Physical and Virtual Intelligent Sensors for Integrated Health Management Systems

Ajay Mahajan[1*], Christopher Oesch[2], Haricharan Padmanaban[2], Lucas Utterback[2], Sanjeevi Chitikeshi[2], Fernando Figueroa[3]

[1] College of Engineering, The University of Akron, Akron, Ohio

[2] Department of Mechanical Engineering and Energy Processes, Southern Illinois University Carbondale, Carbondale, Illinois

[3] NASA, John C Stennis Space Center, Technology Development and Transfer Stennis Space Center, Mississippi

[*] Corresponding author: majay@uakron.edu

Abstract - This paper describes the development of intelligent sensors as part of an integrated systems approach. The integrated systems approach treats the sensor as a complete system with its own sensing hardware (the traditional sensor), A/D converter, processing and storage capabilities, software drivers, self-assessment algorithms and communication protocols. The immediate application is the monitoring of rocket test stands, but the technology should be generally applicable to the Integrated Systems Health Monitoring (ISHM) vision. This paper outlines progress made in the development of intelligent sensors by describing the work done to date on Physical Intelligent Sensors (PIS) and Virtual Intelligent Sensors (VIS). The PIS as discussed here consists of a thermocouple used to read temperature in an analog form which is then converted into digital values. A microprocessor collects the sensor readings and runs numerous embedded event detection routines on the digital data. If any event, i.e. spike, drift, noise, is detected, it is reported, stored and sent to a remote system through an Ethernet connection. Hence the output of the PIS is data coupled with a confidence factor in the reliability of the data. The VIS discussed here is a virtual implantation of the PIS in C++. The VIS is designed to mirror the operations of the PIS; however, the VIS works on a computer at which digital data is provided as the input and is thus portable to any sensor system. This work lays the foundation for the next generation of smart devices that have embedded intelligence for distributed decision making capabilities.

IDEX TERMS**:** Intelligent sensors, ISHM, anomaly detection, data correction.

1. INTRODUCTION

Integrated System Health Monitoring (ISHM) is well on its way to being a way of life for many current and future civil, mechanical and aerospace structures/systems. Current interest in smart sensor networks, development in technologies such as microelectronics, nanotechnology, communication networks and distributed computing, have all contributed to the development of ISHM systems. Hence the need for intelligent sensors as a critical component for ISHM is well recognized by now. An ISHM system is a good application of an intelligent sensing system. The purpose of such a system is to detect and measure certain quantities, and to use the information and knowledge obtained from the measured data, and any prior knowledge, to make intelligent, forward-looking decisions and initiate actions. Even the definition of what constitutes an intelligent sensor (or smart sensor) stems from a fundamental desire to get the best quality measurement data that forms the basis of any complex health monitoring and/or management system. If the sensors, i.e. the elements closest to the measurand, are unreliable then the whole system works with a tremendous handicap. Hence, there is a real need to distribute intelligence to the sensor level, and give it the ability to assess its own health thereby improving the confidence in the quality of the measured data.

Sensing is a significant component of complex and sophisticated systems of today's technology. General theories to treat intelligent sensor systems have been reported in the literature since the mid 1980's [1-3]. Work in industry concentrated on sensors with built in expert systems and look-up tables [4-5]. These sensors, called smart sensors, were described as simple sensing devices with built in intelligence. This intelligence included decision making capabilities, data processing, conflict resolution, communications or distribution of information. The autonomous sensor was defined as a sensor that has an expert system with extensive qualitative tools that allowed it to evolve with time into a better and more efficient system [6]. It differed from the previous models by having a dynamic knowledge base as well as embedded qualitative and analytical functions that gave it a higher degree of operation independence, self-sufficiency and robustness. The underlying philosophy behind the autonomous sensor was closest to Henderson's [7-8] logical sensor models that gave more problem-solving capabilities to the sensor, but excluded any type of dynamic models.

DeCoste [9] described a system, called DATMI, which dynamically maintained a concise representation of the space of local and global interpretations across time that were consistent with the observations. Each of the observations were obtained from a sensor, therefore the number of observation were equal to the number of sensors in the system. Truth and validity of the observations were obtained by cross-referencing with possible and impossible states of the system. DATMI was designed for a complete control system comprising of multiple sensors and actuators, and was the basis for the formalized theory called DATA-SIMLAMT (*Dynamic Across Time Autonomous – Sensing, Interpretation, Model Learning and Maintenance Theory*) which was designed for and is still applicable to each sensor in a complex control system [10]. Other attempts to provide intelligence through evolutionary approaches have been reported by Hoshikawa, *et al* [11]. Redundancy leading to enhanced quality of event sensing has often been reported in the literature [12] and often remains a popular method when one can use multiple sensors or networks of sensors.

The main challenge lies in the development of a standard format of intelligent sensors such that they can provide the measurement as well as the measurement quality for all types of sensors. Significant work has been done by the SEVA Research Group at Oxford in such standardization efforts [13]. Regular updates have shown considerable progress in the development of multiple sensor validity parameters that are independent of the type of sensor and set the stage for future standardization efforts [14-15]. Schmalzel *et al* [16] describe an architecture for intelligent systems based on the smart sensor communication standards such as IEEE 1451.X [17] and lays the foundation for intelligent sensors that are truly standardized, and can contribute significantly to the advanced intelligent health monitoring systems of the future.

