Comparison of Event Detection Methods for Centralized Sensor Networks

Julien Sauvageon
Research Assistant

Alice M. Agogino
Professor

Department of Mechanical Engineering
University of California, Berkeley
{jsauvageon,agogino}@berkeley.edu

Ali Farhang Mehr, Ph.D
Engineer/QSS

Irem Y. Tumer, Ph.D.
Group Lead

Complex Systems Design Group
Intelligent Systems Division
NASA Ames Research Center
{amehr, itumer}@email.arc.nasa.gov

Abstract:

The development of an Integrated Vehicle Health Management (IVHM) for space vehicles has become a great concern. Smart Sensor Networks is one of the promising technologies that are catching a lot of attention. In this paper, we propose to a qualitative comparison of several local event (hot spot) detection algorithms in centralized redundant sensor networks. The algorithms are compared regarding their ability to locate and evaluate the event under noise and sensor failures. The purpose of this study is to check if the ratio performance / computational power of the Mote Fuzzy Validation and Fusion algorithm is relevant compare to simpler methods.

Introduction:

NASA is focusing considerable effort to development Integrated Vehicle Health Management (IVHM) systems for the next generation of space vehicles [1]. IVHM is an advanced form of vehicle instrumentation system that provides the capability to process data instead of simply recording them. The primary purpose of these intelligent sensing systems is to increase safety and reliability while simultaneously reducing maintenance costs. This allows an onboard trend analysis that could detect system degradation and control in-flight systems checkout in addition to permit more efficient system servicing on the ground [2].

The requirements of IVHM have led to the adoption of an approach of monitoring based on a multi-agent system [3,4]. The recent advances in micro electro-mechanical systems technology, wireless communications and digital electronics have made possible the development of low-cost, low-power, multifunctional miniature smart sensors. These

sensors can be deployed throughout a region to build a network for measurement, detection and surveillance applications.

In this study, we propose to focus on centralized network where the data from the sensors converge to a central base doing the analysis and making the decisions. In order to compensate the low-reliability of the sensors, two different network strategies can be dissociated: the model-base network where the data from the nodes are fit into a model [5] and the non model-base network where the network uses the correlation between sensors readings to add some redundancy and ensure robustness. In the first case, the performances of the network will mostly depend on the accuracy of the model. The second case requires the use of an appropriate algorithm that can interpret the redundancy to make the approach robust against failure and to provide a sophisticated knowledge of the environment such as local and global information.

Event detection is one of the most promising applications of sensor networks where a large number of networked nodes are used to identify regions experiencing some particular phenomenon. This study presents a qualitative comparison between different centralized ways to detect and locate events in the case of redundant sensor networks.

This paper compares several methods regarding their ability to detect and locate local events, from the mass of sensors readings:

- Simple Model Fitting Interpolation
- Polynomial Regression
- Distributed Gaussian Method
- Mote Fuzzy Validation and Fusion Method

Here, the sensor network is applied to the surface temperature sensing of an aluminum plate and the local event is a local rise of the temperature, a hot spot. The plate represents the inside aluminum structure of a space vehicle and the hot spot a failure of the thermal protection system.

Ten different cases were generated with different placement of the heating resistance which produces the hot spot. Figure 1 illustrates the case of a network with 6x6 nodes placed in quadrant.
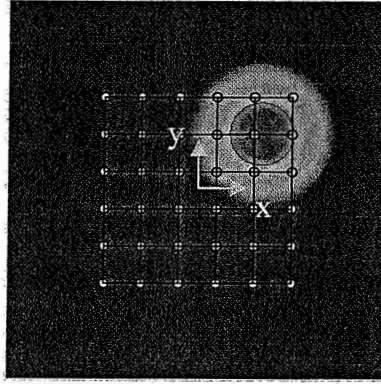
Figure1: Sensors Placement and Temperature Repartition in the Plate

Algorithms:

Model Fitting Interpolation

The method used here for the interpolation is the bicubic technique. It is one of the most common interpolation methods in two dimensions. With this method, the value of the function $f(x, y)$ at a point (x, y) is computed as a weighted average of the nearest sixteen nodes. It is composed of two basic cubic interpolations put together, one for each plane direction. The interpolation is calculated with the formula:

$$T_{Int}(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} \alpha_{ij} x^i y^j$$

The coefficients α_{ij} are computed by Matlab and depend on the interpolated data source properties.

