



Ames Research Center



# Software Development Infrastructure for Small Spacecraft

Flight Software Workshop 2007

Howard Cannon, Craig Pires

11/5/2007



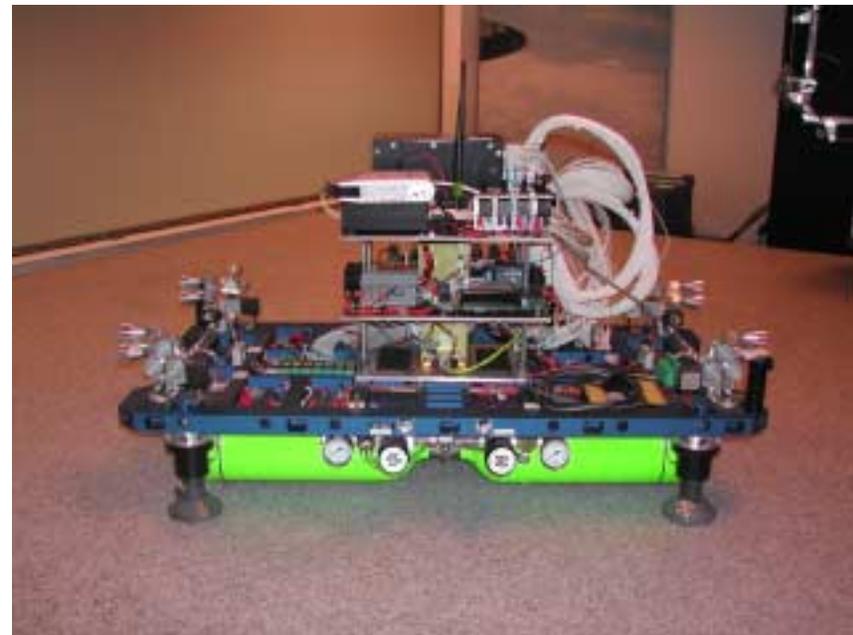
## Overview

### GOAL:

Develop infrastructure and processes for small spacecraft software development using automatic code generation techniques.

### 2007 Objectives:

- Flatsat – Small cart on flat granite table. Develop approach/processes and conduct tool trades using simple example.
- 6 DOF Integration Test – Demonstrate 6DOF control on small spacecraft bus test platform with cold gas thrusters.
- Cruise Phase Simulation – Demonstrate “lost-in-space” algorithm and trajectory correction maneuvers in simulation.