The key requirements of an advanced health monitoring system are that it should be able to detect damaging events, characterize the nature, extent and seriousness of the damage, and respond intelligently on whatever timescale is required, either to mitigate the effects of the damage or to effect its repair. These requirements have been discussed in some detail in earlier reports by Abbott [18-19]. According to Price, *et al* [20] a pure monitoring system is expected only to report damage rather than to formulate a response, but it is preferable that the ultimate objective of responding to damage be borne in mind from the outset. The statement of key

requirements serves to sub-divide the problem as follows: detection of damaging events, characterization of the damage, prioritization of the seriousness of the damage, identification of the cause of the damage, formulation of the response and execution of the response. There is certainly a large demand for ISHM [21] with application in various fields like monitoring structural health, nuclear power, ship harbors, furnaces, turbine engines, thermal plants, etc.

This paper describes the work done on the development of a Physical Intelligent Sensor (PIS) and Virtual Intelligent Sensor (VIS). The goal of the PIS and VIS is to provide on-line autonomous event detection in real-time for the rocket engine test stand at Stennis Space Center. The PIS and VIS are being developed in order to provide an advantage of early detection over traditional methods which often show event anomalies after a problem or disaster has occurred.

2. PHILOSOPHY

The governing Integrated System Health Monitoring (ISHM) vision for an entire process needs to be in an environment conducive for embedded intelligence and decision making. Such an overall system for the rocket test stand has been designed at NASA Stennis Space Center, using the G2 environment from Gensym, Inc.[1] G2 software offers the opportunity to develop layered behaviors analogous to hierarchical autonomous architecture. This software platform allows for the development of complex systems with sensors, actuators and control systems, all capable of having knowledge bases. This is certainly a powerful tool that allows one to embed intelligence in some or all components of a complex system. The work presented in this paper is focused solely on the single sensor level. The central system collects the data from the sensors and external programs, and then applies it to the model for the system contained in its knowledgebase.

The intelligent sensor model described in this paper is foreseen to be a major component in the ISHM vision. An intelligent sensor is anticipated to provide additional information than that of a traditional sensor. The information provided by an intelligent sensor can include actual data,

[1] [Online]. Gensym, Inc. Burlington, MA. Available: www.gensym.com

corrected data, validity of the data, health of the sensor, etc. Figure 1 shows a possible embodiment of an intelligent sensor.

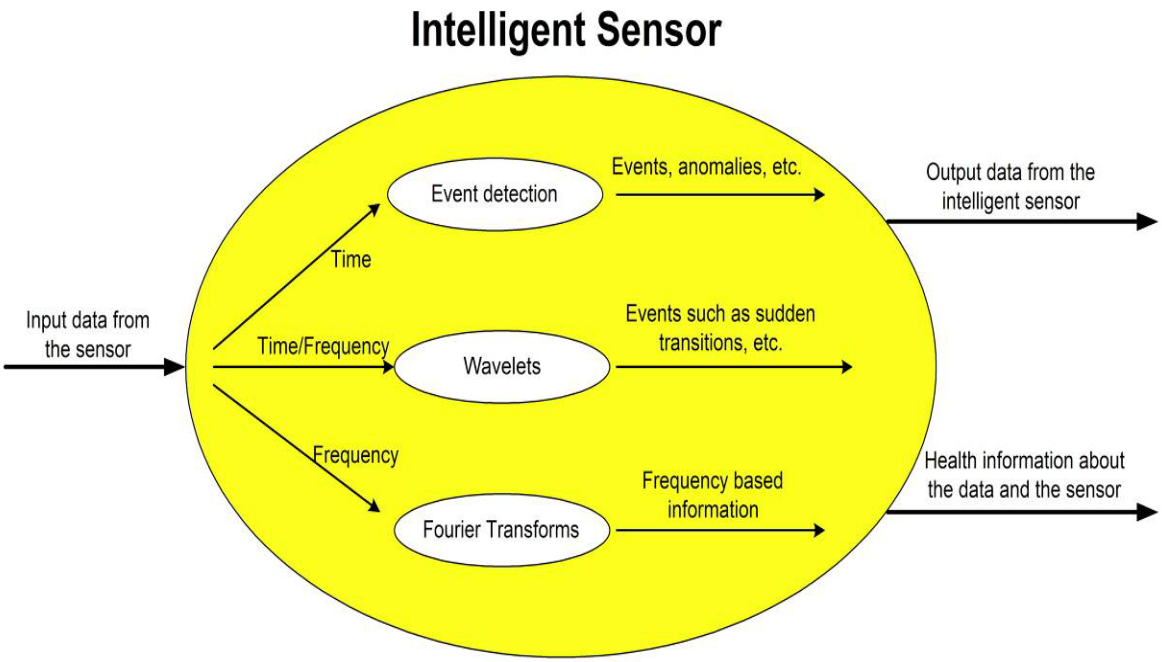

Figure 1: An Intelligent Sensor Embodiment

The intelligent sensor embodiment shown in Figure 1 provides a possible scenario where the input data from a physical sensor is analyzed by various knowledge-based routines. The outputs of the routines provide information which forms a basis for a structured output from the intelligent sensor. The output from the sensor could consist of the raw data, actual or corrected, and health information about the data and the sensor itself. This health information could be in the form of a Condition Assessment Sheet (CAS) which shows a confidence factor level of the data.

The intelligent sensor is developed in two forms: Physical Intelligent Sensor (PIS) and Virtual Intelligent Sensor (VIS). A PIS is an actual sensor with an embedded microprocessor, while a VIS is a software based sensor that functions as a PIS where it is impossible or uneconomical to implement a PIS. The outputs provided to G2 by the PIS and VIS are identical. The PIS and VIS structure for implementation is shown in Figure 2.