Polynomial Regression

The polynomial used for the hot spot regression is a 4rth with the following shape:

$$T_{Reg}(x, y) = \sum_{i=0}^{4} \sum_{j=i}^{4} \beta_{ij} x^i y^{j-i}$$

The coefficients β_{ij} are computed by Matlab using the least square method in order to minimize the quantity $S = \sum_{i=1}^{n} \left(T(x_i, y_i) - \hat{T}(x_i, y_i) \right)^2$ at the nodes.

Where $\hat{T}(x_i, y_i)$ represents the real value of the temperature at the ith node and $T(x_i, y_i)$ the value given by the interpolation.

Distributed Gaussian Method

The idea of this method is to generate a Gaussian curve centered on each node and to do a normalized summation of all of them and find the maximum to detect the Temperature peak.

The Gaussian curve centered on the node *i* is:

$$G_i(x, y) = \hat{T}(x_i, y_i) \times e^{-\frac{(x_i - x)^2}{2\varepsilon^2}} \times e^{-\frac{(y_i - y)^2}{2\varepsilon^2}}$$

Where ε is the standard deviation of the distribution and can be tuned depending on the application.

In order to reduce the geometric effect of the node placement, the summation of the Gaussian function have to be normalized:

$$T_{Gau}(x, y) = \frac{\sum_{i=1}^{n} \hat{T}(x_i, y_i) \times e^{-\frac{(x_i - x)^2}{2\varepsilon^2}} \times e^{-\frac{(y_i - y)^2}{2\varepsilon^2}}}{\sum_{i=1}^{n} e^{-\frac{(x_i - x)^2}{2\varepsilon^2}} \times e^{-\frac{(y_i - y)^2}{2\varepsilon^2}}}$$

Mote Fuzzy Validation and Fusion Method (mote_fvf)

The Mote Fuzzy Validation and Fusion algorithm [6] was developed for wireless sensors network. It is able to distinguish between sensor failure and from environment abnormal behavior and to extract the relevant information from the mass of data of the sensor network. Methods for sensor validation and fusion based on fuzzy logic are unique as they do not require a mathematical model of the system. This algorithm uses the redundancy of the network to compensate the lack of reliability.

The network takes some redundant sensor readings and makes them go through three major steps: validation, fusion and prediction to come up with one single robust value. So far, this algorithm has been applied only to uniform fields where the fusion was done with all the sensors. In order to apply it to non-uniform field (hot spot) the fusion is done locally between few sensors located in restricted areas where the field can be assumed uniform [7]. For this application, one local fused value is generated by 3 sensor values as illustrate figure 2.
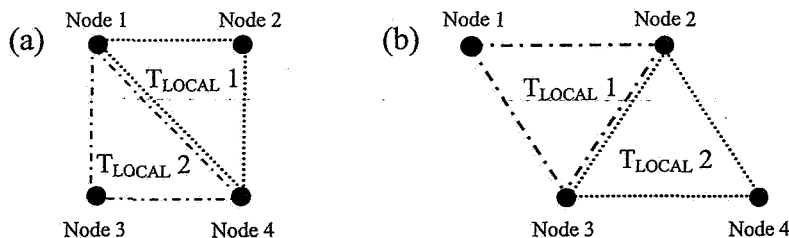

Figure 2: Fused Local Values for (a) Quadrant Repartition (b) Triangle Repartition

The validation part of the algorithm first filters obvious failures based on sensors physical limitation. Then it finds the medium of all the reading by a majority voting system and finally generates a dynamic validation curve in order to assign a confidence value $\sigma_i \in \begin{bmatrix} 0 & 1 \end{bmatrix}$ to the readings x_i. The center of the validation curve, where $\sigma = 1$, is a balance between the medium of the values and a prediction part.