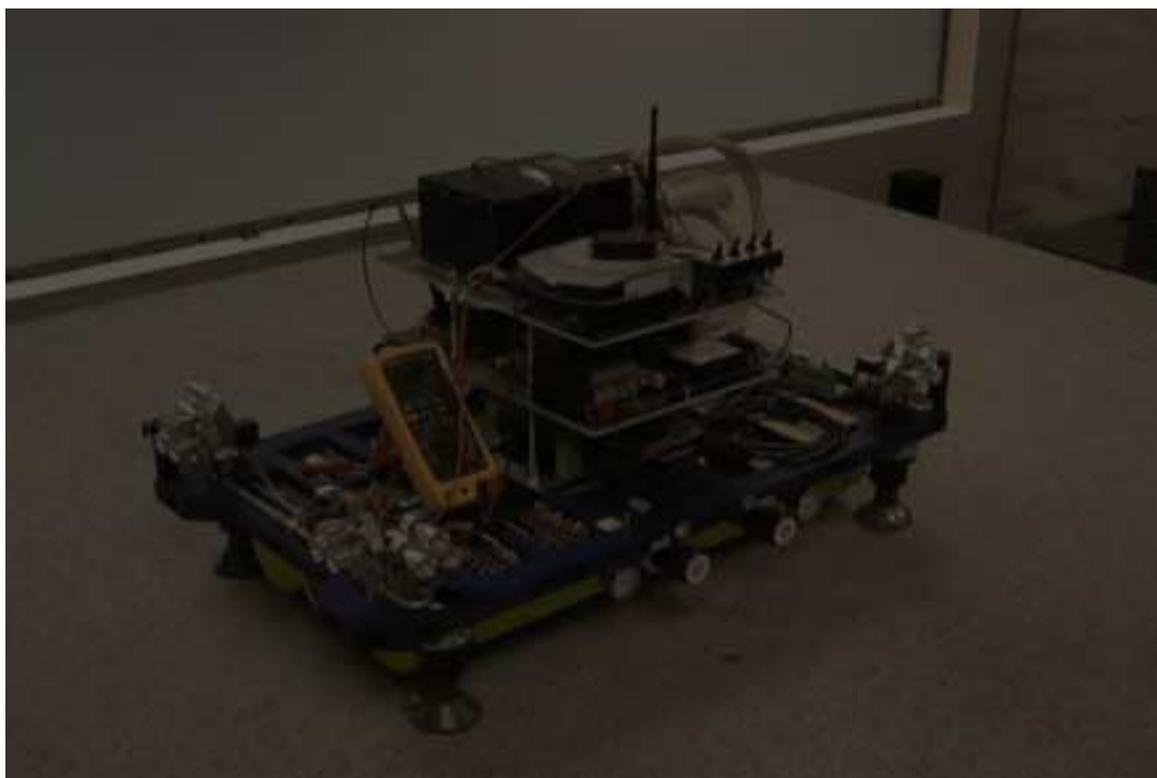




Ames Research Center



## FLATSAT TESTBED





Ames Research Center

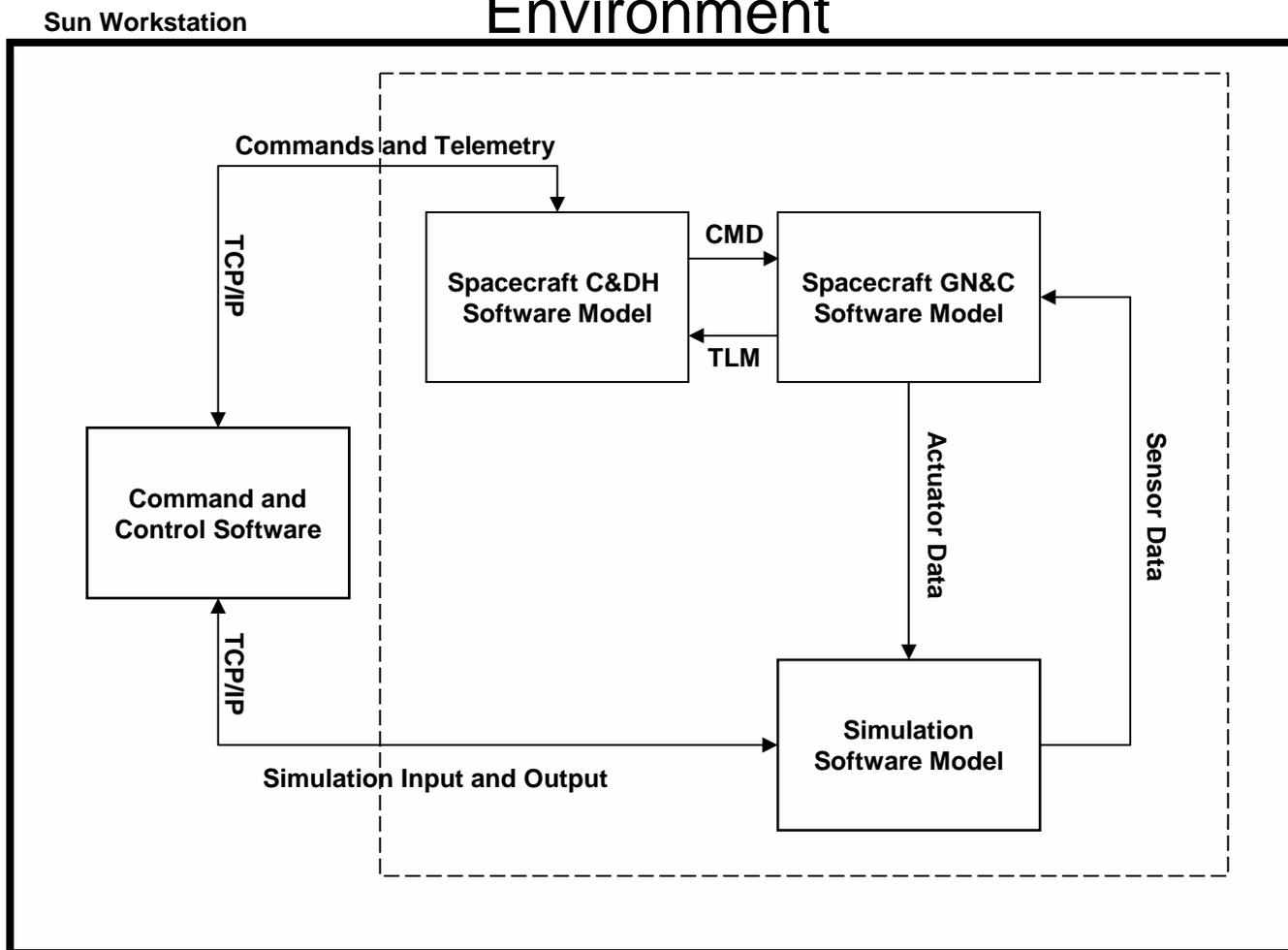


## Common Spacecraft Bus Pressure Test



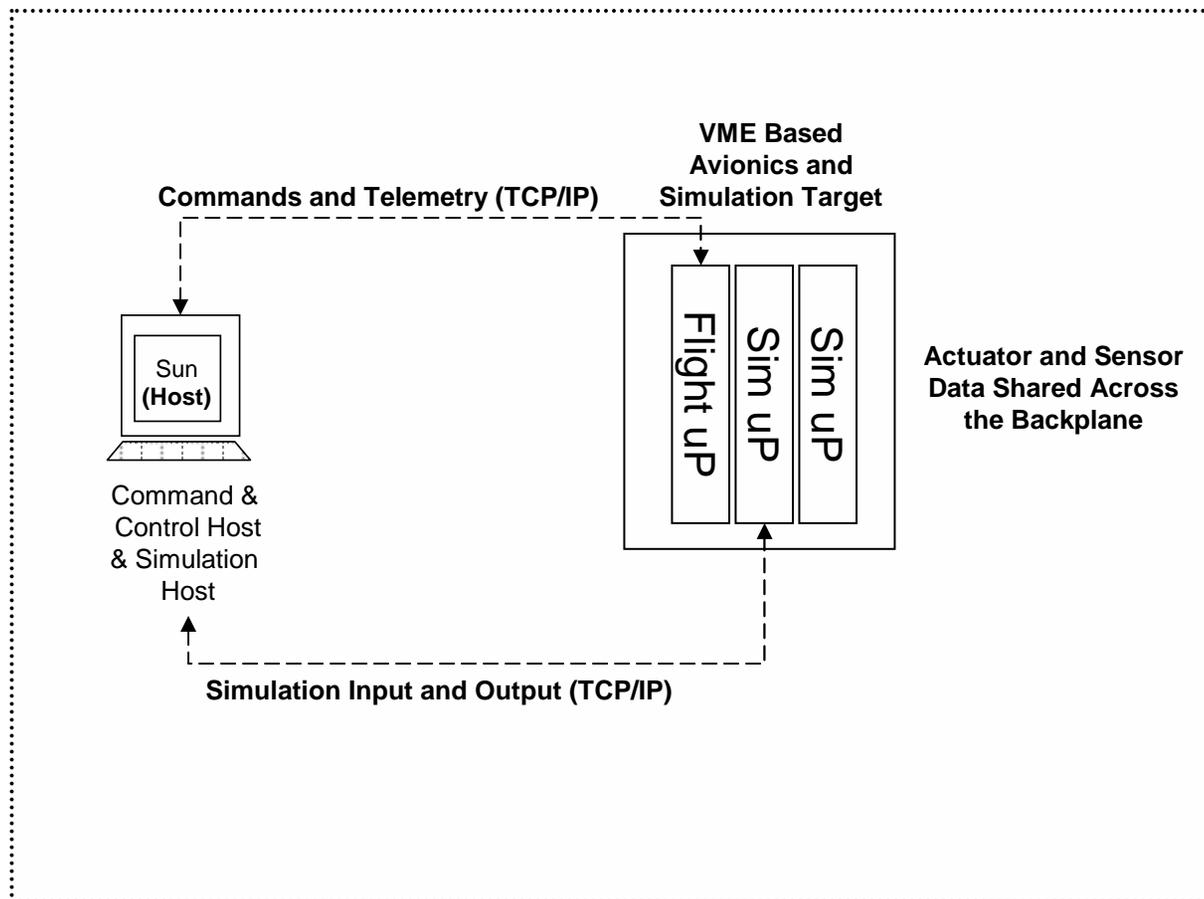


