A Candidate Architecture for Monitoring and Control in Chemical Transfer Propulsion Systems

Michael P. Binder
Sverdrup Technology, Inc.
Lewis Research Center, Group
Brook Park, Ohio

and

Marc G. Millis
Lewis Research Center
Cleveland, Ohio

Prepared for the
26th Joint Propulsion Conference
cosponsored by the AIAA, SAE, ASME, and ASEE
Orlando, Florida, July 16–18, 1990
A CANDIDATE ARCHITECTURE FOR MONITORING AND CONTROL IN CHEMICAL TRANSFER PROPULSION SYSTEMS

Michael Binder
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio 44142

and

Marc G. Millis
NASA Lewis Research Center
Cleveland, Ohio 44135

ABSTRACT

To support the exploration of space, a reusable space-based rocket engine must be developed. This engine must sustain superior operability and man-rated levels of reliability over several missions with limited maintenance or inspection between flights. To meet these requirements, an expander cycle engine incorporating a highly capable control and health monitoring system is planned. This paper discusses alternatives for the functional organization and the implementation architecture of the engine’s monitoring and control system. On the basis of this discussion, a decentralized architecture is favored. The trade-offs between several implementation options are outlined and future work is proposed.

Introduction

The United States has recently renewed its commitment to the exploration and utilization of outer space. Present plans for a space exploration initiative call for returning man to the Moon shortly after the turn of the century. This is to be followed by a permanent Lunar base, and the exploration of Mars and beyond. This venture requires a new class of space-based engines. These are to be man-rated (high reliability), reusable, and continuously throttleable. A basic description of the new engine as it is presently envisioned is given in Table 1. The requirement for space-basing is of critical importance to the development of the engine. Space-basing will seriously curtail the extent of between-flight maintenance and inspection operations, which are customarily performed on earth-based, reusable rocket engines such as the Space Shuttle Main Engines (SSME).

In order to maintain a high level of reliability in the space-based environment throughout several missions, an advanced Integrated Controls and Health Monitoring (ICHM) system is planned for the new engine.

The advanced features and growth requirements for the ICHM system are summarized in Table 2 (Ref. 1); they strongly suggest a capable condition monitoring and control architecture which is open to expansion. The functional organization upon which the architecture is based is called a framework (see Glossary also). This new, space-based engine is still in the definition phase, and no specifications yet exist for the ICHM system features. The discussion here assumes a highly capable ICHM system. Frameworks and architectures which hold potential for high throughputs, capability, and reliability are discussed. Recommendations for future work and further investigations are also offered.

The discussion proceeds as follows:

1) An ICHM General Framework and its constituent functions are described.

2) General features for the implementation of the general framework are listed.

3) The advantages and concerns of multi-processor implementation versus single-processor implementation are discussed.
4) Candidate organizations of multi-processor(s) to implement ICHM functions are described. Those considered are:
   a) Centralized
   b) Functionally de-centralized
   c) Regionally de-centralized

5) Criteria for evaluating these three architectures are defined.

6) A preferred architecture is selected on the basis of the evaluation criteria.

7) Various hardware options for interconnecting the processors in the system are described.

8) Recommendations are made for future investigation of the ICHM architecture.

Description of General Framework

In this section, potential functions for the general framework, shown in Figure 1, are described. This framework reflects the advanced capabilities desired for the ICHM system, as outlined in Table 2. In the figure, functions are represented by blocks; data and information between functions are represented by the connecting lines. Control function blocks are shaded to distinguish them from monitoring functions. Each major function block, numbered in Figure 1 for clarity, is described below. Monitoring functions are described first, followed by the control functions.

Signal-Conditioning (Block 10) and the Data-Reduction-and-Analysis (Block 5) perform most of the basic calculation functions required for monitoring. These functions include:

* Redundancy management & sensor validation, including analytic redundancy using model data.
* Data smoothing (removing noise)
* Parameter estimation
* Performance calculations
* Vibrational and rotational frequency analysis
* Trend determination and prediction
* De-trending (removing gross power changes that drown signal)

A real-time high-fidelity engine model (Block 4) is required for analytical redundancy and parameter estimation in monitoring functions. If necessary, the model can also be used for open loop control of the engine. Models describing internal operation of individual components may also be required to diagnose component condition.