The fusion consists on a weighted average of the values and their confidence values with include a fraction of the predicted value $\hat{x}$ to prevent the system from becoming unstable:

$$x_f = \frac{\sum_1^n x_i \sigma(x_i) + \dfrac{\alpha \hat{x}}{\omega}}{\sum_1^n \sigma(x_i) + \dfrac{\alpha}{\omega}}$$

The prediction part is an exponential weighted moving average time series predicting method.

Finally, the robust fused values obtained at the end of the process are interpolate to have the shape of the temperature field.

Comparison Method:

What to Compare?

In order to compare the algorithm, two cases are considered.

- In the first one, all the nodes in the network are undergoing some noise ($\pm 2.5\%$ and $\pm 5\%$ normally distributed)
- The second case analyses the effect of one failing sensor in different places in the network. (with a node value 10% and 15% higher or lower from the normal value)

For each of them, the methods are tested on their ability to locate precisely the temperature peak and to give a fair estimation of the peak value.

In the first case, 20 different nodes values are generated with random noise and the mean value of the peak location, the standard deviation of the location, the mean peak value and its standard deviation are compared.

In the second case, one and two failures are injected in the network and the peak location and value compared with the ideal case.

As the network performances are also depend on the geographical repartition of the nodes, the influence of the sensor network design and the number of nodes on the different algorithms is also compared.

Four network configurations illustrated figure 4 are studied, with a number of nodes of 16, 36, 64 and 100.
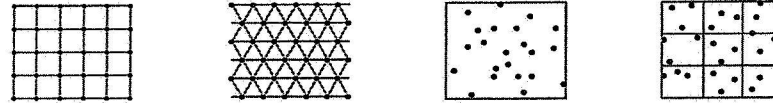
| Quadrants | Triangles | Random | Semi-Random |

Figure4: Network Geometries

In the semi-random configuration, the region is divided into small parts with the same number of nodes in each part to ensure a better coverage.

How to Compare?

In order to compare the algorithms, the output for each case and each algorithm is put in the same plots format shown figure 3.
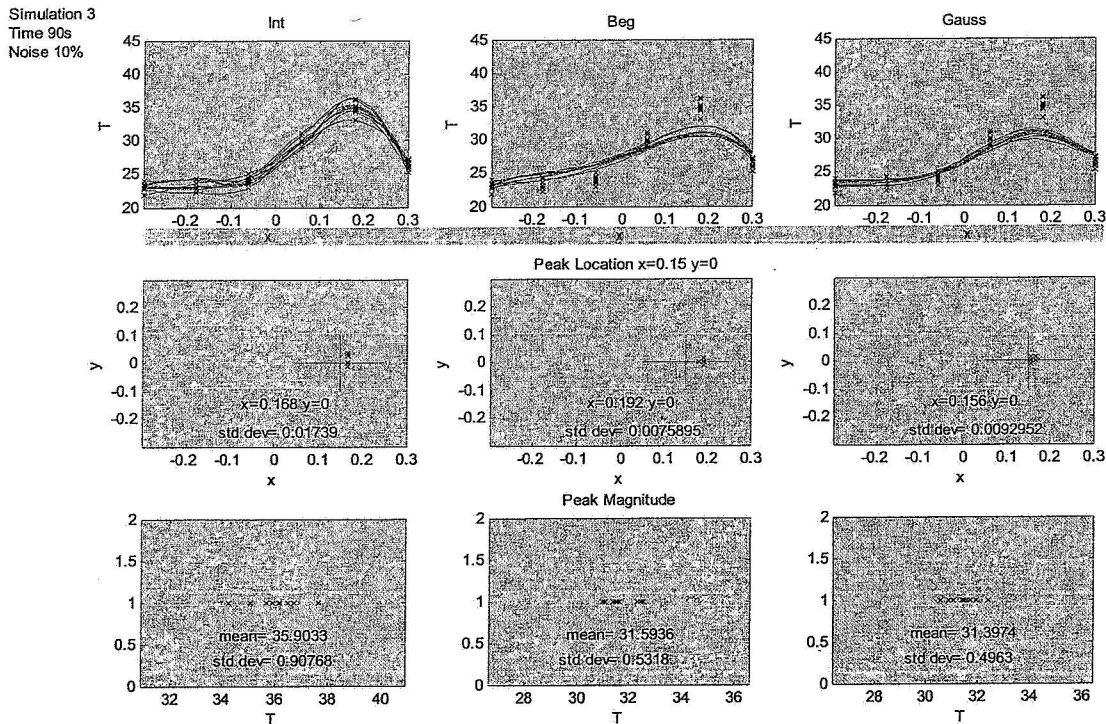

