Real-Time Implementation of Intelligent Actuator Control with a Transducer Health Monitoring Capability

Dipan Jethwa[†], Rastko R. Selmic[†], and Fernando Figueroa[‡]

[†] Department of Electrical Engineering
Louisiana Tech University
Ruston, LA 71272, USA
Email: rselmic@latech.edu
[‡] NASA John C. Stennis Space Center
Stennis Space Center, MS 39529, USA

***Abstract*—This paper presents a concept of feedback control for smart actuators that are compatible with smart sensors, communication protocols, and a hierarchical Integrated System Health Management (ISHM) architecture developed by NASA's Stennis Space Center. Smart sensors and actuators typically provide functionalities such as automatic configuration, system condition awareness and self-diagnosis. Spacecraft and rocket test facilities are in the early stages of adopting these concepts. The paper presents a concept combining the IEEE 1451-based ISHM architecture with a transducer health monitoring capability to enhance the control process. A control system testbed for intelligent actuator control, with on-board ISHM capabilities, has been developed and implemented. Overviews of the IEEE 1451 standard, the smart actuator architecture, and control based on this architecture are presented.**

***Keywords*: smart sensors, smart transducers, IEEE 1451, ISHM**

I. Introduction

Advances in science and technology are enabling the development of increasingly complex electromechanical systems, such as the Space Shuttle, rocket engine test platforms, and the International Space Station. These systems are characterized by high complexity; a very large numbers of sensors, actuators and other components; reduced operating margins; high reliability requirements; and fast fault propagation. This makes it difficult to safely, and cost effectively, operate these systems.

Networks of sensors and actuators gained significant attention in recent years [1], [2], [4], [5] due to their applicability to a wide range of aerospace, environmental, and military applications. However, conventional sensors and actuators utilize diverse, sometimes proprietary, data and communication interfaces. This only adds to the complexity of the system. Recent developments in distributed smart sensors and actuators have enabled significant advances in ISHM capabilities [7], [8]. Sensors incorporating microcontrollers have enabled data processing at the sensor and the development of smart sensors that are fault-aware and fault-tolerant [8]. Smart components that provide a standard interface improve system reliability and lower the cost of installation, operation and maintenance [3]. Improvements in overall reliability can be accomplished by incorporating and utilizing awareness of subsystem conditions to assess the health and status of the system. ISHM can provide a health assessment of each component (transducer) in the system by ascertaining the quality of the information provided by the sensors by considering additional parameters [6], [11], [12], [13].

In this paper we discuss the feedback concept for smart sensors and actuators based on IEEE 1451 standard, integration of such actuators into an ISHM structure, and the control of smart actuators. This work adds to developments at NASA's Stennis Space Center, which address the utilization of smart sensors for integrated health management in safety critical applications. Research and development in this area is based on recent development of IEEE 1451 standard, which we shortly review in the next section.

II. IEEE-1451 Overview

The IEEE 1451 set of standards was developed under leadership from the National Institute of Standards and Testing (NIST). A detailed description of the standards by Lee can be found in [10]. Use of these standards in ISHM is described in [7]. The standard has been divided into six subgroups given in Table 1.

Table 1. Overview of the IEEE 1451 family of standards [10].

IEEE 1451.0	Defines a set of commands, operations, and transducers electronic data sheets (TEDS) for the overall standard. The access to the devices is specified and it is independent of the physical layer.
IEEE 1451.1	Defines communication with the Network Capable Application Processor (NCAP) such as client-server or client-client type of communication between NCAP and other network devices or between several NCAPs.
IEEE 1451.2	Definition of TEDS and an interface between transducer and the NCAP.
IEEE 1451.3	Specifies the interface between the NCAP and smart transducers and TEDS for multi-transducers structure connected to the bus.
IEEE 1451.4	Defines how analog transducers can be interfaced with microprocessors, i.e. specifies TEDS implementation for analog devices.
IEEE 1451.5	Specifies a transducer to NCAP interface and TEDS for wireless communication scenarios.
IEEE 1451.6	Defines a transducer-to-NCAP interface and TEDS using high speed consolidated auto network bus with multiple controllers present in the system.

This work was supported in part by the LaSPACE NASA Grant # 32-0967-58208.