The Limits block (Block 2) checks the sensor data and calculated parameters against red-line limits and warning levels. A red-line usually indicates an immediate shut-down is necessary to save the engine and vehicle. A warning indicates an anomalous condition that is not immediately catastrophic.

Critical-mode Pattern Recognition (Block 3) performs a fast correlation check between multiple parameters. Certain failures may not cause an individual parameter value to exceed its limit, but rather cause a characteristic pattern of changes in several parameters. Such faults would not trigger the limit detection logic (block 2). The critical-mode pattern recognition block recognizes these patterns, however, and warns the safety block (block 12).

Intelligent Monitoring (Block 1) applies both conventional and advanced algorithms to detect and diagnose faults and degradation modes. The intelligent monitor must also assess the severity of the faults detected. This block can also adaptively alter limits and parameters for the other monitoring and analysis functions. (Ref. 2 & 3) Some advanced techniques under consideration include:

* Pattern Recognition (neural networks for example)
* Expert Systems
* Qualitative reasoning (model-based algorithms)

The monitoring system is fully integrated with the control system. Control functions include:
* Safety (emergency cut-off/throttle-down) (Block 12)
* Pre-programmed valve-sequencing (Block 13)
* Basic Autonomic Control (Block 14)
* Performance Optimization (Block 15)
* Life Extension control (Block 16)

The Safety block uses signals from the limit detection logic (block 2) and from the propulsion system to indicate a shut-down or throttle-down. The Valve Sequencer (block 13) provides pre-programmed valve position-vs-time commands for start-up, shut-down, automated pre-start checks, and other standard changes in engine operation.

The Basic Autonomic control (block 14) oversees most engine control functions, including mixture ratio control and throttling. The Valve Translator is shown as part of this function; it converts thrust, mixture ratio, bypass flows, pressures, pump speeds, etc. into the appropriate valve positions. (Ref. 2 & 4)

Performance optimization (block 15) and Life Extension (block 16) functions coordinate with the propulsion system and the monitoring system to conserve propellant, maximize engine performance, and reduce unnecessary strain on engine components. The recommendations from these blocks are screened by the basic controller (block 13) to prevent conflicts with mission safety during critical vehicle maneuvers and in the event of engine degradation.

The general framework of Figure 1 may be implemented in a number of different architectures. Features and options to be addressed by these architectures discussed in the following sections.

General Implementation Features of General Framework

There are a number of general features which must be satisfied by any proposed architecture. These are briefly described as follows.

1) Sensor values that may indicate red-line shut-down must have a connection directly to the limit detection logic (no competition for bus access). The limit detection logic must similarly be connected directly to the relevant controller to effect shut-down when red-line is indicated.

2) The system elements shall use digital electronics for monitoring and control functions. This will allow greater flexibility through programming. Each processor in the system, whether it is centralized, de-centralized, or otherwise must be locally or remotely programmable to accommodate changes in mission profiles, engine characteristic and operating modes (start, stop, idle, mainstage, transients, etc.). Digital-analog hybrids may be used where appropriate (autonomic control for example).

3) The ICHM system elements (processors) must have at least one channel each of electronic redundancy.

4) The ICHM system must guard against false alarms as well as missed detections. This shall be accomplished through self-checking sensors and electronics, and use of analytic redundancy.

5) Data from all sensors and certain analytic results should be available at all times for on-board data storage and for telemetry link with ground support operations.

Multiple-vs-Single Processor Implementation

Traditionally, both rocket and jet engine monitoring and control have been performed by a single processor. As health monitoring and control function requirements have grown, however, multi-processor implementation has become an attractive option to increase system throughput.

Multi-processor implementation permits increased throughput by performing several operations simultaneously (in parallel), Multi-processors also offer a potential for greater flexibility, reliability, and range of function than in a single processor configuration. (Ref. 5)

When considering multi-processor systems, however, certain factors
should be considered. Although multiple processors allow one or more processors to fail without total system failure (graceful degradation), greater system complexity increases the probability that some component will fail. Also, multi-processor implementations must be designed to minimize inter-process communication; communication overhead can significantly reduce advantages in throughput. These types of trade-offs should be investigated further in future research.

