



Model Driven Lunar Habitat Avionics Design

Use of Draper's PARADyM to Guide Design Decisions for Increased System Reliability

Ian R. Claypool
February 26, 2009

The Charles Stark Draper Laboratory, Inc.
555 Technology Square
Cambridge, MA 02139
iclaypool@draper.com



Agenda

- LSS Habitat Modeling Effort
- Designing for Increased System Reliability
- PARADyM Overview
- LSS Habitat Model Development Process
- Preliminary Results



Model Driven Lunar Habitat Avionics Design

LSS HABITAT MODELING EFFORT



Context

- LSS has been studying Avionics for lunar surface applications for several years with the intent being to develop common architecture(s) to support multiple vehicles.
 - The “Avionics” referred to here is any system or component that has an electrical connection.
 - Includes power, thermal, data, sensing, actuation, comms, ...
 - This is the only context to talk about Fault Tolerance, Reliability, ... for Redundant Systems
- Most studies are segmented at subsystem boundaries and do not cover effects resultant from integration of subsystems. This leaves latent risks to be discovered during later systems integration efforts as:
 - Failure modes that manifest as the result of integrating systems
 - Fault propagation across subsystem boundaries
- Draper is funded by NASA ESMD to model the Habitat and Lunar Electric Rover Avionics and evaluate the potential for commercial electronics technology to meet reliability constraints and improve performance
- Current study task is to develop and evaluate a systems based model of the NASA provided reference designs
- This is an iterative approach to refine both the systems and architectures
- Incorporates an integrated analysis of multiple subsystems to expose interdependencies and increase total system reliability
- In turn, this provides a better understanding of the systems of interest and more strategic investment in development of technologies and capabilities



Executive Summary

- A functional model of the Habitat System reference design has been created in Simulink for use with the PARADyM tool.
 - Models the full system down to the Bus Interface Unit (i.e. remote I/O control) level.
- Now in the process of refining our understanding of redundancy and CONOPs for the ECLS Systems to better fold them into the model.
- We are at the point where it is possible to use the model to probe the system architecture and ask questions/evaluate alternatives.
 - What components are driving overall system reliability?
 - Where can we use COTS level reliability with minimal impact to the overall system reliability?
 - Where would we benefit from additional redundancy?
- We have identified potential single point failures and they are already being designed out.
 - Have so far been able to improve system reliability from 95% to 98% (and we are just getting started)
- Please Note: “Reliability” as it is used in this presentation, particularly in the results, is best thought of as a metric for evaluating the architecture.
 - It should not be quoted as a prediction of reliability for this system!



Model Driven Lunar Habitat Avionics Design

DESIGNING FOR INCREASED SYSTEM RELIABILITY



Using Reliability Estimation as a Design Tool

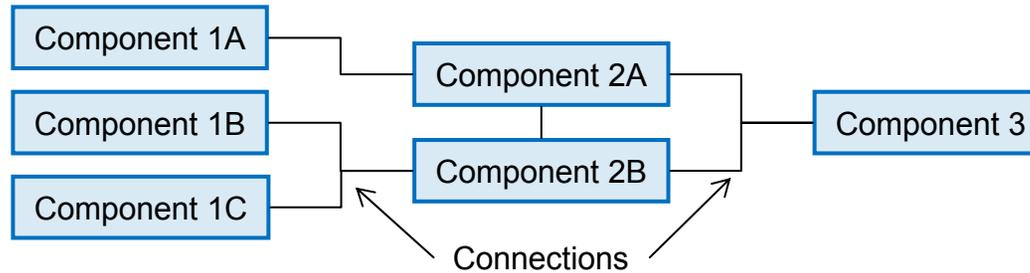
- The ultimate objective of any system is to:
 - perform a defined function
 - within specified operating limits
 - for a desired mission duration.
- The reliability of a system is a measure of its ability to meet this objective.
 - How does the system continue to perform in the face of a component failure?
 - How can the probability of achieving the system objective be increased?
 - Increase component MTBF? Won't work if the source of failure is external to the system – ie. a flock of geese.
 - Add components (redundancy)? Won't help if the components are prone to failure.
 - Both? Fine if you have unlimited \$s and don't mind the extra weight.
 - Other issues:
 - Fault coverage
 - Failure identification
 - Component repair/replacement strategy



Features of System Reliability Analysis

Failures Cause System Configuration to Change

System:
Consists of
components and
connections



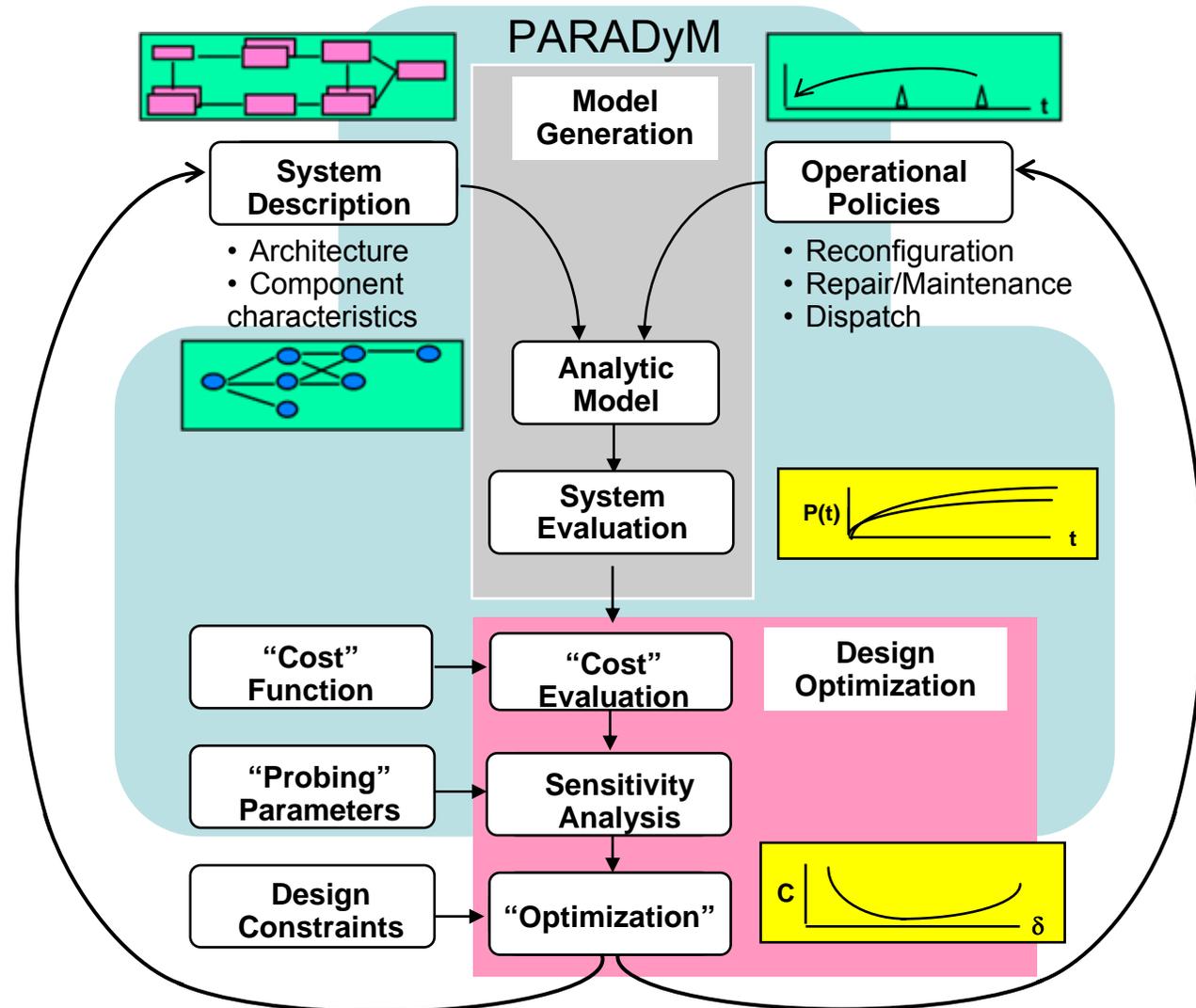
- Component and connection failures move the system into different configurations.
- Some of these configurations do not stop the system's desired function.
- Model evaluates likelihood of getting into each configuration.
- The sum of the probabilities of the operational configurations is the system reliability.
- We use this measure to explore alternative architectures
 - Number of components and their connections
 - Quality of components, approach to maintenance



Refining the System Architecture

Part of a Systematic Process of Design Optimization

- Uses sensitivity to “probe” the design
- Validate design
- Identify:
 - Issues
 - Drivers
 - Opportunities
- Payoff vs. cost of corrective actions





● Sensitivity-Based Methodology

- Provides Systematic Improvement of Design
 - Initial Design May Be Simple or Complex
 - Provides Measure of Robustness / Risk
- Provides Insight Into the Design Decisions
 - Identifies Potential Problem Areas
 - Quantifies the Effects of Design Changes
 - Focuses Attention on Critical Areas
- Provides Well-Balanced Design
 - Distributes Component Contributions Equitably
 - Prevents Over-Design
 - Provides Stopping Criteria

■ Tools and Methods

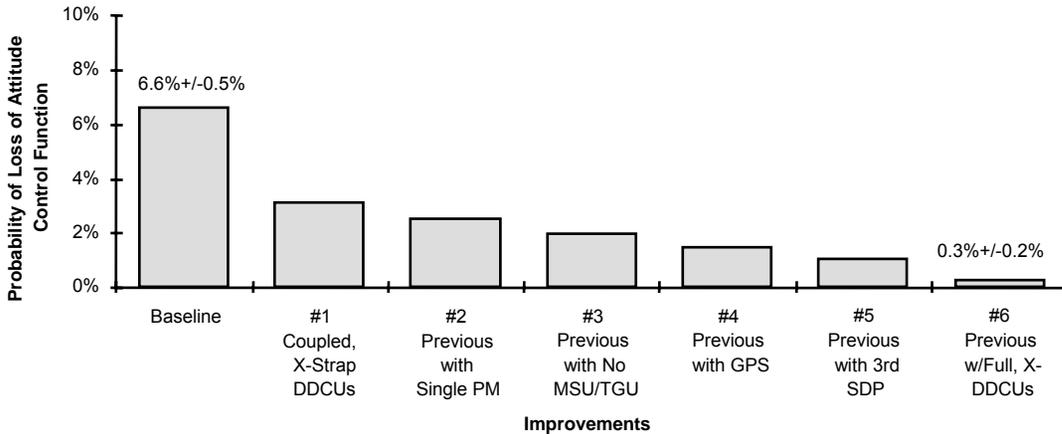
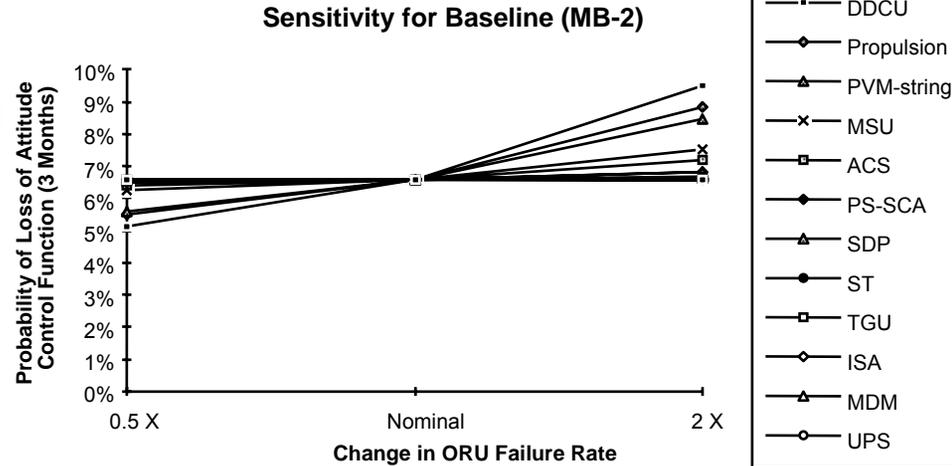
- 0th Order Analysis
- Flexibility in Analysis, Modeling Resolution
- Automated Construction of System-Level, Multiple Fault FMECA (PARADyM)
- Reward Models
- Object Process Network (OPN)
- Multi-Attribute Tradespace Exploration (MATE)
- FFBDs, DoDAF OV-5 (Operational Activity Model)
- Sensitivity-Based Analyses



Space Station Freedom Example

Using Sensitivity Analysis to Improve System Design

- Baseline design showed early Station had 6.6% probability of losing control (= lose Station) in each 3-month period between shuttle visits
- Sensitivity analysis:
 - Exposes drivers that have significant impact on the metric of interest
 - Provides a systematic approach for maximum benefit at minimum cost



- Each improvement driven by sensitivity analysis for maximum benefit at minimum cost
- Loss of attitude control driven from 6.6% to 0.3% for less than 0.1% of development cost



Reliability Estimation with PARADyM

Building Markov Models from System Behavioral Analysis

- PARADyM (Performance and Reliability Analysis via Dynamic Modeling) is Draper's latest system reliability evaluation toolbox
 - Built with the MATLAB®-Simulink® family of products
- PARADyM uses a behavioral model of the system to evaluate nominal and degraded performance
 - Allows for automated generation of Markov models using a performance based definition of what constitutes a failed system state
- Reliability estimation enabled through propagation of component failure rates in Markov model
 - Failure sensitivity analysis is used to find the components that drive system loss

The flowchart illustrates the PARADyM workflow:

- System Component Layout & Interactions:** A block diagram showing various system components like 'LUNAR GEAR SYSTEMS', 'SAFE AND ROBUST', 'ACCELERATION CONTROL', 'RELIABILITY MONITORING', 'VEHICLE CONTROL', 'VEHICLE STATE', 'VEHICLE PERFORMANCE', 'VEHICLE MODELS', 'VEHICLE CONTROL', 'VEHICLE STATE', 'VEHICLE PERFORMANCE', 'VEHICLE MODELS', 'VEHICLE CONTROL', 'VEHICLE STATE', 'VEHICLE PERFORMANCE', 'VEHICLE MODELS'.
- System Behavioral Model:** A Simulink-style block diagram representing the dynamic behavior of the system.
- Model Parsing & Execution Control:** A screenshot of the PARADyM software interface showing simulation parameters and results.

Time for failure injection in dynamic simulation, seconds	2
File name for final report (omit extension)	simple20161012
Truncate evaluation	3
Truncation level	Save as default
- Operational Timeline & Performance Parameters:** A graph showing performance metrics over time, with a vertical dashed line indicating when a failure occurs.
- Nominal & Degraded Performance:** A graph showing two performance curves (red and blue) over time, with a vertical dashed line indicating when a failure occurs.
- Markov Model Assembly & Solution:** A state transition diagram with nodes and arrows, accompanied by differential equations:

$$\frac{dP_1}{dt} = -(a+b)P_1(t)$$

$$P_1(t) = e^{-(a+b)t}$$

$$\frac{dP_2}{dt} = aP_1(t) - bP_2(t)$$

$$P_2(t) = e^{-bt} - e^{-(a+b)t}$$
- System Reliability:** A screenshot of the PARADyM results window showing a table of failure rates and a histogram of accuracy.

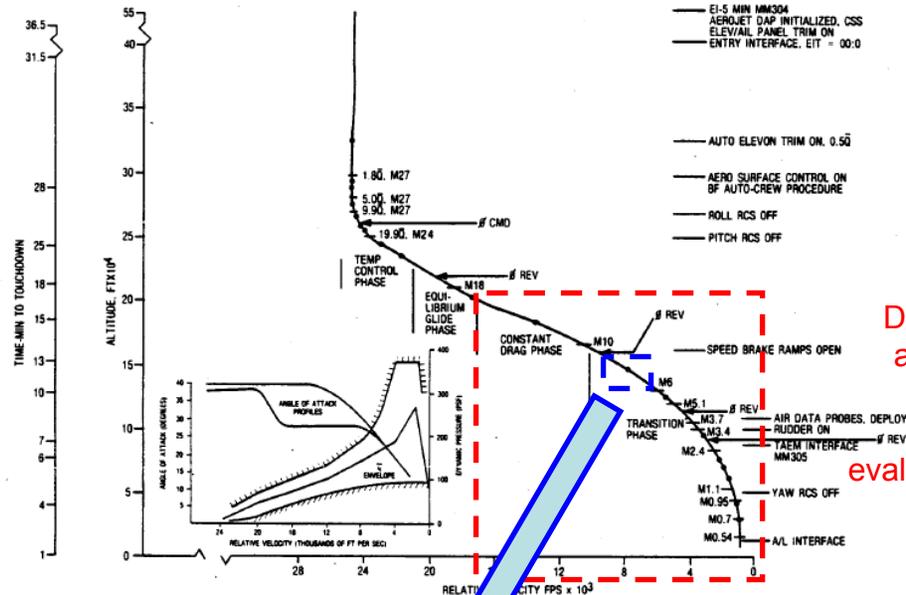
Failure Level	System Rel
0	0.34
1	0.35
2	0.04
3	0.01
- Component Sensitivities:** A bar chart showing the sensitivity of the system reliability to various components.



Using Performance to Capture System Loss

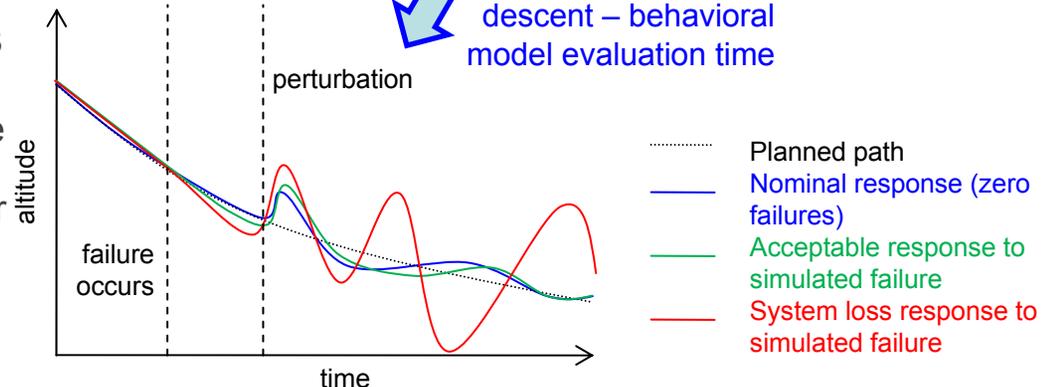
Determining Operational Status from Degraded Performance, not Minimum Equipment

- As the number of potential system configurations grows, it is necessary to utilize a consistent technique to determine operational status
 - This can be done *a priori* (such as a minimum equipment list) or *a posteriori* (modeling system performance metrics)
- Performance modeling utilizes a representative system behavioral model that captures nominal and degraded performance
 - The user quantifies tolerance for system loss in terms of performance metrics
- Using a behavioral model opens up a range of new possibilities
 - Direct simulation of multiple failure modes per component
 - Informed “push-back” on customer requirements
 - Automated generation of Failure Modes & Effects Analysis (FMEA)



Descent and approach to landing – Markov evaluation time

Kafer, G. C. “Space Shuttle Entry / Landing Flight Control Design Description,” 1982.





