From "$_{\textit{Sensor}}$ *Networks*" to "*Sensor Networks*"

Sergio D. Servetto

School of Electrical and Computer Engineering
Cornell University
http://cn.ece.cornell.edu/

***Abstract*— In this position paper, we argue that the mainstream view in our community of what makes for problems in sensor networks that are worthy of attention is very lopsided, focusing much more on pure networking and other classical computer systems issues, than on the equally important but often neglected tasks of modeling of distributed signal fields, and on their acquisition, processing, synthesis and control. We show concrete examples of how these are essential aspects that need to be dealt with in any good system design, and we survey experimental work along these lines currently under way in our lab.**

I. Introduction: Playing Devil's Advocate

Let us imagine a hypothetical scenario for a moment, and see where some of those assumptions take us.

A. A Strange Digital Camera

Digital cameras use solid-state devices as image sensors, formed by large arrays of photosensitive diodes. During the short period of time in which the shutter is open, each diode records the intensity of the light that falls on it by accumulating a charge. This brightness measurement is then digitized and stored in memory, forming a picture. Imagine now that we had one particular digital camera with some rather unusual characteristics:

- Besides measuring brightness, diodes can also compute and communicate with each other.
- There is a central memory storage, where an image is formed based on the readings from all diodes.
- Communication among diodes and with the central storage happens over a wireless channel.

Such a digital camera is not as contrived an example as it may seem: this camera is actually a *wireless sensor network*, in which nodes observe portions of an image, and then need to cooperate so that a copy of the entire image can be delivered to central storage. That is, we view a sensor network as a pure data acquisition device, which incidentally happens to have to deal with some communication aspects during its operation.

In this view of sensor networks, clearly signal processing is "where the action is." Sampling and interpolation questions (how do we form a continuous image based on the information captured by each diode), filtering questions (how do we get rid of noise in the diode readings), data compression questions (how do we produce a compact representation of the image for storage), and signal detection and parameter estimation questions (how do we make inferences about the real world based on information collected by the diodes), all of these are the main technical tasks that go into the design of our digital camera. Sure, there are some annoying issues arising from the fact that nodes have to communicate under resource constraints. But that aspect can be dealt with using standard wireless networking tools; until we find good DSP algorithms to deal with the questions above, we should not have to worry about the wireless networking aspect of this problem.

Now, there is something clearly wrong with this picture, as anyone in this workshop would immediately recognize. And this author does most definitly *not* endorse such a view. But: what exactly is wrong with it? What is it that makes us automatically dismiss this view as nonsensical? Well, I would like to submit that it is the fact that it places such a strong emphasis on the pure DSP aspects of the problem, while "sweeping under the rug" many critical wireless networking issues that are central to any good sensor network system design. Yet, with a reversal of the roles of computer systems and wireless networks on one hand, and of distributed signal processing algorithms on the other, this is the exact kind of view that we hold of the field: one where the task of designing sensor networking systems is one of building wireless networks operating under tight power constraints, whose inputs happen to be signals picked up in the environment.

It is the main contention in this paper that both views (the hypothetical one above, and what this author believes is the mainstream view of the community) are flawed, for the exact same reason: their disproportionate emphasis on only one of these two important aspects. And that perhaps it is time to take a more holistic view of what it means to do systems work in sensor networks.

B. What Is In This Paper

This paper is organized as follows. In Section II we present a brief summary of two sensor networking systems we are currently developing in our lab: a spatial sound system, and a system to recover a description of the geometric structure of a room. In Section III, we address the time synchronization issue: we argue that only by having nodes manipulate physical signals directly (and not just exchanging packets of bits) is it going to be possible to overcome some scalability problems inherent in current proposals. In Section IV we give a description of the tools we have in our lab, and of those we are currently custom-developing, to enable our research work on these topics. Concluding remarks are offered in Section V.