With proper design of the architecture, the potential gains cited above favor a multi-processor implementation for the ICHM system. Different options for allocating ICHM functions amongst the multi-processor resources are discussed below.

**Description of Specific Candidate Architectures**

The architectures discussed in this section are derived from a much larger array of conceivable implementation options. Some architectures were eliminated on the basis of expected response time. Other options were incorporated as features in the remaining architectures. In the end, three basic frameworks were left; these are referred to as centralized, functionally de-centralized, and regionally de-centralized. A brief description of each framework is given below.

This section describes the configuration of processor hardware and software, and the inter-processor communication. The interconnection hardware and protocols are discussed in later sections.

**Centralized**

**Basic Concept:** In a centralized implementation, the multi-processor system acts as a single integrated entity. The functions and programs comprising the ICHM system are dynamically allocated among the processors. Separate multi-processor systems are used for monitoring and control functions (see Figure 2). Some signal processing or autonomic control could be done at the sensor or actuator levels, but all other functions would be performed by the engine-level monitor and controller units.

Data and analysis functions are assigned to processors dynamically in order to maximize efficient utilization of resources and minimize response time. The processors in such an architecture are nearly identical; each can perform several different functions as required. In the event of a processor failure (including redundant channels), the monitor/controller can be immediately reconfigured, creating a potential loss in computation speed but without loss of data or function. (Ref. 5) Because there are no well defined, static assignments of functions or data to each processor, this configuration is still considered centralized. (Ref. 6)

**Functionally de-centralized**

**Basic Concept:** In a functionally de-centralized framework, dedicated processors are assigned to perform each analysis, diagnostic, and control function. Each processor's hardware can be optimized to its specific function, increasing the speed of the monitor and controller dramatically. An example is given in Figure 3. This organization addresses the fact that sensor data is manipulated and analyzed in a number of different ways. Likewise, engine valve settings are derived from a confluence of different algorithms to guarantee safety while simultaneously optimizing engine operation and life. Each block in the general framework may utilize several specialized algorithms. (Ref. 6 & 7)

By allocating the instructions to perform each ICHM function to a specific processor, the inter-processor communication overhead can be reduced; dynamic allocation of resources is not performed. Because the output from one function may be used by several others throughout the system, there will still be considerable inter-process transfer of data.

By using specialized processors, system performance can be increased at the cost of some flexibility; a system utilizing special-purpose hardware may
no longer be fully reconfigurable as in
the centralized multi-processor
configuration. It is important to note
here that the term 'processor' need not
refer to a full digital computer but
may be implemented in both analog and
digital circuitry. It can include FFT
(Fast Fourier Transform) chips, simple
gate logic, etc. (even instruction
caches).

Regionally de-centralized

Basic Concept: In a regionally de-
centralized monitor, the data analysis,
diagnostic functions, and control
calculations are organized according to
engine component or subsystem (referred
to generally here as a 'region'). An
example is given in Figure 4. Each
region processor (or set of processors)
can be optimized to monitor a specific
part, component, or subsystem. For
monitoring and control purposes, these
region processors are self-contained
black-boxes; only that information and
data needed by other systems is passed
out of the region. This should reduce
the network and processor loads
significantly. Even though region
processors pass only relevant
information and data to the other
processor blocks, all sensor data and
meaningful processor outputs are also
recorded or transmitted for ground-
based analysis. (Ref. 1-4,8-10)

The internal monitoring processes for
each component are independent of
those in other system components.
Regional de-centralization takes
advantage of this natural separation
and minimizes the volume of inter-
process communication required.
Communication overhead is a major
limitation to multi-processor system
speed. As with functional de-
centralization, hardware can be
optimized to increase speed, but with
some loss of flexibility.

The regionally de-centralized system
modularity facilitates design of ICHM
hardware and software. Additionally,
the structure of a regionally de-
centralized architecture parallels that
of the engine itself. Each engine
component design group can therefor
have more input in specifying
fault/degradation detection methods and
control characteristics for that
component; each monitor/controller is
designed to interact with the larger
system in a pre-defined manner.