Figure 3: Plots Format used to Compare the Algorithms

Figure 3 represents the case of a 10% noise. Each column corresponds to a different algorithm. The first line is a slice of the temperature field generated by the different

methods. This slice is following the x direction and passing trough the peak. This allows seeing the shape of the field for 20 different noises. The black crosses are the values of the nodes. The second line gives the estimate location of the peak in the x/y plan for the different noises, the value of the average location and the standard deviation while the blue cross represents the real location of the peak. And finally the third line gives the variation of the value of the peak, the value of the average peak and the standard deviation.

Comparison Method

In order to explain the method used to compare the different algorithms for the different cases, this part focuses on the study of one particular network configuration, quadrant repartition with 6x6 nodes.

By processing the algorithms for 10 different hot spot location, it appears that the all the behaviors of the algorithms are a combination of 3 basic ones represented figure 5.

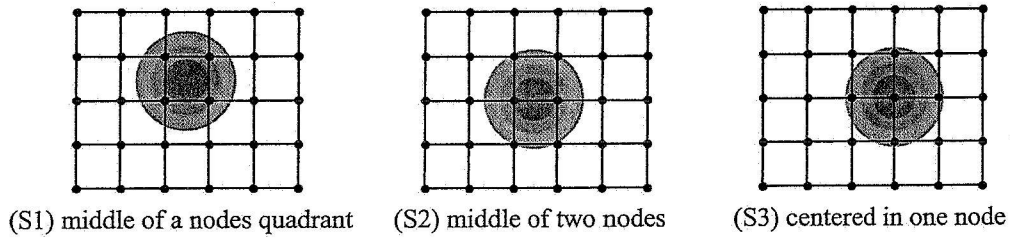(S1) middle of a nodes quadrant   (S2) middle of two nodes   (S3) centered in one node

Figure5: Basic Hot Spot Location

Regarding the number of all the different cases, a qualitative comparison able to have a better synthesis of the results and give more valuable information.

For the estimation of pick location, the variation of the location, the value of the peak and the variation of the value, the algorithms are graded from 5 for good estimation, to for poor. Afterward, the values of the different cases are summed to able a global comparison. A summary of the results is presented Table 1 & 2.

		S1				S2				S3			
		Peak Location		Peak Value		Peak Location		Peak Value		Peak Location		Peak Value	
		Center	Std V	Value	Std V	Center	Std V	Value	Std V	Center	Std V	Value	Std V
Interpolation	noise	5	2	5	1	5	2	5	1	5	5	5	1
	fault	1		0		1		0		4		0	
Regression	noise	5	4	2	4	5	4	1	4	5	4	0	4
	fault	3		2		2		1		3		2	
Gaussian	noise	5	3	2	3	5	4	1	3	5	5	0	3
	fault	3		2		2		1		3		2	
mote_fvf	noise	5	3	4	3	5	4	5	3	5	3	2	3
	fault	5		4		5		4		5		4	

Table 1: Comparison of the Algorithms Different Hot Spot Locations

Results				
	Peak Location		Peak Value	
	Center	Std V	Value	Std V
Interpolation noise	15	9	15	3
fault	6		0	
Regression noise	15	12	3	12
fault	8		5	
Gaussian noise	15	12	3	9
fault	8		5	
mote_fvf noise	15	10	11	9
fault	15		12	

Table 2: Results of the Algorithms Comparison

The results of this studies show that for the 6x6 quadrant network, the Mote_fvf is by far the most efficient algorithm especially in the case of faulty sensors.