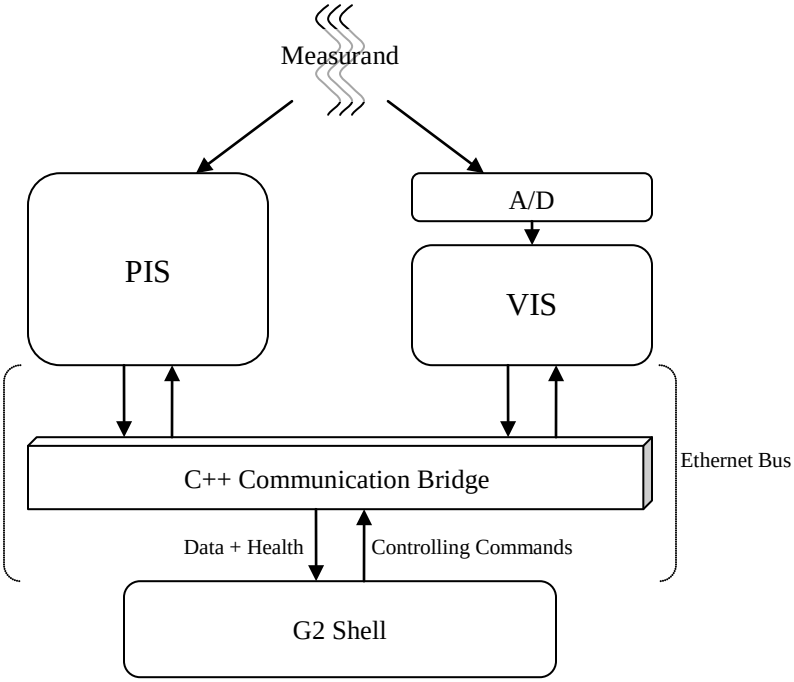

Figure 2: Physical and Virtual Intelligent Sensors

It is seen in the figure that the PIS functions as a complete sensor providing information to the G2 shell. The VIS requires the use of analog to digital converters in order to provide digital data signals. Since the VIS is built for digital input, the technology can provide a powerful testing base for future applications of an intelligent sensor.

The architecture and experimental implementation is described in the following sections.

3. HARDWARE

The PIS (smart sensor) is a combination of a sensing element, a data acquisition chip, a microprocessor and an Ethernet connection. The hardware configuration of the PIS consists of a type K thermocouple, ADC7794[2] analog to digital converter (ADC), a Rabbit RCM3300[3]

[2] [Online]. Analog Devices. Available: www.rabbitsemiconductor.com/products/rcm3300/docs.shtml
[3] Online]. Rabbit Semiconductor. Available: www.rabbitsemiconductor.com/products/rcm3300/docs.shtml

microprocessor core, and a PC operating with Windows XP and Dynamic C 9.0. Figure 3 illustrate the interaction between the components in the PIS.

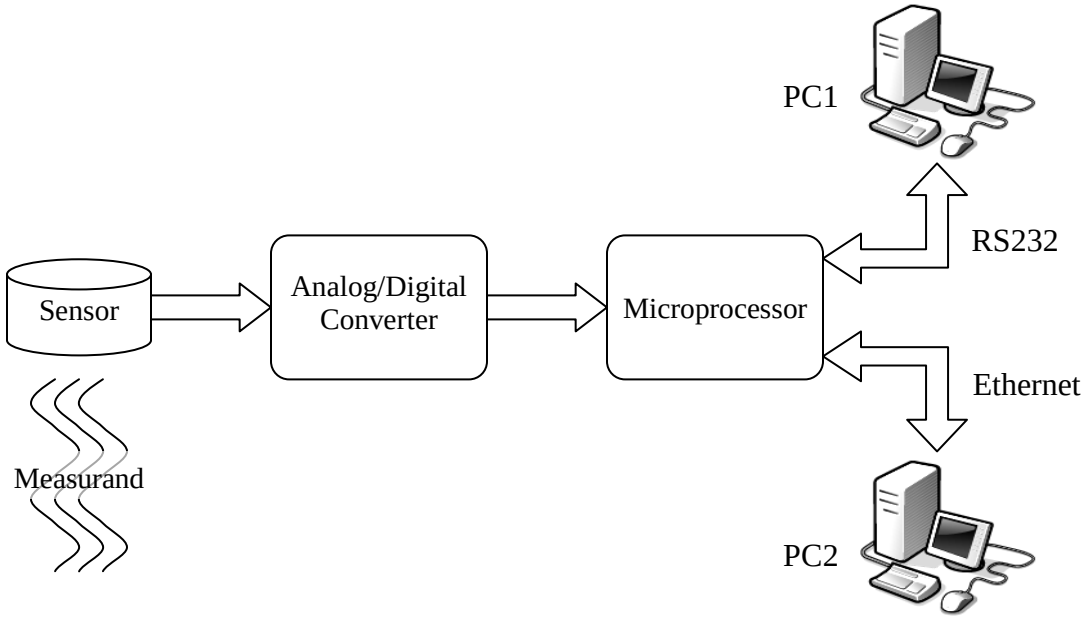

Figure 3: Block Diagram of the PIS

As shown in the diagram, the sensor is connected to the ADC where the analog data from the sensor is conditioned for the microprocessor by converting it to digital data. The ADC is directly connected to the microprocessor which is connected to PC1 via a serial (RS232) programming cable. The microprocessor additionally is connected to the remote PC, PC2, by the Ethernet bus. The Ethernet connection allows one to directly connect the sensor to an Ethernet Bus where a single remote PC can control and collect the information from several smart sensors.