Model Driven Lunar Habitat Avionics Design

PARAD_YM OVERVIEW



PARADyM Overview

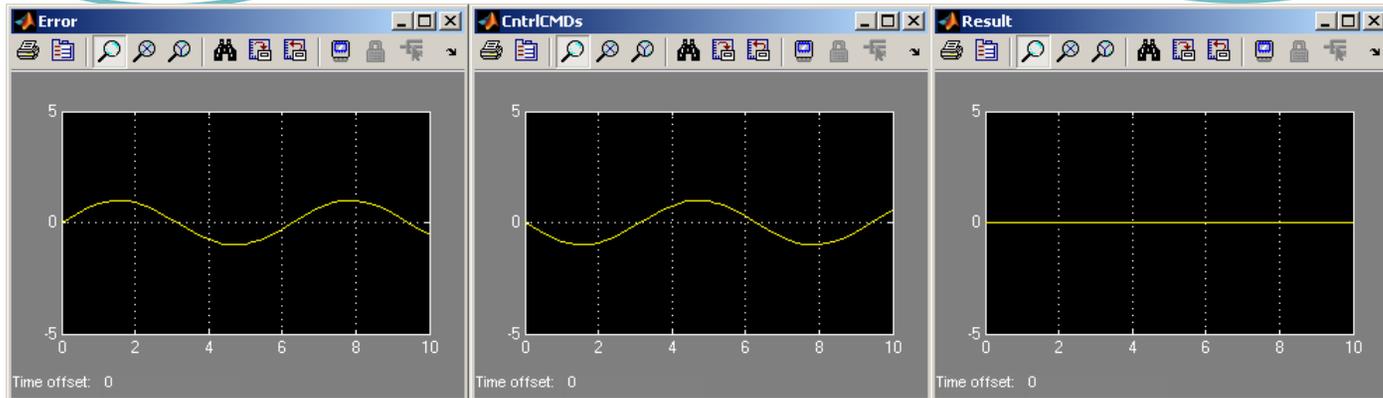
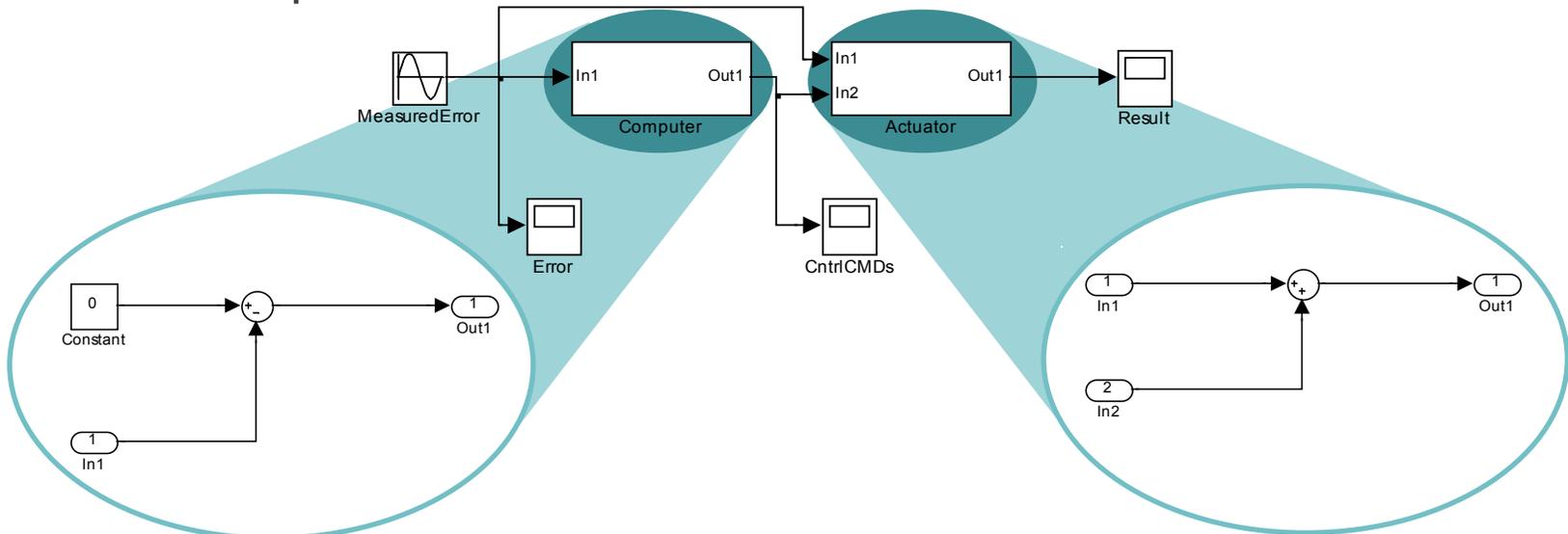
- PARADyM is the Draper design improvement process and uses as a tool the PARADyM software.
- PARADyM is built with the MATLAB[®] and Simulink[®] family of products
 - These products are widely used and supported and have excellent embedded functionality for systems analysis
- The PARADyM software is a toolbox of custom MATLAB functions, graphical user interfaces, and Simulink libraries, which enable a process for concurrent reliability and degraded performance analysis
- How can PARADyM and reliability estimates be used to probe the design of a system and indentify potential improvements?
 - Lets look at a very simple model of a control system:



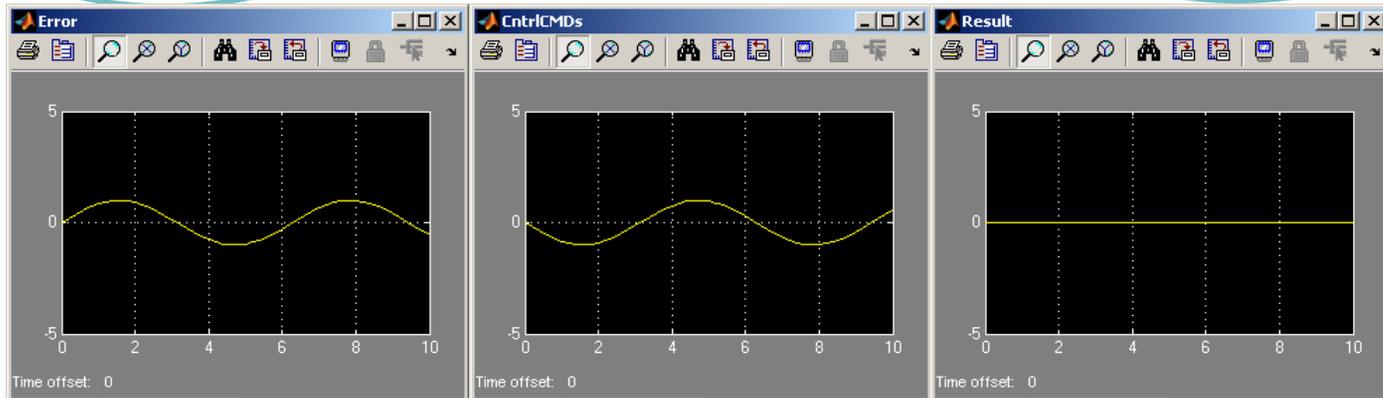
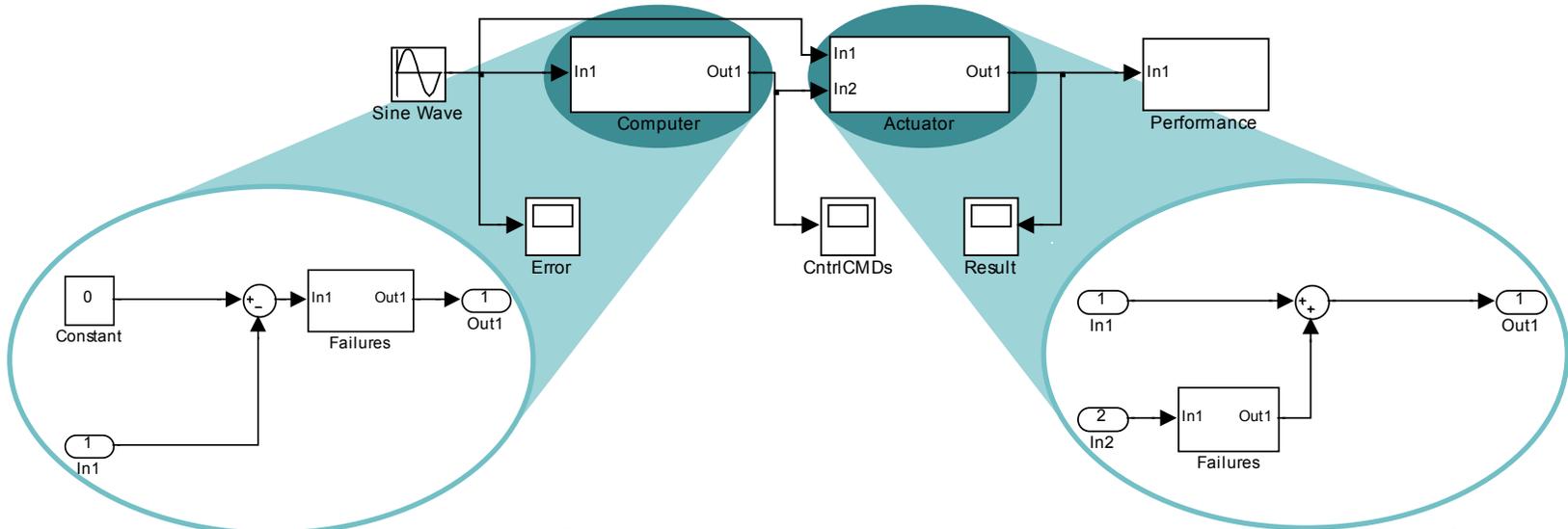
Simple Control System Model

System Modeled in Simulink

- Input is measured error
- Desired output is zero error



- PARADyM Blocks added to the original system
 - Failure Blocks & Performance Block

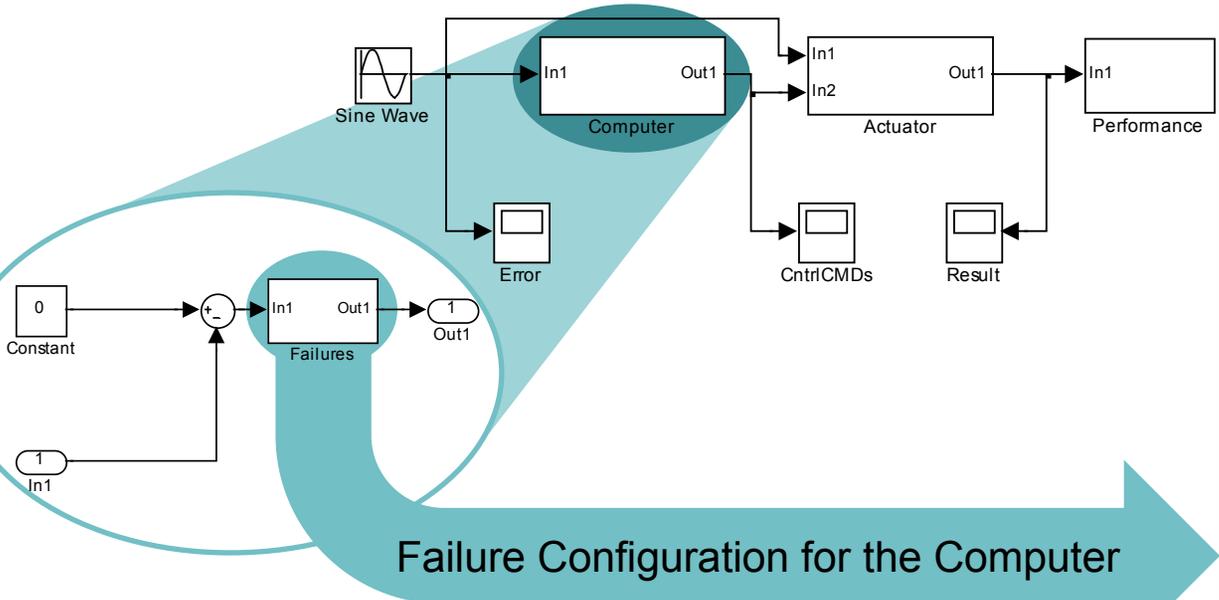




Simple Control System Model

Types of Failures

- Failure blocks provide for a number of different failure modes
 - Anything that can be coded in MATLAB can be injected as a failure



Function Block Parameters: Failures

Failure Specification (mask) (link) —

This failure block modifies an input signal to simulate component failure when used for reliability analysis with PARADyM. Use only one failure block per subsystem, and do not modify its name (i.e. do name call it Failures1, etc.). Check the failures you wish to allow and the corresponding failure rate per hour (the inverse of the MTBF). Currently, this does NOT support symbolic or time-dependent failure rates (i.e. rate must be a static, scalar value). The distinction between the different failure modes is given in the help file.

Parameters —

- Omission
- Full-scale deflection
- Gain change
- Bias
- Stuck at last value

Failure rate for omission:

Failure rate for full-scale deflection:

Value for full-scale deflection:

Failure rate for gain change:

Factor for gain change:

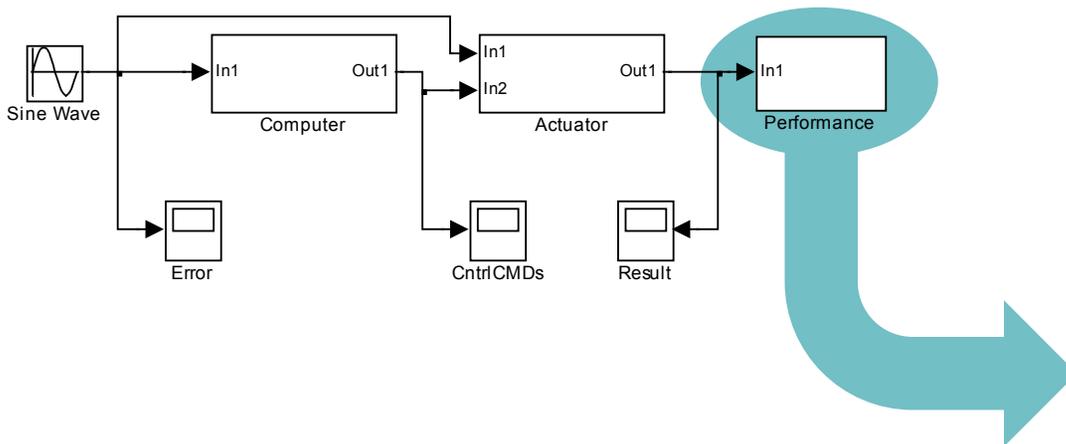
Failure rate for bias:

Value for bias:

Failure rate for stuck at last value:

OK Cancel Help Apply

- Performance blocks provide the metrics for determining if a system is still operating at acceptable levels in the face of the failure/failures that have been injected.
 - Upper and lower limits can be specified relative to the nominal system performance.
 - Transient responses can also be limited.



Sink Block Parameters: Performance

Performance Metric Specification (mask) (link)

This block saves the relevant performance metrics for reliability evaluation using PARADyM. The input should be whatever signal is used for pass / fail criteria when evaluating individual Markov states (i.e., roll rate, thruster chamber pressure, etc.). The steady transient checkbox allows the user to choose if a transient will be used in evaluation. Generally, transient failures allow looser tolerances. The upper and lower bounds for both should be in absolute allowed deviation from the input signal. Multiple inputs can also be used, but should be put into vector format using the Mux or Bus Creator to send the single signal into the Performance block. In this case, the absolute allowable deviation should be input as a row vector in the mask dialogue box.

Parameters

Transient

Maximum allowable absolute deviation from steady state, lower

Maximum allowable absolute deviation from steady state, upper

Transient length, sec

Maximum allowable absolute deviation from transient, lower

Maximum allowable absolute deviation from transient, upper

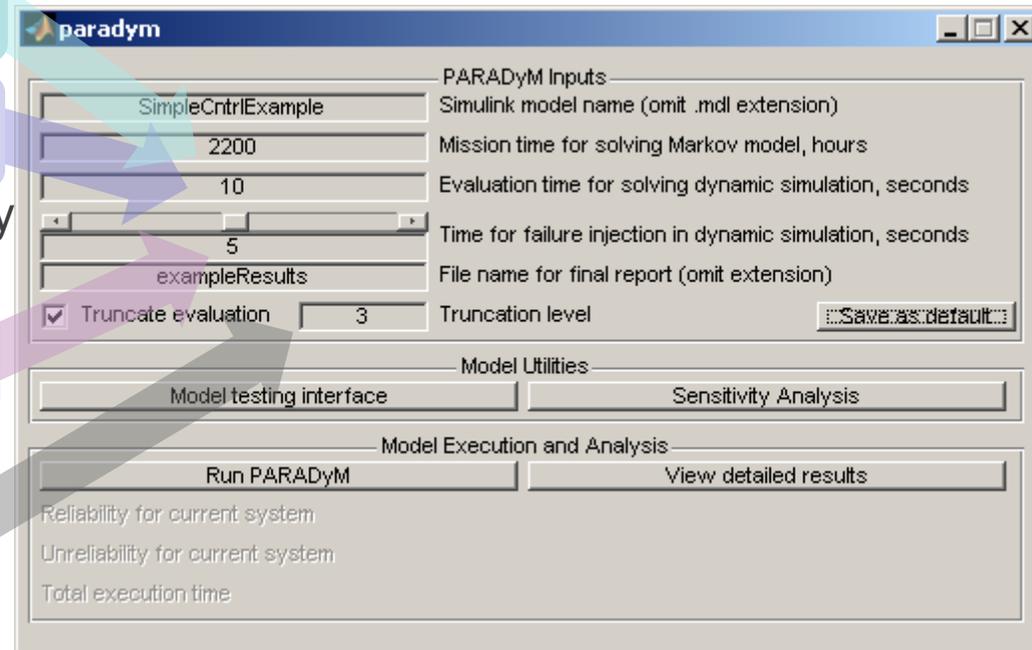
OK Cancel Help Apply



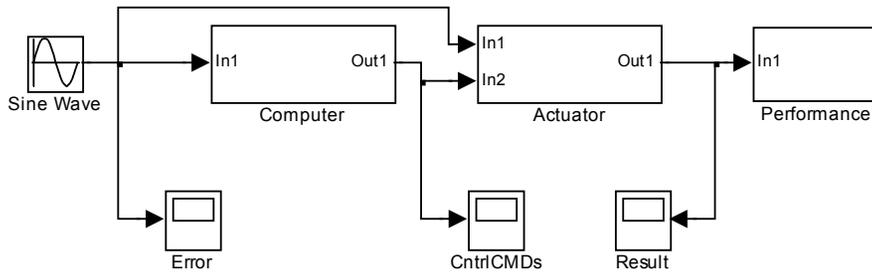
PARADyM Interfaces

Main Window

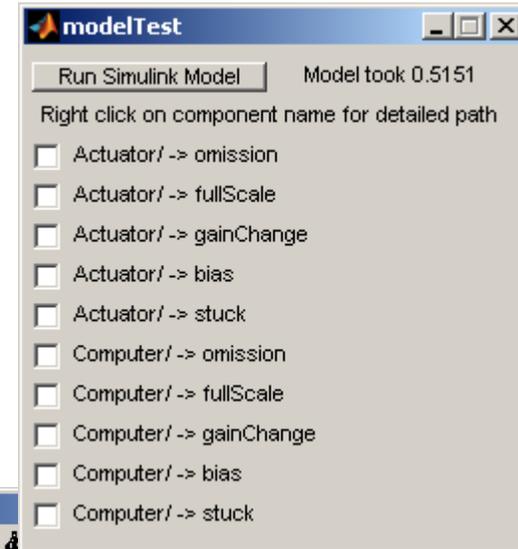
- The main PARADyM window is used to input the parameters which guide the automatic generation of Markov states and evaluation of overall system reliability and performance
 - How long is the system required to be in operation?
 - Over what duration are the dynamics to be evaluated?
 - A period of high sensitivity
 - Steady state
 - When should the failures be injected?
 - After how many failure levels should the evaluation be stopped?



