Constellation Training Facility Support

Jose M. Flores[1]
University of Puerto Rico, Mayaguez, Puerto Rico, 00680

[1] Undergraduate Student, Department of Mechanical Engineering, University of Puerto Rico, Mayaguez Campus

Abstract

The National Aeronautics and Space Administration is developing the next set of vehicles that will take men back to the moon under the Constellation Program. The Constellation Training Facility (CxTF) is a project in development that will be used to train astronauts, instructors, and flight controllers on the operation of Constellation Program vehicles. It will also be used for procedure verification and validation of flight software and console tools. The CxTF will have simulations for the Crew Exploration Vehicle (CEV), Crew Module (CM), CEV Service Module (SM), Launch Abort System (LAS), Spacecraft Adapter (SA), Crew Launch Vehicle (CLV), Pressurized Cargo Variant CM, Pressurized Cargo Variant SM, Cargo Launch Vehicle, Earth Departure Stage (EDS), and the Lunar Surface Access Module (LSAM). The Facility will consist of part-task and full-task trainers, each with a specific set of mission training capabilities. Part task trainers will be used for focused training on a single vehicle system or set of related systems. Full task trainers will be used for training on complete vehicles and all of its subsystems. Support was provided in both software development and project planning areas of the CxTF project. Simulation software was developed for the hydraulic system of the Thrust Vector Control (TVC) of the ARES I launch vehicle. The TVC system is in charge of the actuation of the nozzle gimbals for navigation control of the upper stage of the ARES I rocket. Also, software was developed using C standards to send and receive data to and from hand controllers to be used in CxTF cockpit simulations. The hand controllers provided movement in all six rotational and translational axes. Under Project Planning & Control, support was provided to the development and maintenance of integrated schedules for both the Constellation Training Facility and Missions Operations Facilities Division. These schedules maintain communication between projects in different levels. The CxTF support provided is one that requires continuous maintenance since the project is still on initial development phases.

I. Introduction

The following report entails a full description of the development support provided to the Constellation Training Facility at the National Aeronautics and Space Administration (NASA), Lyndon B. Johnson Space Center through the Undergraduate Student Research Program (USRP). This was a 15 week internship that consisted in both technical and professional experiences under the supervision of mentors which contributed to the ongoing development of the Constellation Program. This report serves as an overview of the Constellation Training Facility and as an explanation of the achievements and experiences acquired during the internship. A detailed description on assigned tasks and tools utilized will be presented. It is expected from the reader, after reading thoroughly this report, to have a better understanding of one of the main projects that will enable astronauts to go back to the Moon, and to have an insight on the contributions made to achieve these goals.

II. Constellation Training Facility

The Constellation Training Facility will provide a venue for training crew members, flight controllers, instructors, and other identified personnel associated with the Constellation Program. This facility will include training devices and supporting infrastructure for the simulation of mission operation tasks required to perform ISS, Lunar, and eventually, Mars missions. The CxTF will provide the resources to support nominal and contingency operations for all mission phases of the Constellation Program.

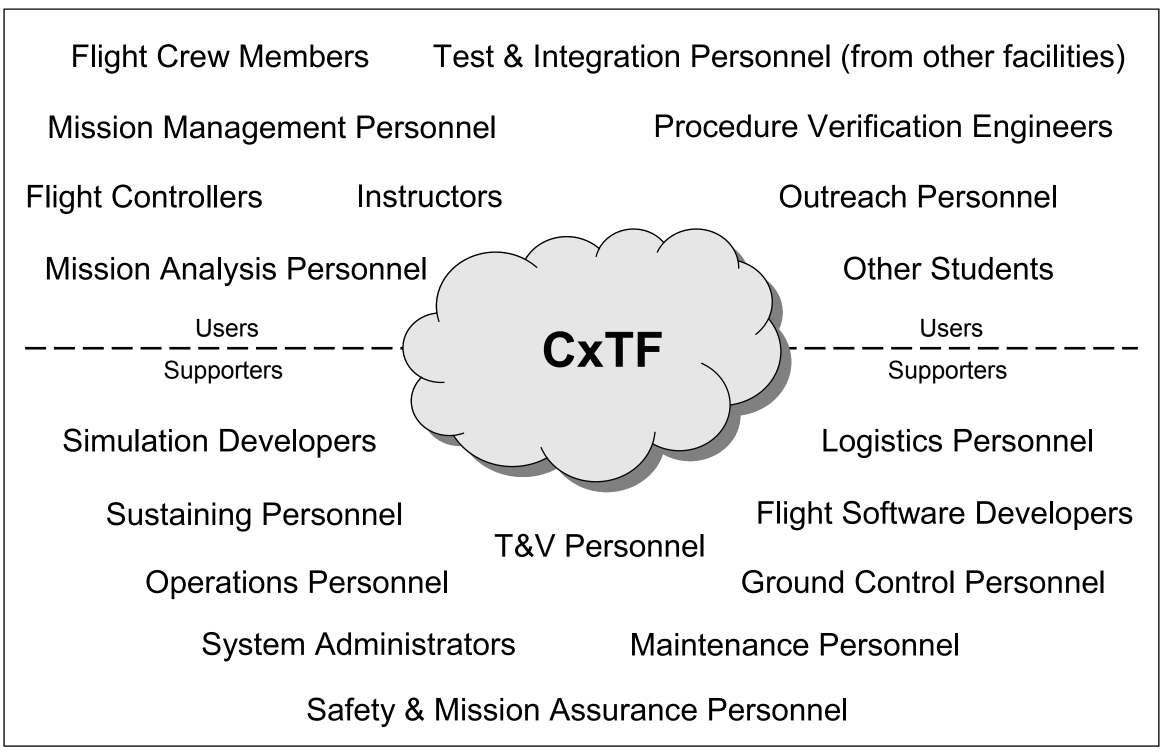


Figure 1. CxTF Stakeholders

The Constellation Program consists of multiple elements. The launch vehicles include the Ares I Crew Launch Vehicle (CLV) and the Ares V Cargo Launch Vehicle (CaLV). The crew vehicles include the Orion Crew Exploration Vehicle (CEV) and the Lunar Lander (Altair). There will be an Earth Departure Stage (EDS) for lunar missions. Finally, the Lunar Surface Systems (LSS) will consist of all additional systems associated with lunar surface operations. The crews for each mission will also interface with Extra-Vehicular Activity (EVA) suits and

Flight Crew Equipment (FCE).

The Constellation Training Facility simulations and training devices will be developed for all Constellation elements, including Ares I, Ares V, Orion, Altair Lander, EDS and LSS. The CxTF will initially support ISS and Lunar Sortie missions, with Lunar Outpost and Mars mission simulation capability in the long term. The trainers will run in conjunction with the Mission Control Center (MCC) and/or the Space Station Training Facility (SSTF). The development philosophy behind CxTF consists of an evolutionary increase of capability over the life of the Constellation program.

The CxTF will provide both part-task (PTT) and full-task (FTT) training environments. The PTT environment will support crew, flight controller, instructor, and other identified personnel training on partial spacecraft systems. The FTT environment will provide full-system and mission training for flight crews and flight control teams.

The PTT environment includes the Orion Part-task Trainers (OPT), Lunar Part-task Trainers (LPT), and Flight Controller Part-task Trainers (FCPT). The OPT and LPT are primarily intended for crew training, while the FCPT is designed for flight controller training. OPT and LPT will include instructor stations with capabilities for malfunction insertion/removal and simulation control. It will also include a crew station, with various degrees of fidelity, depending on the training needs. Some trainers will include image generation capabilities to support crew dynamic skills training. Part-task training simulations may execute real flight software or functional flight software, depending on the application and specific user requirements.