The microprocessor and the PC1 are connected by a RS232 programming cable. The code that is to be stored and ran on the microprocessor is first written in Dynamic C on PC1. The PC1 will first compile the file, creating a flash file with a ".BIN" extension. The flash file is then downloaded to the microprocessor's onboard RAM. Once the flash file is transferred completely, PC1 can be removed. The microprocessor uses SDRAM and FLASH memory to execute the instructions embedded in the flash file. There is a memory backup of 8MB in the form of serial flash on the RCM3300 core module to create files and to store data. This then is transferred to the remote PC2 for further analysis using TCP/IP protocols though the Ethernet

media connected to the microprocessor and the remote PC2. Figure 4 shows the current experimental laboratory set-up of the PIS.

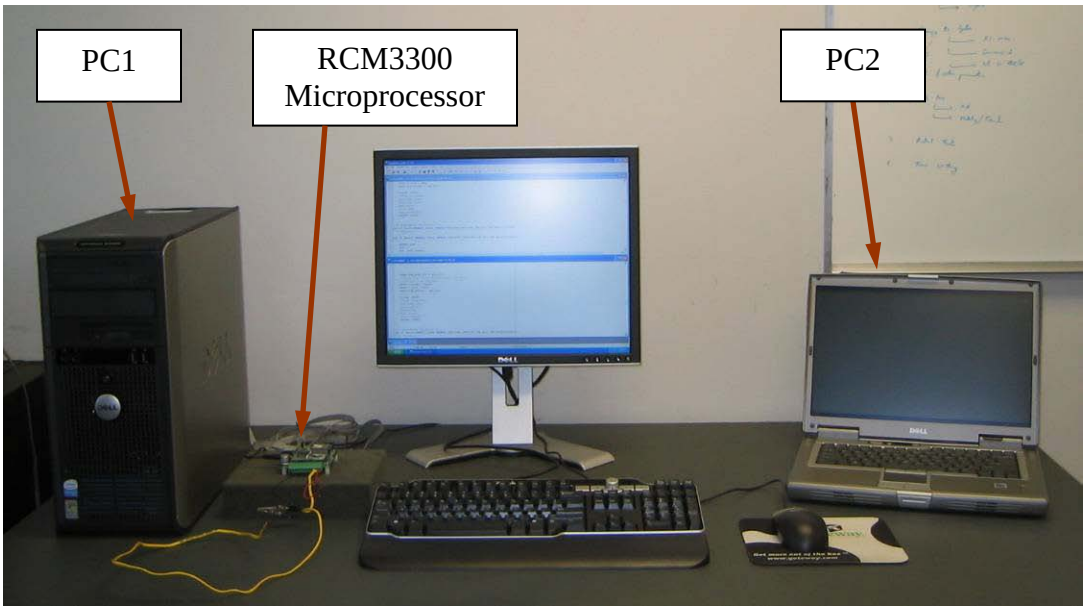

Figure 4: Laboratory PIS Implementation

4. SOFTWARE

The feature extraction algorithms developed by NASA identify prominent events of a sensor signal trace [22]. The routines are written in the 'C' programming language and have been incorporated into a real-time graphical user interface. To function efficiently and to minimize demands on analysis engineers, the algorithms are automatically executed as data are acquired. If an event is detected, the analysis engineer is notified. All events are logged and permanently recorded on disk. The algorithms use statistical information obtained from the time-dependent data stream, along with user defined tolerances and thresholds, to detect features in the data.

The following is a list of routines that have been developed: Drifts, Level Shifts, Spikes, Noise, Iced Sensors, Peaks, Erratic Behavior, Noisy Behavior, Limit Exceedance and Signal Start Bias. Six routines have been implemented on the PIS and include level shifts, flat regions, noise, spikes, drifts, and peaks. Detailed descriptions of each routine can be found in Malloy *et al* [22].

These algorithms were supplied to the authors of this paper to be implemented on the Rabbit microprocessor based PIS that was developed by researchers at Rowan University [16]. Each routine needed to be converted from 'C' programming language to Dynamic C in order to be implemented on the PIS. The process for converting the routines into Dynamic C includes the following:

1. The programmer analyzes the functions, variables and supplied testing data set for each routine.
2. Microsoft Visual C++ is utilized to test a simulated PIS microprocessor configuration in order to analyze memory usage of the data set, output files, and processing files.
3. Functions contained in the supplied routine are converted to library files.
4. The routine's main program is coded into Dynamic C and calls necessary library files generated in step 3, library files that are required for PIS operation, and the artificial data set that was used in step 2.
5. The main program, along with the included library files, are then compiled into a single flash file (".BIN" file extension) and is flashed to the microprocessor, via RS232 serial connection. The microprocessor executes the flash file and generates an output. The output can be displayed in Dynamic C or on a text file sent to the remote PC via the Ethernet link.

The VIS has been developed in order to provide an alternative to the PIS when implementing a PIS is uneconomical or impractical. The VIS not only can work as a smart sensor, but also provides a powerful platform for testing future applications of the PIS on any measurand. The VIS is developed in C++ programming language, which is also the base programming language for G2. This commonality provides superior communication and interaction between the VIS and G2.

