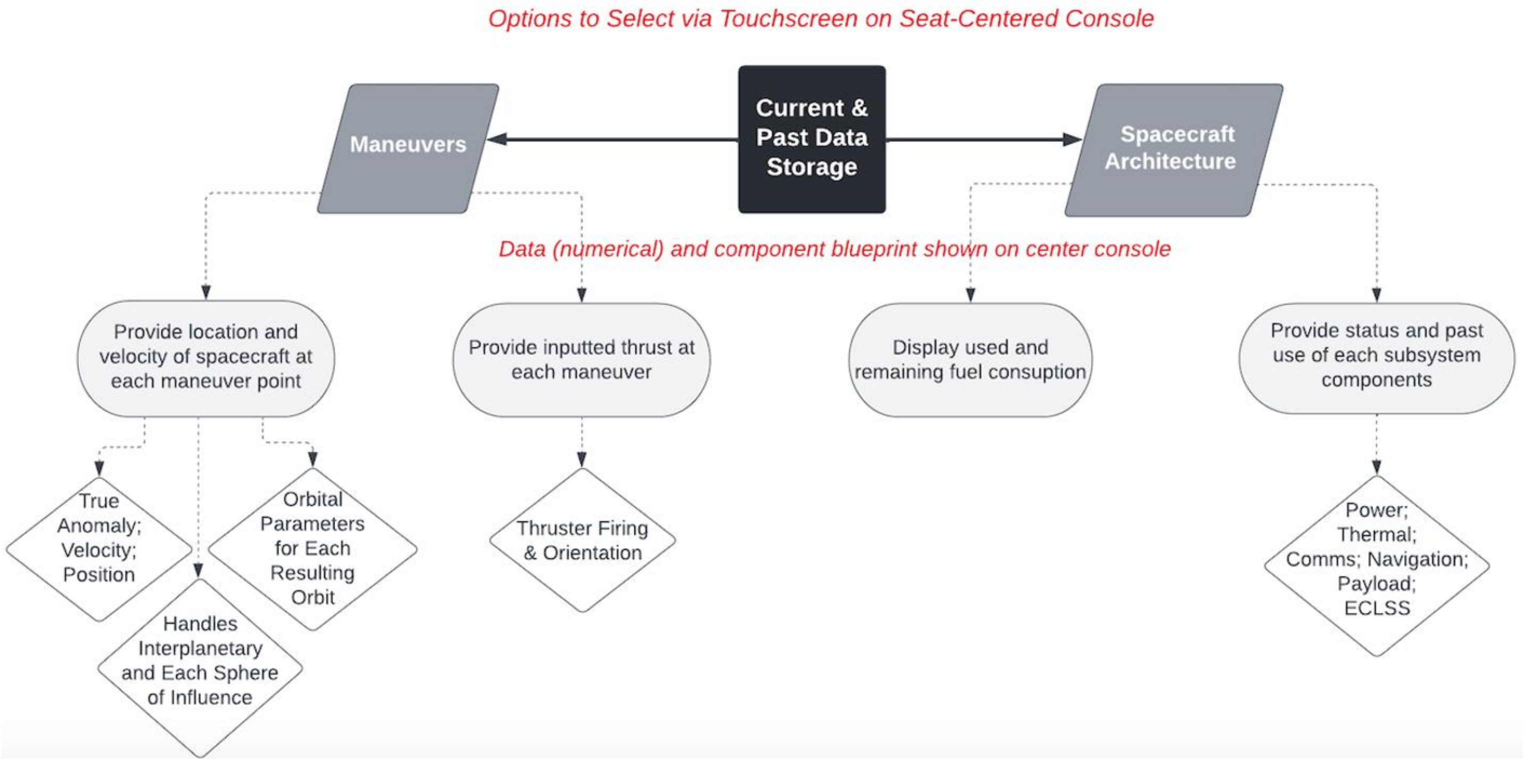


Figure 42: Block Diagram of Current and Past Data Storage Functionality of the proposed Spaceflight Console

As shown in the figure above, along with all of the various options provided by the spaceflight console, crew can at any time review back on past maneuver data or information about their spacecraft architecture. Selecting past maneuver data can show the crew the location and velocity of previous impulse burns, along with the ability to play through a simulation of the maneuver sequencing on the holotable. Moreover, crew can view specifications about their spacecraft, including the used and remaining fuel, power, and communication links. This can be further broken into subcomponents of the spacecraft should crew want to obtain or review information about a select subsystem like ECLSS or payload/cargo, etc..

VI. IMPLEMENTATION

To put this concept of a novel guidance instrument into reality, a virtual simulation will be constructed to replicate the tool as it would appear on board a spacecraft. For demonstration, the previous mission objectives – Earth orbit design and interplanetary/general trajectory planning – will be simulated and evaluated on. Accomplishing this requires building the backend equations and calculations that would represent the spacecraft navigation system and sensor components in C++. The other portion to construct is the front-end visual aid, or what the crew would physically interact with, using unreal engine (UE4) for this initial implementation. Both these requirements are broken down into tasks outlined later in this section. This is with the intent to construct a Spaceflight Console akin to the image below:

Figure 43: Concept to Implement for a Spaceflight Console akin to Thales Cockpit Model Rendering

This rendering shows the mixture of various touchscreens, with the ability to switch between screen displays depending on the mission profile and needed information. It also encompasses hand and manual controls, such as the sliders mentioned for altering Earth orbits or the joystick at hand for proximity or docking options. What is missing from the rendered model above is a holographic representation that the crew can interact with. Such an interactive display can be shown between flight deck chairs, as a small pop-up rendering. A larger holographic rendering can also be provided via a holo-table, wherein the crew can more physically and at large interact with the mission design. This type of a holotable might look like the following:

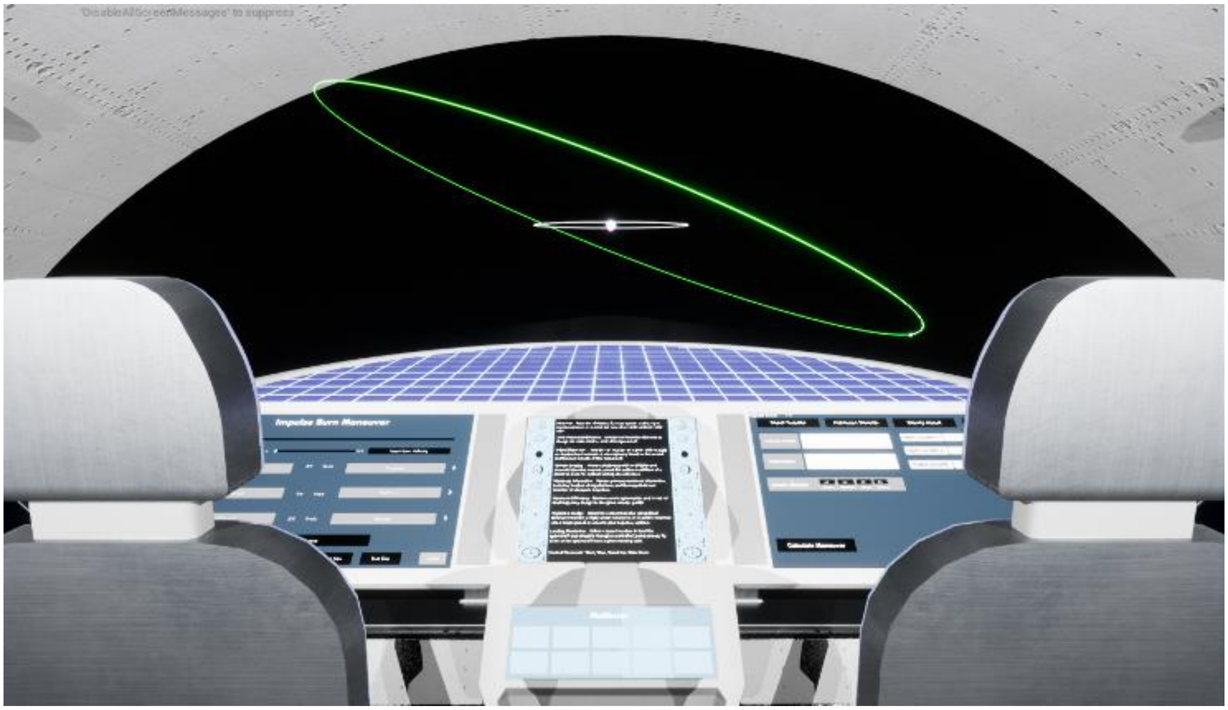


Figure 44: SpaceCRAFT VR Rendering of a Holotable for Spaceflight Console from Pilots Seat

This can show the solar system and orbiting bodies in a more encompassed view, allowing crew to better understand their situational awareness for their spacecraft. In result, the Spaceflight Console can be an interacting mixture of touchscreen displays, hand controls, and holographic renderings of specific mission profiles.