The FTT environment includes the Orion Mission Simulation (OMS) and Lunar Mission Simulation (LMS), and will include high-fidelity Orion and Lander crew stations, full high-fidelity image generation, aural cue capability, and instructor stations co-located with the crew stations. These FTT devices will execute real flight software as a default, but will include the capability to execute functional flight software as required.

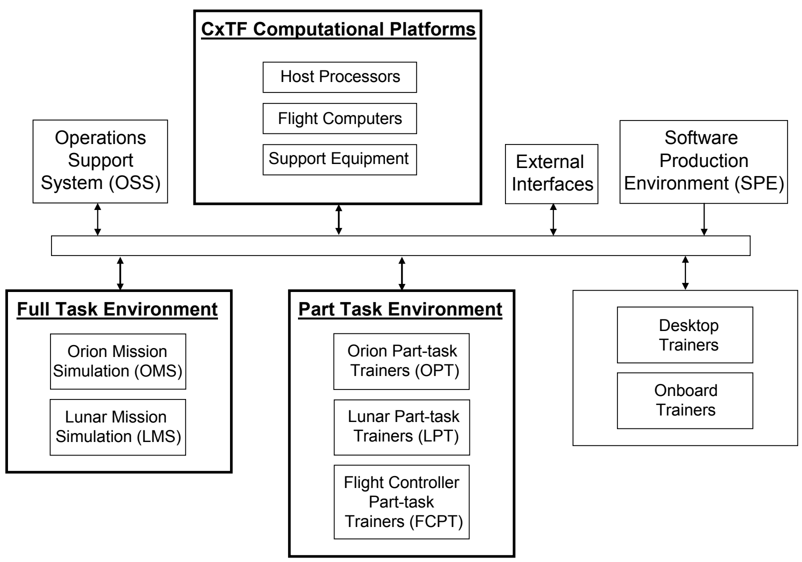


Figure 2. Conceptual CxTF Architecture Overview

Training for the Constellation Program will also make use of various additional training applications, including Desktop and Onboard trainers. The Desktop Trainer is envisioned as a capability for a user to access PTT simulations directly from an office personal computer (PC). The Onboard Trainer is envisioned as a PTT simulation that will run on a laptop computer onboard the flight vehicle.

The CxTF will also provide a platform for procedure verification and test and integration support for other Constellation facilities, such as the MCC at the Johnson Space Center (JSC) or the Launch Control Center (LCC) at the Kennedy Space Center (KSC).

As the Constellation Program continues to mature and formal crew and flight controller task analyses are completed, training objectives and simulator requirements will continue to be evaluated and developed. The overall CxTF design concept will be developed to be able to readily adapt to future program and training requirement needs.

Below is presented a list of essential features of the Constellation Training Facility.

- Part task trainers - for training focused on a single vehicle system or set of related systems
- Full task trainers - for training on complete vehicle(s) and all of the subsystems
- Instructor/Operator stations (IOS) - location that provides an individual instructor, or a team of instructors under the direction of a Team Lead, the ability to monitor, control, and mode the associated full task/part task training facility, and provides the instructor(s) monitoring access to crew and flight controller displays.
- Crew stations - portion of the full task trainer which is a full-scale replica of the CEV crew module, with identical cockpit layout and functional controls and displays. Also, portion of the part task trainer which has a set of controls and displays tailored for crew training in the part task training environment.
- Student stations - portion of the part task trainer which has a set of controls and displays tailored for flight controller training in the part task training environment
- Simulation Control Area (SCA) - provides (a) a team of instructors under the direction of a Simulation Supervisor (Sim Sup) with the ability to monitor flight control disciplines and to monitor, direct, and control (if necessary) the full task training facility working under the Sim Sup's direction, (b) real-time mission monitoring capability of any flight control discipline for instructors, and (c) a team or discipline-specific group of instructors, or individual instructors, with the ability to monitor and control part task training conducted using flight control consoles and assets. This may be located within the Mission Control Center complex.
- Flexible architecture - platform hardware and operating systems are available from multiple vendors
- Audio and video capabilities- to provide a realistic training environment, for communication between student and instructor, communication between CxTF and external facilities, and for out-the-window graphics
- Interfaces to the MCC and SSTF - for integrated/combined simulations
- Systems training - train students to operate the CEV/CLV vehicle systems in a safe, efficient manner
- Mission specific training - enable students to execute tasks planned for actual missions
- Team training - students train with other facilities such as the Mission Control Center (MCC) and/or the Space Station Training Facility (SSTF)
- Procedures verification - fidelity of the CxTF models and its display and controls are sufficient to support procedure verification

The Constellation Training Facility will be developed following a multiple stage approach. The initial phases will develop the CxTF to support missions to the ISS and subsequent phases will develop the CxTF to support lunar missions. The CxTF will evolve in capabilities as the Constellation program progresses. In the early phases of the Constellation program, prior to the first test flight, part-task trainers will be used to train flight controllers to react to command and telemetry streams that pass through the MCC. The part task trainers for flight controller training will be operational with sufficient lead time. Prior to the first manned mission to the ISS an Orion mission simulator, including real-world flight software, will be developed, tested and certified for training the crew and flight controllers. A reuse and commonality strategy will be employed by the development teams to leverage software models, hardware interfaces and data architectures to reduce the development and maintenance costs of the CxTF.

III. Simulation Software development

During the course of the internship software development support was provided to the Constellation Training Facility utilizing the Trick simulation environment. This is the main tool for simulation development that provides a suitable environment to achieve realistic simulations using mathematical models in real and non-real time. Trick enables ways to develop simulations of all the required components of the vehicles and deploy them simultaneously for training purposes. The following is an overview of the Trick simulation environment and its capabilities.

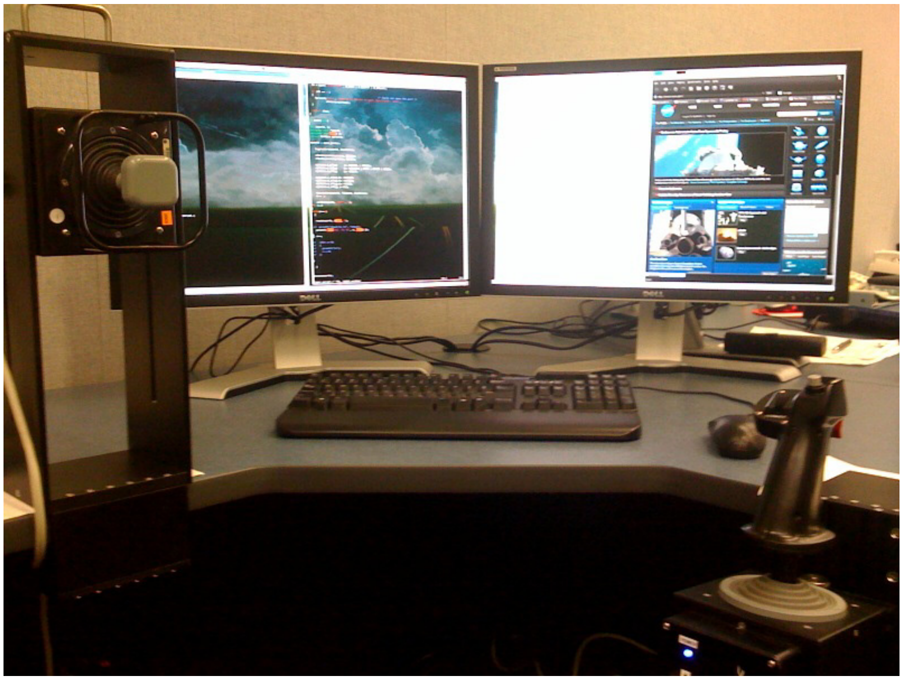


Figure 3. Workstation for simulation software development

A. Trick Simulation Environment

Trick is a generic simulation toolkit that can be used extensively for training astronauts through real-time interactive simulations. A Trick simulation is a run-time executive designed for both real-time and non-real-time applications having both time-based and event-based scheduling requirements, including hardware-in-the-loop, human-in-the-loop, multi-processor and multi-box applications.