Smart actuators following IEEE 1451 standard should support the NCAP function and should include a TEDS either in software or hardware. The term "intelligent" indicates adherence not only to the IEEE 1451 standard, but also the inclusion of a Health Electronic Data Sheet (HEDS). This addition enables actuators to identify their current health status and choose appropriate control actions based upon this information.

III. Intelligent Actuators in ISHM

The smart actuator module is based on the IEEE 1451 standard [9] and is analogous to the smart sensor structure. A smart actuator module includes a Smart Transducer Interface Module (STIM), an NCAP, a memory, a communication module, a power module and an actuator interface. The STIM usually contains an actuator, an actuator interface circuit with the driver, filtering circuitry, and a TEDS (Figure 1). The TEDS contains basic sensor or actuator parameters such as conversion units, calibration data, upper and lower data limits, number of channels, etc. The STIM module serves as a controller for actuators and a communication gateway between the actuator and the NCAP. It also converts digital data into the analog signal required to drive an actuator. Current standards support the usage of multiple actuators on one STIM module.

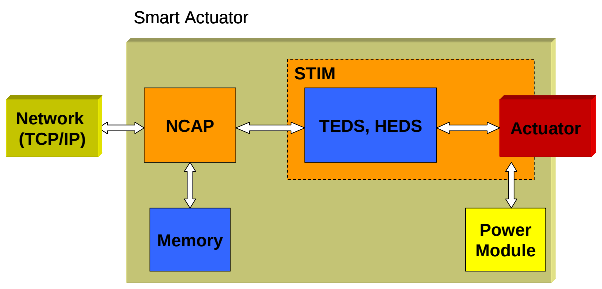

Figure 1. Smart actuator block diagram.

The actuator interface consists of an actuator device driver that provides the necessary signals for actuator operation. A power module provides the power for the whole actuator node. The NCAP is a dedicated gateway between the smart actuator module and the communication network. It communicates with the STIM, provides data conversion between the network and the STIM, and communicates with other smart sensors and actuators on the network. In a distributed sensor and actuator network, the NCAP serves as a node controller and interface to the dedicated network.

Recently, NASA researchers have introduced health electronic data sheet (HEDS). The HEDS contains parameters that can be used to detect a variety of anomalies that can occur in sensors or actuators such as over-range, noise, saturation, deadzone, drift, bias, etc. Instead of traditional control systems where sensors feed data to controllers which in turn provide signals to actuators based on sensor data and desired system performance; a concept of smart HEDS-capable devices requires a new control paradigm with bus-type of information flowing between sensors, controllers, and actuators.

Intelligent Control of Smart Actuators

Conventional control systems include controllers that assume that components are healthy. However, there is a need to consider other scenarios in order to achieve more reliable and safer control. As a result of recent developments in smart sensors technology, additional information about the health of the sensor, accuracy of the data, statistical deviations, the status of neighboring sensors, and more, is available for the central controller to use. Computational processing power can be "attached" closer to the source of information, in case of smart sensors, or closer to the point of action in case of smart actuators. Smart actuators are capable of compensation, self-calibration, health diagnosis, self testing, and communication. New control structures are needed that will focus on health diagnosis, detection of anomalies and failures, and compensation of those.

The IEEE 1451 architecture also allows a distributed control environment. It simplifies the development and implementation of communication with the transducers. IEEE 1451 also provides a remote monitoring capability that is highly demanded by current complex systems.

IV. Health Electronic Data Sheet (HEDS)

HEDS implementation requires that each transducer have a small database type of information pertaining its health status. We have implemented the HEDS in a new packet structure shown in Figure 2. The health packet contains fields that are common to all transducers in the system while providing sufficient information to assess the health of the transducers for the set of faults considered in this implementation. The packet structure fields are briefly described below:

UPPER RANGE (2 bytes, read-only): Indicates the extreme magnitude of the primary data value that can be handled by a system transducer. It is a read-only field.

DEVIANCE (2 bytes, read-only): Indicates the maximum allowed difference between the old and the new standard deviation, calculated for a set of recent transducer data.

HISTORY POINTS (2 bytes, read-write): Keeps a history of the number of times the transducer is accessed.

FLAGS (2 bytes, read-write): The most significant byte corresponds to the transducer number, while the least significant byte corresponds to the type of fault that persists in the transducer. A unique single byte hexadecimal representation for each type of transducer fault is predefined. For example, if transducer 2 is experiencing excessive noise in its readings (a fault type 3), then a fault type has a 2-byte FLAG value of '*x*0203'.