The Interpolation is fairly working in the case of the noise, but is absolutely not fault tolerant. The Regression and Gaussian methods both have the same kind of behavior. They are much more fault-tolerant and have a good peak location but are really poor in giving an approximate value of it. It is understandable for the regression as the hot spot is a local event and as the approximation curve doesn't go through all the value of the nodes. The Mote_fvf gives fair results for noise but is also very fault tolerance. Nevertheless has got a real weakness in giving the peak value for the case S3. Indeed, the algorithm is taking the median of 3 nodes to compute the local values. So when the peak is centered on one node only, this node is among two other and its importance is weakened.

Comparison of Different Network Arrangements:

The methodology previously presented is applied to all the different network designs.
First, let's focus on the effects of the noise on the hot spot detection. The results of the comparison are presented in a form of a matrix of charts Figure 6. Each line represents a different configuration and each column a different algorithm.

Each plot of the matrix represents the evolution of the performance parameters with the number of nodes in the network. The blue line is peak value performance, the red one is the standard deviation of the peak location and the black line represents the location performances.

The peak value variation is not represented as it appears that it almost independent of the geometry of the network. For instance, for a normal distributed noise with a 5% variance, the values has a variance of

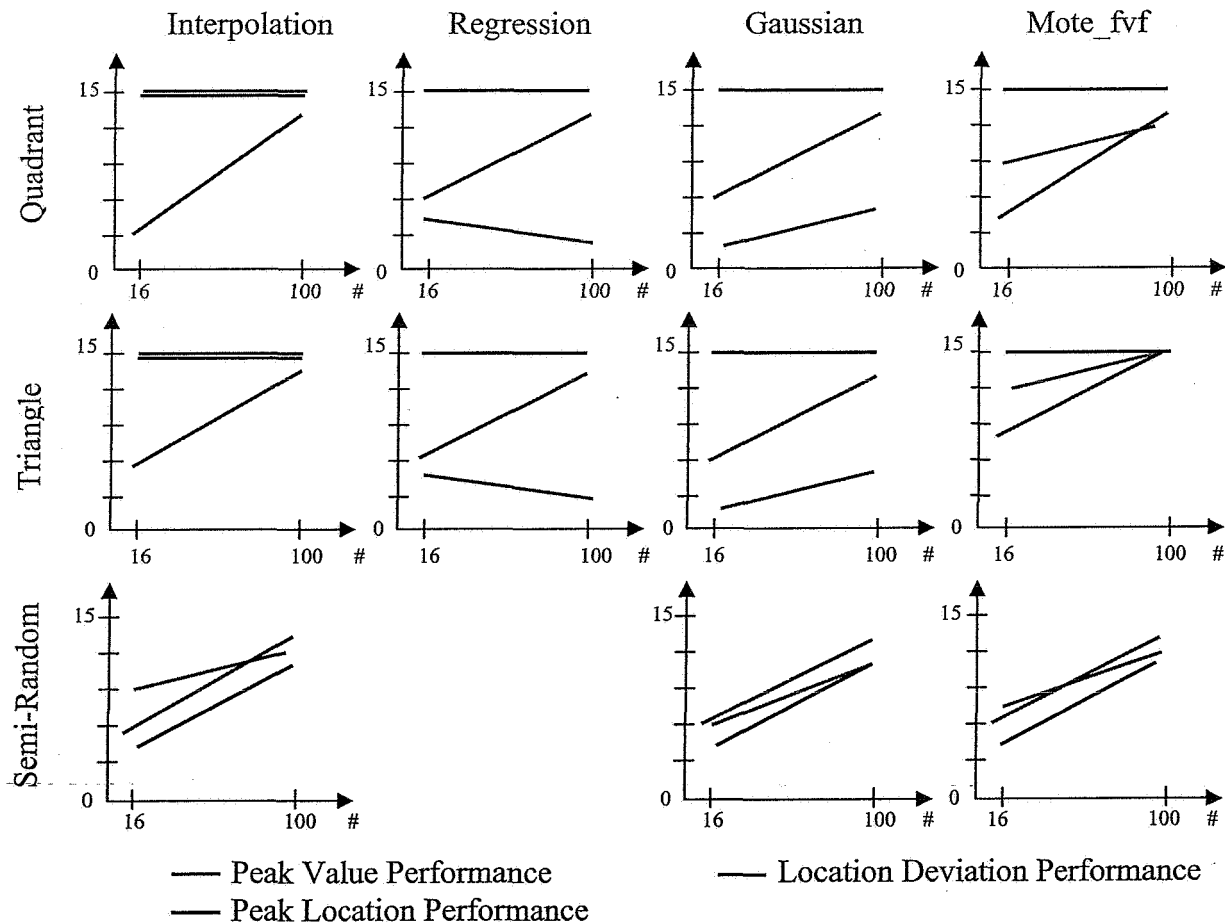|	Interpolation	Regression	Gaussian	Mote_fvf
5% noise	4.08%	0.67%	1.30%	1.52%