Trick simulations are designed to be data driven wherever possible. The run-time executive provides a simple interface that allows the user to configure model parameters, integration schemes, and real-time controls for each application.

Trick takes the burden off the modeler to create simulation executive ware. Trick does this by fusing its own simulation executive with the developer's models. It takes the responsibility of creating all source code for run-time input, data recording output and data communication between model components. The developer's responsibility is reduced to build a simulation definition file and the model code. This plug-in approach coupled with multi-platform and multi-programming support allows modeling groups to create reusable, shared and versatile models across distributed locations.

Trick features the following:

- Supports Linux, MAC, SGI and Sun platforms
- Models developed in C/C++ may be linked in Ada and FORTRAN libraries
- Simulations may be driven by events defined by the developer through run-time input files.
- Simulations are configurable through textual C-like input files.
- Data logging.
- Simulations may dump and load simulation states as well.
- Models can interface with most hardware and link in most low level hardware drivers.
- Facilitates rapid development of high fidelity math models that are easily integrated into the Trick Executive.
- Jobs may be assigned to synchronous/asynchronous P-threads by tagging the job in the simulation definition file.
- Built in support for distributed simulations.

Source code was developed using trick to model the hydraulics system in the Thrust Vector Control System of the Ares I Launch Vehicle.

B. Ares I Thrust Vector Control Hydraulics simulation modeling

Simulation software using Trick was developed for the Thrust Vector Control (TVC) system of the upper stage of the Ares I Crew Launch Vehicle. Ares I is an in-line, two-stage rocket configuration topped by the Orion crew vehicle and its launch abort system. In addition to the vehicle's primary mission of carrying crews of four to six astronauts to Earth's orbit, it may also use its payload capacity to deliver resources and supplies to the International Space Station.

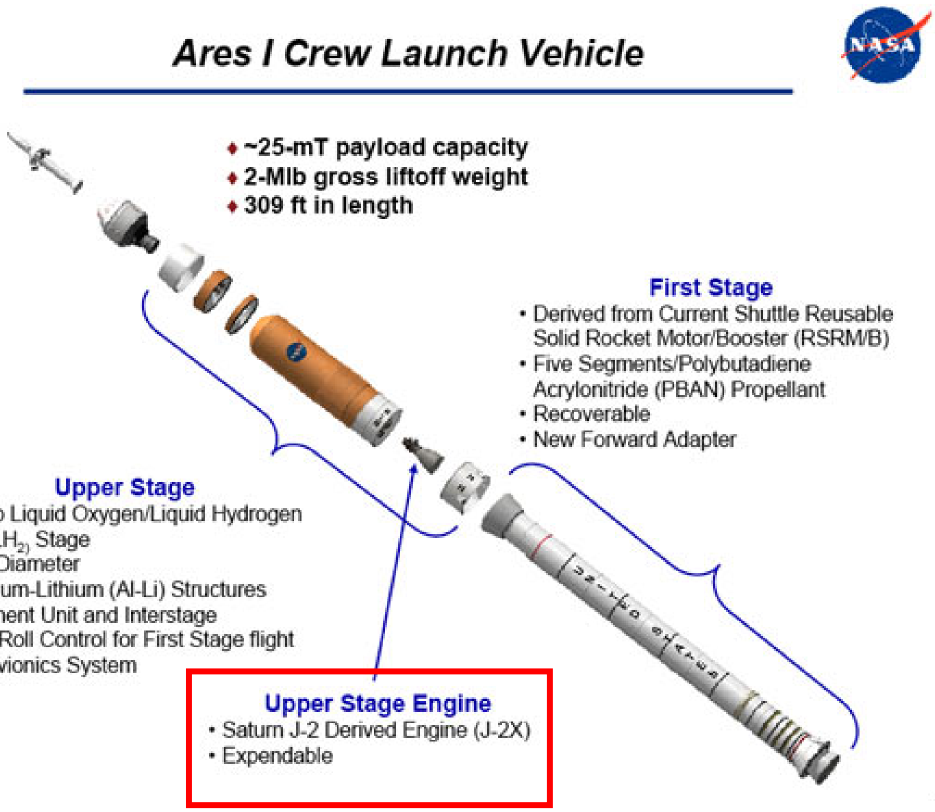


Figure 4. Ares I Crew Launch Vehicle.

During launch, the first-stage booster (See Figure 4) powers the vehicle toward low Earth orbit. In mid-flight, the reusable booster separates and the upper stage's J-2X engine ignites, putting the vehicle into a circular orbit. During the ascent stage the TVC system keeps the gimbals of the J-2X engine in a lock position and during nominal operations it provides maneuvering capabilities for the vehicle.

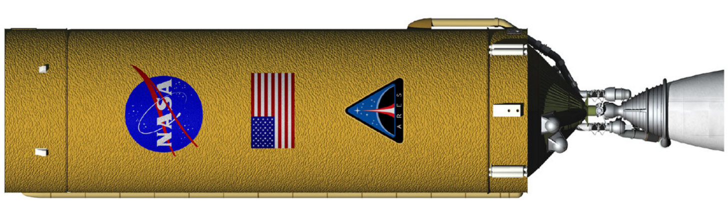


Figure 5. Ares I Upper Stage.

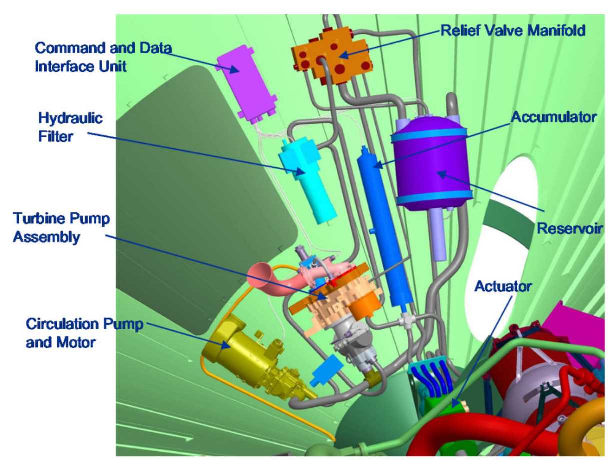


Figure 6. Hydraulic System for Thrust Vector Control of the J-2X engine.

Figure 5. Ares I Upper Stage.

The Upper Stage TVC system consists of various hydraulic components including hydraulic reservoirs, accumulators, circulation pumps, the Turbine Pump Assembly (TPA) and the hydraulic actuators (See Figure 7). Before launch, the gimbals are kept in a locked position using the circulation pumps that are powered externally by the Ground Control Equipment (GCE). During launch, the upper stage gimbals are locked using a tap off from the Main Propulsion System (MPS). Once The Upper Stage is separated the locks are removed thus providing gimbals functionality for vehicle maneuvering. Under nominal conditions the hydraulic actuators will provide the necessary force to maneuver the vehicle with pitch and yaw control.

