

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 1 of 102	

**System Architectural Considerations on Reliable Guidance,
Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

October 22, 2009

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 2 of 102	

Report Approval and Revision History

Approval and Document Revision History

NOTE: This document was approved at the October 22, 2009, NRB. This document was submitted to the NESC Director on December 8, 2009, for configuration control.

Approved Version:	<i>Original Signature on File</i>	12/9/09
1.0	NESC Director	Date

Version	Description of Revision	Office of Primary Responsibility	Effective Date
1.0	Initial Release	Cornelius Dennehy, NASA Technical Fellow for Guidance, Navigation, and Control	October 22, 2009

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 3 of 102	

Table of Contents

Volume I: Technical Assessment Report

1.0	Notification and Authorization	5
2.0	Signature Page	6
3.0	Team List	7
4.0	Executive Summary	8
5.0	Assessment Plan	11
5.1	Background.....	11
5.2	Scope.....	12
6.0	Problem Description, Proposed Solutions, and Risk Assessment	12
6.1	Description of the Problem.....	12
7.0	Data Analysis	13
7.1	The 2 x 2 GN&C System.....	13
7.2	Details on the 2 x 2 OPN Model.....	15
7.3	Details on the 3 x 2 OPN Model.....	19
7.4	Results.....	26
7.5	Concluding Remarks	39
8.0	Findings and Recommendations	40
8.1	Findings.....	40
8.2	Recommendations.....	41
9.0	Alternate Viewpoints	42
10.0	Other Deliverables	42
11.0	Lessons Learned	42
12.0	Definition of Terms	42
13.0	Acronyms List	43
14.0	References	44



**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
4 of 102

List of Figures

Figure 7.1-1. Three Possible 2 x 2 Systems 13
 Figure 7.2-1. 2 x 2 OPN Decision Tree 17
 Figure 7.3-1. Determination of Duplicate Architectures 20
 Figure 7.3-2. Finding Representative Architectures 21
 Figure 7.3-3. Using Representative Architectures to Find Duplicates 22
 Figure 7.3-4. Example of Manually-Drawn Architectures 23
 Figure 7.3-5. Example of Rule Based on a Manually Drawn Architecture 25
 Figure 7.4-1. Architecture Pareto Plots 28
 Figure 7.4-2. Architecture Pareto Plots 30
 Figure 7.4-3. Architecture Pareto Plots 31
 Figure 7.4-4. Architecture Pareto Plots 32
 Figure 7.4-5. The Architectures Described in Table 7.4-1 36
 Figure 7.4-6. The Architectures Described in Table 7.4-2 38

List of Tables

Table 7.1-1. Reliability Expressions for the 2 x 2 Systems in Figure 7.1-1 14
 Table 7.1-2. Indicator for the Channelized and Hybrid 2 x 2 Systems in Figure 7.1-1 14
 Table 7.2-1. Component Properties For Sensor Types A, B, and C 18
 Table 7.2-2. Component Properties For Computer Types A, B, and C 18
 Table 7.2-3. Connection Reliabilities, Connection Weights, and Dissimilar Component Penalties for Each OPN Scenario Run 19
 Table 7.4-1. Reliabilities and Weights for the 1- through 9- Connection Architectures Closest the Utopia Point 35
 Table 7.4-2. Reliabilities and Weights for the 1- through 9- Connection Architectures Closest the Utopia Point 37

Volume II: Appendices

Appendix A. A Comparison of GN&C Architectural Approaches for Robotic and Human-Rated Spacecraft 46
 Appendix B. A Comparison of Fault-Tolerant GN&C System Architectures Using the Object Process Network (OPN) Modeling Language 61
 Appendix C. Study Summary Presentation 83

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 5 of 102	

Volume I: Technical Assessment Report

1.0 Notification and Authorization

A NASA Engineering and Safety Center (NESC) out-of-board assessment was approved on October 5, 2006. Mr. Cornelius Dennehy, NASA Technical Fellow for Guidance, Navigation and Control (GN&C) was selected to lead this assessment. The Assessment Plan was presented and approved by the NESC Review Board (NRB) on April 19, 2007. The Study Kickoff meeting was held at Massachusetts Institute of Technology (MIT) on April 20, 2007. Interim research project progress and status reviews were held at MIT in July 2007, in December 2007 and in March 2008. An Initial Summary was presented to the NRB on December 19, 2008. The Final Report was presented for approval to the NRB on October 22, 2009.

The key stakeholders for this assessment are Mr. Frank Bauer, Chief Engineer for the Exploration Systems Mission Directorate (ESMD); Dr. Brian Muirhead, Constellation Program (CxP) Chief System Engineer; Mr. Howard Hu, Orion Crew Exploration Vehicle (CEV) GN&C lead; Mr. Scott Tamblyn, CEV GN&C Engineering; and Mr. William Othon, CEV GN&C Engineering.

The NESC, MIT, and Draper Laboratory team performed an independent and systematic study on the problem of optimizing the reliability of GN&C architectures with common avionic units.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 6 of 102	

2.0 Signature Page

Submitted by:

Mr. Cornelius J. Dennehy Date

Significant Contributors:

Dr. Gregor Z. Hanuschak Date

Dr. Nicholas A. Harrison Date

Dr. Edward F. Crawley Date

Mr. John J. West Date

Dr. Steven R. Hall Date

Dr. Alejandro D. Domínguez-García Date

Signatories declare the findings and observations compiled in the report are factually based from data extracted from Program/Project documents, contractor reports, and open literature, and/or generated from independently conducted tests, analysis, and inspections.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 7 of 102	

3.0 Team List

Name	Discipline	Organization/Location
Core Team		
Cornelius Dennehy	NESC Lead/NASA Technical Fellow for Guidance, Navigation, and Control	NESC/GSFC
John West	Fault Tolerant Space Systems	Draper Laboratory
Study Team		
Edward Crawley	MIT Study Lead (System Engineering)	MIT Aeronautics and Astronautics Engineering Department
Steven Hall	MIT Study Co-Lead (GN&C Avionics Systems)	MIT Aeronautics and Astronautics Engineering Department
Gregor Hanuschak	Fault Tolerance and System Reliability	MIT Aeronautics and Astronautics Engineering Department
Nicholas. Harrison	Fault Tolerant GN&C Systems	Draper Laboratory
Alejandro Domínguez-García	Fault Tolerance and System Reliability	University of Illinois at Urbana- Champaign
Administrative Support		
Roy Savage	MTSO Program Analyst	LaRC
Donna Gilchrist	Planning and Control Analyst	ATK, LaRC
Christina Cooper	Technical Writer	ATK, LaRC

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 8 of 102	

4.0 Executive Summary

This final report summarizes the results of a comparative assessment of the fault tolerance and reliability of different Guidance, Navigation and Control (GN&C) architectural approaches. This study was proactively performed by a combined Massachusetts Institute of Technology (MIT) and Draper Laboratory team as a GN&C “Discipline-Advancing” activity sponsored by the NASA Engineering and Safety Center (NESC). This systematic comparative assessment of GN&C system architectural approaches was undertaken as a fundamental step towards understanding the opportunities for, and limitations of, architecting highly reliable and fault tolerant GN&C systems composed of common avionic components. The primary goal of this study was to obtain architectural ‘rules of thumb’ that could positively influence future designs in the direction of an optimized (i.e., most reliable and cost-efficient) GN&C system. A secondary goal was to demonstrate the application and the utility of a systematic modeling approach that maps the entire possible architecture solution space.

The NESC team implemented a systematic approach for modeling, enumerating, and comparing simplified GN&C architectures using basic metrics. GN&C systems were decomposed into simple ‘building block’ subunits of Sensors, Computers, and Actuators, and various forms of subunit interconnection were defined for investigation. The resulting subunit/interconnection construct was used as a top-level abstraction for building candidate GN&C system architectures. This model was implemented using MIT’s Object Process Network (OPN) modeling language to more easily enumerate possible architectures, and ultimately identify which of these architectures have optimal properties. Dual and triple redundant GN&C system architectures, employing different reliability classes of components, were modeled using the OPN language. For the purpose of simplicity, it will be assumed that there are only three different types of GN&C avionic components possible for each component class. For example, the model incorporated three different types of GN&C Sensor components generically labeled Type A, Type B, and Type C, but understood to be representative of a low reliability/lightweight/low accuracy Sun Sensor; a medium reliability/medium weight/low-to-medium accuracy star tracker; and a high reliability/high weight/high accuracy Inertial Measurement Unit (IMU). The team assessed the avionic components typically used to implement recent space system GN&C architectures. Based upon this assessment, the team made a critical modeling assumption that the more reliable GN&C components tended to be heavier, more costly, and/or more complex. The team realized there are other modeling assumptions that could have been made, such as the GN&C avionic component with the smallest part count is a more reliable (and probably lower mass) unit than one with a higher part count. However, that alternate model construct did not fit either the team’s desire to keep the model simple and tractable, or the attributes of the GN&C avionic component inventory/technology base.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 9 of 102	

For the purposes of this study, reliability is defined as the probability that a given item will perform its intended function for a given time under a given set of flight conditions. This is most often expressed in terms of an average or mean life, such as Mean Time Between Failure (MTBF). Individual component reliabilities are dependent upon the component failure rate and desired operational time. The reliability relationships used in the model were exponential and governed by the equation $R = e^{-\lambda t}$, where λ is the failure rate and t is the operational time. Operational time is a user-defined parameter and based on the length of the space exploration mission. An operational time of $t = 10$ years was assumed in the models developed. It should be noted that the model assumed perfect coverage – 100-percent – accuracy in detecting and isolating a failure.

All enumerated architectures were evaluated based on two specific metrics: reliability and weight. In this study, the team elected to assume weight as a surrogate indicator of system cost and complexity. The team acknowledges that in some cases system cost can be driven by the complexity factor alone. However to keep the model simple for this top-level study, the assumption made was that weight is a good first-order approximation for system complexity and cost.

The team used an interconnection construct as a top-level abstraction for building a preliminary model of GN&C system architectures. This model was implemented using the OPN modeling language in order to more easily enumerate possible architectures and ultimately identify which architectures have optimal properties. Partial 2 x 2 systems (i.e., systems with up to dual redundancy per component class for two component classes) and 3 x 2 systems (i.e., systems with up to triple redundancy per component class for two component classes) were modeled in OPN. Within the constraints of these models, all possible architectures were rigorously enumerated and the weight/reliability trade-offs of cross-strapping components and using more than one type of component was assessed.

It was found that more reliable components are only beneficial in single string systems, or systems with single point failures. The key finding of this study was that most optimal GN&C system architectures employing component redundancy can be produced from generic connections and the least reliable type of avionic component from each component class. The analysis of the identified optimal architectures show that it is possible to produce nearly all potentially optimal architectures using only the Type A light weight/low-reliability Sensors, Type A light weight/low-reliability Computers, and generic connections. The identified optimal architectures reveal a preference to increase GN&C system redundancy of lighter, less reliable components rather than using smaller numbers of more reliable, heavy components.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 10 of 102	

The NESC team concluded there is merit in mapping the entire possible system architecture solution space. By showing where similar classes of solutions fall within the entire set, it allows one to quantitatively see how certain solution features affect Figure-of-Merit (FOM) performance and derive architecture ‘rules of thumb’. It also allows one to see the optimal system architecture solution boundary (the Pareto Front) and understand how one FOM can be exchanged for another.

The approach documented in this report provides insight into a potential limitation of the ‘Minimum Functionality/Minimum Implementation’ system architecting methodology also known as the ‘Iterative Risk Driven Design Approach’ as described in [ref.6] which uses as its starting point a single-string non-redundant system architecture. This method involves performing iterative trades to improve system safety until the mass margin is exhausted. It is not apparent that this stepwise optimization of a single design can ever achieve the boundary of optimal solutions (the Pareto Front). Even if it somehow did reach the optimal boundary, it is not clear the system architects will have access to and be able to visualize the whole range of optimal solutions. System architects using the ‘Minimum Functionality/Minimum Implementation’ approach should be aware of the technique described in this report and consider using it for comparison.

The team also concluded that with some enhancements, the systematic GN&C/Avionics “building block” OPN modeling techniques employed would serve as an excellent tool for evaluating competing GN&C system architectures for future spacecraft. This OPN-based approach, or other similar modeling tools, would perform the extremely useful up-front function of identifying the most attractive (lowest weight and overall “cost”) GN&C architectural options that satisfy a prescribed set of spacecraft fault tolerance, reliability, and performance requirements. Although less likely, but worth observing, is that such “building block” models could be used to identify the optimal (highest reliability/lowest weight) architectural options for a prescribed number and configuration of connections between adjacent GN&C components.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 11 of 102	

5.0 Assessment Plan

5.1 Background

Historically, the United States (US) human spaceflight programs have had thorough GN&C analysis and design early in their lifecycles, evaluating various GN&C architectural concepts within the trade space of their individual mission goals, constraints, and risk postures. The selected GN&C architectures have been tailored for each program, which is not surprising given the very different mission concepts. The Constellation Program (CxP) will be no different, but top level program requirements drivers for reliability and affordability may flow down to influence GN&C architectural considerations, such as fault tolerance, in ways never before encountered in the US space program.

Accomplishing the objectives of the CxP requires reliable GN&C for multiple spacecraft, both crewed and robotic. The up-front “architecting-in” of reliability is an integral part of the early steps of the GN&C Systems Engineering process. Substandard architectures may not only be unreliable but are typically difficult to fabricate, test, operate, support, service, and upgrade. These architectures can also be prohibitively costly to adapt to evolving mission scenarios as the system lifecycle extends beyond the anticipated time frame of service.

The operators of systems with substandard architectures can have protracted development schedules and high recurring operational costs as a result of not fully informed design decisions made early in a project’s development cycle. Therefore, it was important that a major system development project not prematurely shift its focus to the challenges of implementation before fully defining the appropriate architecture. There is benefit to allocating the time and devoting sufficient attention to defining the optimum system architecture over the lifecycle by producing the maximum return for a given level of risk and resources.

With some of these considerations in mind a comparative assessment activity focused on investigating the fault tolerance and reliability trades between different GN&C architectural approaches was formulated by the NASA Technical Fellow for GN&C. This led to a proactive study being performed by a combined MIT and Draper Laboratory team as a GN&C “Discipline-Advancing” activity sponsored by the NESC. The motivation for performing this study was the observation, both on the part of NESC and MIT, that GN&C systems for exploration prominently stand out among all the future spacecraft systems, as a potential “sweet spot” area where having flight hardware commonality might be of greatest benefit. This systematic comparative assessment of GN&C system architectural approaches was undertaken as a fundamental step towards understanding the opportunities for and limitations of architecting highly reliable and fault tolerant GN&C systems composed of common GN&C avionic

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 12 of 102	

components. The primary goal of this study was to obtain architectural ‘rules of thumb’ which could positively influence and drive future designs in the direction of the most reliable and cost-efficient GN&C systems possible. A secondary goal was to demonstrate the application and utility of a systematic modeling approach that maps the entire possible architecture solution space.

5.2 Scope

The proposed work was a systematic GN&C architecture comparative study performed by an integrated MIT/Draper Laboratory study team under the leadership of Dr. Edward Crawley of the MIT Aeronautics and Astronautics Engineering Department.

This study leveraged analytical methods developed at MIT as part of their program in Technical System Architecture, and their specialized analysis tools/methods used to support the NASA Exploration Systems Mission Directorate (ESMD) Concept Exploration and Refinement (CE&R) study. The MIT tools and methods were extended and applied to the problem of optimum GN&C system architectures. The existing MIT systematic architecture analysis capability, under Dr. Edward Crawley, was used to execute this task. Dr. Steven Hall, formerly of the Draper Laboratory, actively participated in this assessment as the Co-Lead and as the principal researcher on the assessment and evaluation of reliable GN&C avionic architectures.

MIT’s capabilities in system architecting methods were grounded in long-term research studies and benchmarking of best practice in space, automotive, electronics, and oil exploration industries. Historical work included methods developed in support of NASA’s Advanced Planning and Integration office, the creation and continuous refinement of a graduate-level class in Technical System Architecture, and through participation in multiple previous studies supporting NASA’s ESMD.

6.0 Problem Description, Proposed Solutions, and Risk Assessment

6.1 Description of the Problem

Sensors, Computers, and Actuators will be defined as “component classes”. The terminology “I x J OPN model” will be used to describe a model with up to “I” redundancy per component class and up to “J” component classes. For example, J = 2 could designate a model which only has Sensors or which has Sensors and Computers. J = 3 could designate a model with: Sensors; Sensors and Computers; or Sensors, Computers, and Actuators. If J = 3 and I = 2, this could designate a system with up to 2 Sensors, 2 Computers, and 2 Actuators. This paper will discuss the OPN 2 x 2 and 3 x 2 models and their applicability to a 3 x 3 model.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 13 of 102	

For the purpose of simplicity, it will be assumed that there are only three different types of components possible for each component class. In reality, three Sensor Types might include a Sun Sensor, star tracker, and an IMU. However, to be more generic, Types will instead be referred to as Type A, Type B, and Type C.

As a first pass, all enumerated architectures were evaluated based on two specific metrics: reliability and weight. In this study, the team elected to use weight as a surrogate indicator of system cost and complexity. The team did observe and understand that in some cases system cost can be driven by the complexity factor alone. However to keep the model simple for this top-level study the assumption made was that weight is a good first order approximation for system complexity and cost.

7.0 Data Analysis

Section 7.1 will discuss the design of the simple 2 x 2 model, Section 7.2 will give further details on the model, and Section 7.3 will discuss the design of the more complicated 3 x 2 model. Section 7.4 will examine the application of reliability and weight metrics to the enumerated architectures. Finally, Section 7.5 will provide some general concluding remarks.

7.1 The 2 x 2 GN&C System

This section begins the discussion of the design of the 2 x 2 model. Even with just four components (2 Sensors and 2 Computers), numerous architectures can be defined for a 2 x 2 system based on how the components are inter-connected. Each of these architectures will have different total weight and reliability.

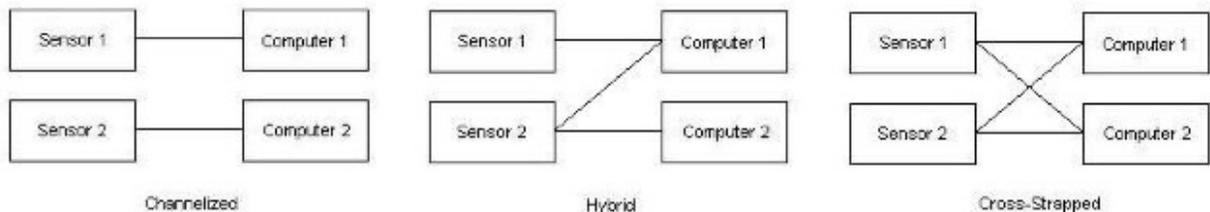


Figure 7.1-1. Three Possible 2 x 2 Systems

Figure 7.1-1 depicts three possible 2 x 2 architectures. The reliability, R , of the three models is shown in Table 7-1.1, where s_j is the reliability of Sensor j , and c_k is the reliability of Computer k .

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 14 of 102	

Table 7.1-1. Reliability Expressions for the 2 x 2 Systems in Figure 7.1-1

Architecture	Reliability
Channelized	$R = s_1c_1 + s_2c_2 - s_1c_1s_2c_2$
Hybrid	$R = s_1c_1 + s_2c_1 + s_2c_2 - s_2c_1c_2 - s_1s_2c_1 - s_1s_2c_1c_2 + s_1s_2c_1c_2$
Cross-Strapped	$R = s_1c_1 + s_1c_2 - s_1c_1c_2 + s_2c_1 + s_2c_2 - s_2c_1c_2 - s_1s_2c_1 - 2s_1s_2c_1c_2 + 2s_1s_2c_1c_2 - s_1s_2c_2 + 2s_1s_2c_1c_2 - s_1s_2c_1c_2$

It is important to note that no matter what the architecture, the reliability of any 2 x 2 model can be generated by taking the cross-strapped expression for R and then eliminating terms from the expression for connections which do not exist and therefore do not contribute to system reliability. Additional indicator variables are added to the cross-strapped reliability expression to specify which terms to eliminate. These indicator variables were correlated with the interconnections between components. A nonzero indicator variable represented a connection, whereas an indicator variable equal to zero represents a missing connection.

Using the methodology described, the following general expression for R is obtained:

$$R = s_1i_{11}c_1 + s_1i_{12}c_2 - s_1i_{11}i_{12}c_1c_2 + s_2i_{21}c_1 + s_2i_{22}c_2 - s_2i_{21}i_{22}c_1c_2 - s_1s_2i_{11}i_{21}c_1 - s_1s_2i_{11}i_{22}c_1c_2 - s_1s_2i_{12}i_{21}c_1c_2 + s_1s_2i_{11}i_{12}i_{21}c_1c_2 + s_1s_2i_{11}i_{21}i_{22}c_1c_2 - s_1s_2i_{12}i_{22}c_2 + s_1s_2i_{11}i_{12}i_{22}c_1c_2 + s_1s_2i_{12}i_{21}i_{22}c_1c_2 - s_1s_2i_{11}i_{12}i_{21}i_{22}c_1c_2$$

Where i_{jk} is the reliability of the connection between Sensor j and Computer k, if such a connection exists and 0 otherwise.

As a sanity check, the reliability expressions for the channelized and hybrid architectures can be derived from the general expression. Assuming perfect connection reliability (i.e., $i_{jk} = 1$ for all connections in the architecture) the channelized and hybrid architectures would be represented by the indicator variables in Table 7.1-2. Plugging these indicator variables into the general expression gives the same reliability expressions in Table 7.1-1.

Table 7.1-2. Indicator for the Channelized and Hybrid 2 x 2 Systems in Figure 7.1-1

Architecture	i_{11}	i_{12}	i_{21}	i_{22}
Channelized	1	0	0	1
Hybrid	1	0	1	1

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 15 of 102	

7.2 Details on the 2 x 2 OPN Model

Like the previously mentioned 3 x 2 and 3 x 3 models, the 2 x 2 OPN model was viewed as a sophisticated Petri net model. In a Petri net model, information-storing tokens move via directed arcs from transitions to and from places to transitions. Note that there may be more than one directed arc feeding from or to a transition or place. Upon arrival at a transition a token is consumed and then processing is completed, and, if appropriate, new tokens were introduced in the places dictated by the directed arcs leading from the transition.

The sequence of transitions in any of the discussed OPN models is a sequence of decision points. At each decision point, a token is replicated with multiplicity equal to the number of possible decisions. The information stored in each token represents a unique possible architecture. Taken together, the tokens enumerate all possible architectures given an initial set of constraints. All tokens are collected when they completely propagate through the model for analysis.

Figure 7.2-1 is a visual representation of the OPN decision tree for the 2 x 2 model and the following questions are the decision points:

- How many Sensors?
 - 1 or 2
- Type assignment for Sensors?
 - If only one Sensor, then choose Sensor A, Sensor B, or Sensor C
 - If two Sensors, then choose two of the same Type of Sensor or one of each of two Types (i.e., the possible combinations would be: AA, AB, AC, BB, BC, and CC)
- How many Computers?
 - 1 or 2
- Type assignment for Computers?
 - If only one Computer, then choose Computer A, Computer B, or Computer C
 - If two Computers, then choose two of the same Type of Computer or one of each of two Types (i.e., the possible combinations would be: AA, AB, AC, BB, BC, and CC)
- Which Sensors are connected to Computer 1?
 - Just Sensor 1
 - Just Sensor 2 (if Sensor 2 exists)

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 16 of 102	

- Both Sensor 1 and 2 (if Sensor 2 exists)
- If Computer 2 exists, which Sensors are connected to Computer 2?
 - Just Sensor 1
 - Just Sensor 2 (if it Sensor 2 exists)
 - Both Sensor 1 and 2 (if Sensor 2 exists)



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
17 of 102

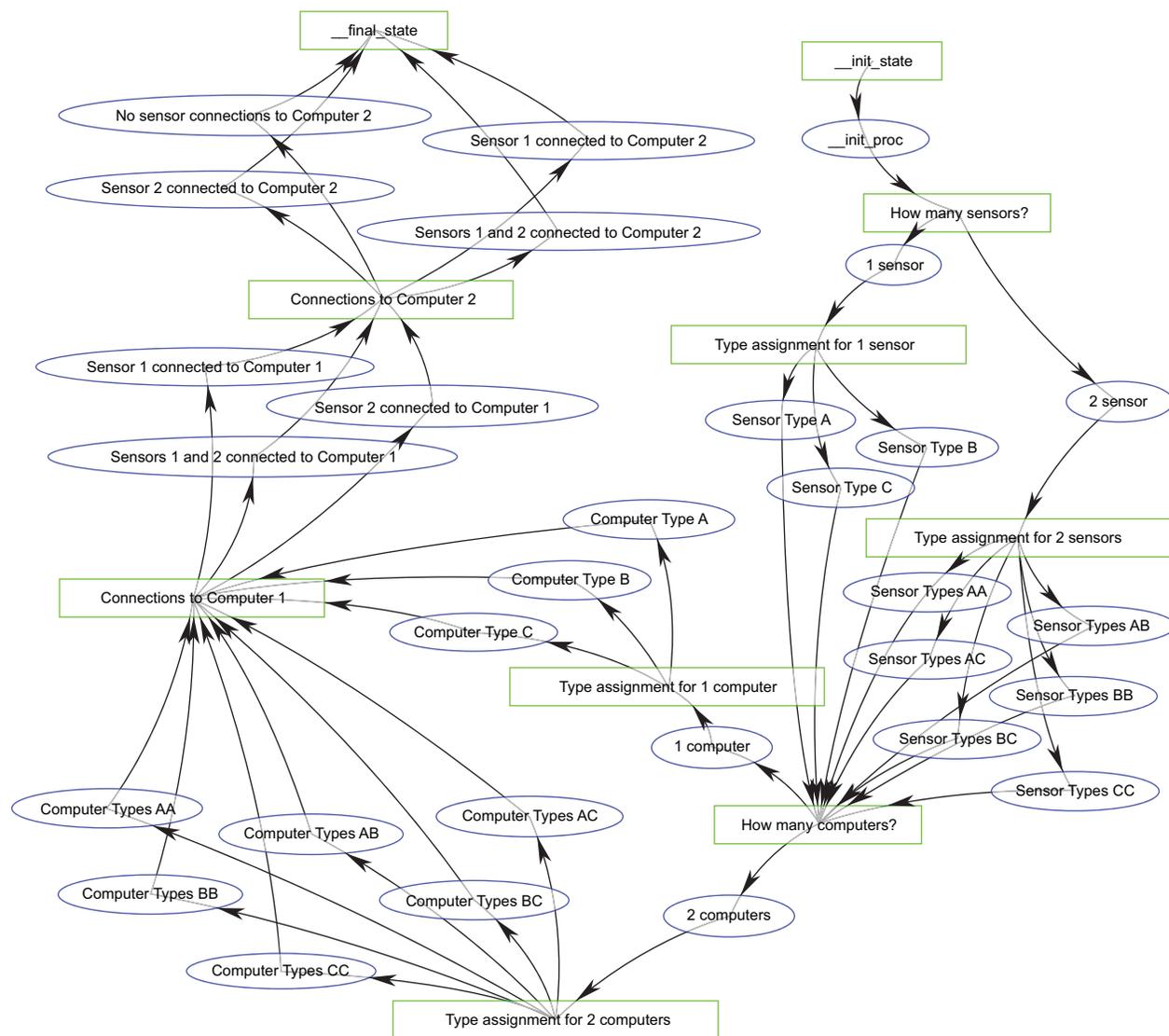


Figure 7.2-1. 2 x 2 OPN Decision Tree

During the process of token propagation, the number of components, component types, and connections were continuously updated for later use in reliability calculations. In addition, the current weight of the system was updated at execution time. Each component type was given its

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft			Page #: 18 of 102

own unique reliability and weight based on the specific make and model of the component. These values were based on real components, but modified slightly to facilitate analysis. Reliabilities are dependent upon the failure rate of the component and the desired operational time for the component. The relationship is governed by the equation $R = e^{-\lambda t}$, where λ is the failure rate and t is the operational time. Operational time is user defined and based on the length of the proposed mission. An operational time of $t = 10$ years was used in the models discussed in this paper. Other component properties are illustrated in Table 7.2-1 and Table 7.2-2.

