System issues in arrays of autonomous intelligent sensors

Robert M. Newman, Elena I. Gaura,

School of Mathematical and Information Sciences
Coventry University
Coventry CV1 5FB, United Kingdom
bobn@coventry.ac.uk

ABSTRACT

This paper considers some of the systems level issues of concerned when building of large scale sensing systems composed of a large number of autonomous intelligent sensors. The work is motivated by the potential offered by monolithic, integrated, intelligent MEMS sensors. They provide very low cost sensing and computational power in a form which may be integrated into 'smart structures', in which structural diagnostic capability is deeply embedded into the structure itself. However, in order to realise them, some intrinsic systems issues must be addressed. Ultimately, the goal of this work is to develop a systems architecture which enables the designer of application systems to deploy intelligent networked sensors as a generic systems component, concentrating on applications issues, rather than the detailed system design problems associated with decentralised, distributed sensor systems.

Keywords: sensor, smart, intelligent, cogent, MEMS, artificial intelligence

1. INTRODUCTION

Developments in intelligent MEMS sensors may be expected to revolutionise the design of all systems based on sensors. Sometimes the claims made for them can be sweeping. Kumar et. al. [1] state:

> Networked microsensors technology is a key technology for the 21st century. Cheap, smart devices with multiple, on board sensors, networked through wireless links and the Internet and deployable in large numbers, provide unprecedented opportunities for instrumenting, and controlling to our advantage, homes, cities, the environment, and battlefields.

While some of these claims may appear at first sight to be overstated, there certainly are some application areas where the effects of the deployment of very large numbers of intelligent MEMS sensors will be profound. One domain where there appears to be very great potential is that of 'smart materials', an application area in which there is particular interest being airframe design. This application will be used as an exemplar throughout the paper, in order to illustrate some of the real systems design issues involved in the deployment of very large arrays of intelligent sensors.

Smart materials in airframes are being proposed to address a number of design goals for future advanced aircraft, two major goals being adaptive aerodynamics and self diagnosis, and possibly repair of the structural integrity of an airframe. The latter is aimed at extending the life of airframes, possibly ultimately leading to the 'eternal aircraft' as proposed in the NASA CTIP [2] project. Both applications depend on the location of sensors throughout the structure of the airframe, with possible need for actuators as well, dependent on the precise nature of the application.

Structural diagnosis systems are derived originally from structural health monitoring systems. If the sensors are not smart, the installation of such systems is also problematic. Each sensor needs to be individually wired to the data processing unit, and the connections must usually be of very high quality to cope with low level analogue signals. The complexity of this wiring, and its sensitivity often means that such systems cannot be permanently fitted.

By providing some redundancy in provision of sensors, reliability may be enhanced at a systems level, even if the reliability of an individual sensor is not better (although a monolithic intelligent sensor might be expected to be more reliable than a multiple component one).

The systems design implications of intelligent MEMS based systems, using redundant provision of sensors, and distributed data processing within the intelligent MEMS sensor, are very different from those of conventional systems. These design problems are also very subtle and complex, particularly for those not familiar with the field. In the more general domain of information systems, where the default system is now a distributed one, system designers have developed generic distributed operating systems which free the applications designer from the need to consider each and every consequence of distribution. Our eventual goal is to develop such an 'operating system' for distributed sensor systems, but the and components of that system will be different from those for more general systems. Several authors have surveyed these system level requirements [3,4], and have considered the importance of the issues outlined above. The major considerations that must be taken into account, and will contribute to the specification for 'operating systems' for intelligent networked sensors are briefly presented below.