## FSW Development: Workstation Simulation (WSIM) Environment



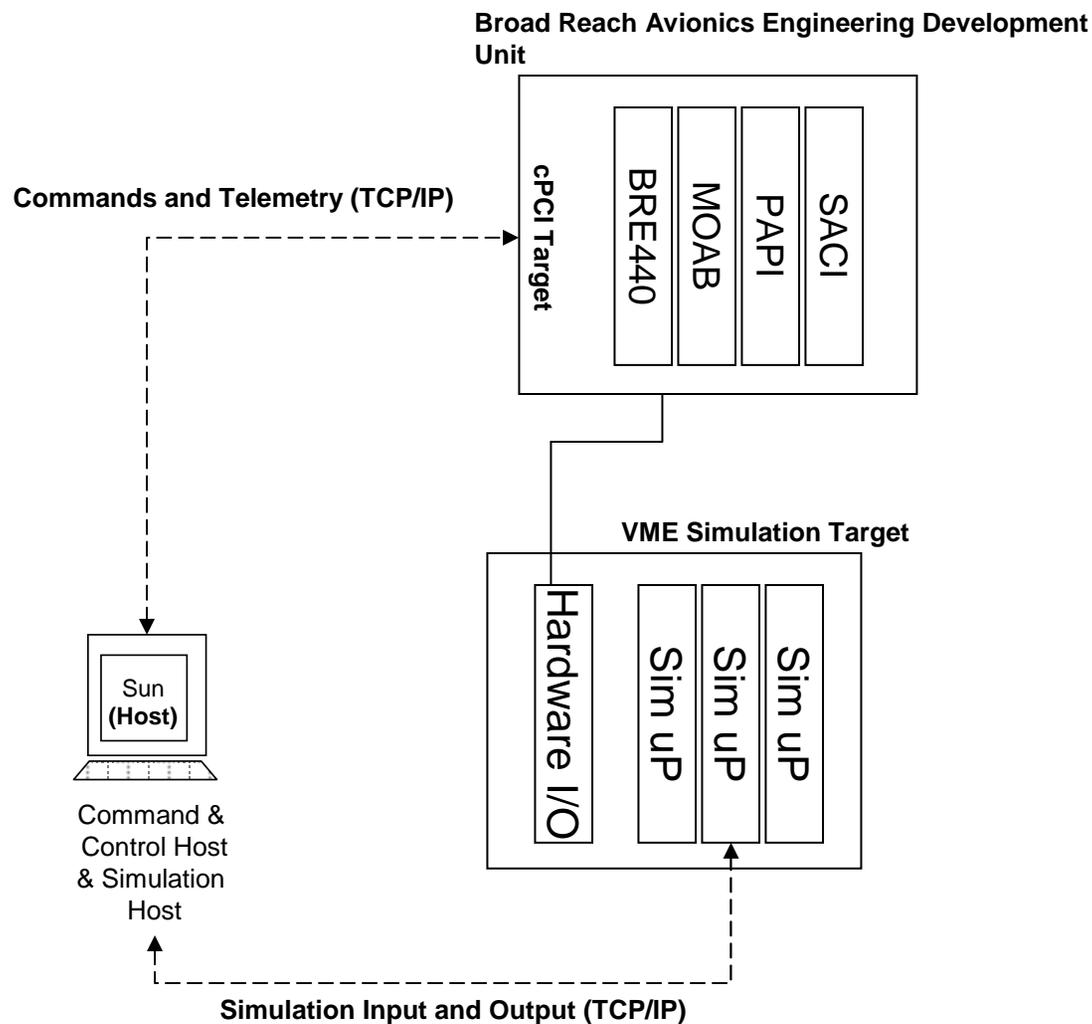


## FSW Development: Processor-in-the-Loop





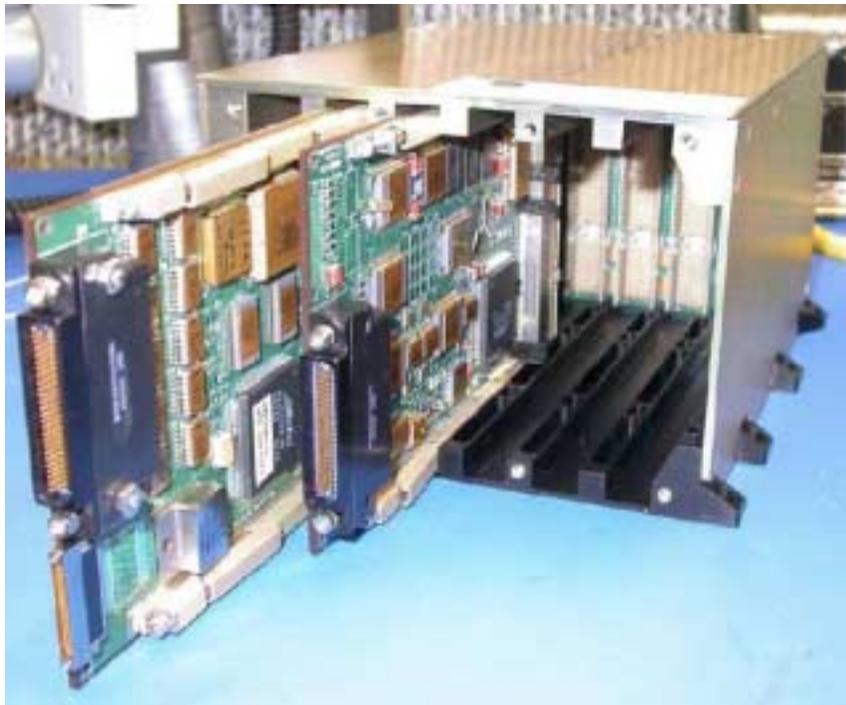
# FSW Development: Hardware-in-the-Loop





# Broad Reach Engineering Development Unit