Table 7.2-1. Component Properties For Sensor Types A, B, and C

Sensor Type	A	B	C
Failure Rate, λ (/year)	0.00015	0.0001	0.00005
Reliability, R	0.9985	0.999	0.9995
Weight (dimensionless)	3	6	9

Table 7.2-2. Component Properties For Computer Types A, B, and C

Computer Type	A	B	C
Failure Rate, λ (/year)	0.0001	0.00004	0.00002
Reliability, R	0.999	0.9996	0.9998
Weight (dimensionless)	3	5	10

Two additional weights and one reliability were also included in the model. A “connection weight” and a “dissimilar component penalty” were included to ensure that weight continues to approximate complexity and cost. Cross-strapping components may not add significant weight to the overall system, but adds to the system complexity and cost. Similarly, dealing with more than one Type of Sensor and/or Computer also increases complexity and cost. Hence, adding these additional weights where appropriate worked as a first step toward simulating an operational system.

The weights associated with connections and dissimilar components were chosen to be consistent with the weights of Sensors and Computers. To do so, assumptions were made and the connections were considered to be, at most, 1/3 of the complexity of the average Computer. In addition, the weight penalty for dissimilar components was set such that it was not larger than the heaviest Sensor or Computer.

These assumptions dictated a certain range of weight values used for connections and dissimilar component parameters. However, rather than presuppose exact values for these weights, multiple OPN runs were executed varying one of the parameters each time. Assuming the

	NASA Engineering and Safety Center Technical Assessment Report	Document #:	Version:
		NESC-RP-06-074	1.0
Title:		Page #:	
System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		19 of 102	

connection reliability would be greater than that of a Computer, the nine OPN scenarios are illustrated in Table 7.2-3.

Table 7.2-3. Connection Reliabilities, Connection Weights, and Dissimilar Component Penalties for Each OPN Scenario Run

OPN Scenario	1	2	3	4	5	6	7	8	9
Connection Reliability	1	1	1	0.99995	0.99995	0.99995	0.9999	0.9999	0.9999
Connection Weight (dimensionless)	0	0	0	0.5	0.5	0.5	5 / 3	5 / 3	5 / 3
Dissimilar Component Penalty (dimensionless)	0	6	9	0	6	9	0	6	9

7.3 Details on the 3 x 2 OPN Model

Implementation of the 3 x 2 OPN model is similar to that of the 2 x 2 model with two notable exceptions. These exceptions relate to the overall system reliability formula and the removal of duplicate architectures to minimize the computer memory required to run the model. The reliability formula was more complicated for the larger models and was handled differently. Although reliability is calculated after the OPN completes execution, it can no longer be easily manually calculated for implementation in Microsoft® Excel®. Instead, symbolic MATLAB® was used to calculate the formula and a MATLAB® script was used to insert the correct “i” indicator values where appropriate. Only after this manipulation was performed could the reliability values be imported into Excel® for implementation.

In addition, care had to be taken to ensure no architecture was represented more than once in the model. Running a larger 3 x 2 OPN model would take an inordinate amount of time and computer memory. It was found that certain architectures could be represented in multiple configurations and this was not taken into account by the 2 x 2 model¹. By producing tokens for all possible configurations of the same architecture, the model required significantly longer run times and used substantial memory.

¹ The 2 x 2 model is smaller than the 3 x 2 model. As a result, there were no memory issues and duplicate architectures were removed in post-processing (i.e., they did not have to be removed in OPN).

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 20 of 102	

An example of duplicate architecture is shown in Figure 7.3-1. A1 and A2 represent the same architecture, in both cases one Sensor of Type A is connected to a Computer of Type A, the other Sensor A is connected to a Computer A and a Computer B, and a Sensor of Type B is connected to a Computer of Type C. A3 represents a different architecture, however, since both Sensors of Type A are connected to a Computer of Type B.

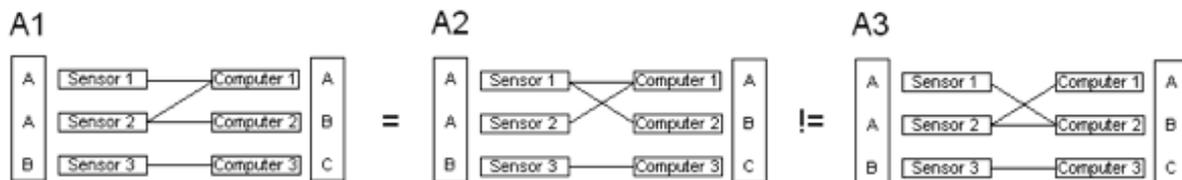


Figure 7.3-1. Determination of Duplicate Architectures

The process of eliminating duplicate architectures began by choosing a representative set of Sensor Types and Computer Types. Ten possibilities were chosen as representative orderings of the three Sensor Types and also the three Computer Types: AAA, AAB, AAC, ABB, ACC, ABC, BBB, BBC, BCC, and CCC.

A1 in Figure 7.3-1 represents a connection pattern between three adjacent components. This connection pattern has four connections: Sensor 1 to Computer 1, Sensor 2 to both Computer 1 and Computer 2, and Sensor 3 to Computer 3. Keeping this connection pattern fixed, the team gave a “Type” identity to the three Sensors based on the 10 possible orderings. For each possible ordering of the Sensor Types, there were 10 possible orderings of the Computer Types, each of which defines a unique architecture. In other words, for any given connection pattern such as A1, there are $10 \times 10 = 100$ possible architectures. The OPN model iterates through all possible connection patterns and finds all 100 possible architectures.

Note that orderings such as ABA and BAA are not taken to be representative orderings. When all possible connection patterns are taken into account, these additional orderings will fail to produce any architecture that cannot be produced by AAB. This is because ABA, BAA, and AAB are equivalent (i.e., all represent two components of Type A and one of Type B). It does not matter in what order the Types are described as long as the case is represented.

Searching for duplicate architectures in the OPN does not require checking 100 possible architectures for each connection pattern. The 100 possibilities for each connection pattern can be represented by 16 representative architectures. As illustrated in Figure 7.3-2, the 10 Sensor Type combinations and the 10 Computer Type combinations can be abstracted to just four



NASA Engineering and Safety Center Technical Assessment Report

Document #:
NESC-RP-06-074

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
21 of 102

representative combinations per component. First, AAA, BBB, and CCC represent the case where all three components are of the same type. Next, AAB, AAC, and BBC represent the case where the first two components were of the same Type and the third component was of a different Type. Furthermore, ABB, ACC, and BCC represent the case where the second and third components are of the same Type, but the first component is of a different Type. Finally, ABC represents the case where all three components were of a different Type.

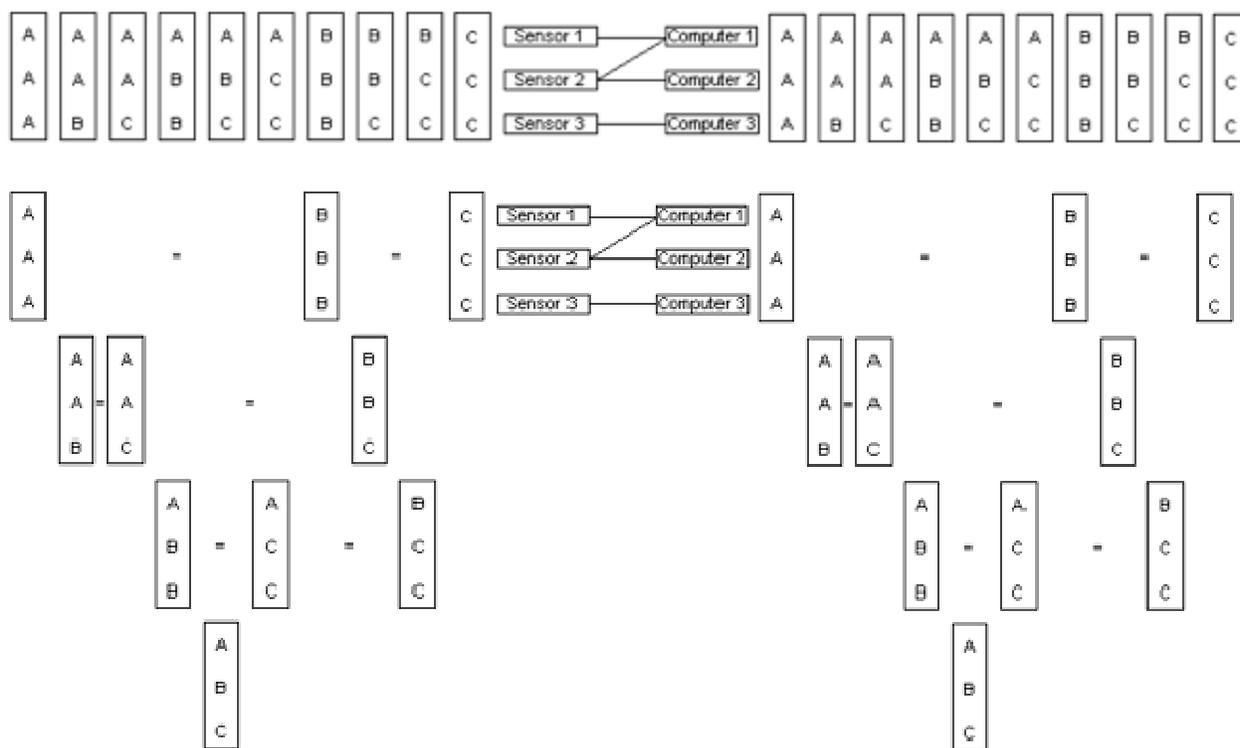


Figure 7.3-2. Finding Representative Architectures



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
22 of 102

Figure 7.3-3 helps demonstrate why these representative architectures work for finding duplicate architectures. To use representative architectures to find duplicates is to claim that if architecture A1 is equivalent to architecture A2, but not A3, then architecture B1 is equivalent to B2, but not B3.

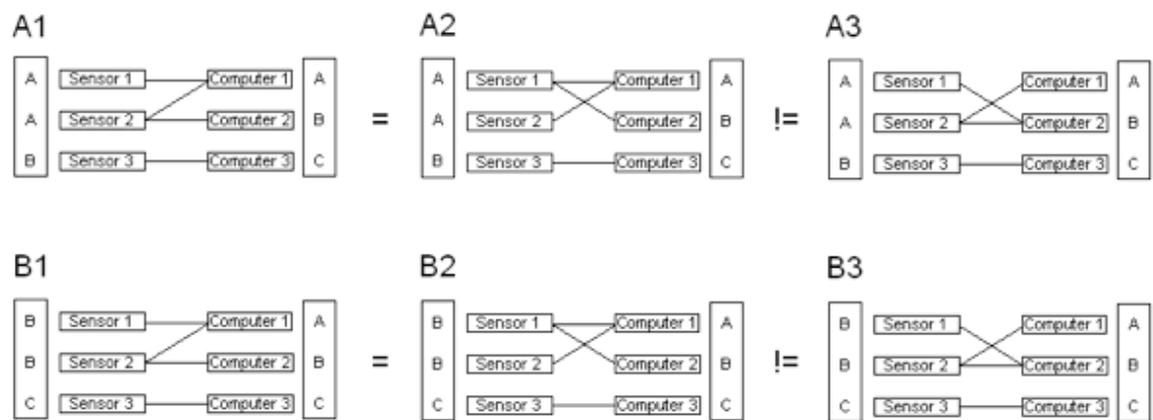


Figure 7.3-3. Using Representative Architectures to Find Duplicates

As previously discussed, A1 and A2 represent the same architecture even though they have different connection patterns. It is arbitrary which form of architecture is chosen as the primary form and which is a duplicate (i.e., either A1 or A2 could be considered the duplicate).

The study team implemented duplicate detection into the model, which was an involved manual process. The assessment had a finite schedule allocation and the development time necessary for automating the duplicate detection process was uncertain. It was therefore decided that a manual (i.e., brute force) method should be used to implement duplicate detection. All possible representative architectures were manually drawn and duplicate architectures were identified. In all, over 100 pages of architectures were drawn and compared.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
23 of 102

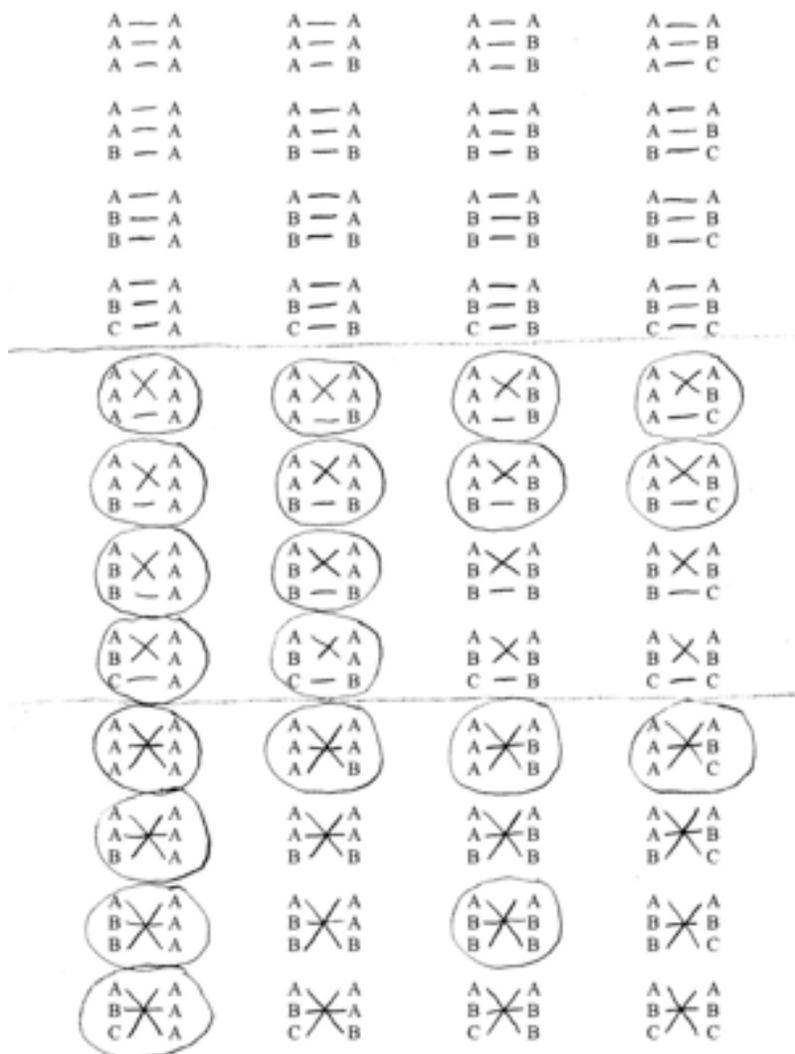


Figure 7.3-4. Example of Manually-Drawn Architectures

Based on the circled representative architectures, rules were created and inserted into the OPN to keep any tokens that will produce duplicate architectures from propagating. Note that all rules are in the form of Boolean expressions starting with “not if” instead of “if”. Although all work was double-checked, it is conceivable that an incorrect rule was entered due to human error. By using “not if” instead of “if”, the default is to pass the token. It was assessed that it is better to retain a duplicate architecture rather than exclude a potentially optimal architecture.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 24 of 102	

The Boolean rules are inserted into the OPN model on the transitions from:

- Which Sensors are connected to Computer 1?
- If Computer 2 exists, which Sensors are connected to Computer 2?
- If Computer 3 exists, which Sensors are connected to Computer 3?

To the places:

- Just Sensor 1
- Just Sensor 2 (if Sensor 2 exists)
- Just Sensor 3 (if Sensor 3 exists)
- Just Sensors 1 and 2 (if Sensor 2 exists)
- Just Sensors 1 and 3 (if Sensor 3 exists)
- Just Sensors 2 and 3 (if Sensor 3 exists)
- Sensors 1, 2, and 3 (if Sensor 3 exists)

Trivial rules govern which Sensors are connected to Computer 1. If a particular token represents an architecture with only two Sensors, it is not possible to make a connection to a Computer from a nonexistent Sensor 3. Therefore, in the two Sensor case, no new tokens are introduced into the places representing, “just Sensor 3”, “just Sensors 1 and 3”, “just Sensors 2 and 3”, or “Sensors 1, 2, and 3”. Similarly, if a particular token represents an architecture with only one Sensor, no new tokens are introduced into the places representing, “just Sensor 2”, “just Sensor 3”, “just Sensors 1 and 2”, “just Sensors 1 and 3”, “just Sensors 2 and 3”, or “Sensors 1, 2, and 3”.

Rules governing connections to Computer 2 and 3 are more complicated. If a token represents an architecture with only two Computers, the final system architecture will be evident after creating the Sensor connections to the second Computer. If a token represents an architecture with three Computers, it is known that there will be a final system architecture after creating the Sensor connections to the third Computer. Connections that will form duplicate (circled) architectures should not be allowed to propagate. Hence rules are followed to block introduction of these tokens.

Figure 7.3-5 shows an example of a rule based on a manually drawn architecture. This rule determines whether a connection should be made between Sensor 3 and Computer 3. Note that, by the time a token reaches the given rule, the connections between Sensor 1 and Computer 2, and between Sensor 2 and Computer 1 have already been defined. No connection should be made if the type definitions for the Sensors and Computers match those represented by the circled architectures (i.e., such tokens will result in the formation of duplicate architectures). In other words, the connection between Sensor 3 and Computer 3 should not be made if Sensor



**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

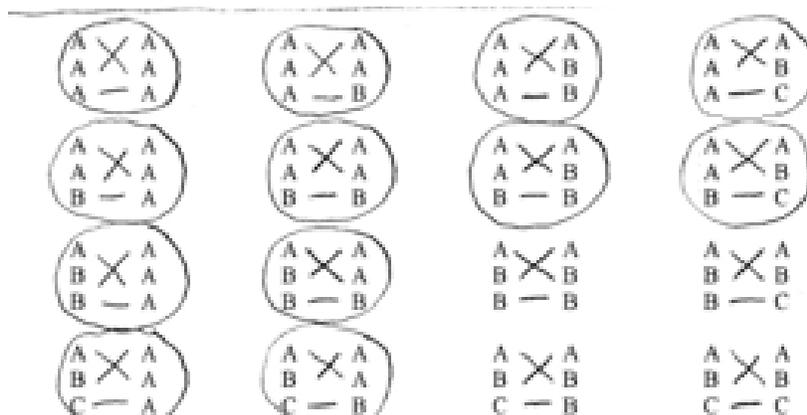
Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
25 of 102

1's Type is the same as Sensor 2's Type, or Computer 1's Type is the same as Computer 2's Type.



```
!(Computer Redundancy < 3) &&
!(Sensor Redundancy < 3) &&
!(
  (i11 == 0 && i12 == 0) ||
  (Sensor Redundancy > 1 && i21 == 0 && i22 == 0) ||
  (
    (i11 == 0 && i12 > 0 && i21 > 0 && i22 == 0 && i31 ==
    0 && i32 == 0) &&
    (
      (Sensor1_Type == Sensor2_Type) ||
      (Computer1_Type == Computer2_Type)
    )
  )
)
```

Figure 7.3-5. Example of Rule Based on a Manually Drawn Architecture

Eliminating duplicate architectures in the 3 x 2 OPN model significantly reduced the number of tokens produced from 51,902 to 9,795 tokens.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 26 of 102	

7.4 Results

Despite eliminating the unnecessary duplicate architectures from the model, attempts to run a 3 x 3 model still resulted in a memory shortage. Although it is unfortunate that the larger OPN model could not be completed, it is important to note that results from the 3 x 2 model can be applied to the 3 x 3 case.

By taking a top-level view of a GN&C system, the interaction between adjacent Sensor and Computer components is identical to the interaction between adjacent Computer and Actuator components. Similar to the Sensors and Computers, there are different Types of Actuators, each with a unique set of properties. A system architect will be able to choose different redundancies for each of these Types. Furthermore, the connection patterns already found between Sensors and Computers are the same as those between Computers and Actuators. Finally, the metrics of weight and reliability can be calculated in the same way.

There were nine scenarios outlined in this section, each scenario was run and Pareto plots were produced for each. Representative plots for the nine scenarios are reproduced in Figures 7.4-1 through 7.4-4. In each scenario, the architectures that simultaneously had both lowest weight and highest reliability were identified. These architectures are “on the Pareto front.” The identified Pareto- architectures were found by zooming in on the “utopia point” at the lower right hand corner of the plot². Note that in most cases, there are multiple identified architectures for each scenario since it is somewhat subjective which architectures are closer to the utopia point. For example: Is an architecture with weight = 17 and reliability = 0.999999596912468 (six “9”s) better than an architecture with weight = 18 and reliability = 0.99999995634079 (eight “9”s)?

The answers to such questions are made clear in the mission requirements context. For a human-rated mission, perhaps a reliability of 0.9999999 (seven “9”s) is required for safety. If this were the case, the architecture with weight = 18 would be better, since the architecture with weight = 17 does not meet the seven “9”s requirement of this example.

In the zoomed out (top) plot of each of the nine scenarios, there appear to be six clusters of architecture data points. The architectures in each cluster have nearly identical reliabilities. Looking from left to right, the first five clusters (i.e., the five clusters with the lowest reliability) are driven by single point failures of any of the six component Types. For example, in an architecture that contains one Sensor and two Computers, or an architecture that contains one Sensor and three Computers, the single Sensor present in the architecture must remain viable in order for the overall system to remain reliable. Sensor Type A has a reliability of 0.9985.

² The utopia point represents the ideal architecture which is 100 percent reliable with weight = 0.



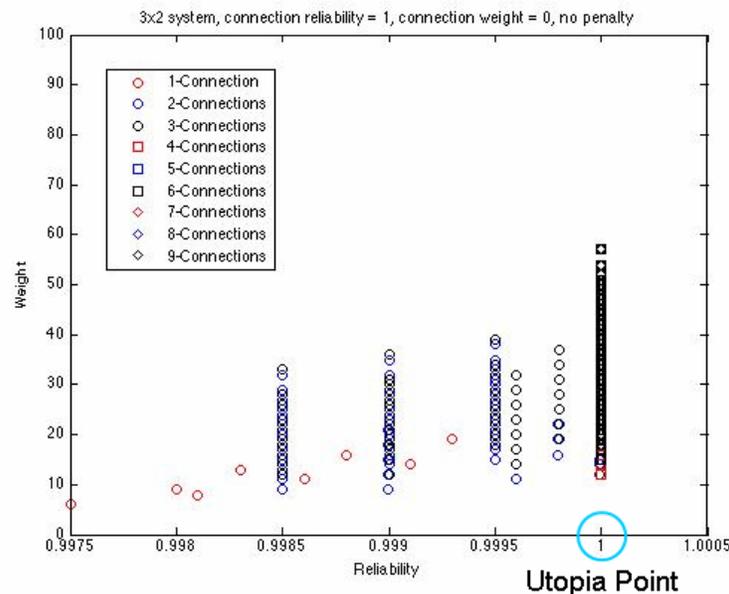
Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
27 of 102

Therefore, a one-Sensor two-Computer or one-Sensor three-Computer architecture that contains Sensor A can have a maximum reliability of 0.9985. Architectures that have a single point failure at Sensor A, define the first (least reliable) cluster of data points. Both Sensor Type B and Computer Type A have the same reliability of 0.999. Therefore, architectures that have a single point failure at a Sensor of Type B, or a single point failure at a Computer of Type A, will fall into the second cluster. Similarly, Sensor Type C's reliability of 0.9995 defines the third cluster, Computer Type B's reliability of 0.9996 defines the fourth cluster, and Computer Type C's reliability of 0.9998 defines the fifth cluster.

The sixth and final (most reliable) cluster contains all other architectures (i.e., architectures free from single point failures). The additional points that do not fall into any of the six clusters are single-string architectures (i.e., architectures that contain one Sensor and one Computer). These architectures contain not one, but two single point failures and are therefore significantly less reliable than an identical architecture with additional Computers or additional Sensors.





Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
28 of 102

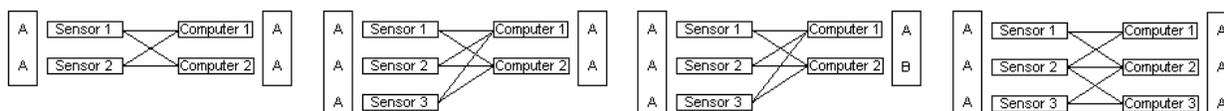
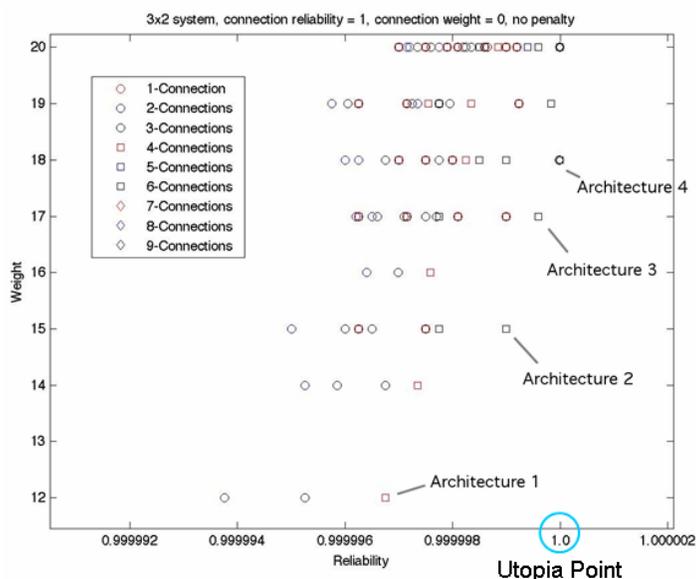


Figure 7.4-1. Architecture Pareto Plots

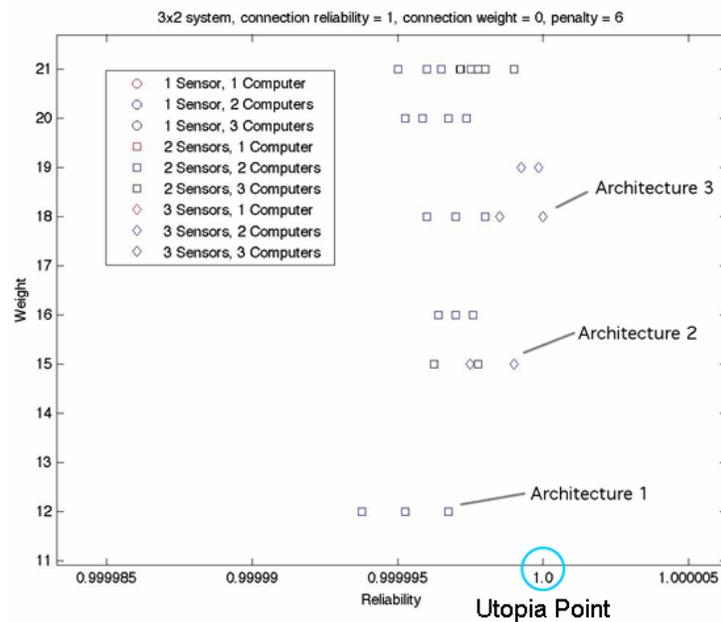
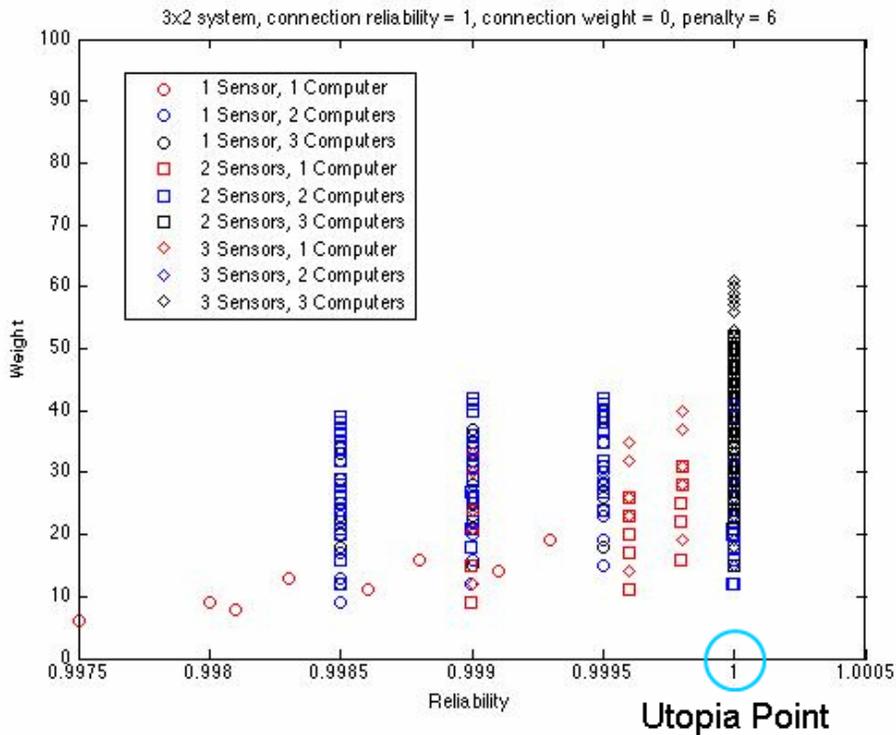
Note: (Top) Pareto plot with added details for number of connections between Sensors and Computers, (Middle) zoomed in version of the same plot, (Bottom) potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 12 and reliability = 0.9999967, architecture 2 has weight = 15 and reliability = 0.9999989, architecture 3 has weight = 17 and reliability = 0.9999959, and architecture 4 has weight = 18 and reliability = 0.999999956



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
29 of 102





**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
30 of 102

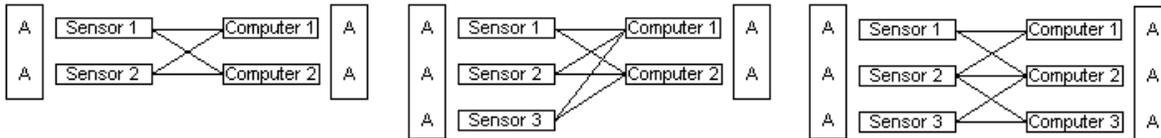


Figure 7.4-2. Architecture Pareto Plots

Note: (Top) Pareto plot with added details for both number of Sensors and number of Computers, (Middle) zoomed in version of the same plot, (Bottom) potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 12 and reliability = 0.99999675437371, architecture 2 has weight = 15 and reliability = 0.999998997632005, and architecture 3 has weight = 18 and reliability = 0.999999995634079



Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
31 of 102

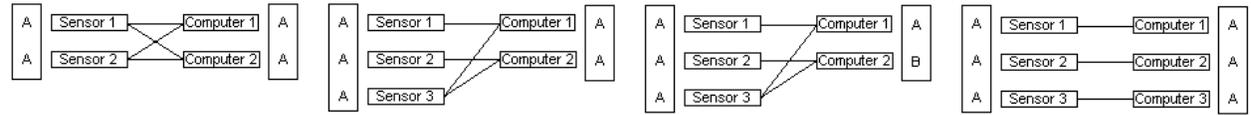
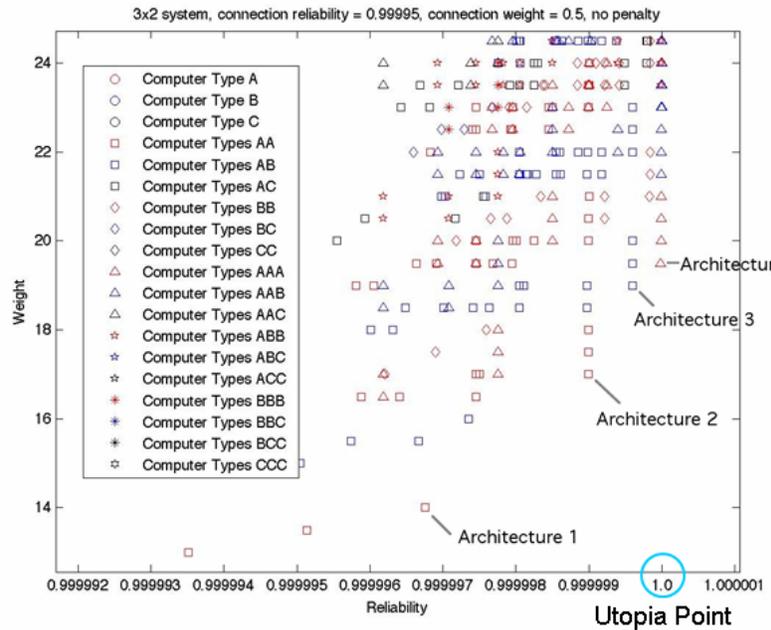
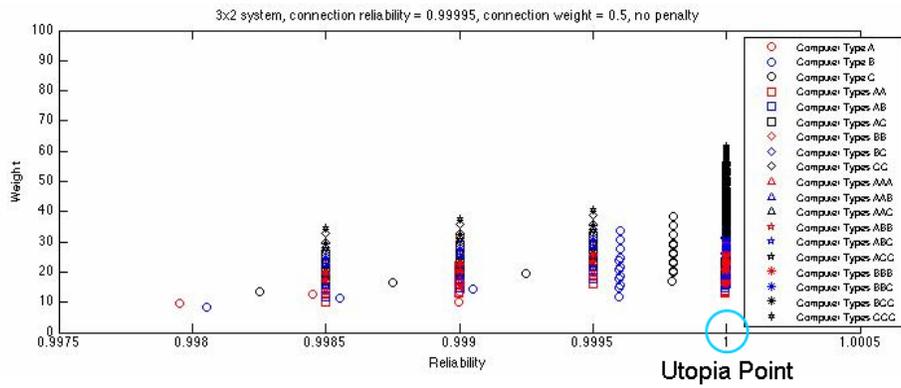


Figure 7.4-3. Architecture Pareto Plots

Note: (Top) Pareto Plot with Added Details for Both Number and Types of Computers, (Middle) Zoomed in version of the same plot, (Bottom) potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 14 and reliability = 0.999996754062388, architecture 2 has weight = 17 and reliability = 0.999998992620742, architecture 3 has weight = 19 and reliability = 0.999999593334442, and architecture 4 has weight = 19.5 and reliability = 0.99999983481890



Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
 32 of 102

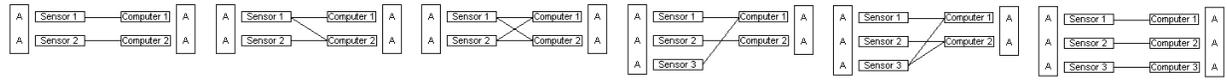
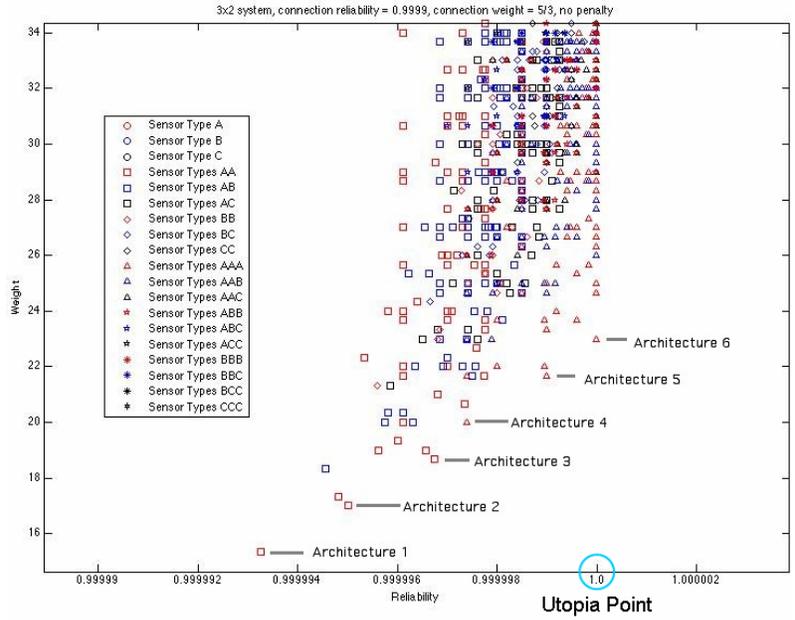
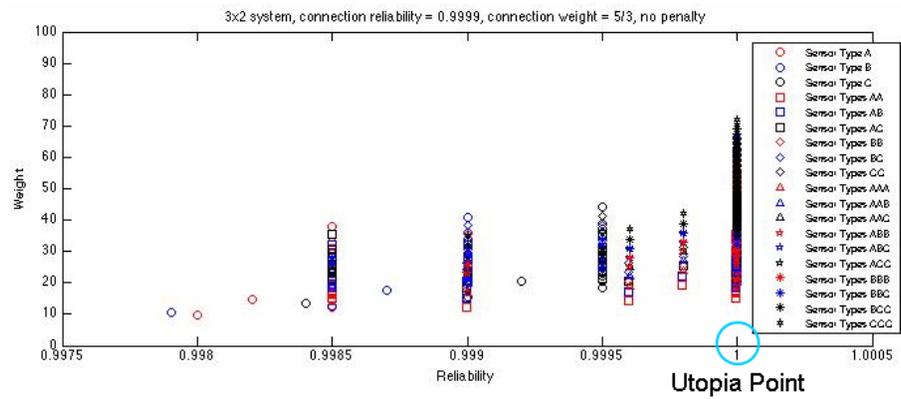


Figure 7.4-4. Architecture Pareto Plots

Note: (Top) Pareto plot with added details for both number and Types of Sensors, (Middle) zoomed in version of the same plot, (Bottom) Potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 15.333 and reliability = 0.999993257523472, architecture 2 has weight = 17 and reliability = 0.99999501059889, architecture 3 has weight = 18.667 and reliability = 0.999996753726173, architecture 4 has weight = 20 and reliability = 0.999997398039817, architecture 5 has weight = 21.667 and reliability = 0.999998992071849, and architecture 6 has weight = 23 and reliability = 0.999999983481890

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 33 of 102	

The ideal case would be that there is no penalty for connection weight or using dissimilar components, which can be seen in Figure 7.4-1. In this scenario, all identified architectures are fully cross-strapped. This is to be expected since additional connections increase reliability yet cost nothing. Figure 7.4-1 also demonstrates that the weight of an architecture's component plays a major role in determining the optimality of the architecture. Recall that components of Type A are the lightest and least reliable, and that components of Type C are the heaviest and most reliable. Even though components of Type A is the least reliable, Sensors of Type A and Computers of Type A are by far the most prevalent component Types in all the optimal architectures. In addition, although components of Type C are the most reliable, no components of this type appear in any of the optimal architectures. Furthermore, there were no Sensors of Type B in the optimal architectures. Architecture three contains a Computer of Type B, but this architecture is no longer optimal once the "dissimilar components penalty" is increased from penalty = 0 to penalty = 6 (see Figure 7.4-2). Since the optimal architectures for penalty = 6 contain no dissimilar components, increasing the penalty to = 9 results in no further changes to the optimal architectures.

Figure 7.4-3 provides a baseline for what can be considered a realistic system. In this scenario, there are no longer perfect connection reliabilities of 100 percent and there is a cost for producing connections (i.e., connection reliability = 0.99995 and connection weight = 0.5). As a result of the 0.5 connection weight, only one fully cross-strapped architecture (Architecture 1) is among the optimal architectures. The other optimal architectures have 3 or 4 connections. Among these, Architectures 2 and 3 are the most interesting. Each has three Sensors and two Computers with two of the Sensors having one connection to a Computer and the last having two connections.

The scenario in which connection weight = 0.5 is similar with the connection weight = 0 scenario. Once again, the component types in the optimal architectures are predominantly of Type A and never of Type C. Only one optimal architecture (Architecture 3) contains a component of Type B (Computer). This architecture is not optimal when the dissimilar components penalty is increased to penalty = 6. There is no change in optimal architectures for connection weight = 0.5 when this penalty is increased from penalty = 6 to = 9.

Figure 7.4-4 depicts the first scenario where connection reliability = 0.9999 and connection weight = 5/3. This scenario produces similar optimal architectures to the connection weight = 0.5 scenario, but with notable exceptions.

Even with the dissimilar components penalty set to penalty = 0, connection weight = 5/3 is sufficiently high as to eliminate any architecture that contains a component type heavier than

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft			Page #: 34 of 102

Type A. This means that Architecture 3 from Figure 7.4-3 is the only connection weight = 0.5 architecture which contains a component of Type B, and is not an optimal architecture for connection weight = 5/3. This also means that the optimal architectures for connection weight = 5/3 will not change if the dissimilar components penalty is increased to penalty = 6 or = 9.

A connection weight of 5/3 makes it more desirable to have architectures with fewer connections. Although Architectures 1, 2, and 4 from Figure 7.4-3 are still optimal architectures when the connection weight is increased to 5/3, the value of these architectures was diminished with the heavier connection weight. As a result, these architectures are no longer closer to the utopia point than Architectures 1, 2, and 4 from Figure 7.4-4.

The six potentially optimal architectures for connection weight = 5/3 produce an interesting set. All legal architectures with four or fewer connections that contain two to three Sensors of Type A and two Computers of Type A are optimal architectures for this scenario. In effect, a system architect is directly trading an increase in weight for additional reliability when connection weight = 5/3.

By reviewing a subset of the possible architectures, specifically a subset in which all members have the same number of connections, the effect of the dissimilar components penalty on the optimal architectures is nearly identical to the penalty's effect on the optimal architectures of the superset. Figures 7.4-5 and 7.4-6 depict the optimal architectures for 3 x 2 systems with 1 through 9 connections when the connection reliability equals one and connection weight equals zero. The reliabilities and weights for these architectures can be found in Tables 7.4-1 and 7.4-2. Again, as the penalty is increased from penalty = 0 to 6, nearly all architectures containing Sensor Type B or Computer Type B cease to be optimal. For architectures with the same number of connections, the same architectures which are optimal for penalty = 0 will be optimal no matter what the connection weight. Similarly, architectures which are optimal for penalty = 6 or 9 remained optimal no matter what the connection weight. Although the overall system weight of any subset member will change if the connection weight is modified, this change will be identical to the transformation seen by any of the other systems in this subset. This is because all systems in each subset have, by definition, the same number of connections. Therefore, among architectures with the same number of connections, the architectures closest to the utopia point will remain closest to the utopia point regardless of the change to the connection weight.



**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
35 of 102

Table 7.4-1. Reliabilities and Weights for the 1- through 9- Connection Architectures Closest the Utopia Point

Note: Assuming a 3 X 2 System with Connection Reliability = 1, Connection Weight = 0, and No Penalty for Dissimilar Components

Connections	Reliability	Weight	On Pareto front?
1	0.999100405	14	
	0.999300245	19	
2	0.999993766	12	
	0.999995260	14	
	0.999996397	16	
	0.999997344	19	
	0.999997754	20	
	0.999998292	22	
3	0.999998741	25	
	0.999995260	12	
	0.999996756	14	
	0.999997499	15	
	0.999998996	17	
4	0.999999984	18	
	0.999996754	12	Yes
	0.999998993	15	
	0.999999594	17	
5	0.999999988	18	
	0.999998995	15	
	0.999999596	17	
6	0.999999992	18	
	0.999998998	15	Yes
	0.999999597	17	Yes
7	0.999999996	18	
	0.999999996	18	
9	0.999999996	18	Yes



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
36 of 102

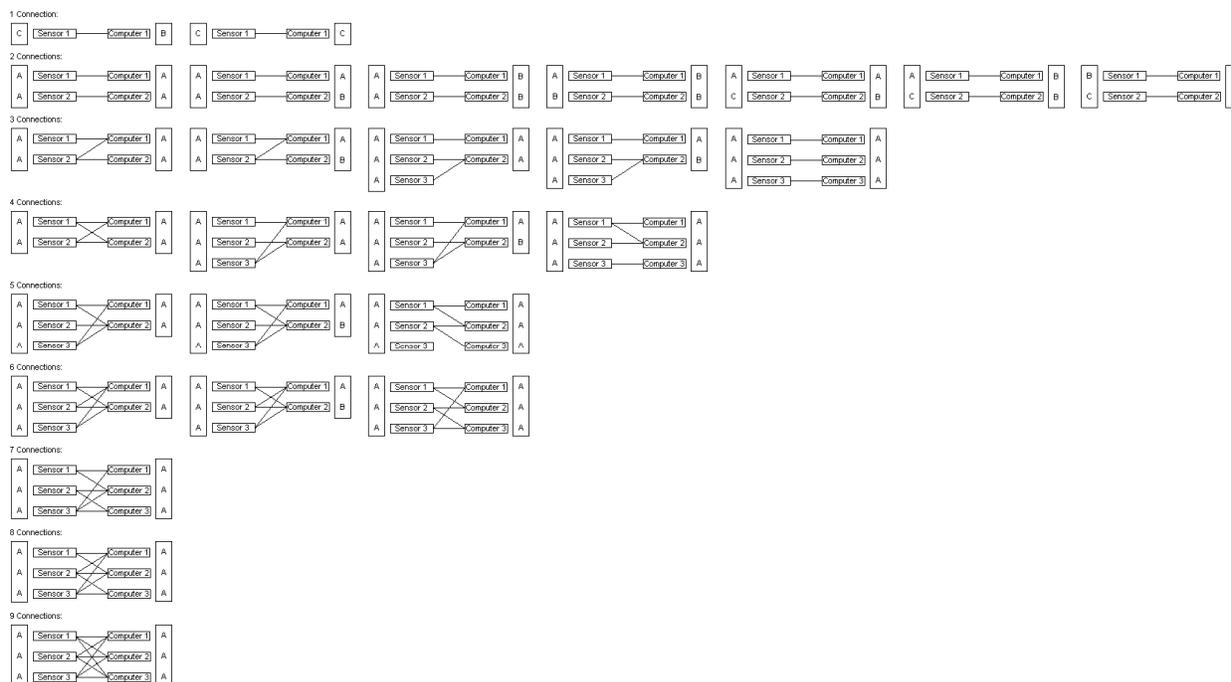


Figure 7.4-5. Architectures Described in Table 7.4-1



**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
37 of 102

Table 7.4-2. Reliabilities and Weights for the 1- through 9- Connection Architectures Closest the Utopia Point

Note: Assuming a 3 X 2 System with Connection Reliability = 1, Connection Weight = 0, and Penalty = 6 for Dissimilar Components

Connections	Reliability	Weight	On Pareto front?
1	0.999100405	14	
	0.999300245	19	
2	0.999993766	12	
	0.999996397	16	
3	0.999995260	12	
	0.999997499	15	
	0.999999984	18	
4	0.999996754	12	Yes
	0.999998993	15	
5	0.999999988	18	
	0.999998995	15	
	0.999999992	18	
6	0.999998998	15	Yes
	0.999999996	18	
7	0.999999994	18	
8	0.999999996	18	
9	0.999999996	18	Yes



Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
38 of 102

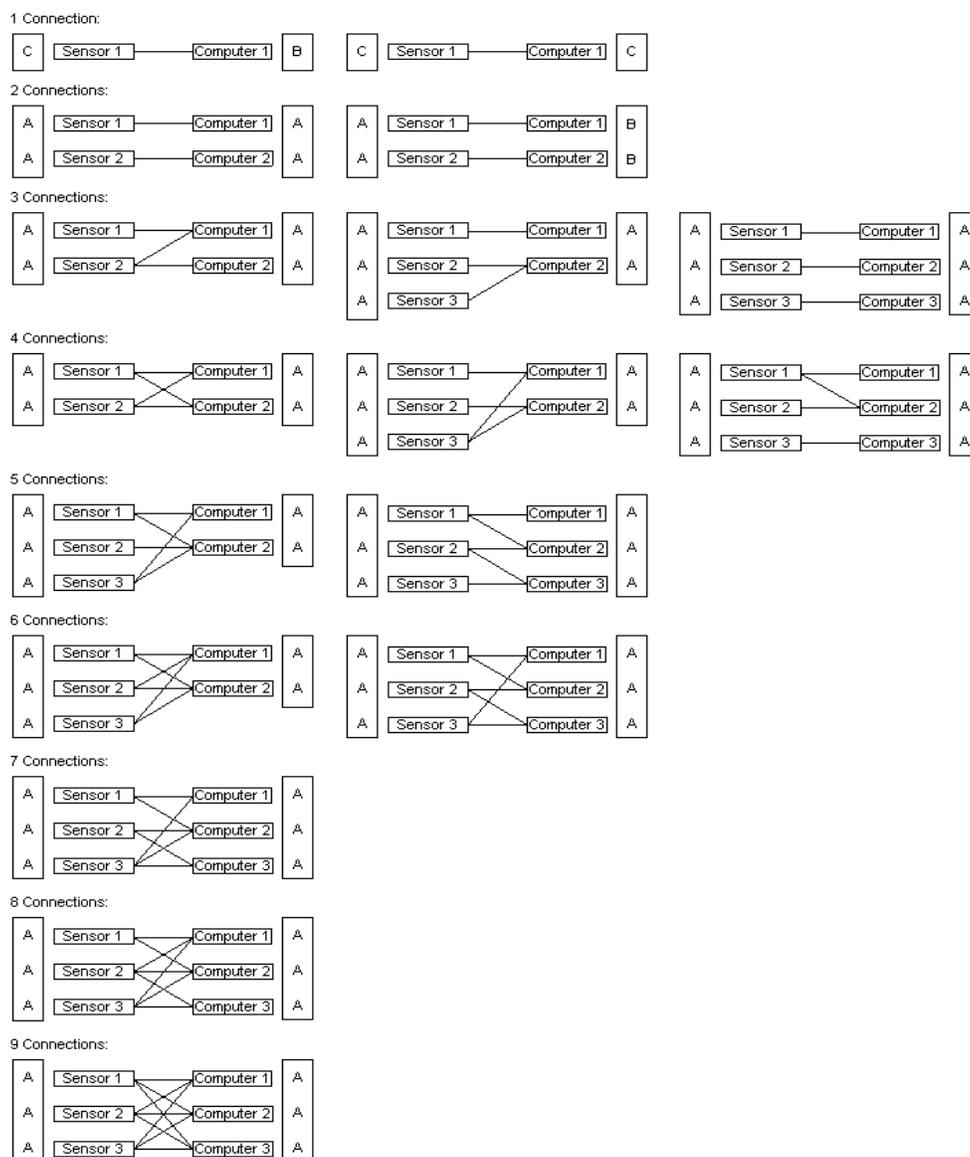


Figure 7.4-6. Architectures Described in Table 7.4-2

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 39 of 102	

7.5 Concluding Remarks

There is merit in mapping the entire possible solution space. By showing where similar classes of solutions fall within the entire set, it allows one to see ‘and how certain solution features effect FOM performance and derive architecture ‘rules of thumb’. It also allows one to see the optimal solution boundary (the Pareto Front) and understand how one FOM can be exchanged for another.

This approach provides insight into a potential limitation of the ‘Minimum Functionality/Minimum Implementation’ system architecting methodology [ref.6] which uses as its starting point a single-string non-redundant system architecture which lies somewhere in the interior of the solution trade space. The method involves performing one trade at a time to improve system safety until the mass margin is gone. It is not clear that this stepwise optimization of a single design can ever get to the boundary of optimal solutions (the Pareto Front). Even if it somehow did reach the optimal boundary it is not clear the system architects will have access to and be able to visualize the whole range of optimal solutions. System architects using the ‘Minimum Functionality/Minimum Implementation’ approach should at least be aware that the technique described in this report is available and they should consider using it to explore the entire system trade space and to provide some comparative outputs for cross-check their results.

With some enhancements the systematic GN&C/Avionics “building block” OPN modeling techniques employed by the MIT/Draper Laboratory would serve as an excellent tool for evaluating competing GN&C system architectures for future NASA spacecraft. This OPN-based approach, or other similar modeling tools, would perform the extremely useful up-front function of identifying the most attractive (lowest weight and overall “cost”) GN&C architectural options that satisfy a prescribed set of spacecraft fault tolerance, reliability and performance requirements. Although less likely, but worth observing, is the fact that such “building block” models could be used to identify the optimal highest reliability/lowest weight architectural options for a prescribed number and configuration of connections between adjacent GN&C components.

On-board GN&C flight software was, by design, not included in this GN&C system modeling and analysis study. Since there are potential for achieving flight software commonalty across multiple spacecraft this aspect of the system architecture should be considered in future work.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 40 of 102	

8.0 Findings and NESC Recommendations

8.1 Findings

The following NESC/MIT/Draper Laboratory study team findings were identified:

- F-1.** Most optimal architectures can be created by using the lowest reliability and lightest components.
 - The analysis of the identified optimal architectures show that it is possible to produce nearly all potentially optimal architectures using only the Type A light weight/low-reliability Sensors, Type A light weight/low-reliability computers, and generic connections.
- F-2.** It is preferable to increase the redundancy of lighter, less reliable components rather than to use smaller numbers of more reliable, heavy components.
- F-3.** There are diminishing returns to adding redundancy, connections, or components with greater reliability.
 - The system will experience only minimal increases in overall reliability for the large gains in system weight.
- F-4.** For the optimal architectures in each scenario, the number of connections drops dramatically as the connection weight penalty is increased.
- F-5.** The impact of the dissimilar components penalty is more subtle than that for connectors, but is still apparent.
 - Some architectures containing both Computer Type A and Computer Type B were identified as potentially optimal when the penalty = 0. However, when the penalty is increased to = 6, these architectures no longer appear to be better than other architectures.
- F-6.** The OPN-based modeling approach employed required a time-consuming manual process for eliminating architectural duplicates.
- F-7.** The OPN-based modeling approach employed was top-level only.
- F-8.** In addition to the reliability metric, the existing OPN model could be modified to identify the optimal GN&C /Avionics architectural options for satisfying other driving

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 41 of 102	

system metrics, such as fault tolerance constraints, avionics power consumption, and/or attitude control performance.

8.2 NESC Recommendations

The following NESC recommendations were identified and directed to the ESMD Chief Engineer, the CxP Program System Engineering organization, and the Orion and Altair GN&C/Avionics Subsystem designers:

- R-1.** Consider the fundamental findings of the NESC/MIT/Draper Laboratory study team to determine, given their reliability modeling assumptions, if the trends identified can be applied to architecting future CxP spacecraft GN&C/Avionics subsystems. *(F-1, F-2, F-3, F-4, and F-5)*
- R-2.** Comprehensively investigate, using the OPN-based model described in this report (or using similar GN&C/Avionics modeling tools and methods) both the pros and cons of incorporating dissimilar GN&C Sensor, Computer, and Actuator components in the GN&C architectures. Study how a small set of crew/flight safety critical GN&C functions can be synthesized/implemented in a backup manner with a limited complement of dissimilar hardware and software components. *(F-5)*
- R-3.** Future work in on reliable GN&C should include the following enhancements of the GN&C/Avionics system modeling approach *(F-6, F-7, F-8, and F-9)*:
 - a) Improve the efficiency of the OPN model processing and its memory management.
 - b) Incorporate provisions to model more than three Types of GN&C Sensors, Computers, and Actuators into the model.
 - c) Include additional intrinsic descriptive details for each component beyond reliability and weight.
 - d) Support expanded GN&C/Avionics architectural layouts in which the component redundancy for any component can be greater than three. For example the enhanced model should be capable of evaluating an architecture consisting of four Sensors, three Computers, and two Actuators.
 - e) Automate the process for eliminating architectural duplicates and for creating rules.
- R-4.** Future work on reliable GN&C should include the addition of the following metrics for higher-fidelity analysis and evaluation of GN&C/Avionics system architectural robustness, reliability, mass, power, volume, and performance *(F-9)*:

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 42 of 102	

- a) The fault tolerance requirements/rules for the specific spacecraft application being modeled. Metrics on the cost of any analytical redundancy (i.e., the redundancy management software) needed to detect and isolate faults should be provided for evaluation.
- b) The impact of common mode hardware failures. The metric could be either the incremental benefit of dissimilar redundancy or the incremental risk of similar redundancy.
- c) The values for Sensor, Computer, and Actuator component Mass, Volume, and Power (MVP).
- d) The 6-DOF spacecraft attitude and position control/knowledge performance metrics.