STANDARD DEVIATION (2 bytes, read-only): Holds the acceptable standard deviation value for the transducer data calculated under normal, healthy system conditions.

OFFSET (2 bytes, read-only): Holds the offset in each transducer of the plant measured during the initial conditions of the plant operation.

CONSTANT (2 bytes, read-only): Number of recent consecutive iterations since the transducer data has remained constant.

UPPER RANGE	DEVIANCE	HISTORY POINTS	FLAGS	STANDARD DEVIATION	OFFSET	CONSTANT	SAMPLE MEANS	DATA	PADDING
2 BYTES	2 BYTES	2 BYTES	2 BYTES	2 BYTES	2 BYTES	2 BYTES	3*w* BYTES	3*w* BYTES	variable

Figure 2. Health Electronic Data Sheet.

CHANNEL NUMBER	CHANNEL SETUP TIME	SENSITIVITY	MANUFACTURER ID LENGTH	MANUFACTURER ID	SCALING	PADDING
1 BYTE	2 BYTES	2 BYTES	1 BYTE	variable	2 BYTES	variable

Figure 3. Transducer Electronic Data Sheet.

SAMPLE MEAN (3w bytes, read-write): Each value is an average of the transducer data set belonging to a window of length *w*. The field holds a sample means of *w* windows, with each mean calculated over a data set of *w* samples.
STANDARD DEVIATION (2 bytes, read-only): Holds the acceptable standard deviation value for the transducer data calculated under normal, healthy system conditions.
OFFSET (2 bytes, read-only): Holds the offset in each transducer of the plant measured during the initial conditions of the plant operation.
CONSTANT (2 bytes, read-only): Number of recent consecutive iterations since the transducer data has remained constant.
SAMPLE MEAN (3w bytes, read-write): Each value is the average of the transducer data set belonging to a window of length *w*. The field holds a sample means of *w* windows, with each mean calculated over a data set of *w* samples.
DATA (3w bytes, read-write): Each value corresponds to the transducer data obtained for *w* number of samples.
PADDING (read-write): This field is multiple bytes in length. It contains all zeroes and is reserved for future purpose.

All of the above HEDS parameters are considered for the health assessment of the transducer using the health check functional layer in the algorithm.

V. Implementation of TEDS and HEDS

We have implemented a prototype structure of TEDS and HEDS on a testbed shown in Figure 5. In the development phase HEDS was implemented on a host PC; however, future ISHM will have both HEDS and TEDS implemented on the STIM. TEDS includes the following fields (Figure 3):
CHANNEL NUMBER (1 byte): Contains the channel number over which that transducer is interfaced with the STIM.
CHANNEL SETUP TIME (2 bytes): Minimum idle time required between two consecutive accesses of the transducers using the NCAP. This is essential to prevent the system from locking up due to processing delays
SENSITIVITY (2 bytes): Maximum physical quantity to be measured divided by the maximum calibrated voltage corresponding to that quantity.
MANUFACTURER ID LENGTH (1 byte): The length of the field that follows this field.
MANUFACTURER ID (variable length): Hexadecimal equivalents of the ASCII characters that comprise the transducer manufacturer's name.
SCALING (2 bytes): Scaling factor required by the controller needed for the access to transducers.
PADDING (variable length): Zeros to fill TEDS packet to 24 bytes in overall length.

The TEDS contains information that remains static during the entire course of the controller operation. Data are not updated and thus does not need to be checked during every access made to the transducer (TEDS checked after every ten transducer accesses).

VI. Controller Functions

Overview: Given a desired reference signal and the status of the system, the controller calculates the necessary actuator signal and issues a command to smart actuators. The command is transmitted through the Ethernet via the NCAP and STIM.

As an illustrative example we have implemented a discrete PID controller given by

$$y(k) = y(k-1) + P(e(k) - e(k-1)) + Ie(k) + D(e(k) - 2e(k-1) + e(k-2))$$

where y is the controller output, e is the error between desired reference signal and obtained output, and P, I, and D are standard proportional, integral and derivative gains. Note that the PID algorithm is just a part of the smart actuator controller. Any other controller can be used and this is not an emphasis of this paper.