2. SENSOR CALIBRATION AND COMPENSATION

Existing systems are dependent on a largely manual process of sensor identification, calibration and compensation. There is a considerable manufacturing spread over the sensors used. Individual sensors need to be calibrated, but if only sensors which calibrate within a small range can be used, the cost of the system rises considerably, thus it is often necessary to provide compensation for an individual sensor's characteristics. The characterisation and calibration of individual sensors is undertaken by providing a controlled stimulus (in an accelerometer, via a calibrated shaker) and measuring the output of the sensor under test. Compensation requires the same process under a range of stimuli and the production of correction tables, which need to be provided individually to the data processing system with which the sensor is to be used. Gaura et. al. [5,6] have described the use of artificial neural networks for the identification and compensation of some commonly found sources of error in micro-machined accelerometers. Sachenko et al. describe the use of an artificial neural network to predict the development of a particular type of fault, sensor parameter drift [7]. Using historical data from sensors the ANN is trained to predict the drift of a sensor over time. The prediction is used to correct the output from the sensors.

Application designers will certainly not wish to be burdened with the detail of sensor compensation, so this is a facility that will need to be included in the sensor operating system.

3. SENSOR PLACEMENT AND STRUCTURE FABRICATION COSTS

. There is a complex, four way trade-off between number of sensors used, degree of redundancy, cost of installation and locality of sensors to the data points of interest. In a 'traditional' pre-intelligent MEMS system the trade off was very much weighted against using a large number of sensors and they would be strategically placed on the structure to be monitored, at sites known to show symptoms of structural stress. The sites for the sensors have to be predetermined by laboratory stress experiments or computer simulation. Staszewski, et al, discuss the use of AI techniques for determining an optimum sensor placement for acoustic impact sensors distributed on the surface of a composite material [8].

Using intelligent MEMS sensors, the cost and processing constraints on the number of sensors used may be relaxed. Instead of placing a few sensors at strategic points of interest, sensors are distributed all over the structure. Although there may not be a sensor exactly at a point of interest, the data value for that sensor can be synthesised using a process of data fusion. Jungert discusses the issues of distributed data fusion, as would be required in an array of intelligent sensors where the only processing power for the system is that contained in the sensors themselves [9]. Koushanfar, Potknojak and Sangiovanni-Vincentelli describe the use of multimodal data fusion [10]. In their example, they obtain a system of fifteen equations to yield the values of twelve variables, giving a redundancy of three equations. Several methods for data fusion have been reported including Kallman filters and artificial neural networks [11,12].

It is not necessary for sensors to be distributed in an orderly way. If sensors are so inexpensive and small that installation is a relatively high cost comparatively, it may be advantageous to use a very low cost installation method, such as embedding them in the matrix of a composite, and allowing them to be randomly distributed through the matrix as the composite is built up. In this case, the se sensors form an 'ad-hoc' network, and one of the first tasks on initialisation is establishment of a virtual topology for the network, as discussed in section 6.

The design of a sensor operating system will need to be sufficiently flexible to account for different types of sensor array, from orderly, placed arrays to random, ad-hoc networks. The application designer is more likely to be interested in the desired data points, than the actual sensor locations, so the operating system will need to have inbuilt the facilities for data fusion, in order to map the actual network topology to the one required by the application.

4. POWER AND HEAT

Passive or conventional smart sensors consume little or no power, but active sensors, particularly those endowed with a large amount of processing capability, may consume significant amounts of power and dissipate substantial amounts of heat. In many applications (see, for instance, Schweibert et al [13]) power, and the associated heat production may be a limiting factor in the design of a system. The design considerations for intelligent sensor networks are therefore often dominated by power considerations. This can be apparent in the detailed design of the sensor electronics, favouring selection of low power processing methods. It can also affect network design and topology establishment. The sensor operating system will require the ability to configure its own internal workings according to the power and heat constraints of a particular application.

5. DATA TRANSMISSION AND COMMUNICATIONS

The wiring requirements are simplified to provision of a simple passive multi-drop network, or, in many cases, wireless connection. The bulk of the computation occurs within the array itself, and all that a structural diagnostic system has to do is to extract the information from a convenient location within the array. The passive network has advantages in terms of simplicity, reliability and flexibility with respect to adjustment of the connectivity of the network to handle sensor failures or other dynamic reconfiguration needs. On the other hand point to point switched networks can have advantages in terms of power use, and this is always an important consideration in sensor networks. Formal descriptions of re-routing algorithms in such networks have been presented previously [14]. Although at first sight wired sensors would seem to be appropriate to smart structure applications (because wires are needed to distribute power to intelligent sensors) wireless networks would be much simpler to connect, particularly if a matrix-embedded random sensor deployment has been adopted.