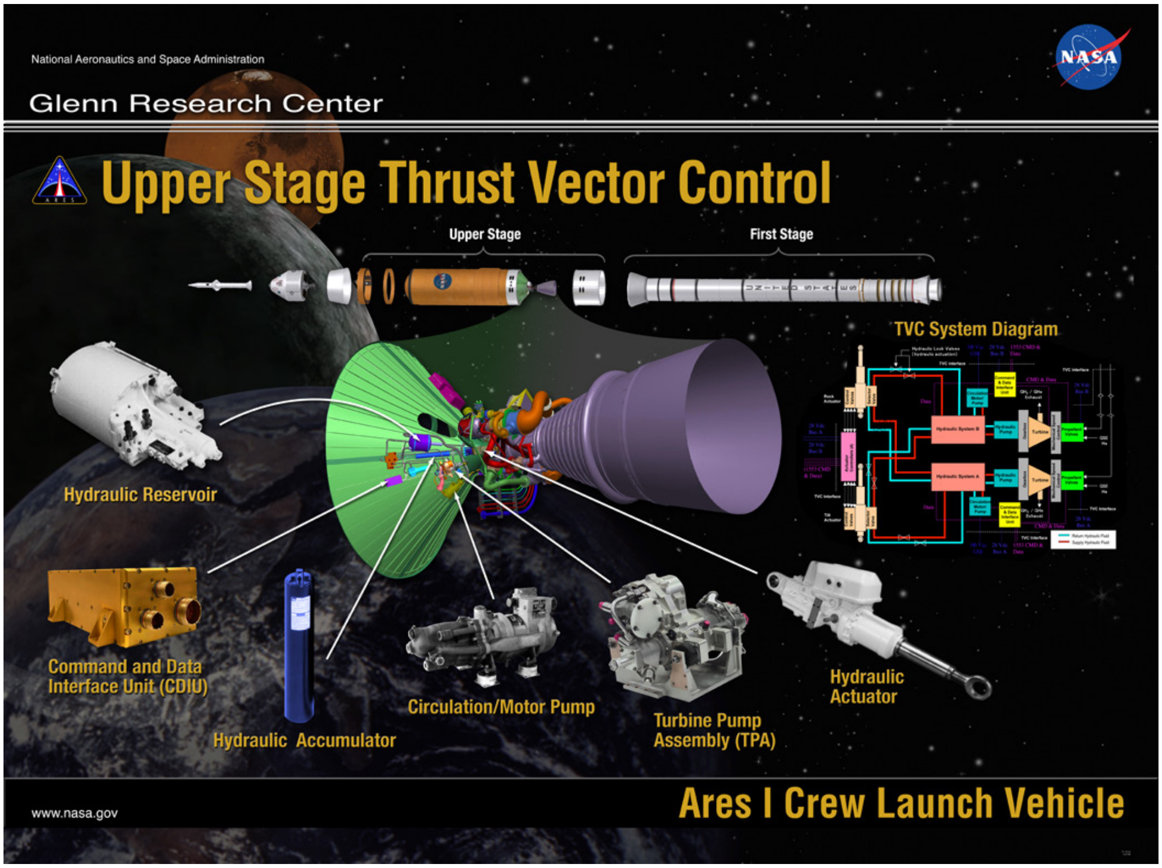


Figure 7. Upper Stage Thrust Vector Control Hydraulic System

The simulation software under development is in charge of monitoring the lock/unlock states as well as the pressure in the hydraulic system during its operation. The software provides malfunction flags that will be enabled as required by training to simulate malfunctions of the TVC. The state of this system will be fed to the Command and Data interface units for them to take the proper actions for each state. The Thrust Vector Control system is a critical system during orbit insertion of the vehicle thus it is required for it to be simulated with high fidelity for crew and flight controller training.

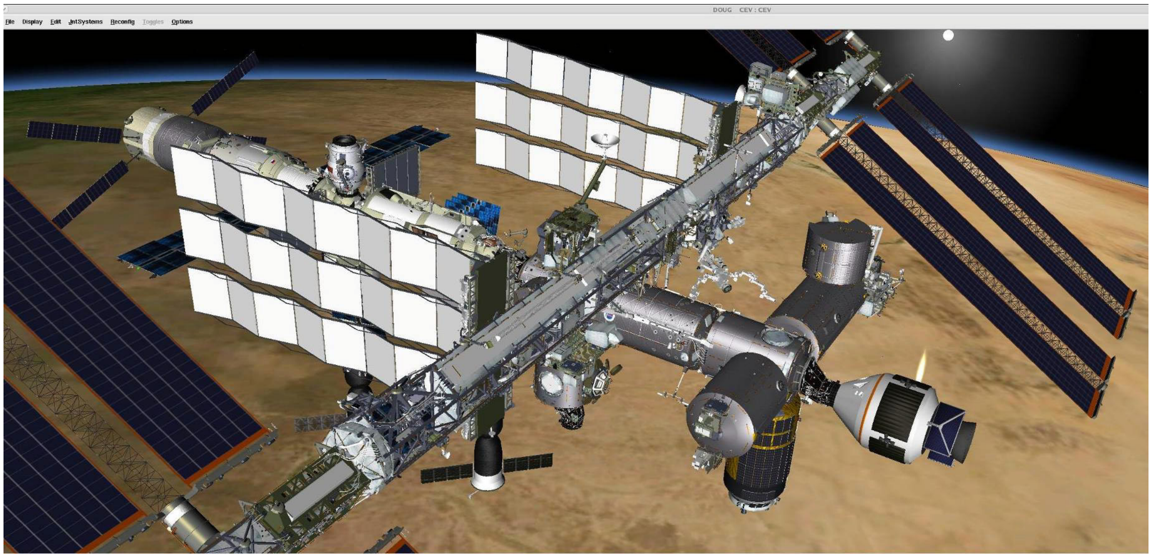


Figure 8. Simulation of CEV docked with the ISS.

The code was compiled under the Trick environment and tested in conjunction with the overall CxTF simulation that consists of multiple simulated systems of the Crew Exploration Vehicle. This part of the mechanical simulations of the CEV it is still under development since it requires specific design data of the launch vehicle provided by the Ares I design group. Once this data is acquired, higher fidelity models can be developed for the hydraulic sub-system of the Thrust Vector Control thus enabling more realistic conditions for the training of all end users.

X. CxTF Cockpit Hand Controllers

As an additional task, software was developed for a set of hand controllers to be used in the long term for the different vehicle trainers. Using standard C libraries under a Linux workstation communication was established between the controllers and the workstation using serial communication standards. The software was developed following portability requirements so it could be used without much manipulation in other systems. Also, adjustments were made to the source code so it could be exported to the Trick simulation environment for future integration in CxTF simulations.

The controllers consist of a Rotational Hand Controller (RHC) and Translational Hand Controller (THC).

- Rotational Hand Controller (RHC)
  - Rotation in 3 axes (Pitch, Roll, Yaw)
  - Trigger Up - Down switch
  - Push Button
  - Toggle Up-Down
  - Slide button
- Translational Hand Controller (THC)
  - Translation in 3 axes (Longitudinal, Vertical, Lateral)

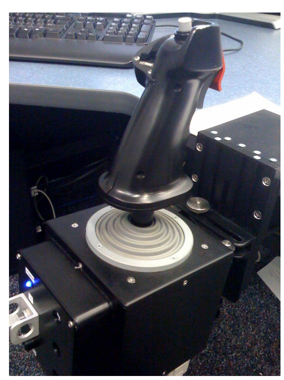


Figure 9. Rotational hand Controller (RHC)

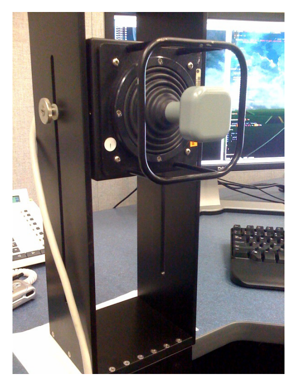


Figure 10. Translational hand Controller (THC)

RHC powers both controllers through a 9 volt power supply and contains a RS-232 serial interface for data transmission. The serial RS-232 port is configured using standard POSIX (Portable Standard for UNIX) terminal control functions and C language libraries.