To implement this work with SpaceCRAFT, a breakdown of the backend coding and the front-end visualization is outlined:

1. *Formation of Required Equations and Extra Formulas*
    - a. *Keplerian Elements & Attitude Conversions*
    - b. *Interplanetary and Within Sphere of Influence Algorithms*
    - c. *Extra Parameters – fuel consumption, phase angle, etc..*
2. *Development of C++ Programming Language to Calculate Given Equations*
    - a. *Implement Constructed Equations for Maneuvers and Initial Orbit*
    - b. *Connection to Required Libraries and NASA SPICE Data*
    - c. *Integration into UE4 VR Level for Input/Output Communication*
3. *Construction of UE4 Virtual Environment and User Interface for Execution*
    - a. *Construction of Spacecraft Environment*
    - b. *Construction of Flight Deck Display*
        - i. *Build Menu Switcher for Each Mission Objective*
            - *a.* Create Menu Buttons and Input Controls
            - b. *Create Map and View Ports*
            - c. *Format Data Output*
        - j. *Generate Solar System Entities*
            - a. *Set Communication Link to Coded Functions*

The formation of required equations and formulas stems from the various mission objectives. As has been stated, this includes determining interplanetary and rendezvous trajectories, optimization for custom target locations, and modifying LEO with Keplerian elements. Future implementations also seek to include the ability for maneuvers in proximity operations and docking, entry, descent and landing displays, and general output of information pertinent to each mission profile.

All this is built from the fundamentals of celestial mechanics, with most of the fundamental calculations referenced in the earlier section IV Spaceflight Console Background Computation. These calculations provide the groundwork for multiple types of trajectory planning, including rendezvousing with another spacecraft, implementing a gravity assist orbit, and generating an optimization of orbits to choose based on time of flight, deltaV, and fuel consumption. Additional controls for proximity and docking would require only the sub-function that translates the input velocity of thrust controls to the vehicles body frame and then calculates the new position based on that impulse burn.

Next, once the baseline functions have been constructed, it will then be implemented into C++ code, the primary language that runs SpaceCRAFT Platform. This platform is a virtual reality sandbox that allows the computation and integration between scripted code and an unreal engine to simulate any type of environment or activity. The backend code utilizes functions that can access additional external math libraries to make some calculations easier, as well as the needed data systems of SPICE kernels, the resource used to accurately represent the solar system planets and moons in real-time. All this code is then connected to UE4 by constructing a JSON script that uploads the required data monitors, as well as referencing the C++ folders inside of UE4 upon constructing the virtual environment. Once all the connections have been implemented, the formulations from the previous step can be implemented. This will enable the construction of the default orbit based on classical elements, as well as the ability to take in velocity changes at a given data point to construct a maneuver or future trajectory map.

Third, the graphical representation of these algorithms is constructed in a ‘level’, which includes a map and actors that is then linked to the C++ classes. UE4 uses its own classification of backend code, called Blueprints, which performs similar functions as C++ would, but in a graphical representation – connecting nodes and blocks with input and output pins for variables. Blueprints will function as the input and output node between the displayed controls and the underlying C++ code. UE4 also generates the meshes that makeup the surface of the planets, the visual design of the orbit trajectories, and the control panel main menu options and data readouts provided to the crew. This encompasses the entirety of the ‘visual’ representation that makes this tool a user-friendly Graphical User Interface (GUI).

To reduce the complexity the user sees, this UE4 portion will be the only interaction the crew has with the tool at large. Hence the C++ code and data systems operate only in the background, hidden from the display. The construction portion of UE4 will include creating a dynamic solar system, wherein the planets and moons move through their respective orbits around the Sun. A starting portal view of the Earth and a spacecraft in LEO will also be generated, constructed from the classical elements. From here the crew member will have the option, via their control panel, to change view ports to see the spacecraft in LEO or see the planets orbiting in a solar system view. As mentioned before, the orbits and trajectory options are all displayed in a 3D rendering to allow more intuitive understanding of the spacecraft reference attitude. And finally, with the user given initial orbit, and the ability to select a launch position using true anomaly parameters, the development of orbital maneuvers can be completed. This includes leaving Earth SOI, viewing interplanetary orbits in a solar system view frame, inputting location target trajectories using the lambert algorithm, and viewing mission profile parameters such as time of flight, fuel consumption phase window, and potential warnings to various flight paths.

A visual breakdown of the process is shown in the following figure.

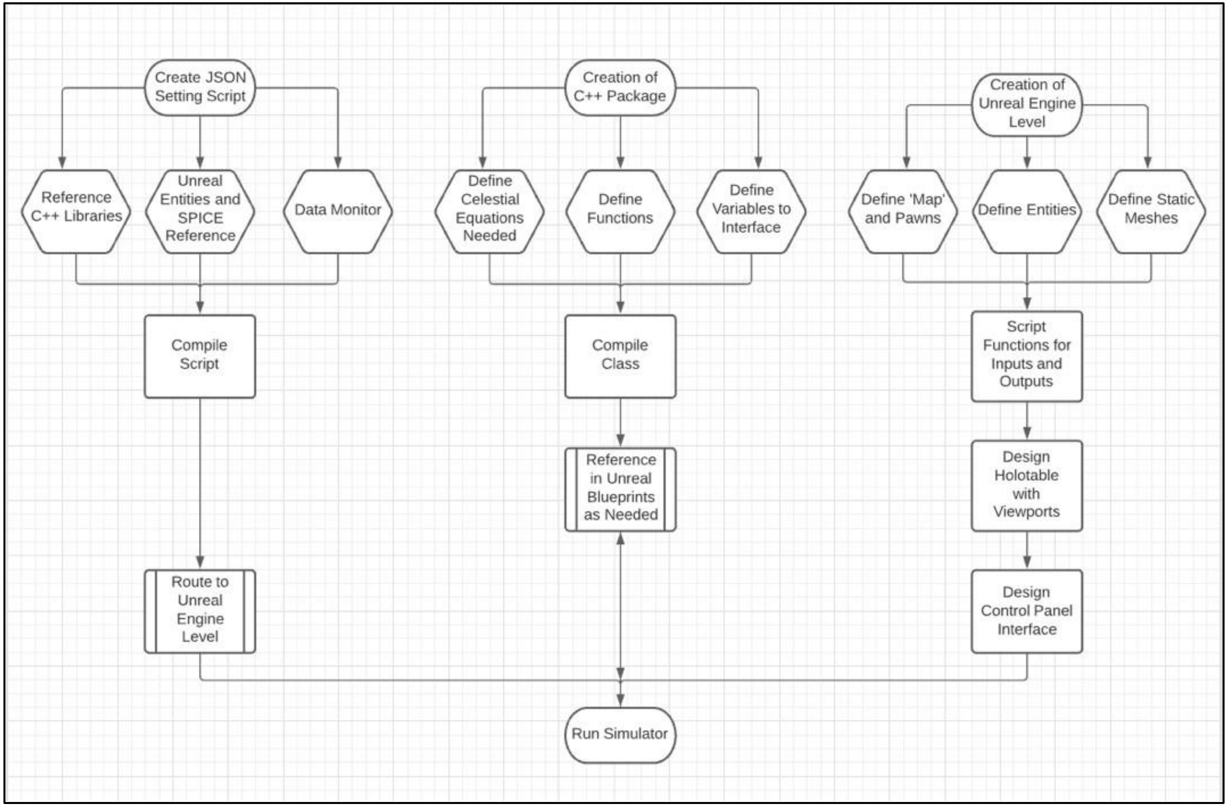


Figure 45: Workflow Diagram for Implementation of SpaceCRAFT Platform

Again, this illustrates the need for creating the equations and back-end code, which also includes scripting a settings JSON file to upload the SPICE data to propagate planetary bodies in a solar frame. This further outlines the general flow of creating the virtual reality side in Unreal Engine, requiring the definition of a map, level, entities, and their static meshes. All of this is the conscripted with a control panel information display and a holo-table rendering that creates the entire simulation.

VII. RESULTS

To demonstrate the envisioned capabilities of the Spaceflight console, the following mission profiles targeting Jupiter and its moon Io, as well as a landing about Earth's moon will be outlined as:

1. Construction of an initial low Earth orbit (LEO) to initialize a maneuver from.
2. Plan out an interplanetary transfer from Earth to Jupiter, using the spacecraft's current location and velocity, and for a given amount of transfer time.
3. Apply the planned maneuver for an impulse burn to arrive on hyperbolic trajectory to Jupiter.
4. Plan out a parking orbit about Jupiter (to base a future maneuver targeting Io).
5. From an initial low Earth orbit, target and enter into a parking orbit about the moon (changing view ports for possible remote sensing about the moon).
6. Plan an impulse burn to initialize a descent to a specified landing site.
7. Implement an autopilot fly down for landing at the given coordinate.