The specification of the sensor operating system must deal with all of these types of connectivity, in a way that is transparent to the applications designer.

6. NETWORK INITIALISATION AND TOPOLOGY

In placed, wired networks the initialisation of the network, and its topology is established by the wired connections. However a wireless, ad-hoc network is unusable until the ‘real’ network has been established. There is a great deal of interest in this topic, particularly with reference to ‘pervasive computing’. Most of this work, in the sensor arena, has been to do with ad hoc radio networked sensors for applications such as environmental and battlefield sensing [15,16,17], however it can also be used in structural sensing, for the reasons discussed above. In a wireless network, every sensor within the range of another sensor’s transmitter is connected, but selecting particular favored routings can have a substantial effect on overall transmitter power required. Other factors such as bandwidth and reliability can also be optimised. The literature above reports several topology establishment methods based on AI and other optimisation techniques.

In ad-hoc networks the sensor operating system will be responsible for initialisation and maintenance of the network, and the discovery of the topological an location information on which higher level functions such as data fusion and fault management depend.

7. FAULT MANAGEMENT

In conventional systems sensors are mechanically attached to the structure under test using screws or other removable attachment. Failed sensors can be removed and replaced, and since they have been pre calibrated, or, in the case of a smart sensor, internally compensated, the new sensor fits into the rest of the system just as the old one did. This is not the case for a sensor embedded in the fabric of the structure under test. Instead, the array needs to be provided with some degree of redundancy, and a means for detecting and swapping out failed sensors. Data fusion techniques [18,19] are used to extract the required data from this array and sensor failure determined by deviation from expected values based on models of the specific application. The authors have proposed an application independent fault management strategy based on self-diagnosis of sensors, in which sensors are removed from the array if faulty. False alarms are also possible, and sensors must be re-introduced to the array if found to be healthy. It is possible to calculate the lifetime of an array in this case [20]. If, in an array consisting of n sensors, the mean time between failures of a sensor is t_{mbf}, the false error detection rate is f, there is a proportion r of redundant sensors and the time to correct a false fault is some multiple C of the fault detection time, then the lifetime L_{rff} of the whole array is given by

$$L_{rff} = t_{mbf}(r - fC)$$

This indicates some of the parameters that are important in the selection of a fault management strategy. The fault detection time of any strategy is an important figure of merit, since it directly affects the error rate from the array as a whole. Any strategy must either have a zero false fault detection rate or must include the re-introduction of a sensor into the array should a fault diagnosis subsequently prove to be faulty. The time to re-introduce a sensor must be a small multiple of the fault detection time, preferably unity or less.

An effective, application independent and transparent fault management system is an important component of the sensor operating system.

8. APPLICATION LEVEL FUNCTIONS

We have put forward the concept of a ‘cogent sensor’ [21], one that participates in the process of transformation from data to information. Of course, what is 'information' depends on the requirements of the application, so each application will require some different processing within a cogent sensor. This requirement conflicts with our goal of a generic systems component unless the sensor part of the system can take responsibility for the downloading of applications level data processing to the sensor web and its distribution within it in a way that is optimal for the overall operating constraints, be they power consumption, computational speed or other constraints. There are several ideas from distributed systems technology that can be used to achieve this goal, agent technology being a currently favoured option.

9. CONCLUSIONS

Several functions, namely, sensor calibration and compensation, data fusion, data transmission and communications, network initialisation and topology, fault management and distribution of application level functions are common requirements for very nearly all intelligent sensor networks, and therefore form the basic components for the system support for a generic intelligent sensor system. Presently we are undertaking a programme to address all of them, and ultimately integrate them into an 'operating system' for distributed sensor networks, providing the engineer using these systems with a straight forward and standardised way of controlling and gathering the information from sensor mechanisms. In our work to date, mainly in the area of fault management, artificial intelligence techniques, particularly artificial neural networks, have provided elegant solutions to some of the design problems for our system. Other researchers have found them of use in the topology establishment and data fusion. Along with intelligent agent technologies, we expect them to play a central part in the workings of our system.

