Embedded microcontroller networks: Acoustic materials health monitoring

Patrick M. Rye*
University of California, San Diego, 9450 Gilman Drive 0416, La Jolla, CA,USA 92093-0416

ABSTRACT

Fiber reinforced polymer matrix (FRP) composites have a rich history of diagnosis and characterized using acoustic emissions techniques. The highly dispersive, attenuating, and anisotropic nature of unidirectional composites places an emphasis on high density local sensing as opposed to low density more-global sensing strategies.

A high density of sensors naturally implies large quantities of data requiring large bandwidth and substantial processing power. By distributing processing with the sensors themselves results in a decreased demand for bandwidth and lower computational power needed at each node in what is now a parallel processing computer. Desired information, time constraints and mechanical considerations place both hard and soft constraints on our network helping to define its architecture. I will present investigated computing architectures and their benefits and limitations as they relate to the various constraints involved.

1. INTRODUCTION

1.1 Network Considerations

My intent in this paper is to layout the considerations that have been taken in designing a network for embedded sensing applications. While we initially attempted to tackle the engineering problem in a hit and miss method of network design, the consideration of several questions, originally outlined by J. Crichlow[1] with some modifications appropriate for our application, directed us towards some suitable networks.

- Which computer network best suits the dominant computational algorithm and/or function to be executed on the computer?
- If the network is designed with a particular algorithm in mind, how easily can other computational tasks be performed?
- How easily can expansion be accommodated?
- How reliable does the network need to be?
- What is the level of complexity in the hardware implementation?
- What will the network cost to implement?

These questions in turn, require us to ask questions of our own project. For example, costs are not only the financial costs associated with components, fabrication and development but also include the penalty to the mechanical fidelity of the FRP composite that is inherent to the inclusion of a dissimilar material. Of these costs, the most important to us is the penalty to the mechanical fidelity of the composite. While the specifics of the effects of network and sensor inclusions are of utmost importance, that is not the purpose of this paper and will not be discussed here.

1.2 Progress

Implementation and analysis of our Networks of interest will be discussed here.

* prye@ucsd.edu; phone 1 858 534-2027; fax 1 858 534-2727; ceam.ucsd.edu

Sensor Systems and Networks: Phenomena, Technology, and Applications for NDE and Health Monitoring 2007, edited by Kara J. Peters, Proc. Of SPIE Vol. 6530, 65301F, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.715213

2. NETWORK CONSIDERATIONS

2.1 Best suited computer network for our algorithm

This is best answered by starting with the question of what algorithm is most appropriate for our application. If we assume that data collected has a spatial and temporal component then one needs to determine which of these play an important part in determining the health of the material.

Let's begin with the assumption that single location temporal changes in our measurements are of greater significance to us than a spatial mapping. We can then assume that with each PE collecting data from an equal set of sensors, the memory load would be roughly balanced. Further, since each PE has all the data it needs to make its evaluation, no communication is necessary to redistribute data. Finally, if each PE is collecting the same data, each should be performing the same task and the processing load should be roughly balanced as well. This leaves little communication necessary on the network except for some occasional administration such as polling and synchronization. This makes a global bus strategy very attractive – see Figure 1.

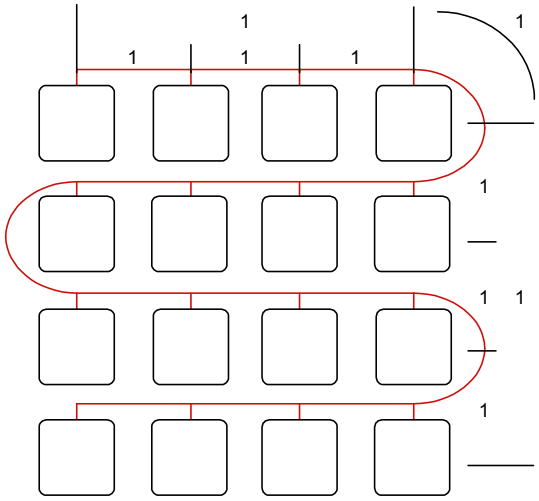

Figure 1. Global Bus network. Similar to the mesh network in that all PEs are peers, but each PE is connected to every other PE via a common bus, as opposed to individual connections.