9.0 Alternate Viewpoints

There were no alternate viewpoints or minority opinions expressed by the members of the NESC/MIT/Draper Laboratory study team.

10.0 Other Deliverables

There are no other deliverables at this time.

11.0 Lessons Learned

There were no lessons learned.

12.0 Definition of Terms

Corrective Actions Changes to design processes, work instructions, workmanship practices, training, inspections, tests, procedures, specifications, drawings, tools, equipment, facilities, resources, or material that result in preventing, minimizing, or limiting the potential for recurrence of a problem.

Finding A conclusion based on facts established by the investigating authority.

Lessons Learned Knowledge or understanding gained by experience. The experience may be positive, as in a successful test or mission, or negative, as in a mishap or failure. A lesson must be significant in that it has real or assumed impact on operations; valid in that it is factually and technically correct; and applicable in that it identifies a specific design, process, or decision

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 43 of 102	

that reduces or limits the potential for failures and mishaps, or reinforces a positive result.

Observation A factor, event, or circumstance identified during the assessment that did not contribute to the problem, but if left uncorrected has the potential to cause a mishap, injury, or increase the severity should a mishap occur. Alternatively, an observation could be a positive acknowledgement of a Center/Program/Project/Organization’s operational structure, tools, and/or support provided.

Problem The subject of the independent technical assessment/inspection.

Proximate Cause The event(s) that occurred, including any condition(s) that existed immediately before the undesired outcome, directly resulted in its occurrence and, if eliminated or modified, would have prevented the undesired outcome.

Recommendation An action identified by the assessment team to correct a root cause or deficiency identified during the investigation. The recommendations may be used by the responsible Center/Program/Project/Organization in the preparation of a corrective action plan.

Root Cause One of multiple factors (events, conditions, or organizational factors) that contributed to or created the proximate cause and subsequent undesired outcome and, if eliminated or modified, would have prevented the undesired outcome. Typically, multiple root causes contribute to an undesired outcome.

13.0 Acronyms List

CaLV	Cargo Launch Vehicle
CE&R	Concept Exploration and Refinement
CEV	Crew Exploration Vehicle
CLV	Crew Launch Vehicle
Cx	Constellation
CxP	Constellation Program
DOF	Degree of Freedom
EDS	Earth Departure Stage
ESMD	Exploration Systems Mission Directorate

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 44 of 102	

FOM	Figure of Merit
IMU	Inertial Measurement Unit
JSC	Johnson Space Center
LaRC	Langley Research Center
LSAM	Lunar Surface Access Module
MTBF	Mean Time Between Failure
MVP	Mass, Volume and Power
NESC	NASA Engineering and Safety Center
NRB	NESC Review Board
TDT	Technical Disciple Team
TIM	Technical Interchange Meeting

14.0 References

1. Cameron, B., Crawley, E., Loureiro, G., and Rebentisch, E., "Value flow mapping: Using networks to inform stakeholder analysis" Acta Astronautica, Volume 62, Issues 4-5, February-March 2008.
2. Dominguez-Garcia, A., Hanuschak, G., Hall, S., and Crawley, E., "A Comparison of GN&C Architectural Approaches for Robotic and Human-Rated Spacecraft" AIAA Guidance, Navigation and Control Conference and Exhibit, 20-23 Aug 2007, Hilton Head, SC.
3. Hofstetter, W., Wooster, P., Nadir, W., and Crawley E., "Affordable Human Moon and Mars Exploration through Hardware Commonality" AIAA-2005-6757 Space 2005, Long Beach, California, Aug. 30-1, 2005.
4. Sahner, R., Trivedi, K., and Puliafito, A. Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package, Kluwer Academic Publishers, 1995.
5. Simmons, W., Koo, B., and Crawley, E., "Architecture Generation for Moon-Mars Exploration Using an Executable Meta-Language" AIAA-2005-6726 Space 2005, Long Beach, California, Aug. 30-1, 2005.
6. Bay, Michael, Davis, Mitchell, and Putney, Blake, "Iterative Risk Driven Design Approach for CEV Avionics", 7th Annual Space Systems Engineering & Risk Management Symposium, Los Angeles, CA, February 2008

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 45 of 102	

Volume II: Appendices

- Appendix A. A Comparison of GN&C Architectural Approaches for Robotic and Human-Rated Spacecraft
- Appendix B. A Comparison of Fault-Tolerant GN&C System Architectures Using the Object Process Network (OPN) Modeling Language
- Appendix C. NESC Stakeholder Outbrief

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft			Page #: 46 of 102

Appendix A. A Comparison of GN&C Architectural Approaches for Robotic and Human-Rated Spacecraft

AIAA Guidance, Navigation and Control Conference and Exhibit, 20-23 Aug 2007, Hilton Head, SC

A Comparison of GN&C Architectural Approaches for Robotic and Human-Rated Spacecraft

Alejandro D. Domínguez-García*, Gregor Z. Hanuschak[†],
Steven R. Hall[†] and Edward F. Crawley[‡]
Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

This paper compares various architectural philosophies used in the design of GN&C systems for NASA human-rated and robotic spacecraft during the last four decades. This comparison gives insight into the options available for the GN&C systems of the new generation of space vehicles to be developed under NASA's Constellation Program (CxP). This study focuses on the component interconnectivity features behind cross-strapping and channelization, the two main architecting approaches for designing fault-tolerant GN&C systems. The features of these two approaches, as well as hybrid versions of these approaches, are explained. The paper also discusses their advantages and disadvantages in terms of complexity, reliability, and fault-tolerance. Several systems developed by NASA for human-rated systems and robotic systems are analyzed and compared from this perspective. The final goal of this paper is to lay the foundations for a more in-depth study to assess commonality among different GN&C architectures for the CxP.

Nomenclature

AA:	Accelerometer Assembly
C&C:	Control and Command
CxP:	Constellation Program
CEV:	Crew Exploration Vehicle
CLV:	Crew Launch Vehicle
CMG:	Control Moment Gyro
CSM:	Command and Service Module
FCC:	Flight Critical Computer
FCP:	Flight Critical Processor
FDIR:	Failure Detection, Isolation, and Reconfiguration
FT:	Fault Tolerance
GN&C:	Guidance, Navigation and Control
GPC:	General Purpose Computer
GA:	Gyroscope Assembly
GiA:	Gimbal Assembly
GPS:	Global Positioning System
ICP:	Instrumentation Control Processor
IMU:	Inertial Measurement Unit
ISS:	International Space Station
LSAM:	Lunar Surface Access Module
N_a :	Actuators redundancy level

*Post-Doctoral Associate, Department of Electrical Engineering and Computer Science, 77 Massachusetts Avenue, Room 10-082. Member AIAA.

[†]Graduate Research Assistant, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 33-409. Student Member AIAA.

[‡]Professor, MacVicar Faculty Fellow, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 33-313. Associate Fellow AIAA.

[§]Ford Professor of Engineering, Department of Aeronautics and Astronautics, and Engineering Systems Division, 77 Massachusetts Avenue, Room 33-413. Fellow AIAA.



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
47 of 102

N_c	: Computers redundancy level
N_s	: Sensors redundancy level
MDM:	Multiplexer/Demultiplexer
NE:	Network Element
NEFU:	Network Element Fifth Unit
RAS:	Rudder Actuation Subsystem
RCS:	Reaction Control System
RCSAS:	Reaction Control System Actuation Subsystem
RFAS:	Right Flap Actuation Subsystem
RRAS:	Right Rudder Actuation Subsystem
RGA:	Rate Gyro Assembly
S:	Sextant
SIGE:	Space Integrated Global Positioning System/Inertial Navigation System
SISO:	Single Input System Output
SPS:	Service Propulsion System
SPSAS:	Service Propulsion Actuation Subsystem
ST:	Scanning Telescope
TAS:	Thruster Actuation Subsystem
t :	Time
λ_a	: Actuator failure rate
λ_c	: Computer failure rate
λ_s	: Sensor failure rate

I. Introduction

The Vision for Space Exploration (VSE) of 2004 has provided NASA with a new direction for human spaceflight and exploration. In order to meet the VSE goals, NASA's CxP will have to acquire and operate a number of new human-rated systems, such as the Orion Crew Exploration Vehicle (CEV), the Crew Launch Vehicle (CLV), and the Lunar Surface Access Module (LSAM), along with other elements for crew transportation (e.g., in-space propulsion stages), as well as for lunar habitation and mobility. There will also be lunar robotic orbiter vehicles and robotic lunar landers. Commonality in exploration system hardware, and software elements offers the opportunity to significantly increase sustainability by reducing, both non-recurring and recurring cost and / or risk. The potential benefit of common GN&C avionics and flight software is considerable, not only in the initial development effort, but in validation and verification, and more importantly in the ongoing maintenance efforts and incremental upgrades that will occur over the life cycle of these spacecraft. With commonality of the onboard components of this system, there is more likelihood that ground control and communications systems could be made more common, yielding a multiplier effect. A comparative assessment of robotic and human-rated GN&C system architectural approaches is currently being performed as part of a proactive NASA Engineering and Safety Center sponsored study of CxP GN&C Commonality at the Massachusetts Institute of Technology (MIT). This study effort was driven by the observation, both on the part of NESC and MIT, that GN&C systems for exploration prominently stand out among all the future spacecraft systems, as an area where commonality might be of greatest benefit. This comparative assessment of robotic and human-rated GN&C system architectural approaches was undertaken as a fundamental step towards understanding the opportunities and limitations of GN&C commonality across the CxP flight elements. This paper documents the results of this comparative analysis yielding an understanding of the fundamental differences (historical and objective) between robotic and human-rated mission GN&C systems.

The paper is organized as follows: Section II gives some high-level background regarding similarities and differences between human-rated and robotic spacecraft. Section III of this paper is a review of architectural approaches used in fault-tolerant GN&C systems. Section IV explains the main architectural features used in GN&C systems of NASA human-rated and robotic spacecraft. Section V presents the observations drawn by the authors from comparing the approaches used in GN&C systems for both human-rated and robotic spacecraft. Concluding remarks are presented in Section VI.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft			Page #: 48 of 102

II. Top Level Comparison of Human-Rated and Robotic GN&C Systems

In this section, we briefly discuss the similarities and differences in the GN&C design process, and operational environment, for both human-rated and robotic spacecraft. We also discuss unique aspects of each type of spacecraft.

The design processes for human-rated and robotic spacecraft GN&C systems have many similarities. The design processes for both require system-level architecture analyses, trade studies, and fault tolerance and reliability analyses to properly balance mission success (risk), performance, mass, power, and cost. Both types of spacecraft are designed using similar discipline-standard analysis, modeling, and simulation techniques. For example, linear frequency domain stability analyses and time-domain non-linear performance simulations are commonly used. Due to industry consolidation, there are only a limited number of GN&C component vendors. Therefore, human-rated and robotic GN&C systems are both implemented using similar (if not identical) sensors, computer processors, and actuators.

In terms of operation, both human-rated and robotic spacecraft operate in similar environmental conditions. They both must survive demanding launch shock and vibration environments. They both must operate in harsh space radiation and thermal / vacuum environments. They both operate in many of the same mission phases, such as low Earth orbital cruise, entry, descent and landing, rendezvous, etc. Furthermore, both types of spacecraft perform some similar mission functions, such as stellar-inertial navigation, angular rate damping, attitude control, and orbital adjustment propulsive maneuvers.

Human-rated spacecraft GN&C systems differ from those for robotic spacecraft in important ways. The designer of a GN&C system for a human-rated system must always keep the physical safety of the human crew members foremost in mind. Thus, it is almost certain that the GN&C system for a human-rated spacecraft will be required to be tolerant to at least two faults (fail operational / fail safe) in order to meet overall spacecraft safety and reliability requirements. Additionally, most mission phases require an abort strategy that can remove the spacecraft (with its crew) from an unanticipated unsafe state. GN&C systems on most human-rated spacecraft to date have been required to operate over short mission durations (e.g., days to weeks), compared to the multi-year duration of robotic missions. The one obvious exception to the general mission duration rule cited above is the International Space Station (ISS). However, in the case of ISS, certain elements of the ISS GN&C system can be replaced on-orbit, such as the control moment gyros (CMGs) that control the station's attitude. Finally, verification and validation, and re-certification costs of any modified GN&C hardware and software are more burdensome for human-rated spacecraft GN&C applications than for robotic spacecraft GN&C.

There are some aspects of spaceflight unique to human-rated missions. It is likely that future human-rated spacecraft will be reused multiple times over a long multi-year life-cycle period, whereas robotic spacecraft are rarely reused. (However, in CxP, reuse of robotic spacecraft is envisioned.) Therefore, the physical, economic, and safety-related impacts of refurbishing, servicing, and/or replacing GN&C hardware on the ground after each mission must be considered early in the GN&C design process. The cockpit panel displays, monitors and alarms, as well as the hand controllers used for piloting manual inputs, which are typically used on human-rated spacecraft are non-existent on robotic spacecraft. For human-rated spacecraft, specialized GN&C training and simulation is required for both the crew and the ground operations team, whereas GN&C training is only required for the ground operations team on robotic spacecraft missions. Human-rated spacecraft are flown by pilots, whereas robotic spacecraft are not. Therefore, far more than their counterparts working on robotic spacecraft, the designers of GN&C systems for human-rated spacecraft must recognize and address the issue of mode awareness.

There are also some aspects of spacecraft systems unique to robotic spacecraft. Robotic spacecraft GN&C system designers often exploit the advantages of flying dissimilar flight hardware. For example, digital fine sun sensors are often used to backup star trackers for precision attitude determination. Magnetometers can be used for backup attitude determination, and thrusters can be used in place of magnetic torquers to unload excess reaction wheel momentum. Separate and dissimilar processors are used to host and execute digital safe hold mode control laws. Attitude control, line-of-sight pointing, and jitter control performance is often a primary design driver for many robotic science spacecraft. High-accuracy pointing is seldom if ever a requirement for human-rated vehicles.

The authors believe that the most important difference between human-rated and robotic spacecraft GN&C systems is the fault-tolerance requirement. Thus, in the remainder of this paper, we will primarily focus on fault tolerance and reliability-related aspects of GN&C systems for both human-rated and robotic spacecraft.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
49 of 102

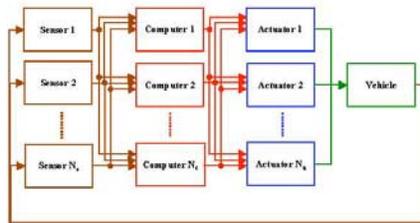


Figure 1. Cross-strapped architecture.

III. Fault-Tolerant System Architecture Approaches for GN&C Systems

Fault tolerance can be defined as the ability of a system to adapt and compensate for, in a planned and systematic way, random failures of system components that can cause the overall system to fail to perform the function for which it was designed.¹ Fault tolerance is achieved with redundancy (component or subsystem functional replication) and appropriate management of that redundancy through failure detection, isolation, and reconfiguration mechanisms. There are good references in the literature addressing these important topics of fault-tolerant systems; for example, see Ref. 2 for redundancy strategies and their management, Ref. 3 for voting algorithms, and Ref. 4 for failure detection. However, equally important is the interconnectivity between different components in the system. In a GN&C system (and in any closed-loop control system in general), there are three main classes of necessary components: sensors, computers, and actuators/actuators.

From the interconnectivity point of view, there are two major architectural philosophies used to achieve fault-tolerance in a GN&C system: cross-strapping and channelization. In the remainder of this section, we will explain in detail the main features of these two philosophies. This section will discuss these philosophies, comparing their advantages and disadvantages.

III.A. Cross-Strapped Architectures

In a fully cross-strapped architecture (Fig. 1), the output of each component is physically connected to the input of each immediate element in the control loop. Thus, every sensor output is physically connected to every computer, and every computer is connected to every actuator. To simplify the explanation further, let's consider that the vehicle in Fig. 1 can be represented by a single input system output (SISO) dynamic model. Then, every sensor ($1, \dots, N_S$) is measuring the same vehicle state variable, while every actuator ($1, \dots, N_A$) can affect the only vehicle control input. The computers ($1, \dots, N_C$) all implement the same appropriate control law to obtain the desired vehicle dynamic performance. In theory, if just one of the sensors, one of the computers, and one of the actuators is functional, there would be enough functionality to control the vehicle. Thus theoretically, the system could deliver its functionality even if $N_S - 1$ sensors failed, $N_C - 1$ computers failed, and $N_A - 1$ actuators failed.

In reality, of course, the system may not successfully tolerate this many failures. Replicating components (adding redundancy) does not ensure fault tolerance. A failed component, if not removed from the control loop, could alter the closed-loop system dynamics and cause the vehicle to become unstable and/or uncontrollable. For example, if $N_S - 2$ sensors have failed, and these failures have been detected and isolated, it will be straightforward to detect the next sensor failure, since there will be a disagreement between the good and bad sensors. However, isolating the failure to the failed sensor is more difficult without additional information, since voting can no longer be used. Built-in-Test equipment can sometimes isolate the failure, but often the probability that built-in test equipment will properly isolate the failure is not as high as would be desired. Thus, the other important aspect of fault tolerance is redundancy management, *i.e.*, the appropriate mechanisms to detect component failures, isolate those failures, and reconfigure the system (making use of the remaining redundant components / subsystems) so that the system remains functional. Therefore, the maximum level of fault tolerance is $\min\{N_S - 1, N_C - 1, N_A - 1\}$, assuming perfect failure coverage of the first $N_S - 1$, $N_C - 1$, or $N_A - 1$ sensor, computer, or actuator failures respectively. If two functional components are necessary to reliably identify the failure of a third, then the level of fault tolerance is $\min\{N_S - 2, N_C - 2, N_A - 2\}$.



Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
 50 of 102

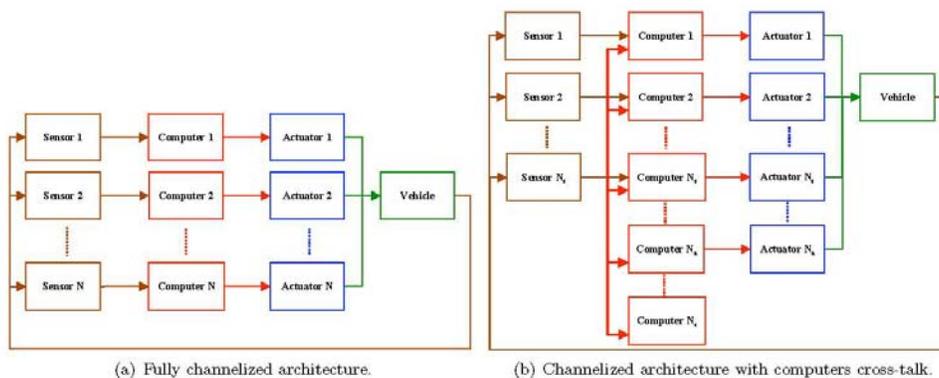


Figure 2. Channelized architectures.

III.B. Channelized Architectures

In a fully channelized architecture (Fig. 2(a)), each component output is connected to the input of a single component. For example, a sensor output ($1, \dots, N_S$) is connected to a single computer ($1, \dots, N_C$), and the output of each computer is connected to a single actuator ($1, \dots, N_A$), where $N_S = N_C = N_A = N$, to create N sensor-computer-actuator strings. As in Section III.A, we consider the case where the vehicle in Fig. 2(a) is a SISO dynamic model. Then, in theory, just one of the N strings would be sufficient for the system to deliver its functionality. It is important to note that in this approach, a single failure of a string component disables the whole string. Thus, the other two components are no longer operational even if they have not failed. Therefore, this architecture can be guaranteed to tolerate at most $N - 1$ failures before the system stops delivering its functionality. The level of fault tolerance will be $N - 2$ if it is assumed that when two operational strings are left, it is impossible to detect and isolate an additional failure.

There is another type of channelized architecture that results when there is cross-talk between the computers (Fig. 2(b)). In this case, the numbers of sensors, computers and actuators are not necessarily the same. The number of computers ($1, \dots, N_C$) is greater than or equal to the number of sensors ($1, \dots, N_S$) and/or actuators ($1, \dots, N_A$), although, as in a fully channelized architecture, each component output is connected to only one component input. Again, we consider the case that the vehicle in Fig. 2(b) is a SISO system. Then, theoretically, the system can tolerate at most $N_S - 1$ sensor failures, $N_A - 1$ actuator failures and $N_C - 1$ computer failures. Thus the maximum level of fault tolerance is $\min\{N_S - 1, N_C - 1, N_A - 1\}$, and there are scenarios where the system is still functional with up to $N_S + N_C + N_A - 3$ failures (whenever the three remaining non-failed elements are a sensor, a computer and an actuator in the same string). Again, the level of fault tolerance will be $\min\{N_S - 2, N_C - 2, N_A - 2\}$ if there is no possibility of detecting and isolating an additional sensor failure when only $N_S - 2$ sensors are non-failed, and similarly when $N_C - 2$ computers, or $N_A - 2$ actuators are non-failed.

One of the advantages of a channelized architecture with computer cross-talk is that a single sensor (or actuator) failure does not disable the other two components of the string (the actuator (or sensor) and the computer). Therefore, this architecture can tolerate more failures than the fully channelized architecture. However, the channelized architecture with computer cross-talk has the disadvantage of adding an additional layer of complexity to the system design. In order to ensure proper operation of this architecture, it is necessary to implement the necessary algorithms to cope with the Byzantine Generals Problem⁵ to ensure that all the computers are receiving the same sensor data through the computer cross-talk.

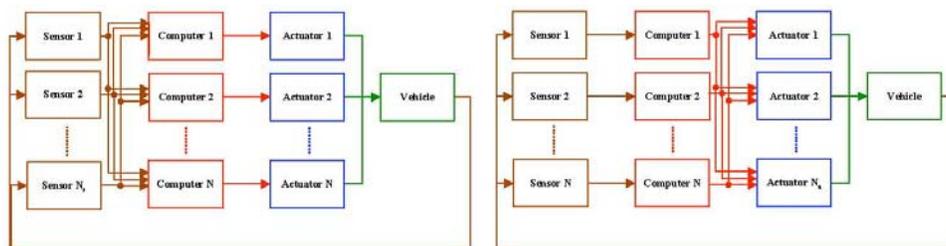
Asymmetric channelization is another advantage of channelized architectures with computer cross-talk. This means that the number of sensors, computers, and actuators does not need to be the same; it is possible to have fewer redundant actuators and sensors than computers. Therefore, if different levels of fault tolerance are demanded for each of these subsystems, it is possible to implement different levels of redundancy among sensors, computers and actuators.



Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
51 of 102



(a) Hybrid architecture: cross-strapped sensing and channelized actuation. (b) Hybrid architecture: channelized sensing and cross-strapped actuation.

Figure 3. Hybrid architectures.

III.C. Hybrid Architectures

Hybrid architectures are a blend of the two architecting approaches already discussed — cross-strapping and channelization. In the hybrid architecture displayed in Fig. 3, the sensors $(1, \dots, N_S)$ are cross-strapped to each computer $(1, \dots, N)$, but each computer independently controls a single actuator $(1, \dots, N)$. Figure 3(b) shows another type of hybrid architecture where the actuators are now cross strapped and the sensors are connected to the computers using a channelized approach. An example of a hybrid architecture of the type shown in Fig. 3(a) is discussed in Ref 6.

III.D. Comparison of Different Architectural Approaches for Fault-Tolerant GN&C Systems

Unreliability, or probability of system failure, is one metric for comparing different architectures. In this paper, we define two different unreliability estimates: the unreliability lower bound and unreliability upper bound. The first estimate, the unreliability lower bound, is computed assuming perfect failure detection and isolation of the first $N_S - 1$ sensor failures ($N_C - 1$ computer failures or $N_A - 1$ actuator failures). The second estimate, the unreliability upper bound, is computed assuming that when two components of any class (sensors, computer or actuators) are left, it is impossible to detect and isolate an additional failure of the same class. These unreliability upper and lower bound estimates are collected in Table 1. Combinatorial analysis was used to obtain these estimates⁷ with the following assumptions:

- Component failures are independent and exponentially distributed, and they are described in terms of a constant failure rate.
- All components of the same class have the same failure rate. Thus, any sensor, computer, or actuator has a failure rate of $\lambda_s, \lambda_c, \lambda_a$ respectively.
- The values of $\lambda_s t, \lambda_c t, \lambda_a t$, where t is the time the system is operational, are assumed to be small enough that
 1. $1 - e^{-\lambda_s t} \approx \lambda_s t$, $1 - e^{-\lambda_c t} \approx \lambda_c t$, and $1 - e^{-\lambda_a t} \approx \lambda_a t$; and
 2. the unreliability function is approximately equal to the sum of probabilities of the smallest-size cut-sets (sets of components which, if failed, the system fails).

The unreliability lower bound estimates collected in Table 1 indicate that, for the same level of redundancy of each class of component ($N_s = N_c = N_a = N$), the fully cross-strapped architecture is the most reliable of all architectures, whereas the fully channelized architecture is the least reliable. This can be easily inferred from the inequalities

$$(\lambda_s t)^N + (\lambda_c t)^N + (\lambda_a t)^N < [(\lambda_s + \lambda_c)t]^N + [(\lambda_c + \lambda_a)t]^N - (\lambda_c t)^N < [(\lambda_s + \lambda_c + \lambda_a)t]^N, \quad (1)$$

$$(\lambda_s t)^N + (\lambda_c t)^N + (\lambda_a t)^N < [(\lambda_s + \lambda_c)t]^N + (\lambda_a t)^N < [(\lambda_s + \lambda_c)t]^N + [(\lambda_c + \lambda_a)t]^N - (\lambda_c t)^N, \quad (2)$$

$$(\lambda_s t)^N + (\lambda_c t)^N + (\lambda_a t)^N < (\lambda_s t)^{N_s} + [(\lambda_c + \lambda_a)t]^N < [(\lambda_s + \lambda_c)t]^N + [(\lambda_c + \lambda_a)t]^N - (\lambda_c t)^N. \quad (3)$$



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
52 of 102

Table 1. Comparison of unreliability upper and lower bound estimates for different fault-tolerant architecting approaches.

Architecture	Unreliability lower bound	Unreliability upper bound	Reliability ranking
Fully Cross-strapped	$(\lambda_s t)^{N_s} + (\lambda_c t)^{N_c} + (\lambda_a t)^{N_a}$	$N_s(\lambda_s t)^{N_s-1} + N_c(\lambda_c t)^{N_c-1} + N_a(\lambda_a t)^{N_a-1}$	1
Fully Channelized. $N_s = N_c = N_a = N$	$[(\lambda_s + \lambda_c + \lambda_a)t]^N$	$N[(\lambda_s + \lambda_c + \lambda_a)t]^{N-1}$	5
Channelized with computer cross-talk. $N_s = N_c = N_a = N$	$[(\lambda_s + \lambda_c)t]^N + [(\lambda_c + \lambda_a)t]^N - (\lambda_c t)^N$		4
Hybrid with cross-strapped sensors. $N_c = N_a = N$	$(\lambda_s t)^{N_s} + [(\lambda_c + \lambda_a)t]^N$	$N_s(\lambda_s t)^{N_s-1} + N[(\lambda_c + \lambda_a)t]^{N-1}$	2(3)
Hybrid with cross-strapped actuators. $N_s = N_c = N$	$[(\lambda_s + \lambda_c)t]^N + (\lambda_a t)^{N_a}$	$N[(\lambda_s + \lambda_c)t]^{N-1} + N_a(\lambda_a t)^{N_a-1}$	3(2)

Similar relations can be established for the unreliability upper bounds, but the reliability ranking remains the same. This reliability ranking is summarized in column 4 of Table 1. It is important to note, however, that there are a number of factors that were not taken into account that may impact this ranking. These factors are the failure coverage probability, which it was assumed to be one for either $N - 2$ or $N - 1$ component failures of the same class; and the assumption that component failures are independent, thus precluding the possibility of common mode failures. These factors must be included if a more detailed analysis of a particular architecture is performed. Furthermore, it assumes that the cross strapping mechanisms are perfectly reliable, which is generally not the case.