II. Two Systems

In this section we review two problems currently being investigated in our lab. These are examples of problems that we believe to be predominantly *systems* problems, although not classical ones. We give here a high level description of

Work supported by the National Science Foundation, under awards CCR-0238271 (CAREER), CCR-0330059, and ANR-0325556.

what the problems are and what challenges they entail, to then justify our claim that these are inherently systems problems, rather than math problems, or array signal processing problems, or information theory problems, or ... etc.

A. *System I: Distributed Wave Field Synthesis (dWFS)*

(Joint work with Georgios N. Lilis – Cornell/ECE.)

1) Description of the Problem: Consider an old problem in acoustics: the synthesis of spatial sounds using arrays of speakers. Commercially available spatial sound systems exist already, using about 5-7 speakers (e.g., surround sound systems for home entertainment). But they require manual calibration, and their effectiveness is limited to a small area that is often referred to as the "sweet spot". If instead of using a few speakers we could use a few *hundred* of them, each one very small, and interconnected over a wireless network (to avoid the clutter derived from dealing with a few hundred cables), the size of the sweet spot could be grown to cover a much larger area. And we believe that having the ability to shape the wave field *almost everywhere* (and not just at isolated points in space) enables applications that are simply not possible with small sweet spots. For example, the creation of virtual environments, such as the meeting room of the Jedi council in the last Star Wars movie, as discussed in Fig. 1.

2) Systems Research Issues: Wave field synthesis is an example of a feedback control problem: we have a plant (the medium where waves occur), we have inputs (the acoustic signals generated by the speakers), we have observations (the acoustic signals measured by the microphones), and we have a controller that chooses inputs as a function of observed outputs to steer the system to a state where a desired output is produced. The design of such feedback control loops is a well understood problem, with many textbooks available on the subject (e.g., [1]).

The dWFS problem is closely related to feedback control problems, but it does *not* fit the classical setup very well. This is because the observer is not a single device that captures the entire signal field, and the controller is not a single device that acts on the entire signal field – instead, sensing and control are performed over a *network* of sensors and actuators. Hence, the performance of this network cannot be analyzed and understood independently of this context. This network has a purpose: to observe and to control a signal field. And that purpose can lead to performance metrics quite different from maximizing throughput, classical definitions of lifetime, etc., which in turn have significant impact on the design and operation of the network. These are classical systems issues, but now very heavily tied to the structure of the signals observed/controlled by the network. The impact of the structure of those signals on the entire network architecture is significant, and hence any good system design has to start with a careful understanding of what those signals are.

B. *System II: Can You Hear the Shape of a Room?*

(This name is chosen by analogy with a well known problem in applied mathematics – see [3]. Fortunately for us, it seems that in our incarnation of the problem there is a bit more hope! Joint work with Mingbo Zhao – Cornell/ECE.)

1) Description of the Problem: The solution of the dWFS problem above (that is, the determination of what signals should be generated by each node in the network of speakers), requires knowledge of a number of parameters in the system. Some of these parameters can be assumed to be constant (e.g., the speed of sound propagation in a room), or if not can be estimated with relatively little effort (e.g., the reverberation characteristics of a stationary room). Others are challenging and most interesting problems in themselves: e.g., the recovery of a geometric description of the room, which is the main focus of this section. At every point in time, the signals generated by each speaker in the dWFS problem will clearly depend on the location of potentially moving objects, etc. Thus, the ability to form a real-time map of the environment is a fundamental component of our dWFS system.

There are multiple alternatives based on which to attempt building such a real time map. After careful evaluation of a number of options, and based on robustness and practicality considerations, in our work we have settled on a simple, two-step approach, that takes advantage of the fact that we have a large number of speakers and microphones in our system:

- In a first "low-level" estimation step, we seek to recover the physical location of all the microphones in the room.
- In a subsequent "high-level" estimation step, we use those known locations to select the most likely geometric boundary that is consistent with those measurements, according to a probability distribution we impose on the space of possible boundaries which captures our a-priori knowledge of structural constraints.

An illustration of this approach is shown in Fig. 2.