The inherent modularity of the
regionally de-centralized architecture
could also allow each component
monitor/controller to be removed from
the engine's instrument bay, and follow
its respective component during
maintenance and engine refurbishment
operations (it may even be built into
the component). This would be
especially important if neural networks
were used in the monitor/controller.
Regional de-centralization may better
facilitate component tracking and
maintenance. (Ref. 8)

Regional de-centralization also
creates an option for physical
distribution of the ICHM architecture.
In the physically distributed system,
regional processor hardware would be an
integral part of the component it
monitors. The processors are therefor
distributed over the entire engine.
Alternatively, the dedicated processors
for each component can be located in a
central engine box, away from their
respective components (regions).
Physical distributed components
processors may require more shielding
and be exposed to harsher environmental
conditions than their centrally located
counterparts. A harness to carry
sensor signals from the component to a
central unit must always be provided
for data storage and telemetry
purposes.

Evaluation of
Candidate Architectures

The critical evaluation criteria for
a rocket engine ICHM system are, in
order of priority:
  1) Reliability - The primary purpose
      of an advanced ICHM system is to
      improve engine reliability. The effect
      of the ICHM system on the engine's
      overall reliability consists of three
      factors:
      a) ICHM capability and throughput -
         Any ICHM system must safely shut-
         down the engine to avoid
catastrophic failure. An advanced
         ICHM system should also help to
extend the operation of a degrading engine to complete the mission, and enhance the efficiency and service life of the engine as well.

b) Probability of ICHM failure - The hardware and software of the ICHM system itself must be very reliable and fault-tolerant (capable of functioning even when several system components fail).

c) Ratio of valid error detections to false alarms - While the system must be sensitive enough to detect faults, it must not be so sensitive that it generates false alarms.

2) Flexibility - A second priority is to make the ICHM system flexible to changes in the mission envelope, engine operating characteristics, and to allow product improvements in engine and component designs.

3) Weight - In an engine of this size (20 klbs), weight and volume are at a premium, the ICHM must be as light and compact as possible.

The above criteria are used in Table 3, which gives an evaluation of each candidate relative to the other architectures described here. The evaluations are based on qualitative judgements only. In order to make more quantitative evaluations, a properly scaled rating system will have to be established for each criterion. The relative importance of each architecture feature to the fulfillment of the various criteria must also be established. Ultimately, the importance of each criterion itself must be gauged in the context of NASA program requirements.

Other considerations - It is important not to overlook considerations such as development cost/time, and maintenance requirements. Typically a system which is more capable and flexible will have greater initial costs but lower upgrade and operations costs. This trade-off has not been explored in depth and requires further investigation. Ultimately, the inherent engine reliability will help determine how much ICHM is really needed; a very robust engine system will require less monitoring than an engine with narrower safety margins.

In the next section, these criteria are applied in order to select a favored architecture from among the three configurations discussed previously.

Selection of an Architecture

A multi-processor monitor and controller system has potential advantages in throughput, capability and flexibility. Furthermore, de-centralized architectures promise potential throughput, capability, and maintainability which are superior to those of the centralized multi-processor system. Redundancy of hardware and software, and a capacity for graceful degradation should guarantee sufficient reliability of the de-centralized systems.

Due to lack of specific, well-defined requirements for ICHM, the choice between the two de-centralized architectures is not clear. Either the functional or the regional de-centralized configuration appear able to satisfy ICHM general requirements.

Presently, the regionally de-centralized architecture is favored in order to maximize throughput, capability, and flexibility. In this architecture, engine components requiring highly interconnected monitoring functions are, in turn, monitored by tightly coupled multi-processors; this tends to reduce network traffic, enhance throughput, and facilitate diagnostics. Due to it's highly modular structure, this configuration allows changes to one engine subsystem with minimal impact on the rest of the system. Overall ICHM reliability is comparable to that of the functionally de-centralized architecture.

Detailed numerical analyses of system timing, network loading, support circuitry requirements, and electronic packaging are beyond the scope of this paper. These issues must be investigated in the course of future research, as recommended in following sections.