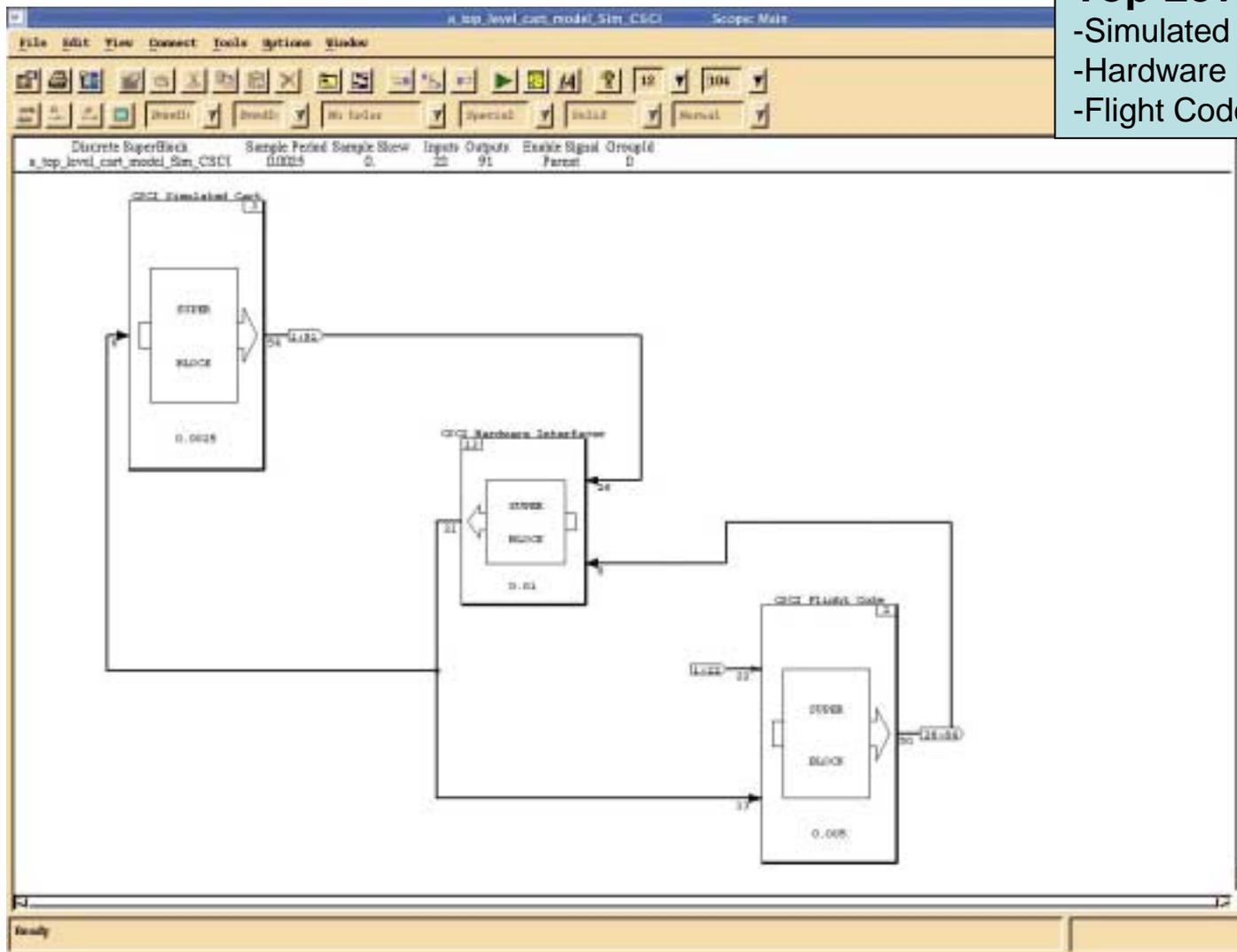
- 8 Slot (5 C&DH, 3 Power) 3U cPCI Chassis
- BRE Starter BRE440
  - 128 Mbyte SDRAM, 8 Mbytes Boot-ROM
  - 200 MHz/400 Mips
- MOAB IO Board
  - 47 AD590 Temp channels
  - 12 Sun sensor channels
  - 24 Analog channels