Beginning with an initial orbit, the crew has the options to change the basic Keplerian elements via on screen sliders or by using a keypad input for specific numerical values. The pilot screen renders as such, with the following trajectory display showing on the holotable in front of the pilot seat, coupled with the Moons orbit (in green) moving in a sped-up timescale:

Semi-major Axis (km): 10000.0
Eccentricity: 0.0
R.A.A.N.: 0.0
Argument of Periapsis: 0.0
Inclination:

Figure 46: Spaceflight Console Keplerian Elements Pilot Display

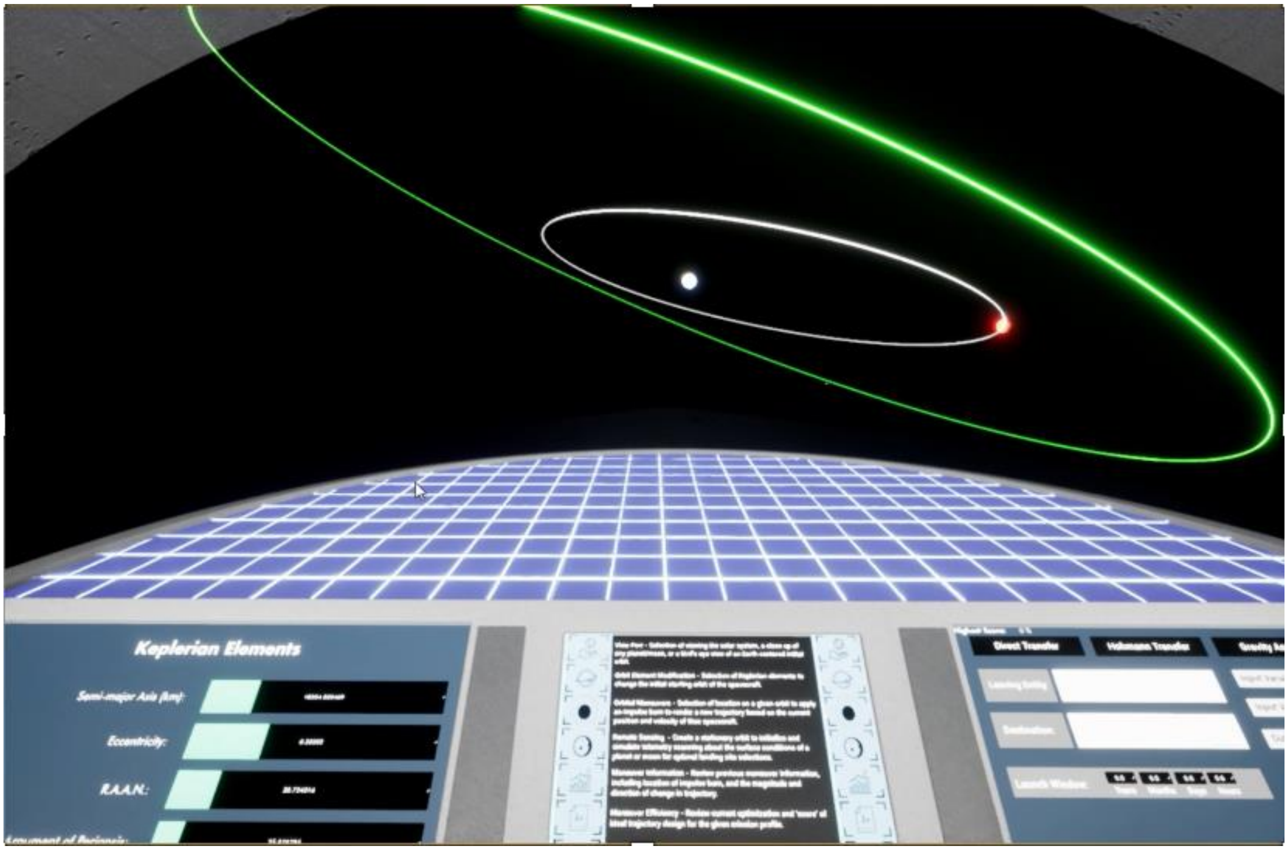


Figure 47: Holotable Rendering of Altered LEO with Moon’s Trajectory

This function tentatively replaces the reliance on mission control to design a parking orbit to launch from, especially if new information or changes in the mission plan require a different initial orbit. Crew thus has the ability to plan and visually comprehend the first stage of their given mission pathway. On the spaceflight console, outputs of the Keplerian elements as well as the current spacecraft position and velocity relative to Earth and in a solar frame is provided to the crew. While this information is not required to input manually into the next step for a maneuver, it aids in providing the crew understanding about the current and projected attitude of their spacecraft.

For planning out the transfer from LEO to entering Jupiter's sphere of influence, the pilot selects from the center console screen the sub-menu for planning out interplanetary travel. This menu provides the ability to plan out interplanetary and within SOI pathways depending on a wide set of mission parameters – direct lambert transfer, optimized direct transfer options, general Hohmann and ideal transfers, and fly-by or gravitational assist for a final destination.

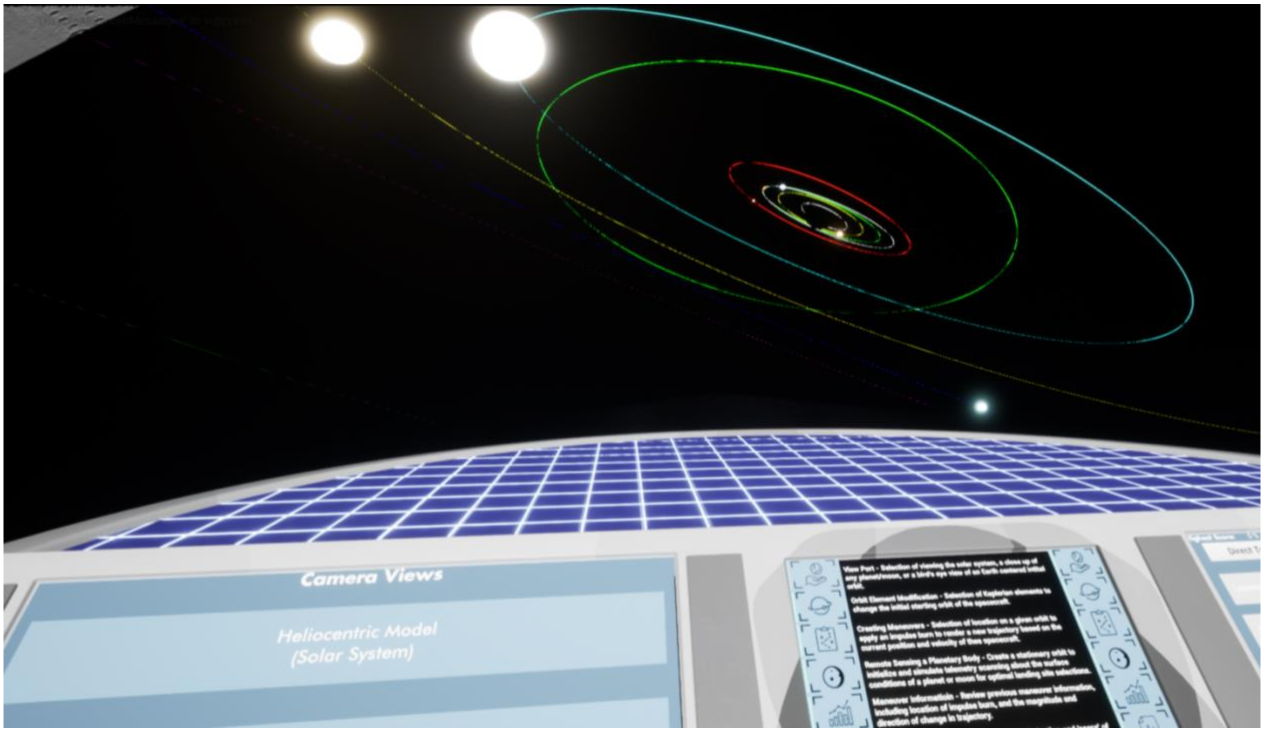


Figure 48: Spaceflight Solar System View Frame of Planets and Orbits for Planning Interplanetary Trajectory

LIGHTING NEEDS TO BE REBUILT (4 unbuilt objects)
BLUEPRINT COMPILE ERROR

Highest Score: 0 %

Launch Point: Spacecraft

Target Body: Jupiter

Maneuver Information
Impulse DeltaV Magnitude: 48.022
Prograde: -12.015627 Normal: -42.447998 Radial In: 18.970974
Escape Velocity Magnitude:
Ejection Angle from Orbit:

Time of Flight: 1 Years 2 Mnths 0.0 Days 0.0 Hours 0.0 Mins.