The overall controller consists of a smart transducer read layer that reads data from the NCAP. Data from the TEDS and the HEDS are then analyzed and processed providing the supervisory controller with the overall system awareness capability. The supervisory controller can have various degrees of complexity and many research results already exist in this area. The supervisory controller chooses an appropriate controller mode based on the status of the sensors and the actuators. For example, a faulty sensor requires the supervisory controller to pick a plant controller designed for that fault. In case of a dead sensor, an open-loop controller is chosen. The controller then feeds data to the data-write layer which provides necessary format conversion and then sends it to the NCAP.

Algorithm Description: This section describes a control algorithm which incorporates the IEEE 1451 standard and the ISHM capability. The algorithm was implemented in MATLAB. The MATLAB functions are applicable to any plant or any control strategy. Any typical plant module which is controlled from a remote PC using IEEE 1451 can be interfaced to this architecture by replacing the block named *Plant*.

Figure 4 shows an actuator controller that uses health information (left) and the client closed loop control model with the presence of the IEEE 1451 prototyping kit (right). The two major blocks communicate via Ethernet.

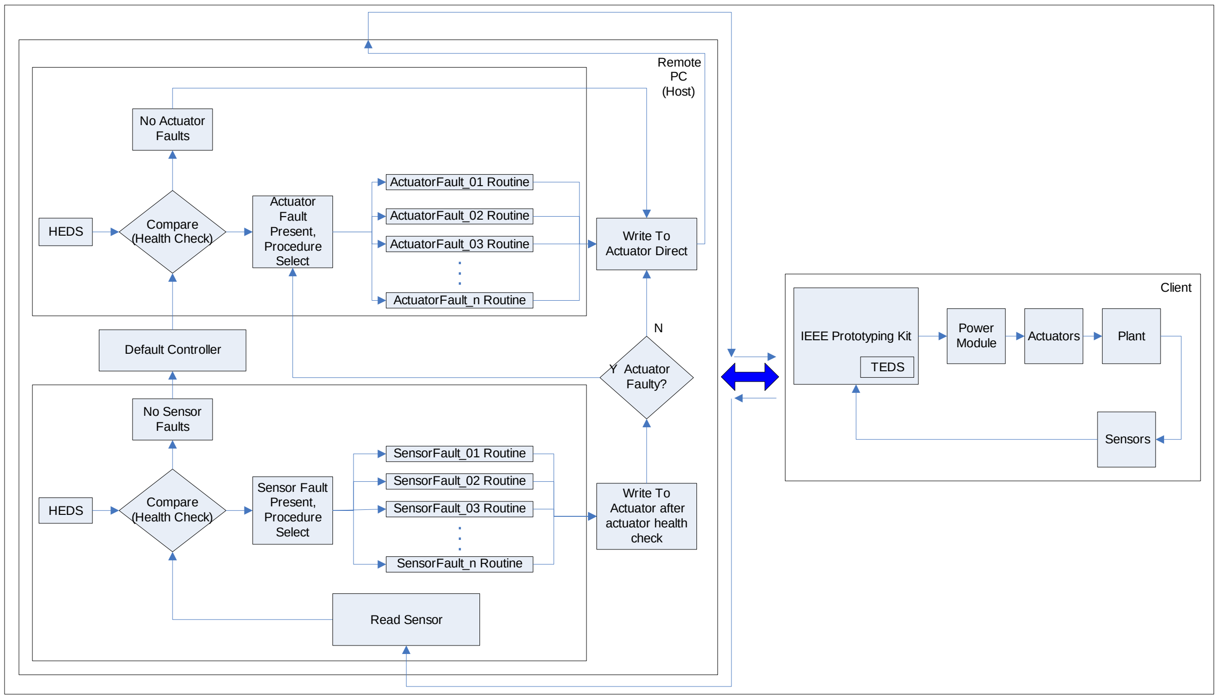

Figure 4. Block diagram representing the program flow and communication links.

A description of the algorithm using ISHM and HEDS is provided below.

- The NCAP of the IEEE 1451 module is initialized. The HEDS packet structure representing the values for healthy conditions for each transducer are initialized and stored at the host computer.
- The plant configuration parameters are then initialized. The parameters corresponding to the plant dimensions and component dimensions are also defined and initialized.
- The following steps are repeated during a control process, unless the attention bit is detected high.