Table 2 shows the ability of each architecture to detect and isolate failures. It also compares the testability of failure detection, isolation and reconfiguration (FDIR) mechanisms and the complexity of developing each architecture. In terms of failure detection, all the architectures are equivalent if the appropriate voting algorithms and built-in-test mechanisms are used. In terms of failure isolation, fully cross-strapped architectures are better than others since, in a properly designed cross-strapped architecture, a single component failure can be isolated without disabling other components. This is different than for fully channelized architectures, where a single component failure will disable all the other components in the same string.

Looking at failure containment, cross-strapping is good for isolating individual failures, but it makes the architecture more prone to common mode failures. In this regard, a fully channelized architecture is more immune to common mode failures; a single failure will take out one sensor-computer-actuator string, but is unlikely to affect the remaining strings. The same is true for a fully channelized architecture with computer cross-talk.

In terms of FDIR testability, fully channelized architectures are easier to test than fully-cross strapped architectures, since each string can be tested individually, while a fully-cross strapped system needs to be tested as a whole. This has a direct impact on the complexity faced while developing and implementing each architecture, since fully cross-strapped architectures are more complex to develop and implement than channelized architectures. As mentioned earlier, the exchange of information between computers adds an additional layer of complexity, since it is necessary to implement the appropriate algorithms to handle the Byzantine Generals Problem.

Table 2. Comparison of failure detection, isolation and containment for different fault-tolerant architecting approaches.

Architecture	Failure detection	Failure isolation	Failure containment	FDIR testability	Complexity
Fully Cross-strapped	High	High	Moderate	Low	High
Fully Channelized	High	Low	High	High	Low
Channelized with computer cross-talk	High	High	Moderate	Moderate	High
Hybrid with cross-strapped sensors	High	Moderate	Moderate	Moderate	Moderate
Hybrid with cross-strapped actuators	High	Moderate	Moderate	Moderate	Moderate



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
53 of 102

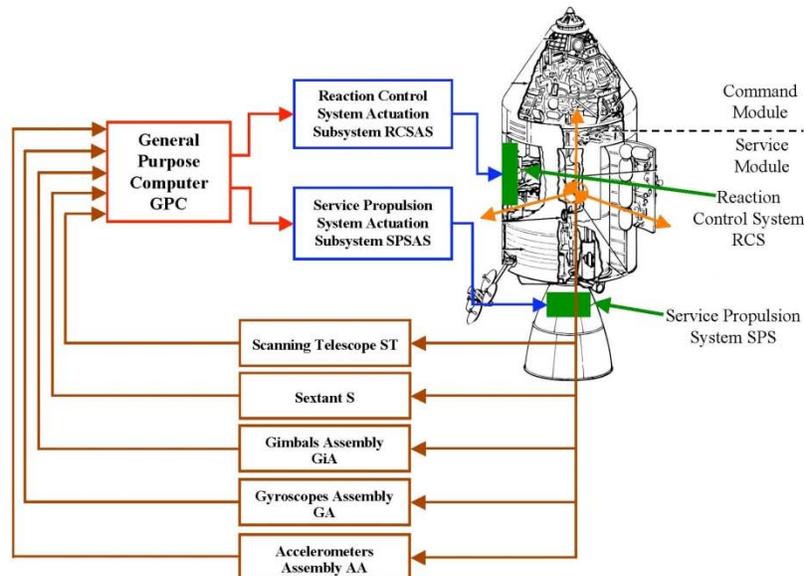


Figure 4. Partial description of the Apollo Command and Service Module GN&C architecture: computer, sensors, reaction control, and service propulsion systems actuation subsystem interconnectivity. Diagram credit: NASA.

It is important to note that there are additional considerations in terms of cost, weight, and volume that we did not discuss here. Often, it is not immediately clear what the best architectural approach is for a particular GN&C system. For example, consider that the configuration and the complexity of the GN&C components will strongly influence spacecraft power needs. Thus, a structured system optimization design process is necessary to formulate a rationale for allocating scarce resources, such as power and mass. Therefore, it is necessary to carry out quantitative analyses and trade studies for each design and mission application to find the best solution in terms of performance, reliability, cost, weight, or any other important metrics.

IV. A Survey of GN&C Systems for Human-Rated and Robotic Spacecraft

The purpose of this section is to explain the main features of the GN&C system architectures of several human-rated and robotic NASA spacecraft. Rather than giving a detailed description of each architecture, this section focuses on the different design approaches used to achieve fault tolerance. For each of the described systems, several references are provided for the reader interested in further details.

IV.A. Apollo Command and Service Module (CSM): Single String

The Apollo CSM was a largely single-string system, yet its lack of redundancy should not be confused with a lack of fault tolerance. Subsystem specialists on the ground were constantly on the lookout for problems. Although the computer onboard the CSM could, and would, extrapolate the state vector, this extrapolation usually only occurred during time-critical mission phases and when the CSM was unable to communicate with Earth. The majority of state vector computation came from the larger and more accurate computer on the ground at the Manned Space Flight Network. This added accuracy improved the performance of the Apollo CSM. Furthermore, sensors and actuators on the Apollo CSM were rigorously tested, as were the integrated circuits used in its computer (which, to achieve improved reliability, were constructed using only one type of logical gate, see Ref 8 for details). Due to this testing, sensors, actuators, and computer were extremely unlikely to fail.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
54 of 102

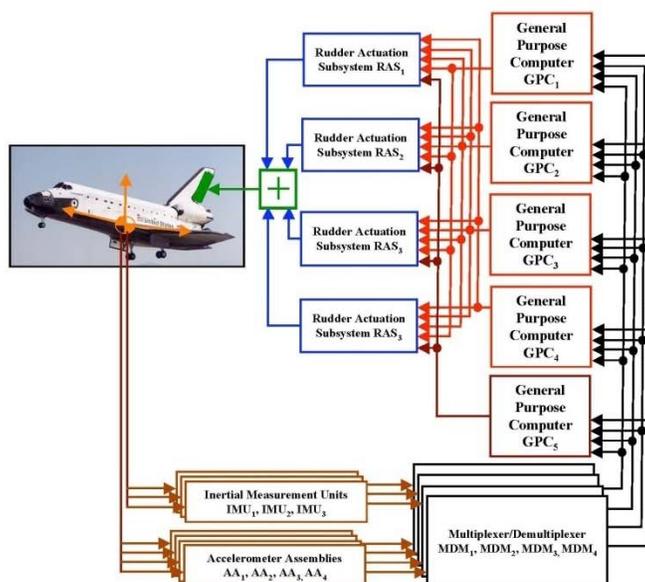


Figure 5. Partial description of the Space shuttle GN&C architecture: computers, sensors, and control surface actuation subsystems interconnectivity. Photo credit: NASA Dryden Flight Research Center.

Figure 4 shows a small portion of the Apollo CSM GN&C system architecture. The primary sensor for the Apollo CSM GN&C system was the IMU, part of the inertial subsystem, and its function was to measure altitude, location, and attitude. It included a gyroscope assembly (GA) as well as an accelerometer assembly (AA). Its platform was mounted in a gimbals assembly (GiA) such that translational accelerations and rotation-rates could be measured in all six degrees of freedom. Drift errors, which occurred due to the mechanical nature of the device, were corrected periodically by collecting data with the two sensors in the optical subsystem: the scanning telescope (ST) and the sextant (S).⁹

There was only one general purpose computer (GPC) on the Apollo CSM. It ultimately accepted signals from all sensors (ST, S, GiA, GA, and AA) and hosted the necessary software to compute, based on sensor measurements, the spacecraft attitude. Based on the current attitude, the GPC would compute the appropriate control commands for the service propulsion system actuation subsystem (SPSAS), to be used by the service propulsion system (SPS) and also the appropriate commands for the reaction control system actuation subsystem (RCSAS). The SPS was responsible for pitch and yaw control during powered flight regimes. The main engine of the SPS was gimballed and the precise direction and duration of main engine firings had to be determined by the computer. The RCS was the system responsible for the spacecraft attitude control during unpowered flight regimes.

A form of fault tolerance can be found in the Apollo CSM GN&C system. The system was designed so that each subsystem (inertial, optical and computer) could be operated independently during an emergency or backup mode. Therefore, the failure of any one subsystem would not disable the entire GN&C system.¹⁰

IV.B. Space Shuttle: Fully Cross-Strapped Architecture

Two of the requirements imposed during the design of the space shuttle were that the avionics system should remain fully operational after any single failure, and fully capable of a safe return to Earth after any two failures.¹¹ This meant that any safety-critical onboard system (including the GN&C system) ought to be two-fault-tolerant. Additionally, voting strategies were preferred as a means of failure detection. Thus, to achieve these design requirements, a fully cross-strapped philosophy, with quadruple redundant copies of almost every single component, was used in the design.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
55 of 102

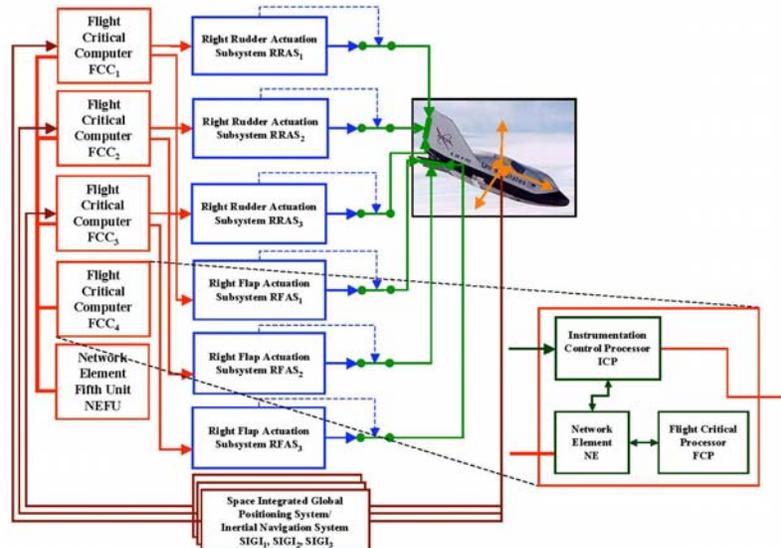


Figure 6. Partial description of the Crew Return Vehicle (X-38 V201) GN&C architecture: computers, sensors, and control surface actuation subsystems interconnectivity. Photo credit: NASA Dryden Flight Research Center.

Figure 5 shows a small portion of the shuttle GN&C system architecture. All the necessary sensors for performing GN&C are triple or quadruple redundant; e.g., there are three Inertial Measurement Units (IMU₁, IMU₂ and IMU₃) and four accelerometer assemblies (AA₁, AA₂, AA₃ and AA₄). The information measured by the sensors is gathered by several Multiplexer / Demultiplexer interfaces (MDM₁, MDM₂, MDM₃ and MDM₄) before it is sent to the computers. Every Multiplexer/Demultiplexer is cross-strapped to five general purpose computers (GPC₁, GPC₂, GPC₃, GPC₄ and GPC₅). Four of these computers (GPC₁-GPC₄) perform the main GN&C functions, and the fifth one (GPC₅) is a backup only used to abort the mission if a common failure mode takes down the other four (GPC₁-GPC₄) at the same time. Each GPC implements mid-value selection voting algorithms as a means to detect and mask sensor failures. A failed sensor is taken out of the control loop so it will not cause detection problems when a second sensor fails. Each GPC has exact copies of the GN&C algorithms so, in nominal conditions, they should produce the same output commands for the different actuation subsystems. Quadruple redundancy is used for the actuation subsystems of the orbiter engines' thrust vector control, and aerodynamic control surfaces (Fig. 5 shows the rudder actuation subsystems RAS₁, RAS₂, RAS₃ and RAS₄). Each actuation subsystem receives commands from the four GPCs and issues a command to a four-port hydraulic valve (attached to the main power actuator) that will act as a "mechanical voter". If one of the four-port valve's input commands is erroneous, either by a failure in a GPC or by a failure in one of the actuation subsystems, then the other three will override the erroneous one and produce the right command to the main power actuator. A failed actuation subsystem is taken out of the control loop so a second failure can be overridden by the two remaining "healthy subsystems". In summary, any GPC failure or any actuation subsystem failure is detected and isolated by the "mechanical voter" four-port hydraulic valve. The reader is referred to Ref. 11 for a more comprehensive explanation of all the features of this system.

IV.C. Crew Return Vehicle (X-38 V201): Channelized Architecture

A channelized architecture was used to design the X-38 V201's GN&C system. Figure 6 shows a portion of that system. The GN&C computer was designed to be two-fault tolerant,¹² however, the GN&C sensors and the control surface actuators were designed with an approach that allows full coverage of single failures and imperfect coverage of second failures.^{13,14}



Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
 56 of 102

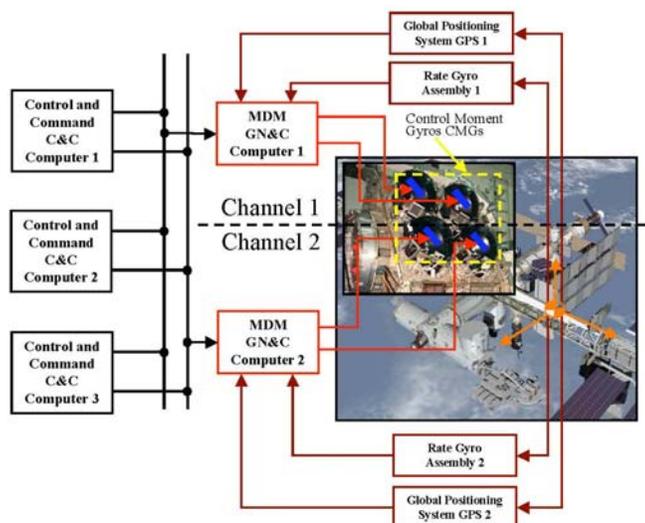


Figure 7. Partial description of the International Space Station GN&C architecture: computers, sensors, and control moment gyros interconnectivity. Photo Credits: (CMGs) Boeing Integrated Defense Systems, (ISS) NASA.

Three redundant space Integrated Global Positioning Systems / Inertial Navigation Systems (SIGI₁, SIGI₂ and SIGI₃) provide spacecraft position, velocity, acceleration, altitude, and attitude rates. The two-fault-tolerant computer is composed of four Flight Critical Computers (FCC₁, FCC₂, FCC₃ and FCC₄) and a Network Element Fifth Unit (NEFU).¹⁵ Each FCC receives a single set of data from a single SIGI, i.e., the SIGIs are not cross-strapped to each computer. To illustrate this, in Fig. 6, it can be seen that the SIGI₁ is connected only to FCC₁, and similarly, FCC₂ and FCC₃ only receive readings from SIGI₂ and SIGI₃ respectively.

The architecture of each FCC is depicted on the right side of Fig. 6. In addition to a Network Element (NE) card, each FCC contains two processors: a Flight Critical Processor (FCP) and an Instrumentation Control Processor (ICP). The NEFU only contains an ICP and a NE, and it was added to achieve the two-fault-tolerance requirement. The ICP of each FCC is the I/O processor, which gathers information from the SIGI directly connected to it, and also outputs control commands to the control surface actuation subsystems and other GN&C actuation subsystems. The ICP of each FCC passes its own SIGI readings to the FCC-NE and then these readings are sent out to other FCC-NE (and also to the NEFU-NE). The exchange of information is done in two rounds to solve the Byzantine Generals Problem. After this exchange of data each FCC-NE have all three SIGI readings and a voting process is carried out to determine the correct measurements. Once each FCC-NE has determined the correct measurements, these are passed to the corresponding FCP. Each FCP contains the GN&C algorithms, and will compute the appropriate control commands for the actuation subsystems. After this, each FCP sends the computed commands to its own NE and another (one round) exchange of data between each FCC-NE takes place. Then each FCC-NE votes the correct commands and these commands are passed to the corresponding actuation subsystems. This arrangement allows separation of the GN&C algorithm development from the FDIR algorithm implementation.¹²

Triple redundant actuation subsystems are used to position each control surface. There are two rudders and two body flaps. Figure 6 displays the right rudder and right flap and their corresponding actuation subsystems. Each control surface actuation subsystem receives a command from a single computer, e.g., RRAS₁ receives a command from FCC₁, RRAS₂ from FCC₂, and RRAS₃ from FCC₃. Each actuation subsystem is connected to the respective control surface through a clutch. If one of them fails, the other two can overpower it and the faulty actuation subsystem will be removed by disengaging its clutch. Built-In-Test failure detection is used to detect a second actuation subsystem failure.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
57 of 102

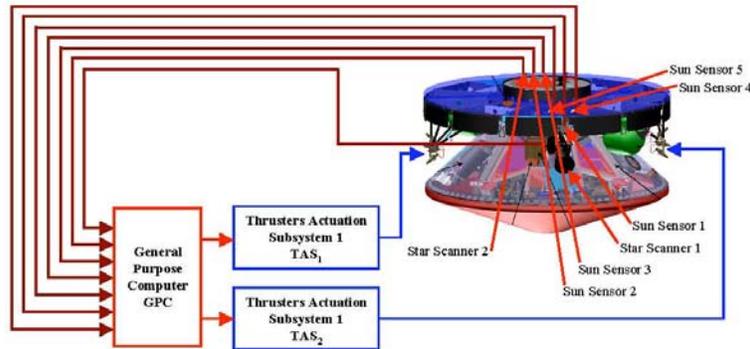


Figure 8. Partial description of the Mars exploration rovers cruise stage GN&C architecture: computers, sensors, and actuation subsystems interconnectivity. Drawing Credits: NASA Jet Propulsion Laboratory.

IV.D. International Space Station (ISS) US Segment: Channelized Architecture

Channelization was used to design the GN&C system architecture of the ISS US segment. The channelization concept used in the ISS comes from the design baseline of its predecessors, the Space Station Freedom and the International Space Station Alpha.^{16, 17}

Figure 7 shows a high-level description of the ISS US segment GN&C system. Three dual-pair-processor computers perform the control and command function of the ISS. These three computers can independently control any of the two GN&C system channels through a Multiplexer/ Demultiplexer GN&C Computer that receives information about the ISS attitude from a Ground Positioning System (GPS) and from a Rate Gyro Assembly (RGA). Each MDM will generate the appropriate control commands for a pair of Control Moment Gyros (CMGs) that control the attitude of the station.

IV.E. Mars Exploration Rovers Cruise Stage: Single String

Like the Apollo CSM, the cruise stage of the Mars Exploration Rovers was largely single string. There is some redundancy in the sensors. The star scanner has a backup system and there are five sun sensors. There is only one computer, but there are two major fault protection algorithms for command loss and battery charge control.¹⁸ Looking at the actuators, there was some redundancy here as well. The eight 1 lbf thrusters, organized in two clusters of four, were all valved to provide redundancy in case of a leaky thruster.¹⁹ The transportation system of the Mars Exploration Rovers is very similar to that of Mars Pathfinder. See Fig. 8 for a top-level diagram and schematic of the cruise stage.

Table 3. Architecting approaches and fault tolerance level for GN&C systems for human-rated and robotic spacecraft

Spacecraft	Architecture	Computer FT level	Sensors FT level	Actuators FT level
Apollo CSM	Single String	0	0	0
Shuttle	Cross Strapped	2	2	2
ISS	Channelized	1	1	2
X-38 V201	Channelized with computer cross-talk	2	1 (limited second)	1 (limited second)
CEV	TBD (Cross-strapped or single String)	TBD	TBD	TBD
Mars Pathfinder	Single String	0	0	0
Mars Exploration Rover	Single String	0	1	1
Phoenix	Single String	0	0	0

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 58 of 102	

V. Observations

The second column of Table 3 displays the architectural approaches for GN&C systems of different human-rated and robotic spacecraft. Columns 3–5 display a breakdown of the levels of fault tolerance required for GN&C subsystems, specifically, sensors, computers, and actuators.

The human-rated spacecraft discussed in this paper were designed using different architectural approaches. Apollo CSM was essentially single string, with limited fault tolerance. If a failure occurred, the mission would be aborted by means of dissimilar backup mechanisms, with degraded performance compared to the primary system, and the spacecraft and crew could be safely brought back to Earth. The lack of fault tolerance in Apollo can be explained by the fact that fault-tolerant computing was a relatively new and emerging field when the Apollo program was being developed. It was not until 1967 (well into the Apollo Program's hardware development phase) that the concept of fault-tolerant computers was formalized.²⁰ Alternatively, to ensure a high-level of reliability, the Apollo program used a high-quality-parts production process.

By the time the space shuttle was under development, the fault tolerance field was more mature. Furthermore, maneuvers such as unpowered landing and entry through final approach demanded stringent performance requirements not just on the main systems, but on the backup systems as well. Thus, backup mechanisms with degraded performance were no longer acceptable. This requirement, together with the fact that aborting a mission after one failure was unacceptable, led to the introduction of a fully cross-strapped, two-fault tolerant architecture for all flight critical subsystems.¹¹

A channelized architectural approach was used for designing the ISS GN&C system. The requirements on the FDIR mechanisms are not as stringent for ISS as for the shuttle, and therefore the GN&C architecture is only one-fault tolerant. The requirements are less stringent because the ISS has slow dynamics, so real-time failure detection and isolation are not as critical as they are in the space shuttle, where an undetected failure during reentry could be catastrophic. Furthermore, the station crew can perform repair activities in sensors and computers without additional ground support.

The X-38 program was meant to develop a reusable crew return vehicle (CRV) for the ISS. A channelized architecture with computer cross-talk was used for the GN&C system. The computer system is two-fault tolerant, while one-fault tolerance, with limited second failure coverage, was implemented for sensors and actuators. The limited fault tolerance of the primary navigation and control system is supplemented by a steerable parafoil, which provides a dissimilar backup mechanism for controlling the spacecraft attitude after reentry. Furthermore, the space shuttle was responsible for delivering the CRV to the ISS. Thus, one additional design requirement was to comply with the shuttle payload weight and size requirements. This may have also had an impact on the level of redundancy used for controlling the position of the control surfaces.

The crew exploration vehicle (CEV) is still under development, and no final decision has been made on the design baseline for the GN&C system. A summary of the GN&C reference design architecture as of 2006 can be found in Ref. 21. The architecture of the CEV GN&C system is still evolving, but it appears likely that a cross-strapped architecture will be implemented. This architecture will have a yet-to-be determined number of computers implementing lock-step processors (self-checking pairs) to achieve a high-level of failure coverage. Thus, no voting mechanisms will be used for computer failure detection.

Single-string architectures have been used for Mars robotic spacecraft with zero computer fault tolerance, and limited fault tolerance at the sensor and actuator level. Two drivers for using single-string architectures are the stringent weight and volume requirements for these missions. Additionally, as is the case in the Mars Exploration Rover mission, fault tolerance was achieved by sending two separate spacecraft, each with its own rover.

The architectural philosophies that have been used for designing the GN&C systems used for NASA's various human-rated spacecraft over the years are substantially different. This is likely due to two factors. First, there are gaps of many years, and in some cases decades, between the design cycles of NASA's human-rated spacecraft systems. As a result of these gaps, NASA and its industry partners must periodically re-learn the process of architecting human-rated spacecraft. Second, rapid technology developments since the 1960s, especially in electronics and avionics, have strongly influenced the design process, resulting in significant architectural changes over the last four decades.

On the other hand, robotic missions of the same class appear to have a very similar architectural basis. For example, the Mars missions, such as Pathfinder, and the Mars Exploration Rovers have similar architectures. This similarity is likely due to several factors. First, the development cycles are shorter compared to human-

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 59 of 102	

rated systems. Second, because of the shorter time between development cycles, there is less technological change between two designs. Third, a wide variety of mission-unique robotic spacecraft GN&C architectures are designed, implemented, and flown each year by NASA and the same teams of contractors. The result is that NASA and its contractors have a substantial and diverse (by mission class) GN&C engineering experience base for robotic spacecraft applications. Thus, legacy designs are often adopted for robotic missions. Finally, it should be noted sometimes there are few viable choices for GN&C system components. For example, Pathfinder, Mars Polar Lander, etc., all use the same single board CPU, in part due to the expense of modifying the layout of a commercial chip to make it tolerant to the space environment.

As would be expected, the GN&C architectures for robotic spacecraft are simpler than for human-rated missions in terms of fault tolerance. In human-rated missions, crew safety is paramount, thus imposing a higher requirement on fault tolerance than in robotic missions, even at the expense of cost, weight, and complexity. On the other hand, significantly more risk is acceptable in robotic missions. Furthermore, risk can sometimes be reduced in robotic missions by an extreme form of channelization, namely, flying multiple, non-redundant spacecraft.

VI. Concluding Remarks

This paper summarizes the first steps in our study of commonality in GN&C systems, for both human-rated and robotic spacecraft. As discussed, the fault tolerance and reliability requirements are the main factors driving the architectural differences between the GN&C systems of human-rated and robotic spacecraft. We defined, from a component interconnectivity point of view, the two main philosophies — cross-strapping and channelization — used to achieve fault tolerance, explaining advantages and disadvantages of each of the resulting architectural design approaches for GN&C systems. This definition helped us to classify the GN&C systems of different NASA spacecraft, and to understand some of the driving factors that led the design teams to choose a particular architecture.

Future work will include a more detailed understanding and analysis of the options available to implement the three main GN&C subsystems, namely sensor, computer, and actuator subsystems. This analysis will help us to understand the options for commonality at the subsystem level. Based on the high-level architectural analysis presented in this paper, and the more detailed work to be conducted at the subsystem level, it will be possible to enumerate numerous feasible architectures for the GN&C systems of the new generation of space vehicles to be developed under NASA's CxP. The ultimate task of the ongoing work will be to carry out a more detailed analysis of performance, reliability, and commonality of this set of architectures, to better inform the design of GN&C systems for the CxP vehicles.

Acknowledgments

The authors wish to acknowledge the encouragement and contributions to this paper from Neil Dennehy of the NASA Engineering and Safety Center, especially on the top-level comparison of the two types of spacecraft GN&C systems as discussed in Section II. The authors would like to thank also John West and Joel Busa at the Charles Stark Draper Laboratory for fruitful discussions of this research. Gregory W. Vajdos from Boeing Integrated Defense Systems and Susan Gomez provided information on the International Space Station GN&C system, which was also appreciated. Robert Rasmussen and Robert Manning of NASA Jet Propulsion Laboratory provided valuable information about the robotic Mars missions.

References

- ¹Gai, E. and Adams, M., "Measures of Merit for Fault-Tolerant Systems," Tech. Rep. CSDL-P-1752, The Charles Stark Draper Laboratory, Cambridge, MA, 1983.
- ²Hammett, R., "Design by Extrapolation: An Evaluation of Fault-Tolerant Avionics," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 17, No. 4, April 2002, pp. 17–25.
- ³Pahami, B., "Voting Algorithms," *IEEE Transactions on Reliability*, Vol. 43, No. 4, December 1994, pp. 617–629.
- ⁴Willsky, A., "A Survey of Design Methods for Failure Detection Systems," *Automatica*, Vol. 12, No. 6, November 1976, pp. 601–611.
- ⁵Laprie, J., editor, *Dependability: Basic Concepts and Terminology*, Springer-Verlag, New York, NY, 1991.
- ⁶Domínguez-García, A., Kassakian, J., Schindall, J., and Zinchuk, J., "On the Use of Behavioral Models for the Integrated Performance and Reliability Evaluation of Fault-Tolerant Avionics Systems," *Proceedings of DASC 2006*, Portland, OR, 2005.