  - 40 RS422 /LVDS transmitters and receivers
  - 48 Discrete Inputs and Outputs
  - MIL-STD-1553
- Solar Array Control Integration (SACI) board.
- Power Switching and Pyro Integration (PAPI) board.





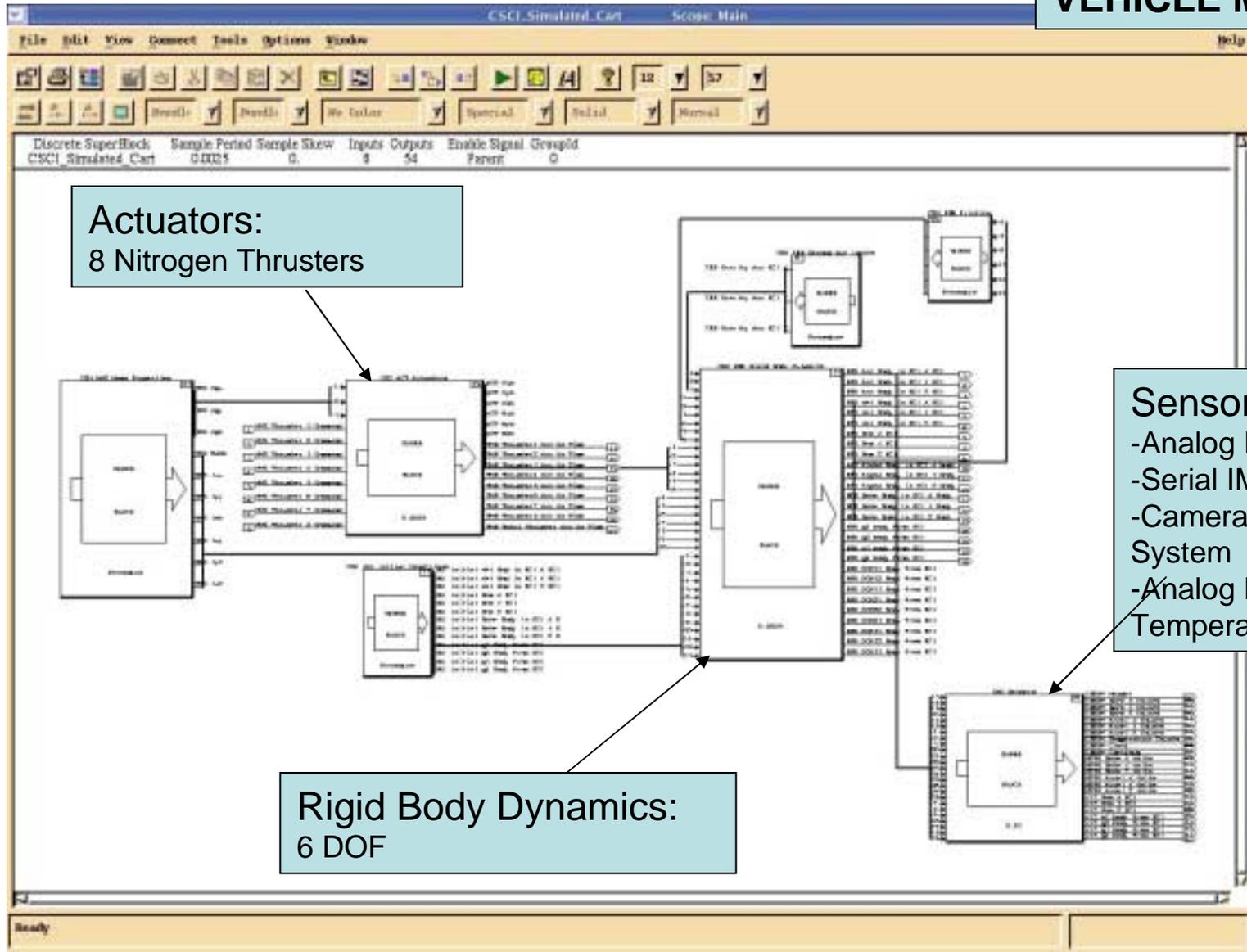
## Top Level Model:

- Simulated Cart
- Hardware Interfaces
- Flight Code





## VEHICLE MODEL





## FLIGHT SOFTWARE MODEL

### Vehicle Health Monitoring:

- Command Checking
- Sensor Limit Checking
- Hardware status

### Telemetry:

- Connect signals to populate
- External Script file creates database by interrogating model
- Currently implemented TCP/IP and 422 in TCM format.

### Command Processing:

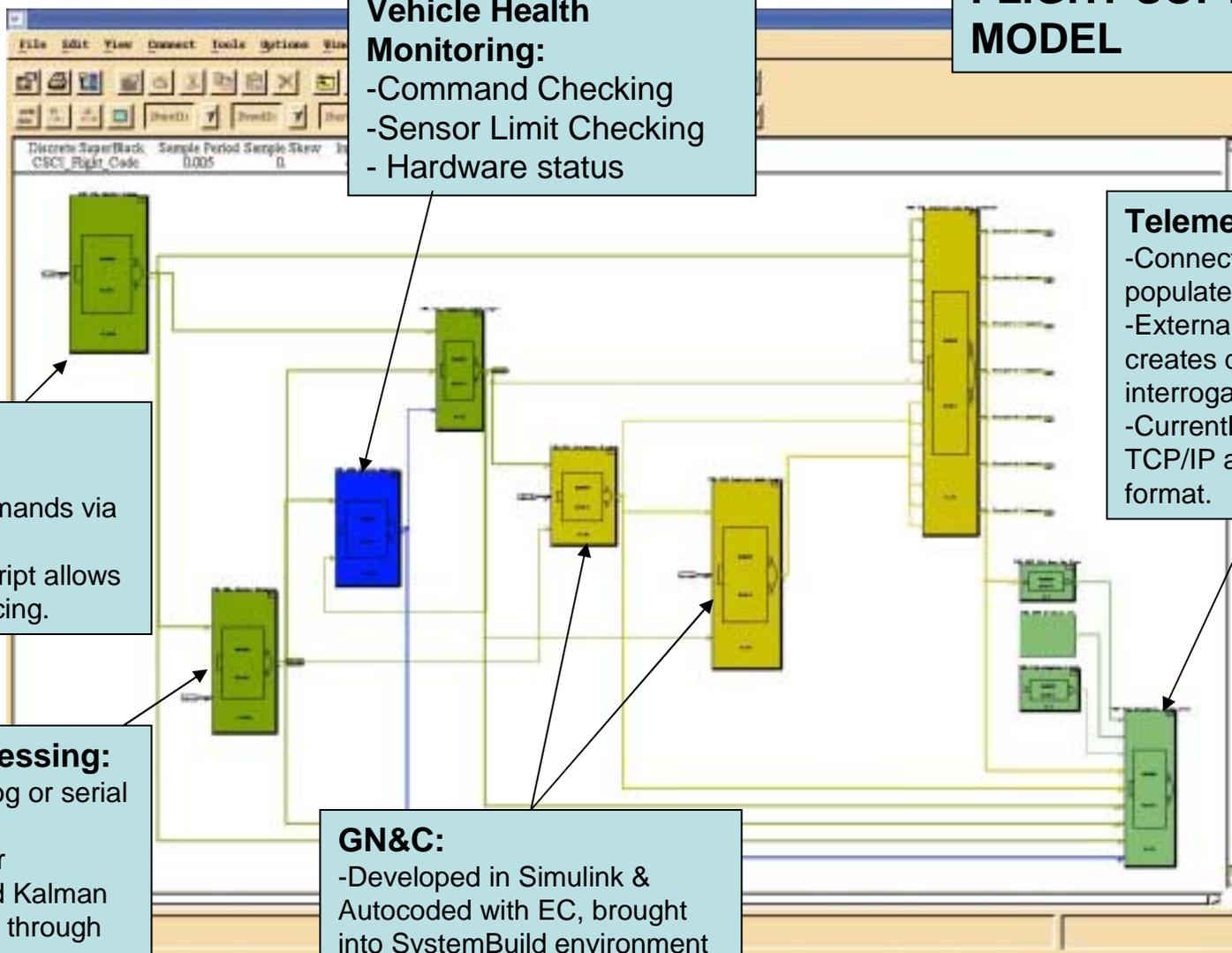
- Receives commands via TCP/IP or 422.
- Compiled in script allows flexible sequencing.