1. Sensor information is read from the remote host.
2. Data are extracted and converted.
3. Data scaling and unit conversion is performed.
4. Data analysis using HEDS information is performed.
5. The algorithm identifies faults, if any, at the transducer.
6. Depending upon the identified fault, the HEDS *flag* field is updated with the corresponding status.
7. Depending upon the identified fault, a corresponding function for tolerating or correcting the fault is selected.
8. The HEDS field values are updated.
9. The data to be sent to the actuator are calculated using controllers – regular controller in a healthy mode, and in case of a fault, a controller for the specific fault.
10. The health of the actuator is evaluated before feeding the control signal.
11. Depending upon the fault type identified for the actuator, a corresponding action for correcting or tolerating the fault is performed.
12. The data is then sent to the actuator at the remote client site.

This algorithm has been implemented at the testbed including two types of faults. A *dead* transducer fault is a case where the transducer data is erroneously stuck at a constant value. A *noisy* transducer is a case when a Gaussian noise power of 2dB has been added to the transducer data.

VII. Implementation Testbed

A block diagram of the testbed built to demonstrate the integration of the IEEE 1451 standard and ISHM capability is shown in Figure 5. The plant used for the testbed implementation was a tank with a level controller from Quanser, Inc. The goal was to control the level of liquid within the tank while considering the health of the transducers used to measure the level of the liquid.

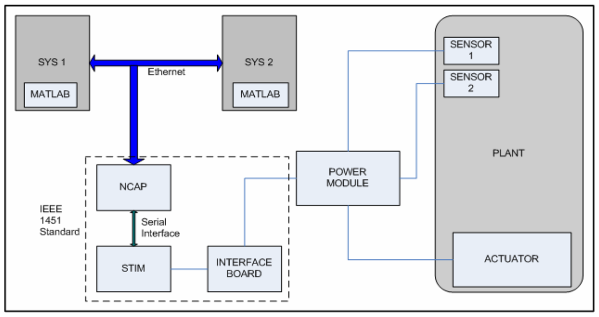

Figure 5. Block diagram of the implemented testbed.

The plant is placed at a remote location. The sensors on the plant provide the local measurement by transferring the primary sensed data to the STIM microcontroller. A null modem was used to connect the NCAP and the STIM. The NCAP communicates with the remote PC via Ethernet and acts as a terminal with its own IP address. The remote PC runs MATLAB software. The algorithm described in Section VI was adopted for the testbed implementation. The algorithm ran on the host PC and controlled the remote plant.

The STIM interfaces with the Quanser pressure sensors and pump. Proper scale factors of the data were stored in the TEDS. The parameters corresponding to the transducer channel were stored in the TEDS. The HEDS packet structure was stored on the host computer. The actuator input was transmitted via Ethernet to the NCAP. Figure 6 shows the actual photograph of the testbed setup developed at Louisiana Tech University.

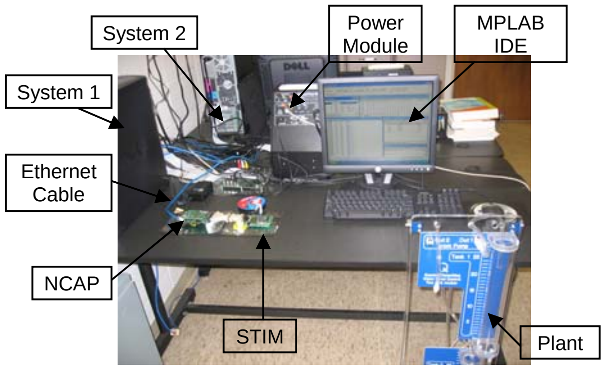

Figure 6. Testbed for intelligent actuator control with a transducer health monitoring capability.

VIII. RESULTS

The testbed was subjected to five different transducer health scenarios to illustrate the significance of the HEDS. Healthy, dead and noisy transducers were considered both for sensors and actuators.

The sampling interval is considered to be approximately 1 second. Since Ethernet does not guarantee a uniform time period for message delivery, the sampling period between each access made to the transducers using Ethernet has to be approximated.