If we make the contrary assumption that the network is acquiring the same type of data – e.g. strain – through out the network, and that said data can be meaningfully represented by an image – be it one dimensional, 2 dimensional, etc., then image processing algorithms are a good first guess; but are they applicable? With edge finding, pattern recognition, feature extraction, and multi-scale frequency analysis, the toolkit is well developed and available for us to use in this application. We can now address our initial question posed to us.

If we consider edge finding as applied to the plane of our panel, differentiation is required across the two special dimensions x and y. Differentiation is approximated by the weighted summation of a point with its two neighbors – see equations (1) & (2). Thus to efficiently implement this algorithm, the general mesh (Figure 2), pyramid and similar grid like networks would work well. Tree-type networks (Figure 3) – where siblings or peers must pass communications through a parent node – would be less efficient precisely because nearest neighbor processing elements (PE) are unable to directly communicate with each other – via message passing (MP) or a shared memory module.

$$I'(x_i) = \left(-\tfrac{1}{2}\right)I(x_{i-1}) + (0)I(x_i) + \left(+\tfrac{1}{2}\right)I(x_{i+1}) \tag{1}$$

$$I''(x_i) = (-1)I(x_{i-1}) + (2)I(x_i) + (-1)I(x_{i+1}) \tag{2}$$

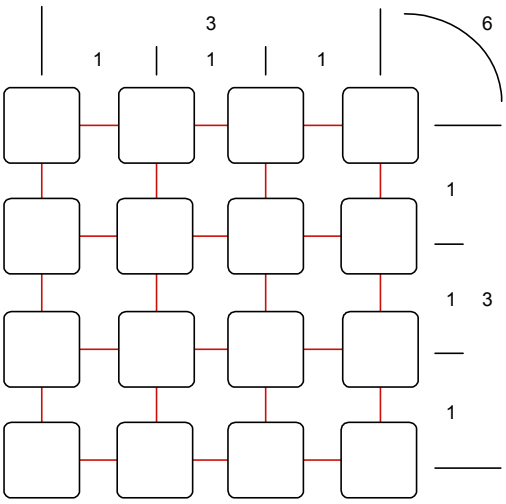

Figure 2. Nearest-neighbor mesh network. A PE's communication is restricted to its neast neighbors only. For communication with other nodes, a message passing protocol must be developed. The numbers represent the effective network distance, thus to pass a message from the PE in the upper left to the PE in the bottom right takes 6 steps. This newtwork can easily be improved by connecting the first and last PE in each row and column making it a torus network.

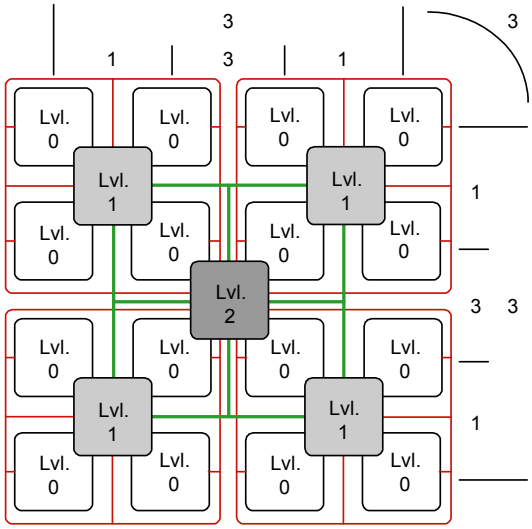

Figure 3. 4-Tree Network. In this implementation of a tree, each node – minus the root (Lvl. 2) – has a parent and three siblings. As implemented here, a bus is shared among siblings of the same parent, allowing direct communication with one another. Communication with a "cousin" is performed via message passing through parent nodes. The pyramid network is related to the tree, but allows communication amongst peers in the network with out parents acting as intermediaries – a combination of the mesh and tree networks.