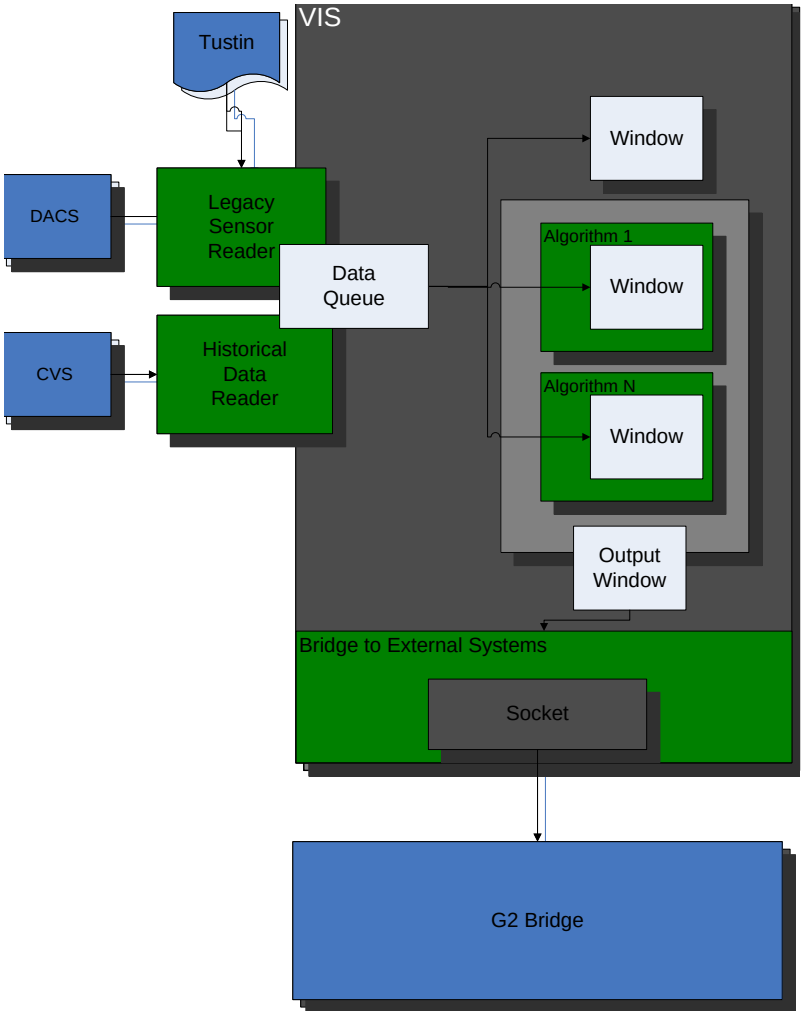

Figure 5: VIS Block Layout

The VIS consists of a number of modules to handle each separate function needed by the VIS. A modular platform is essential for an adaptive VIS. By defining only how the modules must interact, the internal code could be changed easily in order to fit any sensor system. In Fig. 5 regions in blue represent external systems, blocks in green present separate modules, blocks in white represent linked-list classes, and regions in gray represent the algorithm test bed. The modules can be modified to allow the VIS to work in any sensor system as well as using any algorithm. The legacy sensor reader allows for the data to be received from the "dumb" sensor, convert that data into engineering units if needed and send the final data to the VIS for further processing. The historical data reader provides a similar service for files containing recorded data. In this case legacy sensor reader is handling a data stream from the data acquisition and

control system (DACS) monitoring data coming from a rocket test stand. The DACS computer dumps old data with the engineering units to a ".CSV" file at which historical data reader reads from that file. Each module is run in its own thread which is a child of the VIS. The specialized linked-list classes are designed to perform operations such as historical data recording and windowing over the incoming data. After the data input and output modules have been set to work in the sensor system, algorithms can be modified to work using the modified linked list as inputs as well as outputs. The data is then processed using these algorithms and, if desired, sent to another system for post processing or display. In this case the both the raw data and events are forwarded to the G2 knowledge. The data is then archived and interpreted by the G2 knowledgebase which is shown in the results.

6. RESULTS

To demonstrate the PIS capabilities an artificial data set was created with the following anomalies: Level Shift, Flat Region, Noise, Spikes, Drift, and Peak. The data set and outputs from the PIS are shown in Figure 5. The data set has no particular units in either amplitude or time. Subsections A-F describe the routines recognizing the events/anomalies.

A: Level Shifts - Level shifts are detected by fitting linear equations to windows of data and monitoring the fit parameters for excursion. User-definable parameters are used to constrain the routine to the detection of level shifts that are important to the user. The PIS output where the level shift has occurred is shown to have a start time at 20.0 (time units) and end time of 50.0 (time units). The shift started with a stable value of 95.0 (data units) and ended with a stable value of 2.0 (data units).

B: Flat Regions - In order to detect a flat signal, a curve fit is performed. If the resulting slope falls within predefined limits, the difference between each data point and the fitted line is computed. If the user-defined majority of difference values are less than four times the nominal standard deviation of the signal, a flat signal is declared. The PIS shows an output where during the interval between 62.0 and 126.0 (time units) the sensor output was flat. A flat event indicates the possibility of a dead sensor or a sensor with no power supplied. It can be seen from the

graph that there exists no change in the sensor data in the observed period. Since there is absolutely no noise in the observation, the suspicious nature of the sensor is noted as an event.