- The model testing interface allows for stepping through individual failures.
 - This can be used to troubleshoot and validate the modeling.
 - It is also informative in regard to the design being evaluated.



System "Good"

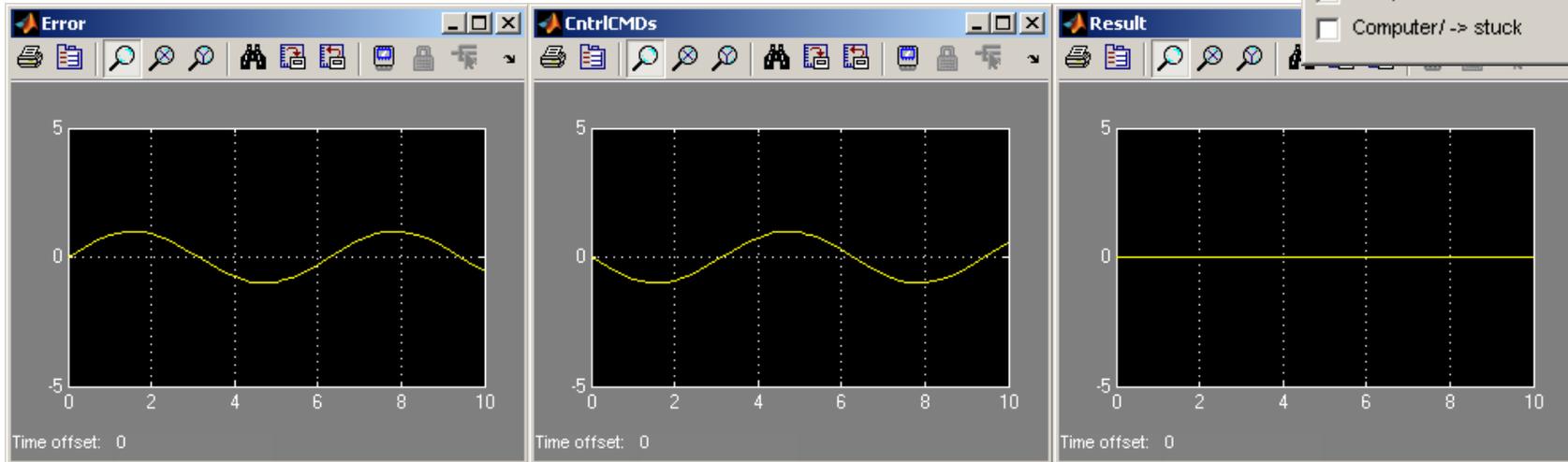


modelTest

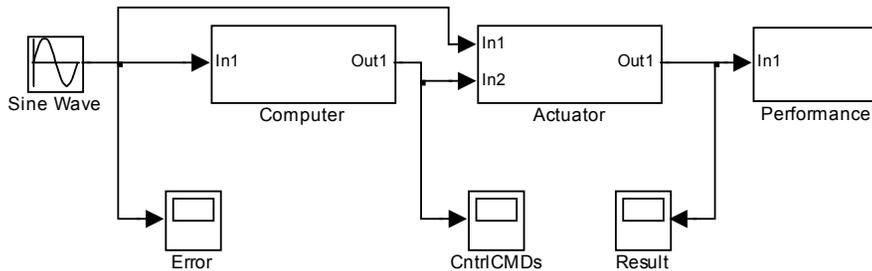
Run Simulink Model Model took 0.5151

Right click on component name for detailed path

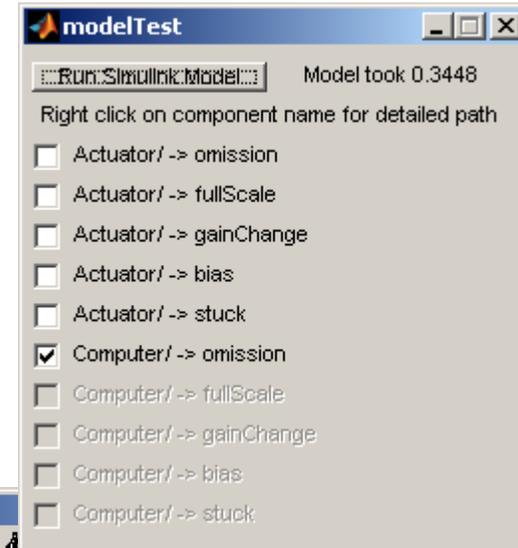
- Actuator/ -> omission
- Actuator/ -> fullScale
- Actuator/ -> gainChange
- Actuator/ -> bias
- Actuator/ -> stuck
- Computer/ -> omission
- Computer/ -> fullScale
- Computer/ -> gainChange
- Computer/ -> bias
- Computer/ -> stuck



- The model testing interface allows for stepping through individual failures.
 - This can be used to troubleshoot and validate the modeling.
 - It is also informative in regard to the design being evaluated.



System "Failed"

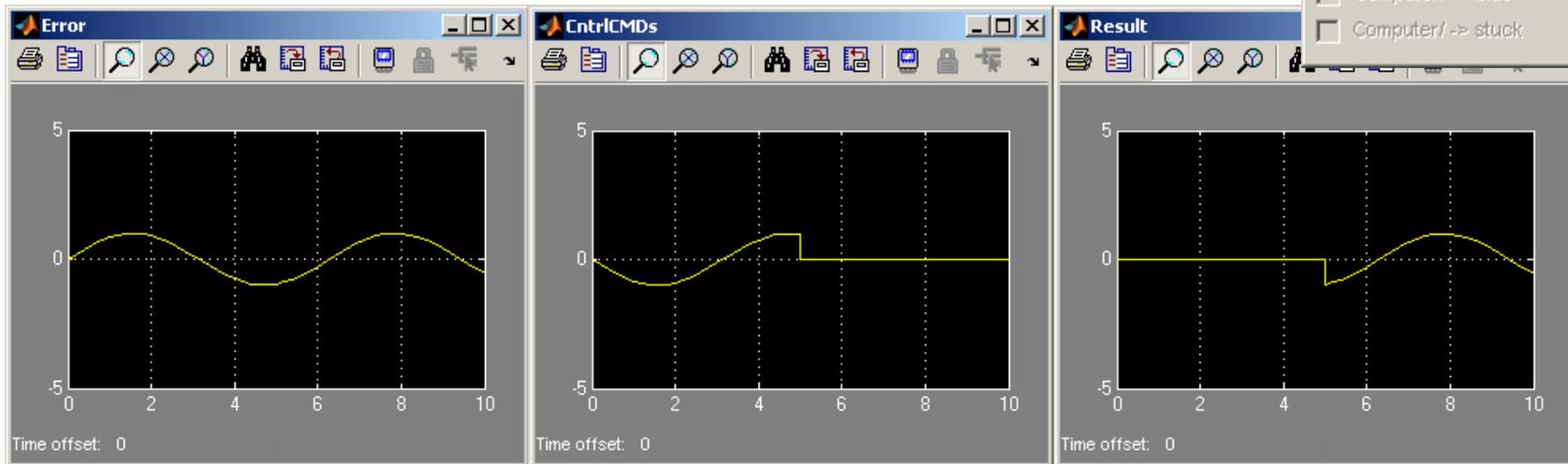


modelTest

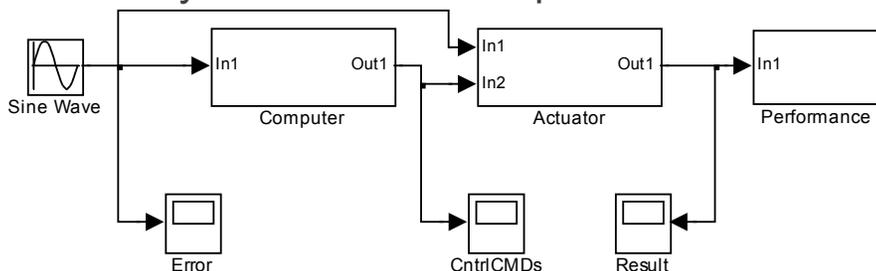
Run: Simulink Model... Model took 0.3448

Right click on component name for detailed path

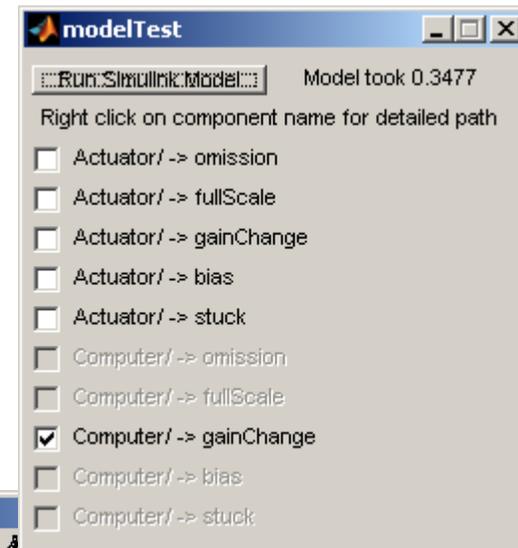
- Actuator / -> omission
- Actuator / -> fullScale
- Actuator / -> gainChange
- Actuator / -> bias
- Actuator / -> stuck
- Computer / -> omission
- Computer / -> fullScale
- Computer / -> gainChange
- Computer / -> bias
- Computer / -> stuck



- The model testing interface allows for stepping through individual failures.
 - This can be used to troubleshoot and validate the modeling.
 - It is also informative in regard to the design being evaluated.
- PARADyM automates this process.



System “?”

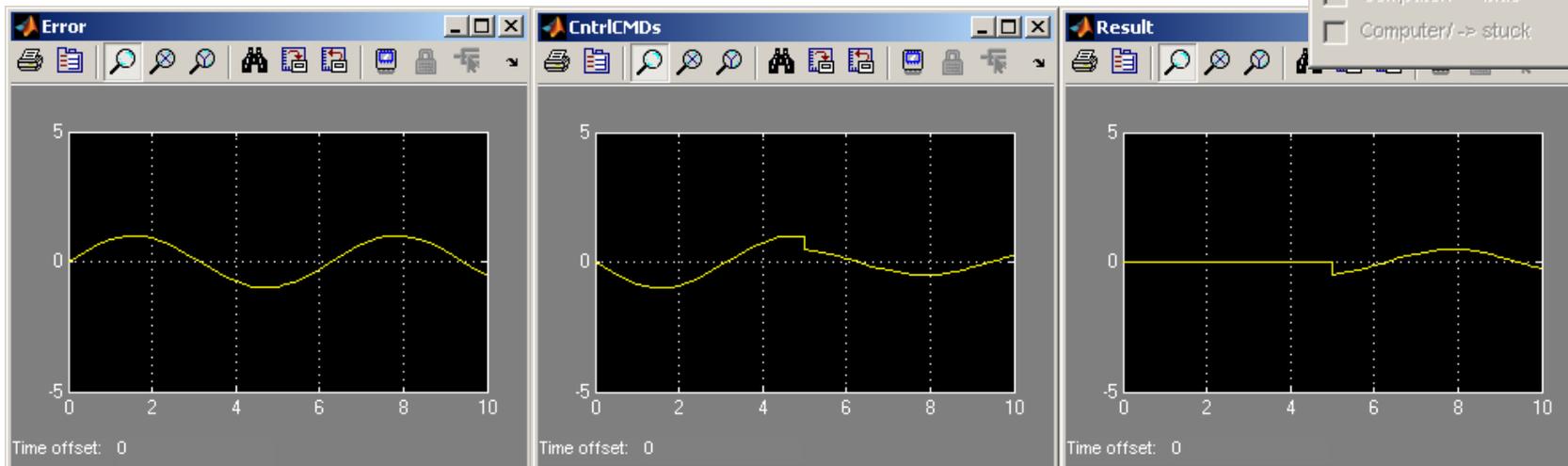


modelTest

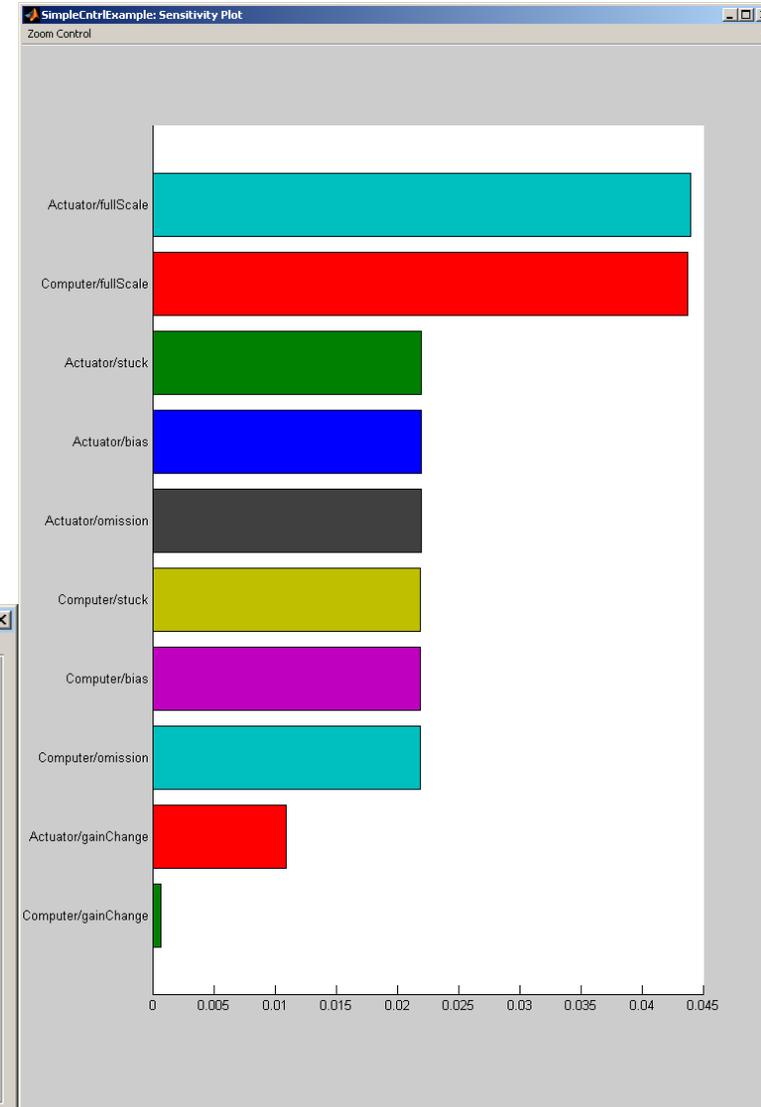
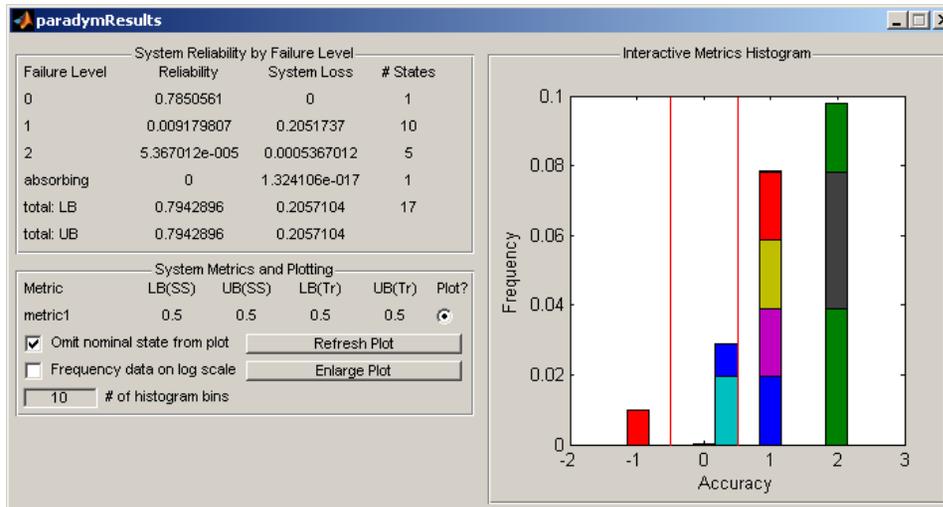
Run: Simulink Model: Model took 0.3477

Right click on component name for detailed path

- Actuator/ -> omission
- Actuator/ -> fullScale
- Actuator/ -> gainChange
- Actuator/ -> bias
- Actuator/ -> stuck
- Computer/ -> omission
- Computer/ -> fullScale
- Computer/ -> gainChange
- Computer/ -> bias
- Computer/ -> stuck



- Results include
 - system reliability/unreliability
 - Interactive graphical summary of good and failed dynamic states
 - Sensitivity analysis showing variation in system reliability due to a 1% change in component reliability.