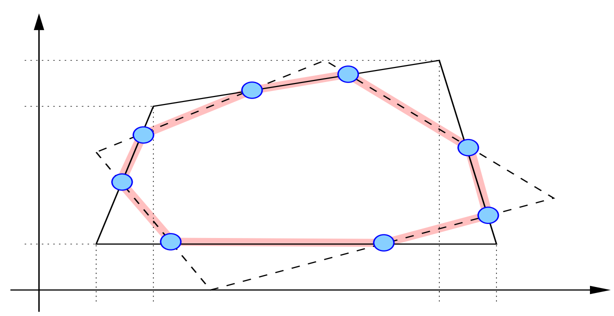

Fig. 2. Illustration for a 2D polygon. Thick black lines denote the real shape we wish to recover, solid circles denote microphones, lighter lines denote the convex hull of the microphone locations, dashed lines denote additional "ghost" faces. The challenge here is, among multiple candidates, and in the presence of noise in the estimate of location of each microphone, to identify the real faces of the polygon.

We note how some might argue that putting nodes on the boundary trivializes the problem – so we hasten to point out that in many practical situations, the boundary is the *only* place where it makes sense to place nodes: one can certainly not keep microphones floating in the air! But then, once nodes are there, and better yet, many of them, it would be foolish to not exploit that fact to recover an accurate geometric descriptor of the boundary. And that is exactly what we do.

2) Systems Research Issues: Estimation of signals constrained by geometric laws is a relatively unexplored area of digital signal processing, for which we have a few good pure

Fig. 1. A meeting of the Jedi council in "Star Wars III: Revenge of the Sith" – the holographic images correspond to council members attending the meeting from a remote location. If we are ever to realistically render a person moving freely around a room as part of a virtual reality system, it will be necessary to have the ability to place a synthetic sound source at any arbitrary location in the room where we expect that person to move. This is possible with array technology (in this case a speaker array), but the number of speakers needed to enable a realistic rendering of a virtual acoustic source is many orders of magnitude higher than anything available today. We believe however that a very large number of micro-speakers, equipped with wireless communication capabilities, can do the trick.

DSP research questions under study at the moment – e.g., no such thing as a "sampling theorem for geometric signals" is known yet. But our two-stage approach to solving this problem does raise a number of important *systems* issues as well.

The geometry identification problem is also closely related to but different from more classical problems in signal detection and parameter estimation, and again, the difference is given by the distributed sensing aspects. Important systems issues that come up in the design of a sensor network for geometry identification purposes are the following:

- How many nodes do we deploy? Forming what topology?
- How do we calibrate and synchronize the network?
- How is the task of computing estimates of locations from signals exchanged by the nodes to be distributed among the nodes and the central controller in the dWFS system?

These are all classical systems issues. But to be able to answer them, first we have to decide on what signals are going to be exchanged among nodes (in our system, acoustic pulses), and on what features are going to be extracted from those signals (in our system, timestamps of arrivals, which in turn creates a need for accurate time synchronization). Only then can we proceed to answer issues of calibration, synchronization needs, topology formation, and distributed computations.

III. Scalability of Time Synchronization

(Joint work with An-swol Hu – Cornell/ECE.)

A. Description of the Problem

The problem of time synchronization in large distributed systems consists of giving all the physically disjoint elements of the system a common time scale on which to operate. This common time scale is usually achieved by periodically synchronizing the clock at each element to an ideal time source, so that the local time seen by each element of the system is approximately the same. Many techniques have been proposed for synchronization in sensor networks [2], [7], [9]. These techniques all rely on nodes exchanging packets with timing information. Using the exchanged timing information, each node can then estimate clock offset and maybe clock skew. However, all of these traditional methods suffer from an inherent scalability problem, as synchronization errors accumulate over multiple hops.