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
60 of 102

⁷Høyland, A. and Rausand, M., *System Reliability Theory*, John Wiley and Sons, New York, NY, 1994.

⁸Williamson, M., "Aiming for the Moon: the Engineering Challenge of Apollo," *Engineering Science and Education*, Vol. 11, No. 5, October 2002, pp. 164-172.

⁹Holley, M., Swingle, W., Bachman, S., Leblanc, C., Howard, H., and Biggs, H., "Apollo Experience Report - Guidance and Control Systems: Primary Guidance, Navigation, and Control System Development," Tech. Rep. NASA TN D-8227, National Aeronautics and Space Administration, Washington, DC, May 1976.

¹⁰Wilson, R., "Apollo Experience Report - Guidance and Control Systems," Tech. Rep. NASA TN D-8249, National Aeronautics and Space Administration, Washington, DC, June 1976.

¹¹Hanaway, J. and Moorhead, R., "Space Shuttle Avionics System," Tech. Rep. NASA SP-504, National Aeronautics and Space Administration, Washington, DC, 1989.

¹²Kouba, C., Buscher, D., Busa, J., and Beilin, S., "The X-38 Spacecraft Fault-Tolerant Avionics System," *Proceedings of the Military and Aerospace Programmable Logic Device International Conferences*, Washington, DC, 2003.

¹³Bedos, T. and Anderson, B., "X-38 V201 Avionics Architecture," Tech. Rep. NASA-20000086667, National Aeronautics and Space Administration, Houston, TX, February 1999.

¹⁴Goodman, J., "GPS Lessons Learned from the International Space Station, Space Shuttle and X-38," Tech. Rep. NASA/CR-2005-213693, National Aeronautics and Space Administration, Houston, TX, November 2005.

¹⁵Racine, R., Leblanc, M., and Beilin, S., "Design of a Fault-Tolerant Parallel Processor," *Proceedings of the Digital Avionics Systems Conference*, Irvine, CA, 2002.

¹⁶Babcock, P., "Channalization: Two-Fault Tolerant Attitude Control function for the Space Station Freedom," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 11, No. 5, May 1996, pp. 9-22.

¹⁷Smith, J., Suchting, S., McDonald, M., and Schikner, J., "Avionics Architecture for the U.S. Segment of the International Space Station Alpha," *Proceedings of the 10th Computing in Aerospace Conference*, San Antonio, TX, 1995.

¹⁸Muirhead, B., "Mars Pathfinder Flight System Design And Implementation," *Proceedings of the IEEE Aerospace Conference*, Snowmass at Aspen, CO, 1996.

¹⁹Muirhead, B., "Mars Pathfinder Flight System Integration and Test," *Proceedings of the IEEE Aerospace Conference*, Snowmass at Aspen, CO, 1997.

²⁰Aviziens, A., "Design of Fault-Tolerant Computers," *Proceedings of the Fall Joint Computer Conference, AFIPS Conference*, Washington, DC, 1967, pp. 733-743.

²¹Tamblyn, S., Hinkel, H., and Saley, D., "NASA CEV Reference GN&C Architecture," *Proceedings of the 30th Annual American Astronautical Society Guidance and Control Conference*, Breckenridge, CO, 2007.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft			Page #: 61 of 102

Appendix B. A Comparison of Fault-Tolerant GN&C System Architectures Using the Object Process Network (OPN) Modeling Language

A Comparison of Fault-Tolerant GN&C System Architectures Using the Object Process Network (OPN) Modeling Language

Gregor Z. Hanuschak¹

Massachusetts Institute of Technology, Cambridge, MA 02139

Nicholas A. Harrison²

Charles Stark Draper Laboratory, Cambridge, MA 02139-4307

Edward F. Crawley³ and Steven R. Hall⁴

Massachusetts Institute of Technology, Cambridge, MA 02139

Alejandro D. Dominguez-Garcia⁵

University of Illinois at Urbana-Champaign, Urbana, IL 61801

John J. West⁶

Charles Stark Draper Laboratory, Cambridge, MA 02139-4307

and

Cornelius J. Dennehy⁷

NASA Engineering & Safety Center, Greenbelt, MD, 20771

This paper summarizes the final results of a study analyzing different Guidance, Navigation and Control (GN&C) architectural approaches for fault tolerance in National Aeronautics and Space Administration's (NASA's) crewed and robotic exploration space systems. GN&C systems were decomposed into simple building block subunits of sensors, computers, and actuators and various forms of subunit interconnection were defined for investigation. The resulting subunit/interconnection construct was used as a top-level abstraction for building candidate GN&C system architectures. This model was implemented using Massachusetts Institute of Technology's (MIT's) Object Process Network (OPN) modeling language in order to more easily enumerate possible architectures and ultimately identify which of these architectures have optimal properties. Dual and triple redundant GN&C system architectures, employing different classes of components, were modeled using the OPN language. The model assumed perfect coverage – 100-percent accuracy in detecting and isolating a failure. Within the constraints of the model, all possible architectures were rigorously enumerated and the weight/reliability trade-offs of cross-strapping components and using more than one type of component were assessed. The study results indicate it is possible to produce nearly all potentially optimal GN&C architectures using generic connections between low-reliability components. The identified optimal architectures reveal a preference to increase GN&C system redundancy of lighter, less reliable components rather than using smaller numbers of more reliable, heavy components.

¹ Graduate Research Assistant, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 33-409, and Student Member AIAA.

² Avionics Systems Engineer, Space Systems Department, 555 Technology Square.

³ Ford Professor of Engineering, Department of Aeronautics and Astronautics, and Engineering Systems Division, 77 Massachusetts Avenue, Room 33-413, and Fellow AIAA.

⁴ Professor, MacVicar Faculty Fellow, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 33-313, and Associate Fellow AIAA.

⁵ Assistant Professor, Department of Electrical and Computer Engineering, 1406 W. Green Street, MC-702, and Member AIAA.

⁶ Program Manager, Space Systems Department, 555 Technology Square.

⁷ NASA Technical Fellow for GN&C, NESC, Goddard Space Flight Center, Mail Code 590.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft			Page #: 62 of 102

Nomenclature

CEV	= Crew Exploration Vehicle
CLV	= Crew Launch Vehicle
CxP	= Constellation Program
GN&C	= Guidance, Navigation and Control
IMU	= Inertial Measurement Unit
MIT	= Massachusetts Institute of Technology
NASA	= National Aeronautics and Space Administration
NESC	= NASA Engineering and Safety Center
OPN	= Object Process Network

I. Background

At the core of NASA's future space exploration is a return to the Moon, where we will build a sustainable long-term human presence. As the Space Shuttle approaches retirement and the International Space Station nears completion, NASA's Constellation Program (CxP) is designing and developing the next fleet of American space-faring vehicles to bring astronauts back to the Moon, and possibly to Mars and beyond. In order to meet their exploration goals, NASA's CxP will have to acquire and operate a number of new human-rated systems, such as the Orion Crew Exploration Vehicle (CEV), the Ares-1 Crew Launch Vehicle (CLV), and the Altair Lunar Lander, along with other elements for crew transportation (e.g., in-space propulsion stages), lunar habitation, and mobility. Robotic systems will include lunar robotic orbiter vehicles and robotic lunar landers. Commonality in exploration system hardware, and software elements offers the opportunity to significantly increase sustainability by reducing, both nonrecurring and recurring cost and/or risk. In particular the potential benefit of common GN&C avionics and flight software is considerable, not only in the initial development effort, but in validation and verification, and more importantly in the ongoing maintenance efforts and incremental upgrades that will occur over the life cycle of these exploration spacecraft. With commonality of the onboard components of this system, there is more likelihood that ground control and communications systems could be made more common, yielding a multiplier effect. This paper summarizes the final results of a comparative assessment of robotic and human-rated GN&C system architectural approaches. This study was performed by a combined MIT and Draper Laboratory team as part of a proactive GN&C "discipline-advancing" activity sponsored by the NASA Engineering and Safety Center (NESC).

This study effort was primarily driven by the observation, both on the part of NESC and MIT, that GN&C systems for exploration prominently stand out among all the future spacecraft systems, as an area where commonality might be of greatest benefit. This comparative assessment of robotic and human-rated GN&C system architectural approaches was undertaken as a fundamental step towards understanding the opportunities and limitations of GN&C commonality across the CxP flight elements.

II. Introduction

CxP has created a need to develop new robotic and human-rated space systems. In an attempt to influence the design of the most collectively reliable and cost-efficient systems possible, the NESC sponsored a commonality study for GN&C systems through the MIT and Draper Laboratories. By modeling, enumerating, and comparing simplified GN&C architectures using simple metrics, this resulting paper presents sound reasoning for making certain architectural choices which, when implemented, would further these reliability and cost-efficiency goals.

In the 2007 AIAA paper, "A Comparison of GN&C Architectural Approaches for Robotic and Human-Rated Spacecraft" (Ref. 2), different architectural approaches for fault tolerance in guidance, navigation, and control (GN&C) systems were analyzed at the topmost level. The study broke down the GN&C systems into simple subunits, i.e., sensors, computers, and actuators, and analyzed how the components were interconnected. This paper expands upon the previous 2007 paper written by the authors. It uses the previous paper's subunit/interconnection construct as a top-level abstraction for building a preliminary model of GN&C system architectures.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 63 of 102	

Although never before used to model GN&C system architectures, the OPN modeling language was used with great success to model mission and hardware architectures (see Ref. 3 and Ref. 5 for details). OPN is a visual and computable meta-language that assists with systems architecting tasks. OPN is typically used to describe and partition the space of architectural alternatives, generate and enumerate the set of instances of feasible system models, and then simulate and order the performance metrics of each model. This language combines visual representation on Pareto plots with mathematical modeling and provides a modeling framework in which it is relatively easy to add new options to understand the effect of new technologies and different configurations. Moreover, as proven in this study, the OPN language is applicable to many “levels” of a given architecture.

Candidate GN&C system architecture models were implemented using the OPN modeling language in order to more easily enumerate possible architectures and ultimately identify which architectures have optimal properties. Following the basic procedure employed in the above references and using Ref. 2 to provide the background for the top-level abstraction used in the model, OPN was successfully employed in the models described in this paper.

Partial 2 x 2 systems (i.e., systems with up to dual redundancy per component class for two component classes) and 3 x 2 systems (systems with up to triple redundancy per component class for two component classes) were modeled in OPN. Within the constraints of these models, all possible architectures were rigorously enumerated and the weight/reliability trade-offs of cross-strapping components and using more than one type of component were assessed.

The described models assume perfect coverage – 100-percent accuracy in detecting and isolating a failure. The models also assume that more reliable components tend to be heavier, more costly, and/or more complicated to deal with. Given these assumptions, it was found that more reliable components are only beneficial in single string systems or systems with single point failures. All optimal architectures employing component redundancy could be produced from generic connections and the least reliable type of component from each component class.

According to Ref. 2, a GN&C system can be represented with sensors, computers, actuators, and how these components are interconnected. Given this abstraction, the completed OPN model discussed in this paper represents all possible GN&C architectures within a given set of constraints. The constraints are defined as the number of component classes, the maximum component redundancy in each component class, and the number of component types for each class.

In this paper, sensors, computers, and actuators will be defined as “component classes.” The terminology “I x J OPN model” will be used to describe a model with up to “I” redundancy per component class and up to “J” component classes. In other words, J = 2 could designate a model which only has sensors or which has both sensors and computers. J = 3 could designate a model with sensors, with sensors and computers, or with sensors, computers, and actuators. If J = 3 and I = 2, this could designate a system with up to two sensors, two computers, and two actuators. This paper will discuss OPN 2 x 2 and 3 x 2 models and touch on their applicability to a 3 x 3 model.

For the purpose of simplicity, it will be assumed that there are only three different types of components possible for each component class. In reality, three sensor types might include a sun sensor, star tracker, and an inertial measurement unit (IMU). However, to be more generic, types will not be designated so specifically – they will instead be referred to as type A, type B, and type C.

As a first pass, all enumerated architectures are evaluated based on two specific metrics: reliability and weight. Note that, with some exceptions, both complexity and cost increase as weight increases; thus, weight is a good first order approximation for these metrics.

Section III of this paper will discuss the design of the simple 2 x 2 model, section IV will give further details on the model, and Section V will discuss the design of the more complicated 3 x 2 model. Section VI will examine the application of reliability and weight metrics to the enumerated architectures. Finally, Section VII concludes and describes the model’s future iterations.

III. A “2 x 2” GN&C System

This section begins the discussion of the design of the 2 x 2 model. Even with just four components (two sensors and two computers), many architectures can be defined for a 2 x 2 system based on how the components are interconnected. Each of these architectures will have different total weight and different total reliability.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
64 of 102

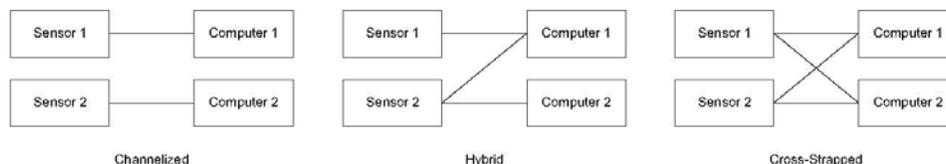


Figure 1. Three possible 2 x 2 systems.

Figure 1 depicts three possible 2 x 2 architectures. The reliability R of the three models is shown in Table 1 where s_j is the reliability of sensor j and c_k is the reliability of computer k.

Table 1. Reliability expressions for the 2 x 2 systems in Figure 1.

Architecture	Reliability
Channelized	$R = s_1c_1 + s_2c_2 - s_1c_1s_2c_2$
Hybrid	$R = s_1c_1 + s_2c_1 + s_2c_2 - s_2c_1c_2 - s_1s_2c_1 - s_1s_2c_1c_2 + s_1s_2c_1c_2$
Cross-Strapped	$R = s_1c_1 + s_1c_2 - s_1c_1c_2 + s_2c_1 + s_2c_2 - s_2c_1c_2 - s_1s_2c_1 - 2s_1s_2c_1c_2 + 2s_1s_2c_1c_2 - s_1s_2c_2 + 2s_1s_2c_1c_2 - s_1s_2c_1c_2$

It is important to note that, no matter what the architecture, the reliability of any 2x2 model can be generated by taking the cross-strapped expression for R and then eliminating terms from the expression for connections which do not exist and therefore do not contribute to system reliability.

Additional indicator variables are added to the cross-strapped reliability expression to specify which terms to eliminate. These indicator variables are correlated with the interconnections between components. A nonzero indicator variable represents a connection whereas an indicator variable equal to zero represents a missing connection.

Using the methodology described, the following general expression for R is obtained:

$$R = s_1i_{11}c_1 + s_1i_{12}c_2 - s_1i_{11}i_{12}c_1c_2 + s_2i_{21}c_1 + s_2i_{22}c_2 - s_2i_{21}i_{22}c_1c_2 - s_1s_2i_{11}i_{21}c_1 - s_1s_2i_{11}i_{22}c_1c_2 - s_1s_2i_{12}i_{21}c_1c_2 + s_1s_2i_{11}i_{21}i_{22}c_1c_2 - s_1s_2i_{12}i_{22}c_2 + s_1s_2i_{11}i_{12}i_{22}c_1c_2 + s_1s_2i_{12}i_{21}i_{22}c_1c_2 - s_1s_2i_{11}i_{12}i_{21}i_{22}c_1c_2$$

where i_{jk} is the reliability of the connection between sensor j and computer k if such a connection exists and is 0 otherwise.

As a sanity check, the reliability expressions for the channelized and hybrid architectures above can be derived from the general expression. Assuming perfect connection reliability, i.e., $i_{jk} = 1$ for all connections in the architecture, the channelized and hybrid architectures would be represented by the indicator variables in Table 2. Plugging these indicator variables into the general expression gives the same reliability expressions in Table 1.

Table 2. Indicator for the channelized and hybrid 2 x 2 systems in Figure 1.

Architecture	i_{11}	i_{12}	i_{21}	i_{22}
Channelized	1	0	0	1
Hybrid	1	0	1	1

IV. Details on the Model

This section gives further detail on the 2 x 2 model. Like the previously mentioned 3 x 2 and 3 x 3 models, the 2 x 2 OPN model can be viewed as a sophisticated Petri net model. In a Petri net model, information-storing tokens move via directed arcs from transitions to places and from places to transitions. Note that there may be more than one directed arc feeding from or to a transition or place. Upon arrival at a transition, a token is consumed, some processing is done, and, if appropriate, new tokens are introduced in the places dictated by the directed arcs leading from the transition.

The sequence of transitions in any of the discussed OPN models is a sequence of decision points. At each decision point, a token is replicated with multiplicity equal to the number of possible decisions. The information stored in each token represents a unique possible architecture. Taken together, the tokens enumerate all possible



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
65 of 102

architectures given an initial set of constraints. All tokens are collected when they completely propagate through the model for analysis.

Figure 2 is a visual representation of the OPN decision tree for the 2 x 2 model and the following questions are the decision points:

- How many sensors?
 - 1 or 2
- Type assignment for sensors?
 - If only one sensor, choose SensorA, SensorB, or SensorC
 - If two sensors, choose two of the same type of sensor or one of each of two types (possible combinations: AA, AB, AC, BB, BC, and CC)
- How many computers?
 - 1 or 2
- Type assignment for computers?
 - If only one computer, choose ComputerA, ComputerB, or ComputerC
 - If two computers, choose two of the same type of computer or one of each of two types (possible combinations: AA, AB, AC, BB, BC, and CC)
- Which sensors are connected to computer 1?
 - Just sensor 1
 - Just sensor 2 (if sensor 2 exists)
 - Both sensor 1 and 2 (if sensor 2 exists)
- If computer 2 exists, which sensors are connected to computer 2?
 - Just sensor 1
 - Just sensor 2 (if sensor 2 exists)
 - Both sensor 1 and 2 (if sensor 2 exists)



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
66 of 102

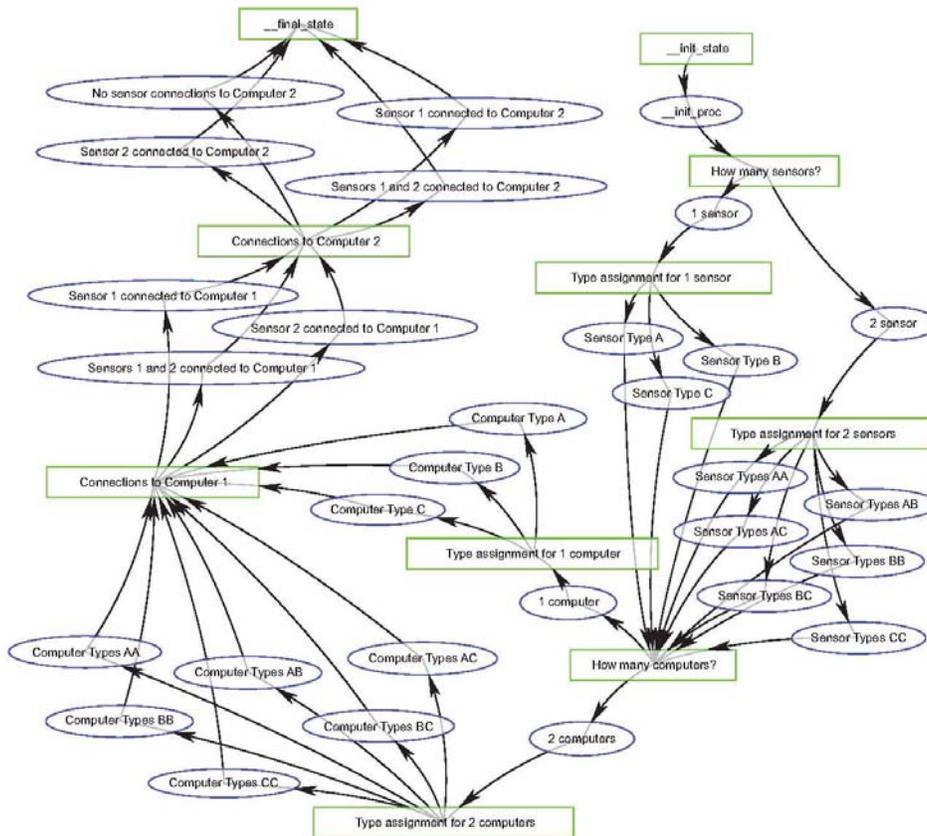


Figure 2. The 2 x 2 OPN decision tree.

During the process of token propagation, the number of components, component types, and connections are continuously updated for later use in reliability calculations. In addition, the current weight of the system is updated at execution time.

Each component type is given its own unique reliability and weight based on the specific make and model of the component. These values were based on real components, but modified slightly to facilitate analysis.

Reliabilities are dependent upon the failure rate of the component and the desired operational time for the component. The relationship is governed by the equation $R = e^{-\lambda t}$, where λ is the failure rate and t is the operational time. Operational time is user defined and based on the length of the proposed mission. An operational time of $t = 10$ years was used in the models discussed in this paper. Other component properties are illustrated in Table 3 and Table 4.



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
67 of 102

Table 3. Component properties for sensor types A, B, and C.

Sensor Type	A	B	C
Failure Rate λ (/year)	0.00015	0.0001	0.00005
Reliability R	0.9985	0.999	0.9995
Weight (dimensionless)	3	6	9

Table 4. Component properties for computer types A, B, and C.

Computer Type	A	B	C
Failure Rate λ (/year)	0.0001	0.00004	0.00002
Reliability R	0.999	0.9996	0.9998
Weight (dimensionless)	3	5	10

Two additional weights and one additional reliability were also included in the model. A “connection weight” and a “dissimilar component penalty” were included to ensure that weight continues to approximate complexity and cost. Cross-strapping components may not add much physical weight to the overall system, but it surely adds to the complexity and cost of the system. Similarly, dealing with more than one type of sensor and/or computer also increases complexity and cost. Hence, adding these additional weights where appropriate worked as a first step toward reality.

The weights associated with connections and dissimilar components were chosen to be consistent with the weights of sensors and computers. To do so, assumptions had to be made. Connections were considered to be, at most, one-third of the complexity of the average computer. In addition, the weight penalty for dissimilar components was set such that it was not larger than the heaviest sensor or the heaviest computer.

These logical assumptions dictated a certain range of weight values used for connections and dissimilar component parameters. However, rather than presuppose exact values for these weights, multiple OPN runs were executed varying one of the parameters each time. Assuming the connection reliability would be greater than that of a computer, the nine OPN scenarios are illustrated in Table 5.

Table 5. Connection reliabilities, connection weights, and dissimilar component penalties for each OPN scenario run.

OPN Scenario	1	2	3	4	5	6	7	8	9
Connection Reliability	1	1	1	0.99995	0.99995	0.99995	0.9999	0.9999	0.9999
Connection Weight (dimensionless)	0	0	0	0.5	0.5	0.5	5 / 3	5 / 3	5 / 3
Dissimilar Component Penalty (dimensionless)	0	6	9	0	6	9	0	6	9

V. A “3 x 2” GN&C System

This section discusses the design of the 3 x 2 model. Implementation of the 3 x 2 OPN model is very similar to that of the 2 x 2 model with two notable exceptions. These exceptions relate to the reliability formula for the overall system and the removal of duplicate architectures to conserve memory.

The reliability formula is much more complicated for these larger models and must be handled differently. Although reliability is still calculated after OPN completes execution, it can no longer be easily calculated by hand for implementation in Excel. Instead, symbolic MATLAB was used to multiply out the formula and a MATLAB script was used to insert the correct “i” indicator values where appropriate. Only after this manipulation was performed could the reliability be imported back into Excel for implementation.

In addition, care had to be taken to ensure no architecture was represented more than once in the model. Running a larger 3 x 2 OPN model would take an inordinate amount of time and computer memory. It was found that certain architectures could be represented in multiple configurations and this was not taken into account by the 2 x 2



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
68 of 102

model.⁸ By producing tokens for all possible configurations of the same architecture, the model took much more time and used much more memory than necessary.

An example duplicate architecture is shown in Figure 3. A1 and A2 represent the same architecture since, in both cases, one sensor of type A is connected to a computer of type A, the other sensor A is connected to a computer A and a computer B, and a sensor of type B is connected to a computer of type C. A3 represents a different architecture, however, since both sensors of type A are connected to a computer of type B.

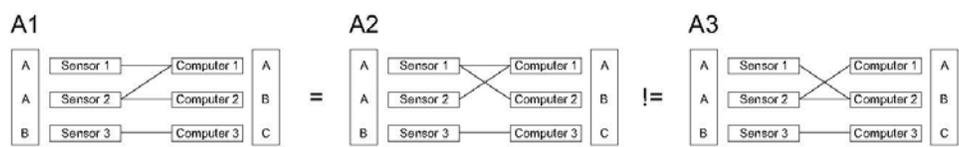


Figure 3. Determination of duplicate architectures.

The process of eliminating duplicate architectures began by choosing a representative set of sensor types and computer types. Ten possibilities were chosen as representative orderings of the three sensor types and also the three computer types: AAA, AAB, AAC, ABB, ACC, ABC, BBB, BBC, BCC, and CCC.

A1 in Figure 3 represents a connection pattern between three adjacent components. This connection pattern has four connections: Sensor 1 is connected to computer 1, sensor 2 is connected to both computer 1 and computer 2, and sensor 3 is connected to computer 3. Keeping this connection pattern fixed, we can give a “type” identity to the three sensors based on the ten possible orderings. For each possible ordering of the sensor types, there are ten possible orderings of the computer types, each of which defines a unique architecture. In other words, for any given connection pattern such as A1, there are $10 \times 10 = 100$ possible architectures. The OPN model iterates through all possible connection patterns and finds all 100 possible architectures for each one.

Note that orderings such as ABA and BAA are not taken to be representative orderings. When all possible connection patterns are taken into account, these additional orderings will fail to produce any architecture that cannot be produced by AAB. This is because ABA, BAA, and AAB are all equivalent – all represent two components of type A and one of type B. It does not matter in what order the letters are written as long as the case is represented.

Luckily, searching for duplicate architectures in OPN does not require checking 100 possible architectures for each connection pattern. The 100 possibilities for each connection pattern can be represented by just 16 representative architectures. As illustrated in Figure 4, the ten sensor type combinations and the ten computer type combinations can be further abstracted to just four representative combinations per component. First, AAA, BBB, and CCC all represent the case where all three components are of the same type. Next, AAB, AAC, and BBC all represent the case where the first two components are of the same type and the third component is of a different type. Furthermore, ABB, ACC, and BCC all represent the case where the second and third components are of the same type, but the first component is of a different type. Finally, ABC represents the case where all three components are of a different type.

⁸ The 2×2 model is much smaller than the 3×2 model. As a result, there were no memory issues and duplicate architectures could be removed in post-processing – they did not have to be removed in OPN.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
69 of 102

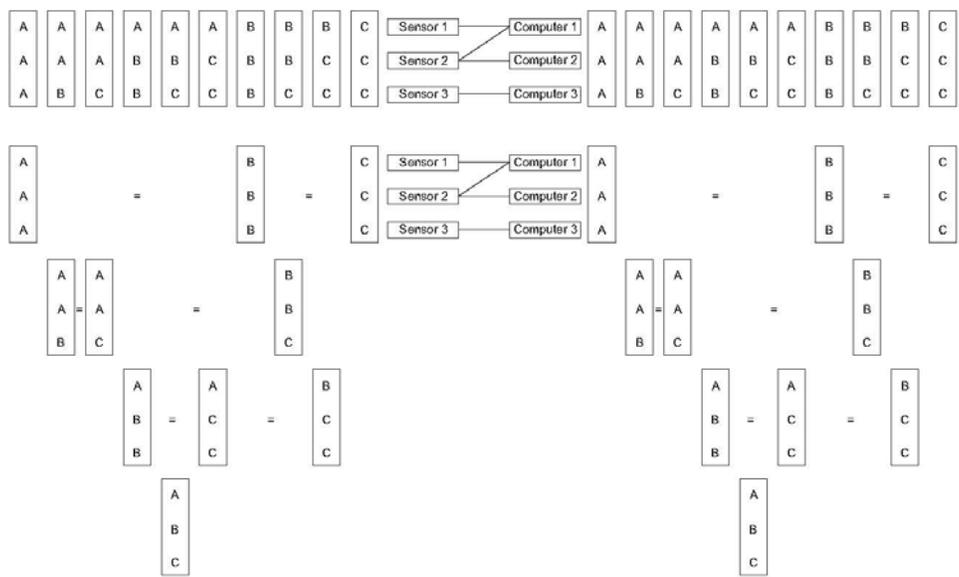


Figure 4. Finding representative architectures.