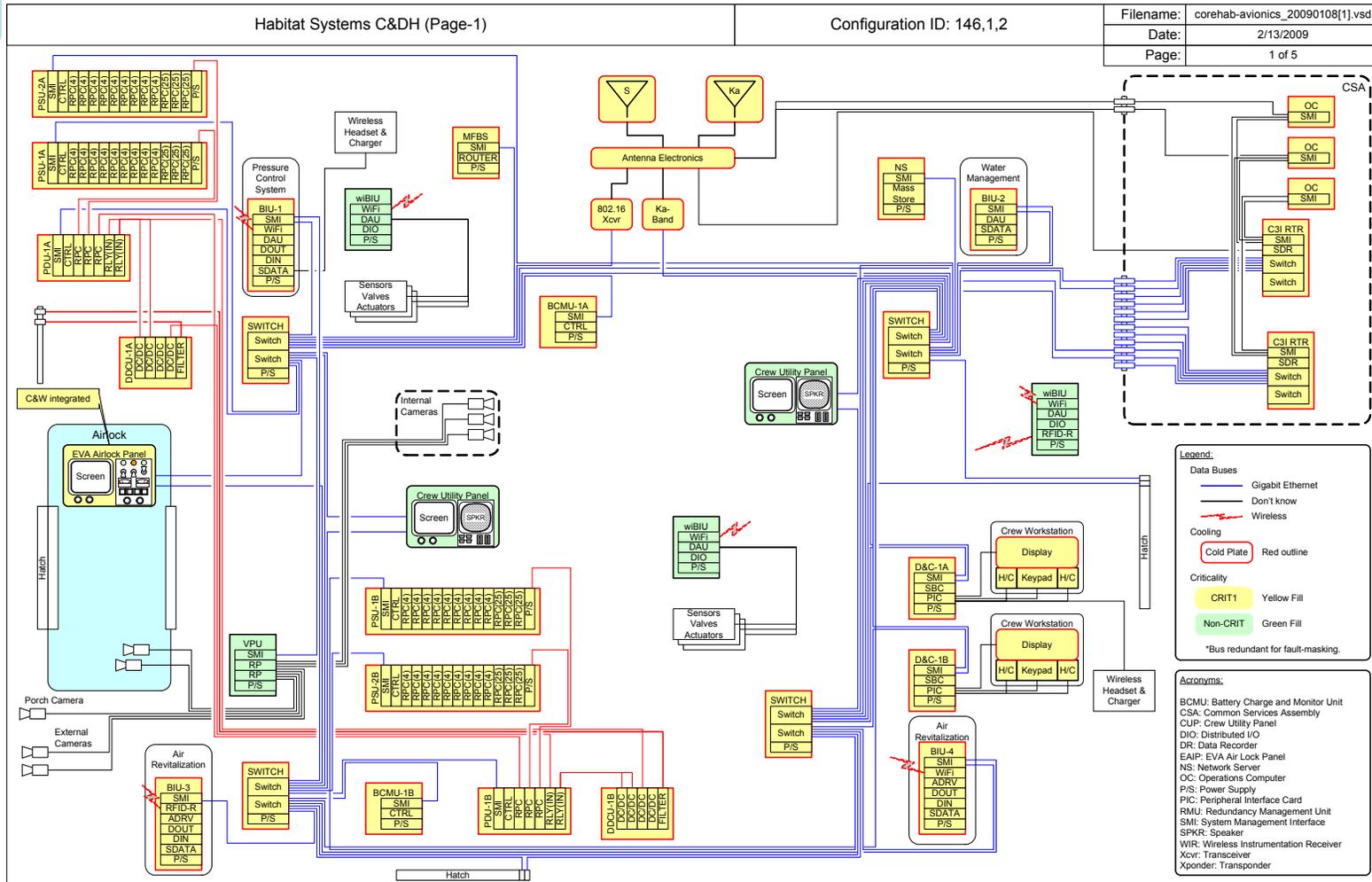
Model Driven Lunar Habitat Avionics Design

LSS HABITAT MODEL DEVELOPMENT PROCESS

Habitation Model Origin

- NASA had high level schematics of the habitat primary systems

- C&DH
- Power
- ECLS

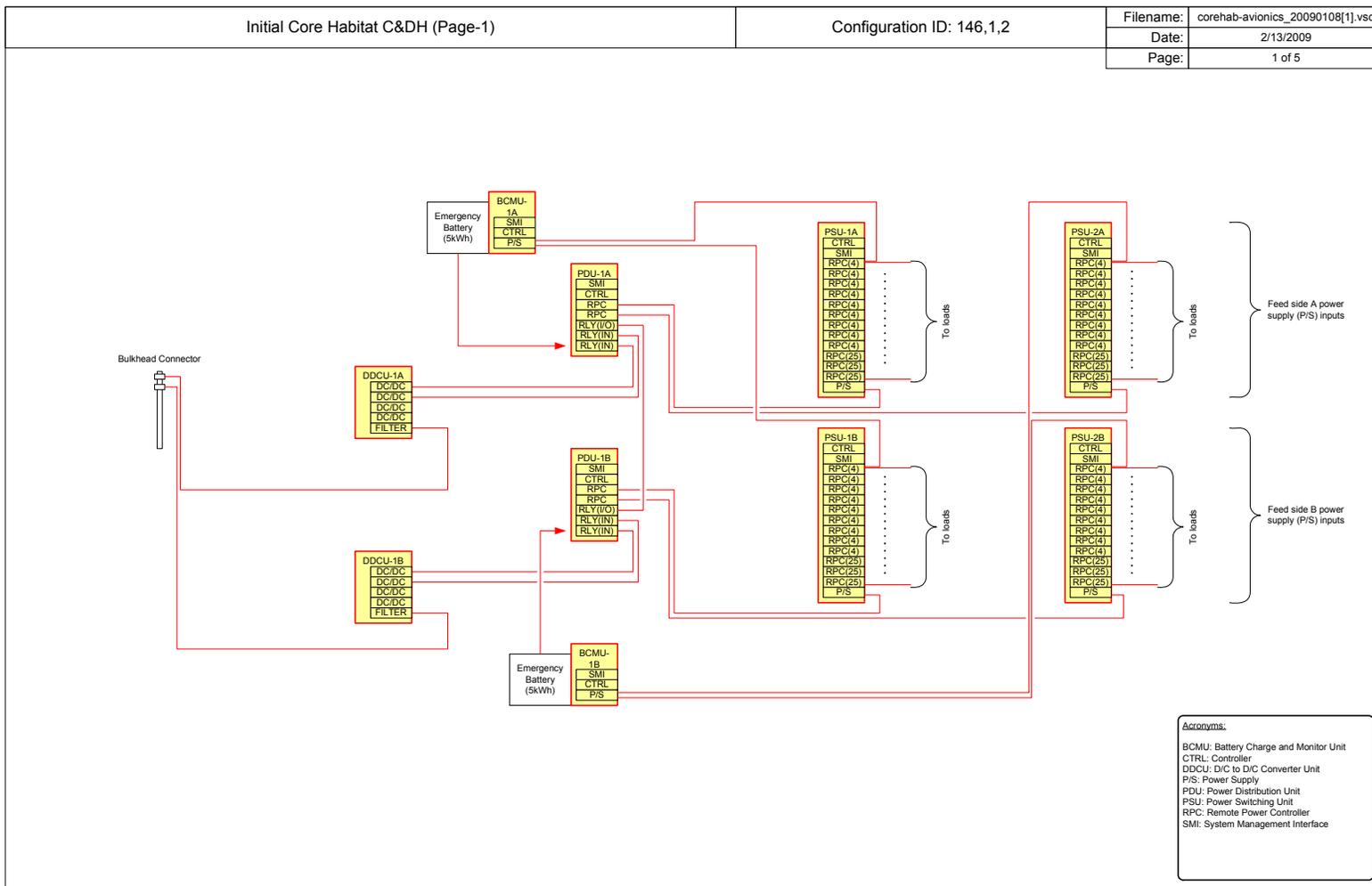




Habitation Model Origin

- NASA had high level schematics of the habitat primary systems

- C&DH
- Power
- ECLS



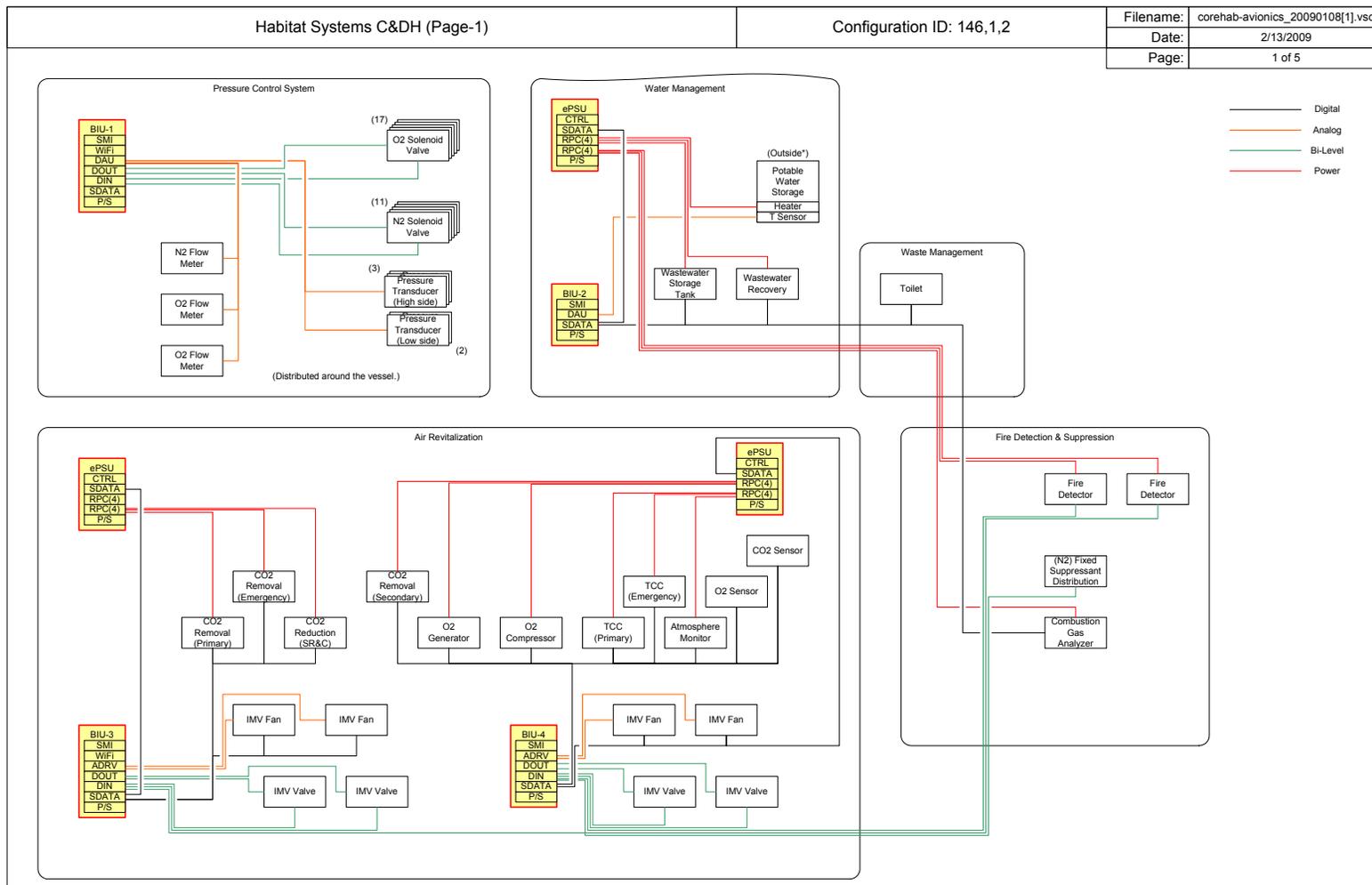
Acronyms:
 BCMU: Battery Charge and Monitor Unit
 CTRL: Controller
 DDCU: D/C to D/C Converter Unit
 P/S: Power Supply
 PDU: Power Distribution Unit
 PSU: Power Switching Unit
 RPC: Remote Power Controller
 SMI: System Management Interface



Habitation Model Origin

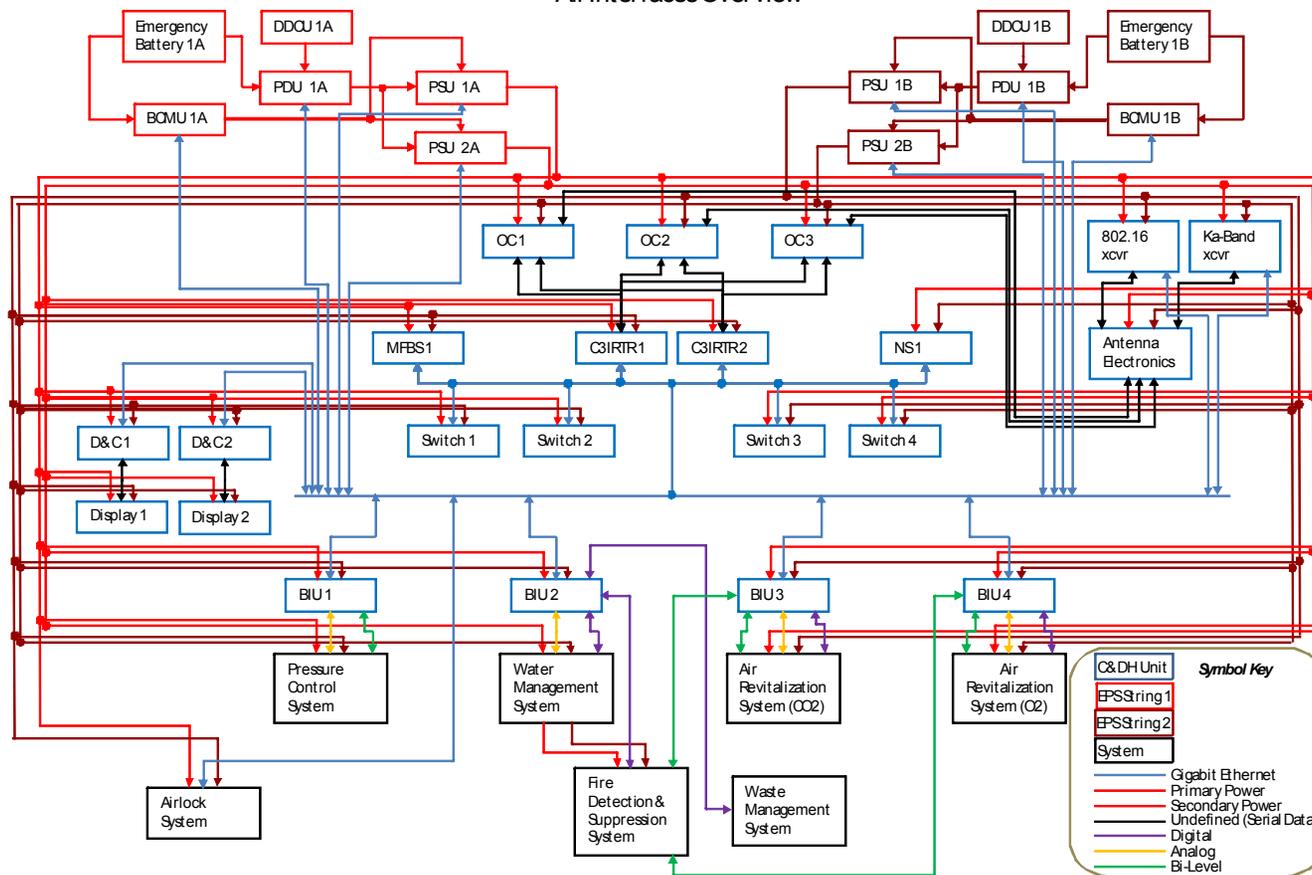
- NASA had high level schematics of the habitat primary systems

- C&DH
- Power
- ECLS



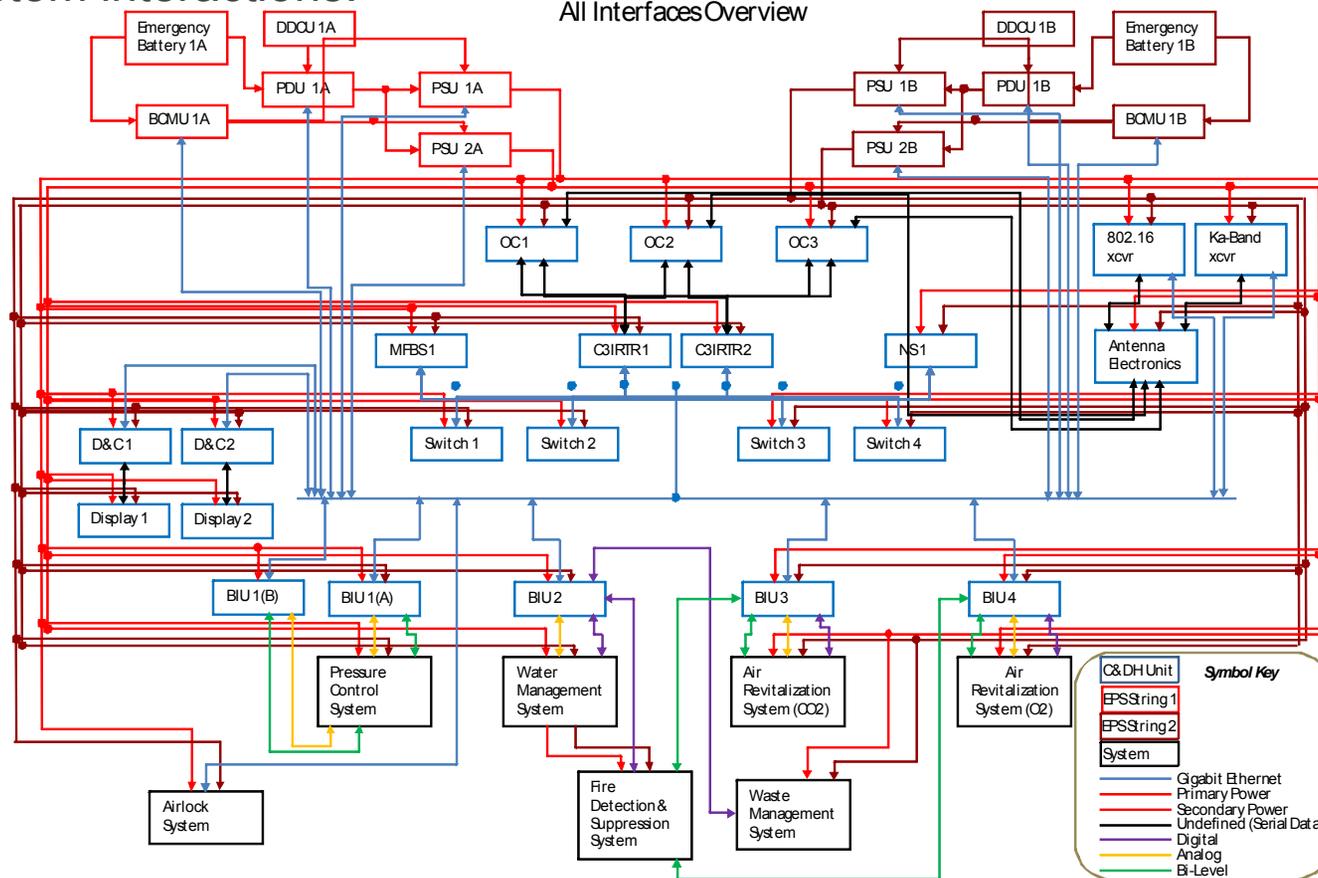
- Draper started by creating a functional diagram that represented how we interpreted the schematics with emphasis on what one component was passing (information, power, commands) to another.