Consider the situation illustrated in Fig. 3. Let us assume

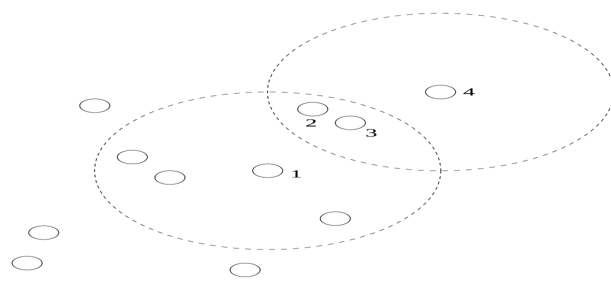

Fig. 3. Assume node 1 has the reference time and all nodes want to be synchronized to this clock. Nodes 2 and 3 are in the broadcast domain of node 1. However, node 4 must be synchronized through nodes 2 and 3.

that we wish to establish a global time scale for this network where node 1 contains the reference time. Nodes 2 and 3 can estimate its synchronization parameters, i.e., clock skew and offset, directly from node 1. This means that nodes 2 and 3 will have an estimate of the clock of node 1, but recall that the estimates will have errors. Node 4, on the other hand, is outside the broadcast domain of node 1 so it must estimate its clock skew and offset relative to node 1 through the use of timing information from nodes 2 and 3. However, since the timing information provided by nodes 2 and 3 regarding the clock of node 1 will have errors, the skew and offset estimates made by node 4 will be further corrupted by timing errors.

This error accumulation process over multiple hops poses a problem, as networks are deployed over increasingly large areas. As the number of hops required for communication increases, so does the synchronization error. To the best of our knowledge, the synchronization performance of all existing algorithms decreases over an increasing number of hops.

B. From Synchronization Bits to Synchronization Waveforms

Based on the current state of the art, it seems that algorithmic methods are not going to be enough to achieve truly scalable time synchronization – current timesync mechanisms do not scale well, and at least to this author it is not clear what could be done to improve the performance of timesync on motes in any substantial way beyond, say, that achieved in [7]. An alternative approach however, developed mathematically in [5], and whose experimental validation is currently under way in our lab [4], would have us deal with the scalability issue by means of a synchronization method in which nodes do not exchange information in the form of packets with bits, but instead by directly manipulating electromagnetic waveforms. A pictorial representation of how synchronization could be achieved is shown in Fig. 4.

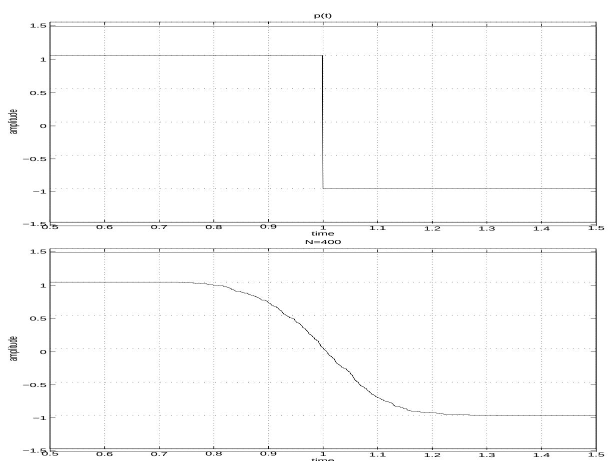

Fig. 4. Assume N square waves are transmitted (one by each node) at random times, and assume the distribution of these random times can be somewhat controlled, to at least make them have all the same mean, and a small variance compared to the duration of the wave. Then, under the assumption of N large, it follows from the Law of Large Numbers that the observed signal is going to be a smoothed version of the square wave, in which the center *zero-crossing* will correspond to the location of the mean of the random times. Thus, an increase in the number of nodes results in the randomness of each individual node being averaged out, and thus in an increase in synchronization performance.

For such a method, whose key distinguishing feature is that it directly manipulates electromagnetic waves exchanged by nodes in the network (as opposed to encoding bits into these waves, and using the information in those bits to synchronize), it was proved in [5] that synchronization can be maintained indefinitely, with an error that actually *decreases* as the network gets larger. Again, as in the systems discussed in Section II, it is by working directly on physical signals that we can achieve synchronization performance capable of scaling to very large and dense networks.