The choice of optimal network is not always clear from algorithm alone. A good example of this is found by exploring the FFT. This can be very crudely explained as the appropriately sorted addition and subtraction even and odd symmetric permutations of all the elements in a vector of data as seen in Table 1[2].

Table 1. Arithmetic arguments for the FFT on a data vector of size 8. The numbers represent the vector indices and the operators are assumed to be operating on each index consecutively. (e.g. `+-+-` equates to `+0-1+2-3...`)

f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)
`++++++++`	`++++----`	`++--++--`	`++----++`	`+-+-+-+-`	`+-+--+-+`	`+--++--+`	`+--+-++-`

Since each processing element (PE) needs to calculate the weighted sum of the values from all the other elements in its vector some obvious options are that each data vector – rows and columns of PEs in the physical sensing – should have a short network distance. This could be accomplished by a bus connecting each row and column to each other (message passing scheme see Figure 4) or some shared memory scheme for the PE vector – crossbar, busses or multi-port memory.

On implementing a Row-Column bus-connected mesh network, the most time efficient way (assuming only modest memory constraints) is passing each PE's data to all other nodes and have them compute their respective sums simultaneously. For 16 bit data points would require a minimum of 16 bytes of memory at each PE and 16 bytes of inter-processor data transmition (excluding headers, commands, acknowledges, checksums, etc...). If memory were more tightly constrained and the network larger or the data resolution greater then this might not be feasible.

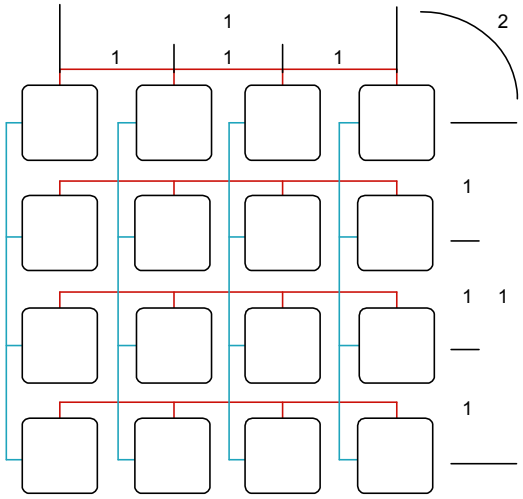

Figure 4. Row-Column Bus Array. Each row and column has its own bus to pass data. This significantly speeds up average communication over global bus network (Figure 1) by allowing simultaneous communications over the network while only increasing the network distance to 2.

A second approach would be to create a tree network. In this case I will assume a binary tree where the i^{th} and $(i+4)^{th}$ PEs are children of the same parent. Using a radix-2 implementation of the FFT one can reduce the memory costs (again assuming a 16 bit data point) for PEs of Lvl. 0 to be 2 bytes, or an 8 fold reduction in the memory requirement. PEs at Lvl. 1 would spend 4 bytes of memory and those PEs occupying Lvl. 2 of the tree would spend 8 bytes of memory during the calculation. That sums to a total of 48 bytes of memory spread throughout the network as a minimum requirement for the calculation. The communication costs are significantly higher, equivalent to that of 40 bytes of data passed serially – once again excluding communications overhead, but also excluding simultaneous communications occurring on different branches of the tree.