Principles of Serial Communication

Serial communication refers to the transfer of computer data one bit a time. Typical devices that use serial communications are network devices, keyboards, mice, modems, and terminals. In this case hand controllers were adapted with a serial communications microprocessor to enable data transfer to and from a workstation. During serial communications each word (i.e. byte or character) of data sent or received is transferred one bit a time. It is said that each bit is either *on* or *off*. The speed of the serial data is expressed in bits-per-second ("bps") or baudot rate ("baud"). This represents the number of ones and zeroes that can be sent in one second. At the beginning of the computer age 300 baud was considered the fastest speed, but with advances in technology computers today can handle RS-232 speeds as high as 430,800 baud.

Figure 11. RS-232 Connector

RS-232 is a standard electrical interface for serial communications defined by the Electronic Industries Association. The most commonly used is RS-232C, which defines a mark (on) bit as a voltage between -3V and -12V and a space (off) bit as a voltage between +3V and +12V. The RS-232 standard defines 18 different signals for serial communications. Only six are generally available in the UNIX environment.

GND - Logic Ground
TXD - Transmitted Data
RXD - Received Data
DCD - Data Carrier Detect
DTR - Data Terminal Ready
CTS - Clear To Send
RTS - Request To Send

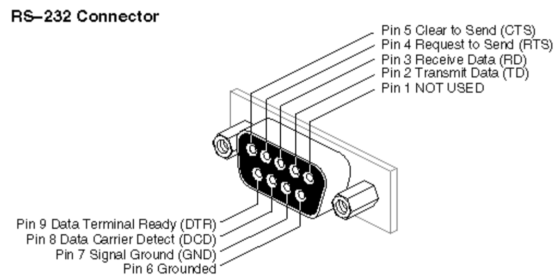


Figure 12. RS-232 Connector showing signal types.

Using the previous concepts source code was developed using the following C libraries to communicate with the hand controllers under the linux environment. Each one of them provides functionalities to the overall script developed for specific actions that have to take place to read data from the serial port.

stdio.h – Standard input/output definitions
string.h – String function definitions
unistd.h – UNIX standard function definitions
fcntl.h – File control definitions
errno.h – Error number definitions
termis.h – POSIX terminal control definitions

Certain steps have to take place in order to read data from the port. In a Linux workstation the serial port can be accessed through a system file where data can be read and written as a normal text file. As a first step the *open()* function is invoked to open file or port to accept read and write commands. If the port can be open this functions returns a positive integer that is used as a tag to identified the port that we are working with. Once the port is open the current settings assigned to the port by the system are read and reconfigured to the specifications of the hand controllers. The port is set for a 9600 baud rate, no parity bit and 8 bit data read support. Also the data is read as raw input meaning that the system will read the data as it comes through the port. Once the serial port is reconfigured an arbitrary byte is written to the port using the *write()* function. This is done as specified by the manufacturer since it is required for the controllers to receive some kind of data before being able to start sending information back to the computer. The data is read using a conditional loop that continuously reads the serial port file for the 9 bytes provided by the controllers and writes them to a file where they will be available for future manipulation.

From the specifications provided by the controller manufacturer it is known that the data values received from the serial port should lie within the ranges shown below in relation to the physical position of the controllers in the different axes. The actual values are read from the serial port file under a Linux machine as character values converted to integer bits that range from 0 to 255. An arbitrary byte needs to be written to the serial port first to then in turn receive these 9 bytes of positional information from the controllers. Notice that bytes 8 and 9 are spare bytes that receive data but are not necessarily related to any axis or switch. These can be programmed by modifying the controller's internal hardware.

BYTE	UNIT	AXIS	VECTOR VALUE	VECTOR VALUE	VECTOR VALUE
BYTE 1	RHC	PITCH	DOWN - 65	CENTER - 128	UP- 189
BYTE 2	RHC	YAW	LEFT - 80	CENTER - 128	RIGHT - 172
BYTE 3	RHC	ROLL	LEFT - 65	CENTER -128	RIGHT - 189
BYTE 4	THC	Y LT-RT	LEFT - 65	CENTER - 128	RIGHT - 189
BYTE 5	THC	Z UP-DN	UP - 65	CENTER - 128	DOWN - 189
BYTE 6	THC	X AFT-FD	AFT - 0	CENTER - 128	FWD - 255
BYTE 7	NONE	SPARE	N/A	FLOAT - 128	N/A
BYTE 8	NONE	SPARE	N/A	FLOAT - 128	N/A
BYTE 9 RHC SWITCHES			TRIGGER DOWN - 1; TRIGGER UP - 2; SLIDE FWD - 4 PUSHBUTTON - 8; TOGGLE UP - 16; TOGGLE DN - 32	N/A	

Table 1. Vector values received from serial port

The source code developed with the above specifications was ported to the Trick simulation environment to be used within simulations. By using common structures that Trick understands parts of the source code were transcribed as source and data files in their respective directories. The conditional loop where the data from the port is continuously read was rewritten as a scheduled job that runs every 0.01 seconds for real time input reading. This is all done in accordance to the standards on which simulations for CxTF are written so it can later be added to full CxTF simulations.

IV. Project Planning and control

Support was provided to Project Planning and Control with the development and maintenance of integrated schedules for both the CxTF project and the Missions Operations Facilities Division (MOFD). The CxTF integrated schedule consisted of important milestones such as

- SDR – Systems Definition Review
- PDR – Preliminary Design Review
- CDR – Critical Design Review
- SAR – Systems Acceptance Review

These Milestones are the driving force between the all the events that need to happen to successfully develop the facility. The schedule also provided information of related dates to these milestones and events that will occur between them. These were updated and maintained by interacting with different sources that provided the necessary information to keep a useful schedule in place.

For the MOFD integrated schedule support was provided by maintaining and updating milestones for the CxTF, Mission Control Center System (MCCS) and Missions Operations Reconfiguration Systems (MORS) projects. The schedule also provided information on integration dates between all the projects or facilities related to MOFD. These are

- OTF (Operations Technology Facility)
- MCCS (Mission Control Center System)
- SSTF (Space Station Training Facility)
- QUAD Integration (CxTF/Cx MCC - ISS MCC/SSTF)

All these facilities will require integration in the future for training purposes. These facilities provide specific products for the training of crew and flight controllers. The Quad integration consists of putting together all MOD facilities necessary to provide realistic training using actual Constellation resources. This means joining the Constellation Training Facility with the Constellation Mission Control Center and the International Space Station Mission Control Center with the Space Station Training Facility and at the same time joining these two sets of facilities into one consolidated simulation environment.

Both integrated schedules required constant interaction with schedulers from different projects. Integrated schedules provide the means to maintain communication between projects of different levels to achieve a better understanding of the events that need to happen for a project or set of projects to be successful.

V. Future work

It is expected that the software developed for the hydraulic system of the TVC for the Ares I simulations is verified and rewritten as necessary to include all recent changes to the design of the real system. The written code is to be integrated to the overall CxTF simulation taking into consideration other systems that are involved with the Thrust Vector Control system.

As part of future developments the feedback from the controllers will be read and manipulated to any extent necessary to convert it into useful parameters during simulations. The code developed under the Trick simulation environment needs to be further improved in terms of how the serial data is read. That is, to make data read more reliable and test it for periods of long duration. This should be added later on added to the CxTF simulation to enable interfaces between the required systems and subsystems and be deployed as needed.

The integrated schedules are to be maintained and observed for future changes in milestones across the Constellation Training Facility and Missions Operations Facilities Division.

VI. Conclusion

All the supported tasks during the 15 week internship at Johnson Space Center in Houston, Texas were completed satisfactorily. Knowledge and skills were developed in many areas including, C/C++, Trick, Thrust Vector Control, Serial Communications Programming, and Project Planning and Scheduling. All contributions made to the Constellation Training Facility will have impacts in current and future events related to the Constellation program overall. NASA will face new challenges in the future with the Constellation Program that will require knowledge and experience in all aerospace related fields. The Undergraduate Student Research Program provides the means to prepare future generations for these new challenges by providing opportunities like the CxTF support provided during this internship.