Lunar Habitat Functional Diagram
 All Interfaces Overview



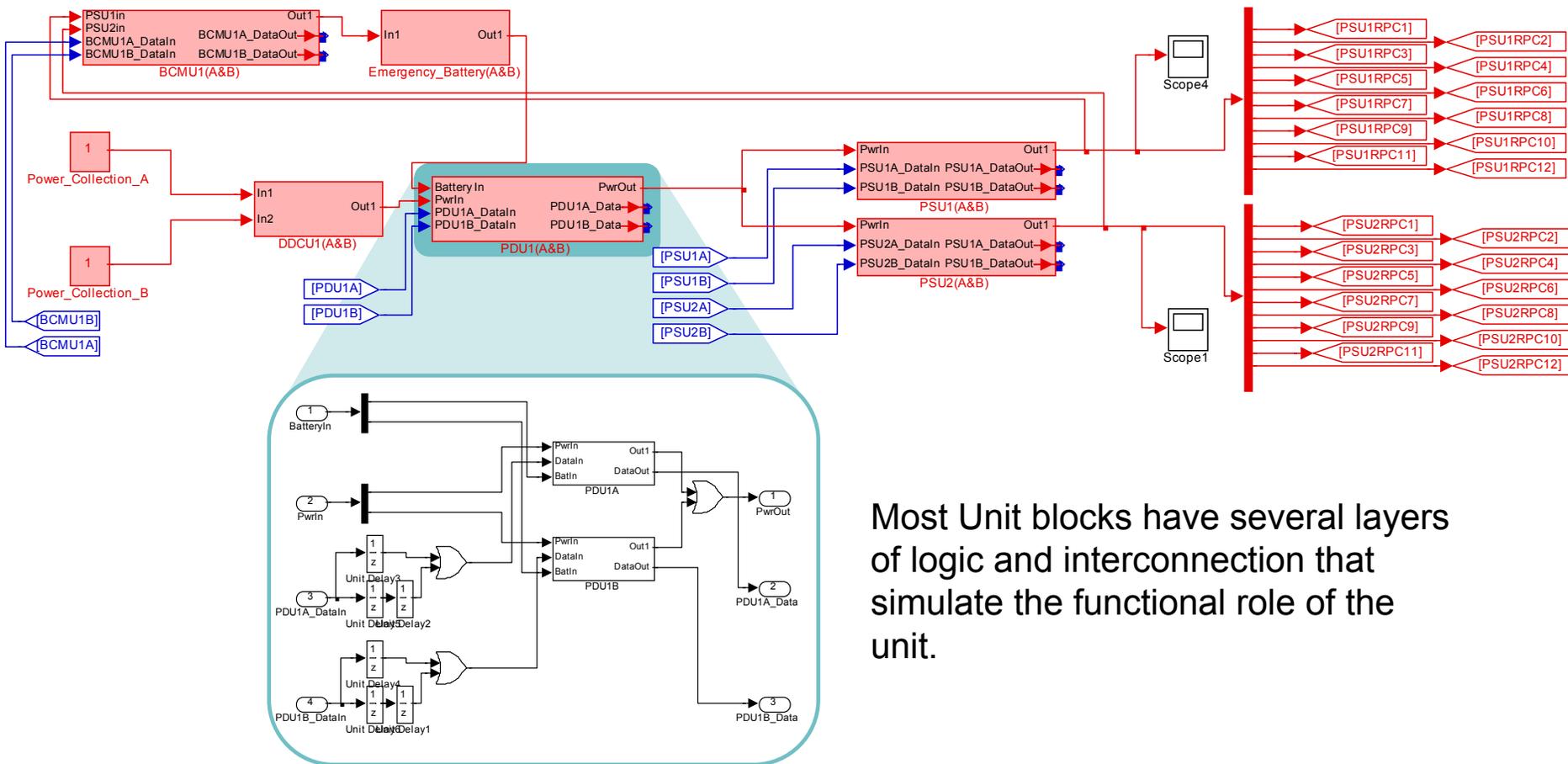
- We iterated on this diagram with NASA until it was clear we had accurately understood the intent of the design.
- The process of developing this diagram helped all parties better understand the system interactions.

Lunar Habitat Functional Diagram
 All Interfaces Overview



- From this functional diagram and the schematics we created a Simulink model of the separate systems:

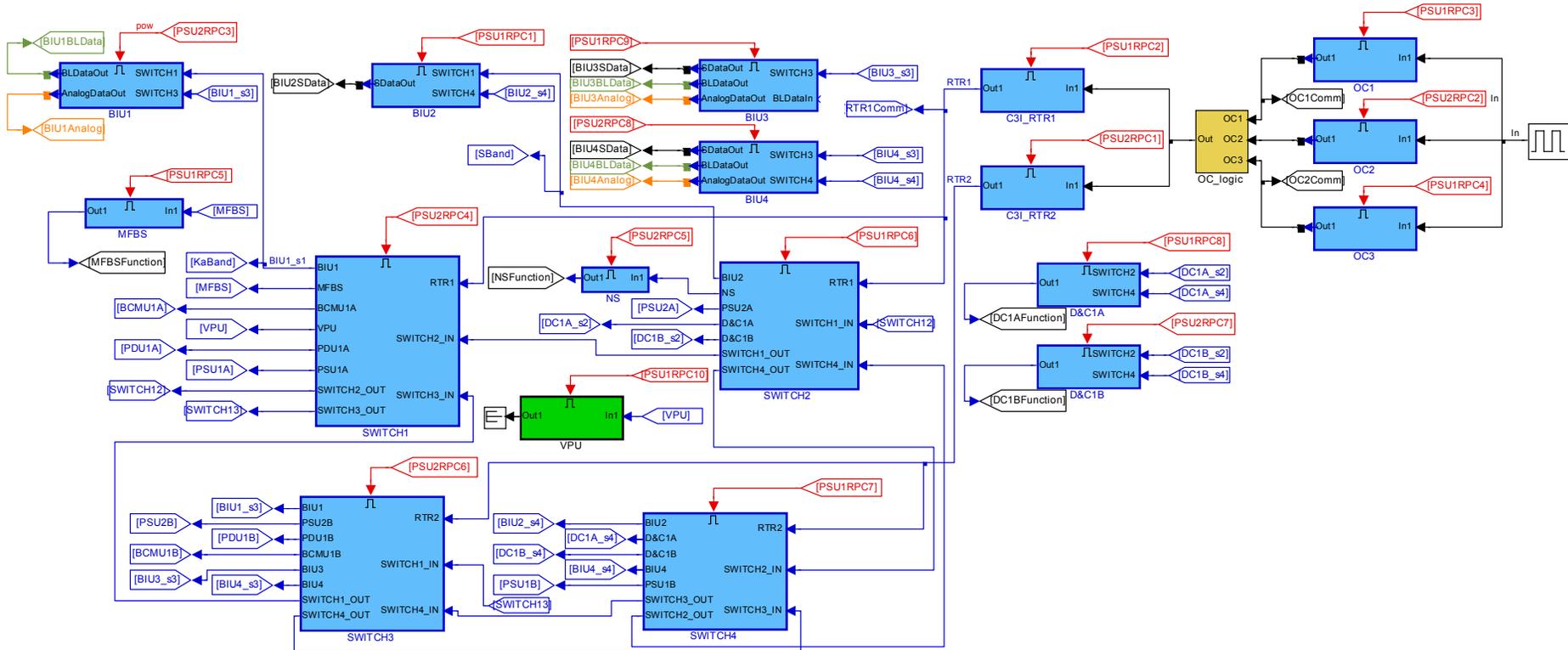
Power



Most Unit blocks have several layers of logic and interconnection that simulate the functional role of the unit.

- From this functional diagram and the schematics we created a Simulink model of the separate systems:

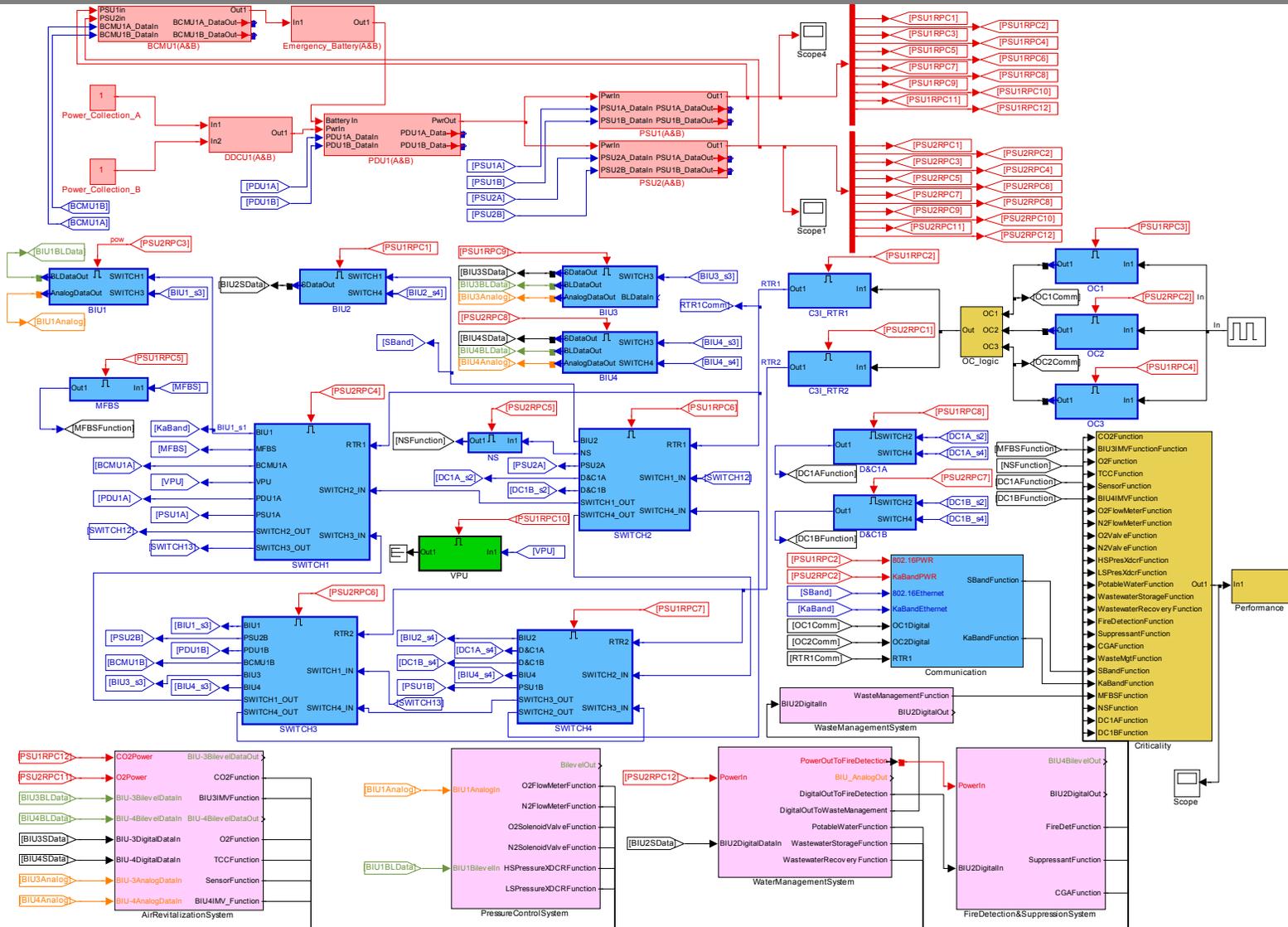
■ C&DH





Habitat Model Development

The Whole Thing





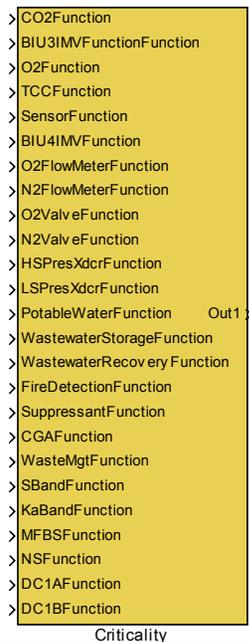
Habitat Model Development Benefits

- The full Simulink model of the Habitat system represents the first time that the entire system was assembled and run together.
 - It showed that there was a flow of data, power and commands through the system.
 - It highlighted some single point failure sources.

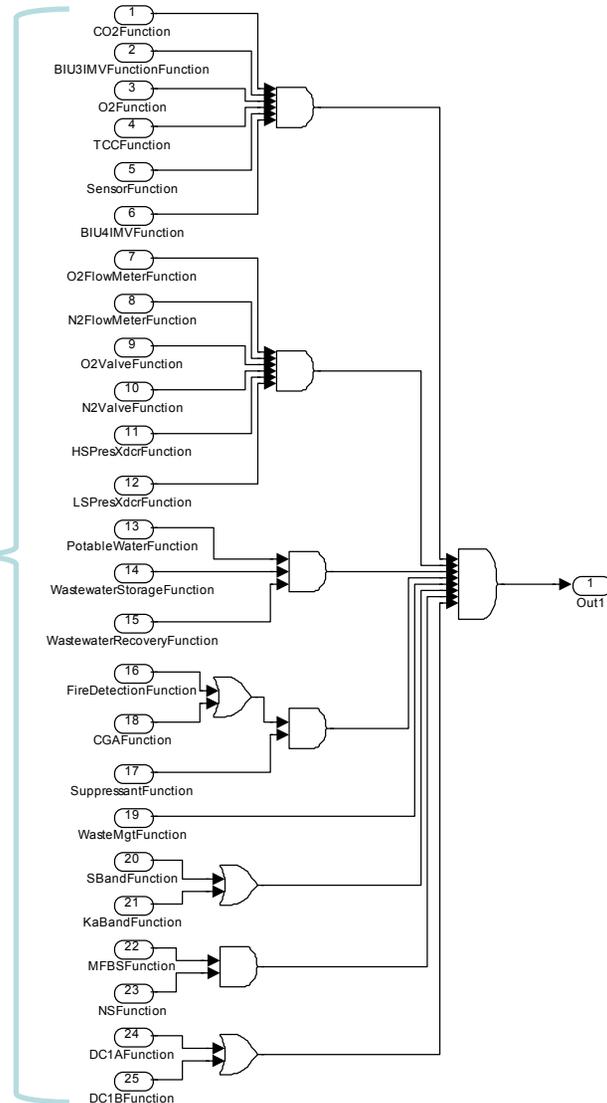


Habitat Model Function

- The input to the entire system are constant “1”s representing power from the solar panels and a square wave representing data and commands generated at the Operational Computers.
- Total model function is reduced to a single output through a “Criticality Evaluator”.



ie, this is the definition of “System is Operational”





Failure Rates

- Failure rates were provided by NASA down to the component level.
 - Based on previous experience (Shuttle, ISS, etc.)
- Most of these component failure rates were aggregated at the unit level.
 - Individual component failures can be explored, however this leads to a state explosion that can be difficult to deal with computationally.
 - Most unit level failures can be treated serially.
 - Any component failing within a unit is probably going to lead to total unit failure.



Model Driven Lunar Habitat Avionics Design

PRELIMINARY RESULTS



Testing the Model

- Evaluate the model down to the Bus Interface Unit (BIU) level with fixed uniform failure rates for Omission only.
 - Provides sanity check
 - Identifies single point failures to the Life Support and Communications Systems
 - Allows for a high level check of the system architecture
 - Where are the possible bottle-necks?
 - What are the possible principle drivers for system reliability?
- Examine model down to the BIU level with actual failure rates.
 - Allows for quick examination of alternatives
- Full model reliability assessment pending better definition of what components are required for successful system operation.



PARADyM Results

Uniform Failure Rate Results Down to BIU Level

- With failure rates for all units set to $1e-5$ (10000 hr MTBF) the overall Habitat Avionics reliability comes out between 0.8756 and 0.8929.
- The assumed $1e-5$ failure rate is a relatively high rate chosen to stress the system model

paradym

PARADyM Inputs

- LSS_Hab_Combined: Simulink model name (omit .mdl extension)
- 2200: Mission time for solving Markov model, hours
- 20: Evaluation time for solving dynamic simulation, seconds
- 5: Time for failure injection in dynamic simulation, seconds
- habResults: File name for final report (omit extension)
- Truncate evaluation: 3: Truncation level

Model Utilities

- Model testing interface
- Sensitivity Analysis

Model Execution and Analysis

- Run PARADyM
- View detailed results

Reliability for the current system is bounded between 0.8756172 and 0.8929388
 Unreliability for the current system is bounded between 0.1243828 and 0.1070612
 757 states took 696 seconds to evaluate

paradymResults

System Reliability by Failure Level

Failure Level	Reliability	System Loss	# States
0	0.516849	0	1
1	0.2874245	0.08052516	30
2	0.07134371	0.02653602	725
absorbing	0	0.01732157	1
total: LB	0.8756172	0.1243828	757
total: UB	0.8929388	0.1070612	

System Metrics and Plotting

Metric	LB(SS)	UB(SS)	LB(Tr)	UB(Tr)	Plot?
metric1	0.5	0.5	0.5	0.5	<input checked="" type="checkbox"/>

Omit nominal state from plot

Frequency data on log scale

10 # of histogram bins

Interactive Metrics Histogram

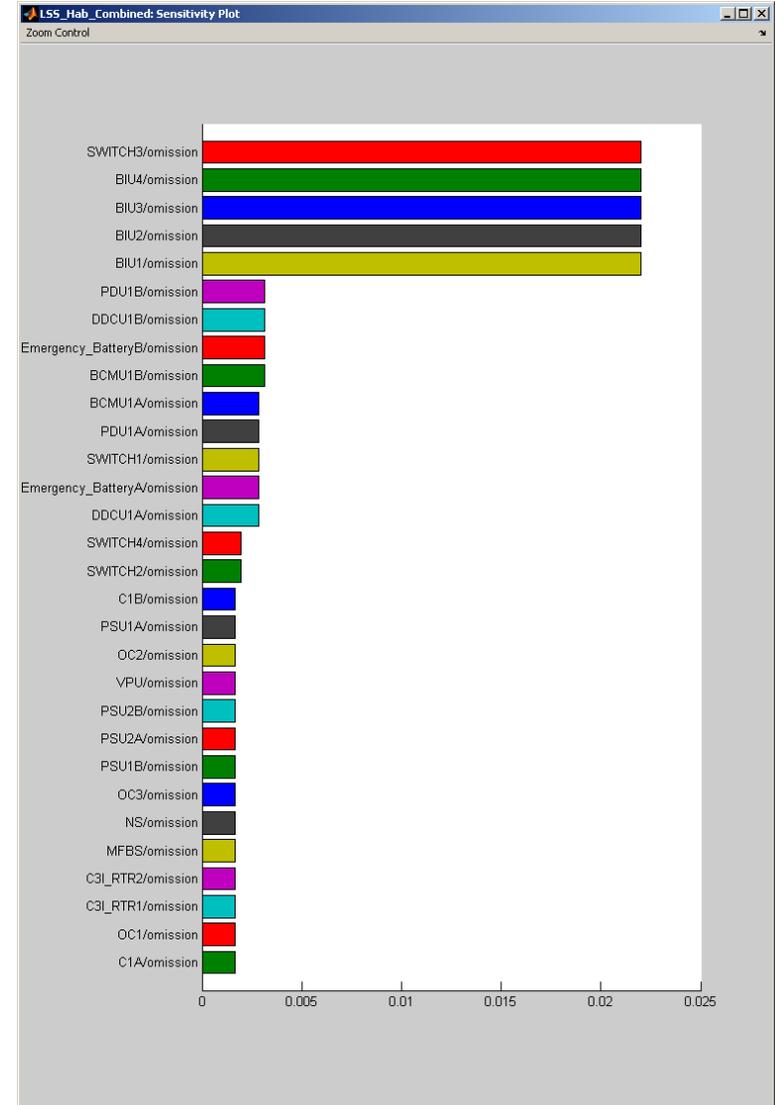
- More than twice as many possible states are still functioning after two failures as not.