During the research and technology program (including a test-bed engine system demo), de-centralization will
first be implemented in software rather than special-purpose hardware where possible. This will accommodate quick changes for experimentation. The required throughput in this case might be attained using general-purpose processors in greater numbers or with greater throughput per processor. Certain hardware specialization will be used, however, as performance requires or convenience indicates (FFT boards, for example). Having described alternative methods of allocating ICHM functions to various processors, methods of implementing the connections between system elements is now discussed.

**Inter-connection Hardware and Protocols**

In this section, issues of interconnecting the sensors, actuators, and processors of the ICHM system (including propulsion and vehicle system interfaces) are discussed.

The connection scheme should be as flexible and modular as possible. The types of information that must be accommodated are:

- Data from sensors
- Data and Status-codes between processors
- Re-programming and reconfiguration data and information from high to low level processors.

In the de-centralized architectures, each processor need not be directly connected to every other. A single standard bus system may be too slow and unnecessarily general for rocket engine ICHM applications.

One possible solution would be to have dedicated signal lines for every sensor and inter-processor link, including sufficient extras for redundancy and expansion. To make the system flexible, signal cables (parallel or serial) might be used between processors (Figure 5a). Unfortunately, this method would not be very light weight nor modular from an operations standpoint. An alternative possibility employs interchangeable connection modules which define the signal paths (Figure 5b).

A more promising solution would be to use two or more standard data/address/control busses (Figure 5c). This would allow faster access between critical modules as well as connection of the overall system. One special option for greater flexibility includes busses that can be divided or connected through switch boxes (small systems only) or interfaces (with bus signal repeaters). (Ref. 5) A bus approach, unlike the dedicated signal line concepts, would allow use of commercially supported hardware and protocols. This would reduce costs and development time. In any case, the engine ‘rack’ must include extra slots and inputs for additional sensors and processors. (Ref. 6)

In addition to the interconnections for real-time monitoring and control, there should be at least one system-wide standard computer bus to perform off-line operations. Some candidate operations include re-programming the ICHM system for a special mission profile, or for a major engine change. Also, this ‘off-line’ bus could be used to transfer information to and from ground-based maintenance facilities (component performance characteristics, degradation modes observed, usage tracking information, etc.).

**Recommendations for Future Work**

A favored architecture for the new ICHM system has been outlined and implementation issues have been discussed. There is much work yet to be done before a detailed ICHM system can be designed for test-bed or flight-rated engines. Several of the most important work elements are outlined below:

1) Specify requirements (trade-studies may be required in several instances):
   a) Define Mission requirements: mission duration, engine operation sequences, reliability, space-basing infrastructure, etc.
   b) Vehicle requirements: size and weight, propulsion system configuration (including auxiliary propulsion for
steering), tanking, aerobraking, etc.
c) Engine requirements: thrust accuracy, throttling range/rates/accuracy, reliability, service life, maintenance, weight, environmental extremes, cost, etc.
d) ICHM requirements
- system functions and capabilities
- speed of calculation/decision
- weight limitations
- cost
- flexibility (to mission, engine changes, ICHM upgrades
2) Specify ICHM algorithms and sensor list.
3) Specify ICHM processors (may involve several different types), interconnection hardware and system design.
4) Construct computer models of candidate architectures. Use these models to extend qualitative work already completed (described herein). Benchmark performance and reliability of architectures; compare size, weight, and flexibility.
5) Conduct trade-studies to determine a realistic system:
   Speed, Advanced capability, and Flexibility
   vs.
   Weight, size, cost, and general-purpose hardware
6) Construct software based on specified algorithms.
7) Simulate full architecture in software (using ICHM software generated in #5 above). Validate architecture and software using dynamic engine simulation.
8) Design and construct hardware for system based on specifications. Final organization of architecture will be set by results of computer models, benchmarking, and experiments.
9) Test full architecture hardware and software using a real-time dynamic engine simulation.
10) Test hardware, software, and architecture generated in steps above as part of technology test bed engine.

Conclusion
This paper has presented a qualitative discussion of the issues which will define the Integrated Control and Health Monitoring (ICHM) system architecture for the new space-based, reusable rocket engines. The operability, reliability, and flexibility requirements for this engine indicate an advanced ICHM architecture.

A general framework has been defined which describes ICHM advanced functions. This description includes a list of techniques and algorithms that will be included as part of the ICHM system. Because many ICHM functions are computationally intensive, a multi-processor implementation is favored.