Figure 49: Spaceflight Console from Pilot Perspective for Lambert Target from Spacecraft to Jupiter

Using the designed parking orbit, crew can set a maneuver node by selecting a point location on screen via a true anomaly slider, a keypad input, or on the holotable's LEO orbit spline. Once a node is selected, crew can then plan out the interplanetary path to go from their current LEO to an arrival at Jupiter. They have the option to map a direct one impulse burn transfer, entering in a time of flight to reach Jupiter's sphere of influence, and viewing an output of the exact impulse burn to apply and the rendered interplanetary spline that shows the projected path the impulse burn corresponds to.

Additionally, crew could select a generalized Hohmann transfer to compare ideal conditions and minimal fuel consumption for an optimized amount of travel time. They also have the ability on screen to view a trade study of different parameters to select the optimal deltaV that pertains to the mission constraints. These parameters can be defined by the crew, but by default the crew can view on screen a 3D contour plot showing the pairing of deltaV magnitude, time of flight, and expected fuel consumption for an array of possible trajectories.

An example of a 3D plot, or a heat map, are currently in future work to be implemented with UE4 in the simulated environment. A normalized surface level map rendered in MATLAB below is akin to what the pilots would use, and a heat map generated using a Python library displays values of deltaV, departure date, and time of flight that crew can use to optimize their choice trajectory.

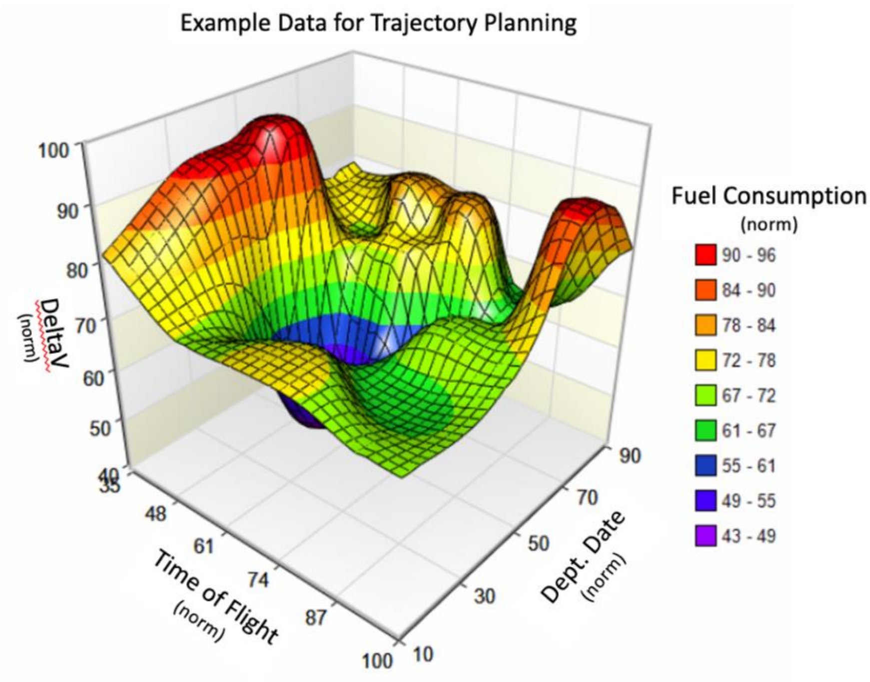


Figure 50: MATLAB Surface Map of Normalized Trajectory Data

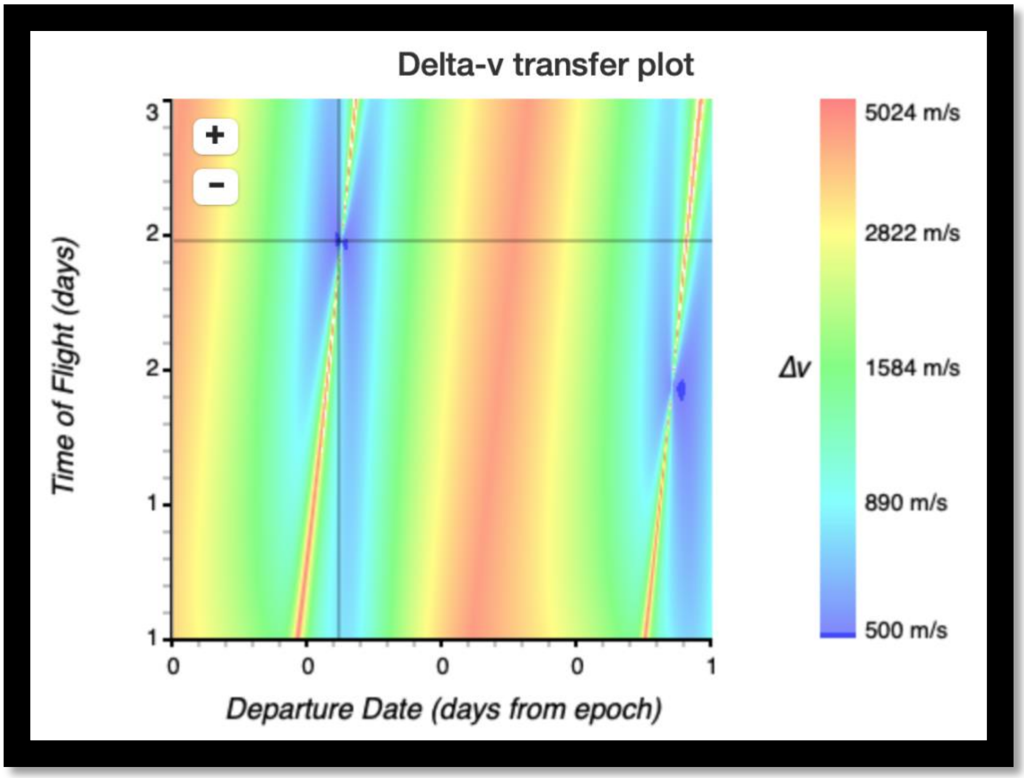
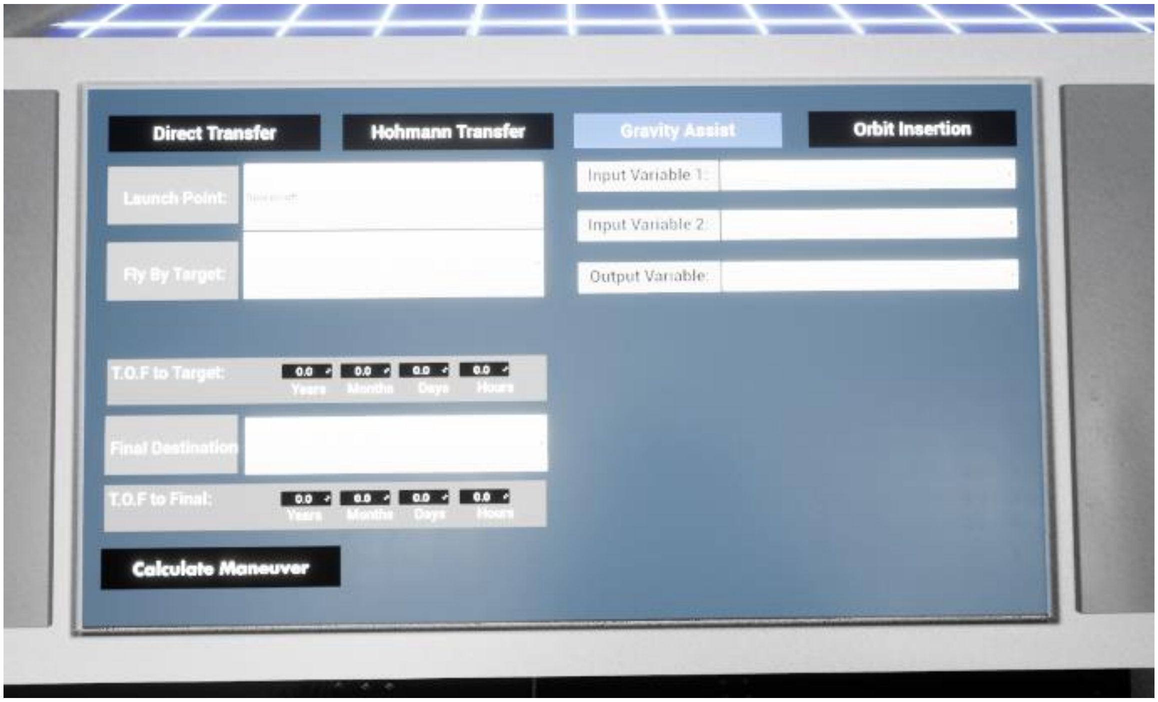


Figure 51: Delta-V Transfer Plot from Earth to Moon [25]

Other parameters to plot include future departure dates (which automatically corresponds location and velocity of the spacecraft and target planet), phase angle to target with, arrival hyperbolic velocity, and ability to apply an intermediate impulse maneuver, etc.. This allows the crew to comprehend the trade-offs for selecting certain elements about the trajectory. It also helps compare different trajectories, numerically with the pertinent data, and visually, with the option to render any data pairing on the holotable to show the projected interplanetary path. Coupled with this, crew can also select a fly-by trajectory or gravity assist sequence for approaching Jupiter and continuing on to Io without entering into an intermediate parking orbit about Jupiter. With future iterations, crew can have the option to select their final target body, and based on their input, they can then select from a list of possible planets to flyby that would provide the most assist in their trajectory path. As shown below, crew can also enter each time of travel for both legs of the journey, or leave them empty to provide a range of possible values for a variety of flight times.