Figure 5 helps demonstrate why these representative architectures work for finding duplicate architectures. To use representative architectures to find duplicates is to claim that if architecture A1 is equivalent to architecture A2, but not A3, then architecture B1 is equivalent to B2, but not B3. This is clearly the case.

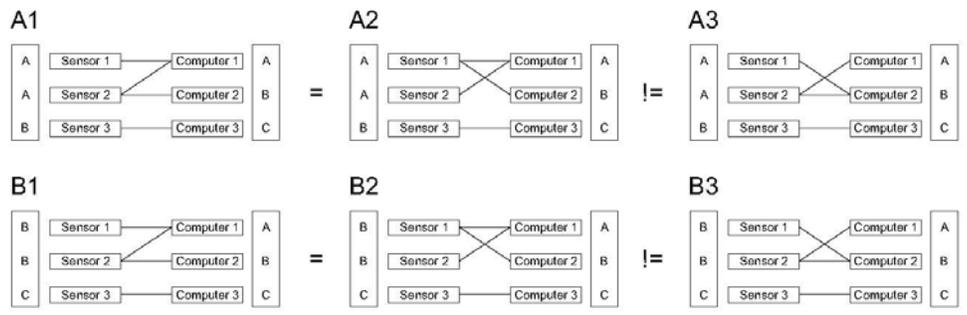


Figure 5. Using representative architectures to find duplicates.

As previously discussed, A1 and A2 represent the same architecture even though they have different connection patterns. It is arbitrary which form of an architecture is chosen as the primary form and which is a duplicate – either A1 or A2 could be considered the duplicate.

Implementing duplicate detection into the model turned out to be a very involved process. The project had a limited time horizon and the development time necessary for automating the duplicate detection process was uncertain. It was therefore decided that a surefire yet brute force method would be used to implement duplicate detection. All possible representative architectures were drawn by hand and duplicate architectures were circled. In all, over 100 pages of architectures were drawn and compared.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
70 of 102

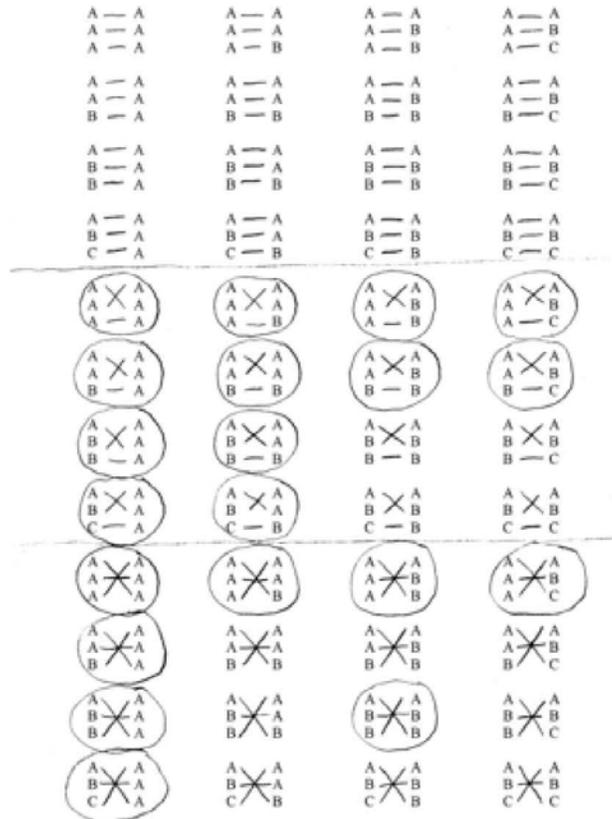


Figure 6. One page of hand-drawn architectures.

Based on the circled representative architectures, rules were created and inserted into OPN to keep any tokens that will produce duplicate architectures from propagating. Note that all rules are in the form of Boolean expressions starting with “not if” instead of “if”. Although all work was double-checked, it is conceivable that an incorrect rule was entered due to human error. By using “not if” instead of “if”, the default is to pass the token. It is better to retain a duplicate architecture rather than exclude a potentially optimal architecture.

The Boolean rules are inserted into the OPN model on the transitions from:

- Which sensors are connected to computer 1?
- If computer 2 exists, which sensors are connected to computer 2?
- If computer 3 exists, which sensors are connected to computer 3?

To the places:

- Just sensor 1
- Just sensor 2 (if sensor 2 exists)
- Just sensor 3 (if sensor 3 exists)
- Just sensors 1 and 2 (if sensor 2 exists)
- Just sensors 1 and 3 (if sensor 3 exists)
- Just sensors 2 and 3 (if sensor 3 exists)
- Sensors 1, 2, and 3 (if sensor 3 exists)



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

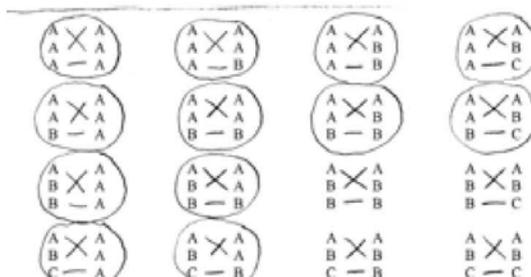
System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
71 of 102

Trivial rules govern which sensors are connected to computer 1. If a particular token represents an architecture with only 2 sensors, it is not possible to make a connection to a computer from a nonexistent sensor 3. Therefore, in the 2-sensor case, no new tokens are introduced into the places representing, "just sensor 3," "just sensors 1 and 3," "just sensors 2 and 3," or "sensors 1, 2, and 3." Similarly, if a particular token represents an architecture with only 1 sensor, no new tokens are introduced into the places representing, "just sensor 2," "just sensor 3," "just sensors 1 and 2," "just sensors 1 and 3," "just sensors 2 and 3," or "sensors 1, 2, and 3."

Rules governing connections to computer 2 and computer 3 are more complicated. If a token represents an architecture with only two computers, it is known what the final system architecture will be after creating the sensor connections to the second computer. If a token represents an architecture with three computers, it is known what the final system architecture will be after creating the sensor connections to the third computer. Connections that will form duplicate (circled) architectures should not be allowed to propagate. Hence rules are put in place to block introduction of these tokens.

Figure 7 shows an example rule based on a hand-drawn architecture. This rule determines whether or not a connection should be made between sensor 3 and computer 3. Note that, by the time a token reaches the given rule, the connections between sensor 1 and computer 2 as well as between sensor 2 and computer 1 have already been defined. No connection should be made if the type definitions for the sensors and computers match those represented by the circled architectures – such tokens will result in the formation of duplicate architectures. In other words, the connection between sensor 3 and computer 3 should not be made if sensor 1's type is the same as sensor 2's type or computer 1's type is the same as computer 2's type.



```
!(Computer_Redundancy < 3) &&
!(Sensor_Redundancy < 3) &&
!(
  (i11 == 0 && i12 == 0) ||
  (Sensor_Redundancy > 1 && i21 == 0 && i22 == 0) ||
  (
    (i11 == 0 && i12 > 0 && i21 > 0 && i22 == 0 && i31 ==
    0 && i32 == 0) &&
    (
      (Sensor1_Type == Sensor2_Type) ||
      (Computer1_Type == Computer2_Type)
    )
  )
)
```

Figure 7. An example rule based on a hand-drawn architecture.

Eliminating duplicate architectures in the 3 x 2 model significantly reduced the number of tokens produced. Before duplicates were removed, the OPN produced 51,902 tokens. After duplicates were removed, the model produced only 9,795 tokens.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 72 of 102	

VI. Results

Despite eliminating the unnecessary duplicate architectures from the model, attempts to run a 3 x 3 model still resulted in a memory shortage. Although it is unfortunate that the larger model could not complete, it is important to note that results from the 3 x 2 model can still be applied to the 3 x 3 case.

Taking such a top-level view of a GN&C system, the interaction between adjacent sensor and computer components is identical to the interaction between adjacent computer and actuator components. Just like sensors and computers, there are different types of actuators, each with a unique set of properties, and a system architect can choose different redundancies for each of these types. Furthermore, the connection patterns already found between sensors and computers are the same as those between computers and actuators. Finally, the metrics of weight and reliability can be calculated in exactly the same way.

The nine scenarios outlined in section IV were run and Pareto plots were produced for each. Representative plots for each of the nine scenarios are reproduced in Figures 8 – 11. In each scenario, the architectures that simultaneously had both lowest weight and highest reliability were identified. These architectures are “on the Pareto front.” The identified Pareto-front architectures for each scenario are reproduced in figures 8 – 11 as well. These architectures were found by zooming in on the “utopia point” at the lower right hand corner of the plot.⁹ Note that, in most cases, there are multiple identified architectures for each scenario since it is somewhat subjective which architectures are closer to the utopia point. Is an architecture with weight = 17 and reliability = 0.999999596912468 (six “9”s) better than an architecture with weight = 18 and reliability = 0.99999995634079 (eight “9”s)? The answers to such questions would be made clear in a mission requirements context. For a human-rated mission, perhaps a reliability of 0.9999999 (seven “9”s) is required for safety. If this were the case, the architecture with weight = 18 would clearly be better, since the one with weight = 17 does not meet the requirement.

In the zoomed out (top) plot of each of the nine scenarios, there appear to be six clusters of architecture data points. Although the clusters appear to be columns of data points, they are not; the architectures in each cluster have nearly identical, but not completely identical reliabilities. Looking from left to right, the first five clusters – the five clusters with the lowest reliability – are driven by single point failures of any of the six component types. For example, in an architecture that contains one sensor and two computers or an architecture that contains one sensor and three computers, the single sensor present in the architecture must remain reliable in order for the overall system to remain reliable. Sensor type A has a reliability of 0.9985. Therefore, a one-sensor two-computer or one-sensor three-computer architecture that contains sensor A can have a maximum reliability of 0.9985. Such architectures, which have a single point failure at sensor A, define the first (least reliable) cluster of data points. Both sensor type B and computer type A have the same reliability: 0.999. Therefore, architectures that have a single point failure at a sensor of type B or a single point failure at a computer of type A will both fall into the second cluster. Similarly, sensor type C’s reliability of 0.9995 defines the third cluster, computer type B’s reliability of 0.9996 defines the fourth cluster, and computer type C’s reliability of 0.9998 defines the fifth cluster.

The sixth and final (most reliable) cluster contains all other architectures – architectures free from single point failures. The few additional points that do not fall into any of the six clusters are single-string architectures, i.e., architectures that contain just one sensor and one computer. These architectures contain not one, but two single point failures and are therefore significantly less reliable than an identical architecture with additional computers or additional sensors.

⁹ The utopia point represents the ideal architecture which is 100 percent reliable with weight = 0.



Title:

**System Architectural Considerations on Reliable
 Guidance, Navigation, and Control (GN&C) for
 Constellation Program (CxP) Spacecraft**

Page #:
 73 of 102

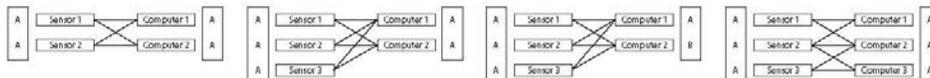
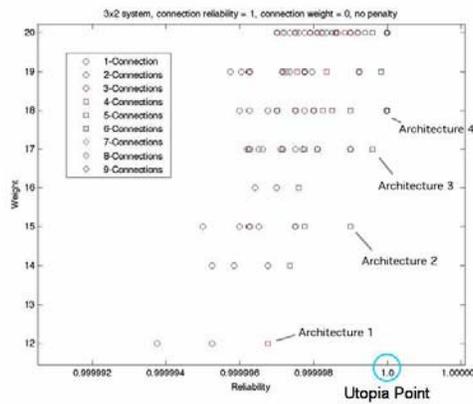
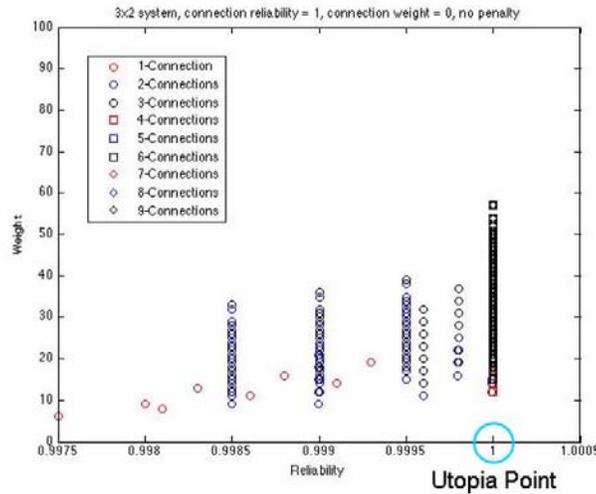


Figure 8. (Top) Pareto plot, with added details for number of connections between sensors and computers, (Middle) zoomed in version of the same plot, (Bottom) potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 12 and reliability = 0.99999675437371 (five 9s), architecture 2 has weight = 15 and reliability = 0.99999899763201 (five 9s), architecture 3 has weight = 17 and reliability = 0.99999959691247 (six 9s), and architecture 4 has weight = 18 and reliability = 0.9999999563408 (eight 9s).



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
74 of 102

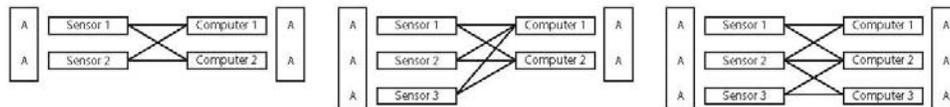
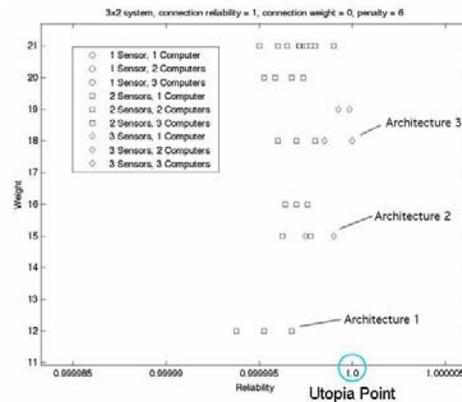
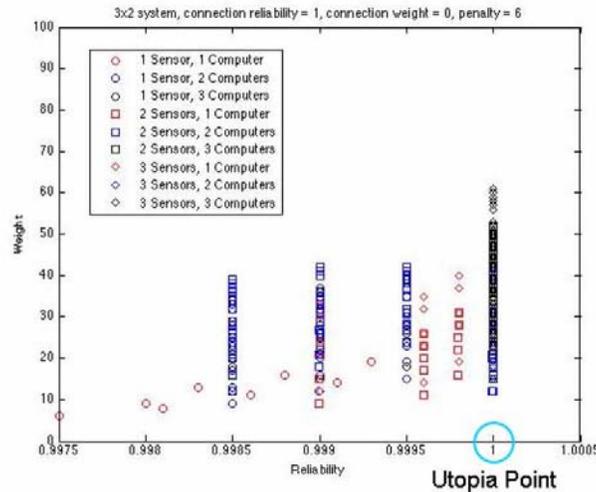


Figure 9. (Top) Pareto plot with added details for both number of sensors and number of computers, (Middle) zoomed in version of the same plot, (Bottom) potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 12 and reliability = 0.99999675437371 (five 9s), architecture 2 has weight = 15 and reliability = 0.99999899763201 (five 9s), and architecture 3 has weight = 18 and reliability = 0.9999999563408 (eight 9s).



NASA Engineering and Safety Center Technical Assessment Report

Document #:
NESC-RP-06-074

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
75 of 102

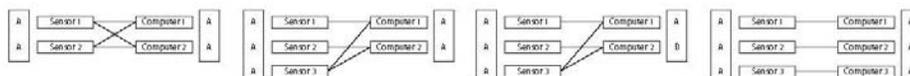
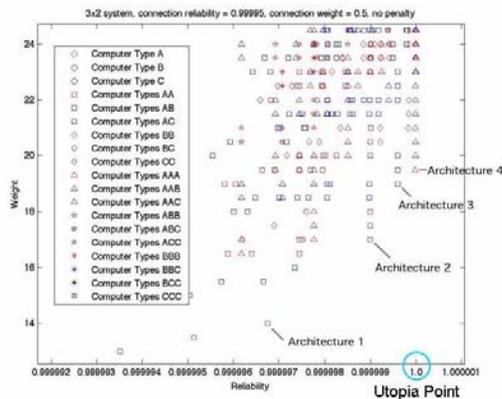
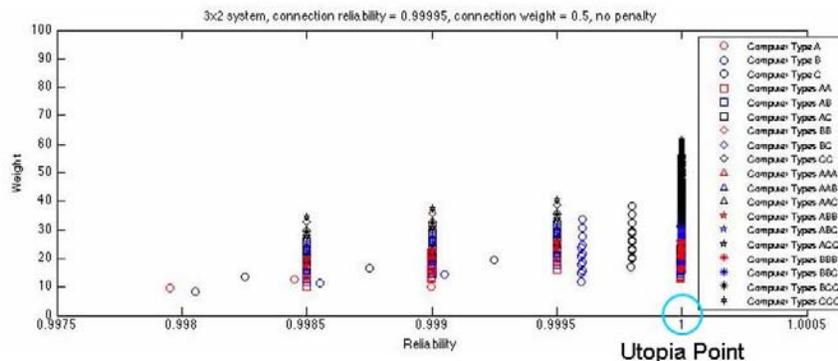


Figure 10. (Top) Pareto plot with added details for both number and types of computers, (Middle) zoomed in version of the same plot, (Bottom) potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 14 and reliability = 0.999996754062388 (five 9s), architecture 2 has weight = 17 and reliability = 0.999998992620742 (five 9s), architecture 3 has weight = 19 and reliability = 0.999999593334442 (six 9s), and architecture 4 has weight = 19.5 and reliability = 0.99999983481890 (seven 9s).



Title:

**System Architectural Considerations on Reliable
 Guidance, Navigation, and Control (GN&C) for
 Constellation Program (CxP) Spacecraft**

Page #:
 76 of 102

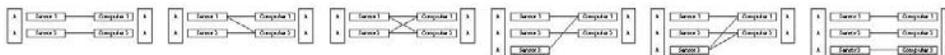
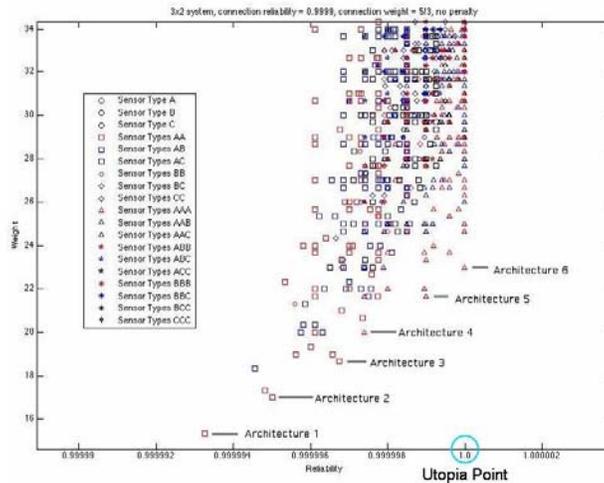
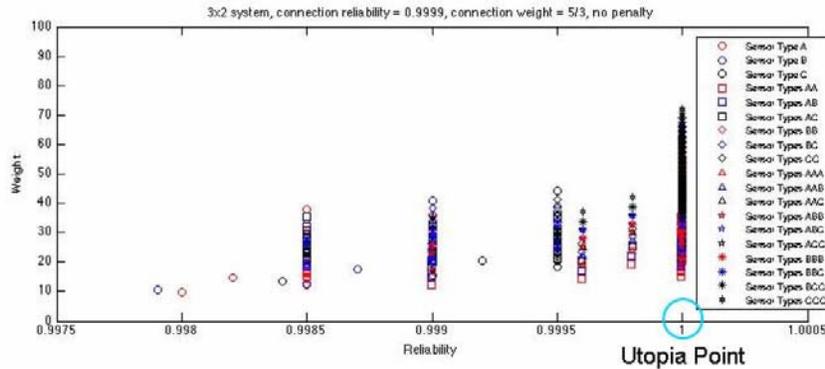


Figure 11. (Top) Pareto plot with added details for both number and types of sensors, (Middle) zoomed in version of the same plot, (Bottom) Potential optimal architectures for this scenario: (From left to right) Architecture 1 has weight = 15.333 and reliability = 0.999993257523472 (five 9s), architecture 2 has weight = 17 and reliability = 0.99999501059889 (five 9s), architecture 3 has weight = 18.667 and reliability = 0.999996753726173 (five 9s), architecture 4 has weight = 20 and reliability = 0.999997398039817 (five 9s), architecture 5 has weight = 21.667 and reliability = 0.999998992071849 (five 9s), and architecture 6 has weight = 23 and reliability = 0.99999983481890 (seven 9s).



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
77 of 102

Many interesting aspects can be found in Figure 8, the ideal case where there is no penalty for connection weight or using dissimilar components. In this scenario, all identified architectures are fully cross-strapped. This is to be expected since additional connections increase reliability yet cost nothing. Figure 8 also demonstrates that the weights of an architecture's component plays a major role in determining the optimality of the architecture. Recall that components of type "A" are the lightest and least reliable and that components of type "C" are the heaviest and most reliable. Even though components of type "A" are the least reliable, sensors of type A and computers of type A are by far the most prevalent component types in all the optimal architectures. In addition, although components of type "C" are the most reliable, no components of this type appear in any of the optimal architectures. Furthermore, there are not even any sensors of type B in the optimal architectures. Architecture 3 contains a computer of type B, but this architecture is no longer optimal once the "dissimilar components penalty" is increased from penalty = 0 to penalty = 6 (see Figure 9, bottom). Since the optimal architectures for penalty = 6 contain no dissimilar components, increasing the penalty again to penalty = 9 results in no further changes to the optimal architectures.

Figure 10 gives a baseline for what is realistic. In this scenario, there are no longer perfect connection reliabilities of 100 percent and there is an actual cost for producing connections. Now, the connection reliability = 0.99995 and the connection weight = 0.5. As a result of the 0.5 connection weight, only one fully cross-strapped architecture (architecture 1) is among the optimal architectures – much different than the scenario where connection weight = 0. The other optimal architectures have just three or four connections. From these, architectures 2 and 3 are the most interesting. Each has three sensors and two computers with two of the sensors having one connection to a computer and the last having two.

The connection weight = 0.5 scenario still has quite a bit in common with the connection weight = 0 scenario. Once again, the component types in the optimal architectures are predominantly of type A and never of type C. Only one optimal architecture (architecture 3) contains a component of type B and this component is again a computer. This architecture is no longer optimal when the dissimilar components penalty is increased to penalty = 6. There is no change in optimal architectures for connection weight = 0.5 when this penalty is increased from penalty = 6 to penalty = 9.

Figure 11 depicts the first scenario where connection reliability = 0.9999 and connection weight = 5/3. This scenario produces very similar optimal architectures to the connection weight = 0.5 scenario, but there are some notable exceptions.

Even with the dissimilar components penalty set to penalty = 0, connection weight = 5/3 is sufficiently high as to eliminate any architecture that contains a component type heavier than type A. This means that architecture 3 from Figure 10, the only connection weight = 0.5 architecture which contains a component of type B, is not an optimal architecture for connection weight = 5/3. This also means that the optimal architectures for connection weight = 5/3 will not change if the dissimilar components penalty is increased to penalty = 6 or penalty = 9.

A connection weight of 5/3 also makes it significantly more desirable to have architectures with even fewer connections. Although optimal architectures 1, 2, and 4 from Figure 10 are still optimal architectures when the connection weight is increased to 5/3, the value of these architectures is diminished with the heavier connection weight. As a result, these architectures are no longer significantly closer to the utopia point than architectures 1, 2, and 4 from Figure 11.

The six potentially optimal architectures for connection weight = 5/3 produce an interesting set. All legal architectures with four or fewer connections that contain two to three sensors of type A and two computers of type A are optimal architectures for this scenario. In effect, a system architect is directly trading an increase in weight for additional reliability when connection weight = 5/3.

When reviewing a subset of all the possible architectures, specifically a subset in which all members have the same number of connections, the effect of the dissimilar components penalty on the optimal architectures of the subset is nearly identical to the penalty's effect on the optimal architectures of the superset. Figures 12 and 13 depict the optimal architectures for 3 x 2 systems with 1, 2, 3, 4, 5, 6, 7, 8, and 9 connections when connection reliability = 1 and connection weight = 0 (the reliabilities and weights for these architectures can be found in Tables 6 through 8). Again, as the penalty is increased from penalty = 0 to penalty = 6, nearly all architectures containing sensor type B or computer type B cease to be optimal. Note that, for architectures with the same number of connections, the exact same architectures which are optimal for penalty = 0 will be optimal no matter what the connection weight. Similarly, architectures which are optimal for penalty = 6 or penalty = 9 will also remain optimal no matter what the connection weight. The reasoning goes as follows. Although the overall system weight of any member of a subset will change if the connection weight is changed, this change will be identical to the change seen by any of the other systems in this subset. This is because all systems in each subset have, by definition, the same number of



**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
78 of 102

connections. Therefore, among architectures with the same number of connections, the architectures closest to the utopia point will remain closest to the utopia point no matter what change is made to the connection weight.

Table 6. The reliabilities and weights for the 1-, 2-, 3-, 4-, 5-, 6-, 7-, 8-, and 9-connection architectures closest the utopia point given a 3 x 2 system with connection reliability = 1, connection weight = 0, and no penalty for dissimilar components.

Connections	Reliability	Weight	On Pareto front?
1	0.999100405	14	
	0.999300245	19	
2	0.999993766	12	
	0.999995260	14	
	0.999996397	16	
	0.999997344	19	
	0.999997754	20	
	0.999998292	22	
	0.999998741	25	
3	0.999995260	12	
	0.999996756	14	
	0.999997499	15	
	0.999998996	17	
4	0.99999984	18	
	0.999996754	12	YES
	0.999998993	15	
	0.999999594	17	
5	0.99999988	18	
	0.999998995	15	
	0.999999596	17	
6	0.99999992	18	
	0.999998998	15	YES
	0.999999597	17	YES
7	0.99999996	18	
	0.99999994	18	
8	0.99999996	18	
9	0.99999996	18	YES



NASA Engineering and Safety Center Technical Assessment Report

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
79 of 102

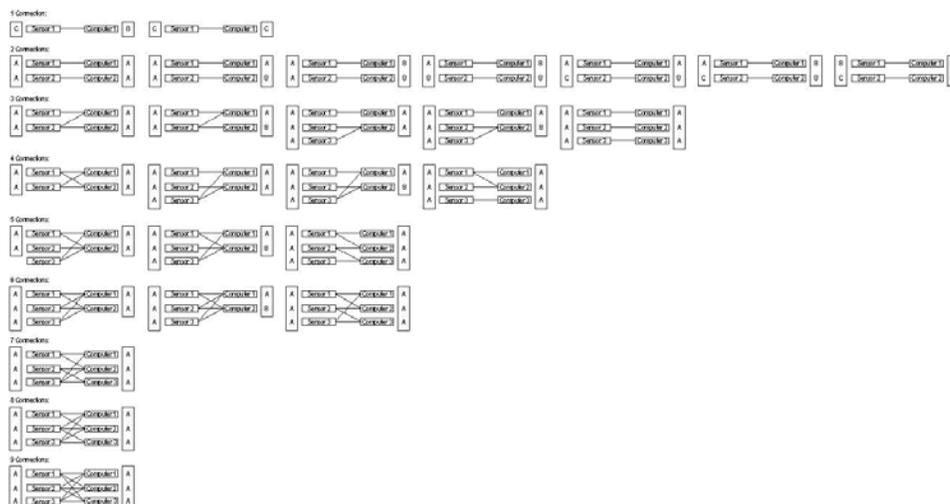


Figure 12. The architectures described in Table 6.