IV. Sensor Networks ≡ Distributed DSP Machines

A. Inferences from Previous Examples

What is the common to all previous three examples? The fact that in these systems, the signals observed, produced, and acted on by the sensor network are central to the operation of the network itself. To design networks of sensors and actuators capable of shaping a sound field or identifying a geometry, a key step, a fundamental one, one that simply cannot be ignored, is the development of an understanding of what these signals are [6], [11]. To synchronize a large network, exchanging bits it appears is not enough, and it became necessary to design suitable electromagnetic signatures for nodes to radiate, and to take into account how those signals interact in the air [4], [5]. And we have two more examples under study, not elaborated on here due to lack of space: suitable modeling of the signals observed by the network leading to a characterization of traffic loads [8], and a modular architecture for data archiving and analysis applications [10].

The key point we are trying to make is this: in all these examples, the system design task simply *cannot* be decoupled from the task of analyzing the structure of the signals observed and acted on by the network being designed. And we believe this is not just an aberration observed in our examples, but that it truly is a statement about the nature of the problem: *there is much more to the system design task in sensor networks than those aspects captured by trying to make a system work on Berkeley motes*. Much of our own work of an experimental nature is therefore focused on the analysis and control of distributed signals, and on the development of hardware tools to enable that work. For that purpose we have established a laboratory at Cornell, described next.

B. Distributed DSP Machines in our Lab

To pursue our goals in sensor networking research we have established a lab whose main purpose is to experiment with the capture and synthesis of high resolution distributed signal fields. We have two pieces of equipment:

- A custom-built large scale acoustic array.
- A home-made array of hammers and strain sensors.

Pictures of the equipment are shown in Figs. 5 and 6, with explanations in their captions.

We note that a likely objection to be raised to this work is that "this is DSP, this is not really sensor networks; where are the radio, the power constraints, the memory limitations, etc?" Yet we hope, through the examples and arguments offered in this work, to have made at least a prima-facie case (to be defended in the presentation at the workshop) that whereas all those certainly are very important issues, they are no more and no less important than the signal analysis and control tasks on which work in our lab is focused.

V. Conclusions

We hope to have convinced the reader that, in the context of sensor networks, signal processing tasks should be considered on equal footing with any other classical network task such as routing, flow control, power management, etc. That is currently

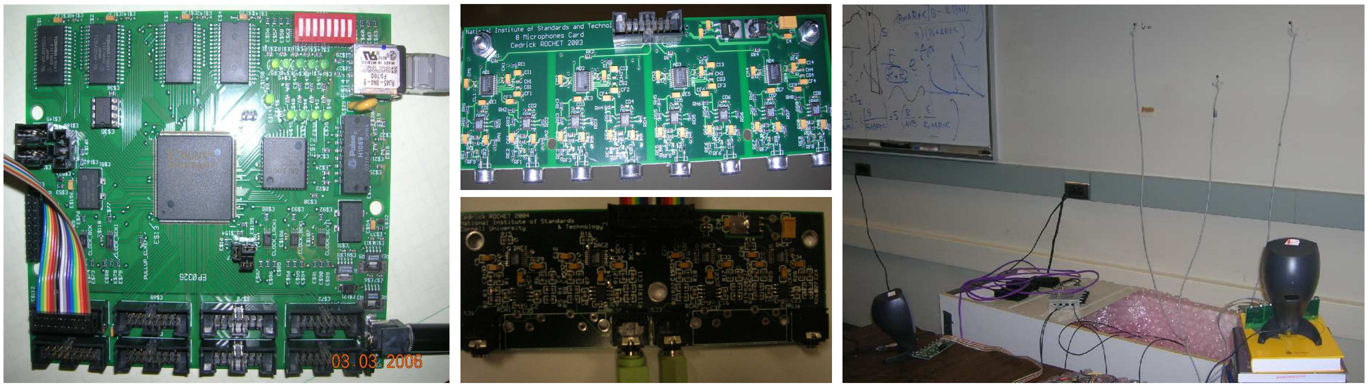

Fig. 5. A large acoustic array, consisting of 256 microphones and 64 speakers, custom developed by Cedrick Rochet, at NIST. A motherboard controls groups of 8 childboards at a time, provides an ethernet interface, and power (left). Each childboard consists of 8 microphones (top center), or 4 stereo jacks (bottom center). The motherboard communicates with device drivers in a host computer running Linux, which provide a simple API to develop C code. Our goal is to effectively demonstrate the placement of a virtual acoustic source in the center of our lab room, by means of implementing a feedback control loop based on *distributed* DSP algorithms for speaker control and for the recovery of room shapes (microphones on the wall, right).