### Sensor Processing:

- Receives analog or serial data.
- Low Pass Filter
- Auto generated Kalman Filter integrated through UCB.

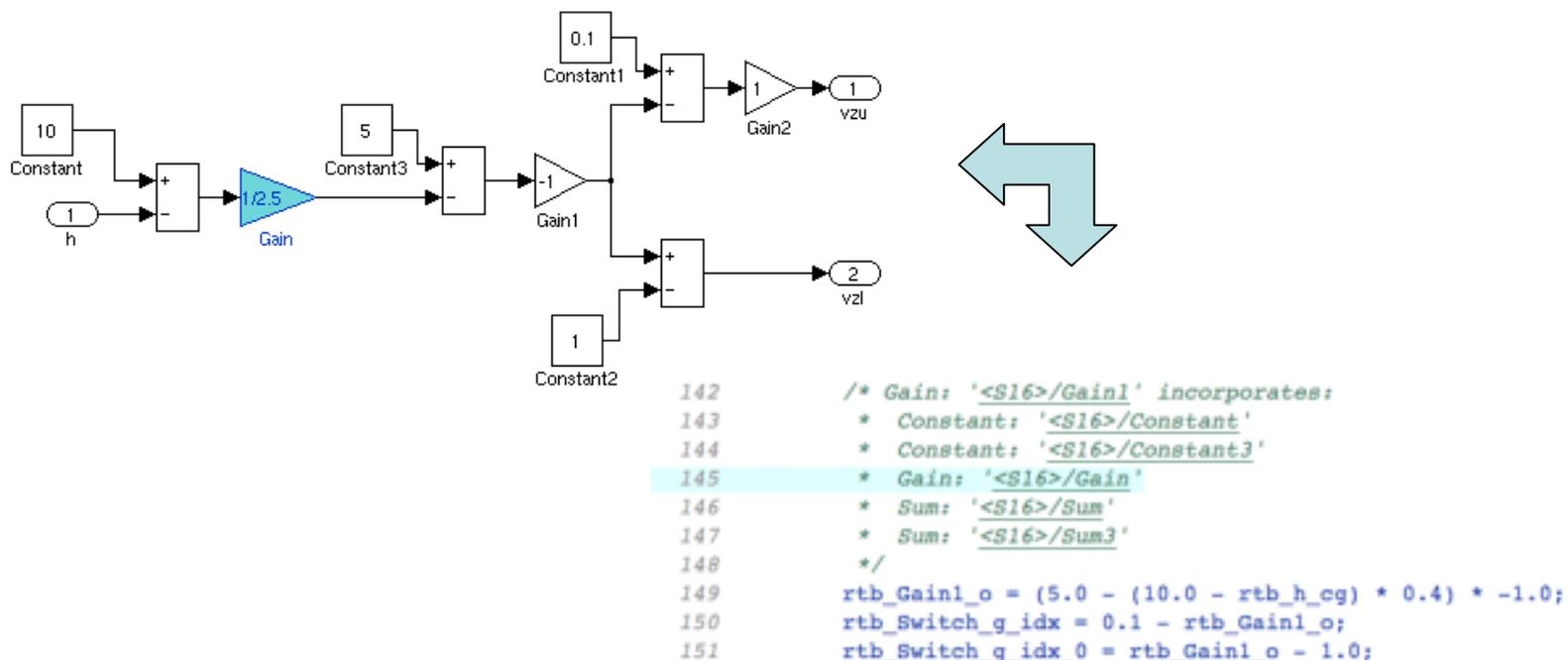
### GN&C:

- Developed in Simulink & Autocoded with EC, brought into SystemBuild environment using UCB.
- Multiple approaches investigated: Bang-Bang, PWPF