Table 7. The reliabilities and weights for the 1-, 2-, 3-, 4-, 5-, 6-, 7-, 8-, and 9-connection architectures closest the utopia point given a 3 x 2 system with connection reliability = 1, connection weight = 0, and penalty = 6 for dissimilar components.

Connections	Reliability	Weight	On Pareto front?
1	0.999100405	14	
	0.999300245	19	
2	0.999993766	12	
	0.999996397	16	
3	0.999995260	12	
	0.999997499	15	
	0.999999984	18	
4	0.999996754	12	YES
	0.999998993	15	
	0.999999988	18	
5	0.999998995	15	
	0.999999992	18	
6	0.999998998	15	YES
	0.999999996	18	
7	0.999999994	18	
8	0.999999996	18	
9	0.999999996	18	YES



Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
80 of 102

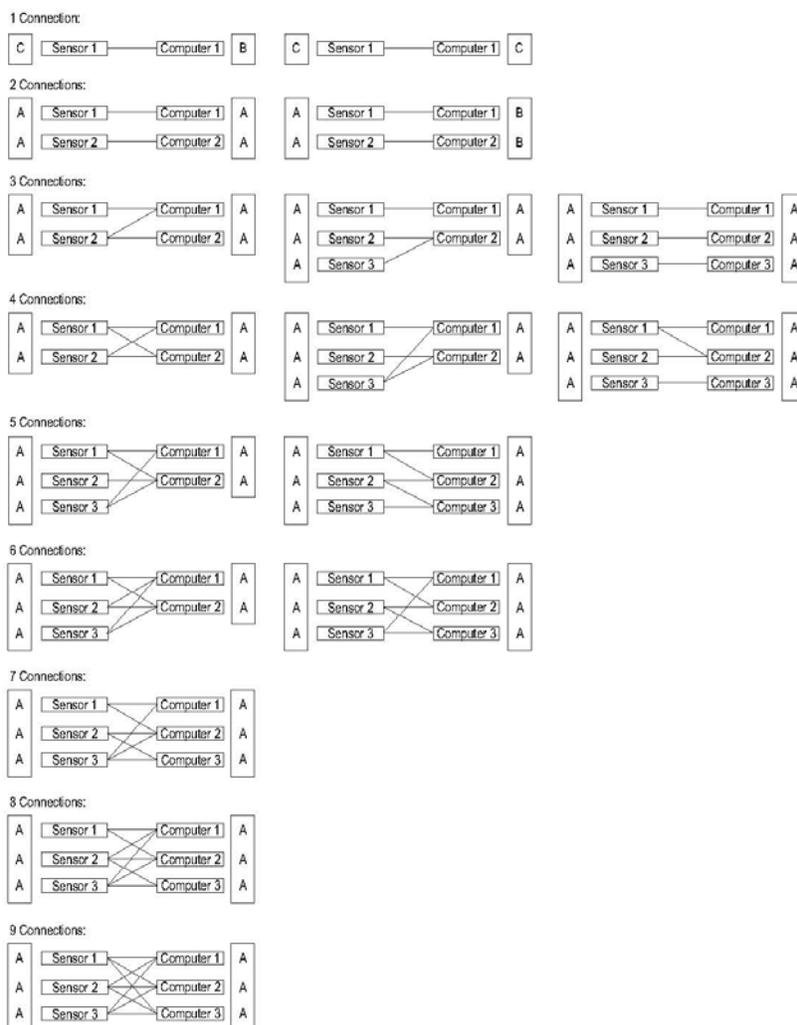


Figure 13. The architectures described in Table 7.

VII. Conclusion

If the only materials on hand were sensors of low-reliability type A, computers of low-reliability type A, and generic connections, it is still possible to produce nearly all potentially optimal architectures. The identified optimal architectures show this and, more generally, that it is preferable to increase redundancy of lighter, less reliable components rather than to use smaller numbers of more reliable, heavy components.

Due to the extremely lightweight nature of “sensor A” and “computer A”, these component types appeared in the optimal architectures in every scenario and almost always with redundancy ≥ 2 . Furthermore, the very heavy yet very reliable “sensor C” and “computer C” never appeared as an optimal architecture in any scenario, not even with redundancy = 1.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 81 of 102	

At a certain point, there are diminishing returns to adding redundancy, connections, or components with greater reliability – the system starts to experience only minimal increases in overall reliability for the large gains in weight. This fact becomes more apparent through the use of the connection weight penalty. Looking at the optimal architectures in each scenario, the number of connections drops dramatically as the connection weight penalty is increased from connection weight = 0 to connection weight = 0.5. When connection weight = 0, there are, as expected, optimal architectures with as many as nine connections. When connection weight increases to 0.5 however, this number drops to three or four.

The impact of the “dissimilar components penalty” is more subtle, but still apparent. In the scenarios where connection weight = 0 and connection weight = 0.5, architectures containing both computer type A and computer type B were identified as potentially optimal when the penalty = 0. However, when the penalty is increased to penalty = 6, these architectures no longer appear to be better than other architectures.

The created OPN models the building blocks of a GN&C system. Now that the base model is complete, more detail can gradually be added until the entire system is encapsulated. First, the OPN language should be made more efficient so it does not run out of memory when trying to run the 3 x 3 model. Next, other architectural layouts should be investigated in which the component redundancy for any component can be greater than three. In order to do so, however, the process of eliminating duplicates and creating rules (which was done by hand) should be automated. Also, it is important that more than three types of sensors, computers, and actuators be incorporated into the model and that further details be included for each component. For example, requirements might suggest that a system needs six degrees-of-freedom of attitude and position knowledge. This requirement cannot be addressed with the current top-level mentality. Finally, further metrics besides reliability and weight should be implemented in the model.

Having such a model in place in OPN would serve as an excellent tool for system design. System requirements may dictate a given reliability and such a model would easily be able to find the lowest weight/lowest cost option for that reliability. Although less likely, if, rather than overall reliability, the system requirement called for a certain number of connections between adjacent components, the OPN model could easily find the optimal high reliability/low weight option for this architecture as well.

In addition to reliability, OPN can be made to find the best option for satisfying other metrics as these metrics are added to the model. As additional component redundancy and component types are added to the model, all possible architectures could eventually be both enumerated and evaluated by OPN.

A detailed consideration of the findings in this paper could be extremely beneficial to the development of the CxP’s robotic and human-rated systems; clearly exploring commonality in GN&C components can reduce both nonrecurring and recurring cost and risk.

Acknowledgments

The authors would like to thank their colleagues at MIT, Draper Laboratory and NASA for their support, technical insights, and help in reviewing this paper. Ralph Roe and the NASA Engineering and Safety Center management team are also to be acknowledged for their support and sponsorship of this GN&C discipline advancing activity.

References

- ¹ Cameron, B., Crawley, E., Loureiro, G., and Rebutisch, E., “Value flow mapping: Using networks to inform stakeholder analysis” *Acta Astronautica*, Volume 62, Issues 4-5, February-March 2008.
- ² Dominguez-Garcia, A., Hanuschak, G., Hall, S., and Crawley, E., “A Comparison of GN&C Architectural Approaches for Robotic and Human-Rated Spacecraft” *AIAA Guidance, Navigation and Control Conference and Exhibit*, 20-23 Aug 2007, Hilton Head, SC.
- ³ Hofstetter, W., Wooster, P., Nadir, W., and Crawley E., “Affordable Human Moon and Mars Exploration Through Hardware Commonality” *AIAA-2005-6757 Space 2005*, Long Beach, California, Aug. 30-1, 2005.
- ⁴ Sahner, R., Trivedi, K., and Puliafito, A. *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*. Kluwer Academic Publishers, 1995.
- ⁵ Cameron Simmons, W., Koo, B., and Crawley, E., “Architecture Generation for Moon-Mars Exploration Using an Executable Meta-Language” *AIAA-2005-6726 Space 2005*, Long Beach, California, Aug. 30-1, 2005.
- ⁶ Chapman, G. T., and Tobak, M., “Nonlinear Problems in Flight Dynamics,” NASA TM-85940, 1984.



**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
82 of 102

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft			Page #: 83 of 102

Appendix C. Study Summary Presentation



Massachusetts Institute of Technology

Study Summary Presentation: Using OPN for Comparing Fault-Tolerant GN&C System Architectures

Gregor Z. Hanuschak (MIT)

Nick Harrison (Draper Laboratory)

Ed Crawley (MIT)

Final – 12/19/08

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 84 of 102	

Task Description

- Analyze different architectural approaches for fault tolerance in Lunar GN&C systems

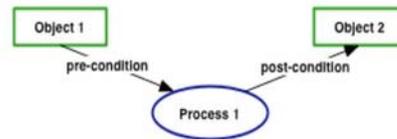
- Determine if performance goals can be met for different levels of reliability by combining like components in different architectures

- Implementing this model in the Object Process Network (OPN) modeling language in order to:
 - More easily enumerate and evaluate all possible architectures
 - Ultimately find those architectures that are most optimal in terms of architecture, performance and proxy for development cost

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 85 of 102	

What is OPN?

- OPN is a **visual** and **computable** meta-language that assists with systems architecting tasks.



- OPN is used to:
 - **Describe** and **Partition** space of architectural alternatives.
 - **Generate** and **Enumerate** the set of instances of feasible system models.
 - **Simulate** and **Order** the performance metrics of each model.
- OPN is a network (a directed graph) of **objects** and **processes** connected by **relationships**



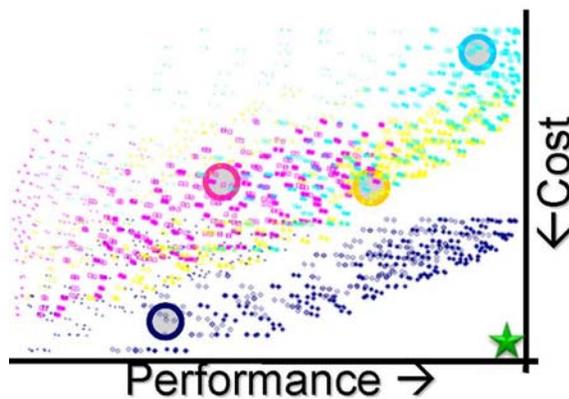
Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
86 of 102

Motivation for the Use of OPN

- OPN allows an engineer to systematically explore the architecture space
 - Gives confidence, as entire space is mapped
 - Modeling process often provides insights
 - Allows team to keep options open: no need to trim decision tree early
- Combines visual representation on Pareto plot with mathematical modeling
- Easy to add new options to understand the effect of new technologies, different configurations



- Applicable to many “levels” of architecture
 - Fleet architecture of ships, aircraft, radars, missiles
 - Missile architecture of propulsion modules, seeker heads, data
 - Propulsion architecture of propellants, staging, sizing

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title:	System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft	Page #: 87 of 102	

Systems Architecture via OPN

- Systems architecting using OPN is a three-step process:
 - Define a set of rules to generate a valid architecture (encoding)
 - Enumerate all possible system architectures (enumerating)
 - Evaluate the performance of these architectures in terms of metrics (evaluating)





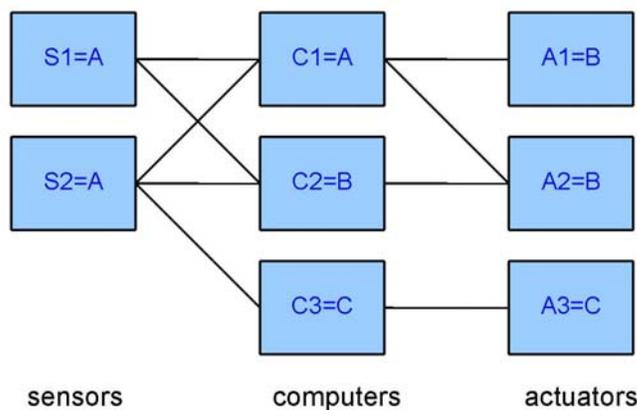
Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
88 of 102

1) Encoding

- GN&C systems can be broken down into simple subunits :
 - Sensors
 - Computers
 - Actuators
 - Interconnections between components



RULES FOR ARCHITECTURE GENERATION:

- Use 3 kinds of elements (sensors, computers and/or actuators)
- Use up to 3 sensors / computers / actuators
- Use 3 different types of sensors, computers and actuators: A, B, C
- Use all the feasible interconnections between these elements



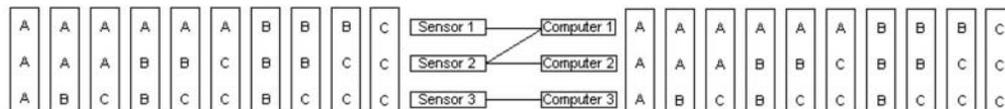
Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
89 of 102

2) Enumerating

- How many possibilities for sensors? $CR_{4\{A,B,C,0\}}^3 - 1 = 20 - 1 = 19$:
{A,B,C,AA,BB,CC,AB,AC,BC,AAA, AAB, AAC, ..., CCC}
- Obviously the same possibilities for computers: 19
- How many possible interconnections between sensors and computers?
 - » Sensor i can be connected to: {C1; C2; C3; C1 and C2; C2 and C3; C1 and C3; or C1, C2 and C3} THEN $7 \times 7 \times 7 = 243$
- $19 \times 19 \times 243 = 87,723$ possible architectures for the 3x2 system
- $19 \times 243 \times 19 \times 243 \times 19 = 405,017,091$ possible architectures for the 3x3 system!
- Not all of these architectures are feasible, as some components are not connected, and some patterns are duplicates. Rules have been developed to eliminate these cases.



	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 90 of 102	

3) Evaluating

-
- Two metrics were used for evaluating performance in this study:
 - **Reliability** as a performance metrics (benefit)
 - **Mass** as a first order approximation to development cost, and to recurring and launch cost
 - The reliability is calculated taking into account the reliabilities of the elements (sensors, computers and interconnections) and how they are connected.
 - The mass of the architecture is calculated as the sum of the masses of all its elements (sensors, computers and interconnections)]
 - Additional “mass” penalties were assessed for use of dissimilar components for actuators, sensors and processors
 - The use of OPN allows automatic evaluation of each enumerated architecture, which helps the architect identify the most optimal ones.



Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
91 of 102

3) Evaluating

Reliability of a of a component
(sensor, computer):

$R = \text{Prob}(0 \text{ fails of component in } t \text{ years}) = e^{-\lambda t} = s_1, c_1 \text{ or } i_{11}$ (Poisson distribution)

Reliability of a sensor-computer link:

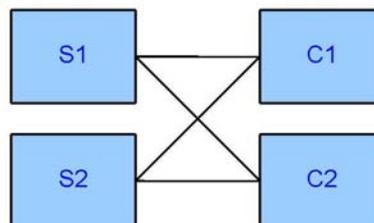
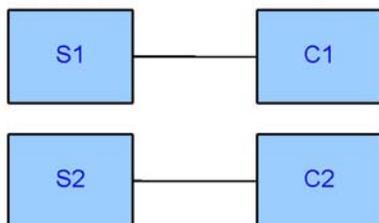
$R_1 = \text{Prob}(0 \text{ fails of sensor AND } 0 \text{ fails of interconnection AND } 0 \text{ fails of computer in } t \text{ years}) = s_1 i_{11} c_1$

Reliability of a 2x2 system
(channelized)

$R = \text{Prob}(0 \text{ fails of link 1 OR } 0 \text{ fails of link 2 in } t \text{ years}) = s_1 i_{11} c_1 + s_2 i_{22} c_2 - s_1 i_{11} c_1 s_2 i_{22} c_2$

Reliability of a 2x2 system (cross-strapped):

$R = s_1 i_{11} c_1 + s_1 i_{12} c_2 - s_1 i_{11} i_{12} c_1 c_2 + s_2 i_{21} c_1 + s_2 i_{22} c_2 - s_2 i_{21} i_{22} c_1 c_2 - s_1 s_2 i_{11} i_{21} c_1^2 - s_1 s_2 i_{11} i_{22} c_1 c_2 - s_1 s_2 i_{12} i_{21} c_1 c_2 + s_1 s_2 i_{11} i_{12} i_{21} c_1^2 c_2 + s_1 s_2 i_{11} i_{21} i_{22} c_1^2 c_2 - s_1 s_2 i_{12} i_{22} c_2^2 + s_1 s_2 i_{11} i_{12} i_{22} c_1 c_2^2 + s_1 s_2 i_{12} i_{21} i_{22} c_1 c_2^2 - s_1 s_2 i_{11} i_{12} i_{21} i_{22} c_1^2 c_2^2$





**NASA Engineering and Safety Center
Technical Assessment Report**

Document #:
**NESC-RP-
06-074**

Version:
1.0

Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
92 of 102

Values used for analysis

Sensor Type	A	B	C
Failure Rate λ (/year)	0.00015	0.0001	0.00005
Reliability R	0.9985	0.999	0.9995
Weight (dimensionless)	3	6	9

Computer Type	A	B	C
Failure Rate λ (/year)	0.0001	0.00004	0.00002
Reliability R	0.999	0.9996	0.9998
Weight (dimensionless)	3	5	10

OPN Scenario	1	2	3	4	5	6	7	8	9
Connection Reliability	1	1	1	0.99995	0.99995	0.99995	0.9999	0.9999	0.9999
Connection Weight (dimensionless)	0	0	0	0.5	0.5	0.5	5 / 3	5 / 3	5 / 3
Dissimilar Component Penalty (dimensionless)	0	6	9	0	6	9	0	6	9

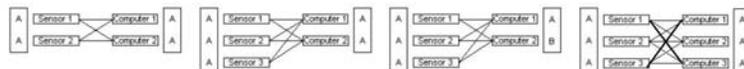
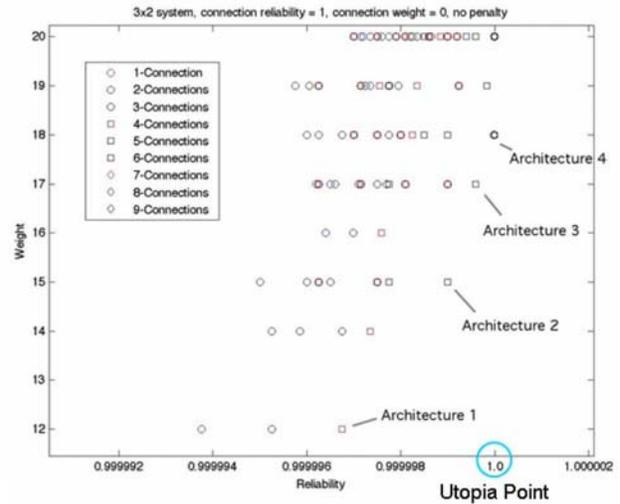
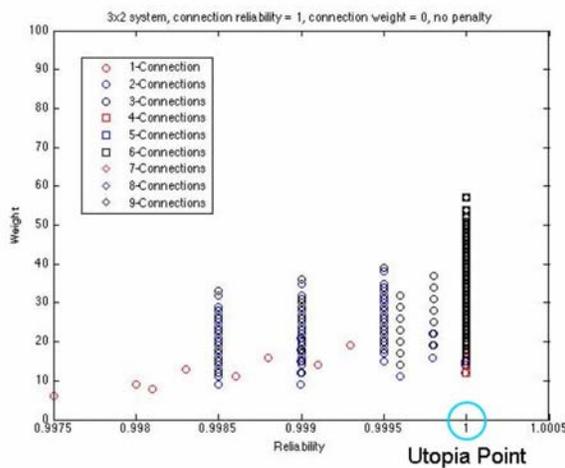


Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
93 of 102

**Results for the ideal case
($i_{jk} = 1, CW=0, DCP=0$)**



Arch 1 Arch 2 Arch 3 Arch 4

Architectures on the Pareto frontier

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 94 of 102	

Subset Overview of Results

- For the case shown on the previous chart:
 - The connection reliability is 1
 - The connection mass (CW) is zero
 - The dissimilar component penalty (DCP) is zero
- All identified desirable architectures are fully cross-strapped
 - Additional connections increase reliability yet cost nothing (in terms for added weight)
- Component weights play a major role in determining the optimality of the architecture
 - Even though components of type “A” are the least reliable, sensors of type A and computers of type A are by far the most prevalent component types in all the optimal architectures
 - Components of type “C” are the most reliable, yet no components of this type appear in any of the optimal architectures
 - Architecture 3 contains a computer of type B, but this architecture is no longer optimal once the “dissimilar components penalty” (DCP) is increased from penalty = 0 to penalty = 6

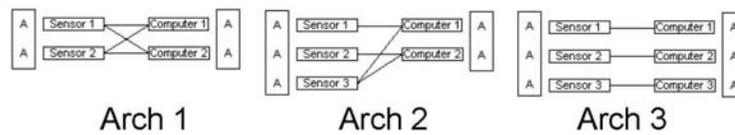
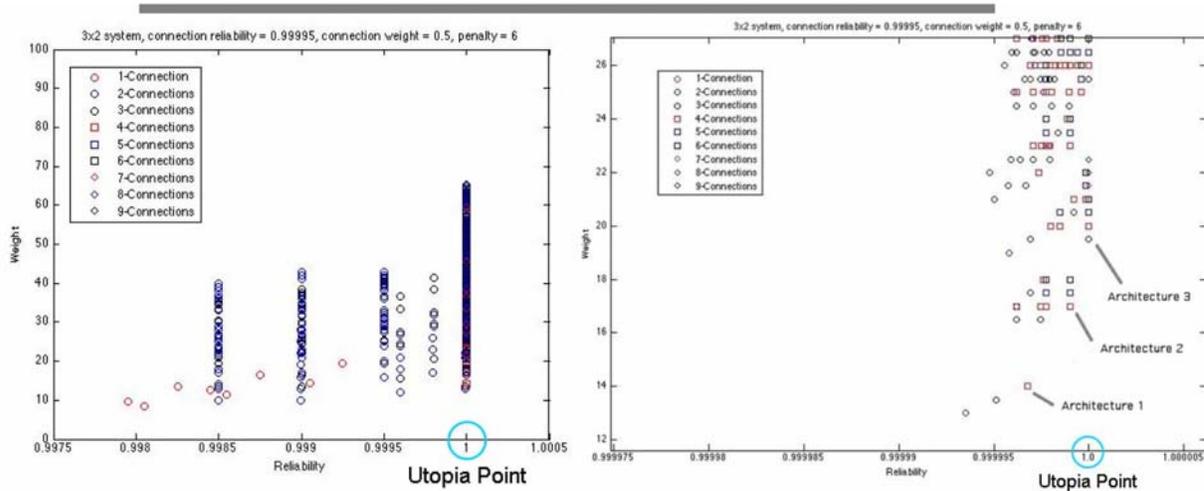


Title:

System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft

Page #:
95 of 102

**Results for a realistic case
($ijk = 0.99995$, $CW = 0.5$, $DCP = 6$)**



Architectures on the Pareto frontier

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 96 of 102	

Results for More Realistic Case

- The case on the previous chart now has finite reliability of connections, finite mass for connections, and a penalty for dissimilar components
- The effect of the dissimilar components penalty on the optimal architectures of specific connection reliabilities and weights is nearly identical to the penalty's effect on the optimal architectures of the superset
- As the penalty is increased from penalty = 0 to penalty = 6, nearly all architectures containing sensor type B or computer type B cease to be optimal
- For architectures with the same number of connections, the exact same architectures which are optimal for penalty = 0 will be optimal no matter what the connection weight
- For, architectures which are optimal for penalty = 6 will also remain optimal no matter what the connection weight
- Although the overall system weight of any member of a subset will change if the connection weight is changed, this change will be identical to the change seen by any of the other systems in this subset

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 97 of 102	

Conclusions

- A systematic method of evaluating the architectures of GNC systems composed of a limited set of common components and connections has been developed.
- Based on the values chosen in this representative calculation:
 - Most optimal architectures can be created by using only the lightest, lowest reliably components
 - It is preferable to increase redundancy of lighter, less reliable components rather than to use smaller numbers of more reliable, heavy components
 - There are diminishing returns to adding redundancy, connections, or components with greater reliability – the system starts to experience only minimal increases in overall reliability for the large gains in weight
 - For optimal architectures in each scenario, the number of connections drops dramatically as the connection weight penalty is increased
 - Dissimilar component penalty has a strong effect of homogenizing the architecture

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 98 of 102	

Recommendations for Future Investigation/Assessment

- Find a more efficient method of determining the reliability of a complex architecture
- Look at other GN&C architectural layouts: 4 sensor, 3 computer, 2 actuators (for example)
- Include more metrics in the analysis: performance, cost, power, degrees of freedom control and sensing, etc

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 99 of 102	

Backup Charts

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 06-074	Version: 1.0
Title: System Architectural Considerations on Reliable Guidance, Navigation, and Control (GN&C) for Constellation Program (CxP) Spacecraft		Page #: 100 of 102	

3) Evaluating

- The goal is to model cost and complexity through weight.
- The complexity (and the cost) of the system is higher when it has many interconnections but **this relationship is not linear**. A correction factor was included to take into account the increase of complexity due to the number of connections → **connection weight (CW)**. A CW = 0 means that connections can be added without any impact on the global cost and complexity.
- The weights associated with connections were chosen to be consistent with the weights of sensors and computers:
 - $CW \leq 1/3$ weight of the average computer
- Analogously, a correction factor was included to take into account the increase of complexity when components of **different types** are used → **dissimilar component penalty**. A DCP = 0 means that different types of components can be used without any impact on the global cost and complexity.
 - $W = \text{weight \{sensors\}} + \text{weight \{computers\}} + N * CW + B * DCP$

where: N is the number of connections
CW is the connection weight
B is 0 if all the elements are of the same type and 1 otherwise
DCP is the dissimilar component penalty
- The weights associated with DCP were chosen to be consistent with the weights of sensors and computers:
 - $CW \leq \max \text{weight \{sensors, computers\}}$



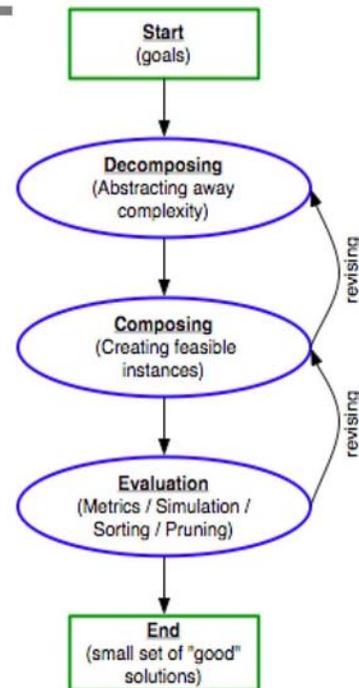
Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
101 of 102

A View of Systems Architecting:

- Systems Architecting starts with “Goals” and end with a “Set of Good Solutions”
 - “good” depends on what’s important to you
- 1st: Decompose the problem
 - Come up with an abstraction that is an stable representation of the system
 - This defines the architecture!
 - In this step, one should figure out the abstract processes and objects that are always part of the system
 - » List the range of specific solutions for each object and process
 - » This should be kept to a relatively small discrete set to start.





Title:

**System Architectural Considerations on Reliable
Guidance, Navigation, and Control (GN&C) for
Constellation Program (CxP) Spacecraft**

Page #:
102 of 102

A View of Systems Architecting:

- 2nd: Composing
 - How do the objects and processes interact?
 - Specifically, what combinations of specific objects and processes make up a valid system?
- 3rd: Evaluation
 - What are the system metrics?
 - » What makes this system “good” for you?
 - Metrics should be a function of the objects and processes in your decomposition
- The other “step”: revision
 - As you proceed down this concept of architecting, expect to revise the previous steps.