Fig. 6. An array of 24 digital hammers, developed by PhD candidate Georgios N. Lilis, MEng student Sining Qi and undergraduate student Jae-Hoon Lee. Each hammer (left) consists of a piece of metal with a coil wrapped around it. When a current is passed through the coil, a magnetic field is created that causes the hammer to displace from its rest position; a simple circuit with a digital interface controls the current flow from an external power source. A simple demo of of this tool is a musical instrument: each hammer hits a pipe of different length, thus producing different sounds (middle). Our goal however is to demonstrate another application of the dWFS concept: here, in conjunction with strain sensors (right), we can synthesize small pressure waves inside an object by having the hammers tap on its surface, and then map these waves at other surface points using strain sensors, to create a tomographic picture of the object's interior without destroying it.

not the case in our community. So, if this presentation can at least plant a seed of doubt and generate some debate in this regard, the main goal of the author in writing this paper will have been accomplished. Feedback is most certainly welcome.

Acknowledgements–The author wishes to acknowledge his collaborators in the various projects mentioned in this paper: Cornell/ECE doctoral candidates An-swol Hu, Georgios N. Lilis, Mingbo Zhao; Cornell/CS doctoral candidate Yong Yao; Prof. J. Gehrke of Cornell/CS; and Prof. Joe Rosenblatt of UIUC/Math. Multiple interactions with these colleagues over long periods of time are the basis on which the author formed the opinions expressed in this writeup. Of course nothing is implied here about their views, and any outrageous mistakes that the author might have included in these pages certainly remain his sole responsibility.

REFERENCES

[1] C.-T. Chen. *Linear System Theory and Design (2nd ed)*. Harcourt Brace College Publishers, 1984.

[2] J. Elson, L. Girod, and D. Estrin. Fine-Grained Network Time Syncrhonization using Reference Broadcasts. In *Proc. 5th Symp. Op. Syst. Design Implementation (OSDI)*, Boston, MA, 2002.

[3] C. Gordon, D. L. Webb, and S. Wolpert. One Cannot Hear the Shape of a Drum. *Bulletin of the American Mathematical Society*, 27(1):134–138, 1992. See also http://www.ams.org/featurecolumn/archive/199706.html.

[4] A. Hu and S. D. Servetto. A Scalable Technique for Cooperative Time Synchronization. Submitted for publication.

[5] A. Hu and S. D. Servetto. On the Scalability of Cooperative Time Synchronization in Pulse-Connected Networks. IEEE Trans. Inform. Theory, to appear. Available from http://cn.ece.cornell.edu/.

[6] G. N. Lilis and S. D. Servetto. dWFS: *Distributed* Wave Field Synthesis. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Proc. (ICASSP)*, Toulouse, France, 2006.

[7] M. Maróti, B. Kusi, G. Simon, and A. Lédeczi. The Flooding Time Synchronization Protocol. In *Proc. ACM SenSys*, 2004.

[8] S. D. Servetto and J. M. Rosenblatt. The Multiterminal Source Coding Problem for Spatial Waves. In *Proc. UCSD Wkshp. Inform. Theory App.*, San Diego, CA, 2006. *Invited paper*.

[9] G. Werner-Allen, G. Tewari, A. Patel, and M. Welsh R. Nagpal. Firefly-Inspired Sensor Network Synchronicity with Realistic Radio Effects. In *Proc. ACM SenSys*, 2005.

[10] Y. Yao, J. Gehrke, and S. D. Servetto. A Modular Architecture for Data Archiving and Analysis Applications in Sensor Networks (tentative title). In preparation.

[11] M. Zhao and S. D. Servetto. Distributed Detection of Geometric Signals (tentative title). In preparation.