REFERENCES

[1] Kumar, S. , Shepherd, D., Zhao, F., Collaborative Signal and Information Processing in Micro-Sensor Networks, *IEEE Signal Processing Magazine,* March 2002.

[2] Miller, W. M. Peterson, K. A. (1998) MEMS reliability: The challenge and the promise, NASA Technical Reports.

[3] Peti, P, Obermaisser, R, Elmenreich, W, Losert, T (2002) An Architecture Supporting Monitoring and Configuration in Real-Time Smart Networks.

[4] Martinez, D, Gruber, M, (2002) Next Generation Technologies to Enable Sensor Networks.

[5] Gaura, E. Rider, R.J. Steele, N. (2000) Closed-loop neural network controlled accelerometer, Proceedings of the I. Mech. E, Part I, Journal of Systems and Control Engineering, vol. 214, no.I2, pp.129-138.

[6] Gaura, E. Rider, R.J. Steele, N. (2000). Developing smart micromachined transducers using feed-forward neural networks: a system identification and control perspective. The IEEE International Joint Conference on Neural Networks, IJCNN'2000, Proceedings, ISBN 0-7695-0619-4, Vol. IV, pp. 353-358, Como, Italy.

[7] A.Sachenko et al. (2002) Metrological Automatic Support in Intelligent Measurement Systems Computer Standards & Interfaces vol. 24, issue 2, pp. 123-131.

[8] Staszewski, W. J., Worden, K., Wardle, R., Tomlinson, G. R. (2000), Fail-safe sensor distributions for impact detection in composite materials. Smart Material Structures, 9 (2000), pp. 298-303. IOP Publishing, London.

[9] Jungert, E (2000) A data fusion concept for a query language for multiple data sources, Proc 3rd International Conference on Information Fusion (Fusion 2000), Paris, Jul 10-13

[10] Koushanfar, F, Potknojak, M, Sangiovanni-Vincentelli, A, (2002) Fault Tolerance Techniques for Wireless Ad-hoc Sensor Networks.

[11] Clarke, J, Yuille, A, (1990) Data Fusion for Sensory Information Processing Systems, Kluwer.

[12] Brookes R, Iyengar, S, (1997) Multi-Sensor Fusion: Fundamentals and Application with Software, Prentice-Hall.

[13] Schweibert, L, Gupta, S, Auner, P, Abrams, G, Iezzi, R, McAllister, P (2002) A Biomedical Smart Sensor for the Visually Impaired.

[14] Gaura, E.I, Newman, R.M., (2003) Intelligent sensing: Neural Network based health diagnosis for sensor arrays, Proc ASME'03 , Kobe, Japan.

[15] Pister, K.S.J., Boser, B.E., (1999)Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes. University of California, Berkeley Electronics Research laboratory Research Summary.

[16] Catterall E, Van Laerhoven, K., and Strohbach, M.. (2002) SelfOrganization in Ad Hoc Sensor Networks: An Empirical Stud. In Proc. Alife VIII: the 8 th International Conference on the Simulation and Synthesis of Living Systems, Sydney, Australia. MIT Press, 2002.

[17] NASA, Advanced Architectures and Automation Branch, Sensor Web Application Prototype (SWAP). NASA GSFC Code 588.

[18] Clark, J.L. , Yuille. A.L.,(1990). Data Fusion for Sensory Information Processing Systems. Kluwer Academic Publishers.

[19] Brooks, R. R. , Iyengar, S. S.(1996). Robust distributed computing and sensing algorithms. IEEE Computer, pages 53--60, June 1996.

[20] Newman, R.M., Gaura, E.I., (2004) A systems level perspective of fault management in large sensor networks. To be published.

[21] Gaura, E. I., Newman, R.M., Smart, Intelligent and Cogent Microsensors – Intelligence for Sensors and Sensors for Intelligence, Proc. MSM'04, Boston, MA, March 2004.