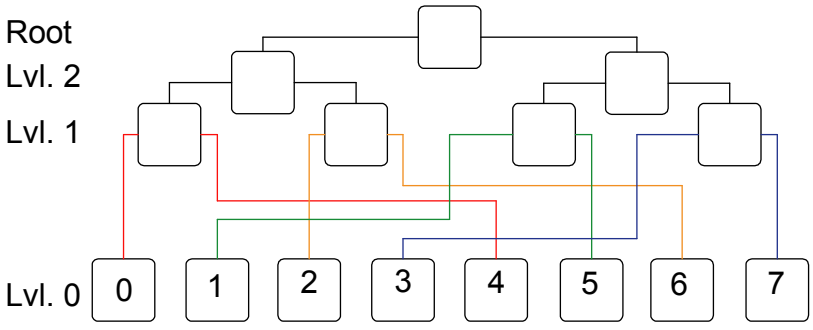

Figure 5. FFT Optimized Tree network

2.2 Highly tailored network and other applications

Consider the situation first outlined in section 2.1 where the temporal characterization of data is of more interest than spatial characterization. If we take the recommendation given of a simply interconnected bus network, the implications for spatial analysis are pretty dire for a reasonably sized network. Either a lot of time must be sent transmitting data – any single PE talking on the bus prevents any other communication throughout the entire network – or a lot of memory is required in each processing element for a global broadcast data smorgasbord.

Consider now using the Row-Column network mentioned in the discussion of the FFT. While memory usage is still pretty high, it should be less than a time optimized single bus network and the fact that many communications can happen in parallel will only increase the speed of any sort of spatial analysis.

2.3 Network size

In general network studies, one must consider the possibility of network expansion and the affect of network scaling on its efficiency. As the premise of this paper suggests, these networks are going to be embedded so while scalability is important in design adaptations for different embedding applications, once embedded the network is done.

With respect to design, it is probably of interest to state that star (cluster) networks, tree networks and most networks with a hierarchy are much more easily expandable, generally with only logarithmic costs to process times. Mesh and peer array type networks tend to grow at a linear or polynomial cost.

2.4 Redundancy and network reliability

The question of reliability should be a no brainier: the purpose of monitoring the health of a composite or any material used in a structure of vehicle. The whole purpose of the network is to improve the safety of the occupants and the investment (building, plane, etc.). So an unreliable network is useless.

Our group has taken two main approaches on this topic. The first is the generally accepted redundancy, rerouting algorithms[3]. The second approach is to simply let it fail...but make sure you know it failed.

Failure in a network can occur in various ways. Communication line breakage, failure of solder joints, program failure, buffer overflow, etc... If one is concerned about breaking copper traces – which really should not be a significant concern – simply make it wider, or duplicate the trace while making it electrically connected. If the first breaks, the second still exists. Program failures are unacceptable, but since their nonexistence cannot be guaranteed watchdog timers, reprogramming methods and (minimally) rerouting algorithms should be developed.

While striving to create a high availability network, or at least a fault tolerant one, is certainly a very good direction, one can always learn something from failure. If we assume that the network fails, that we know with in a reasonable time frame that it fails, and that the failure is not a result of communication conflicts, program bugs, then in the case of a network embedded in a composite material, its failure is indicative of reaching a minimum interlaminar shear stress. Assuming this minimum for breakage of trace or solder is of a magnitude to indicate problems within the composite...then you are done. Thus a network whose failure is indicative of the failure of the composite is itself is a technique for structural health monitoring...though no the one I would want to bet the farm on.

2.5 Network complexity and cost

These two considerations somewhat go hand in hand. From a purely monetary perspective the complexity generally adds to the cost of development, maintenance and likelihood of conflict and failure. Of course complexity is often necessary to solve a problem generally involving dissimilar technologies.

Mechanical costs are not so well defined. There can be quite complex methods of surface treatment of network components for greater matrix adhesion, complex embedding geometries and complex components. These complexities can improve the mechanical fidelity of the composite, whereas a network deemed complex by virtue of its number of components would most likely hurt the mechanical fidelity of the composite.

