DTVC Plan | Emulation of Aerospace Actuation Systems

A Real-Time Controller Hardware-in-the-Loop Platform for Emulation of Aerospace Actuation Systems

Senior Design Students

Kori Eliaz | Electrical Engineering
Dylan Gaub | Computer Engineering
Jake Dorsett | Electrical Engineering

Additional Team Members

Ethan Schultz | Electrical Engineering, Undergraduate Team Member
Claudio Lima | Electrical Engineering, Graduate Team Member

Advisors

Project Advisor: Dr. James Cale
Lab Advisor: Chris Lute
EiR/Woodward Advisor: Matt Heath

Customers

Industry: Woodward, Inc.
Academia: CSU Systems Engineering Department

Summary

The Emulation of Aerospace Actuation Systems Project is part of an effort to validate a controller hardware-in-the-loop (CHIL) laboratory capability for use by CSU students and faculty. One application of this project is to validate an electrical drive alternative being developed by Woodward, Inc. to replace the pneumatically/hydraulically operated Thrust Reverser Actuation Systems (TRAS) currently deployed on commercial and military aircraft. This project will interface a TI TMS320F28379D microcontroller with an OPAL-RT real-time processor to emulate several electrically driven aerospace actuation scenarios using a CHIL platform. Team members will program both the TI board and the OPAL-RT system in C to read/write synchronized digital and analog signals that reflect the behavior of real mechanical motors and drive systems located in a physical testbed at the Aerospace System Emulation and Test (ASET) Lab at the CSU Powerhouse. In addition, the team will produce a detailed system model of the code and associated testbed hardware using the principles of Model Based Systems Engineering in Cameo Systems Modeler. The overall product is a well-documented system model and thoroughly tested code package that can be used by CSU students and faculty for CHIL demonstration and verification purposes on current projects and in the future.

Why is This Project Important?

The trend towards More Electric Aircraft (MEA) is expected to revolutionize the aerospace industry in the coming years, with promises of reduced weight, maintenance, and additional data analytics for real-time system prognostics compared to traditional drive systems. To verify these claims without incurring tremendous R&D costs to the customer, a CHIL platform allows for emulation of an expensive physical system using inexpensive software to test the behavior and benefits of an electromechanical TRAS.

Date	Comments	Version	Approved By
10/25/2021	Initial release	1.0	Jim Cale, Matt Heath

Objectives

Primary Objectives:

Note: These objectives are directly required by the customer for the successful completion of this project:

- Determine the analog and digital I/O requirements for the microcontroller and OPAL-RT
- Develop C code for running a closed-loop CHIL experiment, where the OPAL-RT executes a real-time DC machine system model
- Run the closed-loop CHIL experiment with set failure modes as described in the proprietary document and collect measurement information
- Document the design steps and experimental results in a test report to be delivered to the customer
- Include system requirements flow-through and design information on the CHIL platform in the existing Cameo repository for the testbed

Secondary Objectives:

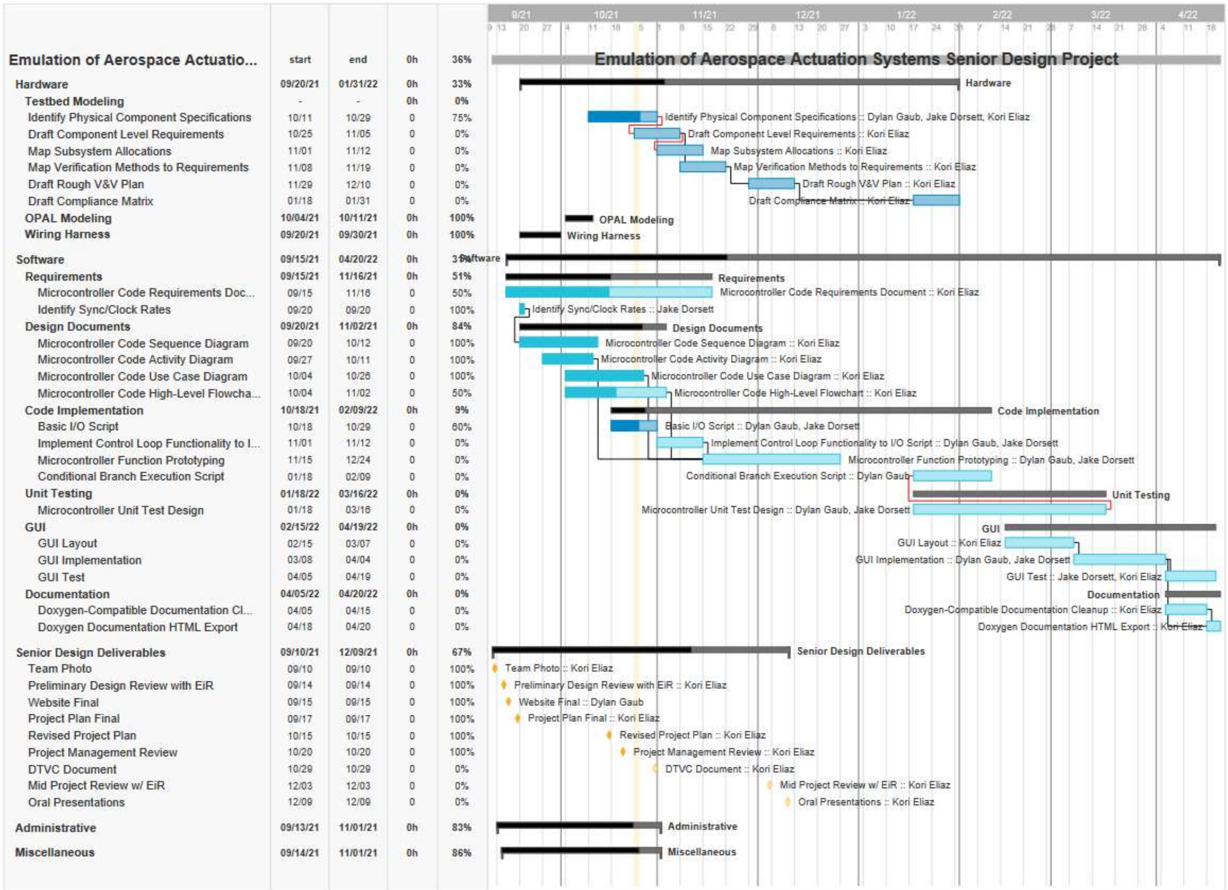
Note: these objectives are not directly required by the customer, but are being pursued by the senior design team to increase readability and overall functionality of the end product:

- Design a Graphical User Interface (GUI) that abides by the ISA101 HMI Standard to represent the current state of the testbed during all tests as well as error codes that are flagged during these tests and the system's real-time response to mitigate these errors
- Create a script to parse through the microcontroller code and establish code coverage of all conditional branches to ensure that all code is being thoroughly tested and verified before official hand-off to the customer for interface with the physical testbed
- Run a documentation tool (e.g. Doxygen) on the finished code to extract an HTML report of all code documentation and increase readability/ease of understanding for the customer upon official hand-off at the end of the project period

Design Requirements

- Inputs to the system (OPAL-RT model to microcontroller):
    - Control Level (duty cycle percentage from 0-100%)
    - Load Torque (-0.2 Nm to 0.2 Nm +/- 3%)
- Outputs from the system (microcontroller to OPAL-RT model):
    - Motor Speed (-600 to 600 rad/s +/- 3%)
    - Motor Armature Current (-2.5 A to 2.6 A +/- 3%)
- Microcontroller must receive and process analog and digital signals strictly within 0-3.3V @ -20 to 20 mA
- Microcontroller must process commands at an absolute maximum rate of 20 us
- GUI must adhere to ISA101 HMI Standards that allows for manual control of a model loaded onto the OPAL-RT. Currently, this is an H-bridge design with 4 DC machines.

Project Timeline



General Test Plan

This system requires testing of both hardware (TI microcontroller and OPAL-RT machine) and software (microcontroller code) components. To accomplish this, a set of hard-coded inputs will be passed to the TI microcontroller to simulate test conditions iterated in the following pages. These inputs will trigger specific hardware outputs in the form of a voltage range that is representative of a certain quantity (load torque, duty cycle ratio, etc.) that are passed via connector to the OPAL-RT machine and cause an appropriate response within a constant Simulink model. This model will generate the appropriate response to represent the behavior of the emulated testbed in the form of a motor speed and armature current to be passed back via connector to the TI microcontroller and read in as input.

Specific targets to be tested within this system include:

- Voltage ranges
- Effects of changing control level (duty cycle)
- Effect of changing load torque
- Effect on simulated motor speed
- Effect on simulated armature current
- Effect on simulated motor temperature

Hardware Tests

Test 1.1	TI GPIO Connections
Components	TI Microcontroller, TI dev board, jumper cables, function generator
Constraints	- Microcontroller can receive 0-3.3V (-0.3V absolute min to 4.6 V absolute max @ -20 to 20 mA) - Microcontroller can output 0-3V (+/- 0.3%)
Stimulus	Sinusoidal waveform (1V amplitude, 1V DC bias @ 1 kHz) generated by function generator
Expected Response	TI sees the input from the Analog Discovery and transmits a shifted output (by 1V) visible on an oscilloscope.
Test Plan	1. Generate the 1V @ 1kHz sinusoidal waveform using the function generator. 2. Connect the voltage output from the function generator to the TI Dev Board on pins 9 (ADC A0) and 11 (ADC A1). 3. Compile and load Actuation_CPU1.c file onto TI microcontroller to take in input from pins 9 and 11, shift by 1V constant value, and output using pins 15 (ADC A2) and 17 (ADC A3). 4. Connect the original input from pins 9 and 11, and output from pins 15 and 17 to an oscilloscope and confirm that the output is shifted by 1V.