C: Noise - The definition of noise in this application is a larger-than-expected variance in the signal for a prescribed length of time. This routine makes a distinction between noisy and excessively noisy parameters based on the expected variance of the signal. The PIS output where there exists excessive noise is detected between the range of 151.0 and 177.0 (time units). During the period the maximum value of the noise is 14.4 (data units) and a minimum value of -2.4 (data units). The routine returns a status of 1 or 2, where 1 is excessive noise and 2 is fine noise. This test shows excessive noise.

D: Spikes - The spike routine uses curve fits of the data to look for positive or negative data excursions which occur within three data samples. The standard deviation of the curve fit error is used to identify possible spikes. Two spikes were detected and are displayed from the output of the PIS. The first spike was detected during the time period between 224.0 to 226.0 (time units). The magnitude of the first spike was 48.0 (data units). Similarly the second spike was detected between 255.0 and 257.0 (time units) with a magnitude of -52.0 (data units).

E: Drift - Drifts are general upward or downward trends in the data. In order to detect a drift, the data is first broken into periods of constant linear behavior. The slope of each period is monitored, and a drift is flagged when one or more consecutive periods exhibit a slope that falls outside predefined bounds. The PIS detected and displayed the output of a drift point in the data. The drift started at 300.0 (time units) and ended at 330.0 (time units). It started with an initial value of 2.0 (data units) and ended at a new final value of 21.0 (data units).

F: Peak - Peaks are detected by monitoring for significantly non-zero values of the slope of successive windows of data. Once a significant slope is detected, it is monitored until it returns to a near-zero state. Peaks which do not meet the minimum duration and height requirements specified by the user are rejected. The PIS output where the data contained a peak event is shown to start at 368.0 (time units) and end at 418.0 (time units). The peak value of the data is 48.0 (data units).

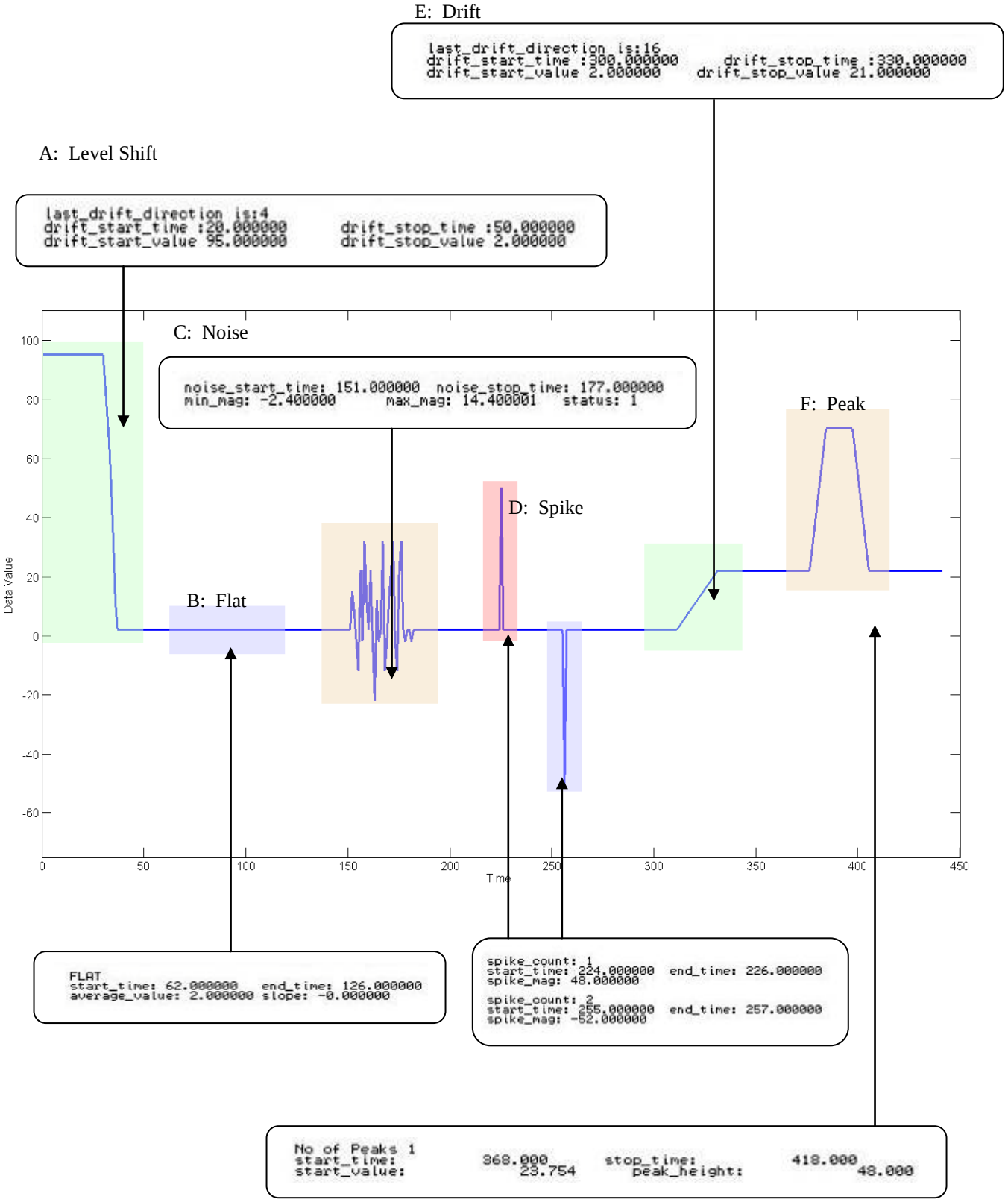

Figure 6: Test Data & Output from the PIS

The VIS is implemented in order to communicate with the G2 environment. The most recent work on the VIS shows the ability to communicate between the VIS and the G2 software via a C++ communication bridge. Additionally, the VIS is capable of detecting the same events as the PIS. Figure 7 shows the current capabilities and structure of the VIS operating in conjunction with G2. It is shown that the VIS is developed into an easy graphical interface for the user.