Acknowledgments

Thanks to the Undergraduate Student Research Program and the Universities Space Research association for providing the internship funding and opportunity. Thanks to the National Aeronautics and Space Administration specifically to the Johnson Space Center Missions Operations Facilities Division for the opportunity of working and collaborating with their current projects. Special thanks to my mentor Shashi Gowda for providing the time to guide and teach me all that was required to have a successful internship. Finally, thanks to all of those that provided support and mentorship on the different tasks that were supported in particular to Brandon Lloyd, Toby Martin, Rachel Obeidzinsky, Veronica Seyl, and Robert Paul.

References

Schild, S., *C++: The Complete Rerence*, 4th ed., McGraw-Hill, 2002.

Sweet, M., "Serial Programming Guide for POSIX operating systems," 5th ed. 2005. URL: http://www.easysw.com/~mike/serial/serial.html#3_1_3

"Constellation Program: America's Fleet of Next-Generation Launch Vehicles," NASA Facts, George C. Marshall Space Flight Center, National Aeronautics and Space Administration, 2008. URL: http://www.nasa.gov/mission_pages/constellation/ares/aresl/index.html

"NASA's Ares I Upper Stage: Powering the Second Phase of a Rocket's Journey to Space," *NASA Facts,* George C. Marshall Space Flight Center, National Aeronautics and Space Administration, 2008. URL: http://www.nasa.gov/mission_pages/constellation/ares/aresl/index.html

"The J-2X Engine: Powering NASA's Ares I Upper Stage and Ares V Earth Departure Stage," NASA Facts, George C. Marshall Space Flight Center, National Aeronautics and Space Administration, 2008. URL: http://www.nasa.gov/mission_pages/constellation/ares/aresl/index.html

Ares I and Ares I Thrust Vector Control System

http://www.nasa.gov/mission_pages/constellation/ares/aresI/index.html

http://www.nasa.gov/pdf/187391main_aresI_fs_nov08.pdf

http://www.nasa.gov/pdf/230922main_1stStage_FS.pdf

http://www.nasa.gov/pdf/231430main_UpperStage_FS_final.pdf

http://www.nasa.gov/pdf/187393main_j2x_fs_nov08.pdf

http://spaceflightsystems.grc.nasa.gov/LaunchSystems/UpperStage/Thrust/

http://www.nasa.gov/pdf/293938main_Ares_I_V_Expanded_nov08.pdf

Serial Programming (External to NASA)

http://www.easysw.com/~mike/serial/serial.html

Constellation Training Facility Support

Jose M. Flores
University of Puerto Rico
November 20, 2008

Agenda

- Personal Background
- USRP Overview
- CxTF Overview
- Contributions to CxTF Project
  - **Software Development**
    - Ares I Thrust Vector Control Hydraulics
    - CxTF Cockpit Hand controllers
  - **Project Planning and Control**
- Acknowledgments

Personal Background

- **Education**
  - **University of Puerto Rico**
    - Bachelor of Science in Mechanical Engineering
    - Senior year. Graduating December 2009
- **Undergraduate Research**
  - **Undergraduate Research on internal channel flow with added roughness**
    - Pratt & Whitney sponsored project
    - Internal channel roughness analysis to improve turbine efficiency
- **Professional**
  - **NASA – LaRC Intelligent Machines and Robotics Laboratory (D203)**
    - Robotic arm development for the Platform for Science Instruments (PSI)
  - **Formula SAE - Colegio Racing Engineering**
    - Powertrain support

Constellation Training Facility

- The Constellation Training Facility will provide a venue for training
  - Crew members
  - Flight controllers
  - Instructors
  - Other identified personnel (procedure verification engineers, mission analysis personnel, etc)

- The facility includes training devices and supporting infrastructure for the simulation of mission operation tasks

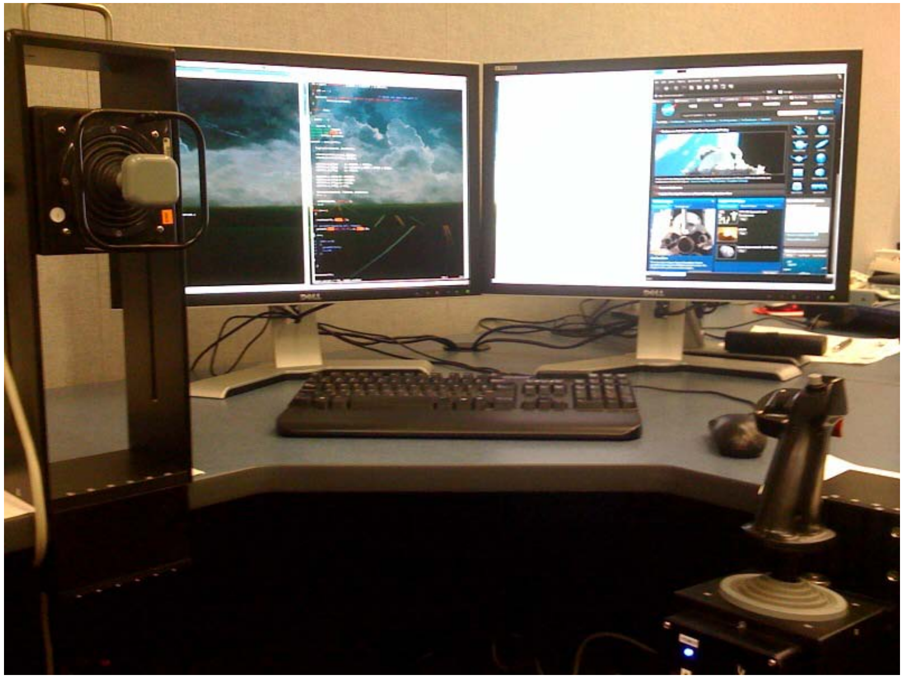
- CxTF will support training for the following:
  - ISS (Lower Earth Operations) missions
  - Lunar missions

Constellation Training Facility Overview

- **Trainers**
  - **Full Task Trainer (FTT)**
    - Orion Mission Simulator (OMS) – Crew Station
  - **Part Task Trainer (PTT)**
    - Orion Part-task Trainer (OPT)
    - Flight Controller Part-task Trainer (FCPT)
  - **Desktop Trainer**
    - Provides training simulation access from office environment
  - **Onboard Trainer**
    - Laptop based simulation suitable for onboard training

Software Development Support

- **Software development support using Trick**
  - **Simulation of the hydraulic system of the Thrust Vector Control (TVC) for the upper stage of the Ares I Crew Launch Vehicle**
  - **Serial communication for hand controllers to be used in CxTF simulations**