Launch Point:

Fly By Target:

T.O.F to Target: 0.0 Years 0.0 Months 0.0 Days 0.0 Hours

Final Destination

T.O.F to Final: 0.0 Years 0.0 Months 0.0 Days 0.0 Hours

Input Variable 1:

Input Variable 2:

Output Variable:

Figure 52: Spaceflight Console of Gravity Assist Maneuver Display from Pilot Perspective

The next step in mission sequence is to enter into a parking orbit about Jupiter from a hyperbolic arrival. Once crew has applied the output deltaV to target Jupiter for the given time of flight, they can select from their console menu to plan an orbital insertion maneuver. Here crew can customize their parking orbit, defining the apoapsis and periapsis altitudes to target.

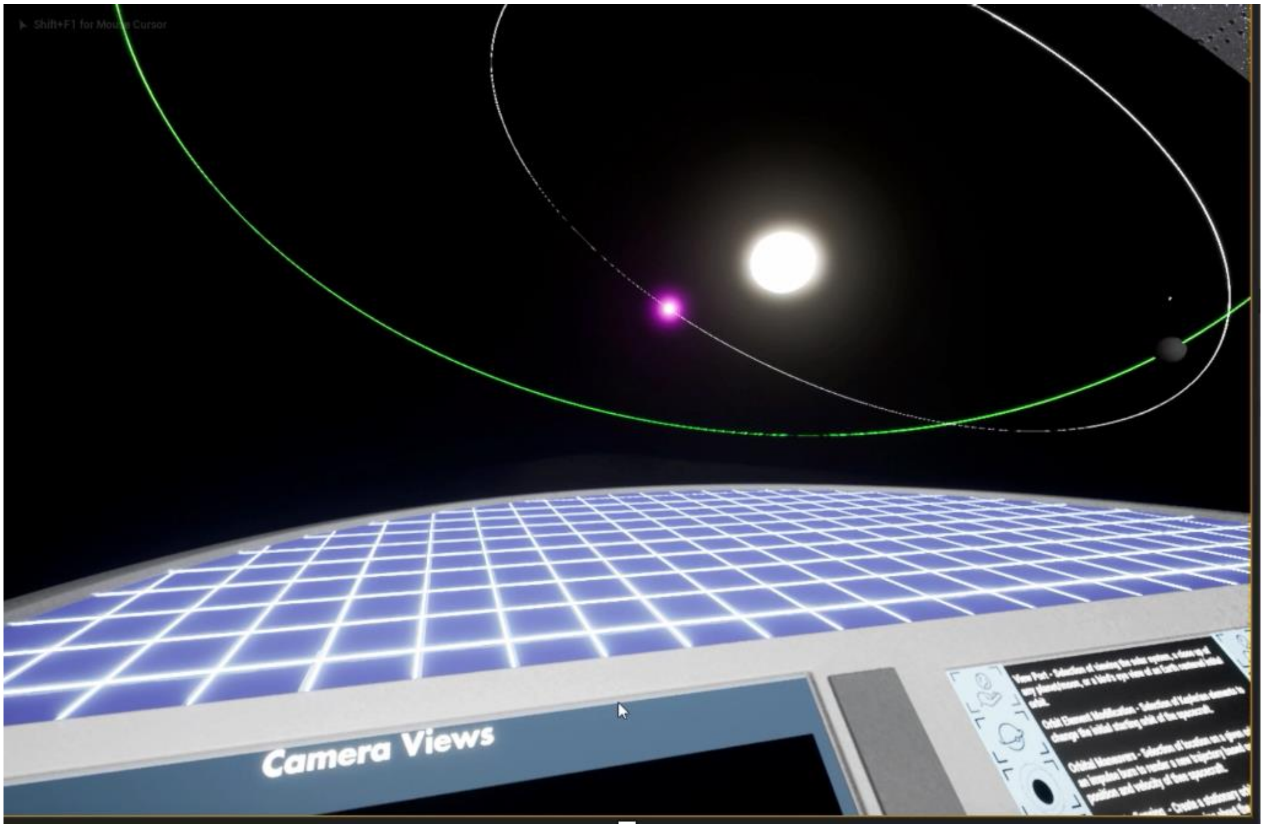


Figure 53: Spaceflight Holotable Rendering of Selected Parking Orbit about Jupiter With Io Orbit Shown (in green)

Figure 54: Spaceflight Console of Pilot Input for Orbital Insertion into Jupiter Sphere of Influence

These two parameters provide an output of information regarding the selected orbit for the crew to reference or change inputs off of. Coupled with the onscreen interaction, the holotable can render the approaching trajectory, point of impulse burn, and resulting parking orbit. Crew can also simulate the timeline through this sequence, viewing their spacecraft travel the projected path on the holotable, allowing crew to speed up the simulated time or pause to analyze certain orientations or maneuvers. With this capability and situational awareness of the mission, crew can accomplish the same planning and functions as mission control. They no longer rely solely on ground control calculations and communication links, and they have the added benefit of visually comprehending each stage of their mission trajectory.

The next stage being depicted follows similar to the first mission stage of selecting a target planet to arrive at. Using the perspective views, the crew can select to display Earth and the moon on the holotable and begin targeting from LEO to enter within a parking orbit about the moon. Switching back to an Earth centered orbit, crew can select from the same drop down of bodies to target Earth’s moon within a given time of travel. On screen, crew can read off the deltaV impulse to apply, the optimal node location for an impulse maneuver, and the launch window parameters for execution. Since they are initially in a parking orbit, the controls, inputs, and outputs for targeting the moon is the same process as the first steps when targeting Jupiter. Again, the resulting trajectory path is displayed on the holotable in front of the crew displays, and the crew can simulate movement through the shown path.