Sensitivity Analysis

Uniform Failure Rate Results Down to BIU Level

- Sensitivity analysis shows that there is an architectural weakness at the Bus Interface Units (BIUs) and Switch 3.
 - In this example the model evaluator is treating the loss of any BIU as the loss of its attached System.
 - Switch 3 is the only one that is tied to BIU3.





Example Analysis

Using the Model to Improve the Design

Uniform Failure Rate Results Down to BIU Level

- A very quick and easy way to use the model and PARADyM is to evaluate the improvement in overall system reliability that can be obtained through the use of higher quality components.
- If the failure rates of the SPF sources (Switch 3 & BIUs) is improved from $1e-5$ to $1e-6$ the overall system reliability bounds go from 87.6%-89.3% to 96.7%-98.2%.

The screenshot shows the PARADyM software interface with the following sections:

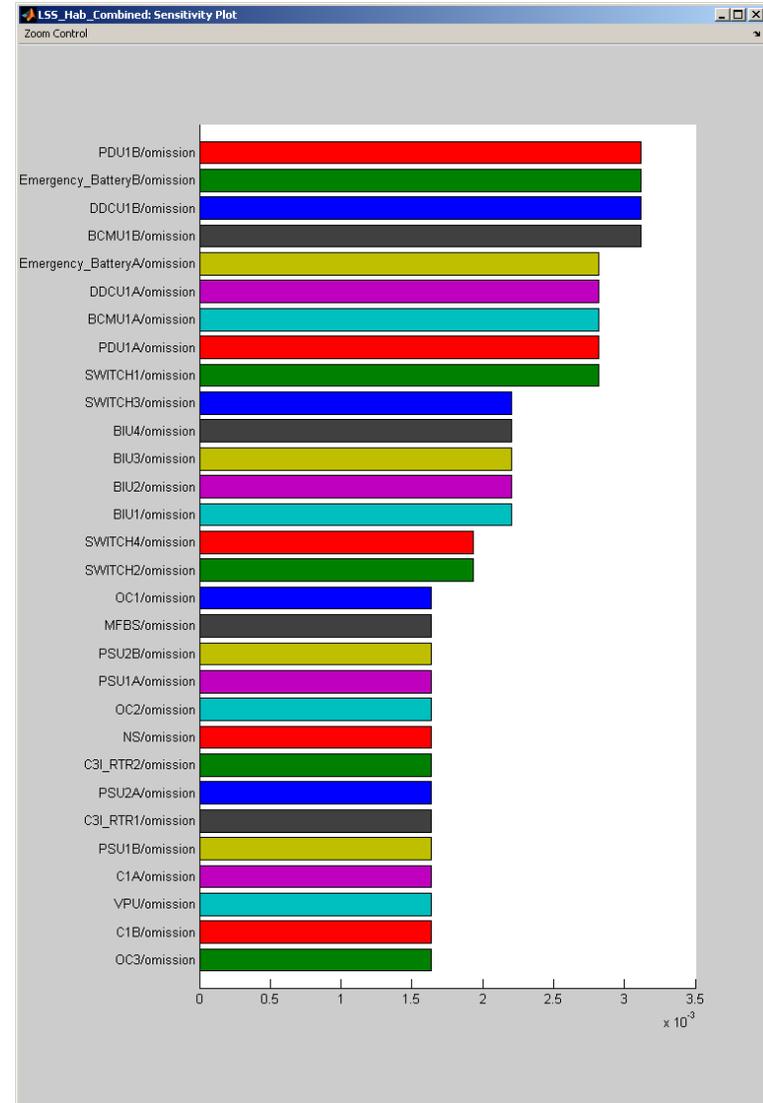
- PARADyM Inputs:**
 - Simulink model name (omit .mdl extension): LSS_Hab_Combined
 - Mission time for solving Markov model, hours: 2200
 - Evaluation time for solving dynamic simulation, seconds: 20
 - Time for failure injection in dynamic simulation, seconds: 5
 - File name for final report (omit extension): habResults
 - Truncate evaluation: (checked)
 - Truncation level: 3
 - Save as default:
- Model Utilities:**
 - Model testing interface:
 - Sensitivity Analysis:
- Model Execution and Analysis:**
 - Run PARADyM:
 - View detailed results:
 - Reliability for the current system is bounded between 0.9667394 and 0.9823618
 - Unreliability for the current system is bounded between 0.03326061 and 0.01763816
 - 757 states took 768 seconds to evaluate



Example Analysis Continued

Uniform Failure Rate Results

- Note that the revised Sensitivity Analysis now shows the BIUs and Switch 3 much lower on the sensitivity scale.
 - EPS components now drive the overall reliability.





Core C&DH and EPS with Actual Failure Rates

Down to Bus Interface Unit Level

- Analysis of the entire model, but with Life Support Systems and Communications treated as units with one failure block each.
- Failure rates from ISS/Shuttle experience
- System reliability between 0.93 and 0.95

paradym

PARADyM Inputs

LSS_Hab_FailureLimited Simulink model name (omit .mdl extension)

2200 Mission time for solving Markov model, hours

10 Evaluation time for solving dynamic simulation, seconds

5 Time for failure injection in dynamic simulation, seconds

exampleResults File name for final report (omit extension)

Truncate evaluation 3 Truncation level Save as default

Model Utilities

Model testing interface Sensitivity Analysis

Model Execution and Analysis

Run PARADyM View detailed results

Reliability for the current system is bounded between 0.9260881 and 0.948259

Unreliability for the current system is bounded between 0.07391188 and 0.05174102

757 states took 702 seconds to evaluate

paradymResults

System Reliability by Failure Level

Failure Level	Reliability	System Loss	# States
0	0.5079306	0	1
1	0.3246174	0.03675594	30
2	0.09354011	0.01498508	725
absorbing	0	0.02217086	1
total: LB	0.9260881	0.07391188	757
total: UB	0.948259	0.05174102	

System Metrics and Plotting

Metric	LB(SS)	UB(SS)	LB(Tr)	UB(Tr)	Plot?
metric1	0.5	0.5	0.5	0.5	<input checked="" type="checkbox"/>

Omit nominal state from plot Refresh Plot

Frequency data on log scale Enlarge Plot

10 # of histogram bins

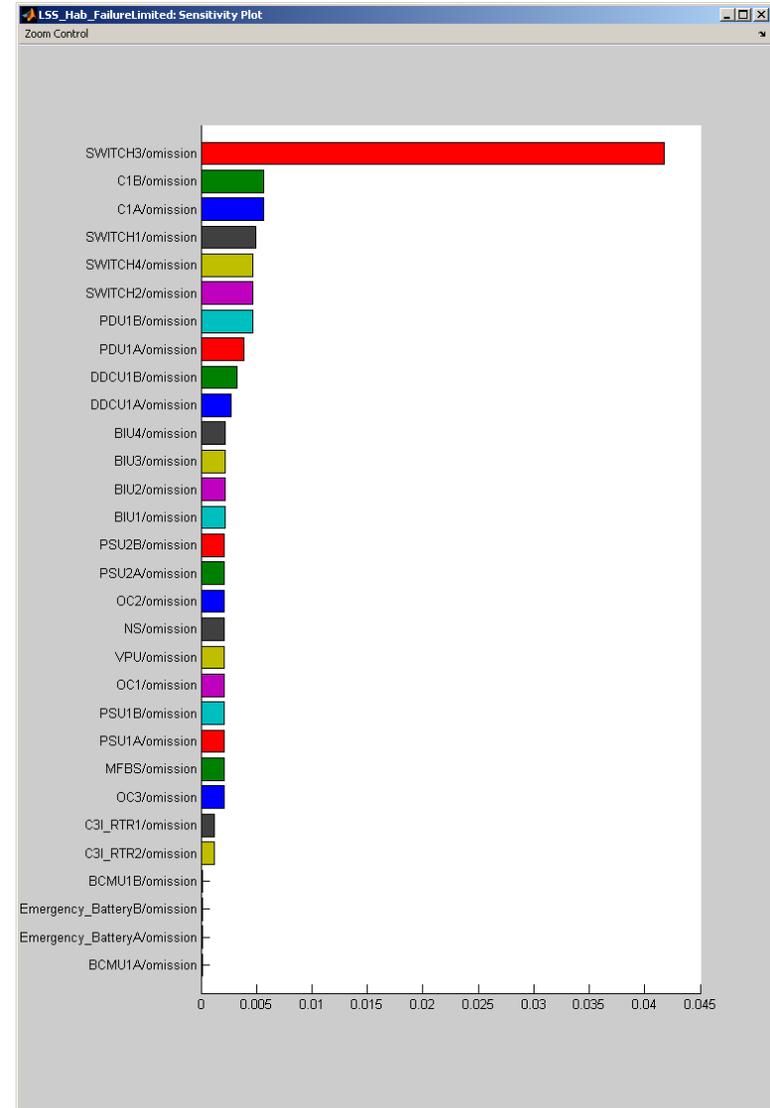
Interactive Metrics Histogram



Preliminary Analysis

Core C&DH and EPS with Actual Failure Rates

- Sensitivity Plot shows that the reliability of Switch 3 is driving over system reliability.
 - This is still the only connection to BIU3.
 - The other switches and the BIUs themselves, have much lower failure rates and so the design is less sensitive to changes in their failure rates.





Preliminary Analysis

Core C&DH and EPS with Actual Failure Rates

- The model was modified to examine the impact of an additional input source to BIU3 on overall reliability.
 - In addition to Switch 3, Switch 4 was used as an input for BIU3.
- This resulted in system reliability between 0.96 and 0.98

paradyM

PARADyM Inputs

LSS_Hab_FailureLimited Simulink model name (omit .mdl extension)

2200 Mission time for solving Markov model, hours

10 Evaluation time for solving dynamic simulation, seconds

5 Time for failure injection in dynamic simulation, seconds

exampleResults File name for final report (omit extension)

Truncate evaluation 3 Truncation level

Model Utilities

Model Execution and Analysis

Reliability for the current system is bounded between 0.958561 and 0.9832662

Unreliability for the current system is bounded between 0.04143904 and 0.01673383

786 states took 1.02e+003 seconds to evaluate

paradyMResults

System Reliability by Failure Level

Failure Level	Reliability	System Loss	# States
0	0.5079306	0	1
1	0.3462994	0.006392338	30
2	0.104331	0.01034149	754
absorbing	0	0.02470522	1
total: LB	0.958561	0.04143904	786
total: UB	0.9832662	0.01673383	

System Metrics and Plotting

Metric	LB(SS)	UB(SS)	LB(Tr)	UB(Tr)	Plot?
metric1	0.5	0.5	0.5	0.5	<input checked="" type="radio"/>

Omit nominal state from plot

Frequency data on log scale

10 # of histogram bins

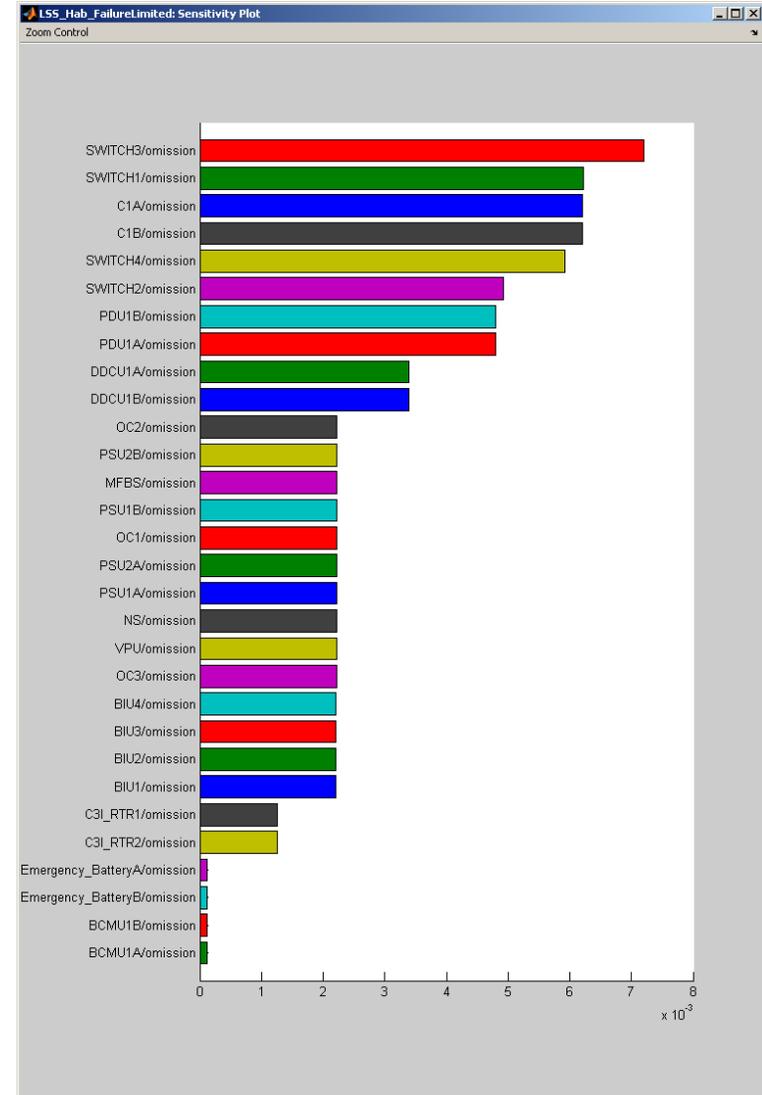
Interactive Metrics Histogram



Preliminary Analysis

Core C&DH and EPS with Actual Failure Rates

- For the revised architecture test case the sensitivity analysis shows that Switch 3 is now on a par with the other Switches and the EPS hardware in terms of impact on overall reliability
- This shows how we use the model and PARADyM to probe and improve the design.





Main Systems Modeling

- All of the main systems have been modeled.
 - There are issues related to the number of solenoid valves, their reliability, and how many are actually required for successful function of the Air Revitalization System.
 - Currently in the process of working through these in discussion with NASA to close on this part of the model.
 - Not able to generate a realistic reliability number for these systems or the full model until we close on this.



Summary

- NASA and Draper are using a systematic model driven process for evaluating the Lunar Habitat Avionics
 - This process makes use of automated tools which use the sensitivity of the system to changes in component reliability to probe the design look for opportunities for optimization.
 - Can be used to systematically evaluate options for future improvements and technology investments.
- This effort has already pointed out ways to improve the design and increase system reliability.
 - Identified and are in the process of removing single point failure sources.
- Next step is to apply this methodology to the LER.



Contacts

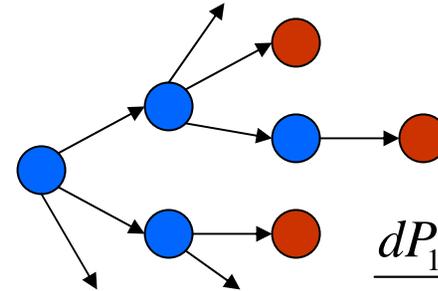
- John West, LSS Avionics Modeling Program Manager, C. S. Draper Laboratory, jwest@draper.com
- Ian Claypool PhD., LSS Modeling Technical Director, C. S. Draper Laboratory, iclaypool@draper.com
- Nick Borer PhD., System Design Engineer, C. S. Draper Laboratory, nborer@draper.com
- Ryan Odegard, Mission Design Staff Engineer, C. S. Draper Laboratory, rodegard@draper.com



Model Driven Lunar Habitat Avionics Design

BACKUP MATERIAL

- Steps to solving a Markov model:
 - Enumerate all of the possible states in the model
 - Quantify the failure rates of the individual components
 - Create a system of differential equations to describe the change of the probability of being in a given state with time
 - Quantify the system life
 - Solve the systems of equations to find the probabilities of being in any one state



$$\frac{dP_1}{dt} = -(a+b)P_1(t)$$

$$\frac{dP_2}{dt} = aP_1(t) - bP_2(t)$$

M

$$P_1(t) = e^{-(a+b)t}$$

$$P_2(t) = e^{-bt} - e^{-(a+b)t}$$

M



LSS Habitat Modeling Effort

From Task 270 Mod 1 SOW

- The cost-effective development of human space exploration systems that are highly-reliable, yet conform to strict mass and power constraints, requires a highly integrated systems engineering and design process.
 - This process relies on an understanding of the complex inter-connectivity and functional inter-dependency between subsystems
 - There is great value in constructing comprehensive functional system models and simulating overall system behavior in response to configuration changes and anomalies originating from either component failures or environmental variation.



LSS Habitat Modeling Effort

From Task 270 Mod 1 SOW

- One particular area of interest is the set of trade-offs between rad-hardened/space-qualified vs. COTS electronic components.
 - Robust performance can be achieved with highly reliable components specifically designed for deployment in harsh operating environments.
 - There are significant impacts to incorporating such components.
 - Bearing the full burden of acquiring hardware with limited, niche applications
 - Current high reliability electronics technology lags commercial technology by nearly a decade
- Alternatively, it may be possible for reliability requirements to be met with commercial components, assuming the use of more advanced architectures employing synchronized redundancy and voting algorithms to prevent failures from adversely affecting mission performance.
 - This approach has the potential to alleviate the impacts of using Space Qualified components, without giving up computing capability.
 - If such architectures were capable of dynamically re-tasking network processors for non-critical tasks when not needed for system redundancy, there is the potential to increase computing resources for science applications.

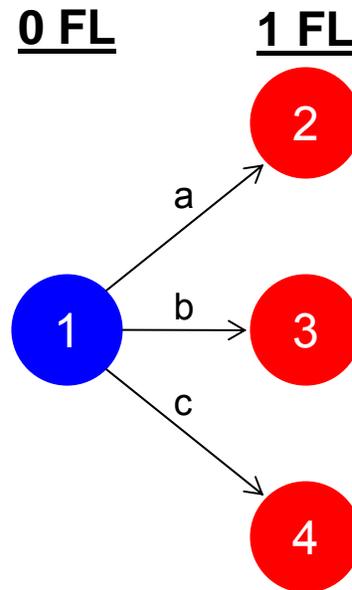
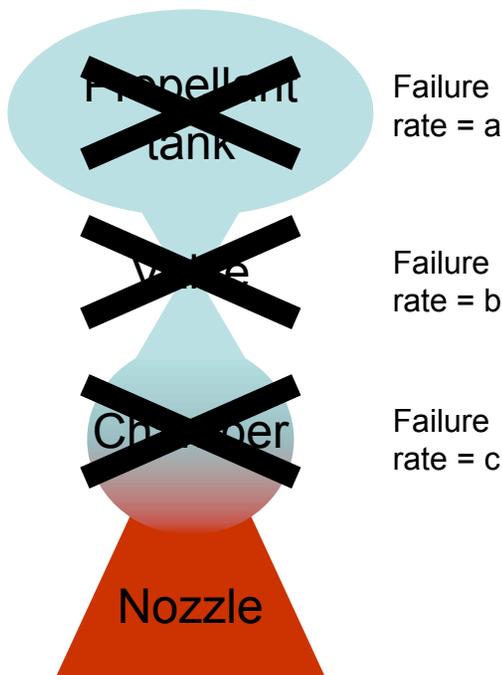


LSS Habitat Modeling Effort

From Task 270 Mod 1 SOW

- NASA and Draper's Objective: Develop a systems model of the lunar habitat.
- Draper is developing a model to be used to evaluate the lunar habitat integrated systems, including avionics, environmental control and life support (ECLS), thermal management, and power.
 - Using the latest conceptual design data available from NASA's Lunar Surface System Project,
- NASA and Draper will use the model in conjunction with Draper's PARADyM tool to analyze and evaluate system-level fault-tolerance and sensitivities to the reliabilities of various components (i.e. computers, sensors, O2 scrubbers, etc.)
 - The model and evaluation will be used to make recommendations for changes.
 - Design Optimization
 - The habitation model will be comprised of a single habitation element, the "core hab", for the purposes of modeling.