# Automatic Code Generation

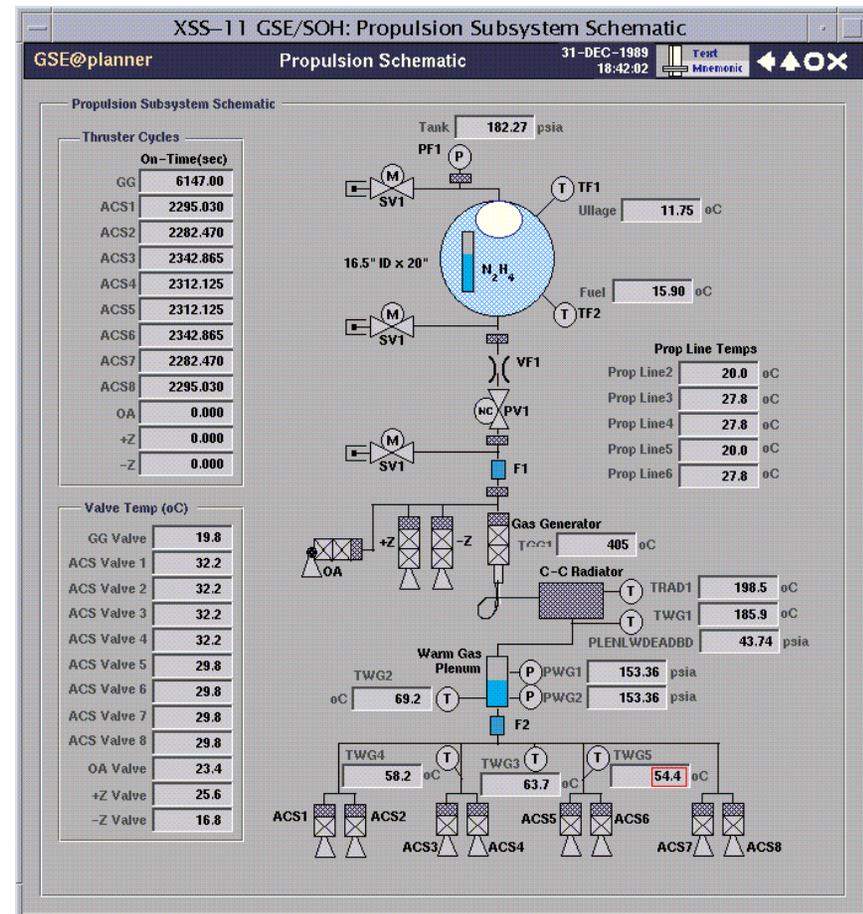


- Simulink supports two way trace-ability between models and generated code
- Code Easy to read, well commented



## Mission Ops Software

- Integrated with commercially available (Octant Technologies) mission operations software
  - CmdBuilder – GUI for spacecraft telecommands and scripting.
  - TelemScope – telemetry monitoring, archiving, and trend analysis.
- Investigations ongoing into ASIST and SCL





## V&V

- Utilizing html based documents for tracking requirements, procedures, specifications, and verification results.
- Unit test scripts exercise low level blocks within the model in the WSIM.
- Perl scripts for automatic execution of integrated tests in WSIM, PIL, and HWIL. Test results and links to data automatically populated in html docs.

Requirement ID	Description	Priority/Status	Acceptance Test Strategy
19.1	The control system shall provide position control of the 1 meter of a command signal when background light	Low/OK	
19.2	The control system shall not allow requirements that require a minimum of 10% of the specified for	Low/OK	
19.4	The control system shall provide for being sensitive to ground less 1.11	Low/OK	
19.5	There shall be no opening individual for every frequency to stop down	Low/OK	
19.6	The control system shall not command status such for location system	Low/OK	
19.7	The control system shall achieve a command position 1 meter from and	Low/OK	
19.8	The control system shall provide for that, maintenance of body position		

Req #	Requirement	Priority/Status (NASA/MSD/PL)	Acceptance Test Strategy
19.1	The operator console shall be located in the left vehicle structure	High/Low	High/Low
19.2	All positive commands and responses feedback shall be provided to users	High/Low	High/Low
19.3	Simple telemetry feedback shall be provided to users in 1000 Hz rate response 200	High/Low	High/Low
19.4	The software shall provide the capability to store the position way point. All subsequent position shall be relative to the last way point	High/Low	High/Low
19.5	The system shall provide real time feedback to the operator regarding position, orientation, distance from, distance to, time, center command	High/Low	High/Low
19.7	The system shall provide capability to store all telemetry feedback to the operator	High/Low	High/Low
19.8	For autonomous mode, the system shall have a resolution system that allows the operator to view the location of the structure in time	High/Low	High/Low
19.9	The system shall have data points buffered to the operator when necessary out of band	High/Low	High/Low

Req #	Requirement	Priority/Status (NASA/MSD/PL)	Acceptance Test Strategy
19.1	The system shall have a software state of health monitor that reports the state of all software and hardware components and	High/Low	High/Low
19.2	The system software shall check and correct the system time	High/Low	High/Low
19.3	There shall be a knowledge base on all processes, resources and operations. Violations shall result in storage of the document, logs	High/Low	High/Low



# Lessons Learned

- **Spiral Development Approach** – Prototype, code, test, and debug early and often.
- **Elimination of Errors** – Eliminated need to manually translate GN&C algorithms to flight code.
- **Reduced Training** – No need to teach control systems experts how to write “good code”.
- **Compatibility** – Using Simulink, SystemBuild and hand written code within the same development framework allows compatibility with various vendors, tools, and legacy code.
- **Enhanced Debugging** – Model based tools provide graphical debugging facilities in addition to standard embedded systems debugging tools.
- **Reuse** – The model based technique lends itself to reusing components. Successfully reused a majority of the software components for the 6 DOF integration tests that were originally built for the granite table tests.
- **Wide Applicability** – Approximately 85% of the software we have developed is automatically generated. Low level software more suited to hand coding. Didn't try to force it.



# Summary

- NASA Ames has been implementing an infrastructure for small spacecraft software development based on Automatic Code Generation techniques.
- Demonstrating this approach on two testbeds.
- Automatic Code Generation seems both feasible and desirable.
- Continuing to refine approach and look at tools/process trades.