Figure 6: Comparison of hot spot detection under noise

Observations

The results are showing that for all cases, the augmentation of the numbers of nodes is a benefit for the performances except for the Peak Value estimation from the Regression algorithm.

The Random placement of sensors gives absurd values. The Semi-Random placement gives a fair estimation of the values except for the Regression algorithm. Indeed, if an area is poor in nodes then the behavior of the algorithm in this area becomes out of control.

The repartition in triangle an quadrant don't change much for the Regression and the Gaussian but it improves the location deviation of the Interpolation and Mote_fvf, as well as the peak value of the Mote_fvf algorithm.

As far as the fault tolerance is concerned, the change of network design doesn't influence the 3 first algorithms. Nevertheless, the Triangle repartition improves a bit the performances of the Mote_fvf, as 3 nodes are covering a smaller area than in the quadrant repartition.

As far as fault tolerance is concerned, the observations are the same as for the case study in the previous part. The network configuration almost doesn't change the performances of the interpolation, regression and Gaussian algorithms. But the triangle repartition improves just a bit the Mote_fvf compared to the Quadrant. Moreover, the augmentation of the number of nodes has also a great benefit on the Mote_fvf.

Moreover, the Mote_fvf performs the same way wherever the faulty sensor is located whereas other algorithm performances depend a lot on where is the fault.

Recommendation

The Mote Fuzzy Validation and Fusion algorithm seems to be the more performing than the 3 other algorithms, especially for the fault tolerance.

But its performances are as more meaningful as the redundancy is relevant in the network. Then, the Mote_fvf algorithm is able to isolate false data and distinguishing

system failures from sensor failures. But it requires more computational power than the other ones.

If this is an issue then depending on the number of nodes, the Distributed Gaussian (for higher number) or the Regression (for lower number) are a good compromise.

References

[1] NASA – Ames Research Center- Integrated Vehicle Health Management (IVHM): http://www.nasa.gov/centers/ames/research/humaninspace/humansinspace-ivhm.html

[2] D. Abbott, et.al. 2002. "Development and Evaluation of Sensor Concepts for Ageless Aerospace Vehicles: Development of Concepts for an Intelligent Sensing System", NASA Report no. NASA/CR-2002-211773, http://techreports.larc.nasa.gov/ltrs/PDF/2002/cr/NASA-2002-cr211773.pdf

[3] M. Hedley, M. Johnson, C. Lewis, D. Carpenter, H. Lovatt, D. Price. "Smart Sensor Network for Space Vehicle Monitoring". In Proceedings of the International Signal Processing Conference, Dallas TX, March 2003.

[4] D.C. Price et.al. "An Integrated Health Monitoring System for an Ageless Aerospace Vehicle". In Proceedings of the International Workshop on Structural Health Monitoring, Stanford, September 2003.

[5] T.J. Johnson et al., "Distributed structural health monitoring with a smart sensor array", Mechanical Systems and Signal Processing 18 (2004) pp.555–572

[6] Y.-J Wen, A.M. Agogino and K. Goebel, "Fuzzy Validation and Fusion for Wireless Sensor Networks", Proceedings of the ASME International Mechanical Engineering Congress,Anaheim, CA, 2004.

[7] J. Sauvageon, "Integrated Systems Health Monitoring Using Smart Dust Mote Sensor Networks: Hot Spot and Peak Strain Detection in Space Vehicles", Master Report 2005, UC Berkeley