Three basic multi-processor architectures have been identified: centralized, functionally decentralized, and regionally decentralized. The evaluation criteria for selecting an architecture have been outlined and trade-offs between the three architectures discussed. Several options were described for making flexible but fast interconnections with the sensor set, and for inter-processor communication.

The conclusion of this paper is that a regionally decentralized architecture is favored in order to maximize throughput, capability, flexibility, and maintainability of the ICHM system. A hierarchical, multiple bus system appears to have the greatest promise for processor interconnections. Further investigation is required to quantitatively verify these conclusions.
References


4) Status reports and technical reviews on Phase I of NASA contract NAS 3-25557, "Reusable Rocket Engine Intelligent Control Framework", conducted by Rockwell International.


Glossary

**Framework** - A description of the system functions, organization, and data flow.

**Architecture** - A description of the electronic implementation of the system framework (may include detail to chip level).

**Processor** - A processor may be a CPU, signal processing chip, logic circuit, instruction cache, neural network, or other devices which process data and information (both analog and digital).

**Centralized** - all system functions are performed by a central unit. If the central unit is a multi-processor, there is no static assignment of function or data to each proc.

**Functionally de-centralized** - each system function for entire engine is performed by a specific processor. Processor hardware may be optimized to its specific function.

**Regionally de-centralized** - each major region or component of the engine has a dedicated processor or multi-processor for monitoring and control. Processor hardware may be optimized by both component and function. Processors may be located in a central bay or distributed about engine.
Table 1
Basic Space-based Engine Characteristics (Ref. 1)

* Human Rated (high reliability - exact quantitative definition varies)

* Engine cycle: Hydrogen/Oxygen Expander (no gas generators or pre-burners; turbines driven by 'warm' propellants from regenerative cooling)

* Re-startable/Reuseable (min. of 5 missions)

* Space-based (minimal inspection, no EVA, and no scheduled maintenance between missions, zero-g start, prolonged exposure to space)

* Continuously throttleable for Lunar descent (thrust ratios as high as 20:1 may be required).

* Thrust: presently projected as 20 klbs class.

* Specific Impulse (Isp): 485 sec. with extendable high area-ratio nozzle.
  : 465 sec. with short nozzle

* Engine flexible to allow different mission duties (orbital transfer, lunar transfer, lunar descent/ascent, mars transfer, mars descent/ascent, trans-mars injection?)

* Engine flexible to allow high mixture ratio operation (to take advantage of Lunar LOX potential).

Table 2
Desirable ICHM System Features (Ref. 1)

* Minimum Requirements
  * Basic safety, emergency shut-down
  * Expander cycle operation (with deep throttling capability and zero-g start)
  * Automated pre-start check-out of engine and ICHM.
  * Automated post-shutdown diagnostic check (using data from last flight)
  * Space environment and effects monitoring.
  * Engine ICHM system fully integrated with engine cluster and vehicle controls.

* Growth Requirements
  * Real-time in-flight engine diagnostics & prognostics
  * Adaptive controls (integrated with monitoring and diagnostics): - to optimize engine performance, propellant utilization - to extend mission effectiveness - to extend engine service life
  * Automated life prediction (ground-based?)
### Table 3
Qualitative Evaluation of Architectures