3. EFFORTS AND IMPLIMENTATION

3.1 Nearest-neighbor mesh network

At the beginning of our investigation we had little idea what out sensing strategy would be and thus determining a specific network was somewhat arbitrary as we could not address the first 2 questions laid out by Crichlow[1]. This left us wanting to design a network that was robust and minimally fault tolerant and would work well with edge finding algorithms. This with the notion that as little electronics as possible should be embedded and that all sensors should be of peer status got us working on a nearest neighbor mesh implementation.

Being limited on hardware implemented communications, software protocols were written. These all required strict synchronization across the network which can be problematic with large distances. Additionally, these protocols required rerouting algorithms in the case of halting and receive omission at an intermediary PE. This was designed to allow messages to arrive at their destination so long as network partition had not occurred – see Figure 6. The major problems with the NN mesh network were latency, timing and scalability. Latency was improved by shifting the network to a torus design and allowing communication to only flow in a single direction along any given column or row. By having alternating directions in the neighboring rows and columns, message delivery could almost be guaranteed – there could be incidences of infinite looping. Workarounds usually involved higher overhead which increased the latency somewhat and added significantly to the protocol governing code space.

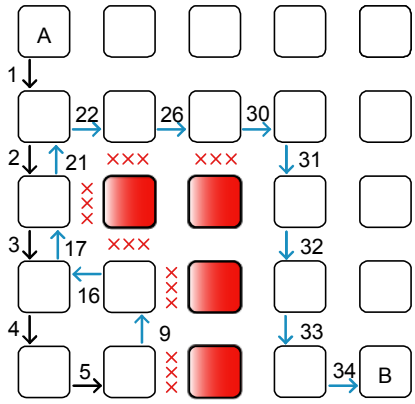

Figure 6. Example of developed message routing algorithm for our nearest neighbor mesh network.

3.2 4-Tree network

Due to the latency and timing issues of the NN mesh network and after viewing dismal performance of the FFT in our non-torus implementation we began work on the 4-tree network – the 2-D representation of which is shown in Figure 5. While receiving significant improvement with this algorithm, we also achieved a scalability that was previously prohibitively expensive in regards to latency especially. With the tree network communications times increased as log of base 2 of the lowest level network size.

The disadvantages were the extra number of PEs required due to the hierarchy scheme – increasing power consumption while decreasing mechanical integrity – and the inefficiency of implementing edge finding algorithms (they still occurred slightly faster than the NN mesh, but required more memory to implement). The lack of hardware implementation of communications in our microcontroller of choice – only 1 UART and 1 SMBus – make implementation of a pyramid network prohibitive but is something we have begun investigating with larger more fully loaded microcontrollers.

3.3 Global bus network

Our final strategy revolves around acoustic emissions sensing. Acoustic emissions are the characteristic sounds emitted when the matrix cracks, when de-bonding occurs, when fibers break, etc. While it is interesting to make spatial comparisons of AE metrics, the generally performed analysis is more analogous to temporal as opposed to spatial delineations in acoustic activity. Thus a distributed network connected by a single bus was our first network of choice (Figure 1). Tests of this network in practice are on going but look promising. Location detection is currently done quite crudely via first arrival techniques – the first sensor to sense the AE or failure is the assumed location of the AE or failure. Image processing techniques perform miserably if at all, and would perform better on a multi-bus system such as the row-column array network, but that is left for future investigation.

4. FUTURE WORK

Our future projects included the continued testing of these networks in use as well as development of networks using more sophisticated microcontrollers in hopes that smaller package versions will be produced in the coming years.

REFERANCES

[1] Crichlow, J. An Introduction to Distributed and Parallel Computing. Bodmin, Cornwall: Prentice Hall Europe 1997

[2] Kevin Loewke, David Meyer, Anthony Starr and Sia Nemat-Nasser, "Structural Health Monitoring Using FFT," *Proc. of SPIE* Vol. 5765 (2004)

[3] Birman, K. Reliable Distributed Systems. New York: Springer 2005