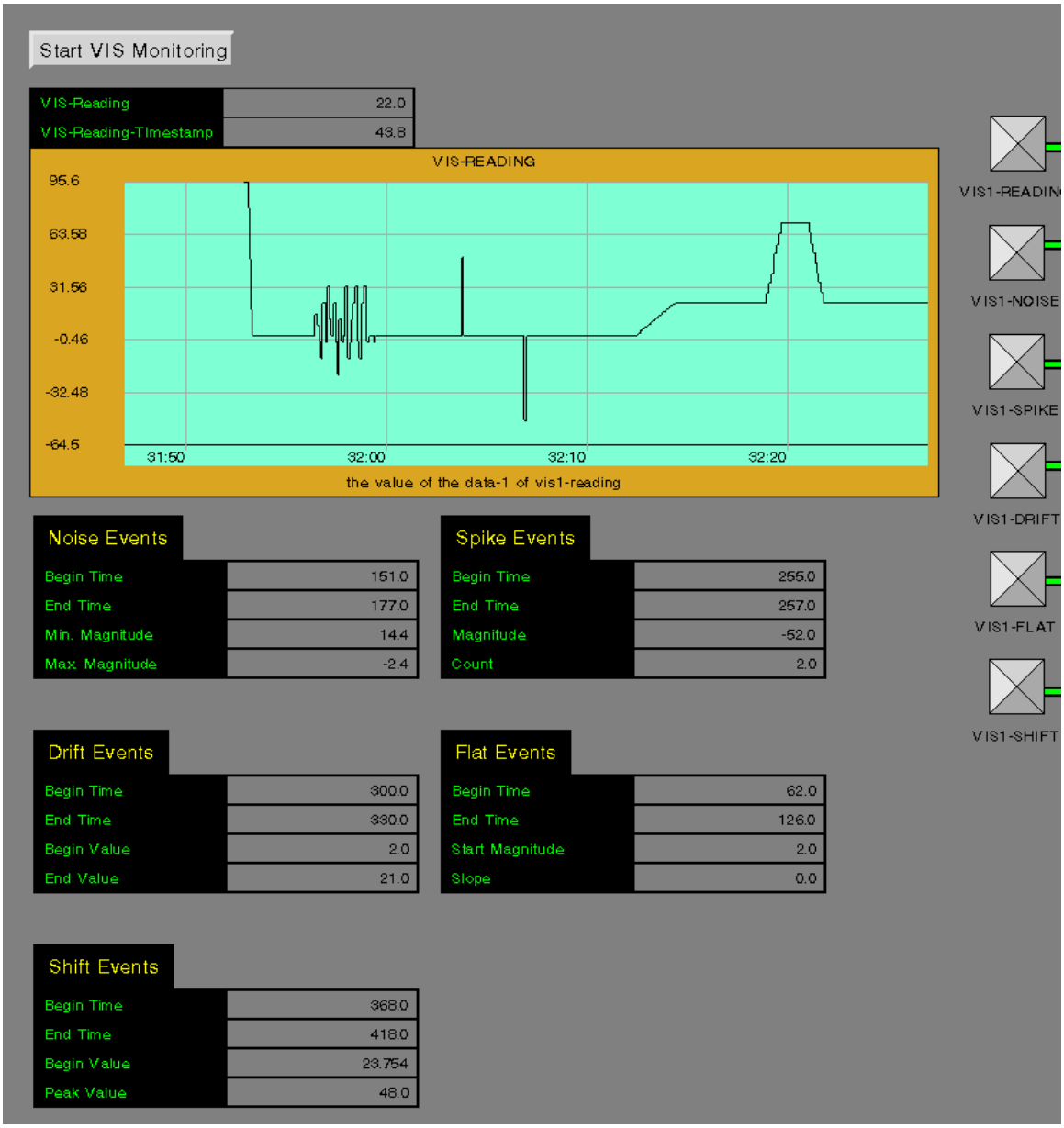

Figure 7: VIS Output

In Figure 7, a graphical representation of the data from the sensor, as well as tables showing the most recent detected anomalies are shown. When an event is detected it is sent from the VIS to G2 where it is archived. G2 also allows for the events to be displayed. The events detected by the VIS are identical to those detected by the PIS.

7. CONCLUSIONS & FUTURE WORK

This paper has presented the latest development of intelligent sensors as part of an integrated systems approach. Under a project undertaken at the NASA Stennis Space Center, an integrated framework is being developed for the intelligent monitoring of smart elements. Two types of sensors, developed to provide the identical information, are shown to present a tool for event detection and intelligence. The Physical Intelligent Sensor (PIS) provides the ability to collect data and supply the collected data along with important interpreted information about the signal. The PIS uses a sensing element, analog to digital converter, and a microprocessor running embedded event detection routines. The Virtual Intelligent Sensor, developed in C++ programming language, provides the same abilities as the PIS, but is implemented on a computer and utilizes digital data sent from a transducer as an input rather than an analog signal from a sensor. This work lays the foundation for the next generation of smart devices that have embedded intelligence for distributed decision making capabilities.