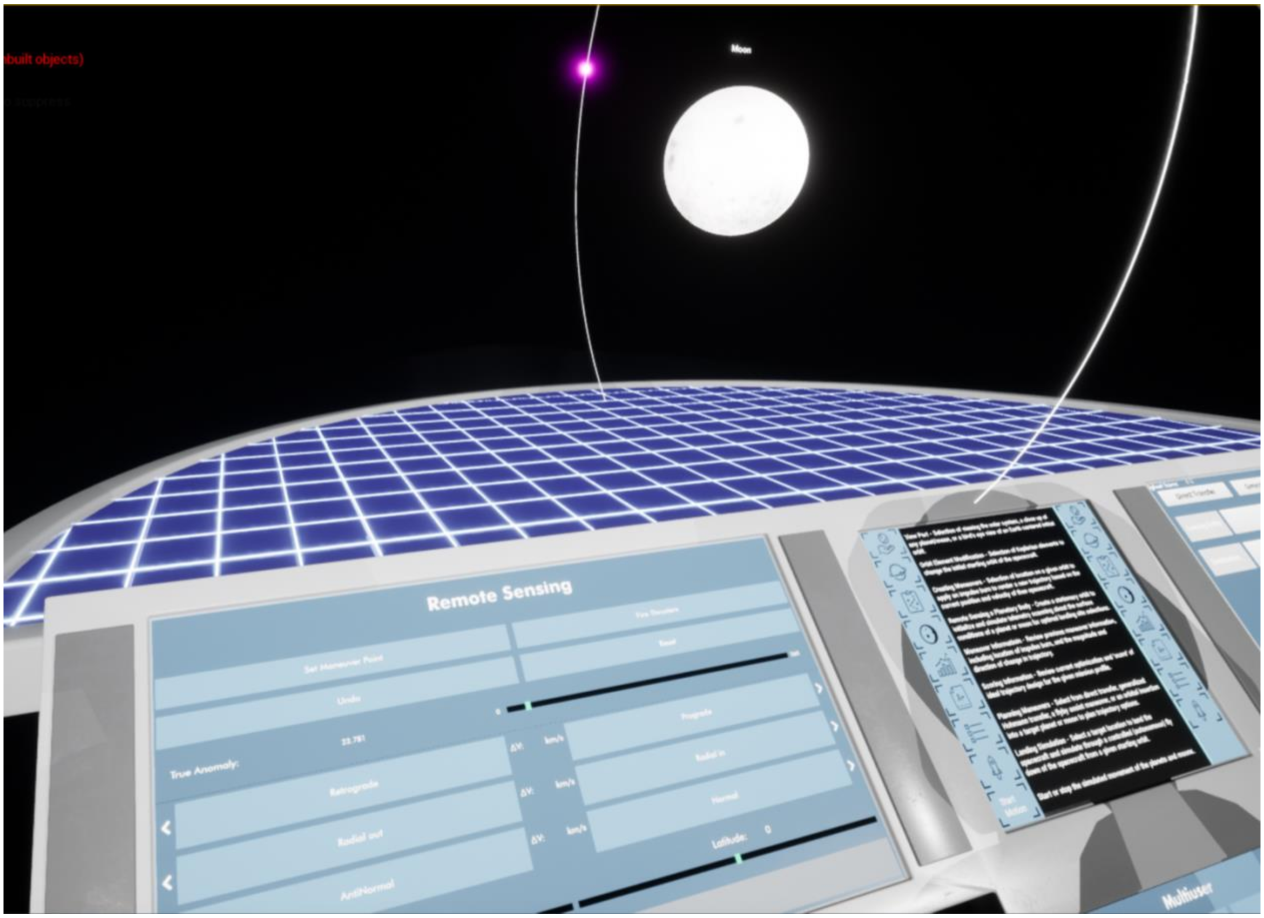


Figure 55: *Spaceflight Holotable Rendering of a Polar Orbit about the Moon*

Upon arrival, crew can establish a set of impulse maneuvers to enter into a parking orbit to begin satellite tracking and detailed mapping of the moon. For constructing remote sensing, crew can start and simulate a scan based on their current orbit from the onscreen console touchscreen buttons. On the holotable, the swath of their orbit will reveal detailed mapping of the surface of the moon, uploading the information to the onscreen console. This topographical information helps the crew select or simply verify the landing conditions for a target site.

With future iterations, this ability to orbit a rogue planet or moon, gathering data about the surface conditions, and visually conveying that information to crew will be essential for distant entry, descent, and landing mission profiles. Especially for unmapped planetary bodies, or for targeted bodies that are extremely far away from Earth ground stations, obtaining and storing all the relevant data onboard the spacecraft will greatly reduce the reliance crew has on mission control, while also providing real-time understanding of the current mission design.

Finally, to complete a full mission sequence, a controlled descent to landing maneuver is conducted. Using the selected landing site coordinates, crew can implement an automated landing simulation to safely land their spacecraft. As shown below, the landing site can be selected on screen, with the resulting trajectory to surface rendered on the holotable.

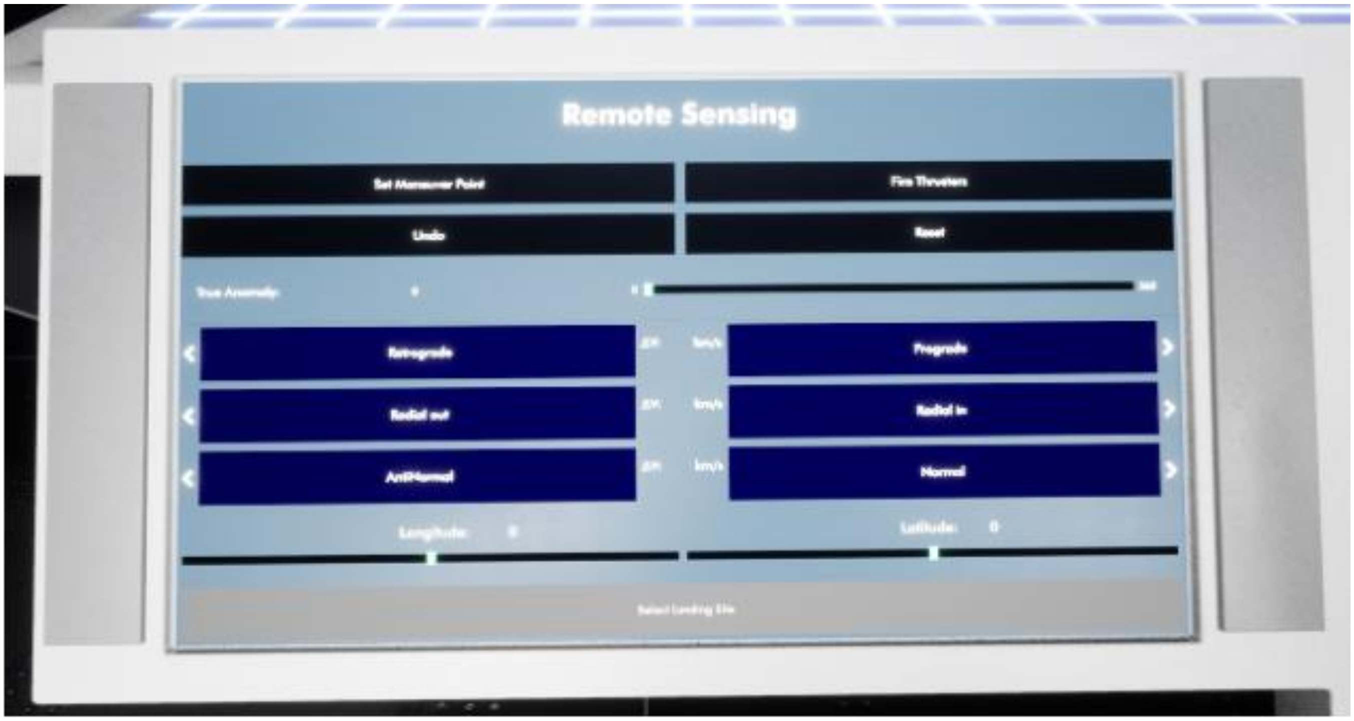


Figure 56: Spaceflight Console of Landing Site Selection from Remote Sensing Panel

Along with the visual aid, each pilot can also obtain on screen their projected path, abort options, relative velocity and location to target, and additional information regarding the impact and structure of their spacecraft.

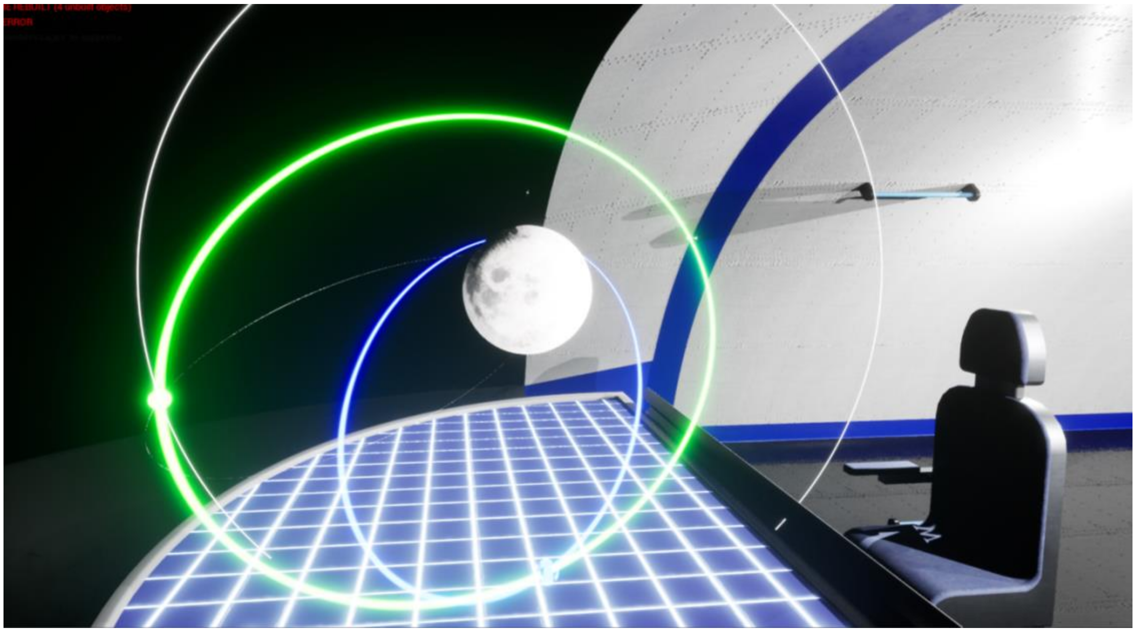


Figure 57: Spaceflight Holotable Maneuver Planning to Landing Site on Moons Surface

In sum, crew can effectively plan out an entire end-to-end mission sequence. Constructing an initial low Earth orbit to set an interplanetary maneuver node from, coupled with trajectory planning and guidance aid, crew can plan the start of their mission, modifying their impact parking orbit, and further adding in the sequence of events to target a moon for remote sensing, selecting a landing location, and descending to a surface target site.