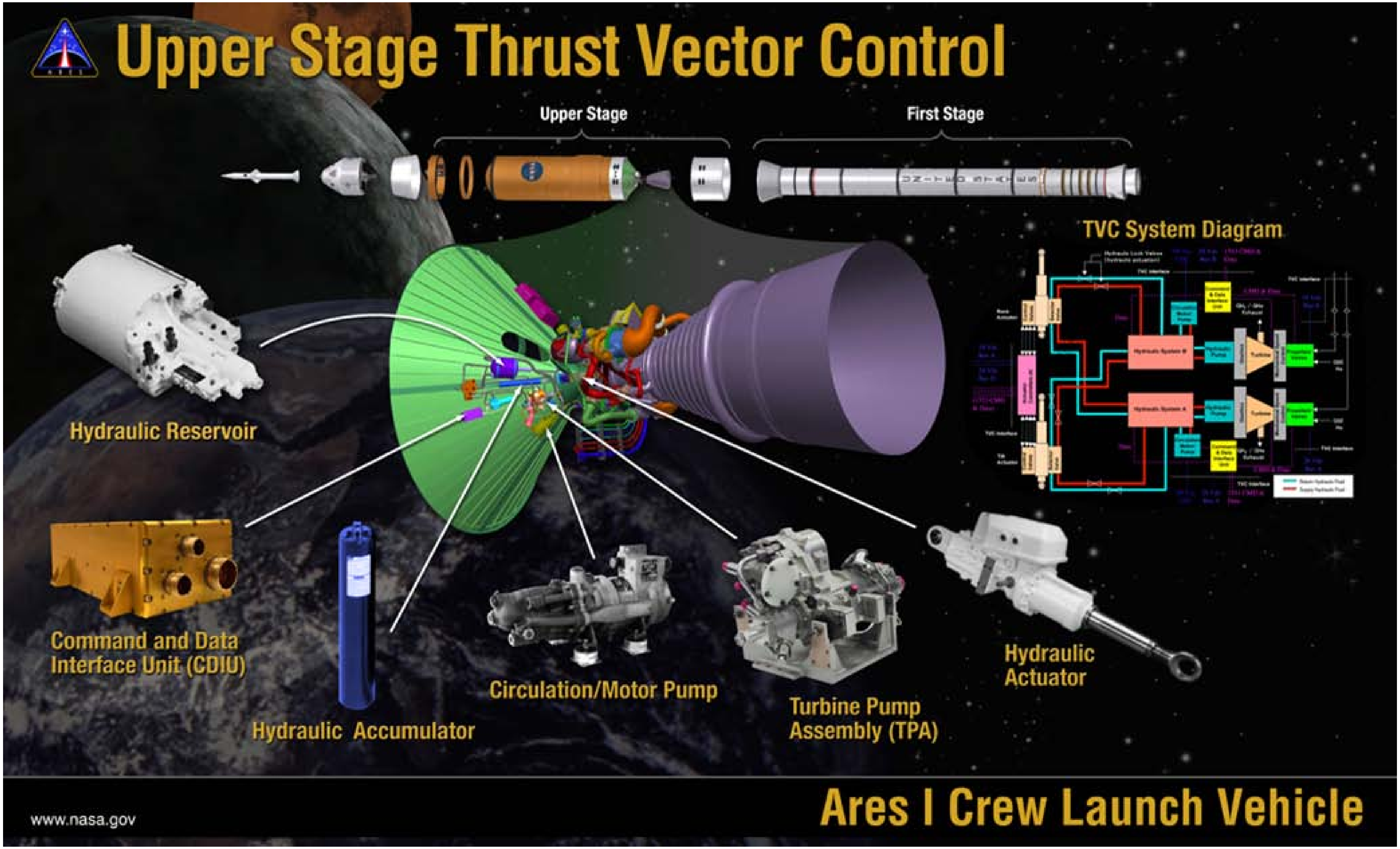


Software Development - Trick

Trick Simulation Environment

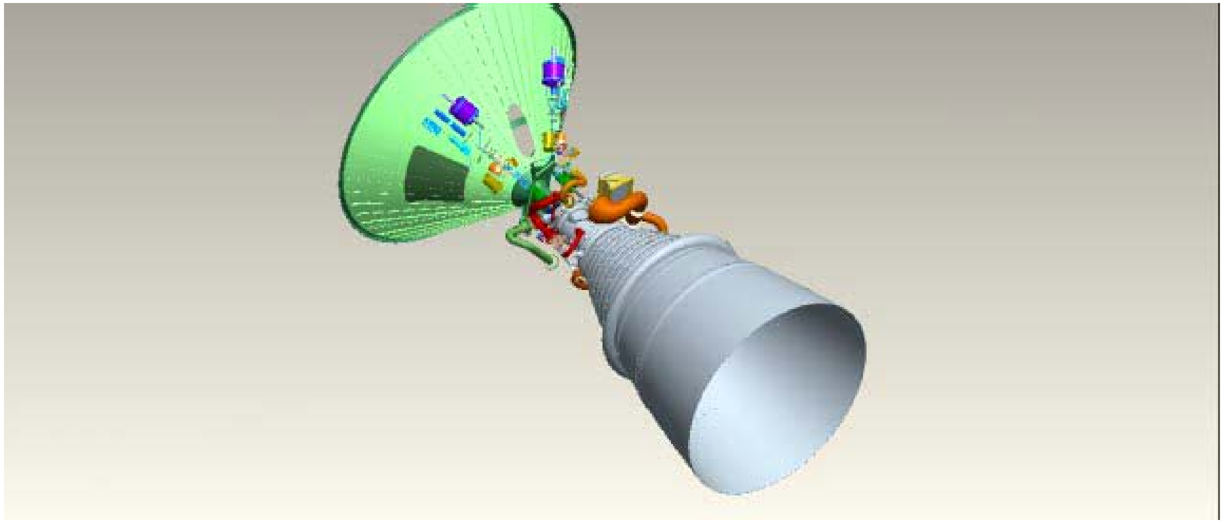
- Generic simulation toolkit that is used extensively for training astronauts through real-time interactive simulations and for engineering analysis
- Run-time executive designed for both real-time and non-real-time applications
  - Time-based scheduling
  - Event-based scheduling
- The run-time executive provides a simple interface that allows the user to configure
  - Model parameters
  - Integration schemes
  - Real-time controls

Ares I Launch Vehicle



Ares I Thrust Vector Control

- The simulation software developed is in charge of monitoring the state of the hydraulic system
  - Pressure
  - Hydraulic fluid level
  - Hydraulic System being used
- The software provides malfunction flags that will be enabled as required by training to simulate malfunctions

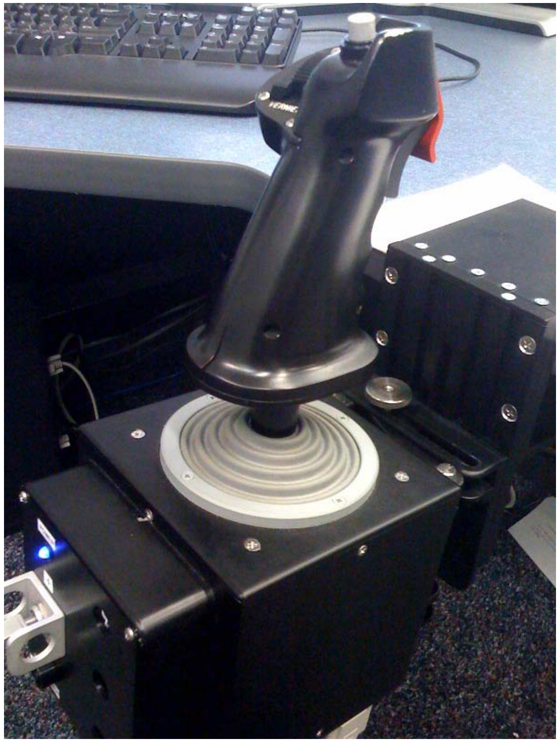


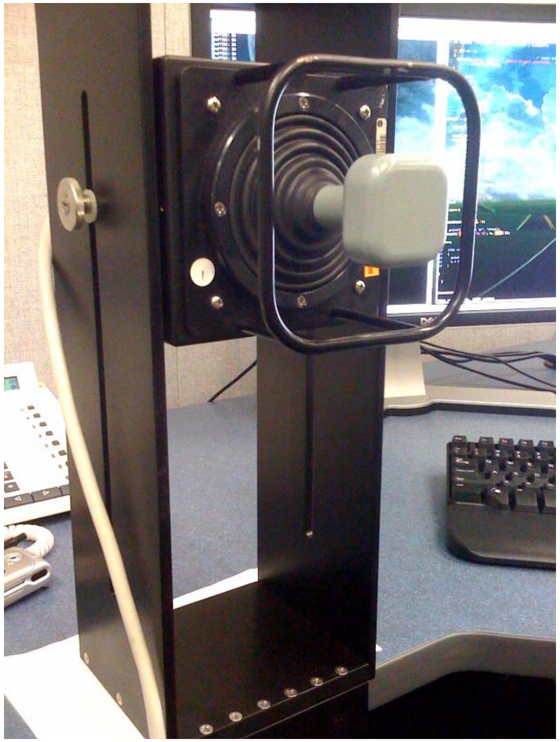
CxTF Cockpit Hand controllers

- **Hand Controllers**
  - **Rotational Hand Controller (RHC)**
    - Rotation in 3 axes (Pitch, Roll, Yaw)
    - Trigger Up - Down switch
    - Push Button
    - Toggle Up-Down
    - Slide button
  - **Translational Hand Controller (THC)**
    - Translation in 3 axes (Longitudinal, Vertical, Lateral)
- **RS-232 Serial connection**