Sensors and actuator are healthy:

When the sensor is in good health, the controller calculates data to be sent to the actuator using the default PID controller. The algorithm then proceeds to perform a health check of the actuator. As the actuator also is identified with no fault, it sends the calculated data to the actuator. The fault type in the flag condition in this case is reset to '00' for both transducers.

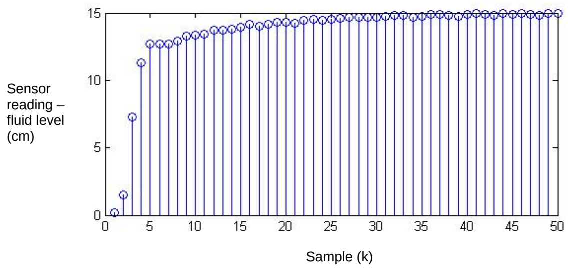

Figure 7. Sensor reading under healthy scenario.

Figure 7 shows the sensor reading of the Quanser Tank module during the experiment. The set point for the liquid level in the tank for this experiment was 15 cm. The results show that the liquid was maintained at the desired level using the default controller and appropriate transducer HEDS. The corresponding actuator voltage during this closed loop control is shown in Figure 8.

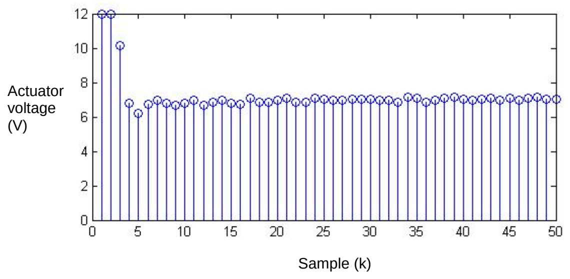

Figure 8. Actuator voltage under healthy scenario.

Sensor is dead and actuator is healthy:

Sensors have finite accuracy and even in steady-state conditions their readings will have some small variations over time. When the sensor generates the exactly same data more than a predefined number of times, identified by the *CONSTANT* field in the HEDS, it is identified as dead. The algorithm chose an open loop controller as a safe mode of operation. Before writing the value to the actuator, an actuator health check was performed. As long as the actuator remains in good health the open loop controller is utilized until the sensor fault no longer exists. Figure 9 and Figure 10 illustrate the results in this case.

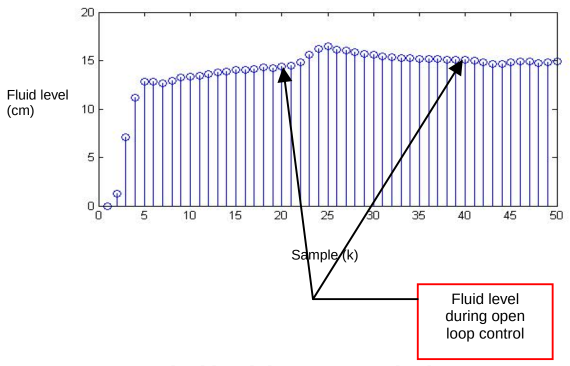

Figure 9. Fluid level during sensor dead scenario.

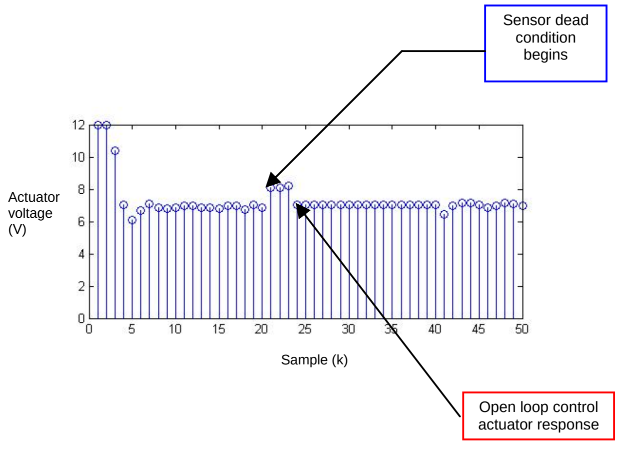

Figure 10. Actuator voltage during when the sensor is dead.