VIII. RESEARCH LIMITATIONS

In the proposed form, the Spaceflight console includes several limitations. For the purpose of this thesis, the “Control” aspect of Guidance, Navigation, and Control will not be considered. The Spaceflight console is a guidance and navigation tool. The manual and autonomous side of controlling the spacecraft for such planned objectives will not be demonstrated. There are certain mission profiles outlined, like rendezvous with another spacecraft, that might require the crew to manually fly their spacecraft with a joystick or hand controllers. These inputs are described in the design of the console, but the specific rendering of the control interface is beyond the scope of this thesis. As such, this thesis aims to demonstrate the overall abilities of a novel Spaceflight console. This thesis does not demonstrate all capabilities for a complete flight-worthy system but will simply demo the concept of a guidance and navigation flight deck.

With the very conception of such a tool, there are a few known difficulties when planning and carrying out orbital maneuvers. The first of these would be the counterintuitive nature of orbital motion as experienced in a relative reference frame. [27] ‘From the perspective of an aircraft pilot, one would assume that a thrust in a forward direction would result in straightforward motion. However, a forward thrust moves the craft upward into a higher orbit. Since objects in higher orbit move more slowly than objects in lower orbit, the craft's eventual relative motion is backward, not forward.’ This can be overcome with simple training of spacecraft orbital mechanics, or with a displayed message on the control panel to inform the crew the orientation is as expected based on the thrust applied.

Another difficulty with a manned control of the spacecraft attitude pertains to safety factors. The first draft of the proposed tool may not factor in safety warnings that can occur due to human error or miscalculation of proximity to other orbiting entities. As such, future iterations of a control panel navigation tool should include warnings and critical sensor alerts that aid in 1) preventing maneuvers within a certain distance to any satellite body, 2) conducting attitude change that renders certain subsystems inoperable, 3) ensuring the plume of a burn does not harm the spacecraft structure or equipment, or that of nearby satellites, 4) and that there is zero relative velocity between two rendezvousing spacecraft. Most of these safety features can be implemented via warnings displayed to the crew, and break error catches within the background code that can trigger such a warning.

Other limitations to the first implementation of this navigation tool stem from the computer science aspect of the tool, as this interface is heavily dependent on SpaceCRAFTs platform. In its current state, the platform functions well with windows operating system (OS), but is still in development with Mac OS and Linux, and requires a minimum level of computational and hardware specs to run optimally [28] As the platform evolves and further develops, so will the capabilities of the trajectory interface, but its dependence on SpaceCRAFT will be inherently tied.

Furthermore, for an initial demonstration, the Spaceflight console will use approximated calculations for satellite orbit determination (OD). This is typical of dynamic models, which factor in the given environment for further accuracy. For instance, determining precise LEO positionings include additional geopotential and atmospheric drag models, along with their errors, for more accurate interpolation between location points [29]. The orbital paths are not propagated using high fidelity techniques like Runge-Kutta or Extended Kalman Filtering. Instead, iterating through a closed path of 360 degrees for true anomaly, finding the location and velocity at each degree, provides the underlying formulation of the orbits, with special cases taken to eccentric values above one. The impact is two-fold. The user cannot simulate their spacecraft moving through each point location on the orbit, and the exact position and velocity of each maneuver in Earths SOI will incur a slight numerical error when contrasted to high fidelity GPS models. Such high-fidelity calculations will be integrated into the console in future developments, but for the purpose of demonstrating the initial concept of the tool, the ability to move through orbits and maneuvers without time-dependence is given higher priority.

Further improvements of a Spaceflight console might include backend algorithms that allow the crew to modify their environment or input the material make-up of the spacecraft. Specific factors, like atmospheric drag or protection from radiation on long duration trajectories, are not considered in the scope of this demonstration, mainly because these parameters are typically not considered in lambert targeting algorithms and mission planning as a whole, with most baseline calculations only using gravitational parameters as an additional input [30] [31].

Finally, upon first pass, this simulated instrument will be to apply it to Science, Technology, Engineering, and Math (STEM) education through the ASTROLab's STEM outreach programs. The idea of this novel instrument will likely take many iterations and improvement before it can be considered a fully comprehensive and flight-ready guidance tool. This initial design also will not be tested in any user experiments. Later installments and versions should include such testing in order to provide constructive feedback and possible improvements on the layout and user-friendliness of the instrument.

IX. SUMMARY AND FUTURE WORK

As demonstrated, the Spaceflight Console provides crew the ability to guide and navigate their spacecraft through a variety of different mission objectives. At each step, crew can view information on a touchscreen display directly in front of them, as well as view a 3D holographic rendering of the environment and corresponding data. For an intuitive console, crew should have the option to input or control their spacecraft through touchscreen and manual buttons/joysticks. Coupled with this, the returning output of information should be conveyed in a mixed format, displaying numerical values or warnings on the pilot's screen while also rendering a visual graphic of the situation on the holotable in front of the pilots' screen. Accounting for all possible methods of inputting and outputting information to crew is crucial to future spaceflight. Pilots will need to be able to handle future long duration spaceflight wherein crew manages GNC functions without actually needing to be in contact with ground stations.

In its current state, the spaceflight console provides the ability to 1) design initial parking orbits, 2) plan and execute impulse burn maneuvers within a bodies sphere of influence or targeting interplanetary bodies, 3) evaluate different trajectory options for direct transfers, fly-bys, or ideal Hohmann situations, 4) maneuver into a new parking orbit and initiate remote sensing for analyzing the surface of a given body, 5) simulate a controlled entry and descent to a chosen landing site. Each of these functions can be tied into a sequence of events crew can plan from stage one. Alternatively, crew can change mission profiles or targets mid-trajectory, allowing for crew to react and account for new information or changes to their tasks.

As shown through the demonstration, the Spaceflight Console contributes to giving crew a visualization tool and interface to account for a range of objectives. It operates onboard, without needing to communicate with ground stations. The console is also a conceptual design, allowing for more advanced controls and displays than might be feasible to construct today. With this advancement, it gives pilots multidimensional optimization methods for a variety of maneuvers, namely by displaying trajectories in a holographic manner while also providing interactive heat map and 3D plots for a plethora of information. Finally, this concept is simulated in a virtual reality environment, allowing further evaluation of the concept and testing and training of the console.

As the spaceflight console continues to expand, further development into proximity and docking procedures will be further integrated into the pilot's control. This includes warnings during manual control, and the ability to open communication links to the docking vehicle. As touched upon previously, a docking simulation akin to SpaceX docking console can easily be implemented into the pilot's touchscreen display. The holotable can also aid in rendering a 3D birds eye view of the docking approach to give the crew additional guidance as they approach.

For the interplanetary trajectory calculations, there is still work left to implement the MATLAB generated 3D or heat map plots into the simulation in Unreal Engine. This provides the unique ability for pilots to evaluate trade-offs and use a range of information for determining the most optimal path. And specifically for the gravity assist maneuver calculations, modifications to the computation for the launch window against time till planetary alignment needs to be corrected and verified its output.

Furthermore, the entry, descent, and landing maneuvers can continue to be expanded upon to allow the crew more control and dictation toward their landing sequencing. At the moment, the spaceflight console allows crew to input a landing site, and then the rest is a controlled, automated landing simulation. Adding in manual landing control, either via on screen controls or joystick and throttle, crew will soon need the ability to land their spacecraft without precomputed conditions or reliance on ground control for the maneuver.

Finally, integrating aerodynamic physics libraries to include drag, perturbations, and radiation factors among others would help make the tool even more realistic and reliable for carrying out the prescribed mission plans. Adding higher fidelity and propagation tools would be a rather straightforward next step to further increase the precision and accuracy of the console's outputs and model.

Nevertheless, this tool will constantly be improved upon and expanded to encompass future mission needs. This thesis provides an initial demonstration of its current capabilities and aims to outline how necessary such a guidance and navigation tool will become as the space industry expands and commercializes.

REFERENCES

[1] Oberhaus, Daniel. “How NASA Certifies New Spacecraft Safe Enough for Humans.” *Wired*, Conde Nast, 13 May 2020, https://www.wired.com/story/how-nasa-certifies-newspacecraft-safe-enough-for-humans/.

[2] Hoffpauir, Daniel. “Space Rendezvous.” *NASA*, NASA, 23 May 2016, https://www.nasa.gov/offices/nesc/articles/space-rendezvous.

[3] Dunbar, Brian. “Docking with Precision.” *NASA*, NASA, 23 Mar. 2004, https://www.nasa.gov/missions/shuttle/f_docking.html.

[4] “A 21st Century Space Capsule.” *Boeing*, https://www.boeing.com/space/starliner/.

[5] Patel, Neel V. “NASA: A Second, Unreported Glitch Could Have Wrecked Boeing's Starliner.” *MIT Technology Review*, MIT Technology Review, 2 Apr. 2020, https://www.technologyreview.com/2020/02/07/349022/nasa-a-second-unreported-glitchcould-have-wrecked-boeings-starliner/.