Test 1.2	Open-Loop Control Level (within bounds)
Components	TI Microcontroller, OPAL-RT, TI dev board, jumper cables, DB37 breakout boards, BNC-miniBNC connector, oscilloscope
Constraints	- Microcontroller can receive 0-3.3V (-0.3 absolute min to 4.6 V absolute max @ -20 to 20 mA) - Microcontroller can output 0-3V (+/- 0.3%) - Output value limited to 12 bits - Control level (duty cycle) range limited to 0-100% (mapped to 0-3 V) - Values above 3V will saturate at 100% duty cycle (cannot exceed 100%)
Stimulus	Hexadecimal value representing a control level percentage, controlled by the user via a slider scale in the system GUI or hard-coded in the microcontroller code.
Expected Response	Example: if the user enters a 50% control level, the voltage transmitted to the OPAL should be 1.5V (+/- 0.3%).
Test Plan	1. Assign a constant control level value of 50% (hard-coded or via GUI). 2. Convert 50% to a 12 bit hexadecimal value (0x7FF). 3. Transmit hexadecimal value via pins 9 and 11 (ADC A0 and A1). 4. Connect jumpers at pins 9 and 11 on the microcontroller to CH00, Group 1B on the OPAL-RT via the DB37 breakout board. 5. Connect Group 1B scope output to an oscilloscope using the BNC-miniBNC connector. 6. Confirm that the output corresponds to test input (following test case in expected response described above).

Test 1.3	Open-Loop Load Torque (within bounds)
Components	TI Microcontroller, OPAL-RT, TI dev board, jumper cables, DB37 breakout boards, BNC-miniBNC connector, oscilloscope
Constraints	- Microcontroller can receive 0-3.3V (-0.3V absolute min to 4.6 V absolute max @ -20 to 20 mA) - Microcontroller can output 0-3V (+/- 0.3%) - Output value limited to 12 bits - Load torque range limited to -0.2 to 0.2 Nm (mapped to 0-3V) - Values above 3V will NOT saturate (will exceed maximum load torque)
Stimulus	Hexadecimal value representing a load torque, controlled by the user via a slider scale in the system GUI or hard-coded in the microcontroller code.
Expected Response	Example: if the user enters 0 Nm (no load torque), the voltage transmitted to the OPAL should be 1.5V (+/- 0.3%).
Test Plan	1. Assign a constant load torque value x (hard-coded or via GUI). 2. Convert x to a 12 bit hexadecimal value. 3. Transmit hexadecimal value via pins 9 and 11 (ADC A0 and A1). 4. Connect jumpers at pins 9 and 11 on the microcontroller to CH01, Group 1B on the OPAL-RT via the DB37 breakout board. 5. Connect Group 1B scope output to an oscilloscope using the BNC-miniBNC connector. 6. Confirm that the output corresponds to test input (following test case in expected response described above).

Note: There are many more hardware tests to be done that will be iterated as the project develops and interfaces are better understood. Notably, the out of bounds versions of tests 1.2 and 1.3, as well as effects of changing control level and load torques on motor speed and armature current.

Software Tests

Test 2.1	Code Coverage
Components	TI Microcontroller
Constraints	- All conditional branches must be flagged for both true and false for each specified parameter.
Stimulus	Comprehensive list of inputs to trigger each conditional branch for both T/F cases.
Expected Response	100% code coverage (each conditional branch traversed twice per parameter).
Test Plan	Follow the procedure as outlined in https://dev.ti.com/tirex/explore/node?node=AFqrv9i9.g9IGxpPpTmHiw__FUz-xrs__LATEST&search=code%20coverage.

Test 2.1	Unit Tests – Test_GetMotorSpeed(), Test_GetArmatureCurrent()
Components	TI Microcontroller
Constraints	- Motor speed limited to -600 to 600 rad/s (can exceed bounds, needs error checking) - Motor armature current limited to -2.5 to 2.5 A (can exceed bounds, needs error checking) - Control level limited to 0-100% (cannot exceed bounds) - Load torque limited to -0.2 to 0.2 Nm (can exceed bounds, needs error checking)
Stimulus	Load torques that cause motor speed and armature current to go negative, positive, and out of bounds.
Expected Response	Negative load torque: motor speed increases, armature current decreases. Positive load torque: motor speed decreases, armature current increases. No load torque: motor speed and current constant.
Test Plan	1. Unit test functions Test_GetMotorSpeed() and Test_GetArmatureCurrent() take in a control level as a percentage (double) and a load torque (double). Range of values passed in (can use linspace if creating unit tests in MATLAB). 2. Unit test functions shall process this range of values and produce a corresponding output plot for verification by inspection. 3. Unit test functions shall have error checking clauses implemented to detect the following cases: a. Motor speed exceeds bounds (throw error code for motor failure) b. Armature current exceeds bounds (throw error code for danger of overheating) 4. Unit test functions shall return a single integer value representing successful completion of the test for tracking purposes.

Note: There are more software tests to be added that will be iterated as the project develops and interfaces are better understood. Notably, additional unit tests to verify that any other system parameters are being tested in and out of bounds.