<u>Sensor with excessive noise and actuator is healthy:</u>

A health check performed on the sensor data identified excessive noise in the sensor readings by comparing the standard deviation calculated on the previous *w* samples (including the present sample that contains the noisy sensor reading), with the allowable standard deviation stored in the *STANDARD DEVIATION* field of the HEDS. The noisy data from the sensor are then filtered. Since the actuator is healthy, the data can be written. The fault type in the flag is reset to '00'. Figure 11 and Figure 12 show the sensor reading with noise as well as the filtered data to indicate the filtering action. For this experiment, the noise is added between the 20th and the 30th samples.

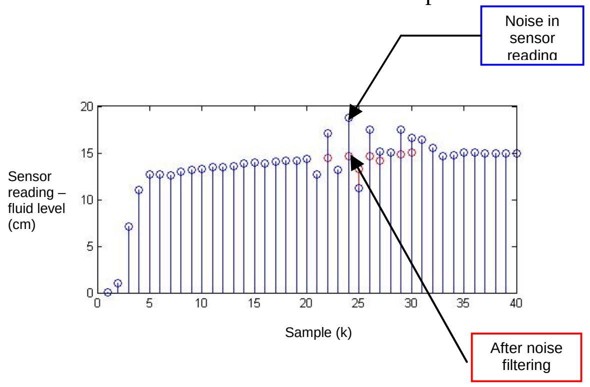

Figure 11. Sensor reading during presence of noise.

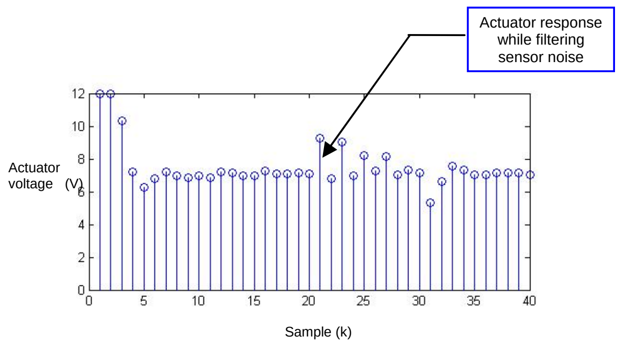

Figure 12. Actuator voltage for the noisy sensor scenario.

<u>Sensor is healthy and actuator is dead:</u>

The sensor is identified as healthy and the actuator is dead. The actuator is shut down to prevent damaging the system and an alarm goes off to notify the operator.

Figure 13 shows the sensor readings during the experiment. The figure indicates that the actuator is detected dead at iteration 25 and hence it is shut down, causing the tank to be emptied.

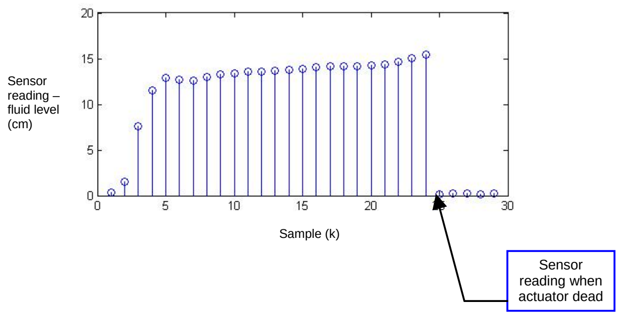

Figure 13. Sensor reading for the actuator dead scenario.

<u>Sensor is healthy and actuator with excessive noise:</u>

The sensor is identified as healthy and the actuator is noisy. The noise in the data is identified by comparing the standard deviation calculated on the averages of the previous *w* samples with the allowable standard deviation stored the HEDS. Figure 14 and Figure 15 show the sensor readings while the actuator noise is filtered. The liquid level in the tank is fairly well maintained, even under the presence of actuator noise.

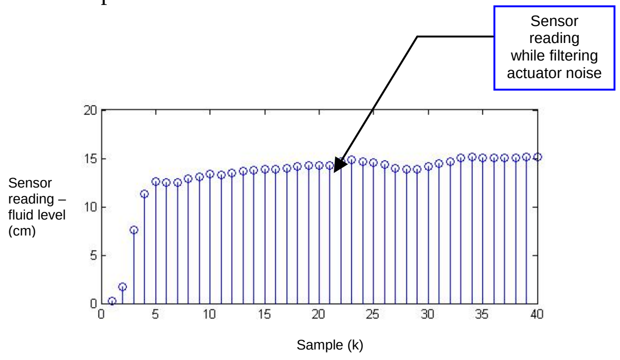

Figure 14. Sensor reading for the actuator noise scenario.