[6] “Apollo 11.” *Wikipedia*, Wikimedia Foundation, 19 Nov. 2021, https://en.wikipedia.org/wiki/Apollo_11.

[7] Ryan-Mosley, Tate. “The Number of Satellites Orbiting Earth Could Quintuple in the next Decade.” *MIT Technology Review*, MIT Technology Review, 30 Apr. 2021, https://www.technologyreview.com/2019/06/26/755/satellite-constellations-orbiting-earthquintuple/.

[8] Harada, Mark. “How Many Planes Are in the Air Right Now?” *Traveltalk*, 30 Mar. 2020, https://www.traveltalkmag.com.au/news/how-many-planes-are-in-the-air-right-now.

[9] Garcia, Mark. “Space Debris and Human Spacecraft.” *NASA*, NASA, 14 Apr. 2015, https://www.nasa.gov/mission_pages/station/news/orbital_debris.html.

[10] Jerez, Juan, et al. *FORCES-RTTO: A TOOL FOR ON-BOARD REAL-TIME AUTONOMOUS TRAJECTORY PLANNING*. 2017, https://www.embotech.com/wpcontent/uploads/ESA_GNC2017_Jerez.pdf.

[11] admin, Written by, and Written by. “Showing What VOR and ILS Aviation Signals Look like in SDR#.” *Rtl*, 2 Oct. 2017, https://www.rtl-sdr.com/showing-what-vor-and-ilsaviation-signals-look-like-in-sdr/.

[12] “Tactical Air Navigation System.” *Wikipedia*, Wikimedia Foundation, 10 Aug. 2021, https://en.wikipedia.org/wiki/Tactical_air_navigation_system.

[13] "U.S. Pilot S Guide. for the TACAN. Tactical Air Navigation System. Model AN/ARN154(V)." *PDF Free Download*, https://docplayer.net/44061048-12-00-u-s-pilot-s-guidefor-the-tacan-tactical-air-navigation-system-model-an-arn-154-v.html.

[14] Dumoulin, Jim. "Communications." *NASA*, NASA, 2000, https://science.ksc.nasa.gov/shuttle/technology/sts-newsref/sts-jsc-comm.html#sts-comm.

[15] "Communications System." *What Interferes with Radio Waves in Space?*, https://www.qrg.northwestern.edu/projects/vss/docs/communications/1-what-interfereswith-radio-waves.html.

[16] "Space Shuttle Orbital Docking System - NASA." *Math and Science @ Work*, https://www.nasa.gov/pdf/593865main_AP_ST_Phys_ShuttleODS.pdf.

[17] Goodman, John. "Space Shuttle Navigation in the GPS Era." *NASA Technical Reports Server (NTRS)*, NASA, https://ntrs.nasa.gov/api/citations/20100039589/downloads/20100039589.pdf.

[18] Goodman, John. "Operational Use of GPS Navigation for Space Shuttle Entry ." *NASA Technical Reports Server (NTRS)*, NASA, 2008, https://ntrs.nasa.gov/citations/20080014095.

[19] Chamitoff, Greg. "Astrodynamics." Part 5 - Targeting Maneuvers. AERO 423 Class Lecture, 21 Nov. 2021, College Station.

[20] "SpaceX Launches Crew with Recycled Rocket and Capsule | CBC News." *CBCnews*, CBC/Radio Canada, 23 Apr. 2021, https://www.cbc.ca/news/science/spacex-recycledrocket-crew-1.5999317.

[21] *Boeing Shuttle Flight Deck*. Boeing/ PBS, 2021, https://pbs.twimg.com/media/CllgGjrUsAAJdjf.jpg. Accessed 21 Nov. 2021.

[22] "How a Touchscreen Controls SpaceX's New Spacecraft." *YouTube*, CNET, 26 May 2020, https://www.youtube.com/watch?v=pgxDH_LNmPw. Accessed 21 Nov. 2021.

[23] Dobry, Wojciech. "Future UI Design without Buttons." *Toptal Design Blog*, Toptal, 28 Sept. 2017, https://www.toptal.com/designers/ui/future-ui-design-without-buttons.

[24] "ISS Docking Simulator." *SPACEX*, 2020, https://iss-sim.spacex.com/.

[25] Moon, Alex. "Launch Window Planner for Kerbal Space Program." *KSP Launch Window Planner by Alexmoon*, https://alexmoon.github.io/ksp/.

[26] Curtis, Howard D. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 2021.

[27] Soller, Jeffrey, et al. “A Trajectory Planning Scheme for Spacecraft in the Space Station Environment.” *NTRS - NASA Technical Reports Server*, NASA , 1991, https://ntrs.nasa.gov/api/citations/19910007758/downloads/19910007758.pdf.

[28] Jakubik, C. (2021). *Build software better, together*. GitHub. Retrieved November 4, 2021, from https://github.tamu.edu/SpaceCRAFT/Platform/wiki/Setup-Platform-Devhttps://github.tamu.edu/SpaceCRAFT/Platform/wiki/Setup-Platform-Dev-EnvironmentEnvironment.

[29] Shou, Ho-Nien. “Orbit Propagation and Determination of Low Earth Orbit Satellites.” International Journal of Antennas and Propagation, Hindawi, 8 May 2014, https://www.hindawi.com/journals/ijap/2014/903026/.

[30] *Free Flyer University Guide*. A.I. Solutions Inc, 2016, http://trylam.com/wpcontent/uploads/2017/01/FreeFlyerUniversityGuide.pdf.

[31] Bate, Roger R., Donald D. Miller, and Jerry E. White. *Fundamentals of Astrodynamics*. New York, NY: Dover Publications, Inc., 1971.

[32] Received by Nikita Beebe, Re: Proposal, 16 Nov. 2021. ISS Navigation Display

[33] Brazzel, Jack. “The Rendezvous Monitoring Display Capabilities of the Rendezvous and Proximity Operations Program .” *NASA*, NASA Technical Reports Server (NTRS), 2013, https://ntrs.nasa.gov/citations/20130003242.

[34] Mccandless, Jeffrey & Hilty, Bruce & Mccann, Robert. (2005). New Displays for the Space Shuttle Cockpit. Ergonomics in Design: The Quarterly of Human Factors Applications. 13. 15-20. 10.1177/106480460501300405.

[35] “Ground Segment.” *Wikipedia*, Wikimedia Foundation, 7 Dec. 2021, https://en.wikipedia.org/wiki/Ground_segment.

[36] “Boeing CST-100 Starliner Crewed Spacecraft - Collectspace: Messages.” *CollectSPACE*, NASA, 2014, http://www.collectspace.com/ubb/Forum35/HTML/000430.html.

[37] James, Rick. “Aircraft Cockpits: Why Are There so Many Buttons!!?” *Pilot Teacher*, 24 Jan. 2022, https://pilotteacher.com/aircraft-cockpits-why-are-there-so-many-buttons/.

[38] Mathi, Sarvesh. “The Touchscreens Controlling SpaceX Dragon on Its Historic Mission.” *Medium*, The Startup, 20 June 2020, https://medium.com/swlh/the-touchscreens-controlling-spacex-dragon-on-its-historic-mission-b0546d26053c.

[39] Seemangal, Robin. “NASA and Microsoft Bring Hologram Technology to the International Space Station.” *Observer*, Observer, 7 Dec. 2015, https://observer.com/2015/12/nasa-and-microsoft-bring-hologram-technology-to-the-international-space-station/.

[40] Mathi, Sarvesh. “The Touchscreens Controlling SpaceX Dragon on Its Historic Mission.” *Medium*, The Startup, 20 June 2020, https://medium.com/swlh/the-touchscreens-controlling-spacex-dragon-on-its-historic-mission-b0546d26053c.

[41] “Gravity Assist.” *Wikipedia*, Wikimedia Foundation, 18 Apr. 2022, https://en.wikipedia.org/wiki/Gravity_assist.

[42] “Maneuvering In Space.” *Office of Aerospace Medicine | Federal Aviation Administration*, https://www.faa.gov/about/office_org/headquarters_offices/avs/offices/aam/cami/library/online_libraries/aerospace_medicine/tutorial/media/III.4.1.5_Maneuvering_in_Space.pdf.

[43] Li, Peng, and Zheng H. Zhu. “Model Predictive Control for Spacecraft Rendezvous in Elliptical Orbit.” *Acta Astronautica*, Pergamon, 13 Mar. 2018, https://www.sciencedirect.com/science/article/abs/pii/S0094576517309426?via%3Dihub.

[44] “GPIS Trajectory Rendering .” *OrbiterWiki*, https://www.orbiterwiki.org/wiki/Main_Page.

[45] *Image: Projected Arrival Hyperbola into Parking Orbit about Planet*. https://ars.els-cdn.com/content/image/1-s2.0-S1674205215003664-gr3.jpg.