The future work in the research and development of these intelligent sensor models requires refinement of embedded intelligent decision making capabilities. Future work also entails combining all routines to be embedded and run simultaneously to provide a true real-time event detection sensor. Further testing of the PIS and VIS using additional sensors will be done in order to trigger the correct event detection routines, even if they occur at the same time. These results will be compiled to test the full functionality of the sensors. Additionally, other intelligent decision making algorithms may need be explored and compared to the provided routines in order to implement learning into the sensor models.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of NASA for funding this work under Grant NNS04AB796. The authors would also like to acknowledge the contributions of researchers at Rowan University for providing the smart sensor prototype and NASA Glenn Research Center for providing the event detection routines.

REFERENCES

1. Ghani, N., 1988,"Sensor Integration in ESPRIT," *IFAC Proceedings,* Karlsruhe, FDR, pp. 323-328.
2. Pinkava, J., 1989,"Towards a Theory of Sensory Robotics," *Robotica*, **8**, pp. 245-256.
3. Lozano-Perez, T., Mason, M. T., and Taylor, R., 1984, "Automatic Synthesis of Fine Motion Strategies for Robots," *International Journal of Robotics Research*, **3**, No. 1, pp. 2-24.
4. AbdelRahman, M. and Smith M. L., 1991 "The Impact of AI On Sensing Technology," *SENSORS,* pp. 16-22.
5. Studt, T., 1994,"Smart Sensors Widen Views on Measuring Data," *R&D Magazine*, pp. 18-20.
6. Figueroa, F. and Mahajan, A., 1994, "Generic Model of an Autonomous Sensor," *Mechatronics*, **4(3)**, pp. 295-315.
7. Henderson, T and Shilcrat, E., 1984, "Logical Sensor Systems," *Journal of Robotic Systems*, **1(2)**, pp. 169-193.
8. Henderson, T., Hansen, C. and Bhanu, B., 1985, "The Specification of Distributed Sensing and Control," *Journal of Robotic Systems*, **2(4)**, pp. 387-396.
9. DeCoste, D., 1991,"Dynamic Across-Time Measurement and Interpretation," *Artificial Intelligence*, **51**, pp. 273-341.
10. Mahajan, A. and Figueroa, F., 1995, “Dynamic Across Time Autonomous - Sensing, Interpretation, Model learning and Maintenance theory (DATA-SIMLAMT),” *Mechatronics,* **5(6)**, pp. 665-693.
11. Hoshikawa, N., Ohka, M. and Yussof, H.B., Dec 2011, “Bottom-Up Approach for Behavior Acquisition of Agents Equipped with Multi-Sensors,” International Journal on Smart sensing and Intelligent Systems, **4(4)**, pp. 583-606.

12. Indu, S., Kesam, V., Chaudhury, S., Bhattacharyya, A., Mar 2011, "Self Organizing Sensor Network to Enhance Event Coverage," International Journal on Smart sensing and Intelligent Systems, **4(1)**, pp. 53-74.
13. Henry, M.P. and Clarke, D.W., 1993, "The Self-Validating Sensor: Rationale, Definitions and Examples," *Control Engineering Practice*, **1(4)**, pp. 585-610.
14. Henry, M., Dec. 1995, "Sensor Validation and Fieldbus," *Computing and Control Engineering Journal*, **6(6)**, pp. 263-269.
15. Henry, M., Oct. 2000, "Plant Asset management via Intelligent Sensors – Digital, Distributed and for Free," *Computing and Control Engineering Journal*, Vol. **11**, pp. 211-213.
16. Schmalzel, J., Figueroa, F., Morris, J., Mandayam, S. and Polikar, R., Aug. 2005, "An Architecture for Intelligent Systems based on Smart Sensors," *IEEE Transactions on Instrumentation and Measurement*, **54(4),** pp. 1612-1616.
17. IEEE Standard for a Smart Transducer Interface for Sensors and Actuators – Network Capable Application processor (NCAP) Information Model [Online]. Available: www.ieee.org.
18. Abbott, D., 2001. "Development and Evaluation of Sensor Concepts for Ageless Aerospace Vehicles", Report 1, CSIRO TIP Report no. TIPP 1516.
19. Abbott, D., 2001, "Development and Evaluation of Sensor Concepts for Ageless Aerospace Vehicles", Report 2, CSIRO TIP Report no. TIPP 1517.
20. Price, D.C., Scott, D.A., Edwards, G.C., Batten, A., Farmer, A.J., Hedley, M., Johnson, M.E., Lewis, C.J., Poulton, G.T., Prokopenko, M., Valencia, P. and Wang, P., 2003. **"**An Integrated Health Monitoring system for an Ageless Aerospace Vehicle" *Proceedings of the Structural Health Monitoring 2003: From Diagnostics & Prognostics to Structural Health Management*, ed. Fu-Kuo Chang, DEStech Publications (Lancaster, PA), pp. 310-8.
21. McDonald, J.R., McArthur, S.D.J. and Burt, G.M., 2001, "Intelligent system applications for power system control and management," *Computing & Control Engineering Journal*, **12(2)**, pp.85-91.
22. Malloy, D.J., Biegl, C., Zakrajsek, J.F., Meyer, C.M. and Fulton, C.E., Sept. 1997, "Development of a Near Real-Time Turbine Engine Testing Diagnostic System using Feature Extraction Algorithms," *Proceedings of the XIII International Symposium on Air Breathing Engines*, ISABE 97-7144, pp.1-10.