CxTF Cockpit Hand controllers

- RS-232 serial interface for data transmission on RHC
- The serial RS-232 port is configured using standard POSIX (Portable Standard for UNIX)





CxTF Cockpit Hand controllers

- **Reading data from the serial port**
  - ***open()* function is called to open port to accept read and write commands**
  - **Current port settings are read and reconfigured to the specifications of the hand controllers**
    - 9600 baud rate,
    - No parity bit
    - 1 stop bit
    - Raw input
  - **Arbitrary byte is written to the port using the *write()* function**
  - **Conditional loop to read the port for 9 bytes of data continuously**

Project Planning and Control

- Updated and Maintained CxTF integrated schedule
  - CxTF Milestones (SDR, PDR, CDR, SAR, etc.)
- Updated and Maintained MOFD (DD) integrated schedule
  - Milestones for CxTF, MCCS and MORS
  - Integration Dates Between Projects
    - OTF (Operations Technology Facility)
    - MCCS (Mission Control Center System)
    - SSTF (Space Station Training Facility)
    - Quad Integration (CxTF/Cx MCC – ISS MCC/SSTF)

Acquired Knowledge and Skills

- C/C++
- Trick Training
- Thrust Vector Control
- Serial Communications Programming
- Project Planning and Scheduling

Contact Information

- Jose M. Flores

DD33

281 483 4848 (B30A/1042)

281 244 8147 (B16/179)

787 220 8689 (mobile)

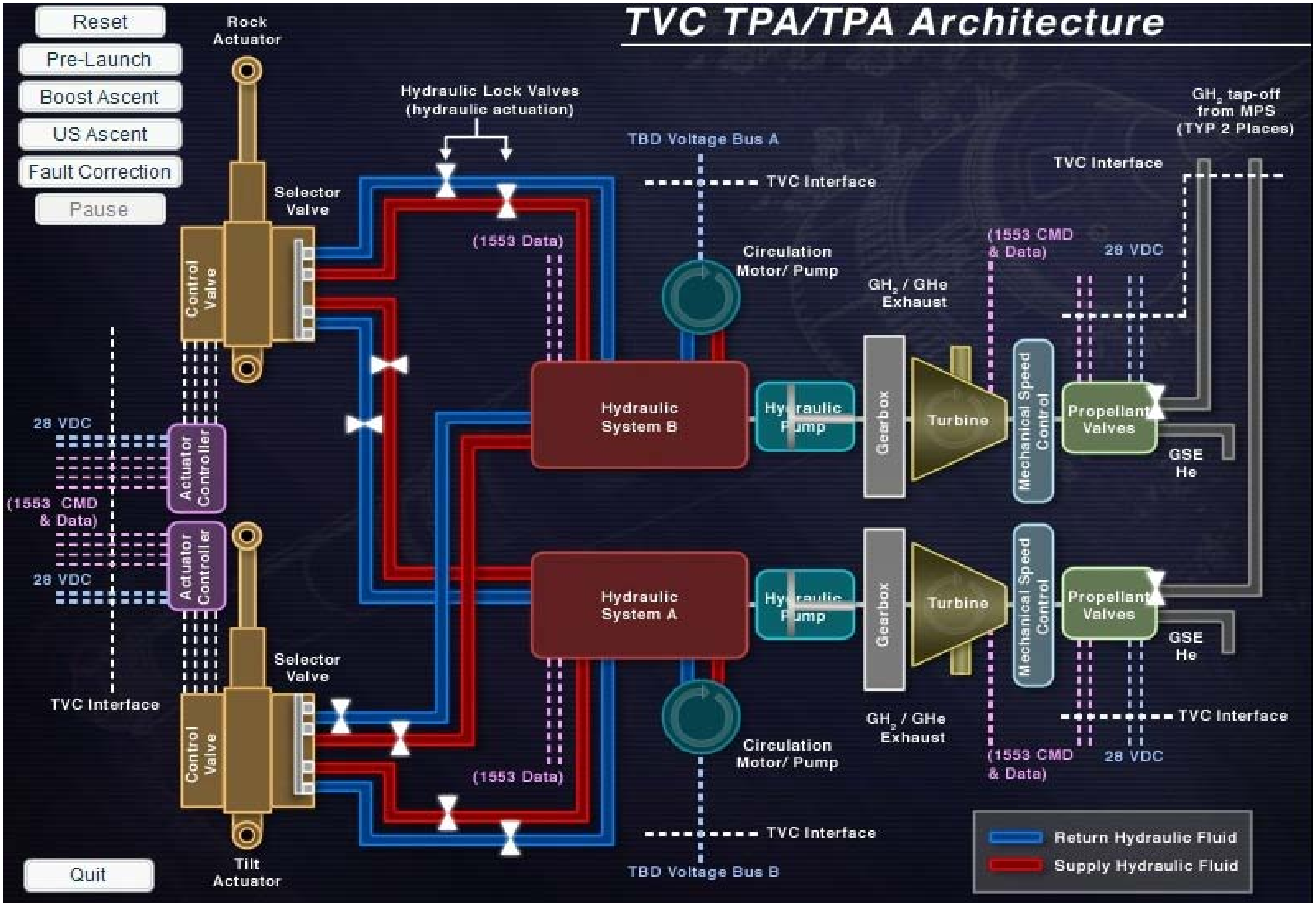
jose.m.flores-1@nasa.gov

jose.marcos.flores@gmail.com

Backup Slides

Upper Stage Thrust Vector Control

- **How the TVC hydraulics works**



CxTF Cockpit Hand controllers

- **Vector values received from the controllers**

BYTE	UNIT	AXIS	VECTOR VALUE	VECTOR VALUE	VECTOR VALUE
BYTE 1	RHC	PITCH	DOWN - 65	CENTER - 128	UP- 189
BYTE 2	RHC	YAW	LEFT - 80	CENTER - 128	RIGHT - 172
BYTE 3	RHC	ROLL	LEFT - 65	CENTER -128	RIGHT - 189
BYTE 4	THC	Y LT-RT	LEFT - 65	CENTER - 128	RIGHT - 189
BYTE 5	THC	Z UP-DN	UP - 65	CENTER - 128	DOWN - 189
BYTE 6	THC	X AFT-FD	AFT - 0	CENTER - 128	FWD - 255
BYTE 7	NONE	SPARE	N/A	FLOAT - 128	N/A
BYTE 8	NONE	SPARE	N/A	FLOAT - 128	N/A
BYTE 9 RHC SWITCHES			TRIGGER DOWN - 1; TRIGGER UP - 2; SLIDE FWD - 4 PUSHBUTTON - 8; TOGGLE UP - 16; TOGGLE DN - 32	N/A	