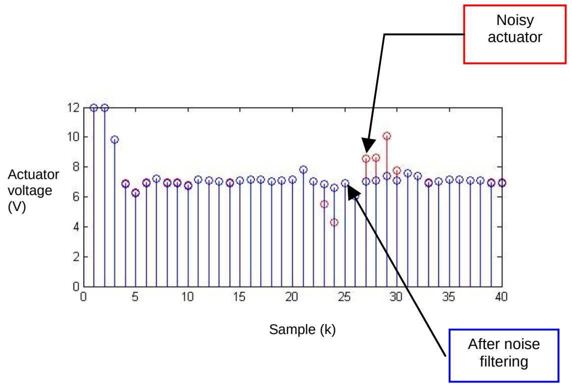

Figure 15. Actuator voltage under presence of noise.

IX. Conclusion

A control concept involving a combination of the IEEE 1451 protocol and Integrated Systems Health Management HEDS has been presented. The concept was realized by developing a control algorithm and was demonstrated through a real time implementation. It is shown how the new concept can be used in combination with standard controllers to utilize health information and thus improve the control and performance of complex electromechanical systems. Furthermore, this work emphasizes the use of standards and an object oriented architecture for data, information, and knowledge, which is suitable for modular, plug & play, and evolutionary control of complex systems.

X. Acknowledgments

The support of LaSPACE and NASA through grant 32-0967-58208 is gratefully acknowledged. The authors also acknowledge the support by Robert Bruce and John Bailey from NASA Stennis Space Center in the context of developing ISHM capabilities.

XI. References

[1] G. Artaud, P. Plancke, G. Furano, R. Magness and C. Plummer, "IEEE 1451: Transducer Networking," *DASIA 2004 Abstract, European Space Agency*.

[2] R. G. Andrei, "Smart silicon sensors/actuators," *Proceedings of International Semiconductor Conference*, pp. 619 – 622, 1995.

[3] R. Bannatyne, "Microcontrollers for the automobile," www.mcjournal.com/articles/arc105/arc105.htm.

[4] F. Bennett, D. Clarke, J. B. Evans, A. Hopper, A. Jones, and D. Leask, "Piconet: embedded mobile networking," *IEEE Personal Communications Magazine*, vol. 4, no. 5, pp. 8-15, Oct. 1997.

[5] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Proc. 7th ACM International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[6] M. A. Demetrious and M. Polycarpou, "On-line fault detection, diagnosis, isolation and accommodation of dynamical systems with actuator failures," in *Adaptive Control of Nonsmooth Dynamic Systems*, Editors G. Tao and F. L. Lewis, Springer-Verlag London, 2001.

[7] F. Figueroa and J. Schmalzel, "Rocket Testing and Integrated System Health Management", Chapter in the book *Condition Monitoring and Control for Intelligent Manufacturing* (Eds. L. Wang and R. Gao), pp. 373-392, Springer Series in Advanced Manufacturing, Springer Verlag, UK, 2006.

[8] F. Figueroa, J. Morris, D. Nicles, J. Schmalzel, D. Rauth, A. Mahajan, L. Utterback, and C. Oesch, "Intelligent sensors and components for on-board ISHM," *Proc. 42nd Joint Propulsion Conference and Exhibit*, California, July 2006.

[9] R. N. Johnson, "Building plug-and-play networked smart transducers," *National Institute of Standards and Technology IEEE 1451 Website*.

[10] K. Lee, "Brief description of the family of IEEE 1451 standards," *National Institute of Standards and Technology*, IEEE 1451 Website (cited April 2008): http://ieee1451.nist.gov/1451%20Family.htm

[11] T. W. Pirinen, J. Yli-Hietanen, P. Pertil and A. Visa, "Detection and compensation of sensor malfunction in time delay based direction of arrival estimation," Proc. *IEEE 2004 International Symposium on Circuits and Systems*, 2004.

[12] R. Selmic, T. Parisini and M. Polycarpou, "Actuator failure dynamics and detectability conditions in nonlinear systems," *American Control Conference*, New York, NY, 2006.

[13] F. A. Zohra and R. Selmic, "Fault aware wireless sensor networks," *IEEE International Conference on Networking, Sensing and Control*, London, UK, 2007.