<table>
<thead>
<tr>
<th>CRITERIA</th>
<th>Centralized</th>
<th>Functionally De-centralized</th>
<th>Regionally De-centralized</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1) Reliability</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Throughput</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Minimal interproc. signals?</td>
<td>POOR</td>
<td>FAIR</td>
<td>VERY GOOD</td>
</tr>
<tr>
<td>ii) Hardware can be optimized to specific functions?</td>
<td>POOR</td>
<td>VERY GOOD</td>
<td>VERY GOOD</td>
</tr>
<tr>
<td>iii) Utilization of all resources maximized?</td>
<td>VERY GOOD</td>
<td>FAIR</td>
<td>FAIR</td>
</tr>
<tr>
<td>b) ICHM Reliability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Redundant hardware/software used?</td>
<td>YES/GOOD</td>
<td>YES/GOOD</td>
<td>YES/GOOD</td>
</tr>
<tr>
<td>ii) Minimum number of connections and processors?</td>
<td>GOOD</td>
<td>FAIR</td>
<td>POOR</td>
</tr>
<tr>
<td>iii) Multi-proc. resources reconfigurable?</td>
<td>VERY GOOD</td>
<td>GOOD</td>
<td>FAIR</td>
</tr>
<tr>
<td>iv) No sources of single-point failure?</td>
<td>FAIR</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>What are they?</td>
<td>failure of scheduler may cause complete system failure</td>
<td>GOOD</td>
<td>some loss of function</td>
</tr>
<tr>
<td>c) Minimal false-alarm rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Redundant cross-check of sensors and processors?</td>
<td>FAIR</td>
<td>GOOD</td>
<td>VERY GOOD</td>
</tr>
<tr>
<td>ii) Internal self-check (sensors and proc.)?</td>
<td>YES/GOOD</td>
<td>YES/GOOD</td>
<td>YES/GOOD</td>
</tr>
<tr>
<td><strong>2. Flexibility of ICHM to Changes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Mission changes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Processors programmable for different operating ranges?</td>
<td>GOOD</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>b) Changes in Engine Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Programmable instructions and params?</td>
<td>GOOD</td>
<td>GOOD</td>
<td>VERY GOOD</td>
</tr>
<tr>
<td>ii) Easy to add or change sensors and functions?</td>
<td>GOOD after system prog.</td>
<td>GOOD after each function proc. programming</td>
<td>VERY GOOD after only affected component proc.</td>
</tr>
<tr>
<td>How are changes made?</td>
<td>GOOD</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>c) Upgrades to ICHM Capability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Programmable instructions and params?</td>
<td>FAIR</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>ii) Easy to add sensors?</td>
<td>GOOD</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>iii) Easy to add functions?</td>
<td>GOOD</td>
<td>VERY GOOD</td>
<td>VERY GOOD</td>
</tr>
<tr>
<td>How are functions added?</td>
<td>reprogram system</td>
<td>add new function processors</td>
<td>add new function proc. to affect comp.</td>
</tr>
<tr>
<td><strong>3. Weight of ICHM Hardware</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) No duplication of non-redundant hardware?</td>
<td>VERY GOOD</td>
<td>GOOD</td>
<td>FAIR</td>
</tr>
<tr>
<td>b) Minimal packaging, wiring, shielding?</td>
<td>VERY GOOD</td>
<td>GOOD</td>
<td>FAIR</td>
</tr>
<tr>
<td><strong>4. Other Considerations</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Low development costs?</td>
<td>FAIR</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>ii) Low operating and upgrade costs?</td>
<td>FAIR</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>b) Maintanability (modular?)</td>
<td>GOOD</td>
<td>GOOD</td>
<td>VERY GOOD</td>
</tr>
</tbody>
</table>
ICHM GENERAL FRAMEWORK

Figure 1
The condition monitor and controller blocks each represent a discrete, tightly coupled multi-processor.

Figure 2
Centralized Framework

Each block in this figure represents a discrete processor or electronic system.

Figure 3
Functionally De-centralized Framework
Data storage & Telemetry

Engine

Fuel Turbopump

Oxidizer Turbopump

A/D

Signal Cond.

Sensors

Drivers

Actuators

Valves

Each component block may be a dedicated multi-processor system

Figure 4

Regionally De-centralized

5a. Cabled Interconnects

processors or electronic modules

cables

5b. Replaceable Backplane with dedicated signal paths.

A

A

Section A-A

5c. Multi-level bus system

Bus 1

Bus 2

Bus 3

inter-bus links: each bus may use a special mapping or extended address to access other busses

bus signal repeater or interconnect as required

Figure 5

Three Candidate Interconnect Schemes
To support the exploration of space, a reusable space-based rocket engine must be developed. This engine must sustain superior operability and man-rated levels of reliability over several missions with limited maintenance or inspection between flights. To meet these requirements, an expander cycle engine incorporating a highly capable control and health monitoring system is planned. This paper discusses alternatives for the functional organization and the implementation architecture of the engine's monitoring and control system. On the basis of this discussion, a decentralized architecture is favored. The trade-offs between several implementation options are outlined and future work is proposed.