- Consider a single monopropellant thruster with only “omission” type failures (failure means fluid, etc. does not propagate)
- Failure rate = 1/MTBF (mean time between failures) = a, b, or c



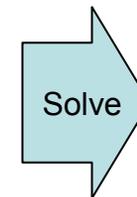
Blue indicates operational state
Red indicates system loss

$$\frac{dP_1}{dt} = -(a+b+c)P_1(t)$$

$$\frac{dP_2}{dt} = aP_1(t)$$

$$\frac{dP_3}{dt} = bP_1(t)$$

$$\frac{dP_4}{dt} = cP_1(t)$$



$$P_1(t) = e^{-(a+b+c)t}$$

$$P_2(t) = ae^{-(a+b+c)t}$$

$$P_3(t) = be^{-(a+b+c)t}$$

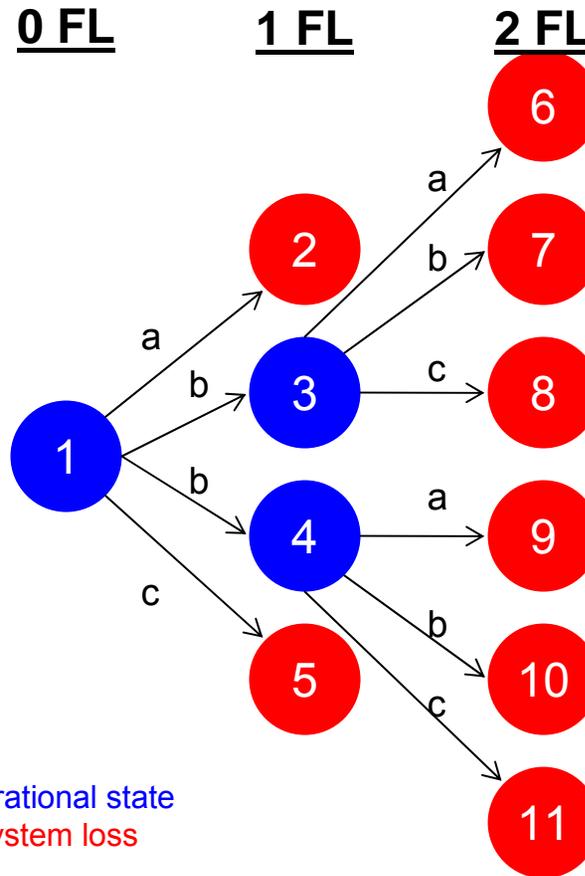
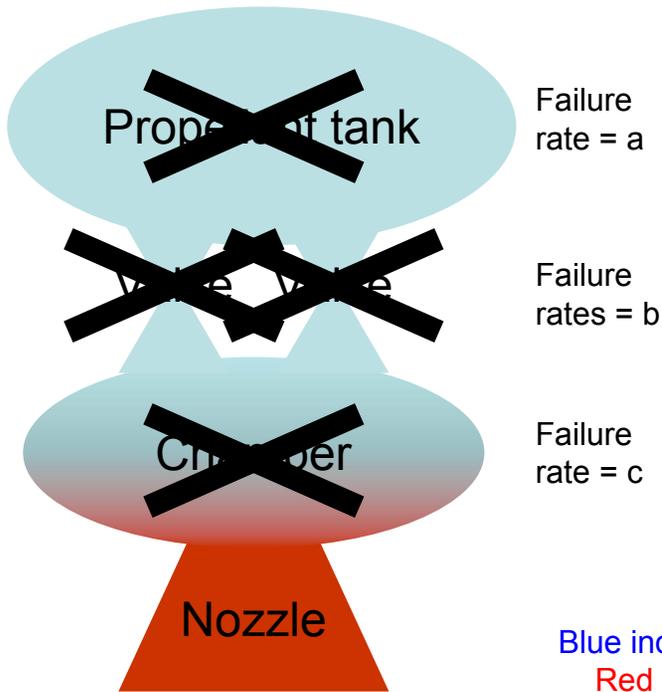
$$P_4(t) = ce^{-(a+b+c)t}$$

Can find probability of being in any failure configuration for any system life t

Probability of system loss = $\Sigma(\text{system loss states})$

Reliability = $\Sigma(\text{operational states})$

- When redundancy is added, the number of states to evaluate increases dramatically



Blue indicates operational state
 Red indicates system loss

It quickly becomes necessary to automate construction of and solution to the Markov model!



- It is relatively straightforward to automatically generate a Markov model and calculate the solution to large systems of ODEs
 - Create a state transition matrix
 - Numerically solve ODEs by stepping in time from initial condition (usually from the nominal state)
- State explosion for large models is still a problem
 - Truncation of Markov model (only build to n^{th} failure level)
 - Aggregation of individual states

$$\mathbf{A} = \begin{bmatrix} -(a+b) & 0 & 0 & \Lambda \\ a & -b & 0 & \Lambda \\ b & 0 & -a & \Lambda \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{O} \end{bmatrix}$$

$$\frac{dP(t)}{dt} = \mathbf{A}P(t)$$

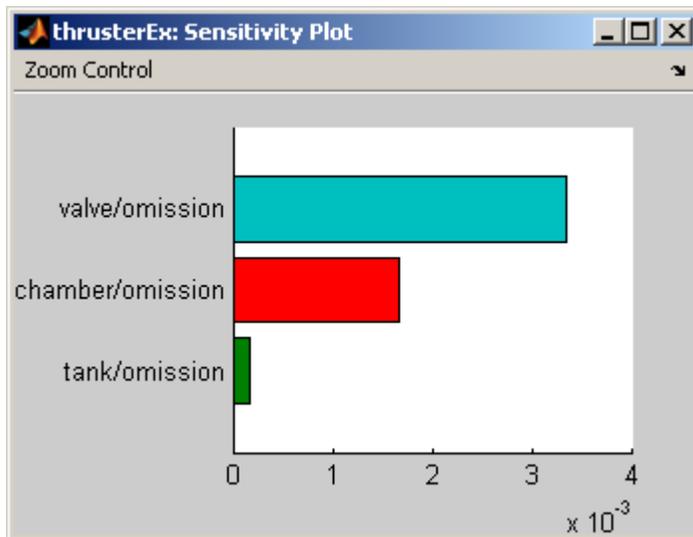
$$P(t + \Delta t) = (\mathbf{I} + \mathbf{A}\Delta t)P(t)$$



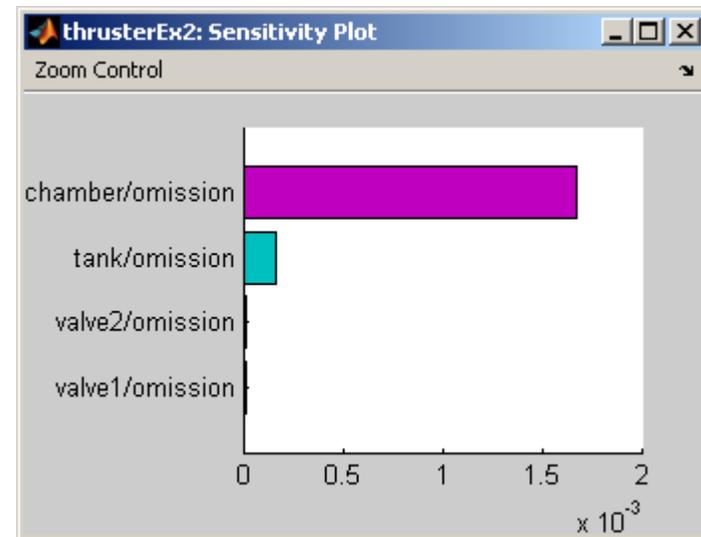
System Failure Sensitivity

What Components Drive the System Loss Probability?

- Comparing reliability numbers across candidate architectures provides little insight into how to best design or improve an existing design
- Instead, the component failure rates can be used to probe the design to determine the area of the architecture that causes the greatest change in system reliability

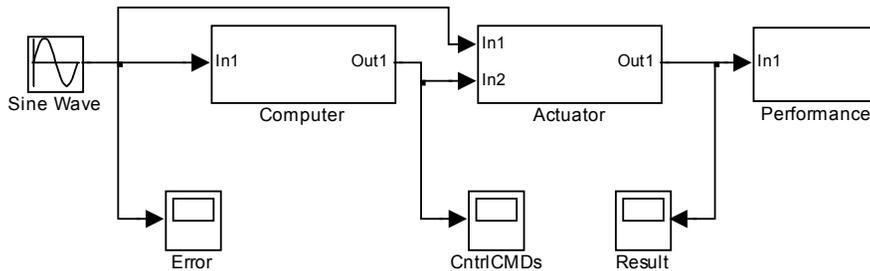
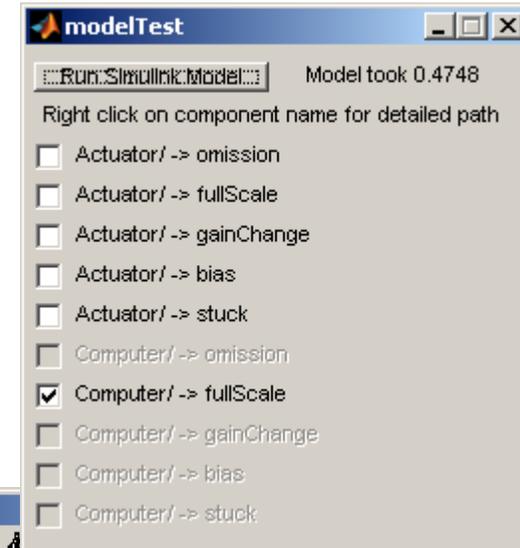


Change in reliability with 1% change in MTBF for single-valve thruster design



Change in reliability with 1% change in MTBF for redundant-valve thruster design

- The model testing interface allows for stepping through individual failures.
 - This can be used to troubleshoot and validate the modeling.
 - It is also informative in regard to the design being evaluated.

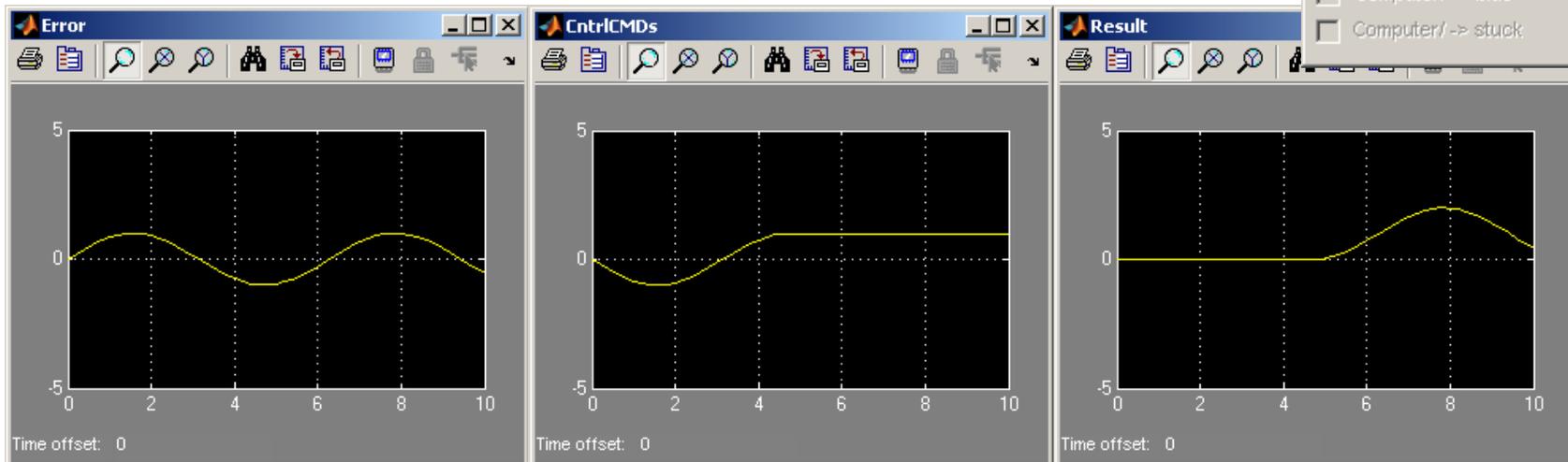



modelTest

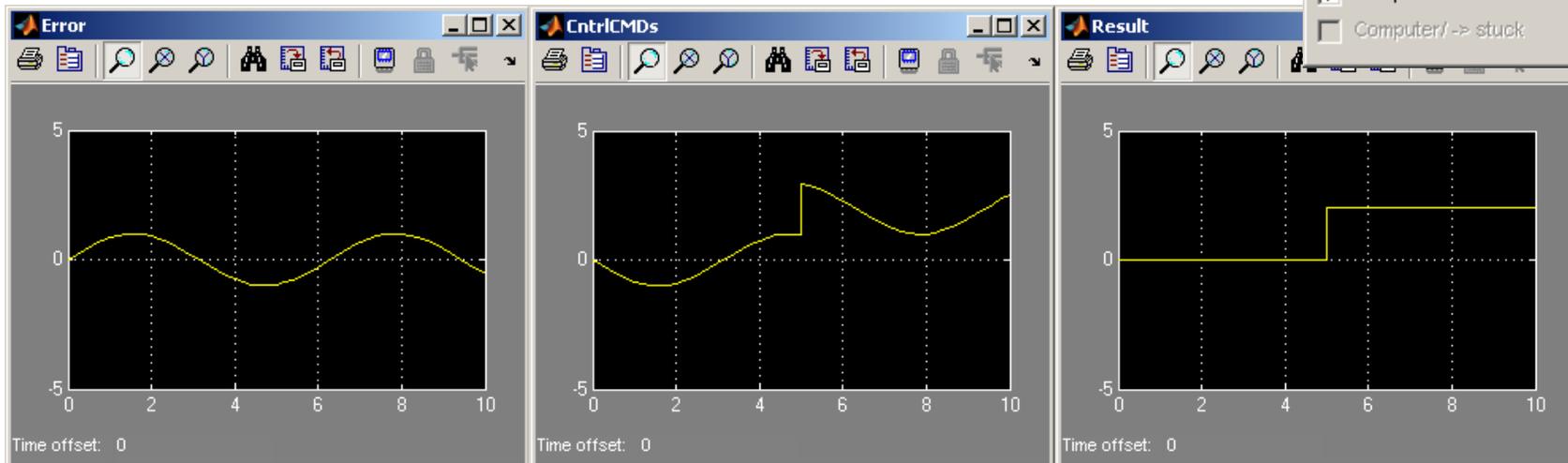
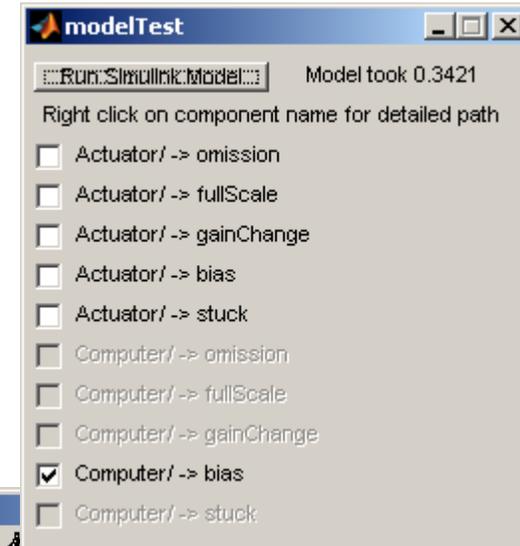
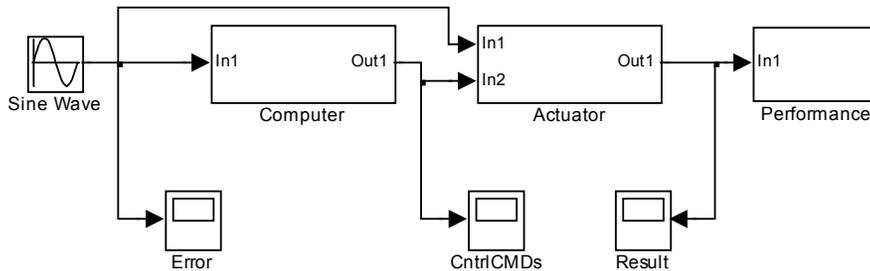
Run Simulink Model: Model took 0.4748

Right click on component name for detailed path

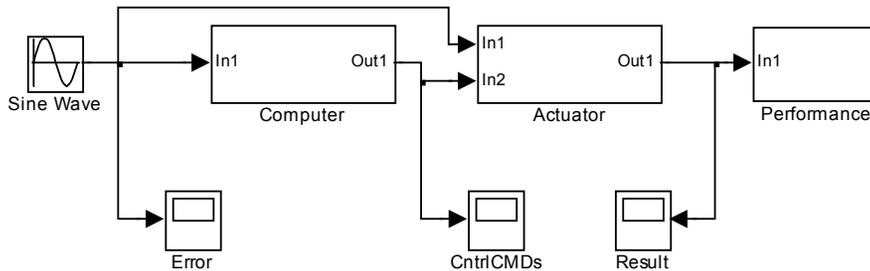
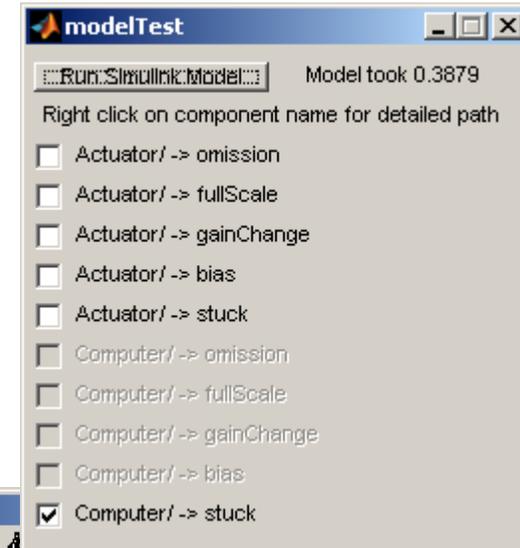
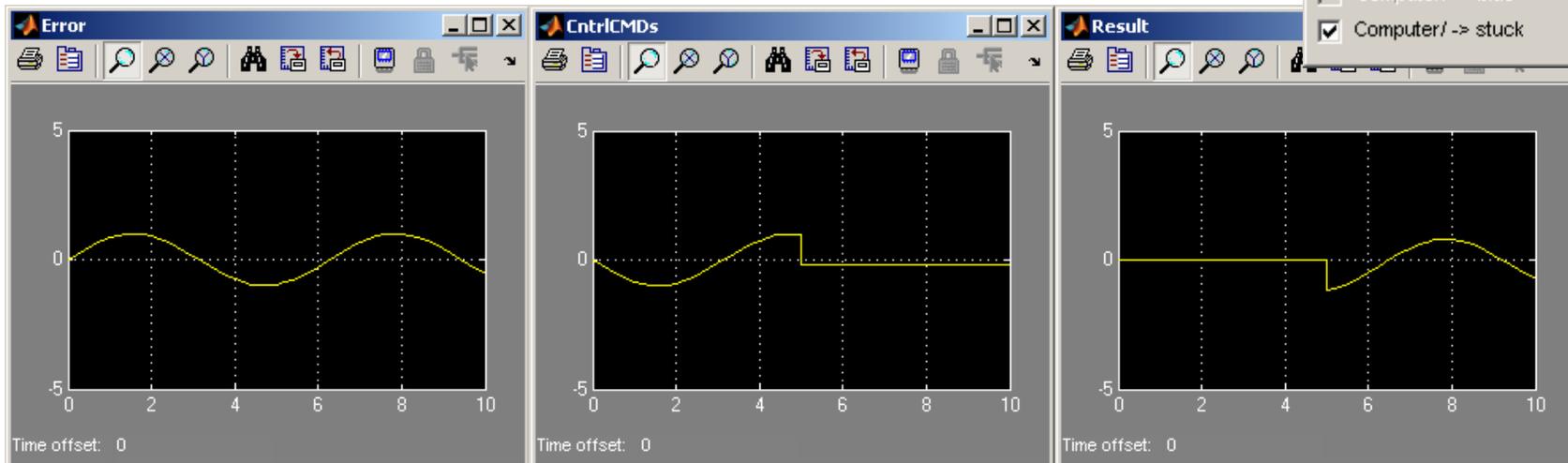
- Actuator / -> omission
- Actuator / -> fullScale
- Actuator / -> gainChange
- Actuator / -> bias
- Actuator / -> stuck
- Computer / -> omission
- Computer / -> fullScale
- Computer / -> gainChange
- Computer / -> bias
- Computer / -> stuck



- The model testing interface allows for stepping through individual failures.
 - This can be used to troubleshoot and validate the modeling.
 - It is also informative in regard to the design being evaluated.

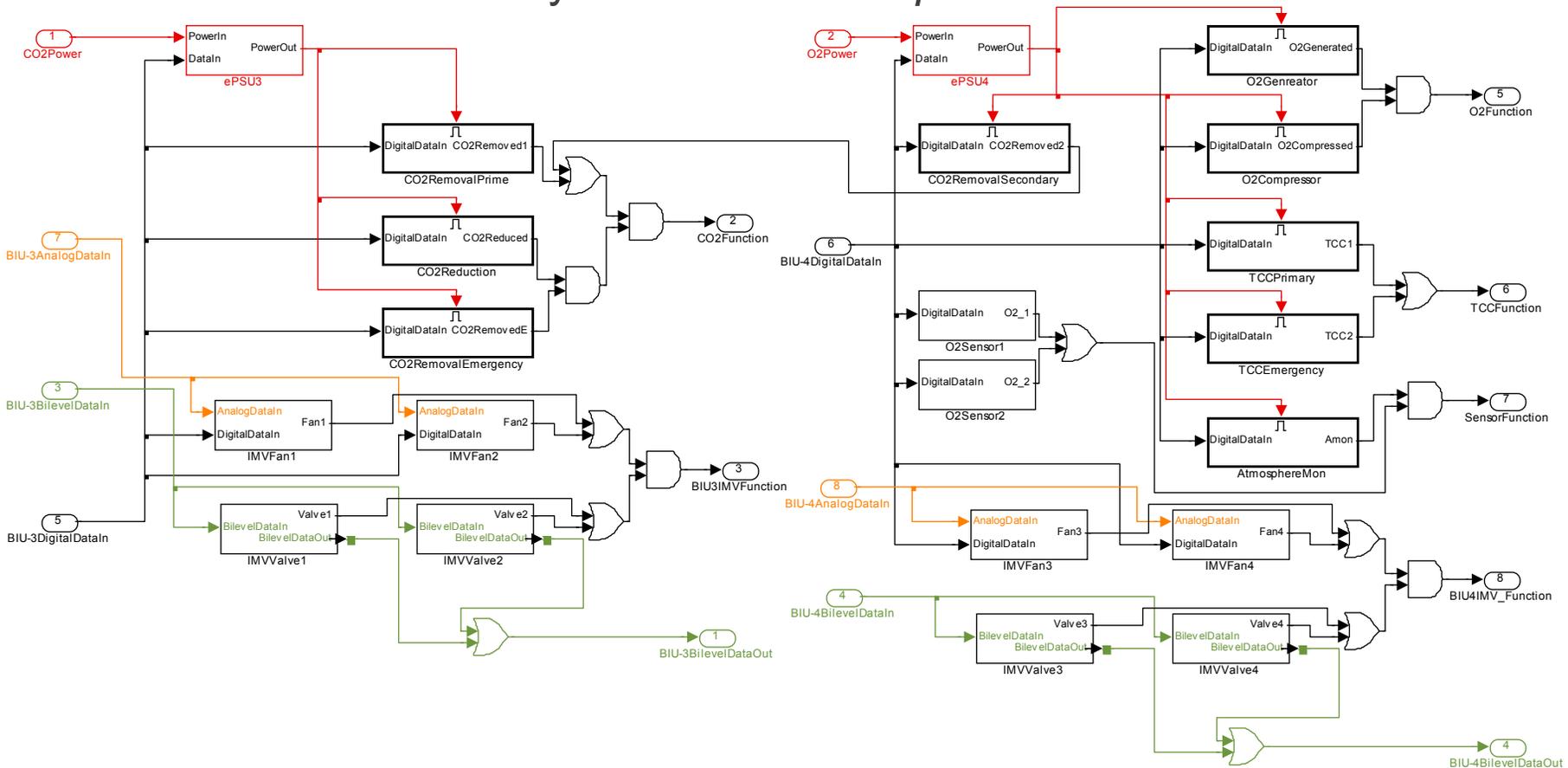


- The model testing interface allows for stepping through individual failures.
 - This can be used to troubleshoot and validate the modeling.
 - It is also informative in regard to the design being evaluated.

- From this functional diagram and the schematics we created a Simulink model of the separate systems:

■ Air Revitalization System – *In Development*

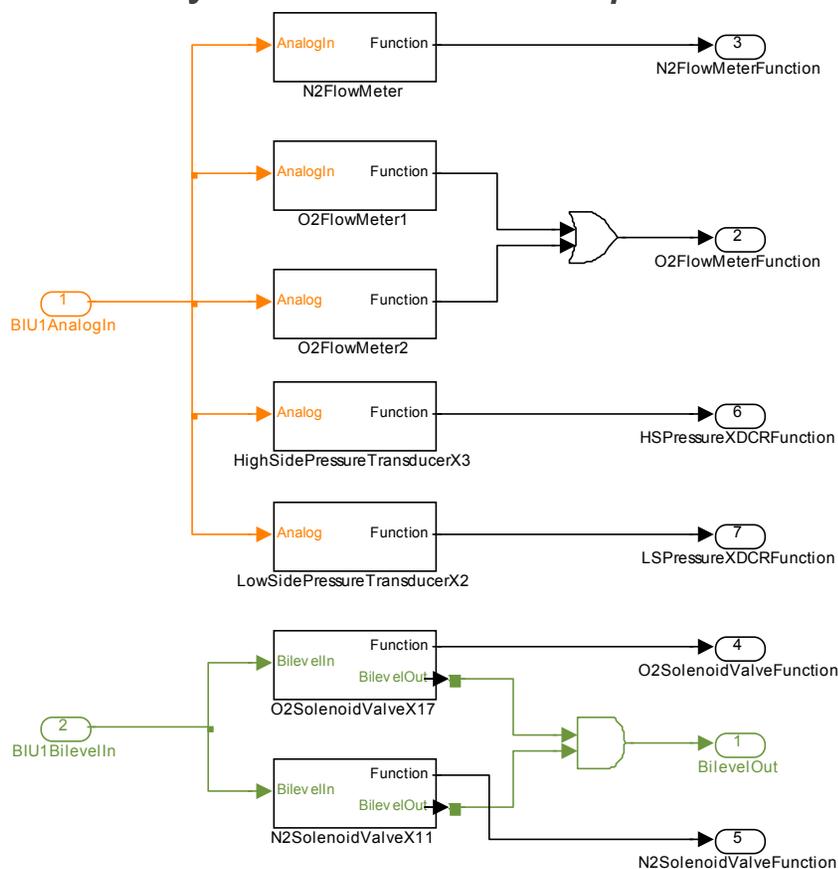




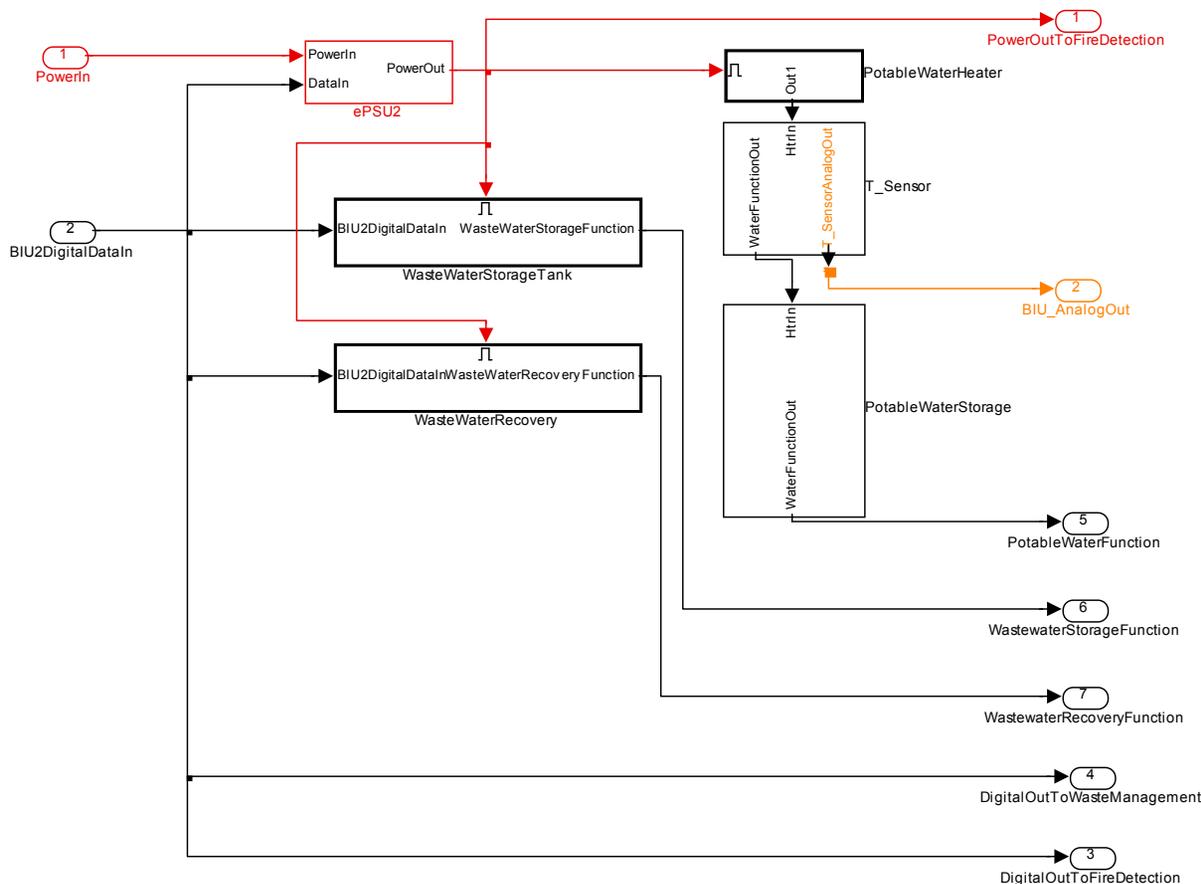
Habitat Model Development

- From this functional diagram and the schematics we created a Simulink model of the separate systems:

- Pressure Control System – *In Development***



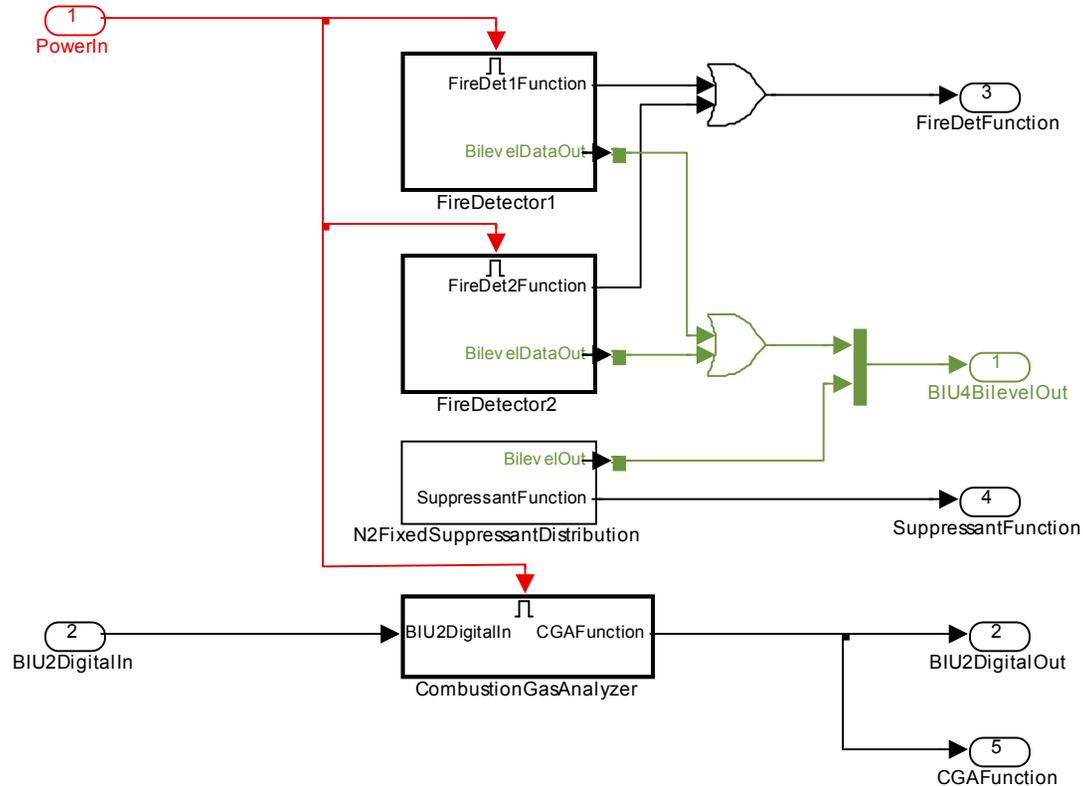
- From this functional diagram and the schematics we created a Simulink model of the separate systems:
 - Water Management System – *In Development*





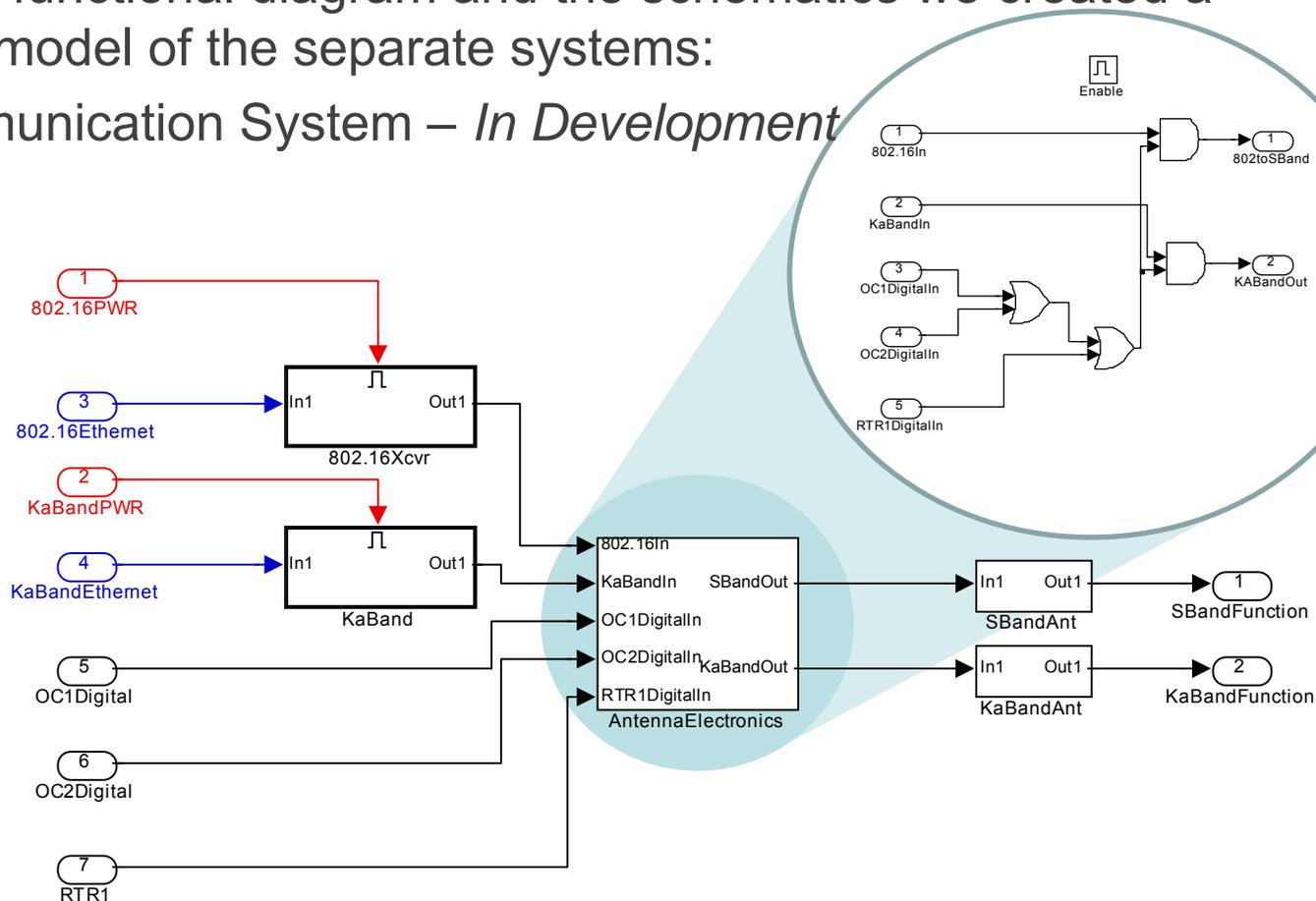
Habitat Model Development

- From this functional diagram and the schematics we created a Simulink model of the separate systems:
 - Fire Detection & Suppression System – *In Development*



- From this functional diagram and the schematics we created a Simulink model of the separate systems:

- Communication System – *In Development*





Habitat Model Development

- From this functional diagram and the schematics we created a Simulink model of the separate systems:
 - Waste Management System – *In Development*
 - PARADyM failure blocks are embedded in most subsystems at the unit level

